

To: PLM Editors  
From: Steve Webber  
Subject: Rules for and Responsibilities of PLM Editors  
Date: 04/02/74

This memo intends to spell out the duties of PLM editors and therefore indirectly to define the duties of PLM writers and updaters. The general philosophy of the PLMs is that they contain the design and implementation descriptions for all of the Multics software. The manuals are divided into what are hopefully reasonable subsets of the system. The subsets are, for the most part, disjoint but some parts of the system design are obviously relevant to several classifications/areas and hence, are partially specified in more than one PLM.

There is no strict format required of PLMs; however, certain Honeywell documentation rules must be followed. A list of the more important format and style rules is attached. In general (within the Honeywell documentation restrictions) the editor of a PLM has complete freedom to format a particular PLM as he sees fit. The content should be approximately that which is mentioned in MAB 007.

The following duties apply to each PLM editor:

1. The editor must make sure the PLM is up-to-date. This means he must elicit the necessary documentation from the appropriate people. It is the editor's responsibility that a PLM be updated when changes are made to the affected part of the system.

During the initial stages of PLM conversion, this duty of the editors is even more important. The initial release of all PLMs is to take place within a few months.

After the initial stage of "catching up" with the documentation, all submissions will require that any necessary documentation be attached.

2. The editor must verify that the final copy of a PLM is accurate, in acceptable Honeywell format, and free of spelling and typographical mistakes. All contributors (writers, typists, etc.) must help in this area but the

---

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

editor is responsible for what gets published.

3. The editor must verify that the PLM writers have followed the rules below for PLM documentation.

The following rules and recommendations apply to all PLMs:

1. PLMs should define all data structures, key variables and terms that are relevant to the particular topic and implementation.
2. PLMs should contain clear descriptions of all algorithms used and all nonobvious or nonstandard coding practices used.
3. PLMs should describe clearly the reasons for doing things in a particular way when there are obvious alternatives.
4. PLMs should give enough information so that a new or interested programmer can easily understand the implementation described by the PLM. It is assumed that the program listings will be used in conjunction with the PLMs by people working at a detailed level. (The program listing contains calling sequences and parameter descriptions.)
5. PLMs should provide an easy mechanism for an interested party to learn the hows and whys of a particular implementation.

# GUIDELINES FOR MULTICS DOCUMENTATION

## STYLE

### Hyphenation

1. the prefixes multi, pre, non, re, pseudo, etc. do not take a hyphen.
2. log-in, logged-in have a hyphen if used as a modifier. log in as a verb is two words. The command login is one word.

### Names

CORRECT	WRONG
DATANET 6600 Front-end Network Processor DATANET 6600 processor	DATANET-6600 DN6600
DSS190 Series 6000 Model 6180 GE TermiNet 300 Honeywell nonprivileged terminal	DSU190, DSS-190 HIS/GE-600 H6180, H-6180 TermiNet 300 we, us, our unprivileged console, typewriter

### Punctuation

Commas and periods go inside quotation marks, colons and semicolons go outside. Literals override this rule; for them, punctuation is outside.

Format

Headings must follow the rules below for capitalization:

first level	<u>FULL CAPS, UNDERLINED</u>
second level	<u>First Caps, Underlined</u>
third level	FULL CAPS
fourth level	First Caps
fifth level	First Caps: text begins on the same line

Section names are in full caps, centered and always begin a page. The word "section" is also in full caps and the section number is in uppercase roman numerals.

Each section begins with the following:

```
.m1 6
.m2 0
.m3 4
.m4 6
.fo 1 "DRAFT: SUBJECT TO CHANGE"X-%"order_number"
.sp 7
.ce 2
SECTION (uppercase roman numeral)
.sp 2
SECTION NAME
.sp 4
```

where the X in the footer line is the arabic value of the section number.

## Figure and Table Titles

1. Keep them short.
2. Use initial caps (excluding words like a, an, the, to, etc.).
3. Place figure title below the figure and table title above the table.

## Front Matter

1. Always use a table of contents (entitled CONTENTS); it must include the section names plus at least the first two heading levels.
2. Use list of figures (entitled ILLUSTRATIONS) when document has two or more figures.
3. Use list of tables (entitled TABLES) when document has two or more tables.

## Sample Table of Contents

```
.m1 7
.he 1 ""CONTENTS""
.he 2 """"
.he 3 """"
.he 4 """"
.he 5 """"Page""
.m2 2
.ro
.pa 3
.fo 1 "DRAFT: SUBJECT TO CHANGE%"order_number"
.nf
Section I      Introduction..... 1-1
                First Phase..... 1-3
.m1 4
.he 1 ""CONTENTS (cont)""
                Second Phase..... 1-5
                    Part A..... 1-9
                    Part B..... 1-10
.sp 2
Section II     Structure..... 2-1
```

## Table of Contents Rules

1. Make all headings first caps regardless of their appearance in text.
2. Use no spaces between items within one section.
3. Indent each new level three spaces.
4. Use .sp 2 between sections.

## Pagination

1. Split lists so that at least two items appear on one page.
2. Split text so that at least two lines appear on one page.
3. Do not start a new heading at the bottom of a page unless at least four lines of text can appear with the heading.
4. Never end a page with a colon (i.e., never end a page with the introduction to a list).
5. Control page breaks using either .bp or .ne.

## MISCELLANEOUS

### Spacing

1. Indent all paragraphs five spaces.
2. Use .sp 2 between paragraphs.

3. Use .sp 3 before headings.
4. Use .sp 2 before and after a list.
5. Indent an example five spaces.

### Lists

1. Use arabic numbers to identify separate items.
2. Use .sp 1 between items of a list.
3. Indent the item number five spaces and the text eight.  
(Use .in 8 and .un 3)
4. If it is necessary to list items within a list, the inner list is identified with lowercase alphabetic characters, starting at the ninth character position (use .in 11 and .un 3). AVOID THIS.