

Road

To: Distribution

From: G.J. Pillsbury

Date: February 13, 1975

Subject: Document Standards, MPM Descriptions

This document establishes a general format standard for MPM Commands, MPM Subroutines, and MPM Subsystem Writers' Guide descriptions. Standard headings and methods of referencing other descriptions and documents are established as well as standard formatting conventions. Also included is a list of miscellaneous conventions and general rules that have been established in order to standardize the descriptions. This discussion is to be considered a supplement to the HIS Publications Standards manual. The latter document should be referred to for more general information about technical manual preparation, e.g., punctuation, effective writing, word usage, etc.

HEADINGS

In general, the command and subroutine descriptions have the following headings:

Name:

Usage

Entry:

Notes

Examples

The Name and the Usage headings must appear in every description; all other headings are optional and depend on the length and complexity of the description. Notes, for example, might require several headings specifically related to subject matter instead of a Notes heading followed by a list of short statements. All of the headings, including any miscellaneous headings that may occur, will be initial caps, underlined (level 2 head). Only Name, at the beginning of every description, and Entry, when it occurs, are followed by a colon. With these two exceptions, descriptive text will begin on a new line, indented five spaces, and a double space between the heading and first

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

line of text. The name of the command or subroutine (and its abbreviation) will appear on the same line as the heading with two spaces between the colon and the name. The other exception, the heading Entry, is treated the same as Name.

Usage and Control Argument Lists

The Usage heading and the supporting text will be treated as shown below. Several examples are illustrated. The first shows the runoff format with the runoff control lines and the succeeding examples show the text as formatted by runoff.

EXAMPLE 1.

Usage

.sp 1

copy path1_i path2_i ... path1_n path2_n -control_args-

.sp 1

where:

.sp 1

.in 24

.un 24

1. path1_i is the pathname of a segment or multisegment file to be copied.

.sp

.un 24

2. path2_i is the pathname of a copy to be created from path1_i. If the last path2_i argument is not given, the copy is placed in the working directory with the entryname of path1_n.

.sp

.un 24

3. control_args can be chosen from the following list of control arguments:

.sp

.un 24

-name, -nm copies multiple names.

.sp

.un 24

-acl copies the ACL.

.sp

.un 24

-all, -a copies multiple names and ACLs.

.sp

.un 24

-brief, -bf suppresses the warning messages

"Bit count inconsistent with current length..." and "Current length is not the same as records used...."

.sp 2

.in 0

The control arguments can appear once anywhere in the command line after the command name and apply to the entire command line.

EXAMPLE 2.

Usage

```
copy path1i path2i ... path1n -path2n- -control_args-
```

where:

1. path1_i is the pathname of a segment or multisegment file to be copied.
2. path2_i is the pathname of a copy to be created from path1_i. If the last path2 argument is not given, the copy is placed in the working directory with the entryname of path1_n.
3. control_args can be chosen from the following list of control arguments:
 - name, -nm copies multiple names.
 - acl copies the ACL.
 - all, -a copies multiple names and ACLs.
 - brief, -bf suppresses the warning messages "Bit count inconsistent with current length..." and "Current length is not the same as records used...."

The control arguments can appear once anywhere in the command line after the command name and apply to the entire command line.

EXAMPLE 3.

In the event that only a single argument is permitted and therefore described, Usage is formatted in the following example.

Usage

```
delete paths
```

where paths are the pathnames of the segments and/or multisegment files to be deleted.

EXAMPLE 4.

Treatment of the declare and call statements in certain subroutines is handled as shown in the example which follows.

Usage

```
declare adjust_bit_count_ entry, (char(168) aligned,  
    char(32) aligned, bit(1) aligned, fixed bin(24),  
    fixed bin);  
  
call adjust_bit_count_ (dn, en, char_sw, bit_count,  
    code);
```

Example of Usage Lines

The following list shows some of the variants that may be encountered:

```
copy path1i path2i ... path1n -path2n- -control_args-
fortran_abs paths -ft_args- -control_args-
set_acl path mode1 User_id1...moden -User_idn- -control_args-
terminate_segno seg_nos
who -control_args- -optional_args-
progress -control_arg- -command_line-
exec_com path -optional_args-
get_quota -control_arg- paths
change_wdir -path-
delete_acl path User_ids -control_arg-
enter_abs_request path -control_args- -optional_args-
```

REFERENCES

There are two types of references: internal and external. Internal references are references to subject matter found elsewhere in the same document, e.g., a reference in Section II of the MPM Reference Guide to material to be found in the previous or following sections. External references are references to subject matter located in documents other than the one in hand, even though those documents may be part of the MPM document set. For example, a reference in the MPM Commands to a description of a subroutine in the MPM Subroutines is an external reference.

