# GOSH Documentation

Norman Feske

12. Januar 2007

## 1 Introduction

GOSH is a tool for converting plain ASCII text to Latex and other document formats. Its main design criteria was to use a syntax that is perfectly readable as plain ASCII text and the support of different target formats. The source files of GOSH have a very simple syntax that is similar to the usenet style. Everyone, who ever wrote a mail using an plain ASCII editor will able to write GOSH texts. GOSH supports multiple target formats by different backends, which are available as separate files. By now, there exists the built-in Latex backend, a simple HTML backend and a man-page backend.

Originally, GOSH was meant as an alternative to writing Latex files by hand. In the meanwhile I do all kind of textual work, papers, documentation, websites and slides using GOSH. As it is a big help for myself, it might be useful for other people, too. Anyhow, this is an early version of GOSH. Its features and usage may change in the future.

## 2 How to get GOSH?

You can download snapshots of GOSH at the `http://os.inf.tu-dresden.de/~nf2/files/GOSH/` of my website `http://os.inf.tu-dresden.de/~nf2`.

## 3 Licence

GOSH and its backends are released under the terms of the GNU General Public Licence. For more information about the GNU General Public Licence visit the official GNU website `http://www.gnu.org/licenses`.

## 4 Usage

GOSH is written in the script language `Tcl/Tk`. Make sure that your installed `Tcl/Tk` on your computer before trying out GOSH. Just check, if you have a program called `tclsh` installed.

GOSH must be called with the source text file as argument and uses standard output for printing its result. For example. a pdf-file of the this text can be created via:

```
> gosh gosh.txt > gosh.tex
> pdflatex gosh.tex
```

In this example, GOSH generates Latex output that is stored in the file `gosh.tex`. This file is then used as input file for `pdflatex`.

A HTML-version of the text can be created by using the HTML-backend:

```
> gosh --style html.gosh gosh.txt > gosh.html
```

The backend to use is specified via the `-style` argument. The `html.gosh` file contains the rules of how to produce the HTML output.

## 5 Text style

Paragraphs are separated from each other be leaving an empty line between them.

If you want to insert verbatim text passages (with monospaced font) - for example source codes, you can mark these lines with a `!` at the beginning of line. For example:

```
void main(int argc, char **argv) {
  return 0;
}
```

### 5.1 Items, Enumations and Descriptions

GOSH supports items by a leading `*`, followed by a space:

- This is an item.
- Items can span over multiple lines. Each subsequent line must be indented by two spaces.

  Even paragraphs within items are possible.

  - Nested items are supported as well.
  - Items can be separated by empty lines to make them better readable in the GOSH text.

Enumerations are marked by a leading # character and behave like items.

1. They can be nested.
2. They can span multiple lines.
3. They can contain multiple paragraphs.

GOSH supports descriptions in a very similar way as items and enumerations. The text to describe is enclosed by colons (`:`), followed by the description text:

**This text** needs some description. The description can span multiple lines and paragraphs. All lines that belong to the description must be indented by two spaces.

Descriptions, Items and Enumerations can be mixed and nested as you like.

### 5.2 Accentuations

GOSH support accentuations for marking **bold** and *italic* text. It uses not the slash (`/`) character to mark italic text because slashes are used in pathnames, which are very likely to appear in GOSH texts. Additionally, underlined text is rarely used. If you want to *mark multiple words* you do not need to (*but you can*) place _ in-between the words. All words `that are written apostrophes` will be considered as monospaced text. This is useful for `filenames` and the like. The beginning and the end of the accentuated text fragment must be on the same line.

*If you need to span accentuations over multiple lines, you need to apply the accentuation at each line. This, way the accentuation is also visible in the GOSH text.*

GOSH detects hexadecimal numbers by a leading `0x` and prints them in a monospaced font automatically.

# 6 Text structure

## 6.1 Head of a GOSH text

The document title is the first text in the document. It is meant to be written centered in the GOSH text. There must be at least one space at the beginning of a title line. Otherwise, GOSH will consider the line as the first paragraph of the document. The title can span multiple lines.

The title is followed by one or more empty lines and the author's names. As for the title, the author's names should be written in a centered way as well.

Example of a header of a GOSH text:

```
            This is the title
              of a gosh the
                document

             Bernd Ullrich
             Uli Berndrich
```

You do not need to specify the author but it is recommended. You can also specify neither the title nor the author. In this case the document is just untitled.

## 6.2 Sections

The chapter's names are underlined with the # character:

```
This is the name of a chapter
#############################
```

Sections are underlined with by = characters:

```
Section
=======
```

Subsections are underlined by ~ characters:

```
Section
~~~~~~~
```

Paragraphs are underlined by `-` characters:

```
Paragraph
---------
```

I choose these underline characters based on their different heights. A `#` is higher than `=` that is higher than ~ that is higher than `-`. This way, the different levels of document structure can be easily differentiated from each other.

### 6.3 Images

Images can be inserted into the document this way:

```
[image filename] This is the caption of the image.
  The caption can span multiple lines. Each line
  must be indented by two spaces.
```

You can control the width of the image in relation to the page width via:

