**Open Software Foundation**
**Request For Comments: 0.1**
**October 1994**

**W. Tuvell (OSF)**

# OSF-RFC TEMPLATE AND STYLE GUIDE

## 1. INTRODUCTION

This document provides a template and style guide for the OSF-RFC[1] series. Its audience consists of the *authors* of OSF-RFC's, and the OSF-RFC Editor *requires* that all OSF-RFC authors observe its instructions. An earlier attempt at providing a template and style guide was given in [RFC 0.0] (see Section 4.1 of that document) — the present document supersedes that one (for the purposes of template and style guide).[2] Barring potential bug-fixes, this template and style guide is now believed to be in final form.

Readers of OSF-RFC's have commented that a consistent style enhances both the readability of each individual OSF-RFC, as well as affording a sense of identity to the OSF-RFC series as a whole. Both of these observations indicate that a consistent style increases the overall value of OSF-RFC's. This gives the Editor sufficient reason to require that this template and style guide be used by all authors.[3]

Among the OSF-RFC's that have been published to date (see [INDEX]), the Editor has observed a wide disparity in the quality of both the *sources* (`.roff`'s) and the styles of the *formatted* versions (`.txt`'s and `.ps`'s), *as originally submitted*. In an effort to regularize that disparity, the Editor has worked with authors to devise a standard style for the OSF-RFC's *as actually published*, but that style has not yet been well documented. This paper provides the heretofore missing documentation.

The `nroff`/`troff` (or `*roff`) tools used by the OSF-RFC series were chosen for their near-ubiquity and virtual standardization within the DCE community.[4] Nevertheless, it is recognized that it would be a hardship for some authors to gain access to them. In such cases, authors are expected to exert a ''good faith'' effort to cast their OSF-RFC's into this template's format (they should be able to get it > 90% right), and then work with the Editor to finish the job.

As usual, readers are assumed to be familiar with [RFC 0.0] before reading this document. They're also assumed to be familiar with `*roff`.

### 1.1. Default Reference

The default reference work for non-OSF-RFC-specific style issues is [CHICAGO].

## 2. GENERAL INSTRUCTIONS

(a) The basic idea is that authors are supposed to retrieve a copy of the most recent reissue[5] of this template

---

1. Early numbers in this series (namely, numbers prior to 62.0) were designated as "DCE-RFC's" instead of "OSF-RFC's", and were sponsored by the DCE SIG (see [RFC 0.0] for historical details). The scope of the series was subsequently broadened to include all OSF initiatives, not just DCE, hence the change of name. The series is now considered to be sponsored by all OSF member organizations, not any specific SIG.

2. Only a few things have changed from the earlier template. The ones visible to authors are: (1) italic/emphasis font(s) (`\*(iB`/`\*(iE`) now prints as underscores or angle-brackets in `.txt`; (2) a bold font toggle has been added (`\*(bB`/`\*(bE`), and it prints as asterisks in `.txt`; (3) the former ''secondary font'' toggle `\*(sB`/`\*(sE`, is now called the ''literal font'' toggle `\*(lB`/`\*(lE`, with a slight change in semantics (matched single quotes in `.txt` instead of double); (4) the former ''display font'' toggle has disappeared (it is no longer necessary, because of change (3)); (5) some ''indent''/''display'' macros have been added (`.iS`/`.iF`, `.cS`/`.cF`, `.dS`/`.dF`); (6) a ''note'' macro has been added (`.nS`/`.nF`).

3. It's also too much of a burden on the Editor to supply styling markup services, so this task is being pushed back onto the authors!

4. This truism is mitigated somewhat by the fact that `*roff` does not provide an *ideal* markup language, and is expected to be replaced by some as-yet-unknown system, at some unforeseeable time in the future.

5. Updates to this document are intended to be quite small, and so will generally be made ''surreptitously'', so as not the flog the OSF membership with OSF-RFC (pseudo-)announcements.

(`rfc0.1.roff`), throw away the substantive contents, and wrap the resulting scaffolding around their OSF-RFC. Authors should not diddle with the scaffolding itself, except for adding document-specific strings and macros, etc. (Readers (not authors) desiring different offprinting conventions can, of course, do anything they want.)

(b) The *`0.1`* designation on this template is to be replaced by the actual OSF-RFC number. The Editor will assign this, either before the author submits the OSF-RFC, or at submission time.