Each type of reference is handled differently. If the internal reference refers to a description within the same volume the reference would appear:

See the memo command description.

or:

See the archive and archive_sort command description (in this document and the MPM Subsystem Writers' Guide, Order No. AK92, respectively).

If the reference is to be a description contained in another document, the reference is stated as:

See the user_info_ subroutine description in Section II of MPM Subroutines, Order No. AG93.

However, for the sake of convenience, it is usual in a set of volumes such as the MPM, to reference all volumes by their full title and order number in the preface of each volume, and in text omit the underline and shorten the reference to:

See the user_info_ subroutine description in Section II of MPM Subroutines.

Other external references must be to subject headings, section, title, and order number in the case of Honeywell publications (e.g., see "Password Trap" in Section VIII of the System Administrator's Manual, Order No. AK50). In the very rare event that it becomes necessary to reference a non-Honeywell publication, please seek advice from a member of the Documentation group.

Dot Tilde

Each time a reference is made within a description, the dot tilde (.~) comment control line should be inserted so that a complete reference index is available. On 2741-type terminals, the dot tilde is represented by the characters .\t.

As the first line of any description, use the .sr Charsw -1 control. This automatically creates a char segment. In this way the reference can easily be checked to determine if it has been changed. As an example:

```
.~ Under "Name"--first paragraph, user_info_ subroutine  
.~ referenced to Section II of MPM Subroutines.
```

TERMINOLOGY STANDARDS

Currently, the usage line in the MPM descriptions offers a rich variety of terminology for the same terms. While author creativity should not be stifled, certain standard ways of referring to common terms is mandatory from a user's point of

view. Henceforth, the following "standard" terminology will be used:

<u>Current Variations</u>	<u>Standard</u>
control argument control_argument control_arg ctl_arg	control_arg
optional argument arg option	optional_arg
person name identifier user User_name Person person person identifier name personid	Person_id
project name identifier project project_name projectid project identifier project	Project_id
segment/directory pathname pname pathname path segname source_name	path
entryname entry_name entryname ename	entryname
access control identifier acname	User_id
segment number segno	seg_no
reference name refname name	ref_name

The following terms will be "standard" for text:

entryname
 pathname
 terminal (NOT console except for the operator)
 multisegment file
 record rather than page when discussing quota
 I/O switch instead of I/O stream
 CPU--all uppercase
 newline character; but new line of text
 command and subroutine description, not writeup
 -output_file path instead of -output_file f
 the FORTRAN language; but the Multics fortran command
 the user types (not enters) input on his/her terminal
 programs print (not type) output or responses

GENERAL RULES/CONVENTIONS

1. Identify the default; do NOT say "This is not the default."
2. Do not leave blank lines in the runoff file; use .sp instead.
3. Do not number "Notes" or "Examples". Use introductory sentence for clarity in separating a number of examples, particularly if they are lengthy.
4. All descriptions must include at least one example; generally, several should be shown.
5. Do not use quotes around names of command, subroutines, operators, or I/O switch names.
6. Be sure to inform the user of access requirements where necessary.
7. Include default condition information, where applicable, within the argument description, as well as in the first paragraph under "Notes".
8. For control_args (and others), let the reader know how many can be selected (e.g., "can be one of the following:"). This will also be shown in the usage line, i.e.,

only one permitted	control_arg
several, if desired	control_args
9. Use plural, as paths, control_args, instead of an expression such as xxx1 ... xxxn, unless the command requires argument pairs (e.g., copy, or set_bit_count).

10. Refer to suffixes as, "appends the suffix runoff to the segment name." Do not place the suffix name in quotes, and do not use the dot (period) before the name in such references. Examples are:

the list segment
the absout segment
the suffix runoff

11. Refer to "segments or multisegment files" in the storage system context; but use "file" in the I/O system context to mean segments or multisegment files.
12. Daemons and special identifications are generally not initial cap, e.g., I/O daemon, GCOS daemon; however, SysDaemon when referencing the Project_id. Also, refer to GCOS environment simulator unless referencing the publication, GCOS Environment Simulator, Order No. AN05.
13. If a command accepts arguments that are associated with another command (e.g., runoff arguments can be given in the rfa command), the form to be used for such arguments is <short name of command>_arg. For example, rf_arg and ear_arg in the runoff_abs command.