

# ***Toc*, a multipass framework for troff**

John Ankarström

23 June 2021

## **Introduction**

*Toc* is a very simple solution – in 50 or so lines – to the problem of generating tables of contents and other forms of forward references in troff documents. It consists of a shell script called *toc* and a macro package called *toc.tmac*.

The *toc* script is a wrapper around troff, passing the document through the typesetter three times. In every pass, it sets the values of two registers named *te* and *to*.

The *toc.tmac* package defines two macros, also named *te* and *to*. If the *te* register is non-zero, the *te* macro hands its argument to the *tm* request, which prints it on standard error. If the *to* register is non-zero, the *to* macro invokes the *rd* request, reading and inserting lines from standard input into the troff document.

## **Operation**

In the first pass, the *toc* script sets *te* = 1 and *to* = 0. This enables the *te* macro, printing its arguments on standard error. The lines printed on standard error are piped to the second invocation of troff.

In the second pass, *toc* sets *te* = 1 and *to* = 1. This enables the *to* macro as well, inserting the lines given on standard error into the troff source. The *te* macro is still activated and accordingly prints its arguments on standard error yet another time. Like before, the output is piped to the third invocation of troff.

In the third and final pass, *toc* sets *te* = 0 and *to* = 1. This disables *te*, which means that nothing is printed on standard error, but *to* is still enabled, once again inserting the input on standard error into the document.

Three is the number of passes necessary – and sufficient – for generating forward references, such as tables of contents. Two passes are not enough, as the generated references may push a referent to the next page, rendering the generated references incorrect. To account for the addition of the generated references, a third pass is needed.

## **Usage**

### **Macro package**

The *te* and *to* macros do not apply any formatting to or perform any processing of their input. The *te* macro prints its arguments on standard error verbatim. For example, the request

```
.te .nr &a \n%
```

will print

```
.nr &a \n%
```

on standard error. Let us assume the page number is 3. When *to* is invoked, it will literally insert

```
.nr &a 3
```

into the troff source.

As such, *toc* provides the tools needed to create forward references, including tables of contents, but the exact formatting must be programmed by the user himself.

## **Script**

The *toc* script is designed to be called instead of troff. It is not a troff preprocessor, but a wrapper around it. Any arguments given to *toc* are passed to troff (or whichever program specified in the *TROFF* environment variable).

Pre- and post-processors can be added by setting *TROFF* to a custom shell script:

```
$ cat xroff
#!/bin/sh
... | groff "$@" | ...
$ TROFF=./xroff toc document.t
```

## Examples

### Table of contents

```
.so toc.tmac
.de he
. ft B
. sp 1v
\${*
. br
. ft
. te .the \n% \${*
..
.de the
. nr _ \${1
. shift
. ta 0 \n(.luR
. tc .
\${*      \n_
. tc
. br
..
.sp |1i
.to
.\" ...
.he First heading
.\" ...
.he Second heading
```