The LaTeX2HTML Translator

Nikos Drakos,
Computer Based Learning Unit,
University of Leeds.

March 30, 1997

This document accompanies LaTeX2HTML Version 96.1[1]

Abstract

LaTeX2HTML is a conversion tool that allows documents written in LaTeX to become part of the WorldWide Web. In addition, it offers an easy migration path towards authoring complex hypermedia documents using familiar word-processing concepts.

LaTeX2HTML replicates the basic structure of a LaTeX document as a set of interconnected HTML files which can be explored using automatically generated navigation panels. The cross-references, citations, footnotes, the table of contents and the lists of figures and tables, are also translated into hypertext links. Formatting information which has equivalent "tags" in HTML (lists, quotes, paragraph breaks, type styles, etc.) is also converted appropriately. The remaining heavily formatted items such as mathematical equations, pictures or tables are converted to images which are placed automatically at the correct positions in the final HTML document.

LaTeX2HTML extends LaTeX by supporting arbitrary hypertext links and symbolic cross-references between evolving remote documents. It also allows the specification of *conditional text* and the inclusion of raw HTML commands. These hypermedia extensions to LaTeX are available as new commands and environments from within a LaTeX document.

This document presents the main features of LaTeX2HTML, and describes how to obtain, install, and use it.

---

[1] This manuscript was updated to version 96.1 as indicated by the changebars by Herbert W. Swan `<dprhws@edp.Arco.com>` and converted to LaTeX 2_ε by Michel Goossens `<goossens@cern.ch>`.

Contents

---

[2] This feature is supported only for users of LaTeX 2_ε.

List of Figures

List of Tables

---

[3] http://csep1.phy.ornl.gov/csep.html

1 Overview

This document describes the LaTeX2`HTML` translator which:

- breaks up a document into one or more components as specified by the user[4],
- provides optional, customizable iconic navigation panels on every page which contain links to other parts of the document, or other documents,
- hadles inlined equations ($\sum_{i=1}^{n} x_i = \int_0^1 f$), handles equation alignment ($A_{B_{C+D}}$), right-justified numbered equations (see equation 2), tables (see Table 2), or figures (see Figure 2), and *any arbitrary environment*[5],
- figures or tables can be arbitrarily scaled and oriented and shown either as inlined images or "thumbnail" sketches
- can produce output suitable for browsers that support inlined images or character based browsers (as specified by the user),
- handles definitions of new commands, environments, and theorems even when these are defined in external style files[6],
- handles footnotes[7], tables of contents, lists of figures and tables, bibliographies, and can generate an index,
- translates cross-references into hyperlinks and extends the LaTeX cross-referencing mechanism to work not just within a document but *between documents* which may reside in remote locations,
- translates LaTeX accent and special character commands (e.g. Ȧ Øö £©¶) to the equivalent ISO-LATIN-1 or Unicode character set where possible,
- recognizes hypertext links (to multimedia resources or arbitrary internet services such as sound/video/ftp/http/news) and links which invoke arbitrary program scripts, all expressed as LaTeX commands,
- recognizes *conditional text* which is intended only for the hypertext version, or only for the paper (DVI) version,
- can include raw HTML in a LaTeX document (e.g. in order to specify interactive forms),
- can deal sensibly at least with the *Common LaTeX commands* summarized at the back of the LaTeX blue book [1],

---

[4] The user can specify the *depth* at which the document should not be broken up any further.

[5] These are passed to LaTeX and then converted to images which are either included in the document or are made available through hypertext links.

[6] This allows the definition of HTML macros in LaTeX !

[7] Like this!

- will try and translate any document with embedded LaTeX commands irrespective of whether it is complete or syntactically legal.
- can be configured to translate equations and/or tables either as GIF images or as HTML 2.1 or 3.0 markup, as browsers become available which are suitable for the task.
- links symbolic references across document segments which are independently processed.

A selection of documents illustrating the different contexts in which LaTeX2HTML has been used is available at
http://www-dsed.llnl.gov/files/programs/unix/latex2html/manual/.

2 User Manual

To use LaTeX2HTML simply type `latex2html <file>.tex`. The `.tex` suffix is optional and will be supplied by the program if it is omitted by the user. This will create a new directory called `file` which will contain the generated HTML files, some log files and possibly some images. To view the result use an HTML viewer such as NCSA Mosaic or Netscape Navigator on the main HTML file which is `file/file.html`. This file will contain navigation links to the other parts of the generated document.

It is possible to customize the output from LaTeX2HTML using a number of command line options (see Section 2.1) with which you can specify how to break up the document, where to put the generated files, what the title is, what the signature at the end of each page is, what to put in the navigation panel, what kind of extra information to include about the document, whether to retain the original LaTeX section numbering scheme, etc.

Also, the LaTeX2HTML script includes a short manual which can be viewed by saying `%nroff -man latex2html`.

2.1 Command Line Options

The command line options described below can be used to change the default behavior of LaTeX2HTML. Alternatively, the corresponding environment variables in the initialization file `.latex2html-init` may be changed, in order to achieve the same result.

-split *num* Stop splitting sections into separate files at this depth. A value of 0 will put the document into a single HTML file. The default is 8.

-link *num* Stop revealing child nodes at each node at this depth. (A node is a part/chapter/section/subsection/subsubsection etc.). A value of 0 will show no links to child nodes, a value of 1 will show only the immediate descendents, etc. A value at least as big as that of the `-split` option will produce a table of contents for the tree structure, rooted at each given node. The default is 4.

-nolatex Disable the mechanism for passing unknown environments to LaTeX for processing. This can be thought of as "draft mode" which allows faster translation of the basic document structure and text, without fancy figures, equations or tables.

This option has been superceded by the `-no_images` option (see below).

-external_images Instead of including any generated images inside the document, leave them outside the document and provide hypertext links to them.

-ascii_mode Use only ascii characters and do not include any images in the final output. In ascii mode the output of the translator can be used on character based browsers which do not support inlined images (the `<IMG>` tag).

-t *top-page-title* Name the document using this title.

-up_url *URL* Specifies a universal resource locator (URL) to associate with the UP button of the top-level navigation panel.

-up_title *string* Specifies a title associated with this URL.

-down_url *URL* Specifies a URL for the NEXT button of the bottom-level navigation panel.

-down_title *string* Specifies a title associated with this URL.

-contents *URL* Specifies a URL for the table of contents button for document segments that would not otherwise have one.

-index *URL* Specifies a URL for the index button for document segments that otherwise would not have an index.

-external_file *filename-prefix* Specifies the prefix of the `.aux` file that this document segment should read. The `.aux` extension will be appended to this prefix. This file is necessary for document segments to obtain figure and section numbers from LaTeX.

-dir *output-directory* Redirect the output to this directory.

-no_subdir Place the generated HTML files in the current directory. The default behaviour is to create (or reuse) another file directory.

-prefix *filename-prefix* The `filename-prefix` will be prepended to all `.gif`, `.pl` and `.html` files produced, except for the top-level `.html` file. This will enable multiple products of LaTeX`2HTML` to peacefully coexist in the same directory. Do **not**, however, attempt to *simultaneously* run multiple instances of LaTeX`2HTML` using the same output directory, or various temporary files will overwrite each other.

-font_size *size* This option provides better control over the font size of environments processed by LaTeX. *size* must be one of the font sizes that LaTeX recognized; i.e. `10pt`, `11pt`, `12pt`, etc. The default size is the same as the LaTeX document. Whatever size is selected, it will be magnified by the installation variables, $MATH_SCALE_FACTOR or $FIGURE_SCALE_FACTOR.

-no_tex_defs If specified, the translator will make no attempt to interpret raw TeX commands. This feature will enable sophisticated authors the ability to insert arbitrary TeX commands in environments that are distined to be processed by LaTeX anyway (e.g. figures, theorems, etc.)

-ps_images Use links to external postscript images rather than inlined GIF images.

-address *author-address* Sign each page with this address.

-no_navigation Disable the mechanism for putting navigation links in each page.

-top_navigation Put navigation links at the top of each page.

-bottom_navigation Put navigation links at the bottom of each page AS WELL as the top.

-auto_navigation Put navigation links at the top of each page. If the page exceeds `$WORDS_IN_PAGE` number of words (the default is 450) then put one at the bottom of the page as well.

-index_in_navigation Put a link to the index page in the navigation panel if there is an index.

-contents_in_navigation Put a link to the table of contents in the navigation panel if there is a table of contents.

-next_page_in_navigation Put a link to the next logical page in the navigation panel.

-previous_page_in_navigation Put a link to the previous logical page in the navigation panel.

-info *string* Generate a new section *About this document ...* containing information about the document being translated. The default is to generate such a section with information on the original document, the date, the user and the translator. An empty string (or the value 0) disables the creation of this extra section. If a non-empty string is given, it will be placed in the contents of the *About this document ...* section instead of the default information.

-reuse *reuse_option* The *reuse_option* specifies how or whether image files are to be shared or recycled, and accepts three valid options:

- **0** Do not ever share or recycle image files. This choice also invokes an interactive session prompting the user about what to do about a pre-existing `HTML` directory, if it exists.
- **1** Recycle image files from a previous run if they are available, but do not share identical images that must be created in this run.
- **2** Recycle image files from a previous run and share identical images from this run. (This is the default).

Section 4.5 provides additional information about image reuse.

-no_reuse Do **not** share or recycle images generated during previous translations. This is equivalent to **-reuse 0**. (This will enable the initial interactive session during which the user is asked whether to reuse the old directory, delete its contents or quit)

-init_file file Load the specified file. This Perl file will be loaded after loading $HOME/.latex2html-init (if one exists). It can be used to change the default options.

-no_images Do not attempt to produce any inlined images. The missing images can be generated "off-line" by restarting LaTeX2HTML with the option `-images_only`.

-images_only Try and convert any inlined images that were left over from previous runs of LaTeX2HTML.

-show_section_numbers Show section numbers. By default the section numbers are not shown in order to encourage the use of particular sections as standalone documents. In order to be shown, section titles must be unique and must not contain inlined graphics.

-html_version *version* This specifies the HTML version to generate. Currently, versions 2.1, 2.2, 3.0 and 3.1 are available. The default version is 2.0.

-vs Print the current version of LaTeX2HTML.

-h Print out the list of options.

2.2 Extending the Translator

As the translator covers only partially the set of LaTeX commands and because new LaTeX commands can be defined arbitrarily using low level TeX commands, the translator should be flexible enough to allow end users to specify how they want particular commands to be translated.

2.2.1 Adding Support for Specific Style Files

LaTeX2HTML provides a mechanism where code to translate specific style files is automatically loaded if such code is available. When the use of a style file such as `german.sty` is detected in a LaTeX source document, the translator looks for a file `LATEX2HTMLDIR/styles/german.perl`. If one is found, then it will be loaded into the main script.

This mechanism will help to keep the core script smaller as well as make it easier for others to contribute and share solutions on how to translate specific style files. The current distribution includes the files listed in Table 1. These will provide good examples of how you can create your own extensions to LaTeX2HTML. The problem however, is that writing such extensions requires an understanding of Perl and of the way LaTeX2HTML is organized. More user-friendly interfaces will be investigated.

At the moment a rudimentary mechanism is provided so that a user can ask for particular commands and their arguments either to be ignored or passed on to LaTeX for processing (the default behavior for unrecognized commands is for their arguments to remain in the HTML text). Commands that are passed on to LaTeX are converted to images which are either "inlined" in the main document or become accessible via hypertext links. Simple extensions using the commands above may be included in the `LATEX2HTMLDIR/latextohtml.config` file or in each personal `HOME/.latex2html-init` initialization file.

`.perl` file	Description
`alltt`	Supports the LaTeX 2$_{\varepsilon}$'s `alltt` package
`changebar`	Provides rudimentary change bar support
`color`	Causes colored text to be processed as ordinary text by LaTeX2HTML
`french`	Special support for the French language
`epsfig`	Processes embedded figures not enclosed in a `figure` environment
`floatfig`	Processes floating figures
`german`	Special support for the German language
`graphics`	Supports commands in the `graphics` package
`graphicx`	Supports the alternate syntax of graphics commands
`heqn`	Alters the way displayed equations are processed
`htmllist`	Provides support for fancy lists
`makeidx`	Generates indices
`texdefs`	Supports raw TeX commands
`wrapfig`	Supports wrapped figures

Table 1: Supported LaTeX2HTML style files

2.2.2 Asking the Translator to Ignore Commands

Commands that should be ignored may be specified in the `.latex2html-init` file as input to the `ignore_commands` subroutine. Each command which is to be ignored should be on a separate line followed by compulsory or optional argument markers separated by `#`'s e.g.[8]:

```
<cmd_name>#{}# []# {}# [] ...
```

`{}`'s mark compulsory arguments and `[]`'s optional ones.

Some commands may have arguments which should be left as text even though the command should be ignored (e.g. `mbox`, `center`, etc.). In these cases the arguments should be left unspecified.

Here is an example of how this mechanism may be used:

```
&ignore_commands( <<_IGNORED_CMDS_);
documentstyle # [] # {}
linebreak# []
center
<add your commands here>
_IGNORED_CMDS_
```

---

[8] It is possible to add arbitrary Perl code between any of the argument markers which will be executed when the command is processed. For this however a basic understanding of how the translator works and of course Perl is required.