(c) Similarly, the author should insert the correct full title, short title (for the page header), authors' names (both for document header and for page footer), addresses, date (month and year), etc.

(d) The formatting instructions for each OSF-RFC are indicated in the ''FORMATTING INSTRUCTIONS'' section near the top of the `.roff` file. Separate commands are given for `nroff`/`.txt` version (which must *always* be supported by every OSF-RFC) and for the `troff`/`.ps` version (which is optional per OSF-RFC, at the author's discretion). The author is responsible for making sure these instructions are correct (it is only rarely that an author will have to change the defaults in the template.

Note that many of the intricacies of this style guide simply disappear if an author is satisfied with supporting only the `nroff`/`.txt` version, and the requirements of this style guide will be partially relaxed for such authors. For example, such authors do not have to deal with font changes, and they can hard-code quotation marks instead of invoking `\*(qB`/`\*(qE`, etc. Such RFC's should abort if they are formatted with `troff`, as follows:

```
.if t \{\
.tm *** Troff not supported for this RFC.
.ex \}
```

(e) The template sets up several `*roff` options that are not discussed explicitly here, and are usually not needed by authors, but authors can read about them in the template itself if they really want to.

(f) There are no defined tools to support any of the above — the author must ferret out the places in the template that need to be modified. (That's easy to do.)

## 3. *ROFF COMMANDS; MM STRINGS AND MACROS

The following standard `*roff` requests, and `mm` macros and strings are useful in OSF-RFC's:

(a) `.P` — Paragraph. This macro (and not a blank line) should always be used to indicate paragraph breaks. Using `.P`'s liberally enhances readability of the source file (stacked `.P`'s do not introduce additional blank lines on output).

(b) `.H` *`N`* `"`*`Title`*`"` — Section headings. The required capitalization style of the *`Title`* is illustrated as follows:

(i) First level (*N* == 1):
*CAPITALIZE ALL LETTERS*

(ii) Second level (*N* == 2):
*Capitalize Initial Letters of All (Significant) Words*

(iii) Higher levels (*N* >= 3):
*Capitalize first letter of first word only*

(c) `.br` — Break line. Used for special effects, such as in the previous list item on section headings.

(d) `\*(EM` and `\(mi` — Em-dash and en-dash. `\*(EM` produces a true em-dash in `troff`, a double-hyphen in `nroff`. `\(mi` (literally, a minus sign) produces a good en-dash in `troff`, a single hyphen in `nroff`. Em-dashes should be preceded and followed by whitespace (not abutted up against the words before and after them). One good place to use em-dashes is to introduce definitions, as has been done throughout this template. A good place for en-dashes is in dash lists.

(e) `.ta` *`M +N`* `...` — Tabs. This can be used for tabular/columnar output, and it works equally well for both `nroff` and `troff`. The best way to compute the tab stop positions (*M*, *N*, etc.) is via the width

(`\w$`*`string`*`$u`) function (where *`string`* denotes the widest string in its column). Also recall that tabbed input works best inside a no-fill (`.nf`) scope. See the actual examples illustrated in this template.

(f) `.ne` *`N`* — ‘‘Need’’ requests, for ‘‘block mode’’ text (text that shouldn’t be broken across page breaks). Use judiciously: long blank sections at the bottom of pages is unsightly, and it’s often better to forgo block mode in such cases. A common choice is to use block mode for tables and pictures, and sprinkled within code fragments (to keep logically related code blocked together).

(g) `.DS`/`.DE` (especially with standard indentation: `.DS I`) — Display macros. Experience shows that these macros tend to be a little clumsy:

(i) They don’t allow fine-grain control over blocking (i.e., `.ne` *N*’s make no sense inside them), though that can be overcome by using sequences of `.DS`/`.DE`’s.

(ii) They can’t be mixed with footnotes in any combination, but that’s a useful thing to want to do.

(iii) They can’t be used to block the author’s address at the end of a OSF-RFC with the ‘‘AUTHOR’S ADDRESS’’ heading).

(iv) The centering option (`.DS C`) does not take into account the 3 space offset in the `.txt` version of OSF-RFC’s, and shouldn’t be used.

