Mg, a simple macro package for troff

John Ankarström

2021-06-21

Introduction

Mg is a simple macro package for troff with the following features:

1. It is designed to be easy to understand and to customize by editing the source code.
2. It makes use of the extended support for environments offered by many modern troff implementations.
3. It is designed to be practically easy to use. Macros are consistently one letter long and written in lowercase.

While *mg* does provide macros for many common tasks, including footnotes, it is at the end of the day an idiosyncratic macro package, written to serve the author's personal needs. Users of *mg* are encouraged to

a) modify the source code according to their own needs, as well as
b) use built-in troff requests for some things that other packages might provide custom macros for.

All in all, *mg* aspires to abstract as little as possible from the underlying troff requests and registers. In its author's humble opinion, it is the ideal macro package for learning troff.

In this document, the fundamental concepts of *mg* are explained. The reader is encouraged to inspect the document's source code in order to see how the macro package is used in practice.

Environments

Mg makes heavy use of named environments, supported by implementations such as GNU troff and Neatroff. Environments obliviate the need for many special registers that a macro package (and its user) would otherwise need to keep track of. For example, *ms* keeps track of the document's font size in the PS register. For the font size of headings, it has yet another register. *Mg* has no such registers. If the user wishes to modify the default font size, he or she can simply switch to the relevant environment and set the font size as desired using regular troff requests:

```
.\" set heading font
.h
.fam H
.ps +1p
```

Troff saves the font settings in the environment, so that the next time the environment is invoked, the desired font family and point size are automatically restored.

The environments are initialized as soon as the first "block-level" macro is called. At the initialization of each environment, the default environment (0) is copied, meaning that all environment-relevant settings defined before the first macro call are applied to all *mg* environments. It is thus remarkably simple and intuitive to set, for example, the default font of a document:

```
.fam N
.t
Document title
.p
First paragraph.
```

In addition to the normal environment-relevant settings, *mg* manually associates a few special registers with the current environment:

1. `sp`, the amount of vertical space to add before an environment
2. `sq`, the amount of vertical space to add before a different type of environment
3. `ti`, the indentation of the first line in some environments (currently only p)

These can be set inside a given environment to control its behavior when invoked. The only exception are the margin and footnote environments (`_m`, `_f`), which are treated specially and do not support these registers.

Global settings

Thanks to the reliance on environments, there are only four global string values controlling the state of *mg*:

1. `mt`, the margin at the top of the page
2. `mb`, the margin at the bottom of the page
3. `tt`, the text in the top margin
4. `tb`, the text in the bottom margin

The `tt` and `tb` registers are passed to troff's `tl` request and should be formatted accordingly. By default, `tt` is empty and `tb` is set to `''%''`, displaying the page number in the center.

External macros

Mg defines a number of macros. Some of them are used internally by *mg* itself; these carry an underscore prefix and are going to be explored later. For now, we will focus on the external macros provided by *mg*.

Inline macros

There is a group of macros that provide convenient inline formatting. All take three arguments: the text to be formatted, the text to be placed immediately after and the text to be placed immediately before. The inline macros are the following:

1. `b`, bold font
2. `c`, constant-width font
3. `i`, italic font
4. `i`, bold italic font

For example, the following request outputs "*mg*.":

```
.i mg .
```

Note that `c` uses the font family and point size set in the `l` environment (see below).

Block-level macros

There is a large group of macros that provide block-level formatting:

1. `d`, centered date (or given string, or nothing, if given `\&`)
2. `h`, heading
3. `l`, literal display (for source code)
4. `p`, paragraph
5. `s`, subheading
6. `t`, centered title

The `t` and `d` macros can be used at the beginning of a document to create a centered header:

```
.t
Document title
.d \&
First author
.d \&
Second author
.d \" current date
```

Here, you can see that the `d` macro may be used for other things than dates.

Other macros

Finally, there are a few macros that belong to neither category:

1. `(`, begin footnote
2. `)`, end footnote
3. `q`, quotation
4. `w`, want space

The footnote macros (and) are used like this[1]:

```
.p
A line of text\c
.(
A footnote.
.) .
```

Mg does not account for cases where a footnote, because of its height, pushes the place where it is referenced to the next page. In such cases, the reference will simply appear on the page following the footnote. When this happens, the user must account for it manually.

The q behaves like an inline macro by default, surrounding text in quotation marks, but if no arguments are given to it, it starts an indented block quotation instead:

```
.q
This is an indented quotation.
.p
This is an inline
.q quotation .
```

The w macro is a wrapper around troff's ne, which ensures that a given amount of space is available on the page — otherwise, a line break is issued — but unlike ne, which takes an exact amount of space as its argument, w takes a declarative specification describing the amount of space desired in terms of *mg* environments. For example:

```
\" want space for...
.w s p \" a subheading of one line + a paragraph of one line
.w s pp \" a subheading of one line + a paragraph of two lines
.w ss p \" a subheading of two lines + a paragraph of one line
```

Internal macros

Mg's internal macros are generally not meant to be used outside of *g.tmac*. _E, _e and _ec, a group of macros controlling the environment, are excepted from this rule, as it may be desirable to define custom environments for advanced documents.

Frequently anticipated questions

How do I customize the default appearance of a document?

All environment settings, like point size, font family and indentation[2], are configured with the standard troff requests. If you set the point size at the beginning of the document, before any *mg* macros have been called, it will be correctly set for the entire document. This works because *mg*'s environments initially copy all their settings from 0, the default environment.

For example, if you wanted to write a document with the New Century Schoolbook font at 9 points and a vertical spacing of 12 points, you would start the document like this:

```
.so g.tmac \" include mg
.fam N
.ps 9p
.vs 12p
```

How do I customize the default appearance of a given environment?

To configure the layout and font settings of a specific environment, you can switch to that environment and use the relevant troff requests:

---

[1] Note that (and) take an optional argument, which is output immediately before or after the footnote number.

[2] For a complete list of settings that are associated with the environment, see 5.26 Environments in the full GNU troff manual (`info '(groff.info)Environments'`).

```
.q
.ps +1p
```

If you are merely configuring the environment without printing anything in it, you can also use the `_e` macro:

```
._e q
.ps +1p
```

How do I customize the default appearance of the margin text?

```
._E _m \" margin environment
.ps +1p
```

How do I customize the default appearance of footnotes?

```
._E _f \" footnote environment
.ps +1p
```

How do I prevent a section from being broken up by a page break?

Some macro packages have a concept of "keeps", sections that are kept together across page breaks. *Mg* does not define any helper macros for this purpose by default. The user is recommended to use troff's `ne` request:

```
.q
.ne 7
Four score and seven years ago...
```

In some cases, *mg*'s own `w` macro may be useful:

```
.w qqqqqqq
.q
Four score and seven years ago...
```

How do I define my own environments?

Environments are a feature built into troff, accessed via the `ev` request, but because *mg* extends environments with additional functionality, it provides special macros to be used instead of `ev`:

1. `_E`, set environment (same as `ev`)
2. `_e`, set extended environment
3. `_ec`, copy environment (same as `evc`)

The `_e` macro can be used to activate any environment that supports *mg*'s extensions (see *Environments* above). The following code configures an environment called `n` and defines a corresponding macro:

```
._e n
. _ec 0 \" copy default environment
. ps -1p
._e
.
.de n
. br \" finish current environment
. _e n \" activate new environment
..
```