Bibtex2html User Documentation Version 1.02

Grégoire Malandain

May 19, 2004

Foreword: this documentation aims to describe the `bibtex2html` tool. As it is written by the developper, and not by users, it may not be convenient for the latter. Please feel free to comment and react on it: by doing so, you could help us to improve it.

General information

`bibtex2html` is a BibTeX to HTML translator. It is written in C.

Documentation

- http://www-sop.inria.fr/epidaure/personnel/malandain/codes/bibtex2html.html, bibtex2html on-line manual.
- ftp://ftp-sop.inria.fr/epidaure/Softs/bibtex2html/bibtex2html.ps.gz, gzipped postcript.
- ftp://ftp-sop.inria.fr/epidaure/Softs/bibtex2html/bibtex2html.pdf, pdf.

Distribution

Current version is 1.02, it can be downloaded from

- ftp://ftp-sop.inria.fr/epidaure/Softs/bibtex2html/.

The distribution contains:

- Full source,
- Some icons,
- User manual, in Postcript, PDF and HTML

Acknowledgment

This little software has been built in the same spirit than bib2html of Eric Marchand which kindly provides me with its sources (in C++). As I was not smart enough to modify it to obtain some desirable features (as links between pages), I have prefered to develop a new code (in C).

Contents

1 Presentation

The purpose of this software is to automatically produce html pages from BibTeX files, and to provide access to the BibTeX entries by several criteria: year of publication, category of publication, keywords (see section 3 for the definition of these categories), author name, and keywords, from an index page.

Moreover cross-linking is generating between pages to provide an easy navigation through the pages without going back to the index.

Each entry is formatted as the bibliographic style `unsrt.bst` did, except that the title would be in bold. Thus, processing the BibTeX files with the following

```
\documentclass{article}
\bibliographystyle{unsrt}
\begin{document}
\nocite{*}
\bibliography{file1.bib,file2.bib,file3.bib,...}
\end{document}
```

should give an idea of what the output will look like. Moreover, it may provide with error messages which corresponds to syntax error (see sections B.1 and B.2 as a first help).

2 Getting started

This package was originally designed to translate BibTeX files (generally with a `.bib` extension) into a bundle of html files allowing a user-friendly navigation in the bibliography.

2.1 Installing the sources

2.1.1 On unix/linux

I assume you get a package from the above mentionned ftp site. It is named `bibtex2html-X.XX.extension` where `X.XX` denotes a version number. The extension can be either `.zip` or `.tar.gz`. Thus running

```
% unzip -d   bibtex2html-X.XX.zip
```

or

```
% tar -xvzf bibtex2html-X.XX.tar.gz
```

should create a directory named `bibtex2html-X.XX` containing the sources and some other files.

2.1.2 On windows

I assume you get a package from the above mentionned ftp site. It is named `bibtex2html-X.XX.extension` where `X.XX` denotes a version number. The extension can be either `.zip` or `.tar.gz`. Uncompress the archive in some directory, say `C:\greg\bibtex2html\`

2.2 Compiling

2.2.1 On unix/linux

Go into the directory where the sources are

```
% cd bibtex2html-X.XX
```

and compile with `make`

```
% make
```

Notice that there exists a preprocessor macro `ICONSDIR` that is used to define the default path for the icon files. One may want to change it to make it correspond to one's configuration. To do that, you will have to edit the `Makefile`.

2.2.2 On Windows

The following described the compilation with Microsoft Visual C++ 6.0 (under Windows 2000).

1. **Create new project/workspace**
   - **File → New** opens a windows. Choose the **Projects** tab, and select **Win32 Console Application**.
   - As **Location**, choose for instance the directory where the sources are (e.g. `C:\greg\bibtex2html\`).
   - Enter a **Project Name**, e.g. `bibtex2thml_win32`: it changes the **Location** into `C:\greg\bibtex2thml_win32`. You can change again the location (e.g. into `C:\greg\bibtex2thml_win32`).
   - Choose to create a new Workspace.
   - Click on **OK**, a new window pops up.
   - Choose **An empty workspace** and click on **Finish**.
   - Confirm your choice by clicking **OK** in the last windows.
2. **Add the sources to the project**
   - **Project → Add to Project → Files...**

- Select the source files (the .c and the .h) from the directory where they are (`C:\greg\bibtex2html` in this example), and click **OK**.

3. **Tune the settings**

   - **Project → Settings** opens a windows.
   - Choose the settings you want to change (at the top left, default is `Win32Debug`).
   - In tab **Link**, choose the executable name, e.g. `bibtex2html.exe`
   - In tab **C/C++**
     - Make sure that the define **WIN32** is already indicated in the `Preprocessor definitions` (else you have to add it).
     - In the `Project Options`, you have to indicate the path to the include files (the .h files)

       `/I"C:\greg\bibtex2html"`

       as well as the macro `ICONSDIR` that is used to define the default path for the icon files

       `/D"ICONSDIR=\"C:\\greg\\bibtex2html\\icons\""`

4. **Build the executable**

   Build → Build bibtex2html.exe

   The executable `bibtex2html.exe` will be created in the directory `C:\greg\bibtex2thml_win32`.

2.3 Generating html pages from BibTeX files

Note for Windows users `bibtex2html` can only be used from a MS-DOS windows. Thus, open such a windows, you will be allowed to type on-line commands. By specifying the full path towards `bibtex2html.exe`, i.e.

```
C:\greg\bibtex2thml_win32\bibtex2html.exe
```

you will be able to use it.

What is described below is also valid for Windows OS, by replacing the command `bibtex2html` by `bibtex2html.exe`.

According that you have installed (copied) `bibtex2html` in some place listed in your `${PATH}` variable (for unix/linux users), or that you are using the full path towards `bibtex2html` (or`bibtex2html.exe`), you can process your `.bib` files. While parsing its arguments, `bibtex2html` considers that all arguments that do not match a command line option (see section 7.1) are input files. The following command will then create a bundle of html files from the BibTeX files given in the command line.

```
% bibtex2html file1.bib file2.bib ... -force
```

You may access to each generated html file (see below for details) through the `index.html` as starting point. Notice the `-force` option, that will create the needed subdirectories (see section 3) if they do not exist.

If you also want to navigate from file to file, you may allow *links* between files with the `-html-links` option.

```
% bibtex2html file1.bib file2.bib ... -force -html-links
```

You may also want to use icons as links towards distant files or pages. The simpliest way to do that is

```
% bibtex2html file1.bib file2.bib ... -force -copy-icons
```

The option `-copy-icons` will copy the icon files from the distribution (according that the preprocessor macro `ICONSDIR` has been set properly, see the section 2.2 about compilation) to the local subdirectory `./Icons/`, that will be created if necessary thanks to the `-force` option.

If the icons already exist locally (in the subdirectory `./Icons/`), specifying the `-icons` option is then sufficient, i.e.

```
% bibtex2html file1.bib file2.bib ... -force -icons
```

2.3.1 Moving the html pages

According you like the generated output, you may want to move it at some more convenient place on your web server storage space. Thus, if you do not use icons

```
% cp -rp index.html Author Biblio Category Keyword Year DEST/
```

will copy all the output file in the `DEST` directory (for unix/linux users) If you use icons, you just have to add the `Icons` into the above command.

