Starlink Project
Starlink User Note 199.12

P W Draper
A J Chipperfield
M D Lawden

26 January 2009

Star2HTML
Converting Starlink Documents to Hypertext
Version 1.6
User's Manual

Abstract

Star2HTML lets you write (or convert) a Starlink document so that you can create two versions of it from a single source file. A paper version is produced by LaTeX, and a hypertext version (suitable for browsing on the web) is produced by LaTeX2`HTML`. You can tailor each version to its own medium by marking selected text as *LaTeX-only* or *HTML-only*.

Star2HTML also includes a set of document templates for producing Starlink documents in a standard style (such as *Starlink User Note*). They also define new LaTeX commands for adding extra links to the hypertext version of your document (without affecting the paper version).

This document explains these new facilities, and gives advice on good practice and on how to deal with some specific formatting problems when converting a document to hypertext. You are assumed to be familiar with LaTeX.

The advice in this document, in particular that found in Section 5, covers the LaTeX2`HTML`-based document conversion system and not the more modern `tex4ht`-based system.

Contents

1 Introduction

Starlink documents (SUN, SGP, *etc.*) are produced by LaTeX. Star2HTML converts Starlink documents to HTML for use on the World Wide Web. It has the potential to make the entire set of Starlink documents appear as a single hypertext document by linking all the separate documents together. It allows a single document source file (`.tex`) to produce a paper version (when processed by LaTeX) and a hypertext version (when processed by LaTeX2`HTML`).

1.1 Getting started

Copy one of the Starlink template documents into a working directory:

```
# cp $STARLINK_DIR/share/sun.tex .
```

This file is a skeleton LaTeX source file and will give your document a standard style and extra hypertext facilities.

Get a paper copy to look at (or use `xdvi` if you prefer):

```
# latex sun
# dvips sun
# lpr -P1 sun.ps
```

You've produced a Starlink User Note! It's got a title page with an abstract, a contents page, and third page (page 1) containing the body of the document.

You've got a paper version of the document. Now use Star2HTML to produce a hypertext version:

```
# star2html sun
```

A lot of alarming messages, dots, and general gobbledygook appear on your screen. Don't worry about it – this is Unix not Windows. You now have a hypertext version of the document. It's stored in a new directory called `sun.htx` (which will have a lot of files in it, none of which need concern you). Notice that you've produced a paper version and a hypertext version from the same source file (`sun.tex`).

Now use a web browser to look at the hypertext version:

```
# showme ./sun
```

Compare the two versions.

The same information is there, it's just been rearranged on the page. Some extra information has been added to the hypertext version. There's a navigation band at the top (you may have to scroll up to see it), a button to get hardcopy, a couple of entries in the contents list, and some extra identification at the bottom. Also, some elements in the header (like `CCLRC`) appear as hypertext links to external pages that describe what they are.

Star2HTML and the templates do all this automatically. All you have to worry about is filling in the template (and deleting the bits you don't want), replacing those place-holders (like `[author]` and `[title]` with meaningful text, and writing the document. The following sections will help you do this.

1.2 Converting an existing document

A basic conversion of an existing LaTeX document can be done very quickly. It consists of incorporating the appropriate template in the source file, and then changing (or deleting) the place-holders. However, to get the best from the system, some additional tasks are required.

To convert an existing LaTeX document, edit it to:

(1) Incorporate one of the Star2HTML document templates and adapt it to your document.

(2) Add extra hypertext links to your document and allow other Starlink hypertext documents to link to your document.

(3) Adapt the two versions (paper and hypertext) of your document to their respective media.

(4) Correct any formatting problems in the hypertext version that result from the conversion process.

These four tasks are described in later sections of this document.

Start with task 1 (see "Using a template", Section 2), which will produce a basic conversion of your document. You can sort out any linking, formatting, and media problems later.

Run the converted document through LaTeX to make sure you haven't harmed the paper version. Then produce a hypertext version, using `star2html`, and check the result, using `showme`. Use the commands shown in "Getting started" (Section 1.1).

Sometimes the hypertext version doesn't look right on the screen. You've got formatting problems. Check the common problems in Section 5 for possible solutions.

1.3 Some background

Star2HTML comprises two Perl scripts and a set of templates.