2.2.3 Asking the Translator to Pass Commands to LaTeX

Commands that should be passed on to LaTeX for processing because there is no direct translation to HTML may be specified in the `.latex2html-init` file as input to the `process_commands_in_tex` subroutine. The format is the same as that for specifying commands to be ignored. Here is an example:

```
&process_commands_in_tex (<<_RAW_ARG_CMDS_);
fbox # {}
framebox # [] # [] # {}
<add your commands here>
_RAW_ARG_CMDS_
```

3 Hypertext Extensions to LaTeX

This section describes how you can define hypertext entries in your `HTML` documents from within your LaTeX source. These are implemented as new LaTeX commands which have special meaning during the translation to `HTML` but are mostly ignored when processed by LaTeX.

All the new commands used in the sections below are defined in the file `html.sty` which is part of the distribution. *This file must be included in any LaTeX document using these features* by one of the following methods:

- Including `html` as an optional argument to the LaTeX `documentstyle` command;
- Including `html` in the LaTeX 2_ε `usepackage` command; or
- Pasting the definitions of individual commands directly into the text!

3.1 Hyperlinks in LaTeX

Arbitrary hypertext references are available through the commands `htmladdnormallink` and `htmladdimg`.

These are defined as follows:

```
\htmladdnormallink{<text>}{<URL>}
\htmladdimg{<URL>}
```

The first command expects some text as the first argument and a URL as the second argument. When processed by LaTeX (i.e. in the `dvi` or `ps` output files), the URL will have no effect. But when processed by the translator, the URL will be used to provide an active hypertext link (to another file, picture, sound file, movie, etc) e.g.

```
\htmladdnormallink{<URL>}
{http://www.ncsa.uiuc.edu/demoweb/url-primer.html}
```

A similar command called `htmladdnormallinkfoot` takes the same two arguments and has the same effect when generating HTML as `htmladdnormallink`. But when processed by LaTeX it shows the URL as a footnote.

In a similar way, the argument of the `htmladdimg` command should be a URL pointing to an image. This URL is ignored in the LaTeX hard copy output.

3.2 Including Arbitrary HTML Markup

Raw HTML or HTML+ tags and text may be included using the environment `rawhtml`. This can be used to take advantage of new HTML+ facilities as soon as they become available. A particularly good use of this feature is in the creation of interactive electronic forms from within a LaTeX document. When producing the paper (DVI) version of a document the `rawhtml` environment is ignored.

Here is an example:

```
\begin{rawhtml}
<HR>
<FORM ACTION="http://cbl.leeds.ac.uk/nikos/doc/error.html">
<OL>
<LI> <INPUT TYPE="checkbox" NAME="wp" VALUE="word"> Word for
Windows.
<LI> <INPUT TYPE="checkbox" NAME="wp" VALUE="wp"> Word Perfect.

<LI> <INPUT TYPE="checkbox" NAME="wp" VALUE="latex"> LaTeX.
<LI> Plain Text Editors (Please Specify): <INPUT TYPE="text" NAME="other_ed">
</OL>
So, what do think (comments please): <BR>
<INPUT TYPE="text" SIZE=45,4 NAME="other_wp">

<INPUT TYPE="submit" VALUE="submit this form but don't expect much!">
</FORM>
<HR>
\end{rawhtml}
```

The result is shown below:

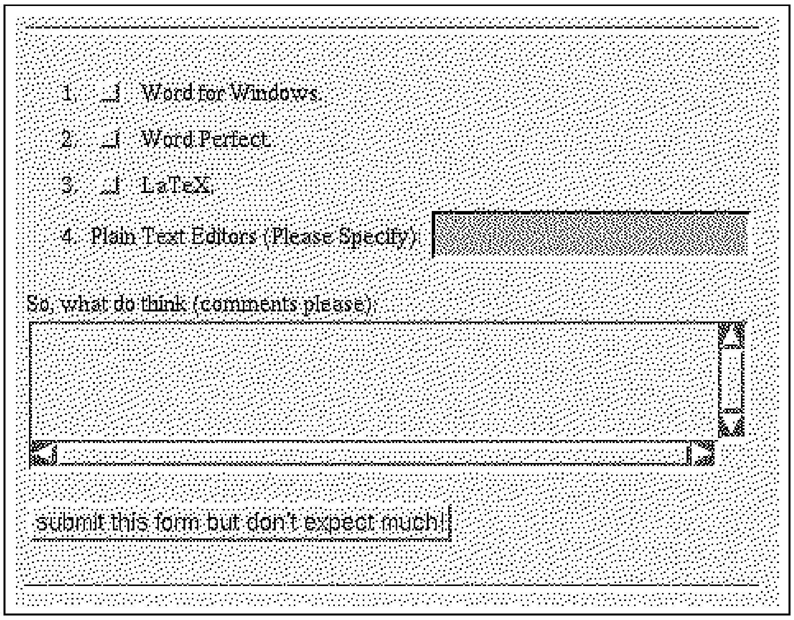
Warning: Avoid using LaTeX commands involving counters (e.g. numbered figures or equations) in conditional text because this may disrupt the values of the counters in the electronic version.

3.3 Conditional Text

Conditional text can be specified using the environments `latexonly` and `htmlonly`. These allow writing parts of a document which are intended only for electronic delivery or only for paper-based delivery.

This would be useful for example in adding a long description of a multimedia resource in the paper version of a document. Such a description would be redundant in the electronic version, as the user can have direct access to this resource.

Here is an example of the use of the `latexonly` environment:



1. ☐ Word for Windows
2. ☐ Word Perfect
3. ☐ LaTeX
4. Plain Text Editors (Please Specify):

So, what do think (comments please):

Figure 1: An electronic form. Of course in the online version of this document the form above would be active.

```
\begin{latexonly}
\begin{figure}
    \begin{center}
    \fbox{\includegraphics[width=4in]{eform.ps}}
    \end{center}
    \caption{An electronic form. Of course in the online version of this
     document the form above would be active.}
\end{figure}
\end{latexonly}
```

There are also shorthand notations to accomplish the same thing with less typing. The `\latex{}` command causes everything within the braces to be processed by LaTeX, but ingored by LaTeX2`HTML`. Conversely, the `\html{}` command causes everything within the braces to be ignored by LaTeX and processed by LaTeX2`HTML`. Finally the command `\latexhtml{}{}` causes everthing within the first set of braces to be processed exclusively by LaTeX, with the contents of the second set of braces processed solely by LaTeX2`HTML`.

3.4 Symbolic References Shown as "Hyperized" Text

In printed documents cross-references are shown through a *numerical or symbolic indirection* e.g. "see Figure 1" (numeric indirection), or "see section "Changes" (symbolic indirection). LaTeX2`HTML` can mirror this mechanism using the same numeric or symbolic references, or when these are not appropriate by using iconic references.

In a hypertext document however, cross-references can be shown without any indirection just by highlighting a relevant piece of text. This can make a document more readable as it removes unnecessary information.

A single new LaTeX command `hyperref` can be used for specifying how a cross-reference should appear both in the printed document and in the hypertext version.

Assuming that the label `sec:cond` is defined somewhere in a document the command `hyperref` which takes 4 arguments can be used in the same document as follows:

```
\emph{Is the concept of
\hyperref
% This will be highlighted in the hypertext version
{conditional text} % argument #1
% This will be shown in the printed version
% followed by a numeric reference ...
{conditional text (see Section }   % argument #2
% ... followed by this text
{ for more information)}  % argument #3
% This the common label
{sec:cond} % argument #4
a good idea? }
```

Here is how it will be shown:
Is the concept of conditional text (see Section 8.6 for more information) a good idea?

In the hypertext version what would appear is:
Is the concept of conditional text a good idea?
Of course conditional text would be an active hypertext link.

Another command also defined in `html.sty` is `htmlref` which has the same effect as `hyperref` during the conversion to HTML. It takes two arguments, some text and a label. In the HTML version the text will be "hyperized" pointing to the label. In the paper version the text will be shown as it is and the label will be ignored eg

```
With \texttt{htmlref} \htmlref{it's easy to make links}{fig:example}.
```

In the HTML version it will be shown as:
With `htmlref` it's easy to make links.

3.5 Symbolic References Between Living Documents

The method of the previous section to generated symbolic hyperized links can easily be extended to *external* documents processed by LaTeX2HTML. When LaTeX2HTML processes a document, it generates a `perl` file named `labels.pl` which contains a list of all the symbolic labels that were defined, along with their locations. The name of this file is preceeded by the filename prefix to allow different document segments to share the same directory. Links to an external document are then possible once a connection is established to that document's `labels.pl` file. This connection is established by the `\externallabels` command:

```
\externallabels{<URL to directory of external document>}
               {<external document labels.pl file>}
```

The first argument to `externallabels` should be a URL to the directory containing the external document. The second argument should be the full pathname to the `labels.pl` file belonging to the external document. Note that for *remote* external documents it is necessary to copy the `labels.pl` file locally so that it can be read when processing a local document that uses it. The command `externallabels` can be used once for each external document in order to import the *external labels* into the current document. A warning is given if `labels.pl` cannot be found.

If a symbolic reference made in either of the commands described in Section 3.4 is not defined within the document itself, LaTeX2HTML will look for that reference in one of the external files.[9] After any modifications in an external document (sections added/deleted, segmentation into different physical parts, etc.) a new `labels.pl` will be generated. If the `externallabels` command in another document contains the correct address to the `labels.pl` file, then the cross-references will be realigned after running the local document through the translator.

There is also a mechanism analogous to the `label-ref` pairs of LaTeX, which can be used only within a single document. These labels are called *internal labels*, as opposed to the external labels defined above.

Either type of label is defined with a LaTeX `label` command. Labels can be referenced *within* a document using a `ref` command. When processed by LaTeX, each `ref` command is replaced by the section number in which the corresponding `label` occurred. When processed by the translator, each `ref` is replaced by a hypertext link to the place where the corresponding `label` occurred. This mechanism can be extended to external documents by the `\externalref{}` command:

```
\externalref{<symbolic label in remote document>}
```

The argument to `externalref` may be any symbolic label defined in any of the external documents in the same Such references to external symbolic labels are then translated into hyperlinks pointing to the external document.

3.5.1 A Cross-referencing example

To understand this mechanism better consider how you would maintain a link to this section (of the hypertext version of this document) from one of your documents without using labels. Sure enough you can get the name of the physical file that this section is in. This however is very likely to change and any links to it will become invalid. To update your link, the name of the new file must be found and your link changed by hand. Also there is no general updating mechanism so the only way to find out if your document is pointing to the right place is by actually following the link and then doing a manual update[10].

Now consider how it could be done with symbolic labels. First you have to import the labels used in this document by copying the file labels.pl, saving it say in `/tmp/labels.pl`, and then adding anywhere in your document:

---

[9] Care must be taken to ensure that critical symbolic references are unique across related documents.

[10] Link validation can be done automatically but the updating must be done manually when filenames have changed (assuming no other symbolic label mechanism is available).

```
\externallabels{http://cbl.leeds.ac.uk/nikos/tex2html/doc/manual}
               {/tmp/labels.pl}
```

After that you can use the label `crossrefs` defined at the beginning of this section[11] as follows:

```
\externalref{crossref}
```

This will be translated into the appropriate hyperlink to this page. If there are any changes in this document and you would like to bring your document upto date you have to copy labels.pl again and rerun the translator on your document. Of course if I move the directory containing the HTML files for this document somewhere else then you would have to make a change in the argument of the `externallabels` command to reflect this.

It is obvious that some level of collaboration is required between authors trying to maintain cross-references between different documents. Using symbolic labels makes this a lot easier (especially for documents written by the same author).

3.6 Document Segmentation[12]

One of the greatest appeals of the World Wide Web is its high connectivity through hyperlinks. As we have seen, the LaTeX author can provide these links either manually or symbolically. Manual links are more tedious because a URL must be provided by the author for every link, and updated every time the target documents change. Symbolic links are more convenient, because the translator keeps track of the URLs. Earlier releases of LaTeX2`HTML` required the entire document to be processed together if it was to be linked symbolically. However, it was easy for large documents to overwhelm the memory capacities of moderate sized computers. Furthermore, processing time could become prohibitively high, if even a small change required the entire document to be reprocessed.

For these reasons, program segmentation was developed. This feature enables the author to subdivide his document into multiple *segments*. Each segment can be processed independently by LaTeX2`HTML`. Hypertext links between segments can be made symbolically, with references shared through auxiliary files. If a single segment changes, only that segment needs to be reprocessed (unless a label is changed that another segment requires). Furthermore, the entire document can be processed without modification by LaTeX to obtain the printed version. The top level segment that LaTeX reads is called the *parent* segment. The others are called *child* segments.

Program segmentation does require a little more work on the part of the author, who will now have to untertake some of bookkeeping formerly performed exclusively by LaTeX2`HTML`. The following four LaTeX extensions carry out segmentation:

`\segment{file}{sec-type}{heading}` This commands indicates the start of a new program segment. This segment resides in `file.tex`, represents the start of a new LaTeX sectional unit of type `sec-type` (e.g., `section, chapter,` etc.), and has a heading of `heading`. A variation of this command, `\segment*`, is provided for segments that are not to appear in the table of contents. These commands perform the following operations in LaTeX:

---

[11] You either have to guess the role of each label by looking at the `labels.pl` file or by asking the author!

[12] This feature is supported only for users of LaTeX 2_ε.

1. The specified sectioning command is executed.
2. LaTeX will write its section and equation counters into an auxiliary file, named `file.ptr`. It will also write an `\htmlhead` command to this file. This information will tell LaTeX2HTML how to initialize itself for the new document segment.
3. LaTeX will then proceed to input and process the file, `file.tex`.

The `\segment` and `\segment*` commands are ignored by LaTeX2HTML.