A viable alternative is to use the other display macros described below (`.iS`/`.iF`, `.cS`/`.cF`, `.dS`/`.dF`), which have font toggles built into them, and can be used with `.ne` *`N`* requests to allow fine-grain control over blocks of text. Economy of mechanism would argue for just using these and not `.DS`/`.DE`.

(h) `.if n`/`.if t` — These directives can be used to special-case `.txt`/`.ps` constructs. For an example, see the double-arrow in the ‘‘SAMPLE NOTE’’ below.

## 4. OSF-RFC STRINGS AND MACROS

The following strings and macros are defined in this template. By convention, the words ‘‘Begin/End’’ are used for strings, while ‘‘Start/Finish’’ are used for macros.

(a) Fonts. The standard fonts used in (PostScript versions of) OSF-RFC’s are:

(i) Times — For the body of the document.
(ii) Courier (Constant-Width) — Literal font, for ‘‘object-language’’.
(iii) Helvetica — For document meta-data (headers, footers, etc.).

The following strings are defined, to invoke the most common usages of these fonts:[6]

(i) `\*(iB`/`\*(iE` — Italics begin/end (true italics in `.ps`, underscores in `&.txt`). To be in most places where emphasis is desired.

(ii) `\*(bB`/`\*(bE` — Bold begin/end (boldface in `.ps`, asterisks in `.txt`). To be used sparingly. A good place to use this is for **definitions**, where new terms are introduced or defined for the first time.

(iii) `\*(lB`/`\*(lE` — Literal (i.e., object-language) begin/end. It prints in the literal font in `troff`, as matched single quotation marks in `nroff`. It should be used for ‘‘object language’’ words, such as:

[a] Code excerpts.
[b] Manifest constants.
[c] Names of functions.
[d] Names of files.
[e] Namespace names.

---

6. These strings are usable in both `nroff` and `troff`, though they actually toggle fonts only in `troff`, of course.

[f] Etc.

(iv) `\*(jB`/`\*(jE` — Literal italics begin/end (true literal italics in literal font family in `.ps`, angle-brackets in `&.txt`). To be used *inside* `\*(lB`/`\*(lE`.

(b) `.aL`/`.LI`/`.LE` — Lists. Lists enhance readability and referability, and should be used liberally. The `.aL` macro should be used for all lists (except as noted otherwise in this paper). By its definition, `.aL` generates list-item markers of the following form:

(i) 1st level: (a) (b) (c) ...
(ii) 2nd level: (i) (ii) (iii) ...
(iii) 3rd level: [a] [b] [c] ...
(iv) 4th level: [i] [ii] [iii] ...
(v) 5th level: {a} {b} {c} ...
(vi) 6th level: {i} {ii} {iii} ...

In-line lists should be numbered (by hand): (1), (2), (3), .... Lists within notes (`.nS`/`.nF`) should be single-level dash lists (use: `.LI "\(mi"`), or preferably avoided altogether.

To generate a single list item without a (visible) list item marker, use an unpaddable space as the item marker: `.LI "\ "`.

Finally, list items should in general be separated by a blank line, but that can be adjusted on a case-by-case basis by judicious use of `.nr Ls` *N* (0 <= *N* <= 6) requests (as has been done at various places in this template).

(c) `.iS`/`.iF` — Indent. These macros merely indent, the same distance as `.aL`.

(d) `.cS`/`.cF` — Code display, without indentation. These macros do the following: (1) toggle no-fill/fill (`.nf`/`.fi`) mode; (2) toggle constant-width/normal font. They do not invoke block mode automatically, but the author can do that by hand if desired (via `.ne` *N*).

This macro pair is usually used for extended code excerpts (see Appendix B).

(e) `.dS`/`.dF` — Display with indentation. These macros do the same as (and are used more often than) `.cS`/`.cF`, except that they additionally indent the same distance as `.aL`.

Example:

```
#include <dce/rpc.h>

void
rpc_ns_binding_lookup_next (
    rpc_ns_handle_t       lookup_context,
    rpc_binding_vector_t **binding_vec,
    unsigned32            *status );
```

(f) `.nS [`*STRING*`]`/`.nF` — Notes.