The `star2html` command invokes one of the Perl scripts which calls the LaTeX2`HTML` utility to do the actual conversion to hypertext format. LaTeX2`HTML` is described in its own manual[1] (MUD/152). However, you shouldn't need to refer to it unless you have specialised needs. Always use the `star2html` command rather than the `latex2html` command to convert Starlink documents to hypertext.

The templates have two functions:

[1] http://astro.dur.ac.uk/ pdraper/latex2html/manual2002/

- Provide LaTeX definitions for standard styles of Starlink document.
- Add extra LaTeX commands to the basic repertoire provided by LaTeX2HTML. These are used to link Starlink documents together.

The templates are based on LaTeX 2$_\varepsilon$ and all documents should now be produced using this.

The second Perl script, `xlabel`, is a utility to help in labelling sections of the hypertext document so that other documents can link to them. Its use is described in "Labelling sections" (Section 3.4).

2 Using a template

The templates are skeleton LaTeX documents with preambles giving the document a standard style and extra hypertext facilities. They are stored in directory `/star/share` and have filenames which are based on the name of the Starlink document class for which they have been designed (`sun.tex`, `ssn.tex`, *etc.*).

Changing the place-holders in the template will be trivial for LaTeX experts, but it can be intimidating for novices, so the process is described in detail below.

2.1 Changing place-holders

The sections of the template you need to change are contained between comments like:

```
% ? Section description
...
% ? End of section description
```

Some just indicate where you should put any special commands required by your document, others contain sections of code which should be commented in or out (with %) depending upon your requirements and yet others contain place-holders indicated by being enclosed in brackets, `[number]` for example. Change the place-holders as appropriate for your document. Some place-holders may not be relevant to your document, in which case delete the lines that contain them.

The section headings to look for are:

```
% ? Specify used packages
```

Any LaTeX extension packages which are used in your document should be listed
here in a
`usepackage` command like:

The section contains commented out commands for the commonly-used `graphicx` package – don't forget to use the `draft` option of the package to save time when writing your document.

% ? Document identification

Replace or delete the following place-holders:

- `[number].[version]` – replace with the sequence number and version number of the document code (like `199.3`).
- `[author]` – the name of the author (or authors).
- `[date]` – the date of issue (like `3 June 1997`).
- `[title]` – the main title of the document.
- `[software-version]` – the version number of the software being described (like `Version 1.1-2`). If you are not writing about a software package, or there is no version number, delete this line.
- `[manual-type]` – the type of manual you are writing (like `User's Manual`). If this line isn't appropriate, delete it.
- `[Text of abstract]` – every document, except the most trivial, should have an abstract. Put the text of the abstract here. It will appear under the heading "Abstract" on the front page of the paper version. In the hypertext version, it will appear in the first section of the document and will be given the title *Abstract*. If you don't want to supply an abstract, delete this line, and also delete the other sections which are used to create abstracts:

  ```
  % ? Heading for abstract if used
  ...
  % ? End of heading for abstract
  ```

  and

  ```
  % ? New section for abstract if used
  ...
  % ? End of new section for abstract
  ```

  and

  ```
  % ? Document Abstract. (if used)
  ...
  % ? End of document abstract
  ```

  One point to note is that if you want to include a paragraph break in your abstract, you must use the `\par` command rather than just leaving a blank line in the text.

% ? Document specific \providecommand or \newenvironment commands

Use this section to add any extra definitions that you want to use in the body of your text. Most documents don't have anything added to this section, but some (*e.g.* SUN/95) define a large number of new commands here. Just leave this section alone if you don't want to add anything to it.

```
% ? Add picture here if required for the LaTeX version
% ? Add picture here if required for the hypertext version
```

Use these sections to add any pictures that you want to display on the front page of your document. Just leave the sections alone if you don't want to add a picture in this position.

SUN/9 gives details of how to include PostScript files in LaTeX documents and there are examples of simple `\includegraphics` commands in both sections but you will probably need something more elaborate. PostScript files can be included in both versions or you can use `\htmladdimg{<URL>}` to include a `.gif` file for the hypertext document.

```
% ? Latex Copyright Statement
```

This section is required for Starlink documents but may be deleted by others. The statement specified in the `\stardoccopyright` command will appear at the bottom of the second page of the paper version; the HTML version will, by default, have the copyright statement in the footer of each page.