`\internal[type]{prefix}` This command directs LaTeX2HTML to load intersegment information of type `type` from the file `<prefix><type>.pl`. Each program segment must be associated with a unique filename prefix, specified either through a command line option, or through the installation variable `$AUTO_PREFIX`. The information `type` must be one of the following:

internals This is the default type, which need not be given. It specifies that the internal labels from the designated segment are to be input and made available to the current segment.

contents The table of contents information from designated segment are to be made available to the current segment.

sections Sectioning information is to be read in. Note that the segment containing the table of contents requires both contents and sections information from all other program segments.

figure Lists of figures from other segments are to be read.

table Lists of tables from other segments are to be read.

index Index information from other segments are to be read.

`\startdocument` Only the parent segment contains `\begin{document}` and `\end{document}` statements. The child segments, therefore, cannot be processed separately by LaTeX (without modification). They can be processed separately by LaTeX2HTML, provided it is told where the end of the end the LaTeX preamble is. This is the function of the `\startdocument` directive. It substitutes for `\begin{document}` in child segments, but is otherwise ignored by both LaTeX and LaTeX2HTML.

`\htmlhead{sec-type}{heading}` This command is not normally placed in the document at all. It is automatically passed from parent to child via `file.ptr`. It identifies to LaTeX2HTML that the current segment is a LaTeX sectional unit of type `sec-type`, with the specified heading. This command is ignored by LaTeX itself.

3.6.1 A segmentation example

The best way to illustrate document segmentation is through a simple example. Suppose that a document is to be segmented into one parent and two child segments. Let the parent segment be `report.tex`, and the the two child segments be `sec1.tex`, and `sec2.tex`. The latter are translated with filename prefixes of `s1` and `s2`, respectively. This example is included with the version 96.1 distribution of LaTeX2`HTML`, with more prolific comments than are shown here.

The text of `report.tex` is given below:

```
\documentclass{article}          % Must use LaTeX 2e
\usepackage{html,makeidx,color}

\internal[figure]{s1}            % Include internal information
\internal[figure]{s2}            % from children
\internal[sections]{s1}
\internal[sections]{s2}
\internal[contents]{s1}
\internal[contents]{s2}
\internal[index]{s1}
\internal[index]{s2}

\begin{document}                 % The start of the document
\title{A Segmentation Example}
\date{\today}
\maketitle
\tableofcontents
\listoffigures

% Process the child segments:

\segment{sec1}{section}{Section 1 title}
\segment{sec2}{section}{Section 2 title}
\printindex
\end{document}
```

This file obtains the information necessary to build an index, a table of contents and a list of figures from the child segments. It then proceeds and typesets these.

The first child segment, `sec1.tex` is shown below:

```
\begin{htmlonly}
\documentclass{article}
\usepackage{html,color,makeidx}
\input{sec1.ptr}
\end{htmlonly}
\internal{s2}
\startdocument
Here is some text.
\subsection{First subsection}
Here is subsection 1 \label{first}.
\begin{figure}
\colorbox{red}{Some red text\index{Color text}}
```

```
\caption[List of figure caption]{Figure 1 caption}
\end{figure}
Reference\index{Reference} to \ref{second}.
```

The first thing this child segment does is establish the LaTeX packages it requires, then loads the counter information that was written by the `\segment` command that invoked it. Since this segment contains a symbolic reference (`second`) to the second segment, it must load the internal labels from that segment.

The final segment `sec2.tex` is shown below:

```
\begin{htmlonly}
\documentclass{article}
\usepackage{html,makeidx}
\input{sec2.ptr}
\end{htmlonly}
\internal{s1}
\startdocument
Here's is another section \label{second}.
Plus another\index{Reference, another} reference \ref{first}.
\begin{figure}
\fbox{The figure}
\caption{The caption}
\end{figure}
```

This segment needs to load internal labels from the first one, because of the reference to `first`. These circular dependencies (two segments referencing each other) are either not allowed or handled incorrectly by the Unix utility `make`, without resorting to time stamps and some trickery. A *time stamp* is a zero-length file whose only purpose is to record its creation time. Besides evaluating segment interdependence, another function of `make` is to provide intersegment navigation information.

A sample `makefile` is included in the distribution which correctly generates the fully linked document. The first time it is invoked, it runs:

- LaTeX on `report.tex` twice;
- `dvips` to generate `report.ps`;
- LaTeX2`HTML` on `sec1.tex`;
- LaTeX2`HTML` on `sec2.tex`. At this point, `sec2.html` is completely linked, since the labels from the `sec2` were available.
- LaTeX2`HTML` on `sec1.tex` to pick up the labels from `sec1`;
- LaTeX2`HTML` on `report.tex`.

Proper operation of `make` depends on the fact that LaTeX2`HTML` updates its own internal label file only if something in its current program segment causes the labels to change from the previous run. This ensures that LaTeX2`HTML` is not run unnecessarily. It is also important that the information page be supressed by specifying `-into ""` on all but the last program segment.

3.7 Hypertext Links in Bibliographic References (Citations)

If a report or a book that is cited (using the `cite`) command is available (or there is information about it) on WorldWide Web then it is possible to add the appropriate hypertext links in your bibliographic database (the `.bib`) file.

Here is an example of a bibliographic entry for the original LaTeX [1] book:

```
@book{lamp:latex,
title = "LATEX User's Guide \& Reference Manual, 2nd Edition",
year = 1994 ,
author = "Leslie Lamport",
Publisher = "Addison-Wesley Publishing Company, Inc.",
note = "Online information on TeX and LaTeX is available at
\htmladdnormallink{http://curia.ucc.ie/info/TeX/menu.html}
{http://curia.ucc.ie/info/TeX/menu.html} and
\htmladdnormallink{http://es-sun2.fernuni-hagen.de/info2html?(latex.info)Top}
{http://es-sun2.fernuni-hagen.de/info2html?(latex.info)Top}"}
```

No other modifications are required for this to work - LaTeX and BibTeXshould work as normal.

For those who use the Harvard style for references there exists a special conversion add-on package[13].

3.8 Customizing the Navigation Panels

The navigation panels are the strip containing "buttons" and text that appears at the top and perhaps at the bottom of each generated page and provides hypertext links to other sections of a document. Some of the options and variables that control whether and where it should appear have already been mentioned in Section 2.1.

A simple mechanism for appending customized buttons to the navigation panel is provided by the command `htmladdtonavigation`. This takes one argument which LaTeX2`HTML` appends to the navigation panel. For example,

```
\htmladdtonavigation
   {\htmladdnormallink
      {\htmladdimg{http://server/mybutton.gif}}
      {http://server/link}}
```

will add an active button `mybutton.gif` pointing to the specified location.

Apart from these facilities it is also possible to specify completely what appears in the navigation panels and in what order. As each section is processed, LaTeX2`HTML` assigns relevant information to a number of global variables. These variables are used by the subroutines `top_navigation_panel` and `bottom_navigation_panel`, where the navigation panel is constructed as a string consisting of these variables and some formatting information.

These subroutines can be redefined in a system or a user configuration file (`LATEX2HTMLDIR/latex2html.config` and `HOME/.latex2html-init` respectively). *Any combination of text, HTML tags, and the variables mentioned below is acceptable.*

---

[13] http://www.arch.su.edu.au/~peterw/latex/harvard/

The control panel variables are:

Iconic links (buttons)

PREVIOUS Points to the previous section

UP Points up to the ”parent” section

NEXT Points to the next section

NEXT_GROUP Points to the next ”group” section

PREVIOUS_GROUP Points to the previous ”group” section

CONTENTS Points to the contents page if there is one

INDEX Points to the index page if there is one

Textual links (section titles)

PREVIOUS_TITLE Points to the previous section

UP_TITLE Points up to the ”parent” section

NEXT_TITLE Points to the next section

NEXT_GROUP_TITLE Points to the next ”group” section

PREVIOUS_GROUP_TITLE Points to the previous ”group” section

If the corresponding section exists each iconic button will contain an active link to that section. If the corresponding section does not exist, the button will be inactive. If the section corresponding to a textual link does not exist then the link will be empty.

The number of words that appears in each textual link is controlled by the variable `WORDS_IN_NAVIGATION_PANEL_TITLES` which may also be changed in the configuration files.

Below is an example of a navigation panel (the “.” is the Perl string concatenation operator and “#” signifies a comment).

```
sub top_navigation_panel {

    #  Start with a horizontal rule (3-d dividing line)
    "<HR> ".

    # Now add few buttons with a space between them
    "$NEXT $UP $PREVIOUS $CONTENTS $INDEX $CUSTOM_BUTTONS" .

    "<BR>\n" . # Line break

    # If ‘‘next’’ section exists, add its title to the navigation panel
    ($NEXT_TITLE ? "<B> Next:</B> $NEXT_TITLE\n" : undef) .
```

```
    # Similarly with the ``up'' title ...
    ($UP_TITLE ? "<B>Up:</B> $UP_TITLE\n" : undef) .

    # ... and the ``previous'' title
    ($PREVIOUS_TITLE ? "<B> Previous:</B> $PREVIOUS_TITLE\n" : undef) .

    #  Horizontal rule (3-d dividing line) and new paragraph
    "<HR> <P>\n"
}
```

4 Special Features

This section describes some special important considerations to keep in mind when using LaTeX2HTML.

4.1 Variation on HTML versions

The Hypertext Markup Language (HTML) is an evolving standard, with different versions supporting different features. In order to make your documents viewable by the widest possible audience, you should use the most basic HTML version in common usage. This is currently version 2.0 , which is the default version for LaTeX2HTML. It supports images, server-side image maps, interactive forms, and minimal typographic elements (bold, itallic and teletype). This version adheres to the ISO Latin 1 (ISO-8879) character set. Other HTML versions supported by LaTeX2HTMLare as follows:

Version 2.1 HTML version 2.1 is essentially identical to HTML version 2.0, with some extensions for internationalization. Most importantly, the default character set is no longer ISO 8859-1 but ISO-10646 (Unicode). This is a 16-bit character set and can thus display a much larger set of characters. There are also provisions for bidirectional languages (e.g. in Arabic, the text is written from right to left, but numerals from left to right), and provisions in HTML to determine the character set and the language used. Not all of the symbols are available in TeX, LaTeX2HTML, or any browser yet available. However, the version 2.1 emulation of LaTeX2HTMLis at least a start.

Version 2.2 This version supports all the features of version 2.1, plus the HTML3 Table Model . This feature is already available in many HTML 2.0 browsers, including Netscape Navigator V1.2.

Version 3.0 This version adds to version 2.2 some of the HTML 3.0 textual formatting features, including centering, flush right, flush left, and underlining.

Version 3.1 In addition to all fo the features of version 3.0, this version adds support for math markup . Currently the only browser which can display this markup is Arena.

4.2 Internationalization

A special variable `LANGUAGE_TITLES` in the initialization or configuration files determines the language in which some section titles will appear. For example setting it to

```
$LANGUAGE_TITLES = ``french'';
```

will cause LaTeX2HTML to produce “Table des matières” instead of “Table of Contents”. Furthermore, the value of `\today` is presented in a format which is customary in that language.

The only languages currently supported are “french”, “english” and “german,” but it is trivial to add support for another language in the file `latex2html.config`. As a guide here is the entry for the French titles:

```
sub french_titles {
 $toc_title = "Table des mati\\`eres";
 $lof_title = "Liste des figures";
 $lot_title = "Liste des tableaux";
 $idx_title = "Index";
 $bib_title = "R\\'ef\\'erences";
 $info_title = "\\`A propos de ce document...";
 $abs_title = "R&eacute;sum&eacute;";
 $pre_title = "Pr&eacute;face";
 $app_title = "Annexe";
 @Month = ('', 'janvier', 'f&eacute;vrier', 'mars', 'avril', 'mai',
              'juin', 'juillet', 'ao&ucirc;t', 'septembre', 'octobre',
              'novembre', 'd&eacute;cembre');
}
```

In order to provide full support for another language you may also want to replace the navigation buttons which come with LaTeX2HTML (which are by default in English) with your own. As long as the new buttons have the same filenames as the old ones there should not be a problem.

4.3 Image Conversion

LaTeX2HTML converts equations, special accents, external postscript files, and LaTeX environments it cannot directly translate into inlined images. This section describes how it is possible to control the final appearance of such images. For the purposes of the discussion below,

“small images” will refer to equations, special accents and any other image generating LaTeX commands, while

“figures” will apply to any image generating LaTeX environments (eg figure, table, minipage etc).

The size of all “small images” depends on a configuration variable `MATH_SCALE_FACTOR` which specifies how much to enlarge or reduce them in relation to their original size in the postscript version of the document. For example a scale factor of 0.5 will make all images half as big while a scale factor of 2 will make them twice as big. Larger scale factors result in longer processing times and larger intermediate image files. A scale factor will only be effective if it is greater than 0.

The configuration variable `FIGURE_SCALE_FACTOR` performs a similar function for "figures". Both of these configuration variables are initially set to 1.6.

For finer control, several parameters affecting the conversion of a single "figure" can be controlled with the command `htmlimage`, which is defined in `html.sty`. The one argument of `htmlimage` is a string of options separated by commas. The options are:

- `scale=<`*`scale factor`*`>`
- `external`
- `thumbnail=<`*`scale factor`*`>`
- `map=<`*`server-side image map URL`*`>`
- `usemap=<`*`client-side image map URL`*`>`
- `flip=<`*`flip_option`*`>`
- `align=<`*`alignment`*`>`

The `scale` option allows control over the size of the final image.