3 Description of the generated html pages

By default, html pages are generated in

- the current directory, namely './', it only concerns the index file,
- and subdirectories: `./Author/`, `./Biblio/`, `./Category/`, `./Keyword/`, `./Year/`;
- an other subdirectory `./Icons/` may be needed if icons are desired to indicate links towards urls or documents.

Each directory contains a number of html files, that correspond to a certain sort of the bibliography.

3.1 Bibliography by authors

`bibtex2html` will generate one html file per indexed author (or editor if relevant) in the subdirectory `./Author/`. In these files, entries are sorted by categories.

By default, all the authors (and some editors, see section 4.1) are indexed but it is possible to change this behavior by specifying either the authors to be indexed (and all the other will not be indexed): see the sections 4.1.

Each generated file will be named after the author's name, namely the last-name followed by a '-' and the initials of the firstnames, e.g. `BACH-JS.html` for the above example. In case of unknown author, an additional file named `ZZZZZZZ-UNKNOWN.html` is generated.

3.1.1 Comparison of names

In large bibliographies, the same author name can be written in different ways. E.g. you may find either Jean-Sebastien Bach, J.S. Bach, J.-S. Bach, or J. Bach.

Two names are considered to be the same if the last names are exactly equal, and if the initials of the first names are also the same. From above, it comes that Jean-Sebastien Bach, J.S. Bach, and J.-S. Bach will be considered as designing the same person. But J. Bach will be someone else.

It is possible to allow a more sloppy comparison, by comparing only the first initials (in case of the numbers of initials differs), and thus to consider J. Bach as the same person than J.S. Bach (see section 7.1).

3.2 Bibliography by keywords

`bibtex2html` will generate one html file per indexed keyword in the directory `./Keyword/`. In these files, entries are sorted by categories.

By default, all the recognized keywords are indexed, but it is possible to change this behavior by specifying either the keywords to be indexed (and all the other will not be indexed), or the ones not to be indexed (and all the other will be indexed): see the section 4.2 for details.

Each generated file will be named after the keyword.

3.3 Bibliography by categories

Categories have been defined as a group of several BibTeX entry types, they are by order :

Books and proceedings	@BOOK, @PROCEEDINGS
Thesis	@PHDTHESIS
Articles in journal or book's chapters	@INBOOK, @INCOLLECTION @ARTICLE
Conference's articles	@INPROCEEDINGS
Internal reports	@TECHREPORT
Manuals, booklets	@BOOKLET, @MANUAL
Miscellaneous	@MASTERSTHESIS, @AUDIOVISUAL @FILM, @MISC, @UNPUBLISHED

One html file is generated per category in the `./Category/` directory. In these files, entries are sorted by year of publication.

3.4 Bibliography by years

For each year found in the BibTeX files, two files are produced in the `./Year/` subdirectory: The first one is named `year.html` and `year.complete.html` where *year* is a four digit number corresponding to the year (or `UNKNOWN` if the year is not specified).

In addition to the formatted entry, `year.complete.html` contains the following fields (if given): annote, comments, abstract, and the original BibTeX entry. They can be reached by links from other pages.

In each file/page, the entries of the year are sorted with respect to the above described categories.

3.5 Complete bibliographies

In addition to the above generated files, two files that contain the whole bibliography are also created. Both of them are produced in the `./Biblio/` subdirectory. Their name are built from a generic name (see section 5.2). The first one, `complete-bibliography.html`, is the complete bibliographie in html format, while the second one `complete-bibliography.bib`, is a BibTeX file. They both are sorted with respect to the field `year`: this can be changed by the command line option `-sort-criterium` (see section 7.1).

3.6 Index file

An index file, `index.html`, is created in the current directory. It contains links towards all the other generated and is thus a good starting to browse the bundle of files.

4 Finer Control of the output

Most of the control can be done by some appropriate `@string`, that should be into a file (the choice of the syntax allows you to put it into your BibTeX file) that have to be passed to the command line. It will also be used later (see section 5.1).

4.1 Selecting authors

By default, all the authors (or editors if relevant) are indexed, and then one file will be generated per different name in the subdirectory `./Author/`. Relevant editors are the ones of @book, @proceedings, or @inbook if there are no authors.

However, one may want only to index some of them, and then all the others are conidered as being non-indexed. It can be done by specifying the authors to be indexed in lines like

```
@string{ author_to_be_indexed = "G. Malandain" }
```

or

```
@string{ authors_to_be_indexed = "G. Malandain and J.-S. Bach" }
```

The second part of strings is interpreted as authors in BibTeX entries and follows then BibTeX's syntax. These lines may be repeated as many times as necessary, ie:

```
@string{ authors_to_be_indexed = "G. Malandain and J.-S. Bach" }
@string{ authors_to_be_indexed = "Mozart, W. A. and J.-B. Lully" }
```

Similarly, one may want not to index some of the authors, and then all the others are considered as being indexed. It can be done by specifying the authors not to be indexed in lines like

```
@string{ author_not_to_be_indexed = "G. Malandain" }
```

or

```
@string{ authors_not_to_be_indexed = "G. Malandain and J.-S. Bach" }
```

Both keys, `authors_to_be_indexed` and `authors_not_to_be_indexed`, are exclusive. If both are present, only the authors specified by the `authors_to_be_indexed` key will be effectively indexed.

4.2 Selecting keywords

By default, all the keywords are indexed. However, one may want only to index some of them, and then all the others are conidered as being non-indexed. It can be done by specifying the keywords to be indexed in lines like

```
@string{ keyword_to_be_indexed = "Medical image processing" }
```

or

```
@string{ keywords_to_be_indexed = "matching, registration" }
```

Similarly, one may want not to index some of the keywords, and then all the others are considered as being indexed. It can be done by specifying the keywords not to be indexed in lines like

```
@string{ keyword_not_to_be_indexed = "bull shit" }
```

or

```
@string{ keywords_not_to_be_indexed = "bull shit, not relevant" }
```

Please notice that, even if not indexed, these keywords will appear in the generated files: using such clear-cut keywords (as above) is perhaps not a good idea.

Both keys, `keywords_to_be_indexed` and `keywords_not_to_be_indexed`, are exclusive. If both are present, only the authors specified by the `keywords_to_be_indexed` key will be effectively indexed.

4.3 Generating one single file

Instead of generating a number of files, it may be desirable to produce only one file:

- the input BibTeX files contain only a few items
- it is the bibliography of a single project, or author
- the input BibTeX file(s) need to be merged and sorted (to output BibTeX file, use `-style bibtex`) so that it will be more easily checked and corrected
- ...

this is can done with the command line option `-single-output`. This single file follows the same rules than the complete bibliography already described in section 3.5, except that it is created in the current directory, `./`, instead of in the `./Biblio/` directory. It can thus be sorted according to the same different criteria

4.4 Changing the title of the pages

You surely want to change the title of the generated pages, especially the one of the index file. This is possible by setting the values of different *keys* within `@string` (see section 5.1).

