Date: 25 August 1976

From: Cec Erickson

Subject: Info Segments

This MTB describes the format and contents of system info segments. It supersedes the information given in Section XI of MTB-291.

STYLE

Info segments provide immediate aid to online users. Info segments are not intended to offer instruction in the use of Multics. Info segment users are frequently busy and impatient; they often are using terminals that print slower than they can read. Therefore, when writing an info segment, use the following guidelines:

1. Be brief, concise, and terse.
2. Minimize words; minimize white space.
3. Assume some knowledge and experience on the part of the reader.
4. Use active verbs in the present tense.
5. Provide only essential facts; reference the MPM for complex detail.
6. Remember that info segments do not replace, but rather summarize the MPM.

PHYSICAL APPEARANCE

Info segments must be readable on all terminals supported by Multics. Therefore:

1. Employ a maximum line length of 79 characters.
2. Avoid tabs and needless spacing -- they slow output. The only indent allowed is ".in 3"; e.g., see "Syntax for Command Info Segs" later in this MTB.
3. Avoid underlining. Underlined text is illegible on many terminals.

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

4. Avoid control characters; they may not be transparent when the syst is accessed via certain terminals. This means that the use of 006 in info segments is discontinued and is replaced by conventions described in this MTB.
5. Do not put the string "(END)" at the end.
6. Use MPM conventions regarding punctuation, capitalization, etc.

NAMING CONVENTIONS

The help command respects the star convention for segment names. Therefore, the following naming conventions should be followed:

1. XXX.info for command XXX or subroutine XXX; e.g., help.info, clock_.info. (If the command has a short name, add the name <short name>.info to the info seg.)
2. XXX.changes.info for changes to XXX; e.g., help.changes.info.
3. XXX.status.info for status of bug fixes to XXX; e.g., basic.status.info.
4. XXX.gi.info for text info segment on general information; e.g., master_directories.gi.info. (A second name should be XXX.info.)
5. MRn-n.gi.info for system release summary info segments; e.g., MR4-0.gi.info. (A second name should be MRn-n.info.)

SYNTAX OF INFO SEGMENTS

Rules for the command and subroutine description info segs are strict so that the user can search efficiently for particular items. Rules for all other info segs have fewer restrictions. The syntax for each type of info seg and some examples are given on the following pages.

MAKING AN INFO SEGMENT--MECHANICS

1. Make a runoff segment whose first two lines are:
   .ll 79
   .na

2. All section titles must end with a colon and the only capital letter in the title is the first letter of the first word (e.g., "Control arguments:" NOT "Control Arguments:").

3. An indent of three (.in 3) is used in the "Arguments" and "Control arguments" sections (with ".un" before each argument) so that the arguments themselves stand out.

4. In long lists of things, put in two blank lines (or ".sp 2") every 10 or 15 items (that way, user will be asked if he needs more help). Also, if a section has several paragraphs, put in two blank lines between some of the paragraphs.

5. Try not to use a colon anywhere else except at the end of a section title (see item 2 above).

6. Make a runout segment, using the following control arguments:
   -in 0
   -npgn
   (e.g., "rf foo -sm -in 0 -npgn")

7. Rename the runout segment according to proper info seg rules:

   a. For individual modules, the primary name is <long name>.info and the added name is <short name>.info (e.g., set_acl.info, sa.info).

   b. For "text" info segs on general information, the primary name is <description>.gi.info and the added name is <description>.info (e.g., acl_matching.gi.info, acl_matching.info).

   c. For "changes" info segs, make the second component "changes"; do NOT insert the string "_changes" (e.g., set_acl.changes.info). This same rule applies to "bugs", "status", whatever. We want to be able to find things through components.

   d. For system release summary info segs, the primary name is MRn-n.gi.info and the added name is MRn-n.info (e.g., MR4-0.gi.info, MR4-0.info).

8. SAVE your runoff segment until the info seg is installed (in case changes are required). Editing macros will be written to transform an installed info seg back to a runoff segment for future updates.

SYNTAX FOR COMMAND INFO SEGS

```
.ll 79
.na
<date><TWO SPACES><long name>, <short name>
.sp
Syntax: <short name> <arg1> <arg2> <-control_args>
.sp 2
Function: <short phrase, not sentence; usually begins with verb, e.g., prints,
creates>.
.sp 2
Arguments:
.in 3
.un
<arg1><FIVE SPACES><description>. (DEFAULT-- phrase telling default)
.un
<arg2><FIVE SPACES><description>.
            [NOTE:  If only one argument, still use this format.]
.sp 2
.un
Control arguments:
.un
-<name1><FIVE SPACES><description>.
.un
-<name2><FIVE SPACES><description>.
.un
-<name3><FIVE SPACES><description. . . . . . . . . . . . . . . . . . .
   continued line of description (indented THREE SPACES)>.
.sp 2
.in 0
Access required:
.br
<description--this section not needed for most commands>.
.sp 2
Notes:
.br
<First paragraph of notes, i.e., issues critical to proper use of command>.
.sp
<Second paragraph of notes>.
.sp
      .
      .
      .
            [NOTE:  If there are several paragraphs, put ".sp 2" between some
                    of the paragraphs; i.e., every 10 lines or so.]
.sp 2
Examples:
            [NOTE:  This section should only be used in rare cases, e.g.,
                    extremely difficult command, or one that is not in the
                    MPM.]
```