The `external` option will cause the image not to be inlined (images are inlined by default). External images will be accessible via a hypertext link.

The `thumbnail` option will cause a small inlined image to be placed in the caption. The size of the thumbnail depends on the scale factor. The use of the `thumbnail` option implies the `external` option.

The `map` and options specify that the image is to be made into an active image map. (See section 4.6 for more information.)

The `flip` option specifies a change of orientation of the electronic image relative to the printed version. The `<`*`flip_option`*`>` is any single command recognized by `pnmflip`. The most useful of these include:

rotate90 or r90 This will rotate the image clockwise by 90°.

rotate270 or r270 This will rotate the image counterclockwise by 90°.

leftright This will flip the image around a vertical axis of rotation.

topbottom This will flip the image around a horizontal axis of rotation.

The `align` option specifies how the "figure" will be aligned. The choices are: `top, bottom, middle, left, right` and center. The `middle` option specifies that the image is to be left-justified in the line, but centered vertically. The `center` option specifies that it should also be centered horizontally. This option is valid only if the `HTML` version is 3 or higher, or if the `NETSCAPE` configuration variable is set. The default value is `BOTTOM`.

In order to be effective the command `htmlimage` and its options **must be placed inside the environment on which it will operate**.

4.3.1 An embedded image example

The effect of the LaTeX commands below can be seen in the thumbnail sketch of Figure 2.

```
\begin{figure}
    \htmlimage{thumbnail=0.5}
    \centering\includegraphics[width=5in]{figure.ps}
    \caption{A sample figure showing part of a page generated by
       \latextohtml{} containing a customized navigation panel
       (from the \protect\htmladdnormallinkfoot{CSEP project}
        {http://csep1.phy.ornl.gov/csep.html}).}
       \label{fig:example}
\end{figure}
```

The `htmlimage` command is also often useful to cancel-out the effect of the configuration variable `FIGURE_SCALE_FACTOR`. For example to avoid resizing a color screen snap despite the value of `FIGURE_SCALE_FACTOR` it is possible to use `htmlimage{scale=0}`.

4.4 Figures, Tables and Arbitrary Environments

These are here to show how the translator handles figures, tables and other environments. Compare the paper with the online version.

gnats	gram	$13.65
	each	.01
gnu	stuffed	92.50
emur		33.33
armadillo	frozen	8.99

Table 2: A sample table taken from [1]

Here are some some automatically numbered right-justified equations

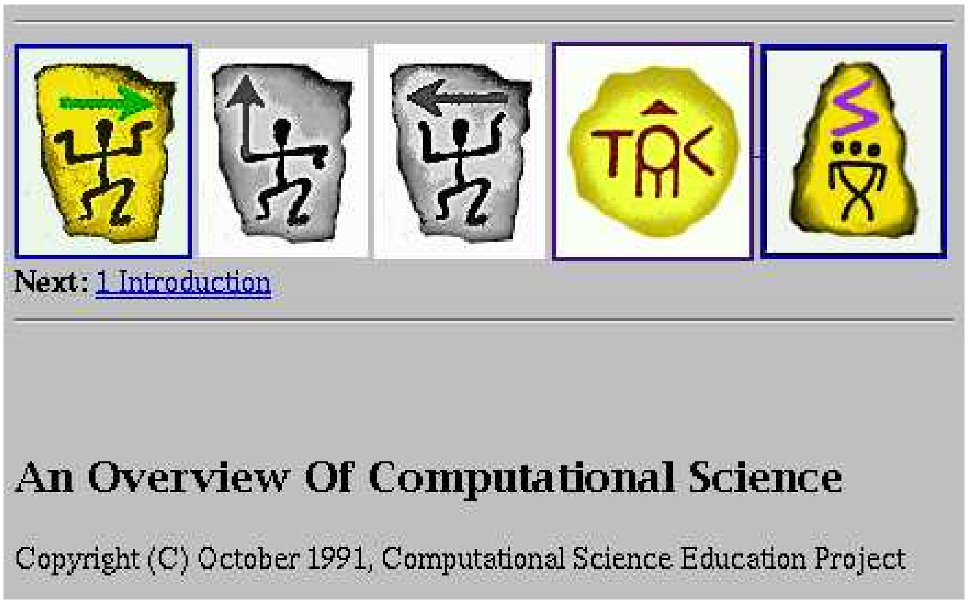
$$\Phi_{l+1,m,n} = (\Phi + h\frac{\partial\Phi}{\partial x} + \frac{1}{2}h^2\frac{\partial^2\Phi}{\partial x^2} + \frac{1}{6}h^3\frac{\partial^3\Phi}{\partial x^3} + \ldots)_{l,m,n} \tag{1}$$

with some gratuitously áccënted text in between them.

$$\begin{multline}\frac{\Phi_{l+1,m,n} - 2\Phi_{l,m,n} + \Phi_{l-1,m,n}}{h^2} + \frac{\Phi_{l,m+1,n} - 2\Phi_{l,m,n} + \Phi_{l,m-1,n}}{h^2} + \\ \frac{\Phi_{l,m,n+1} - 2\Phi_{l,m,n} + \Phi_{l,m,n-1}}{h^2} = -I_{l,m,n}(v) \tag{2}\end{multline}$$

4.5 Image sharing and recycling

It is not hard too see how reasonably sized papers, especially scientific articles, can require the use of many hundreds of external images. For this reason, image sharing and recycling is of critical importance. In this context, "sharing" refers to the use of one image in more



Next: 1 Introduction

An Overview Of Computational Science

Copyright (C) October 1991, Computational Science Education Project

Figure 2: A sample figure showing part of a page generated by LaTeX2`HTML` containing a customized navigation panel (from the CSEP project[15]).

than one place in an article. "Recycling" refers to the use of an image left over from a previous run of LaTeX2`HTML`. Without them, every instance of an image would have to be regenerated every time even the slightest change in the document were made.

All types of images can be shared. These include "small images" and figures with or without thumbnails and image maps. Furthermore, most images can also be reused. The only exception are those which are *order sensitive*, meaning that their content depends upon their location. Examples of order sensitive images are `equation` and `eqnarray` environments. This is because their figure numbers are part of the image. Figures and tables with captions, on the other hand, are order insensitive because the figure numbers are handled by `HTML` and are not part of the image itself.

If you have a document with a great many displayed equations, you might try using the `heqn` style package. Inclusion of this style file has absolutely no effect on the printed version of the article, but it does change the way in which LaTeX2`HTML` translates displayed equations and equation arrays. It causes the equation numbers of the `equation` environment to be moved outside of the images themselves, so that they become order-independent and hence recyclable. Images that result from the `eqnarray` environment are also recyclable, as long as their equation numbers remain unchanged from the previous run. An optional `\nonumber` command is recognized in each line of the equation array. A side-effect of this approach is that equation numbers will appear on the *left* side of the page. The `heqn` package requires the `html` package.

4.6 Active Image Maps

Image maps are images with active regions in which a Web surfer can click to send him off to another sector of cyberspace. LaTeX2HTML can design either inline "figures" or external ones (with or without a thumbnail version) to be image maps. However, `HTML` requires a URL of a `HTML` *map file* which defines the coordinates of each active region in the map with a destination URL. Usually, this map file is kept on the server machine, however HTML version 3 also allows it to reside on the client side for faster response. Both configurations are supported by LaTeX2HTML through the `htmlimage` options `map` and `usemap`, respectively.

Keeping such a map file up to date manually can be tedious, especially with dynamic documents under revision. An experimental program `makemap` can help automate this process. This program (which is really a `perl` script) takes one mandatory argument and one optional argument. The mandatory argument is the name of a *user map* file, defined below. The optional argument is the name of the directory where the `HTML` map file(s) are to be placed.

The best way of describing how this works is by example. Suppose that a document has two figures designated to become active image maps. The first figure included a statement like,

```
\begin{figure}
\htmlimage{map=/cgi-bin/imagemap/BlockDiagram.map,...}
. . .
\end{figure}
```

The second figure had a line like,

```
\begin{figure}
\htmlimage{map=/cgi-bin/imagemap/FlowChart.map,...}
. . .
\end{figure}
```

A typical user map file, named `report.map`, might contain the following information[16]:

```
#
#  Define the location(s) of the labels.pl file(s):
#
+report/  URL
#
#  Define map #1:
#
BlockDiagram.map:
label1  rect    288,145 397,189
label2  rect 307,225 377,252
label2  default
#
```

---

[16] This file is designed for an NCSA server. CERN servers use "`rect`" instead of "`rectangle`," specify a radius instead of an outer point in the circle, and enclose point coordinates by parentheses.

```
#  Define map #2
#
FlowChart.map:
label3  circle  150,100 200,100
label4  default
```

In this file, comments are denoted by a # sign in column 1. The line beginning with `+bench` states that the symbolic labels are to be found in the `labels.pl` contained in the directory `report`, and that its associated URL is as stated. Any number of external `labels.pl` files may be so specified. The block diagram image has two active regions. The first is a rectangle bounded by corners (288,145) and (397,189), while the second is a rectangle bounded by corners (307,225) and (377,252). These coordinates can be obtained with the aid of a program such as `xv`. If the user clicks in the first rectangle, it will cause a branch to the URL associated with symbolic label `label1` defined in the `labels.pl` file found in directory `report`. The single active region in the flow chart figure is a circle centered at (150,100) and passing through point (200,100). Clicking in this region will cause a branch to symbolic label `label3`. Labels `label2` and `label4` will be visited if the user clicks anywhere outside of the explicit regions. If any labels are not defined in any of the `labesl.pl` files mentioned, they will be interpreted as URLs without translation.

The `HTML` image maps are generated and placed in directory `report` by invoking the command

```
makemap report.map report
```

4.7 Other style files

An optional style file `htmllist` has been provided which produces fancier lists in the electronic version of the document . This file defines a new LaTeX environment, `htmllist`, which causes a user-defined item mark to be placed at each new item of the list, and which causes the optional description to be displayed in bold letters. The mark is determined by the `\htmlitemmark{}` command. This command accepts either a mnenomic name for the item mark from a list of icons established at installation, or the URL of a mark not in the installation list. The `htmlitemmark` must be used inside the `htmllist` environment in order to be effective, but it may be used more than once to change the mark within the list. The item marks supplied with LaTeX2`HTML` are `BlueBall, RedBall, OrangeBall, GreenBall, PinkBall, PurpleBall, WhiteBall,` and `YellowBall`. The `htmllist` environment is identical to the `description` environment in the printed version.

An example of usage is:

```
\begin{htmllist}
\htmlitemmark{WhiteBall}
\item[Item 1:] This will have a white ball
\item[Item 2:] This will also have a white ball
\htmlitemmark{RedBall}
\item[Item 3:] This will have a red ball
\end{htmllist}
```

This will produce:

Item 1: This will have a white ball

Item 2: This will also have a white ball

Item 3: This will have a red ball

There are also optional style files `floatfig` and `wrapfig`, which provide support for the `floatingfigure` and `wrapfigure` environments, respectively. These environments allow text to wrap around a figure in the printed version, but are treated exactly as an ordinary figure in the electronic version. They are described in Goossens, et. al. [2].

4.8 Indicating Differences Between Document Versions

LaTeX2`HTML` supports the `changebar style file` package by Johannes Braams `<JLBraams@cistron.nl>`, for inserting "change bars" in a document in order to indicate differences from previous versions. This is a very primitive form of version control and there is much scope for improvement.

5 Getting LaTeX2HTML

One way LaTeX2`HTML`may be obtained is through one of the three Comprehensive TeX archive Network (CTAN) site nearest you. They are located in the United States `<ftp.shsu.edu>`, the United Kingdom `<ftp.tex.ac.uk>` and Germany `<ftp.dante.de>`.
The latest version will appear in uncompressed format in the directory,
`tex-archive/support/latex2html`.
A compressed `tar` file of the source and related files may be obtained from anonymous ftp to
ftp://www-dsed.llnl.gov/files/programs/unix/latex2html/sources/latex2html-96.1.tar.gz.
Two other ftp sites are ftp://ftp.mpn.com/pub/nikos/latex2html-96.1.tar.gz and
ftp://ftp.rzg.mpg.de/outgoing/latex2html-96.1.tar.gz. Other FTP sites nearer to you can be found using `Archie` at http://hoohoo.ncsa.uiuc.edu/archie.html or
http://www.pvv.unit.no/archie/ (faster).

Warning: Some FTP sites may not carry the latest version.

Updates and patches are posted on the LaTeX2`HTML` server at
http://www-dsed.llnl.gov/files/programs/unix/latex2html

If you get the compressed `tar` version, save it into a file, say `latex2html-96.1.tar.gz` and then extract the files with

```
% gzip -d latex2html-96.1.tar.gz
% tar xvf latex2html-96.1.tar
```

You should then have the following:

- A `README` file;
- A `Changes` file;

- The `latex2html` Perl program;
- A `texexpand` Perl program [17];
- A `latex2html.config` configuration file;
- An `install-test` installation and testing Perl script;
- A `dot.latex2html-init` sample initialization file;
- A `texinputs/` subdirectory containing various LaTeX style files;
- A `versions/` subdirectory containing code for specific HTML versions;
- A `makemap` Perl program;
- An `example/` subdirectory containing the segmentation example described in Section 3.6;
- A `.dvipsrc` file;
- A `pstogif` perl script.
- Two `pstoppm.ps` files.
- A `docs/` subdirectory containing a version of this manual.
- An `icons/` subdirectory containing various icons.
- A `styles/` subdirectory containing Perl code for handling some style files.