There are two different kinds of title: the one that appears on the top of the navigator window (between <TITLE> and </TITLE> in the html syntax) and the one that will appear at the top of the display page (and that takes place after the <BODY> declaration in the html syntax). The name of the *key* describing the first kind of title will have the `******_file_title`, while the one of the second kind of title will have the `******_page_title` (see section 6.2).

Changing the titles of the index page (see section 3.6) can be achieved by changing the values of the `index_file_title` and `index_page_title` *keys*. E.g., the defaults are set by

```
@string{ index_file_title = "List of publications: index" }
@string{ index_page_title = "List of publications: index" }
```

Similarly,

- for the bibliographies by authors in the `./Author/` subdirectory (see section 3.1), titles are given by

  ```
  @string{ author_file_title = "Publications of %s %s" }
  @string{ author_default_file_title = "Publications of unknown author(s)" }
  @string{ author_page_title = "Publications of %s %s" }
  @string{ author_default_page_title = "Publications of unknown author(s)" }
  ```

  the two conversion specifications '%s' being for the firstname and the lastname, of the author, the *default* titles being used when there are no authors

- for the bibliographies by keywords in the `./Keyword/` subdirectory (see section 3.2), titles are given by

  ```
  @string{ keyword_file_title = "Publications about '%s'" }
  @string{ keyword_page_title = "Publications about '%s'" }
  ```

  the conversion specification '%s' is for the keyword itself

- for the bibliographies by categories in the `./Category/` subdirectory (see section 3.3), titles are given by

  ```
  @string{ category_file_title = "%s" }
  @string{ category_page_title = "%s" }
  ```

  the conversion specification '%s' is for the name of the category, that can not be changed

- for the bibliographies by years in the `./Years/` subdirectory (see section 3.4), titles are given by

  ```
  @string{ reduced_year_file_title = "Publications of year %d" }
  @string{ reduced_year_default_file_title = "Publications with no year" }
  @string{ reduced_year_page_title = "Publications of year %d" }
  @string{ reduced_year_default_page_title = "Publications with no year" }
  ```

  or

  ```
  @string{ complete_year_file_title = "Publications of year %d" }
  @string{ complete_year_default_file_title = "Publications with no year" }
  @string{ complete_year_page_title = "Publications of year %d" }
  @string{ complete_year_default_page_title = "Publications with no year" }
  ```

  the conversion specification '%d' being for the year, the *default* titles being used when no year is given

- for the complete bibliography (only the html one) in the `./Year/` subdirectory (see section 3.5), titles are given by

  ```
  @string{ complete_biblio_file_title = "All publications sorted by %s" }
  @string{ complete_biblio_default_file_title = "All publications" }
  @string{ complete_biblio_page_title = "All publications sorted by %s" }
  @string{ complete_biblio_default_page_title = "All publications" }
  ```

  the conversion specification '%s' being for the name of the sort criterium, that can not be changed, the *default* titles being used when no sorting criterium is given

- for the single output (see section 4.3) titles are given by

  ```
  @string{ single_output_file_title = "All publications sorted by %s" }
  @string{ single_output_default_file_title = "All publications" }
  @string{ single_output_page_title = "All publications sorted by %s" }
  @string{ single_output_default_page_title = "All publications" }
  ```

  with the same remark as above.

4.5 Changing the background of pages

The description of the body of html pages can either be done in style sheet (see sections 6.7.2 and 7.4) or in the body declaration. The body starting element can be changed through the `body_tag.start` key. E.g.

```
@string{ body_tag.start = "<body bgcolor=\"yellow\">\n" }
```

will display a yellow background (notice the `\"` in the syntax, please refer to section 5.1.3 for more details). Here there are some attributes of the body element:

bgcolor Specifies the background color for the document body. See below for the syntax of color values.

text Specifies the color used to stroke the document's text. This is generally used when you have changed the background color with the BGCOLOR or BACKGROUND attributes.

link Specifies the color used to stroke the text for unvisited hypertext links.

vlink Specifies the color used to stroke the text for visited hypertext links.

alink Specifies the highlight color used to stroke the text for hypertext links at the moment the user clicks on the link.

background Specifies a URL for an image that will be used to tile the document background.

4.6 Changing the styles of (sub)titles

The page title of each page is enclosed between the `page_title_tag.start` key and the `page_title_tag.end` key, while the subtitles if pages are enclosed between the `page_subtitle_tag.start` key and the `page_subtitle_tag.end` key. Thus, changing these *keys* will allow you to change the style of the (sub)titles. E.g.,

```
@string{ page_title_tag.start = "\n<h1 align=\"center\">\n" }
@string{ page_title_tag.end = "\n</h1>\n" }
@string{ page_subtitle_tag.start = "\n<h2 align=\"center\">\n" }
@string{ page_subtitle_tag.end = "\n</h2>\n" }
```

will result in a very simple display (notice the `\"` in the syntax, please refer to section 5.1.3 for more details).

The options `-print-default-context`, `-print-context`, and `-print-full-context` (see section 7.1), will allow you to see the default values of these *keys*.

4.7 Generating or not part of the bundle

It is also possible to generate only a sub-part of the bundle of files, by setting the values of different *keys* within `@string` (see section 5.1).

The '1' value means the creation/generation of the files, while the '0' value means no creation.

- the bibliographies per author, described in section 3.1 and generated in the `./Author/` subdirectory, will not created if the `author_create_pages` *key* is set to 0, i.e.

  ```
  @string{ author_create_pages = 0 }
  ```

- the bibliographies per keyword, described in section 3.2 and generated in the `./Keyword/` subdirectory, will not created if the `keyword_create_pages` *key* is set to 0, i.e.

  ```
  @string{ keyword_create_pages = 0 }
  ```

- the bibliographies per category, described in section 3.3 and generated in the `./Category/` subdirectory, will not created if the `category_create_pages` *key* is set to 0, i.e.

  ```
  @string{ category_create_pages = 0 }
  ```

- part of the bibliographies per year, the ones of name `year.html`, described in section 3.4 and generated in the `./Year/` subdirectory, will not created if the `reduced_year_create_pages` *key* is set to 0, i.e.

  ```
  @string{ reduced_year_create_pages = 0 }
  ```

- part of the bibliographies per year, the ones of name `year.complete.html`, described in section 3.4 and generated in the `./Year/` subdirectory, will not created if the `complete_year_create_pages` *key* is set to 0, i.e.

  ```
  @string{ complete_year_create_pages = 0 }
  ```

- the complete bibliographies, described in section 3.5 and generated in the `./Biblio/` subdirectory, will not created if the `complete_biblio_create_pages` *key* is set to 0, i.e.

  ```
  @string{ complete_biblio_create_pages = 0 }
  ```

- the index file, described in section 3.6 and generated in the current directory `./` will not created if the `index_create_pages` *key* is set to 0, i.e.

  ```
  @string{ index_create_pages = 0 }
  ```

Warning: with the current version, this is not compatible with the use of links between the pages.

5 Customizing the output

The default behavior of `bibtex2html` should correspond to some consensual expectation of such a tool. However, one may want to adapt it to more specific need. The aim of this section is to explain how things can be changed.