Copyright is assigned to STFC by default.

```
% ? Latex document Table of Contents (if used)
```

Delete this entire section if you don't want a table of contents in the paper version (you will always get one in the hypertext version). We recommend that you always include this section unless your document is short and unstructured.

2.2 Using SST macros

If your document uses the standard macros defined for the Simple Software Tools (SST) package on VMS, then you should add the command:

```
\input{/star/share/sst}
```

to the "Document specific . . . commands" section of the template. The new definitions are a direct replacement for the old macros and should not cause any problems for existing documents unless you have modified the old macros. If you have done so, you must work out how to incorporate your changes into the new definitions.

3 Adding links

Star2HTML automatically adds some links to a document it converts; for example, the section headings in the contents list link to the section text. However, you can add extra links if you wish. These can be to other parts of your document (Internal links), or to other pages on the web (External links). A type of External link that is very common is one to another Starlink document; a special command is provided to create this.

3.1 Internal links

If your document uses the cross-referencing facilities of LaTeX (*i.e.* the `\label{key}`, `\ref{key}` commands), then Star2HTML will automatically create a link from the `\ref` to the `\label`. The link will normally be indicated by a small button, when viewed with a browser (the actual value will be used if the `-aux` switch of the `star2html` command is used, but these values aren't normally useful in hypertext).

If you want to use some descriptive text for the link instead of the button (or if you want to add a new internal link), use the command:

```
\htmlref{text}{label}
```

`text` is the text that will be highlighted as the link (it will also appear in the paper version of your document as ordinary text). `label` is the name of a label defined somewhere in your document by the `\label` command.

For example, the source code for the heading of the current top-level section might be:

```
\section{\label{adding_links}Adding links}
```

You could then create a link to this section by a reference like:

```
... refer to the \htmlref{Adding links}{adding_links} section.
```

This will appear in both versions as:

```
... refer to the Adding links section.
```

but in the hypertext version, "Adding links" is indicated as a link. When activated, it will take the reader to that section.

As another example, the standard SST macros define a label for every routine description. If a reference is made to a routine, say PISAPLOT, in the main text, a link to the full description of the routine can be made by:

```
\htmlref{PISAPLOT}{PISAPLOT}
```

The word "PISAPLOT" will appear in both versions, but in the hypertext version it will be a link to the description of the routine.

3.2 External links

To add a link to an external web page, use the command:

```
\htmladdnormallink{text}{URL}
```

`text` will appear in both versions, but in the hypertext version it will be a link to the web page with the specified `URL`.

To add the URL as a footnote in the paper version, use the following command instead:

```
\htmladdnormallinkfoot{text}{URL}
```

3.3 Links between Starlink documents

The LaTeX command `\label` allows cross-references within normal LaTeX documents. Similarly, the Star2HTML command `\xlabel` allows hypertext links to be set up from other Starlink documents to your document. Its format is:

```
\xlabel{label}
```

Other documents can create a link to it by using `\xref`. The format of `\xref` is:

```
\xref{text}{doc}{label}
```

`text` will appear in both versions, but in the hypertext version it will be a link to the place labelled `label` (with `\xlabel`) in the Starlink document `doc`.

Obviously, if other Starlink documents contain `\xlabel` commands, you can use `\xref` in your document to add links to them.

It will often be necessary to include both a `\label` and an `\xlabel` command at the same point in a document in order to anchor internal and external links to the same place. In this case, the name of the label can be the same in both commands since `\xlabel` modifies it to produce a different anchor name.

You can create a link to any Starlink document by a command like:

```
\xref{SUN/x}{sunx}{}
```

This link will take you to the beginning of the document as the templates contain an `\xlabel` with a blank label on the title page of the document. Not all Starlink documents are available in hypertext form but the link will even work in that case – browsers will usually just display the paper version.

There is one further step required to create a link to another local Starlink document – you must run the hypertext linker `hlink`, after running `star2html` on your source file, so that the links between the two documents can be resolved. If you do not do this then all the cross references will be passed to the Starlink central document server for resolution. Thus, if your document contains links to other Starlink documents, you can convert, link it locally, and view it by:

```
# star2html sunx
# hlink
# showme ./sunx
```

The Starlink hypertext linker, `hlink`, is more fully described in SUN/188.