6 Requirements

The translator makes use of several utilities all of which are freely available on most platforms. You may use `Archie` to find the source code of any utilities you might need.[18]

The requirements for using LaTeX`2HTML` depend on the kind of translation you would like to perform, as follows:

1. **LaTeX commands but without equations, figures, tables, etc.**
    - `Perl` (version 4.0 - RCSfile: perl.c,v - Revision: 4.0.1.8 - Date: 1993/02/05 19:39:30 - Patch level: 36)

      Warning: You really DO need Perl at patch level 36 or later. Versions of LaTeX`2HTML` earlier than 0.7a4 work **only** with Perl 4 at patch level 36. Later versions of LaTeX`2HTML` work both with Perl 4 at patch level 36 and Perl 5. **No** version of LaTeX`2HTML` will work with Perl 4 at earlier patch levels.
    - `DBM` or `NDBM`, the Unix DataBase Management system.

---

[17] Written by Robert S. Thau `<rst@edu.mit.ai>`.

[18] http://www.pvv.unit.no/archie/

2. **LaTeX commands with equations, figures, tables, etc.**
   As above plus
   - `latex`,
   - `dvips` (version 5.516 or later) or `dvipsk`.
   - `gs` (Ghostscript version 2.6.1 or later).
   - The `pbmplus` OR `netpbm` library. Some of the filters in those libraries are used during the postscript to GIF conversion.
3. **Segmentation of large documentation**
   If you wish to use this feature, you will have to upgrade your LaTeX to the LaTeX 2_ε level.
4. **Transparent inlined images**[19]
   If you dislike the ugly white background color of the generated inlined images then you should get either the `netpbm` library (instead of the older `pbmplus`) OR install the `giftrans`[20] filter by Andreas Ley `<ley@rz.uni-karlsruhe.de>` Version 1.10.2 is known to work without problems but later versions should also be OK.

If `ghostscript` or the `pbmplus` (or `netpbm`) library are not available it is still possible to use the translator with the `-no_images` option.

If you intend to use any of the special features of the translator (see Page 10) then you have to include the **`html.sty`** file in any LaTeX documents that use them.

Because by default the translator makes use of inlined images in the final HTML output, it would be better to have a viewer which supports the `<IMG>` tag, such as NCSA Mosaic or Netscape Navigator. If only a character based browser is available or if you want the generated documents to be more portable then the translator can be used with the `-ascii_mode` option (See Section 2.1).

7 Installing LaTeX2HTML

To install LaTeX2HTML you **MUST** do the following:

1. **Specify where Perl is on your system.**
   In each of the files `latex2html`, `texexpand`, `pstogif`, `install-test`, and `makemap`. modify the first line saying where Perl is on your system.

   Some system administrators do not allow Perl programs to run as shell scripts. This means that you may not be able to run any of the above programs. *In this case change the first line in each of these programs from*

   ```
   #!/usr/local/bin/perl
   ```

   to

---

[19] http://melmac.corp.harris.com/transparent_images.html

[20] ftp://ftp.rz.uni-karlsruhe.de/pub/net/www/tools/giftrans.c

```
: # *-*-perl-*-*
    eval 'exec perl -S  $0 "$@"'
    if $running_under_some_shell;
```

2. **Specify where the external utilities are on your system.**
   In the file `latex2html.config` give the correct pathnames for some directories (the `latex2html` directory and the `pbmplus` or `netpbm` library) and some executables (`latex, dvips, gs`). Note that it is possible to use LaTeX2HTML even if you don't have some of the external utilities.

   While you're at it you may want to change some of the default options in the same file.

3. **Run `install-test`.**
   This Perl script will make some changes in the `latex2html` file and then check whether the pathnames to any external utilities (specified during the previous step) are correct. It will not actually install the external utilities.

   Don't forget to make `install-test` executable (using the `chmod` command) before using it if necessary. You may also need to make the files `pstogif`, `texexpand` and `latex2html` executable if `install-test` fails to do it for you.

   If for any reason you have trouble running `install-test` do not despair. Most of what it does is to do with checking your installation rather than actually installing anything. To do a **manual installation** just change the variable `LATEX2HTMLDIR` in the beginning of the file `latex2html` to point to the directory where the LaTeX2HTML files can be found.

This is enough for the main installation but you may also want to do some of the following:

- **To use the new LaTeX commands which are defined in `html.sty`:**
  Make sure that LaTeX knows where the html.sty file is, either by putting it in the same place as the other style files on your system, or by changing your TEXINPUTS shell environment variable, or by copying html.sty in the same directory as your LaTeX source file. If you are still using LaTeX V2.09, you must change the `html.sty` from `html2e.sty` to `html209.sty`. *Also make sure that TEXINPUTS includes “..” or LaTeX2HTML won't work properly.* On some systems, the command `latex` is really a shell script which sets some environment variables and calls the real LaTeX. If this is so, make sure that this shell script sets TEXINPUTS properly. This environment variable is not to be confused with the LaTeX2HTML installation variable `$TEXINPUTS` described next.

- There is an installation variable in `latex2html.config` called `$TEXINPUTS`, which tells LaTeX2HTML where to look for LaTeX style files to process. It can also affect the input path of LaTeX when called by LaTeX2HTML, unless the command `latex` is really a script which overwrites the `$TEXINPUTS` variable prior to calling the real LaTeX. This variable is overidden by the environment variable of the same name if it is set.

- The installation variable `$PK_GENERATION` specifies which fonts are used in the generation of mathematical equations. A value of "0" causes the same fonts to used as those for the default printer. Because they were designed for a printer of much greater resolution than the screen, equations will generally appear to be of a lower quality than is possible. To cause LaTeX2HTML to dynamically generate fonts that are designed specifically for the screen, you should specify a value of "1" for this variable. If you do, then check to see whether your version of `dvips` supports the command line option `-mode`. If it does, then also set the installation variable `$DVIPS_MODE` to a low resolution entry from `modes.mf`, such as `toshiba`. If `dvips` does *not* support the `-mode` switch, then leave `$DVIPS_MODE` undefined, and verify that the `.dvipsrc` file points to the correct screen device and its resolution.

- The installation variable `$AUTO_PREFIX` allows the filename prefix to be automatically uniquely set to the base filename prefix of the document being translated. It can be especially useful for multiple-segment documents.

- On certain Linux systems it is necessary to uncomment the line `use GDBM_File` in the `latex2html` and `install-test` scripts, to define the interface to the database management routines.

- The `makemap` script also has a configuration variable, `$SERVER`, which must be set to either `CERN` or `NCSA`, depending on the type of web server you are using.

- **To set up different initialization files:**
  For a "per user" initialization file, copy the file `dot.latex2html-init` in the home directory of any user that wants it, modify it according to her preferences and rename it as `.latex2html-init`. At runtime, both the `latex2html.config` file and `$HOME/.latex2html-init` file will be loaded, but the latter will take precedence.

  You can also set up a "per directory" initialization file by copying a version of `.latex2html-init` in each directory you would like it to be effective. An initialization file `/X/Y/Z/.latex2html-init` will take precedence over all other initialization files if `/X/Y/Z` is the "current directory" when LaTeX2HTML is invoked.

  > **Warning:** This initialization file is incompatible from previous versions of LaTeX2HTML. Users must either update this file from their home directories or delete it.

- **To make your own local copies of the LaTeX2HTML icons:**
  Please copy the `icons` subdirectory to a place under your WWW tree where they can be served by your server. Then modify the value of the `$ICONSERVER` variable in `latex2html.config` accordingly.

  > **Warnings:** If you cannot do that bear in mind that these icons will have to travel from Livermore, California!!! Also note that several more icons were added that were not present in previous versions of LaTeX2HTML.

- **To make your own local copy of the LaTeX2HTML documentation:**
  This will also be a good test of your installation. To do it run LaTeX2HTML on the file

`docs/manual.tex`. You will get better results if you run LaTeX first on the same file in order to create some auxiliary files.

- **To join the community of LaTeX2`HTML` users:**
  More information on a mailing list, discussion archives, bug reporting forms and more is available at
  http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html

8 Changes from Previous Versions

The previous versions of the translator and some patches are available. A detailed list of changes which includes full credits to all who have contributed bug fixes or other code is also available with the LaTeX2`HTML` distribution in the file `Changes`.

8.1 Changes upto v96

Tables and math support for HTML 3.0 Tables can now be specified in HTML, without having to generate a GIF image. Code is also available for HTML 3.0 equations, but it is not currently turned on, since no browser can handle it (yet!) (Courtesy of Marcus Hennecke `<marcush@crc.ricoh.com>`)

Font generation at screen resolutions The quality of equation bitmaps is greatly improved by enabling the $PK_GENERATION configuration variable. When this is done METAFont will be invoked through `dvips` to generate fonts more suitable to screen viewing. Ideally, this should be done by setting the `mode` switch to `dvips`, but unfortunately not all version of `dvips` support this option. For those that don't, a `.dvipsrc` file is supplied with this distribution. Sidik Isani (`<isani@cfht.hawaii.edu>`) provided this change.

Improved support for active image maps Both server and client-side maps are supported. Image maps can either be inline or external. External maps can be associated with a thumbnail image. A separate script `makemap` helps resolve external URLs in image maps. (Herb Swan `<dprhws@edp.Arco.com>`)

Improved image sharing and recycling The older version of image recycling often caused images to overwrite each other due to confusion in the bookkeeping. This has been fixed. It is also no longer necessary to keep generating the same image which is being used repeatedly in a document. Furthermore, images with thumbnails can now be recycled, as well as active image maps. Only images of the correct size are recycled. (Herb Swan)

Document segmentation Large documents can now be divided into independently processed segments. Additional command line switches provide intersegment navigation information, while automatically generated Perl files pass symbolic references and counter information between segments. (Herb Swan, assisted by Michel Goossens `<Michel.Goossens@cern.ch>`)

Command parsing made more like LaTeX's Constructs like `\hello2` are now treated as macro `hello` followed by argument '2', rather than as macro `hello2`. Added and commands. (Marcus Hennecke)

Graceful termination upon interrupt When LaTeX2HTML is interrupted by a termination signal, all child tasks are also terminated. This provides a more orderly shutdown than that offered by previous versions. (Herb Swan)

Support for textual font size changes HTML 2.1 support is now available fo ISO 10646 Unicode character sets by specifying `-html_version 2.1`. HTML 3.0 support is provided for textual font size changes (like `<SMALL>`), used in conjunction wich a generated optional style sheet. (Marcus Hennecke)

More inline math generated in `HTML` Small inline equations which can be typeset in `HTML` **are** typeset in `HTML`. This further reduces the nuber of GIF files that need to be generated.

Hypertext references can now be external The commands `htmlref` and `hyperref` now accept labels defined in an external document if the internal reference is not found. (Herb Swan)

Additional style files are now available These were provided by Herb Swan:

- **htmllist** Defines a fancier list environment. (It is the same as the `description` environment in the paper version.)
- **heqn** Redefines the `equation` environment so that equation numbers are handled in `HTML`. This causes equations in this environment to be recyclable. It also causes equation arrays to be recyclable if their equation numbers do not change from the previous run.
- **floatfig** Provides support for this environment by making it look like an ordinary figure in the electronic version.
- **wrapfig** (Same comment as for `floatfig`).
- **graphics** Defines elements of the standard LaTeX2e `graphics` package.

More support for LaTeX 2_ε Provided support for the `ensuremath` command. The last *option* to the `babel` package is interpreted as the LaTeX2HTML language style file to load. User-defined commands and environments can now have an optional argument. Stubs have been provided for `enlargethispage` and `suppressfloats`. LaTeX 2_εpackages `alltt, graphics, graphicx, color` and `epsfig` were provided. This manual itself was transformed into standard LaTeX 2_ε. (Herb Swan and Michel Goossens)

Orientation manipulation for figures The `flip=` option of `htmlimage` causes the program `pnmflip` to be called prior to the generation of the GIF file. This allows the electronic version of the image to be oriented differently than the paper version. (Herb Swan)

Better treatment of null images If for some reason an image produced a null GIF file, then no reference is made to that file and the program proceeds gracefully. Furthermore, such null images do not need to be regenerated. This would occur in `HTML` 3.0 images whose caption is enclosed in a `parbox`, for example. (Marcus Hennecke)

Independent control over top/bottom navigation panels There are now separate subroutines to control the top and bottom navigation panels. (Jens Krinke `<krinke@ips.cs.tu-bs.de>`).

Labels, equations and images in section headings All of these items are now permitted inside a section heading, and in the `\title{}` command. However, the equations may not look very good because of the size difference. (Herb Swan)

Other changes The following other numerous changes are in no particular order:

- Fixed the treatment of `@{}` expressions in HTML 3.0 tables.
- Made the `reuse` command line switch accept an option.
- Made the `.tex` filename suffix optional. (This will save a bit of typing!)
- Added a `-debug` command line switch.
- Support unbreakable spaces (~) and itallic correction.
- Fixed a bug in `pstogif` which caused `ppmquant` never to be called.
- The `\special` command can now be used outside of a figure environment. This is useful for specifying a PostScript prolog to `dvips`.
- Added the configuration variable $TEXINPUTS to point to the path of LaTeX style files that LaTeX2`HTML` should interpret.
- Added the configuration variable $DVIPS_MODE to specify:
  1. That `dvips` understands the `-mode` switch, and
  2. Which METAFont mode to use for dynamic font generation.
  3. The default output file prefix.
- Fixed up comment removal code to remove all comments during preprocessing.
- Make `$LATEX2HTMLSTYLES` a path list (and support '~' as a way of specifying the home directory).
- Replace double dashes with a single dash.
- Removed extraneous spaces in comma separated citations.
- Changed encoded '~' character to a real '~' character at end of conversion
- Eliminated the spurious newline that was appended to macro expansions.
- Improved the error reporting of `install-test`.
- Remove whitespace from beginning of unnumbered section names.
- Add other language support for TOC, figures, tables, and `\today`.
- Ignore `"|` and `"-` strings in `german` documents.