5.1 Principle

There are a large number of variables that allows to parameterize the behavior of `bibtex2html`. I choose to put those indications in some @STRING{ ... } so that they can be put into bibtex files (even if they may yields warnings). The file containing those indications, if independant from the bibtex files to be processed, has simply to be added to the list of files to be processed. Thus, the syntax is always

```
@string{ key = the_value_I_want_to_set_for_key }
```

There are a lot of *keys* that allows to control the output, all of them can be printed out by running `bibtex2html` with the `-print-keys` options, i.e.

```
% bibtex2html -print-keys
```

Some (most?) of them are of no interest at all. Then, one surely want to see only the used ones. This can be achieved at different time of the processing.

```
% bibtex2html -print-default-context
```

print out the values of non-null customization *keys* after the *style* have been loaded, but before any input file has been read.

```
% bibtex2html -print-context
```

print out the values of non-null customization *keys* after the *style* have been loaded, and after the input files have been read. Thus, the user can check that his own customization has been taken into account.

5.1.1 Admissible values

The expected values for the *keys* as in

```
@string{ key = the_value_I_want_to_set_for_key}
```

are either integer values, e.g.

```
@string{ author_index.cells_per_row = 3 }
```

or strings, e.g.

```
@string{ index_page_title = "Hello, world" }
```

In some cases, it could be useful to assign empty strings, i.e. `""`, to some keys.

5.1.2 Files as admissible values

Sometimes, especially in case of complete change of the look of the pages, some of these keys may take a very long string as value (i.e. several lines in the created files). To handle this case, it is also possible to give filenames as key values. Thus, specifying

```
@string{ body_tag.start = biblio-body-start.html }
```

will not write "biblio-body-start.html" in the output file(s), but will copy the contents of the file `biblio-body-start.html` in the output file(s). This could be the easiest way to assign a certain number of input lines to some key's value (see section 6.7.2).

5.1.3 Strings as admissible values

A string is supposed to be delimited by double quotes, i.e. `"..."`, as in

```
@string{ index_page_title = "Hello, world" }
```

However, the double quote sign is also used in the HTML syntax as in

```
<h1 align="center">
```

To differentiate between the two cases, the double quote as a string delimitor or as a HTML syntax sign, the later has to be written `'\"'` as in the C/C++ syntax, yielding something like

```
@string{ page_title_tag.start = "\n<h1 align=\"center\">\n" }
```

5.2 Files and directories names

The names of the subdirectories as well as the names of some files (the main index, and the generic name of the *complete bibliographies*) can be changed (see section 3). Default are given by:

```
@string{ directory_authors = "Author" }
@string{ directory_biblio = "Biblio" }
@string{ directory_categories = "Category" }
@string{ directory_keywords = "Keyword" }
@string{ directory_years = "Year" }
@string{ directory_icons = "Icons" }

@string{ filename_complete_biblio = "complete-bibliography" }
@string{ filename_index = "index" }
@string{ filename_extension = "html" }
```

5.3 Links and Icons

5.3.1 Links

One of the desirable features of `bibtex2html` is to offer an easy and intuitive navigation inside the bundle of generated html pages. This is controlled by the `use_html_links_between_pages` key. Thus,

```
@string{ use_html_links_between_pages = 1 }
```

will make the links to be generated between pages, while

```
@string{ use_html_links_between_pages = 0 }
```

will disable this feature. The `use_html_links_towards_urls` key controls the creation of links towards distant files or urls, i.e. the ones referenced by the fields `postcript`, `pdf`, and `url` in the BibTeX entries.

See also the `-html-links` command line option.

5.3.2 Icons, links towards distant pages

From an aesthetic point of view, it may be desirable to see the links towards distant files or urls as icons (small images). This can be achieved globally either by the `use_icons` key, i.e.

```
@string{ use_icons = 1 }
```

or by the `-icons` command line option.

A finer control can also be achieved for each of the possible link type, i.e. `postscript`, `url`, and `pdf`. E.g.

```
@string{ bib_postscript.icon = "my_own_ps.gif" }
```

specifies the name if the icon file for the links towards postscript files (assuming that this icon file is in the `./Icons/` subdirectory), while

```
@string{ bib_postscript.use_icon = 1 }
```

turns on the use of icons for the links towards postscript files only (0 turn it off). The same can be achieved for `url` and `pdf`, by replacing `postscript` by either `url` or `pdf` in the two above keys.

6 Controlling the layout

Because of the large number of *keys* controlling the output, it will be a tremendous task to describe all of them. Thus, *layout* files can be produced to see where the different *keys* are used. You should try

```
% bibtex2html some_file.bib -style layout -force
```

in some safe directory. You will notice that it will create

- the following subdirectories `directory_authors`, `directory_biblio`, `directory_categories`, `directory_years`, and `directory_icons`, that are named after their *key* names, and not after these *keys*' values. They contains files that are named automatically (see section 3) but with the `.layout` extension.
- the file `filename_index.layout` in the current directory
- the file `filename_complete_biblio.layout` in the `directory_biblio` subdirectory

Similarly, a single output (see section 4.3) can also be generated with the *layout* style.

```
% bibtex2html some_file.bib -style layout -force -output single.layout
```

In these `.layout` files, the value of all the *keys* needing a string (see section 5.1.1) will be the name of *key* itself between less than and greater than signs. They provide then an easy way to really see where each *key* is written. Of course, it will depends on the values of the integer *keys* that also controls the output, and that won't be changed with respect to your or to the default tuning. Such `.layout` files were used to produced most of the following descriptions (where some extra spacing has been added for clarity).

Remember that there exists one command line option, `-print-keys`, that list all the existing *keys*, and several other options, `-print-default-context`, `-print-context`, and `-print-full-context`, that list the keys' values (see section 7.1).

6.1 Files generic layout

The generic layout of the produced files is described by the following

```
<file_tag.start>

   <head_tag.start>
      <file_title_tag.start>
         <******_file_title>
      <file_title_tag.end>
   <head_tag.end>

   <body_tag.start>
      <header_of_body>
      <******_header_of_body>
      <page_title_tag.start>
         <******_page_title>
      <page_title_tag.end>
      <header_of_contents>
      <******_header_of_contents>

      ...
      this inner part will be described later
      ...

      <******_footer_of_contents>
      <footer_of_contents>
      <separator>
      <disclaimer.content_field.start>
      <disclaimer.prefix>
        <disclaimer.content.start>
          <disclaimer_1>
        <disclaimer.content.end>
        <disclaimer.content.start>
          <disclaimer_2>
        <disclaimer.content.end>
      <disclaimer.content_field.end>
      <separator>
      <date_tag.start>
         automatically produced date
      <date_tag.end>
      <author_tag.start>
         <author_name>
      <author_tag.end>
      <separator>
      <credits>
      <******_footer_of_body>
```

```
      <footer_of_body>
   <body_tag.end>
<file_tag.end>
```

E.g., in the default implemented *style*, the values of `file_tag.start` and `file_tag.end` are

```
@string{ file_tag.start = "<html>\n\n" }
@string{ file_tag.end = "\n</html>\n" }
```