```
[image filename 50%] This image will appear with the
  size of a half page-width.This works only with the
  Latex-backend of GOSH.
```

The image can be rotated via:

```
[image filename 5°]
```

Of course, you can specify both parameters for the same image, too.

### 6.4 Tables

The table feature of GOSH is very preliminary, but it is still usable for a lot of cases. Tables are drawn by `-` and `|` characters. Each line of a table must be led by at least one space. The caption of a table can be written similar to the caption of an image.

| directory | filename | size |
|---|---|---|
| /etc | crontab | 651 |
| | csh.login | 65 |
| | exports | 114 |
| /sbin | route | 46680 |
| | portmap | 12016 |

**Tabelle 1:** This is the caption of the table.

**Note** Two sequent tables without any text in-between cause problems with the current version of GOSH.

### 6.5 References

Chapters, sections and subsections can be referenced within the document by enclosing the corresponding name with brackets. For example, read section 3 carefully. This makes it very easy to insert references but has the drawback, that all referenced section names need to be different to avoid ambiguous references.

Images can be referenced by their filenames (without the extension). I use to store all images of the document in a separate `img/` folder. This avoids naming conflicts of image filenames and section names. Tables can be referenced by their identifiers.

If there is no matching identifier within the document, GOSH assumes the text within the brackets is an external reference (citation). So you can insert citations just the same way as references. This is very practical if you use Latex and Bibtex.

GOSH detects HTML-links by a heading `http://`. For example, take a look at Atari.org `http://www.atari.org`. The HTML-backend supports the link text and a title to be optionally specified. The link text is delimited from the *URL* by a `-`, enclosed with spaces:

```
[http://www.atari.org - The Atari Headquarter]
```

A title can be optionally specified with parenthesis:

```
[http://www.dhs.nu - Dead Hackers Society (click here)]
```

## 7 The HTML backend

By default, the HTML produces pure HTML code without any fancyness. For the different textual styles, the corresponding HTML tags are used.

There exist the following command line options, which take effect on the HTML output:

**`-html-toc`** This option lets GOSH create a table of contents. The entries of the table link to their corresponding section.

**`-html-sec-enum`** By default, chapters and sections are not enumerized. By using this option, you can make GOSH to prepend section numbers to the headlines.

**`-html-p-colored`** This option allows any section type to be colored by a different color. The colors are defined inside the backend and can be over-defined via an additional style as described in section 8.1.

**`-html-p-justify`** This option sets the alignment of paragraphs to justified.

## 8 Advanced features

### 8.1 Tweaking the GOSH output

The output of GOSH can be easily tweaked by supplying multiple `-style` arguments. The style-files will be processed in the specified order. For example if you want to tweak the HTML ouput to not contain a header and tail, you can tweak the `html.gosh` style by another `rawcontent.gosh` style:

```
> gosh --style html.gosh --sytle rawcontent.gosh gosh.txt > gosh.html
```

The `rawcontent.gosh` file contains only the empty versions of the functions for outputting the head and tail:

```
proc produce_head_html {} {
}

proc produce_tail_html {} {
}
```

The `slides.gosh` backend is another example of this technique. It slightly modifies the Latex output of GOSH to create Foiltex output. This way, you can use GOSH to create `slides` very easily.

### 8.2 Raw text

There are things like formulars, which are not supported by GOSH but by the target format (such as Latex). You can insert source code fragments of your target format directly into your GOSH text file by marking such lines by a colon, followed by a space at the beginning of line:

```
: This text will not be touched by
: GOSH. It will be directly written
: out in its original form.
```

When using this method, you loose the feature of GOSH to create different target formats of your document because the raw content will certainly conflict with the syntax of other formats (such as HTML).

### 8.3 Annotations

When writing papers, one often wants to make annotations to preliminary revisions that should be printed in a sligtly accentuated way - so that it is easy to differentiate annotations from real text. As GOSH comments are completely ignored by GOSH, you will not be able to make printable annotations via GOSH comments. Instead, you can use a pipe symbol, followed by a space to mark annotated lines. Within such annotations, you can use items, enumerations, descriptions and accentuations. This makes it easy to convert annotations to real text by just clearing the leading pipe symbol and space.

```
| This is an *annotation* and will
| be written in italic style.
```

## 9 Troubleshooting

In this section you will find some hints for the use of GOSH. It will grow as soon as people will report problems to me.

**“Error: cannot figure out what you mean with”** When GOSH is unable to parse its input file correctly, it outputs an error message “Error: cannot figure out what you mean with”, followed by the trouble-making text. You should revisit this text passage. Mostly, this message is caused by wrong indentation. You should make sure that you indented items, enumerations, descriptions and captions with two spaces and no TABS. Do not use TABS. GOSH does not recognize TABS.

## 10 Known bugs and limitations

- Sequent tables without any text in-between cause trouble.
- The line where an error occured is not printed in the error message. This sounds simple to fix but it is not.
- All references must be written completely on one line, including the brackets.

## 11 Contact

If you have comments, tips or bug reports regarding GOSH, please do not hesitate to contact me via

**Email** nf2@inf.tu-dresden.de

For new versions of GOSH and information about its development, you might visit

**My Website** `http://os.inf.tu-dresden.de/~nf2`.