- Make `\hrule` insert less vertical whitespace.
- Added an `\htmlrule` command to draw a horizontal rule, even within a figure caption.
- The `texexpand` program now ignores anything after the

  `\end{document}`

  command, except when it is shielded by the `verbatim` environment.
- The optional caption argument is now the one which goes to the list of figures/tables, not the required argument.
- Added a `\LaTeXe` command.
- Fixed bug in the treatment of the `\circ` math symbol.
- A comment that appears on the first line of an included file is now properly removed.
- Added a call to `replace_user_references`, to allow style files to add new types of crossreferences.
- Removed the expansion of '`\\`' to '`\\ `' in verbatim.
- Figure numbers now appear in captions enclosed in parboxes.
- All deferred warning messages are now correctly reported.
- The "`scale=`" option of `\htmlimage` now works.

8.2 Changes up to v95

Much improved inlined equation baseline alignment! (Thanks to Mark Segal `<segal@spud.asd.sgi.com>`) Inlined equation bitmaps are now aligned correctly depending on whether they contain subscripts, superscripts etc.

Support for internationalization (Thanks to Martin Boyer `<gamin@ireq-robot.hydro.qc.ca>`) A global variable `LANGUAGE_TITLES` can now be used to change the language in which some section titles (eg "Table of Contents") are printed. It is also very easy to add support for more languages.

Compatibility with Perl 5

More efficient implementation There has been a major overhaul of the way the source text is parsed and analyzed in order to reduce the memory requirements of this process.

This has been achieved by spawning off separate Unix processes to deal with each of the `input`'ed or `include`'d files. As each process terminates all the space that it used is reclaimed. Asynchronous communication between processes takes place using the Unix DataBase Management system (DBM or NDBM) which should be present on your system.

To take advantage of these changes, it is necessary to split the source text into more than one files which can be assembled using the LaTeX`input` or `include` commands.

"Off-line" Image Generation Two new options `-no_images` and `-images_only` allow "off-line" image conversion. The advantage of using these options is that the translation can be allowed to finish even when there are problems with image conversion. In addition it may be possible to fix manually any image conversion problems and then run LaTeX2HTML again just to integrate the new images without having to translate the rest of the text. More instructions on how to do this are included in the "Troubleshooting" section of the LaTeX2HTML manual.

Can now use either the `pbmplus` or the `netpbm` libary. If `netpbm` is used then it is no longer necessary to get and install `giftrans` in order to generate transparent inlined images.

Also, a new option `map=<`*`image map URL`*`>` in the command `htmlimage` can turn an included postscript image into an active image map.

New options

-no_images Do not attempt to produce any inlined images. The missing images can be generated "off-line" by restarting LaTeX2HTML with the option `-images_only`.

-images_only Try and convert any inlined images that were left over from previous runs of LaTeX2HTML.

-no_reuse Do **not** reuse images generated during previous translations. (This will enable the initial interactive session during which the user is asked whether to reuse the old directory, delete its contents or quit)

-no_subdir Place the generated HTML files in the current directory. The default behaviour is to create (or reuse) another file directory.

-ps_images Use links to external postscript images rather than inlined GIF images.

Several small changes and bug fixes

- It is no longer necessary to get `Giftrans` if `NETPBM` is available
- Fixed problems with support for `german.sty`
- Fixed problems with multiple bibliographies. Each bibliography is now treated as a separate section.
- Added support for "dotless i's and j's".
- Fixed to resolve figure and table numbers when captions contain accented characters.
- Added support for hierarchical indices with duplicate index keys
- Fixed problem with HTML encodings of ISO-LATIN1 characters creeping into converted images of figures and tables.
- Added new global variable `PAPERSIZE` to make it easier to change the default behavior when converting large images
- A new configuration variable (`TRANSPARENT_IMAGES`) can be used to stop any inlined images generated from "figure" environments from being transparent.
- Fixed problem which occurs when `getcwd.pl` is not part of the Perl library.

- The option `-address ""` is now valid.
- Fixed a problem with tables of contents.
- Fixed problem with the use of the `finger` command when trying to find out the name of the user.
- Added some support for `tabbing` environments.
- HTML heading elements no longer contain other markup.
- Fixed problem with optional argument in citation commands
- Added some more support for LaTeX2e.
- Fixed problem with displayed equations forcing all the remaining equations to be displayed.
- Fixed bug with converting postscript images containing more than 256 colors.

8.3 Changes upto v0.6.2

Image Conversion

- The `pstogif` script has been rewritten in Perl and it now accepts more options for specifying color depth, scale factors and pixel density.
- LaTeX `figure` and other environments are now processed in 24-bit color.
- Figure and table captions are now converted into HTML rather than becoming part of the inlined image. This means that any cross-references in the captions will become active.
- It is now possible to control for each generated image:
  - its size
  - whether it should be inlined or left as an external image
  - if left as an external image whether it should be accessible via a **generated thumbnail sketch** or a textual hypertext link

  These options are available from a new LaTeX command defined in html.sty called `htmlimage`.
- Equations and other inlined images are converted into transparent GIFs rather than XBMs. **This makes it possible to reduced their storage requirements to about 1/6th of what was previously required!**
- The default size of equations and other small inlined images can be changed by setting the variable `MATH_SCALE_FACTOR` in the configuration files.
- The default size of figures, tables, and other large inlined images can be changed by setting the variable `FIGURE_SCALE_FACTOR` in the configuration files.
- Dvips is now called with the `-M` option which stops it from invoking `METAFONT`.

Optimization

- Some effort has gone into reducing the amount of memory required during conversion. **The impact of these changes can be very significant**. In particular, files containing large numbers of LaTeX commands (eg `.aux` or `.bib`) files are handled much better.

Backwards Incompatible Changes

- The option `-allbitmaps` has been removed. This is no longer necessary as LaTeX2HTML can now generate transparent GIF images.
- The definition of the command `htmladdnormallink` in the file `html.sty` has changed and the URL will no longer appear as a footnote in the paper (DVI) version. The translation of this command into HTML is **not** affected.

  The previous functionality can be obtained with a new command `htmladdnormallinkfoot` which will add the URL as a footnote in the paper version.
- The script `pstoxbm` is no longer distributed as it is not used.

Bug Fixes

- The `install-test` script now recognizes version numbers correctly and gives better warnings. It also makes executable any scripts that need to be so.
- Stopped using the `nslookup` program to try and guess e-mail addresses (they were used in signing each generated page). This has been replaced with a simple call to `finger`.
- Fixed problem which could cause "Parameter overflow" in TeX during image conversion.
- The `texexpand` script has been changed extensively. Some nasty looping problems are now avoided, and the tracking down of included files has been improved.
- Some reported incompatibilities with some Unix shells (bash,OSF) have been fixed.
- Displayed equations now appear correctly on separate lines and are right justified.
- Fixed bug with nested environments.
- Fixed problem which caused the wrong numbers to be assigned to sections with the same titles when the `-show_section_titles` is used (thanks to Brian Toonen `<toonen@mcs.anl.gov>`).
- Fixed problem with the `hyperref` command which caused the wrong "hyperized" text to appear in the final document.
- `item` optional arguments can now contain one level of nesting eg `item[[First Choice]]`.
- Fixed problems with the "image caching and reuse" mechanism which avoids converting images unnecessarily..

Other Changes

- LaTeX2HTML now generates `<meta>` HTML tags which can be used by indexing scripts[21] which generate information for the ALIWEB[22] search and retrieval tool. The information in the `<meta>` tags contains the title of each separate HTML file.
- A new configuration variable `DEBUG` can be used to preserve intermediate files for debugging.
- Some problems with respect to compatibility with the HTML2.0 standard have been addressed. No more "unquoted attribute value literals".
- All the navigation icons are now "lynx friendly"; their `ALT` attribute now has a meaningful value.
- A new LaTeX command `htmlref` makes it a lot simpler to create hypertext links intended only for the HTML version of the document.
- `DVIPSK` is now recognized by the installation script
- Added support for the `changebar.sty` file by David B. Johnson (dbj@titan.rice.edu).
- It is no longer necessary to add style file names into the `DONT_INCLUDE` variable as LaTeX`2HTML` now does not attempt to translate file included files ending in `.sty`.
- The small invisible bitmaps that used to mark anchors have been replaced with the invisible character ` `.
- It is no longer necessary to use full path names when including external postscript files.
- If a file `.latex2html-init` is found in the "current directory" then this will be loaded automatically after loading the other default configuration files.
- Changed the naming convention of generated HTML files. The "top" document is `<FILE>.html` as before but all other "nodes" are named `nodeN.html` where `N` is an integer. Also, the file containing the footnotes is now called `footnode.html`.
- A special extension to LaTeX`2HTML` for those who use the Harvard style for references can be obtained from http://www.arch.su.edu.au/~peterw/latex/harvard/.

8.4 Changes upto v0.5.3

- The files `LATEX2HTMLDIR/styles/german.perl` and `LATEX2HTMLDIR/styles/makeidx.perl` have been fixed so that they are consistent with changes in the main script.
- A problem with disappearing spaces after some equations and other environments has been fixed.

---

[21] http://www.ai.mit.edu/tools/site-index.html

[22] http://web.nexor.co.uk/aliweb/bin/aliwebsimple.pl

8.5 Changes upto v0.5.1

Navigation Panel The navigation panel (see section 3.8) is now fully configurable and much better looking.

Automatic Signatures The signatures at the bottom of each page are now constructed using `nslookup` (Thanks to Alberto Accomazzi `<alberto@cfa.harvard.edu>`)

Numerous fixes including...

- "dead" `next_page` buttons when the `-split` option was used, and
- internal LaTeX2HTML markers appearing in section titles.

8.6 Changes upto v0.4

Accents and Special Characters LaTeX accent commands, special characters and accent commands defined in `german.sty` are translated to equivalent ISO-LATIN-1 characters when that is possible (partly thanks to code supplied by Franz Vojik `<vojik@de.tu-muenchen.informatik>`).

Auto-loading of Style-Specific Code The translator now supports a mechanism for including Perl code extensions (see Section 2.2.1) which are specific to particular style files.

This mechanism will help to keep the core script smaller as well as make it easier for others to contribute and share solutions on how to translate specific style files. The current distribution includes the files `german.perl`, `french.perl`, `html.perl` and `makeidx.perl`.

Installation and Testing To make it easier to install the translator for more than one user there is now a configuration file (LATEX2HTMLDIR/latex2html.config) which contains installation specific information and default values for the various options. This makes it unnecessary to have a `HOME/.latex2html-init` for each user. If a user does have a `HOME/.latex2html-init` then this will be loaded after LATEX2HTMLDIR/latex2html.config.

A new Perl script (`install-test`) is now available with the distribution which will install the translator, and then perform some tests on the availability of some external programs giving, appropriate warnings.

Navigation Panel Extensions The following new links have been added to the navigation panel which appears in each page (where this is appropriate):

- a link to the next "logical" page (this allows reading each page in the same order as with a paper-based version - as opposed to structure navigation)
- a link to the previous "logical" page
- a link to the table of contents
- a link to the index

HTML Style File Extensions The HTML style file `html.sty` now includes definitions of new environments for:

> **Inclusion of Raw HTML** A new LaTeX environment `rawhtml` allows arbitrary HTML tags to be included in a LaTeX document. This is useful for taking advantage of HTML+ features as they become available (e.g. interactive forms). The HTML commands are ignored when producing the DVI version of the document.
>
> **Conditional Text** The new environments `latexonly` and `htmlonly` allow their contents to appear only in final DVI or in the HTML version of a document respectively.
>
> A new command `hyperref` can be used to specify the way in which cross-references should be shown in the DVI version and the HTML version of the document.

Right Justification of Equations The translator adds enough whitespace at the beginning of each equation bitmap to push it to the right-hand margin. The width of each line can be set in the configuration file using the variable `LINE_WIDTH`.

Inlined Images All inlined images are now generated by calling `latex` and `dvips` only once at the end of the conversion process (but each generated image still has to be filtered individually). Also, a warning is given if an image cannot be converted.

Numeric Labels The original (LaTeX generated) numeric labels are used instead of the arrow navigation icon for cross-references where possible. To do this the translator uses the information in the `aux` file which is generated by LaTeX. If the `aux` file is out of date a warning is given.

New Options

> `-auto_navigation` This puts a navigation panel at the top of each page as usual. But if the page exceeds a user-settable number of words (the default is 450 words) then a navigation panel is also placed at the end of the page. This option is active by default.
>
> `-index_in_navigation` Adds a link to the index in the navigation panel.
>
> `-contents_in_navigation` Adds a link to the table of contents in the navigation panel.
>
> `-next_page_in_navigation` Adds a link to the next "logical" page in the navigation panel.
>
> `-previous_page_in_navigation` Adds a link to the previous "logical" page in the navigation panel.
>
> `-bottom_navigation` This puts a navigation panel at the bottom of each page.
>
> `-top_navigation` This puts a navigation panel at the top of each page (the default).