6.2 Files specific layout

Most of the keys presented above are common for all the generated files. They are denoted as *generic*. However, some of them, the ones containing `******` above and that will qualified as *specific*, depend of the type of generated files. These different types are:

- the bibliographies by authors in the `./Author/` subdirectory (see section 3.1), and the name of the *keys* are built by replacing `******` by `author`.
- the bibliographies by keywords in the `./Keyword/` subdirectory (see section 3.2), and the name of the *keys* are built by replacing `******` by `keyword`.
- the bibliographies by categories in the `./Category/` subdirectory (see section 3.3), and the name of the *keys* are built by replacing `******` by `category`.
- the bibliographies by years in the `./Years/` subdirectory (see section 3.4), and the name of the *keys* are built by replacing `******` by either `reduced_year` or `complete_year`.
- the complete bibliography (only the html one) in the `./Year/` subdirectory (see section 3.5), and the name of the *keys* are built by replacing `******` by `complete_biblio`.
- the index in the current directory `./` (see section 3.6), and the name of the *keys* are built by replacing `******` by `index`.
- the single output (see section 4.3) and the name of the *keys* are built by replacing `******` by `single_output`.

Thus the beginning of a bibliography by author, i.e. a file in the `./Author/` subdirectory is described by the following

```
<file_tag.start>

   <head_tag.start>
      <file_title_tag.start>
```

```
      <author_file_title>
   <file_title_tag.end>
<head_tag.end>

<body_tag.start>
   <header_of_body>
   <author_header_of_body>
   <page_title_tag.start>
      <author_page_title>
   <page_title_tag.end>
   <header_of_contents>
   <author_header_of_contents>
   ...
```

while the beginning of a bibliography by keyword, i.e. a file in the `./Keyword/` subdirectory, is described as follows

```
<file_tag.start>

   <head_tag.start>
      <file_title_tag.start>
         <keyword_file_title>
      <file_title_tag.end>
   <head_tag.end>

   <body_tag.start>
      <header_of_body>
      <keyword_header_of_body>
      <page_title_tag.start>
         <keyword_page_title>
      <page_title_tag.end>
      <header_of_contents>
      <keyword_header_of_contents>
      ...
```

6.3 More details about the end of files

At the end of the files, will be written, in order (see section 6.2)

- `******_footer_of_contents`
- `footer_of_contents`
- `separator`, if there is something else, among the disclaimers, the date, the author, and the credits to be written afterwards. It can be something like

  ```
  @string{ separator = "\n<br><hr size=\"2\" width=\"100%\"><br>\n\n" }
  ```

Notice the \" in the syntax, please refer to section 5.1.3 for more details.

- *Disclaimer*. Disclaimer(s) is(are) written if the value of the `******_write_disclaimer` key is not 0. The layout of disclaimers is as follows

```
<disclaimer.content_field.start>
   <disclaimer.prefix>

   <disclaimer.content.start>
      <disclaimer_1>
   <disclaimer.content.end>

   <disclaimer.content.start>
      <disclaimer_2>
   <disclaimer.content.end>

<disclaimer.content_field.end>
```

  There are two strings of disclaimer: by default, the first one is in english and the second one is in french. If you want to disable the second one, insert

```
@string{ disclaimer_2 = "" }
```

  in your customization (or BibTeX) file.

- `separator`, if applicable, between *disclaimers* and *date* or *author*
- *Date*. If the value of the `******_write_date` is not 0. The value of the date is automatically determined.
- *Author of the pages*. If the value of the `******_write_author` is not 0. An author's name can be given with the `author_name` key, if not it is automatically determined.
- `separator`, if applicable, between *date* or *author* and *credits*
- *Credits*. If the value of the `******_write_credits` is not 0. The value of the credits string is given by the `credits` key.
- `******_footer_of_body`
- `footer_of_body`

6.4 The inner content of files

The inner content of files (excepted the index file) is made of several subparts similar to the following one

```
<page_subtitle_tag.start>
  ... some predefined subtitle ...
<page_subtitle_tag.end>
<list_tag.start>
  <item_tag.start>
    ... bibtex item #1 ...
  <item_tag.end>
  <item_tag.start>
    ... bibtex item #2 ...
  <item_tag.end>
  ...
<list_tag.end>
```

There is a subtitle, generally predefined strings (e.g. categories) or years, followed by a list containing items.

6.5 The inner content of the index file

This content depends on the generated specific files, where the specific part of the keys will be denoted by `******` (see section 6.2).

Obviously, the `index_create_pages` and `******_create_pages` *key* should not be 0 (see section 4.7),

According that the `******_build_index` is also not 0, the index of the `******` pages will be generated in the index page as a table:

```
<page_subtitle_tag.start>
   predefined sub title
<page_subtitle_tag.end>
<******_index.table.start>
  <******_index.row.start>
     <******_index.cell.start>
        generated page #1
     <******_index.cell.end>
     ...
     <******_index.empty_cell.start>
     <******_index.empty_cell.end>
  <******_index.row.end>
<******_index.table.end>
```

there are two cells elements, one, `******_index.cell`, for the filled cells, and the other, `******_index.empty_cell`, for the cells needed to finish rows. If the `use_html_links_towards_urls` key is not 0, each cell content is a html link towards the corresponding page.

The number of cells in a row can be controlled by the `******_index.cells_per_row` key.

6.5.1 Authors and keywords

The above behavior is a little different for the `author` and `keyword` specific files. First of all, rows will contains only names or keywords with the same name's first letter.

An extra first column can be added to the index table to contain these first letters: this is controlled by the `******_put_initials_in_index` key. As above, there are two cell elements to describe the layout of the cells of this extra first column: `******_index.extra_cell` and `******_index.empty_extra_cell`.

Moreover, before the index table, an additional table containing only these first letters can be generated, that allows then a direct access to the names beginning with this letter. This is controlled by the `******_build_initials_index` key. This table is described by

```
<******_initials.table.start>
  <******_initials.row.start>
     <******_initials.cell.start>
        A
     <******_initials.cell.end>
     ...
     <******_initials.empty_cell.start>
     <******_initials.empty_cell.end>
  <******_initials.row.end>
<******_initials.table.end>
```

the number of cells in a row being controlled by the `******_initials.cells_per_row` key.

The display of the names (for the `author`) in the index table can also be parametrized with the `******_arg1_in_cell` element for the first name, and the `******_arg2_in_cell` element for the last name.

The display of the keywords in the index table can be parametrized with the `******_arg1_in_cell` element.

6.6 BibTeX items layout

The implemented layout follows the one of `unsrt.bst`, thus the final display (in browsers) of the generated files should look like the output of the LaTeX file of section 1. Description of the BibTeX items together with the associated fields is provided in section B.1.

In the following, we will first present how the standard fields of BibTeX items are displayed, how links towards distant pages are displayed, how keywords (if any) are displayed, and how the remaining (additional) fields are displayed.

6.6.1 Standard fields

`booktitle` enclosed between the `bib_booktitle.start` and the `bib_booktitle.end` keys.

`journal` enclosed between the `bib_journal.start` and the `bib_booktitle.end` keys.