*SAMPLE NOTE: The* `.nS`/`.nF` *macros are used for ''meta-comments'', like this one. For example,* Notes to Reviewers *and* Open Issues *are of this character. The* `.nS` *macro indents a standard amount; in the* `.ps` *format it reverses primary and emphasis fonts (roman ↔ italic), and all font toggles (such as* `\*(lB`/`\*(lE`*) still work. The author supplies an optional* *STRING*, *which should be in all capitals, either at the beginning of the* `.roff` *or on a per-invocation basis (the default is* `NOTE`*); the macro itself supplies a '':''.*

(g) `\*(f!`/`.FS`/`.FE` — Footnotes. The string `\*(f!` introduces footnotes; it resolves to a bracketed number in `nroff`, and to a superscript in `troff`. The macros `.FS` and `.FE` bracket the body of the footnote.

Note that if `\*(f!` occurs at the end of a sentence which is not the end of a paragraph, the first word of the following sentence, preceded by 2 spaces, should appear on the same line as the `\*(f!`. (Otherwise, `*roff` is not smart enough to put 2 spaces between the sentences.) Examples occur in this document.

(h) `\*(qB`/`\*(qE` — Quotations. These strings resolve to either double quotation marks (in `nroff`), or pairs of single quotation marks (in `troff`).

## 5. SPECIAL TOPICS

(a) Standard tools (`tbl` for tables, `pic` for pictures, `eqn` for equations) that work well in `troff` but not in `nroff`. There are really only 2 general solutions to these problems:

(i) Draw the desired thing in ‘‘typewriter art’’, and include it within a `.dS`/`.dF` (or similar) scope. This has the advantage of being simple, but it is not as beautiful as it could be.

Example:

```
+-------------------------------+
| Here are some words in a box. |
+-------------------------------+
```

(ii) Use typewriter art within a `.if n \{ ... \}` scope for `.nroff`, but use the fancier tools within a `.if t \{ ... \}` scope for `troff`. This is a beautiful (i.e., ‘‘best’’), but it has the disadvantage of introducing multiple sources for the same content (and therefore doubling the author’s work).

If you use any non-standard tool to help you (i.e., other than `tbl`/`pic`/`eqn`), include the source for that, too. For example, if you use `xfig` to draw a picture, include the corresponding `.fig` source, too, as well as the `.pic` generated by `fig2dev −L pic`.

Example:

Here are some words in a box.

(iii) It’s also possible to use `.ta` to simulate `tbl` for some simple tables, but that’s not a general solution. If you’re going to go to that level of trouble, use the preceding solution instead.

(b) Authors may, at the author’s option, embed ‘‘diffmarks’’ in OSF-RFC’s, via the usual *margin character* construct (`.mc` *c*). When first published, OSF-RFC’s displaying diffmarks will be archived for a short term (approximately one month), and named with the extention `.diffmk` (i.e., `.txt.diffmk`, `.ps.diffmk`). However, the long-term archived versions of OSF-RFC’s (`.roff`/`.txt`/`.ps`) will not include the diffmarks (the `.mc` request will always be defined to be null in the archived `.roff`). This is done because the diffmarks are not usually desired beyond the initial publication of a OSF-RFC, and besides if diffmarks are desired they can always be reinstated by removing this null definition from the `.roff` and regenerating the `.txt.diffmk`/`.ps.diffmk`.

## 6. MISCELLANEOUS

(a) All sentences should start with a capital letter, and end with a period. (Some authors tend to write definitions or lists without an initial capital letter, or without a terminating period, but that should be avoided.) The only exception to this is sentences that begin with a keyword having a specified syntax beginning with a lowercase letter (though in most cases, such constructions can be avoided).[7]

(b) Two spaces should separate sentences (some authors tend to use only one); one space separates other things. Thus, two spaces should occur after periods (also question-marks, etc.) that separate sentences. But only one space occurs after colons, or after the number that identifies footnotes, etc.

---

7. Similarly, such words are not capitalized if they occur in section headings.

Note that `*roff` recognizes periods at the end of input lines, so it's OK to end lines in `.roff` with periods. But `*roff` is not smart enough to do the same thing with other sentence-ending punctuation (question marks, exclamation marks, etc.), so the author must not end lines with these. Similarly, the author must be careful about ending sentences with footnote marks (the first word of the next sentence must occur on the same line as the footnote mark, separated from it by two spaces — see examples in this template).

(c) Lines in the source (`.roff`) file should be limited to 72 characters in length, for maintainability reasons. This can always be accomplished with the ''*newline*'' construction.