3.4 Labelling sections

If you want to label a complete section, for other documents to refer to, you should include an `\xlabel` as part of the section title. This is because the link will be to the *exact point* in the document where the `\xlabel` occurs, rather than to the start of the section (as with normal `\ref` and `\label`).

So that others can easily link to sections of your document, we recommend that you `\xlabel` all sections and subsections with the section title in lower case, non-alphanumeric characters removed and words separated by underscore.

For example, the subsection you are now reading has been labelled as follows:

```
\subsection{\xlabel{labelling_sections}Labelling sections}
```

and other documents can create a link to it by using `\xref` as in:

```
See SUN/199 for more on
\xref{labelling sections}{sun199}{labelling_sections}.
```

You can use the `xlabel` command from the Unix shell to ensure that your document has the recommended `\xlabel`'s for each section and subsection. The command:

```
# xlabel document
```

Will produce a new version of the document, in file `new`*document*, containing any additional required `\xlabel` commands. It will also warn you if there are any duplicated section labels or if an existing `\xlabel` differs from the recommendation. See the description of `xlabel` (in Appendix B) for more information on this.

When you are happy with the result, copy `new`*document* back to *document*.

3.5 Links to routine descriptions

The SST routine description macros include an `\xlabel` command for each routine so, if you use them, your routines will already be labelled to allow links from other documents. By the same token, you can include links to routine descriptions in other documents which have used the SST macros.

For example:

```
\xref{CHR\_LEN}{sun40}{CHR_LEN} finds the used length of a string.
```

4 Adapting to the media

You may wish the hypertext version of your document to differ from the paper version. To restrict text to the *paper* version, use the environment:

```
\begin{latexonly}
This text will only appear in the LaTeX version.
\end{latexonly}
```

To restrict text to the *hypertext* version, use the environment:

```
\begin{htmlonly}
This text will only appear in the hypertext version.
\end{htmlonly}
```

Note:

- The `\begin{htmlonly}`, `\end{htmlonly}`, `\begin{latexonly}` and `\end{latexonly}` statements should be on separate lines from other text or LaTeX commands.
- Both text and commands may be included within the environments, but problems can arise if they contain any of the LaTeX sectioning commands (see "Selecting sections", Section 4.1).

Single lines of HTML-only and LaTeX-only text can be controlled using the commands:

```
\html{text for the hypertext version}
```

and

```
\latex{text for the paper version}
```

Or you could use the conditional statement:

```
text for the \latexhtml{paper}{hypertext} version.
```

4.1 Selecting sections

If you want to have a section that only appears in the paper version of your document, then your document preamble must define new commands to replace the LaTeX sectioning commands, for example:

```
% Latex only sections, subsections etc.
% Use these in a latexonly environment.
\providecommand{\latexonlysection}[1]{\section{#1}}
\providecommand{\latexonlysubsection}[1]{\subsection{#1}}
\providecommand{\latexonlysubsubsection}[1]{\subsubsection{#1}}
```

You must then use these new commands instead of the normal sectioning commands, for example:

```
\begin{latexonly}
\latexonlysection{LaTeX readers beware...}
LaTeX readers are missing out by not reading the HTML version of this document.
\end{latexonly}
```

If a sectioning command appears in an `htmlonly` environment, you get the warning message "`Cannot find \end{htmlonly}`", but the result should be satisfactory.

5 Common conversion problems

We have already mentioned that Star2HTML uses LaTeX2HTML to do the conversion to hypertext. LaTeX2HTML usually makes a good attempt at this conversion. However, it can produce disappointing results, particularly for complex sections, such as the cover page and subroutine/application descriptions in Starlink documents.

Some LaTeX structures cannot be translated into HTML. Normally, LaTeX2HTML handles these by either ignoring them, or by passing them back to LaTeX for processing and then converting the result to in-line images (complex maths symbols, for example). The result occasionally requires tweaking by hand. Sometimes the result is so poor that you need to re-write parts of your document in order to generate a more acceptable result.

5.1 Things to avoid

5.1.1 Raw TeX

LaTeX2HTML is designed to deal with LaTeX commands of the form `\command{arg1}{arg2}...` and cannot, in general, deal satisfactorily with raw TeX commands. If this problem occurs, you'll notice "=" signs in inappropriate places, or missing parts in your hypertext version. The only thing to do is to use LaTeX equivalents of the TeX commands.