`title` enclosed between the `bib_title.start` and the `bib_title.end` keys.

names of authors or editors are enclosed between the `bib_firstname.start` and the `bib_firstname.end` for the firstname, and between the `bib_lastname.start` and the `bib_lastname.end` for the lastname.

`series` enclosed between the `bib_series.start` and the `bib_series.end` keys (only when both the `volume` and `series` fields are filled).

6.6.2 Links towards distant pages

There are 3 fields that are supposed to be links towards distant fields: `pdf`, `postscript`, and `url`. They all will be displayed in the same way. The below explanation is done with `url`, but it can be replaced either by `pdf` or `postscript`.

1. If links toward distant pages have to be created, that is controlled by the `use_html_links_towards_urls` key (see section 5.3.1)

   (a) If icons are to used, i.e. the values of both `use_icons` and `bib_url.use_icon` keys are not 0 (see section 5.3.2), the link, i.e. the value of the `URL` field, will be displayed as an icon

   ```
   <a href="link">
   <img align=middle src="../<directory_icons>/<bib_url.icon>">
   </a>\n
   ```

   (b) else, if a string is provided in the `bib_url.link` key to represent the link, then it is display as

   ```
   <bib_url.link_field.start>
     <a href="link"><bib_url.link></a>\n
   <bib_url.link_field.end>
   ```

   please note that the whole link is enclosed between the `bib_url.link_field.start` and `bib_url.link_field.end` keys

   (c) else the link itself is written to specify the link, i.e.

   ```
   <bib_url.link_field.start>
     <a href="link">
       <bib_url.content_field.start>
         link
       <bib_url.content_field.end>
     </a>\n
   <bib_url.link_field.end>
   ```

   please note that the value of the `URL` field (the link string) is additionally enclosed between the `bib_url.content_field.start` and `bib_url.content_field.end` keys

2. else, if no links toward distant pages have to be created the value of the `URL` field is displayed by

```
<bib_url.link_field.start>
  <bib_url.prefix>
  <bib_url.content_field.start>
    link
  <bib_url.content_field.end>
<bib_url.link_field.end>
```

6.6.3 Keywords

Keywords are displayed as follows

```
<bib_keywords.prefix>
<bib_keywords.content_field.start>
  keyword #1
<bib_keywords.content_field.end><bib_comma>
...
<bib_keywords.content_field.start>
  last keyword
<bib_keywords.content_field.end><bib_dot>
```

Moreover, if links to be generated between pages have to be created, that is controlled by the `use_html_links_between_pages` key (see section 5.3.1), the indexed keywords (see section 4.2) are links towards their own pages.

6.6.4 Additional fields

Additional fields are the `abstract`, `annote`, and `comments` BibTeX fields, and the BibTeX item key, entry, and file

They all will be displayed in the same way. The below explanation is done with `abstract`, but it can be replaced either by `annote`, `comments`, `key`, `entry`, or `file`.

First, it should be remarked that the display of these additional fields will depends on the type of generated files, i.e. a *specific* key, `******_write_bibtex_abstract`, (see section 6.2) controls the output. If its values is

0: nothing is written

1: if the value of the `use_html_links_between_pages` key is not 0, the contents of the field is a link towards the corresponding BibTeX item in the adequate `year.complete.html` file (see section 3.4). This link is written as follows:

```
<bib_abstract.link_field.start>
  <a href="../<directory_years>/year.complete.<filename_extension>#bibtex_key">
    <bib_abstract.link>
  </a>
<bib_abstract.link_field.end>
```

2: the contents of the field is written in the file as follows

```
<bib_abstract.content_field.start>
  <bib_abstract.prefix>
  <bib_abstract.content.start>
    abstract
  <bib_abstract.content.end>
<bib_abstract.content_field.end>
```

6.6.5 Additional keys

These keys allows to control the look of the generated files

dot is controlled by the `bib_dot`

comma is controlled by the `bib_comma`

and is controlled by the `bib_and`. Notice that it should be encapsulated by spaces, *ie*

```
@string{ bib_and = " and " }
```

6.7 Building your own layout

6.7.1 Adding your own input

There are four places (see the generic layout in section 6.1) where you can put your own input:

- after the beginning of the body and before the page title, with the `header_of_body` and the `******_header_of_body` keys;
- just after the page title, with the `header_of_contents` and the `******_header_of_contents` keys;
- at the end of the content (and before some ad-hoc stuff: disclaimers, date, etc) with the `footer_of_contents` and the `******_footer_of_contents` keys;
- at the very end of the page, with the `footer_of_body` and the `******_footer_of_body` keys;

For each of these four places, there is a common part for all the generated files, and a specific part, prefixed by `******`, that depends of the files' type and will be written in all the files of the same type. As above, `******` can be either `author`, or `keyword`, or `category`, or `reduced_year`, or `complete_year`, or `complete_biblio`, or `index`, or `single_output`.

6.7.2 Style sheets and similar stuff

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to web documents. It is done by attaching styles to pages with the following statement

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

inserted after `</title>` and before `</head>`. At the same place, one may write `<meta>` elements, or insert `<script>` elements, etc.

Thus, one may have a large number of lines, e.g.

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
<meta name="GENERATOR" content="bibtex2html 2.00">
<meta name="description" content="A bibliography designed for my own use">
<meta name="keywords" lang="en" content="bibtex2html, bibliography">
....
```

to be inserted between `</title>` and `</head>`.

One solution for the insertion is to add these line to the `head_tag.end`, without forgetting the `</head>`. The drawback of this solution is that it makes you write a long string for the key's value.

One better solution is to write all these line in one single file and to assign the filename to the key's value. In our example it comes to build a file, say `biblio-head-end.html`, containing

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
<meta name="GENERATOR" content="bibtex2html 2.00">
<meta name="description" content="A bibliography designed for my own use">
<meta name="keywords" lang="en" content="bibtex2html">
....
</head>
```

and to have

```
@string{ head_tag.end = biblio-head-end.html }
```

in your customization (or BibTeX) file.

6.8 Default specific layout

There exists many *specific* files, and then a lot of *specific* customization keys. However, one may want to change all the same keys in all the specific files. This can be achieved by using the `default` key as `******`. This only works for the NULL strings.

7 Practical information

7.1 Command line option

Running `bibtex2html` without any arguments should give the full list of options. Running it with `--help` give additional information about each option.

`--help` display some help

`-bibtex` from now on, input files are considered to be BibTeXfiles (default behavior).

`-check-icons` will check the icon files in the local subdirectory `./Icons/`. Implies also the option [-icons.

`-complete-equality-of-names` | `-ceon` names are supposed to be the same if both last names are equal, if the initials of the first names are the same, and there are the same number of initials. Eg, J. Bach and J.S. Bach do not represent the same name, but J.S. Bach and J.-S. Bach do.

`-copy-icons` will copy the icon files from the distribution installation (according that the preprocessor macro `ICONSDIR` has been set properly) to a local subdirectory `./Icons/`. Implies also the option [-icons.

`-create-directories` will create the necessary subdirectories (i.e. `./Author/`, `./Biblio/`, `./Category/`, `./Keyword/`, `./Year/`, and `./Icons/` if needed) if they do not exist.