`-show_section_numbers` This allows each section (page) title to be numbered in the same way that it would be numbered by LaTeX. This requires an up to date `aux` file (generated by running LaTeX) and that each title is unique. If the `aux` file is not up to date then a warning is given. Unfortunately if the title contains inlined images the numbering for that title will be lost.

`-reuse` This allows images generated during previous invocations of the translator to be "reused" without going through the initial interactive session. The same behavior is obtained by setting the variable `REUSE` to 1 (the default) in the configuration file. Note that images which may depend on contextual information (e.g. numerical labels) cannot be reused and are always re-generated.

Bug Fixes and Minor Changes

- The links from the table of contents in single page documents (created using the option `-split 0`) is now working as expected.
- Newlines are translated to the `<BR>` tag.
- Fixed some problems in dealing with new command macros.
- `texexpand` now respects commented `input` and `include` commands. Also (thanks to Franz Vojik `<vojik@de.tu-muenchen.informatik>`), it now looks in subdirectories when expanding files.
- The ALIGN attribute of inlined images is now BOTTOM instead of TOP.
- Fixed problem with the environment variable `TEXINPUTS`.
- Added some support for optional user-defined labels (bullets) in list environments.

More details on all the changes are available in the file `LATEX2HTMLDIR/Changes`.

8.7 Changes upto v0.3.1

These changes are mostly due to patches contributed by Robert S. Thau `<rst@ai.mit.edu>`:

- Nested environments with the same name are now dealt with properly.
- Commands that are passed to LaTeX for processing which have environments in their arguments (e.g. a `parbox` command which an `itemize` environment as an argument) are now processed correctly. A general mechanism for users to specify the syntax of commands that should be passed to LaTeX is described in Page 10.
- Fixed a problem with recognizing the `special` command.
- Fixed bug in the generation of the index.

8.8 Changes upto v0.3

Image Recycling Images for equations, tables, figures, special characters etc. generated by the translator are recognized during subsequent runs. The user is then asked whether old images should be reused (or whether the old images should be deleted and regenerated). This offers tremendous improvements in speed after an initial successful translation.

Cross-References Between (Local or Remote) Documents Cross-references between two or more documents (possibly on remote locations) can be established via symbolic labels which are independent of the physical realization of these documents.

Such cross-references will be maintained with a simple re-translation[23] even after one or more of the documents have been broken into different physical parts or moved.

The mechanism is based on the new commands (see Section 4.3) `externallabels` and `externalref` which are an extension of the simple `label-ref` pairs.

New Options

`external_images` This provides hypertext links to where generated images (for equations, tables, figures etc) are stored externally. (The default is to "inline" generated images in the main body of the text.)

`ascii_mode` This switches all the navigation icons to their ascii equivalents. Also generated images are stored externally as with the `external_images` option above. The `ascii_mode` option makes documents more portable as it allows them to be viewed on browsers that do not support inlined images.

Special Command Style File A style file `html.sty` is now included in the distribution. This contains the definitions (syntax) of some special LaTeX commands mainly for providing external hypertext links. This should be included in LaTeX files that use the any hypermedia extensions (see Section3).

Minor Changes and Bug Fixes

- Equations, equation arrays and theorems are now numbered correctly even when they are individually passed to LaTeX for processing.
- Fixed problem with `label` commands appearing in section headings.
- Fixed problems with verbatim environments, bibliography items, generated file names, the options `nolatex` and `no_navigation`, the `thanks command`, the name of an HTML tag (`<HEAD>`),
- The appearance of footnotes has been improved.
- Some inconsistencies in the `pstoxbm` and `pstogif` scripts have been fixed.
- Parts of the documentation have been rewritten, restructured and some new sections have been added.

The following were contributed by Robert S. Thau `<rst@ai.mit.edu>`:

New Texexpand This fixes problems with the standard version and handles the inclusion of style files which need to processed by the translator. Appropriate modifications to the main script were also made to work with new version of `texexpand`.

---

[23] This is true for documents under the same server but for remote documents a little more is required (see the example in Section3.5.1)

Handling of Raw TEX Added support for simple raw TeX commands such as `special` and simple instances of `def`[24]. For the messier cases, the definition is scooped up and moved to the preamble. This allows the translator to handle the simple, but nonstandard, postscript figure inclusion macros as well as an awful lot of other stuff done with gratuitous `defs`.

More Options

`dont_include` This can be used to specify style files that should not be included in the translation.

Other Changes

- Added support for nested math mode expressions and general list environments in order to handle *the document from Hell!!!*.
- Fixed problems in the translation of bibitems, the substitution of macro definitions, and the processing of unrecognized commands in the preamble.

8.9 Changes upto v0.2

- Added a command line option to switch off the navigation links at the top of each page.
- The navigation icons are now part of the distribution.
- Added a customizable separator between the main body of the text in a page and the child links from that page.
- The order of the navigation keywords at the top of each page is the same as that of the navigation icons.
- The arguments of `verbatim` environments are now translated into fixed width fonts.
- A warning is given if a `bbl` (bibliography) file is needed but not found.
- The installation is now (mostly) done by setting variables in just one file.
- The `pstoxbm` script now uses environment variables set in the initialization file.
- Fixed bug in translating sequences of special HTML characters (e.g. &, <, etc.)
- Fixed bug in the handling of the `$$`-form of the dislay math environment.
- Fixed bug in the handling of the *-forms of environments.
- Added sections on how to embed hyperlinks in a LaTeX document (see Page 10) and on how to extend the translator (see Page 8) in the documentation.

---

[24] Where "simple" is defined roughly as "they could have used `newcommand`, but didn't".

8.10 Changes upto v0.1.1

- Fixed bug about empty lines being inserted in environments that cannot tolerate them (e.g. `math`).
- Changed the format of inlined images coming back from LaTeX from GIF to XBM. This looks better on grayscale and color monitors.
- Fixed problem with commands being passed on to LaTeX after their arguments had been translated (this affects the commands `psfig`, `fbox`, `framebox`, and `parbox`).

9 Known Problems

Here are some of the problems of the current version:

Correctness and Efficiency The translator cannot be guaranteed to perform as expected. Several aspects of the implementation need optimization and improvement. Apart from possible bugs the translator may place heavy demands on your resources.

Unrecognized Commands and Environments Unrecognized commands are ignored and any arguments are left in the text. Unrecognized environments are passed to LaTeX and the result is included in the document as one or more inlined images.

Cross-references References in environments that are passed to LaTeX for processing (e.g. a `cite`, or a `ref` command), are not processed correctly. `label` commands are handled correctly.

Order-Sensitive Commands Commands which affect global parameters during the translation, and are sensitive to the order in which they are processed may not be handled correctly. In particular, counter manipulation (e.g. `newcounter, setcounter, stepcounter`, etc) commands may cause problems.

Index The translator generates its own index by saving the arguments of the `index` command. The contents of the `theindex` environment are ignored.

New Definitions New definitions (`newcommand`, `newenvironment`, `newtheorem` and `def`), will not work as expected if they are defined more than once. Only the last definition will be used throughout the document.

Scope of declarations and environments If the scope of a declaration or environment crosses section boundaries, then the output may not be as expected, because each section is processed independently.

Math mode font size changes Math mode font changes made outside the math mode are not honored. Thus the two equations in

```
$a_b$ and {\LARGE $a_b$}
```

would come out looking the same. The trick is to write

```
$a_b and $\mbox{\LARGE $a_b$}$.
```

10 Troubleshooting

Here are some curable symptoms:

Cannot run any of the Perl programs If your Perl installation is such that Perl programs are not allowed to run as shell scripts you may be unable to run `latex2html`, `texexpand` `pstogif` and `install-test`. In this case change the first line in each of these programs from

```
#!/usr/local/bin/perl
```

to

```
: # *-*-perl-*-*
    eval 'exec perl -S  $0 "$@"'
    if $running_under_some_shell;
```

The `install-test` script gives uninformative error messages If for any reason you have trouble running `install-test` do not despair. Most of what it does is to do with checking your installation rather than actually installing anything. To do a **manual installation** just change the variable `LATEX2HTMLDIR` in the beginning of the file `latex2html` to point to the directory where the LaTeX2HTML files can be found.

Also, make sure that the files `pstogif`, `texexpand` and `latex2html` are executable, and if necessary use the Unix `chmod` command to make them executable.

It just stops Check the style files that you are using. It is likely that you are using a style file which contains raw TeX commands. In such a case start LaTeX2HTML with the option `-dont_include <style file name>`. Alternatively, add the name of the style to the variable `DONT_INCLUDE` in your `HOME/.latex2html-init` file. If you don't have such a file then create one and add the lines:

```
$DONT_INCLUDE = "$DONT_INCLUDE" . ":<style file name>";
1;  # This must be the last line
```

Another reason why LaTeX2HTML might stop is that the LaTeX source file itself contains raw TeX commands. In this case you may put such commands inside a `latexonly` `(see Section` 3.3) environment.

Perl cannot parse the `latex2html` script Update your Perl to patch level 36. You can check which version of Perl you are using by invoking Perl with the `-v` option. Earlier versions of Perl than that shown above have caused problems due to tighter control over syntax.

It crashes (dumps core) as soon as it starts Update your Perl 4 to patch level 36 or later (Perl 5).

You can check which version of Perl you are using by invoking Perl with the `-v` option.

While you wait for your technical support people to upgrade Perl you could try invoking Perl from within LaTeX2HTML with the `-d` (debug) option. Then, when LaTeX2HTML starts, it will immediately fall into the Perl debugger. To continue just press `c <CR>`.

`dvips` complains about incorrect arguments Please use a version which supports the command line options `-M -S, -o and -i`. “Recent” versions at least after 5.516 do support them.

It gives an `Out of memory` message and dies If you are using version LaTeX2`HTML` 0.7 or later try splitting your source file into more than one files using the LaTeX commands `input` or `include`. Also, use the `-no_images` option.

As a last resort you may consider increasing the virtual memory (swap space) of your machine. As an indication of what you might be able to do on your machine, a very long book (about 1000 printed pages) required about 24MB of RAM and over 150MB of swap space to convert on a local Sun Sparc ELC running SunOS 4.1.3.

It gives “dbm” related error messages LaTeX2`HTML` 0.7 and later requires the Unix DataBase Management system (DBM or NDBM) in order to run. This is usually part of each Unix operating system but if you don't have it then you may need to get it. Use Archie[25] to find one.

The `verb"ABC"` command doesn't work This is a nasty bug. Please use any characters other than quotes eg `verb+ABC+`

Cannot get the “tilde” (`~`) to show The trick here is to use the command `\~{}`.

Alternatively it is possible to use something like

```
\htmladdnormallink{mylink}%
  {\begin{rawhtml}http://host/~me/path/file.html\end{rawhtml}}
```

or

`\htmladdnormallink{mylink}{http://host/\%7Eme/path/file.html}`

Warning: Some browsers may not be able to interpret the `%7E` as a “tilde” character.

Macro definitions don't work correctly As mentioned in other places plain TeX definitions cannot be converted. But you may also have problems even when using LaTeX definitions (with `newcommand` and `newenvironment`) if such definitions make use of *sectioning or verbatim* commands. These are handled in a special way by LaTeX2`HTML` and cannot be used in macro definitions.

In general the macro handling mechanism is inefficient and very fragile. Avoid using macros if possible.

`input` commands There is a bug in the expansion of `input` commands which causes a problem when more than one `input` command appear on the same line. There is no quick fix other than suggesting that you insert a newline after `input` commands in the source .tex files.

`input` commands in verbatim environments These cause problems. There is no fix yet.

---

[25] http://www.pvv.unit.no/archie/

Optional arguments in description environments If you have optional arguments for the `item` command in a description environment containing nested "]" characters then these may not show up correctly. To avoid the problem enclose them in {}'s eg `\item[{[nested [angle [brackets] are ok]]}]`

LaTeX2HTML behaves differently even when you run it on the same file If you notice any strange side-effects from previous runs of LaTeX2HTML try using the option `-no_reuse` and choose `(d)` when prompted. This will clear any intermediate files generated during previous runs. Note that this option will disable to image reuse mechanism.

Cannot convert postscript images which are included in the LaTeX file It is likely that the macros you are using for including postscript files (e.g. `epsffile`) are not understood by LaTeX2HTML. To avoid this problem enclose them in an environment which will be passed to LaTeX anyway e.g.

```
\begin{figure}
\epsffile{<postscript file name>}
\end{figure}
```

Another reason why this might happen is that your shell environment variable `TEXINPUTS` is undefined. This is not always fatal but if you have problems you can use full pathnames for included postscript files (even when the postscript files are in the same directory as the LaTeX source file). Alternatively try setting TEXINPUTS to ".::". With some TeX and LaTeX installations setting TEXINPUTS to ".::" may cause problems in the normal operation of LaTeX. If you get errors such as LaTeX complaining that it can no longer find any style files then you must set TEXINPUTS to `"<path to your LaTeX installation>:."` if you want to use both LaTeX and LaTeX2HTML.

Some of the inlined images are in the wrong places This happens when any one of the inlined images is more than a page (paper page) long. This is sometimes the case with very large tables or large postscript images. In this case you can try specifying a larger paper size (eg "a3", "a2" or even "a0") instead of the default ("a4") using the LaTeX2HTML variable `PAPERSIZE` in the file `latex2html.config`.

Another reason why this may happen is that by default the `dvips` program reverses the postscript pages it generates. If your `dvips program` behaves in this way try changing the line `$DVIPS = "dvips";`

to

`$DVIPS = "dvips -r0";`

in the file `latex2html.config`.