(d) No line in the `.roff` file should end with a space character. If a space character is desired in that position, append the zero-width non-printing character `\&` to the end of the line (after the space), or use some such construct as `"\ "`.

(e) No tab characters should appear in a OSF-RFC (e.g., some mailers eat tabs), though this is sometimes unavoidable, as in field delimiters for `.ta` requests. In particular, code fragments should not contain tabs.

(f) Plurals of ''strange'' words can be formed with or without an apostrophe. Thus: ''RFC's'' or ''RFCs''. See [CHICAGO].

(g) ''Requirements'' should be clearly tagged with the string `REQT:`.

(h) References to other sections in OSF-RFC's must be given by ''author's divisions'', not by page numbers. For example, ''Section 1.2.3.4'' (with a capital ''S''), or ''Appendix D.3.2.1'' (with a capital ''A'').

(i) The abbreviations ''i.e.'' and ''e.g.'' have 2 periods in them, and are always separated from surrounding text by punctuation such as commas or parentheses. Similarly for ''etc.'', except that it only has 1 period in it.

(j) Run a spell-checker.

(k) In general, OSF-RFC's should be thoroughly proofread before they are sent to the Editor. The Editor makes an editorial perusal of RFC's, but should not be expected to make up for authorial deficiencies.

## 7. ACKNOWLEDGEMENTS

(a) An Acknowledgements section, if one is desired, occurs as the last numbered section, before the Appendices.

(b) All persons acknowledged here (or elsewhere in a OSF-RFC) must be tagged at least once with their organizational affiliation.

## APPENDIX A. THIS IS AN APPENDIX

(a) Appendix headings are generated by the `.aH` macro. This macro denotes Appendices by letters, not numbers.

### A.1. This is a Subsection within the Appendix

(a) Subsections of Appendices are generated by the usual `.H` $N$ ($N$ >= 2) macros.

## APPENDIX B. EXTENDED CODE EXCERPTS

(a) Extended code excerpts should be put in appendices. They should be delimited by the code display macros, `.cS`/`.cF`.

(b) It is important to recognize that the line length of code samples (everywhere, not just in appendices) must be adjusted to fit onto the `.txt`'s lines, which are only 69 characters wide (because the `nroff` width (`W`) register is initialized to 72 in the template, and the offset (`O`) register is initialized to 3).

(c) Recall (as was stated above), no tab characters should occur in code excerpts.

(d) Also, remember that the backslash character is a metacharacter for `*roff`, so any `\` characters that occur in code excerpts have to be changed to `\e`. A common place for this to occur is in the newline character (`\n`) in `printf()` statements.

(e) This style guide does not enforce any specific coding style, other than to say, ''use accepted DCE coding standards''.

## REFERENCES

(a) References have the form: `[`*`mnemonic`*`]` (in particular, don't use the predefined `mm` reference strings and macros). The following mnemonics are established, and authors can invent others not matching these:

(i) [RFC M.m] — OSF-RFC references.

(ii) [Tuve] — First 4 letter of individual author's last name (in this case, Tuvell).

(iii) [Tuve 1], [Tuve 2], etc. — Successive references by same author. Year references ([Tuve 89], [Tuve 90a], [Tuve 90b]) are also acceptable.

(iv) [JLMT] — Initial letters of (up to 4) co-author's last names (in this case, Johnson-Leser-Mackey-Tuvell).

(b) Every reference must be referenced *at least once* from the body of the document.

(c) The list of references is generated by a `.VL 12 0` macro, giving the result illustrated below. (Note that mnemonics having spaces in them must be printed with unpaddable spaces (`"\ "`) in the `.VL` list.)

[CHICAGO] *The Chicago Manual of Style*, 13th Edition, 1982, The University of Chicago Press. Also, *Chicago Guide to Preparing Electronic Manuscripts*, 1987, The University of Chicago Press.

[INDEX] W. Tuvell, ''Index to OSF-RFC's'', various dates.

[RFC 0.0] W. Tuvell, ''Introduction to OSF-RFC's (Required Reading for All Users)'', May 1992.

## AUTHOR'S ADDRESS

Walter Tuvell
Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142
USA

Internet email: walt@osf.org
Telephone: +1-617-621-8764