`-force` the same as `-create-directories`

`-help` the same as `--help`

`-html-links` use html between the generated pages and towards distant files or urls.

`-icons` use icons to indicate links toward distant files or urls.

`-medline` from now on, input files are considered to be MedLine files. (default behavior).

`-no-html-links` do not use html between the generated pages and towards distant files or urls.

`-no-icons` do not use icons to indicate links toward distant files or urls.

`-output` the same as `-single-output`

`-partial-equality-of-names` | `-peon` names are supposed to be the same if both last names are equal and if the initials of the first firstnames are the same. Eg, J. Bach and J.S. Bach represent the same name.

`-print-default-context` print the context, i.e. the values of the non-null customization *keys* after the *style* have been loaded.

`-print-context` print the context, i.e. the values of the non-null customization *keys* after the *style* have been loaded, and after the input files (if any) have been read.

`-print-full-context` the same as `-print-context` but print also the null customization *keys*.

`-print-keys` prints all the existing customization *keys*

`-single-output %s` will generate only one file, containing the whole bibliography. Its name is the argument coming after `-single-output`. It will be written on *stdout* if the name is "-". See section 4.3.

`-sort %s` the same as `-sort-criterium`

`-sort-criterium %s` specify the sort criterium, that will be used to generated single output/file containing the whole bibliography (all the inputs)

- `-sort-criterium cat` will be sorted with respect to to the categories defined in section 3.3.
- `-sort-criterium year` will be sorted with respect to the year of publication.
- `-sort-criterium name` will be sorted with respect to the authors' names (in alphabetical order).

`-style %s` allow to specify an output style

- `-style bibtex` will generated BibTeX files.
- `-style html` will generated html files.
- `-style layout` will generated layout files: they are generated with the name of the *key* between less than and greater than signs as *key value* for all the *key* needing a string (see section 6).

`-use-html-links` the same as `-html-links`

`-use-icons` the same as `-icons`

7.2 Examples of output

- I use the computational geometry bibliography, `geom.bib`, for testing. The version I got contains 13824 BibTeX items, there are 8016 different names among 27487 authors and 1851 different keywords among 8807 keywords. Due to the size of the generated bundle of files, I can not keep it as an output example. The BibTeX file can retrieved by anonymous ftp from ftp://ftp.cs.usask.ca/pub/geometry/ or use some web search interface

  - http://www.cs.ruu.nl/usr/cgi-bin/bib-search/geom
  - http://www-ma2.upc.es/~geomc/geombib/geombibe.html
  - http://cgm.cs.mcgill.ca/geombib.html
  - http://netlib.bell-labs.com/netlib/search.html

- http://www-sop.inria.fr/epidaure/BIBLIO/ presents the bibliography of our research team.

- http://public.rz.fh-wolfenbuettel.de/~hoeppnef/bib/index.html, maintained by Frank Höppner, contains references, abstracts and BibTex entries of papers about fuzzy clustering, pattern recognition, machine learning, data mining, intelligent data analysis, knowledge discovery in databases, etc.

7.3 Other *TEX to HTML translators

Most of them have been found by searching on the web.

- BIBTEX to html translators
  - http://membres.lycos.fr/sgalland/english/tools/bib2html/, set of script LaTeX and Perl, by Stéphane Galland.
  - http://www.irisa.fr/vista/bib2html/bib2html.html, C++, by Eric Marchand.
  - http://www.cs.cornell.edu/home/wkiri/bib2html/, flex and C, by Kiri Wagstaff.
  - http://www.cs.dartmouth.edu/~dfk/bib2html.html, by David Kotz.
  - http://www.research.att.com/~mff/strudel/Bib2Html/README.html
  - http://www.lri.fr/~filliatr/bibtex2html/, Objective Caml, by Jean-Christophe Filliatre.

- LATEX to html translators
  - http://pauillac.inria.fr/hevea/, Objective Caml, by Luc Maranget.
  - http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html, LaTeX2HTML

- TEX to html translators
  - http://hutchinson.belmont.ma.us/tth/, TTH.

7.4 Other useful URLs

- http://www.w3.org/ the world wide web consortium (short, W3C)
- http://www.w3schools.com/ tutorials on HTML, XHTML, CSS, etc.

8 FAQ

Here *frequently* means at least once.

8.1 How to print the bibtex key/file associated to each entry?

This can be achieved by tuning some configuration *keys*, and was sketched in section 6.6.4.

E.g., to print out the bibtex keys for the files of the `./Author/` directory, one has to add the following

```
@string{ author_write_bibtex_key        = 2 }
```

in your customization (or BibTeX) file. For other files, i.e. the ones of the `./Biblio/`, `./Category/`, `./Keyword/`, and `./Year/` directories, the customization keys are `******_write_bibtex_key` with `******` being respectively `complete_biblio`, `category`, `keyword`, `reduced_year`, and `complete_year`.

The same stands for the `******_write_bibtex_file` customization keys that allow to write out the file where the bibtex item can be found.

8.2 I just want my publications

This can be achieved by tuning some configuration *keys*. Put what follows in some file (say `greg.cfg`).

```
%
% do not generate all these files
% what remains is only the "Authors" pages
%
@string{ keyword_create_pages = 0 }
@string{ category_create_pages = 0 }
@string{ reduced_year_create_pages = 0 }
@string{ complete_year_create_pages = 0 }
@string{ complete_biblio_create_pages = 0 }
@string{ index_create_pages = 0 }

%
% I just want one author (me of course)
% and I want to write the file ('MALANDAIN-G.html')
% in the current directory
%
@string{ directory_authors = "." }
@string{ author_to_be_indexed = "G Malandain"}

%
% this is more tricky: the default header of the file body and
```

```
% the default footer of the inner content
% contain an explicit link to the index file.
% We have to override them with an NON-EMPTY string.
% ‘bibtex -print-context‘ may help to identify such a problem.
%
@string{ default_header_of_body = " " }
@string{ default_footer_of_contents = " " }
```

and run `bibtex2html` with all your BibTeX files plus this configuration file (you still can customize the output of course: in this case the above file should be the last one in the line command), *i.e.*

```
% bibtex2html file1.bib file2.bib ... greg.cfg
```

it will just generate one html file with the publications of the desired author.

A List of changes

may 19, 2004

- More documentation about `@string`, please read section 5.1.3.
- Enable compilation on Windows OS (suggested by Ramon Miralles Ricos).
- Add `ps`: does exactly the same than the `postscript` (changes suggested by S. Saint-Jean).

may 6, 2004

- Correct a bug in reading medline files: lines like "`AU  - van Beethoven L`" were improperly read.
- correct some minor problems

july 28, 2003

- better management of strings, allows '#' to concatenate strings
- allows use of `crossref`
- better management of links (in case of partial writing)
- correct minor problems

april 30, 2002

- Addition of the key `bib_and` (see section 6.6.5).
- Keywords are now also searched into the `title` and `abstract` fields, so that entries exhibiting the keyword in these fields (but not in the `keyword` field) are indexed.
- Accept bibliographic files in MedLine format such the ones that can be saved after a search in PubMed.

november 22, 2002 Additional field `file` added to BibTeX items (that allows to known from in which file an item can be found). Nicer reformat of Bibtex output.

june 17, 2002 New version, allowing full customization.

november 27, 2001 Add icons for links.

september 24, 2001 Add keywords as a sorting criterium..