Unacceptable quality of converted images Try changing the size of the image (See Section 4.3).

The bibliographic references are missing Run `latex` and then `bibtex` on the original source file in order to generate a `bbl` file. LaTeX2`HTML` requires a `bbl` in order to generate the references.

The labels of figures, tables or equations are wrong This can happen if you have used any figures, tables, equations or any counters inside conditional text i.e. in a `latexonly` or a `htmlonly` environment.

Problems after changing the configuration files Please make sure that the last line in the configuration files (ie `.latex2html-init` and `latex2html.congif`) is:

```
1; # This is the last line
```

This is a Perl quirk...

Problems when producing the DVI version If you are using any of the new LaTeX commands which are defined in the `html.sty` file make sure that `html.sty` file is included e.g. as one of the optional arguments to the `documentstyle` command.

Of course you also have to make sure that LaTeX knows where the html.sty file is, either by putting it in the same place as the other style files on your system, or by changing your TEXINPUTS shell environment variable[26].

Some of the fonts are translated incorrectly There is a fault in way the LaTeX scoping rules have been interpreted in LaTeX2`HTML`. Consider this:

```
\ttfamily fixed-width font.
\begin{something}
nothing here
\end{something}
default font.
```

When processed by LaTeX, the effect of the `tt` command is delimited by the beginning of the environment "something" so that "default font" will appear in the default font. But LaTeX2`HTML` will not recognize "something" as a delimiter and "default font" will appear in the wrong font.

To avoid this problem until it is fixed you may delimit the scope of some commands explicitly using `{}`'s i.e.

```
\texttt{fixed-width font}.
\begin{something}
nothing here
\end{something}
default font.
```

---

[26] If don't know how to do either of these things, copy (or link) html.sty to the directory of your LaTeX document...

Using `Ghostscript 3.X` you can no longer generate inlined images for equations If you have a version of LaTeX2HTML later than 0.6.1, go to the LaTeX2HTML directory and run `install-test` again. This should fix it.

With earlier versions of LaTeX2HTML you can fix it by replacing the file `pstoppm.ps` in the LaTeX2HTML directory with a newer one that accompanies `Ghostscript 3.X`. Alterhatively you can avoid using pstoppm.ps by changing the way `GS` is invoked in the file `pstogif`, using something like

```
open (GS, "|$GS -q -sDEVICE=ppmraw  -sOutputFile=$base.ppm $base.ps");
```

Cannot get it to generate inlined images Try a small test file e.g.

```
% image-test.tex
\documentstyle{article}
\begin{document}
Some text followed by \fbox{some more text in a box}.
\end{document}
```

You should see something like:

```
This is LaTeX2HTML Version  (Wed Dec 1 1993) by Nikos Drakos,
Computer Based Learning Unit, University of Leeds.

OPENING /usr/cblelca/nikos/scripts/tex2html/tests/image-test.tex

Reading ....
Translating ...0/1.....1/1......
Generating images using latex ...
This is TeX, C Version 3.14t3
12222_images.tex
LaTeX Version 2.09 <7 Dec 1989>


Generating postscript images using dvips ...
This is dvips 5.521 Copyright 1986, 1993 Radical Eye Software

' TeX output 1993.12.03:1050' -> 12222_image
(-> 12222_image001) <tex.pro>[1]
Initializing... done.
Ghostscript 2.6.1 (5/28/93)
Copyright (C) 1990-1993 Aladdin Enterprises, Menlo Park, CA.
  All rights reserved.
Ghostscript comes with NO WARRANTY: see the file COPYING for details.
GS>GS>Writing 12222_image001.ppm
GS>pnmcrop: cropping 119 rows off the top
pnmcrop: cropping 961 rows off the bottom
pnmcrop: cropping 208 cols off the left
```

```
pnmcrop: cropping 484 cols off the right

Doing section links .....
Done.
```

If there is a problem somewhere during the conversion from postscript to GIF you can try to do it manually so that you can find out where the problem is. Here is one way to do it (Please use the `pstoppm3.ps` file instead of `pstoppm.ps` if your version of ghostscript is later than 3.0):

```
cblelca% latex image-test.tex
This is TeX, C Version 3.14t3
(image-test.tex
LaTeX Version 2.09 <7 Dec 1989>
(/usr/TeX/tex.lib/inputs//paper.sty
Document Style `paper' <28 Nov 89>.
(/usr/TeX/tex.lib/inputs//pap11.sty) (/usr/TeX/tex.lib/inputs/\-doublespace.sty)
(/usr/TeX/tex.lib/inputs//smaller.sty)) (/usr/TeX/tex.lib/inputs\-/psfig.sty
psfig/tex 1.9
)
No file image-test.aux.
[1] (image-test.aux) )
Output written on image-test.dvi (1 page, 652 bytes).
Transcript written on image-test.log.
cblelca% dvips -o image-test.ps image-test.dvi

This is dvips 5.519 Copyright 1986, 1993 Radical Eye Software
' TeX output 1993.11.12:1412' -> image-test.ps
<tex.pro>. [1]
cblelca% gs -dNODISPLAY pstoppm.ps
Initializing... done.
Ghostscript 2.6.1 (5/28/93)
Copyright (C) 1990-1993 Aladdin Enterprises, Menlo Park, CA.
  All rights reserved.
Ghostscript comes with NO WARRANTY: see the file COPYING for details.
GS>(image-test) ppm1run
Writing image-test.ppm
GS>quit
cblelca% pnmcrop image-test.ppm >image-test.crop.ppm
pnmcrop: cropping 61 rows off the top
pnmcrop: cropping 110 rows off the bottom
pnmcrop: cropping 72 cols off the left
pnmcrop: cropping 72 cols off the right
cblelca% ppmtogif image-test.crop.ppm >image-test.gif
```

STILL cannot get it to generate inlined images for equations etc. If you have no problems with the `image-test.tex` file but you still cannot convert the images in some of your files have a look in the directory of the generated HTML files for two files `images.tex` and `images.log`. Do you notice anything unusual in them? Copy `images.tex` in the directory of your original LaTeXfile and run `latex` on `images.tex`. Can you see any errors in `images.log`? If yes can you fix `images.tex` to get rid of the errors? After fixing images.tex you can put it back in the directory of HTML files created by LaTeX`2HTML` and run LaTeX`2HTML` on the original document using the option `-images_only`.

If you get into a mess try running LaTeX`2HTML` with the options `-no_reuse` and `-no_images` eg

```
cblipca% latex2html -no_reuse -no_images test.tex
This is LaTeX2HTML Version 95 (Tue Nov 29 1994) by Nikos Drakos,
Computer Based Learning Unit, University of Leeds.

OPENING /tmp_mnt/home/cblelca/nikos/tmp/test.tex
Cannot create directory /usr/cblelca/nikos/tmp/test: File exists
(r) Reuse the images in the old directory OR
(d) *** DELETE *** /usr/cblelca/nikos/tmp/test AND ITS CONTENTS OR
(q) Quit ?
:d

Reading ...
Processing macros ....+.
Reading test.aux ....................
Translating ...0/1........1/1.....
Writing image file ...

Doing section links .....

*********** WARNINGS ***********

If you are having problems displaying the correct images with Mosaic,
try selecting "Flush Image Cache" from "Options" in the menu-bar
and then reload the HTML file.

Done.
```

Then try to have a look in the file `images.tex` (as described earlier) and perhaps fix it. Once you are happy that `images.tex` is OK run LaTeX`2HTML` again with the option `-images_only`.

The options `no_reuse, no_images` and `images_only` are available with LaTeX`2HTML` version 0.7 or later.

Some problems in displaying the correct inlined images, may be due to the image caching mechanisms of your browser. With some browsers a simple "Reload Current Document" will be enough to refresh the images but with others (eg Mosaic) you may need to request for the cache to be refreshed. With Mosaic try selecting "Flush Image Cache" from "Options" in the menu-bar and then reload the HTML file.

It cannot do slides, memos, etc, ... If you use `slitex` you can go a long way just by replacing the `slides` argument of the `documentstyle` command with something like `article` just before using LaTeX2`HTML`. One problem may be that all your slides will end up in the same HTML file. If you use `lslide.sty` you may get much better results (use Archie[27] to find this or any other style files).

11 Support and More Information

A LaTeX2`HTML` mailing list has been set up at the Argonne National Labs (thanks to Ian Foster `<itf@mcs.anl.gov>` and Bob Olson `<olson@mcs.anl.gov>`). The LaTeX2`HTML` mailing list archive[28] is available.

To join send a message to:
`latex2html-request@mcs.anl.gov`
with the contents
`subscribe`

To be removed from the list send a message to:
`latex2html-request@mcs.anl.gov`
with the contents
`unsubscribe`.

12 General License Agreement and Lack of Warranty

This software is distributed in the hope that it will be useful but **without any warranty**. The author(s) do not accept responsibility to anyone for the consequences of using it or for whether it serves any particular purpose or works at all. No warranty is made about the software or its performance.

Use and copying of this software and the preparation of derivative works based on this software are permitted, so long as the following conditions are met:

- The copyright notice and this entire notice are included intact and prominently carried on all copies and supporting documentation.
- No fees or compensation are charged for use, copies, or access to this software. You may charge a nominal distribution fee for the physical act of transferring a copy, but you may not charge for the program itself.
- If you modify this software, you must cause the modified file(s) to carry prominent notices (a Change Log) describing the changes, who made the changes, and the date of those changes.

---

[27] http://www.pvv.unit.no/archie/

[28] http://cbl.leeds.ac.uk/nikos/tex2html/doc/mail/mail.html

- Any work distributed or published that in whole or in part contains or is a derivative of this software or any part thereof is subject to the terms of this agreement. The aggregation of another unrelated program with this software or its derivative on a volume of storage or distribution medium does not bring the other program under the scope of these terms.

This software is made available **as is**, and is distributed without warranty of any kind, either expressed or implied. In no event will the author(s) or their institutions be liable to you for damages, including lost profits, lost monies, or other special, incidental or consequential damages arising out of or in connection with the use or inability to use (including but not limited to loss of data or data being rendered inaccurate or losses sustained by third parties or a failure of the program to operate as documented) the program, even if you have been advised of the possibility of such damages, or for any claim by any other party, whether in an action of contract, negligence, or other tortious action.

The LaTeX2`HTML` translator is written by Nikos Drakos, Computer Based Learning Unit, University of Leeds, Leeds, LS2 9JT. Copyright ©1993, 1994, 1995. All rights reserved.

13 Credits

Several people have contributed suggestions, ideas, solutions, support and encouragement. Some of these are Roderick Williams, Ana Maria Paiva, Jamil Sawar and Andrew Cole here at the Computer Based Learning Unit.

The idea of splitting LaTeX files into more than one components linked with hyperlinks was first implemented in Perl by Toni Lantunen at CERN. Thanks to Robert Cailliau `<cailliau@cernnext.cern.ch>` of the WorldWide Web Project also at CERN for giving me access to the source code and documentation (although no part of the original design or the actual code has been used).

Robert S. Thau `<rst@edu.mit.ai>` has contributed the new version of `texexpand`. Also, in order to translate the *document from hell* (!!!) he has extended the translator to handle `def` commands, nested math-mode commands, and has fixed several bugs.

The `pstogif` script uses the `pstoppm.ps` postscript program originally written by Phillip Conrad (Perfect Byte, Inc.) and modified by L. Peter Deutsch (Aladdin Enterprises).

The idea of using existing symbolic labels to provide cross-references between documents was first conceived during discussions with Roderick Williams `<rodw@cbl.leeds.ac.uk>`. Eric Carroll `<eric@ca.utoronto.utcc.enfm>` suggested providing a command like `hyperref`. Franz Vojik `<vojik@de.tu-muenchen.informatik>` provided the basic mechanism for handling foreign accents. The `-auto_navigation` option was based on an idea by Todd `<little@com.dec.enet.nuts2u>`. Axel Belinfante `<Axel.Belinfante@cs.utwente.nl>` provided the code in the `makeidx.perl` file as well as numerous suggestions and bug reports.

Verena Umar `<verena@edu.vanderbilt.cas.compsci>` (Computer Science Education Project) has been a very patient tester of some early versions of LaTeX2`HTML` and many of the current features are a result of her suggestions.

Thanks to (thanks to Ian Foster `<itf@mcs.anl.gov>` and Bob Olson `<olson@mcs.anl.gov>`) at the Argonne National Labs for setting up the LaTeX2`HTML` mailing list[29].

Special credit is due Marcus Hennecke `<marcush@crc.ricoh.com>` for his many extensive revisions, Mark Noworolski `<jmn@eecs.berkeley.edu>` for coordinating V 95.3, Sidik Isani `<isani@cfht.hawaii.edu>`, for his improvement in GIF quality, and to Herb Swan `<dprhws@edp.Arco.com>` for coordinating V 96.1 of LaTeX2`HTML`.

Many others, too many to mention, have contributed bug reports, fixes, and other suggestions. Keep them coming!

References

[1] Leslie Lamport, *LaTeX User's Guide & Reference Manual, 2nd Edition.* Addison-Wesley Publishing Company, Inc., 1994. Online information on TeX and LaTeX is available at http://curia.ucc.ie/info/TeX/menu.html and http://es-sun2.fernuni-hagen.de/info2html?(latex.info)Top.

[2] Michel Goossens, Frank Mittelbach, Alexander Samarin, The LaTeX Companion. Addison-Wesley, 1994 ISBN 0-201-54199-8

---

[29] http://cbl.leeds.ac.uk/nikos/tex2html/doc/mail/mail.html