For example use:

```
\newfont{\ssttt}{cmtt10 scaled 1095}
```

instead of:

```
\font\ssttt=cmtt10 scaled 1095
```

or

```
\providecommand{\_}{\texttt\symbol{95}}
```

instead of:

```
\providecommand{\_}{\tt\char'137}
```

and so on.

5.1.2 Parboxes

These are always translated into GIFs and often have an incorrect size. Try to avoid them if possible. If you can't, you might have to fix things by hand.

5.2 Things that rarely look right

5.2.1 Tabbing environments

Don't use the `tabbing` environment – the converter makes a mess of it and the only solution is to edit the HTML files. Use the `tabular` environment instead.

5.3 Irritants

5.3.1 Unwanted blank lines

Consider the source text:

```
Some text
\latex{for latex readers.}
\html{HTML readers.}
```

You'd expect the generated text to look like:

```
Some text for latex readers.
```

in the paper version, and

```
Some text for HTML readers.
```

in the hypertext version. However, the hypertext version comes out like:

```
Some text

for HTML readers.
```

because the `\latex` line is replaced by a blank line and consequently ends up as a paragraph break. To get around this you must include some text on the `\latex` line, as in:

```
Some text \latex{for latex readers.} \html{for HTML readers.}
```

Or you could get the same effect using the `\latexhtml` command:

```
Some text \latexhtml{for latex readers.}{for HTML readers.}
```

5.3.2 Tildas in URLs

To get a tilda into a URL, use:

```
\htmladdnormallink{My home page at RAL}{http://star.rl.ac.uk/\~{}me}
```

instead of:

```
\htmladdnormallink{My home page at RAL}{http://star.rl.ac.uk/~me}
```

which just throws away the tilda.

5.4 Using the `htmladdimg` command to replace figures

The `\htmladdimg` command is used to display an image stored in an external file. Its form is:

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

where the `URL` is the hypertext address of the image (probably just the name of an `image.gif` file in the default directory).

It is tempting to use this command to replace existing figures with better colour graphics. However, you cannot use this command inside a `figure` environment. Instead, you should use something like:

```
\begin{latexonly}
  \begin{figure}
    \label{This figure}

          body of figure

    \caption{Latex eps version}
  \end{figure}
\end{latexonly}
```

However, you will not get a figure number in the caption.

5.5 Hints from the Starlink Software Librarian

The following list of recommendations was compiled by the Starlink Software Librarian, Martin Bly, who has issued many Starlink documents submitted for release and has noted the things that have caused him trouble:

- Don't reference specific version numbers of documents, *e.g.* use "SSN/26" rather than "SSN/26.1".
- Use a / to separate document-type from document-number, *e.g.* use "SSN/26" rather than "SSN26" or "SSN 26".
- In the paper version of your document, the first page (page 1) of the main body of your text should appear on the right-hand-side when it is printed double-sided (or sent for photocopying), otherwise your document looks amateurish. This main body is preceded by the title page, abstract, and contents list (which have page numbers written in Roman numerals).

  You can ensure the correct layout by including a `\cleardoublepage` immediately before your "page 1". The latest templates include this automatically but you should check existing documents for this problem.
- Use `\begin{quote}` ...`\end{quote}` to surround `\begin{verbatim}` ...`\end{verbatim}` texts. If the area of text is large or long, use `\small` inside the quote environment.

6 Finally

If you've spent many hours fine-tuning the layout of your LaTeX document, you will probably be disappointed with the Star2HTML conversion – most of your extra bells and whistles will be thrown away without trace. Remember, it's not yet possible to obtain the level of formatting control in HTML that LaTeX offers (although this is situation is changing as HTML develops).

More problems which may occur with LaTeX2HTML are described in LaTeX2HTML manual[2] (see MUD/152). If you run across undocumented problems, please pass them on (hopefully with fixes) to the Starlink support list. A support page[3] for this package is maintained and will list any new problems.

