Texi2ipf 2.2 – a Texinfo to IPF converter

Based on work by M. Groeber and B. A. Erlenbacher

Martin "Herbert" Dietze

1 Introduction

1.1 Historical notice

The *texi2ipf* program is mainly Beverly A. Eschenbacher's work from 1988, 1989 and 1990 as his *texi2roff* was the base for Marcus Groebner making a Texinfo to IPF converter out of it in 1992/93.

As I needed a tool to convert my master's thesis written in Texinfo to IPF I tried it but found it not suitable for my needs – it did not treat *@def** templates correctly and produced blank pages where I had menus in my sources. So I decided to take a closer look at the sources and add the features I found missing and fix a few bugs I found.

I don't claim to have done much on the program, it is still mainly Beverly's and Marcus' work but as nobody else seems to have maintained it any more I have decided to make my modifications public, they have made the program a bit more usable for me after all.

The result works – let's say – not too bad, the sources look like what they are: the result of at least three people doing this or that :-)

1.2 Further notices

The program comes shipping as an OS/2 executable (compiled and linked using IBM's VisualAge C++ tools) with Texinfo documentation (you are just reading it!) and the C sources.

Asking me for license stuff I can only say "no idea!" as it is not only my work. As far as I'm concerned my stuff can be considered as GPL'ed, nobody pays nobody nothing. You may also want to take a look at the original license shipped with *texi2roff* in the file *copyright* for all that was written by Beverly.

If anybody of you out there adds some useful feature or fixes a bug (there *should* be some more) please feel free to send me a mail, I am always interested in further enhancements. My email address is:

        herbert@wiloyee.shnet.org

I will try to release all future versions to at least those two ftp sites / directories:

1. `ftp.cdrom.com/pub/os2/`

2. `ftp.leo.org/pub/comp/os/os2/leo/`

By the way: You may find this manual a bit too verbose for such a small utility, it is indeed, but this also makes it a good demo file for the program's capabilities...

2 Using it

2.1 Usage

The program reads the Texinfo source, translates it and writes the result to stdout. It has no further options. A sample invocation looks like:

```
texi2ipf texi2ipf.texi >texi2ipf.ipf
```

Redirecting the standard output to a file via `">"` creates us an ipf source that can be compiled via:

```
ipfc -inf texi2ipf.ipf
```

If encountering a Texinfo command it does not recognize (there are still some) it will ignore it and emit a warning message. If the Texinfo commands ignored is not important for the ipf output the warnings can be ignored. Only if the program runs into a situation it cannot recover from it will abort with an error message. This is only very rarely the case (e.g. not correctly nested `@..` `@end ..` environments).

The program can, of course, only handle standard Texinfo commands. It will parse all `@ifinfo ..` `@end` ifinfo environments and ignore all `@tex ..` `@end tex` environments. How it treats some special Texinfo features will be described in the next section (see Section 2.2 [Features], page 2).

2.2 Features

Overall features

Like mentioned in the previous section (see Section 2.1 [Usage], page 2) the program parses only `@iftex ..` `@end iftex` environments. As *info* is a hypertext format like *inf* is there is only a small subset of features not covered by both formats. One of them is the missing handling of bitmaps in info, for this I use the pseudo-ipf tag `@ipfline{}` described more below. Here's a list of further features:

- `@def*` templates produce an entry in the main index. Using the `@cindex` command will do this, too. Texinfo-like indices (`@printindex`) are not (yet) supported. See Section 2.3 [Limitations], page 3, for more information.

- Texinfo menus are simulated. This is not really necessary but looks better than blank pages. Please, note that menu descriptions continueing in the line(s) after the menu entry will only be handled correctly if indented more than 25 columns.

- The program will create a node for every document structuring command except the `@*heading` commands (You see two of them in this section!).

- For being able to enter plain ipf code into the Texinfo document I provide a pseudo-tag `@ipfline{}`. Its handling will be described more verbosely below.

Entering plain ipf

As info knows no images inside info documents we may want to enter plain ipf code into our Texinfo documents still being able to process them via *makeinfo*, *texi2dvi* etc. For this task *texi2ipf* recognizes a pseudo-tag `@ipfline{}` containing plain ipf code as argument. To avoid conflicts with other formatting tools you should use it the following way:

```
@c Define this macro somewhere near the beginning of your document.
@c It expands to nothing for "makeinfo" and is necessary to avoid
@c errors when creating an Info file.
```

```
@c The "@ifinfo" block is necessary because the "@macro" command is
@c only known to "makeinfo". Please, note that older versions of
@c "makeinfo" do not recognize the "@macro" command either.

@ifinfo
@macro ipfline{line}
@end macro
@end ifinfo

@c [...]

@c Now somewhere inside the document enter the ipf commands as
@c arguments to "@ipfline{}". In this example we use plain
@c ipf to include a bitmap and some explaining text only into
@c the ipf output. Like the macro above we put in into a "@ifinfo"
@c block because the (empty) macro "@ipfline{}" would cause
@c errors to any formatter but "makeinfo" or "texi2ipf".

@ifinfo
@ipfline{The following example shows a simple LAN consisting of}
@ipfline{three machines connected by a ring&colon.}
@ipfline{:p.:artwork name='lan.bmp' align=center.}
@end ifinfo
```

This will enter the text and the bitmap into the ipf output just the way it is inside of the
`@ipfline{}` arguments.