If an argument is optional, make sure it is enclosed in braces in the syntax line AND make sure the description of that argument explains what happens if the argument is omitted. Put this information at the end of the argument description, in parentheses, preceded by the string "DEFAULT-- "; e.g., (DEFAULT-- working directory).

Make sure the short name of the command is shown at the top of the info seg and also make sure the short name is used in the "Syntax" section.

Both "Syntax" and "Function" sections are mandatory. "Arguments" and/or "Control arguments" sections are required if the command takes arguments and/or control arguments. If default values are associated with any control argument, explain at the end of the control argument description; e.g., in the pl1 command, the last part of the -severityN description would be (DEFAULT-- N is 1).

08/16/76 copy, cp

Syntax: cp path1A path2A ... path1N {path2N} {-control_args}

Function: copies segments and multisegment files.

Arguments:

path1i	pathname of segment or multisegment file to be copied; star convention accepted.
path2i	pathname of copy to be created; equal convention accepted. (DEFAULT-- working directory with entryname of path1)

Control arguments:

-name, -nm	copies all names.
-acl	copies ACL.
-all, -a	copies all names and ACL.
-brief, -bf	suppresses warning messages.

Access required:
Read access required to path1. Status access required to the directory containing path1. Append access required to the directory containing path2. Modify access also required if the -nm, -acl, or -all control argument is use

Notes:
The control arguments apply to all paths given in command.

SYNTAX FOR SUBROUTINE INFO SEGS

```
.ll 79
.na
.tr !
<date><TWO SPACES><subroutine name>
.sp
.in 3
.un
<call ...>
.sp
.un
<dcl ...fixed!bin(1)...>
.in 0
.sp 2
Notes: <special instructions, structure, etc.>
```

If there is a pointer to a structure, put the structure in the "Notes" section. If the structure is an include file, say so.

"Notes" section is optional. Not needed unless there is a structure or very special information that user must be told.

Call line(s) should contain the "standard" value of items if standard values exist. These are put in braces immediately following the item (e.g., ref_name{""}, seg_sw{0}, ...).

Declare line(s) should use ! as a translate character between things like "fixed" and "bin" so they will not end up on different lines.

Notice that with the proposed format, there is only one heading (Notes:) and the user gets the call and declare lines when he types:

```
     help <subroutine name>
```

Those subroutines having many entry points (e.g., hcs_) will have one info seg for each entry point plus a separate info seg that merely lists the entry points (e.g., hcs_.info will contain a list of all hcs_ entry points).

08/03/76 hcs_$initiate

```
call hcs_$initiate (dir_name, entryname, ref_name{""}, seg_sw{0},
   copy_ctl_sw{1}, seg_ptr, code);

dcl hcs_$initiate entry (char(*), char(*), char(*), fixed bin(1), fixed bin(2),
   ptr, fixed bin(35));
```

SYNTAX FOR CHANGES INFO SEGS

```
.ll 79
.na
<date><TWO SPACES><module_name changes>
.sp
<date><TWO SPACES><changes synopsis in 68 characters or less>:
.sp
<short summary paragraph highlighting changes>
               .
               .
               .
               .
.sp 2
<date><TWO SPACES><changes synopsis in 68 characters or less>:
.sp
<short summary paragraph highlighting changes>
               .
               .
               .
               .
```

This format will be used for both types of changes info segs:
--logs (usually lots of information arranged chronologically)
--post release (changes since last release that are not yet in published documents)

In logs, the most recent information is at the top. Notice that each date-synopsis line is a section title; i.e., it is preceded by two blank lines and ends with a colon. This format means the user can get useful information by having help print out all the titles in the info seg.

In all changes info segs, other titles may be used after the summary paragraph. However, a summary paragraph must be given first.

SYNTAX FOR GENERAL INFORMATION INFO SEGS

```
.ll 79
.na
<date><TWO SPACES><name>
.sp
     [rest of segment consists of short paragraphs describing the named topic;
     make SURE there are references to more information (generally some manual,
     perhaps some other info segs)]
```

There are no required titles. Keep in mind that this IS an info seg and like other info segs should be brief and to the point. The only requirement is at least one reference to detailed information on the topic.