7 References

The following documents give further information of value to authors of Starlink documents:

Charles, A.C.	SC/9	LaTeX cookbook.
Lawden, M.D.	SGP/28	How to write good documents for Starlink.
Lawden, M.D.	SGP/50	Starlink document styles.
Charles, Anne	SUN/9	LaTeX – A document preparation system.
Warren-Smith, R.F.	SUN/188	HTX – Hypertext cross-reference utilities.
Bly, M.J.	SUN/201	LaTeX2HTML – LaTeX to HTML conversion.
Drakos, Nikos	MUD/152	A LaTeX to HTML translator.
Lamport, Leslie		LaTeX User's Guide & Reference Manual, 2nd ed, Addison-Wesley, 1994.

[2] http://astro.dur.ac.uk/ pdraper/latex2html/manual2002/
[3] http://astro.dur.ac.uk/~pdraper/star2html/star2html.html

A The star2html command

The star2html command sets up a Starlink-specific environment before calling LaTeX2HTML to produce a hypertext version of a Starlink document. A typical invocation, is simply:

```
# star2html sun199
```

However, there are some optional switches which you might find useful. The full format of the command is:

```
# star2html [-a author(s)] [-aux] [-c copyright_statement]
            [-d document_date] [-m email_address] [-s 0|1]
            [-obsolete] [-t title] [arguments for LaTeX2HTML] document
```

The switches (all of which are optional) have the following meanings:

-a authors – The names of the document authors. If not supplied, the value of the command \stardocauthors will be used. If this doesn't exist, the name of the person running the application will be used. The value appears in the footer of every page of the hypertext version.

-aux – This controls whether or not the document .aux file is used. (This file is produced by running LaTeX on the source file, and contains a description of the section numbering). If this switch is omitted, the .aux file is not used. This default is sensible because the structure of the paper version may differ from the structure of the hypertext version. For example, optional diagrams and/or sections may be specified by \htmlonly and \latexonly environments.

-c copyright_statement – A copyright statement to override the default which will appear in the footer of each page. The default statement is found from the \stardoccopyright command if there is one in the document, or failing that from the environment variable HTX_COPYRIGHT if defined, or failing that the default is: "Copyright © YYYY Council for the Central Laboratory of the Research Councils" (where YYYY is the current year). To prevent any copyright statement from appearing, specify "-c none" or, in the absence of a \stardoccopyright command or explicit -c "copyright_statement", set HTX_COPYRIGHT to "none".

-d document_date – The date when the document was created. If not supplied, the value of the \stardocdate command will be used. If this doesn't exist, today's date will be used. The value appears in the footer of every page of the hypertext version.

-hcserver URL – A URL pointing to the hardcopy server to be used for the document. Should be an copy of the hcserver script from the HTX package. If this is not set a check of the variable HTX_HCSERVER is made, if that is not set then no changes to the hardcopy server URL are made.

`-m e-mail_address` – A contact e-mail address for the document. If not supplied, the value of the environment variable `HTX_EMAIL` will be used. If this isn't set, then the e-mail address "`starlink@jiscmail.ac.uk`" will be used. The value appears in the footer of every page of the hypertext version. If you are only responsible for producing your own documents, you may find it convenient to set this environment variable in your `.login` script.

`-obsolete` – This flag indicates you want to mark the document as obsolete. This is done by displaying an image with the word OBSOLETE written into it as the background.

`-s 1|0` – This switch controls whether the input file will be split into parts, or not, before being processed by LaTeX2HTML. The split occurs at `\section`, `\subsection`, `\sstroutine` and `\routine` statements. Splitting up the document may considerable speed up processing and consequently this is the default.

`-t title` – The title of the document. If not supplied, the value of the `\stardoctitle` command will be used. If this fails, a name generated from the document type will be used. The value is used as a title for the first page, and appears in the footer of every page of the hypertext version.

`LATEX2HTML arguments` – Any unknown arguments before the file name will be passed to LaTeX2HTML (do not use the `-dir` and `-no_subdir` arguments).

`document` – The name of the document to be converted into HTML (the file type `.tex` is optional).

Some error messages always appear, but they can usually be ignored.

B The xlabel command

The xlabel command is used to ensure that all sections and subsections in the document include an \xlabel command labelling them with the section heading in lower case with all non-alphanumeric characters removed and words separated by underscore.

The format of the command is:

```
# xlabel document
```

This will process the file *document* and create a new version, new*document*, containing any additional \xlabel commands required. The file type .tex is optional and will be added if it is omitted.