2.3 Limitations

Disregarding the fact that the program is still pretty buggy <g> there are a few features I would
like it to have but yet it has not:

- The program does not yet recognize all Texinfo commands. This is not so bad as most
  of them are not of interest for producing ipf output. But this still leads to unnecessary
  warning messages.

- There is no support for the `@defindex` and `@printindex` commands. This results in almost
  empty pages only containing "Sorry, no .. index!". I plan to add this but I have no idea
  when.

2.4 Compiling it

First thing I did when starting working on it was changing the code from K&R C to ANSI C. So
it will not compile on a pre-ANSI C compiler any more. It should not make any problems with
non-Unix compilers any more (emx/gcc offers all the Unix functions not available with other
PC compilers).

I have compiled the program using the following compilers / operating systems:

- GNU CC under AIX and OS/2.
- IBM's xlC compiler under AIX.
- IBM's CSet++ 2.1 and VisualAge C++ 3.0 under OS/2.

The makefile contains predefined sections for those compilers that only need to be uncom-
mented.

By the way: Yes I know the code *is* ugly. This is partially my fault as I did not get into the way it works too much when I first started modifying it. It also was quite unreadable when I took my first look at it...

3 Change log

3.1 Release 2.2, July 1997

- Fixed a minor bug with templates. As the template tags are parsed separately things like '&' inside e.g. a `@defun` tag slipped through our fingers.

- Fixed a not so minor bug with menus. Sometimes the end of menu environments were not recognized correctly. This lead to plenty of handywork (try the Texinfo documentation with the old version and you will know what I mean).

- Added a Rexx file written by Hans-Michael Stahl for those who do not like typing more than necessary :-)

3.2 Release 2.1, February 1997

- Added this documentation.

- Added enhanced support for footnotes. Produces real IPF footnotes now instead of Text in brackets.

- Changed the handling of all `@*heading` commands and added the missing `@chapheading` command. They will no longer produce nodes but rather produce real headings like in the Info file.

  This makes it more consistent to the commands' definition saying they do not generate Table-Of-Contents entries. Note, that all the other `@chapter`, `@*section`, `@unnumbered*` commands do *not* generate headings inside the text window as these lines appear in the title of the frame anyway.

- Upped stack size under OS/2 to 0x30000 Bytes.

- Changed the `setclear()` and `findtag()` routine. The program now no longer crashes with large `@set` names.

- `@ignore` will now be handled like real comments.

- Fixed a bug in `eat_first_word()` that made the program crash sometimes.

- Fixed the handling of a few `@def*` commands (index entry).

- Added routine `remove_texicmds()` for better formatting of automatically created index entries from `@def*` commands.

- Menu entry descriptions can now be continued in the next line(s) if indented more than 25 columns. Also use `remove_texicmds()` when processing menu entries.

- Added support for the `@bye` tag for leaving the document before the physical end of file.

3.3 Release 2.0, December 1996

- Converted any K&R stuff to ANSI C (prototypes etc.). Still the code looks rather shrouded :-)

- Changed the table.h for a better handling of all the `@def*` stuff. Now any definition will start with a new paragraph. The leading "–" will not appear anymore. I also added the `@deftp` command and the pseudo- Texinfo command `@ifhtml` (ignored, no error message) that may be used by users of texi2html and the `@macro` command (ignored, no error message).

- Added my name to the banner :-)

- Removed the `#include <string.h>` from texi2ipf.h and added it to those .c files that needed it.

- Texinfo menus will now be simulated in the output .ipf file. This is not really necessary but it looks better than blank pages.

- `@include` statements inside of menus will no longer crash the program.

- All Texinfo `@def*` commands will now produce an index entry in the main index. Support for seperate indices like "Concept index", "Types index" etc. may be possible in future but has not yet been added. The `@printindex` command still produces the "Sorry, no ... index" entry in the resulting .IPF file. Commands dealing with custom indices will now produce a more verbose error (ignore) message.

- Changed the makefile. Compiles well under Unix systems (AIX and LINUX) and with IBM's VisualAge C++ for OS/2.

- Added support for pseudo-Texinfo command @ipfline{} (my invention) for entering IPF commands into the Texinfo source file. The syntax is: "@ipfline{<Sequence of IPF code not longer than one line>}" and it should be put into `@ifinfo` / `@end ifinfo` blocks to avoid trouble with e.g. texi2dvi or texi2html.

Table of Contents