B The entries

B.1 Entry types

Standard entries [1, 2]

@ARTICLE An article from a journal or magazine. Required fields: `author`, `title`, `journal`, `year`. Optional fields: `volume`, `number`, `pages`, `month`, `note`.

@BOOK A book with an explicit publisher. Required fields: `author` or `editor`, `title`, `publisher`, `year`. Optional fields: `volume` or `number`, `series`, `address`, `edition`, `month`, `note`.

@BOOKLET A work that is printed and bound, but without a named publisher or sponsoring institution. Required field: `title`. Optional fields: `author`, `howpublished`, `address`, `month`, `year`, `note`.

@CONFERENCE The same as @INPROCEEDINGS, included for Scribe compatibility.

@INBOOK A part of a book, which may be a chapter (or section or whatever) and/or a range of pages. Required fields: `author` or `editor`, `title`, `chapter` and/or `pages`, `publisher`, `year`. Optional fields: `volume` or `number`, `series`, `type`, `address`, `edition`, `month`, `note`.

@INCOLLECTION A part of a book having its own title. Required fields: `author`, `title`, `booktitle`, `publisher`, `year`. Optional fields: `editor`, `volume` or `number`, `series`, `type`, `chapter`, `pages`, `address`, `edition`, `month`, `note`.

@INPROCEEDINGS An article in a conference proceedings. Required fields: `author`, `title`, `booktitle`, `year`. Optional fields: `editor`, `volume` or `number`, `series`, `pages`, `address`, `month`, `organization`, `publisher`, `note`.

@MANUAL Technical documentation. Required field: `title`. Optional fields: `author`, `organization`, `address`, `edition`, `month`, `year`, `note`.

@MASTERSTHESIS A Master's thesis. Required fields: `author`, `title`, `school`, `year`. Optional fields: `type`, `address`, `month`, `note`.

The default `type` is *Master's thesis*.

@MISC Use this `type` when nothing else fits. Required fields: none. Optional fields: `author`, `title`, `howpublished`, `month`, `year`, `note`.

@PHDTHESIS A PhD thesis. Required fields: `author`, `title`, school, `year`. Optional fields: `type`, `address`, `month`, `note`.

It is the same as @MASTERSTHESIS, but with the default `type` set to *PhD thesis*.

@PROCEEDINGS The proceedings of a conference. Required fields: `title`, `year`. Optional fields: `editor`, `volume` or `number`, `series`, `address`, `month`, `organization`, `publisher`, `note`.

@TECHREPORT A report published by a school or other institution, usually numbered within a series. Required fields: `author`, `title`, `institution`, `year`. Optional fields: `type`, `number`, `address`, `month`, `note`.

The default `type` is *Technical report*.

@UNPUBLISHED A document having an author and title, but not formally published. Required fields: `author`, `title`, `note`. Optional fields: `month`, `year`.

In addition to these entries, I add the following

@AUDIOVISUAL Some audiovisual material. The same as @TECHREPORT, except that the default `type` is *Video*.

@FILM Some audiovisual material. The same as @AUDIOVISUAL, included for internal compatibility.

B.2 Fields

`address` Usually the address of the publisher or other type of institution. For major publishing houses, van Leunen [3] recommends omitting the information entirely. For small publishers, on the other hand, you can help the readger by giving the complete address.

`annote` An annotation. It is not used by the standard bibliography styles, but may be used by others that produce an annotated bibliography.

`author` The name(s) of the author(s), in the format described in the LaTeX book.

`booktitle` Title of a book, part of which is being cited. See the LaTeX book for how to type titles. For book entries, use the title field instead.

`chapter` A chapter (or section or whatever) number.

`crossref` The database key of the entry being cross referenced. NOT IMPLEMENTED YET.

`edition` The edition of a book (for example, *Second*). This should be an ordinal, and should have the first letter capitalized, as shown here; the standard styles convert to lower case when necessary.

`editor` Name(s) of editor(s), typed as indicated in the LaTeX book.If there is also an author field, then the editor field gives the editor of the book or collection in which the reference appears.

`howpublished` How something strange has been published. The first word should be capitalized.

`institution` The sponsoring institution of a technical report.

`journal` A journal name. Abbreviations are provided for many journals; see the Local Guide.

`key` Used for alphabetizing, cross referencing, and creating a label when the `author` information is missing. This field should not be confused with the key that appears in the `\cite` command and at the beginning of the database entry. NOT IMPLEMENTED YET.

`month` The month in which the work was published or, for an unpublished work, in which it was written. You should use the standard three-letter abbreviation, as described in Appendix B.1.3 of the LaTeX book.

`note` Any additional information that can help the reader. The first word should be capitalized.

`number` The number of a journal, magazine, technical report, or of a work in a series. An issue of a journal or magazine is usually identified by its volume and number; the organization that issues a technical report usually gives it a number; and sometimes books are given numbers in a named series.

`organization` The organization that sponsors a conference or that publishes a manual.

`pages` One or more page numbers or range of numbers, such as 42–111 or 7,41,73–97 or 43+ (the '+' in this last example indicates pages following that don't form a simple range). To make it easier to maintain Scribe-compatible databases, the standard styles convert a single dash (as in 7-33) to the double dash used in TeX to denote number ranges (as in 7–33).

`publisher` The publisher's name.

`school` The name of the school where a thesis was written.

`series` The name of a series or set of books. When citing an entire book, the the title field gives its title and an optional series field gives the name of a series or multi-volume set in which the book is published.

`title` The work's title, typed as explained in the LaTeX book.

`type` The type of a technical report (for example, *Research Note*).

`volume` The volume of a journal or multivolume book.

`year` The year of publication or, for an unpublished work, the year it was written. Generally it should consist of four numerals, such as 1984.

http://www.ecst.csuchico.edu/~jacobsd/bib/formats/bibtex.html provides a description of other field that may exist. The ones that are already recognized are

`keyword` To specify keywords. Keywords are separated by commas (",") or semicolon (";")

`keywords` id.

`pdf` To specify the way to the pdf file, may be something like FTP://WWW....

`postscript` To specify the way to the postscript file, may be something like FTP://WWW....

`ps` id.

`url` To specifiy some url, may be something like HTTP://WWW....

`abstract` The abstract of the article

`comments` Some comments ...

`alt...` Always ignored.

`opt...` Always ignored.

`archive` Always ignored

`cites` Always ignored

`isbn` Always ignored

`issn` Always ignored

`location` Always ignored

`nickname` Always ignored

`precedes` Always ignored

`site` Always ignored

`succeeds` Always ignored

`update` Always ignored

References

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

[2] Oren Patashnik. BibTeXing. Documentation for general BibTeX users, 8 February 1988.

[3] Mary-Claire van Leunen. *A Handbook for Scholars*. Knopf, 1979.