A label is created from the heading of each \section and \subsection in the specified document by removing common font-changing commands (such as \texttt, \texttt, \em and \emph), replacing all spaces by underscore, removing all non-alphanumeric characters (underscore counts as alphanumeric here) and folding to lower case. If the sectioning command does not already contain an \xlabel command specifying the generated label, one is inserted.

The labels generated will be displayed and should be inspected for peculiarities, probably caused by unusual LaTeX commands in the heading.

Warnings will be displayed if:

- The document contains an \xlabel command specifying the same label for a different section. Note that duplicates for \xlabel's not in section or subsection headings will not be detected.
- The section already contains an \xlabel command specifying a different label.
- The section heading resulted in an empty label.

When you are happy with the result, copy new*document* back to *document*.

C The structure of a converted hypertext document

Star2HTML bases the structure of the hypertext version of a LaTeX document on the original structure defined by commands like `\section`. However, it changes the format of the title page, adds buttons and links to aid navigation within the document, and adds identification information to each page. It does this automatically.

It splits a document up into pages. Each section, subsection, and so on, is stored in its own page. Each page is divided into bands, separated by a horizontal line. The top-level page is the *Title Page*. This is the first page you see when you start reading the document.

The Title Page is divided into the following bands:

(1) A *Navigation Band*. This contains a row of navigation buttons (some of which are blanked out), followed by a line containing a link to the document Abstract (by default).

(2) A *Title Band*. This contains information from the title page of the paper document (title, document number, author, date, *etc.*), but presented in a different order and style. References to CCLRC/RAL, PPARC, and Starlink are converted into links to their associate web pages. A button is added to help people copy the document.

(3) A *Contents Band*. This looks like two bands because a horizontal line appears between the heading “Contents” and the list of section titles (which are links to the section pages).

(4) A *Navigation Band*. This is a duplicate of the band at the top of the page. It only appears in long pages.

(5) An *Identification Band*. Shows the document title, code, author, date, and e-mail address.

The other pages are made up of similar bands, some of which don’t always appear.

(1) A *Navigation Band*. The lines below the navigation buttons show the titles of the Next, Up, and Previous pages. “Up” means the next level up in the structural hierarchy, thus if the page is a subsection, it will point to the containing section.

(2) A *Text Band*. This contains the author’s text for the section.

(3) A *Structure Band*. This shows the titles of the subsections in this section, with links to the pages containing the text. If the section has no subsections, this band won’t appear.

(4) A *Navigation Band*. This is a duplicate of the band at the top of the page. It only appears in long pages.

(5) An *Identification Band*. Same as for the Title Page.

If your document has an index it will be converted to an index page with links from each entry to the indexed entry. The last page in the hypertext document will be called “About this document . . . ”. This gives details about how and when the hypertext document was produced. It has no equivalent in the paper version.

D Changes in release 1.6

In this release Star2HTML has been upgraded to use LaTeX2HTML version 2000.2 exclusively (this is also part of the Starlink release). The old compatibility version of LaTeX2HTML has been removed as has version 98.2. Tests have shown that the new system is largely compatible with 98.2, so documents updated to work with that should continue to work, **however**, all documents that used the compatibility version of LaTeX2HTML *will now need to be updated*, so you must follow one of the routes described in the next subsection.

D.1 Incompatibilities with previous documents

Unfortunately changes in LaTeX2HTML version 98.2 and 2002.2 mean that several major incompatibilities with previous versions of Star2HTML documents continue.

To correct old documents you and take full advantage of the new facilities you'll need to modify your documents so that they use the current templates provided. You may also need to slightly modify any media sensitive commands that you have added.

The simplest way to update your document is to change it to use the appropriate new template (*e.g.* `sun.tex`) and to use the new version of `sst.tex`, if appropriate. Another change you must make is to modify any `\latexonly{}` commands to use the equivalent `\latex{}` command. Note that `\latexonly` as an environment (`\begin{latexonly} \end{latexonly}`) can continue to be used, this is just a change to the command form.

Note that if you are using `sst.tex` then it *must* now be included in the document prologue, not in the text. Now if you have not made any other changes to differentiate your on-line and paper documents, then you should be able to convert it.

If you have added customized `\providecommands` to your document then you will also need to change any commands that look like:

```
\providecommand{\mycommand}{this version is only for LaTeX}
\begin{htmlonly}
    \renewcommand{\mycommand}{this version is only for LaTeX2HTML}
\end{htmlonly}
```

to:

```
\providecommand{\mycommand}{this version is only for LaTeX}
\begin{htmlonly}
    \providecommand{\mycommand}{this version is only for LaTeX2HTML}
\end{htmlonly}
```

This is especially important if you have any section control commands in the redefinitions (`\providecommand` and `\renewcommand` are now dealt with at different stages in the conversion process). In general if your `\renewcommand` is just a replacement command then you should now use `\providecommand`.

If your document is very customized and you do not want to change the whole prologue, or the effort seems too great, then try the following sequence of changes:

- Move the %\end{rawtex} line to the end of the "debugging" section (all commands visible in your document now override any that LaTeX2HTML may know about in any fashion, i.e. builtins, from style files etc., so must be hidden). Now rename %\end{rawtex} to %end{latexonly} and %\begin{rawtex} to %begin{latexonly}.
- Move your sst.tex definitions into the prologue.
- Change any \renewcommands in htmlonly environments to \providecommands (this includes the SST definitions).
- Change any \latexonly{} commands to \latex{} (leave your latexonly environments as they are).
- Modify the font declaration in the SST definitions to:

  ```
  \latex{\newfont{\ssttt}{cmtt10 scaled 1095}}
  \html{\providecommand{\ssttt}{\tt}}
  ```

  Removing any other such definitions and adding a command:

  ```
  \providecommand{\html}[1]{}
  ```

  To the standard part of the prologue.
- An additional space issue means you also need to change:

  ```
  \stardoccategory \stardocnumber \\
  ```

  to:

  ```
  \stardoccategory\ \stardocnumber \\
  ```

- And finally add the following definition to the standard part of the prologue.

  ```
  \providecommand{\latexhtml}[2]{#1}
  ```

If all this doesn't work then you'll have to do the conversion properly.

E Changes in release 1.5

Star2HTML was changed internally to work with the autotools based build system.

F Changes in release 1.4

The document templates have been updated to correct an error that centered the HTML document title, type and version and also to remove the Rutherford Appleton Laboratory URL, which is now invalid. The string "CCLRC / Rutherford Appleton Laboratory" is now a single hypertext link to CCLRC.

The templates also now include a copyright statement for the paper document and the star2html script will add a copyright statement to the footer of each page unless it is specifically prevented. (The action of the -c switch and HTX_COPYRIGHT environment variable has changed.) By default the copyright statement will assign copyright to STFC.

This document has been updated to describe the new scheme.

G Changes in release 1.3

In this release Star2HTML has been upgraded to include LaTeX2HTML versions 98.2 and 97.1. Since the time of the last Star2HTML release LaTeX2HTML itself has been under constant development (by a keen group of users organised using a discussion group) and there have been many changes which produce a generally better translation. The key features of these changes are:

- Much improved image generation. Images are now anti-aliased, resulting in much clearer text, and are generated much quicker.
- Images of the same text are now shared. This reduces the number of images required.
- LaTeX tables are now translated into HTML ones.
- HTML 3.2 is now generated (this allows improved formatting like: centering, left-right justification, table captions etc.).
- LaTeX 2_ε is now supported.
- Many more LaTeX 2_ε classes and packages are now supported (this means less images and more intelligent translations). *
- Large speed and memory usage improvements. These mean less waiting around for a translation to complete.
- Maths formulae are more carefully translated (*e.g.* plain text is used whenever possible, HTML superscript and subscript are also used).
- `\providecommands` can now be more complex (*i.e.* can include other newcommands). This means existing documents can be translated with less effort.
- The HTML produced is generally much cleaner (redundant `<P>`'s have mostly gone).
- HTML 4 support is available. *
- ISO-Latin and unicode characters are now translated. *

Changes in Star2HTML, other than to support LaTeX2HTML, are the addition of the "`-s`" and "`-c`" flags and a new command `latex2html`.

The "`-s`" flag (which is on by default) greatly speeds up document conversion by splitting it up into smaller parts. The "`-c`" flag (which has no effect by default) adds a copyright statement to the document footer (this can be set permanently by the `HTX_COPYRIGHT` environment variable).