

To: Distribution  
From: T. H. Van Vleck  
Date: April 11, 1977  
Subject: System Error Message Documentation

A method has been devised for documenting error messages produced by the system. Comments are inserted into the source of all programs which produce system messages; these comments are sections of the error message documentation. An automatic procedure collects and sorts these messages, and produces a runoff file which lists all messages.

This operation will be performed as part of the installation of new systems, at MIT and Phoenix. The message documentation will be distributed to customer sites in runoff form in >doc>MRxx on the release tapes.

Message documentation is required for hardcore programs which call `syserr` or `protection_audit_`; for answering service programs which call `sys_log_`; and for programs run in daemon processes or in the answering service which produce any messages.

All programmers and auditors are responsible for insuring that correct comments exist in every module which gets changed.

### Information Format

All error message documentation is contained in a single comment somewhere in the body of the program; the very bottom of the source is acceptable.

Each comment must bracket the error message documentation with a pair of lines of the form

```
BEGIN MESSAGE DOCUMENTATION
```

```
:
```

```
END MESSAGE DOCUMENTATION
```

The newline character on the BEGIN line is required. Within the bracketing lines, each message has a block of documentation.

---

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

Each message documentation block begins with the marker

Message:

on a line by itself. Characters may precede the M, but nothing may follow the colon. The characters preceding the M are taken to be a "prefix" which will be stripped off all other lines in the block. Thus, the double quote for ALM comments or the three blanks inserted by indent will be removed.

Following the start-of-block indicator, an example of the message is provided. Lower case or mixed case should be used for literal text, and items in all capitals for substitutable items. For those messages which are produced by `syserr$error_code`, the string `ERROR_CODE` should be used for the message expanded from `error_table_`. For example, one might place the lines

Message:

```
asd: acle_count err on DIRNAME>ENAME.
```

In the comment at the end of `asd.pl1`.

Following the error message template, an empty line must be supplied, followed by the documentation for the error message. Several headings are required, beginning paragraphs. A simple "macro" facility is provided to expand commonly-used sentences. The headings are:

**S:** Where the output is written and whether it causes a crash. Possible values for this field are:

```
$crash
$beep
$term
$info
$log
source (stream)
BOS typewriter
```

**T:** When the message can occur. Some possible values are:

```
$init
$run
$shut
Daemon driver startup.
```

**M:** An explanation of the message and any substitutable parameters. This section should describe the state of events which caused the message, and the probable reasons for that state. Probable consequences of the problem must also be described. For example, if no users can log in after a certain error, the documentation should say so. Sometimes it is

appropriate to break the discussion under this heading into several paragraphs, of increasing technical sophistication.

A: What action the operator should take if he sees the message. Some possible values are:

\$ignore  
\$recover  
\$contact  
\$contact\_sa

Paragraphs should be separated by a single blank line.

Messages which are logged only (syserr code 4, answering service code 0) should still be documented, since the message documentation will be used by site personnel to decipher the various logs.

### Example

The documentation for asd\_.pli would look like this:

/\* BEGIN MESSAGE DOCUMENTATION

Message:

asd\_! acle\_count err on DIRNAME>ENAME.

S: \$info

T: \$run

M: The count of ACL entries in entry ENAME of directory DIRNAME does not match the number of ACL entries found on the chain. Directory DIRNAME may have been damaged in a crash.

A: \$contact

Message:

asd\_! mylock err on DIRNAME

S: \$crash

T: \$run

M: Directory DIRNAME was found locked when attempting to lock it. This indicates a logic error in the supervisor, or CPU or memory hardware problems. This error crashes the system.

A: \$recover

## END MESSAGE DOCUMENTATION \*/

Processing

The command

```
extract_message_doc path1 epath {control_arg}
```

will extract the error messages from segment path1 and place them in the error message segment epath.msgs. If path1 is an archive, each of its components will be processed separately.

For each message, one or more lines are added to the output file. First, the name of the source segment is output, enclosed in brackets. Then, the text of the message (which may be multiple lines) is output, followed by a title and a newline. The message documentation follows, ending with an exclamation point and newline.

The output file resulting from the extraction of all messages can be processed with standard commands. To produce a runoff file, the following operations might be used:

```
&attach
qx
r error.msgs
1,$s/^[.]*] /~/
w temp
q
ss temp -dm !
qx
r title.page
r temp
1,$s/!$//
1,$s/^-.*$/ .if headline "&"\c
~&/
1,$s/if headline "~"/if headline "/
1,$s/^-"/"
1,$s/^-//p
1,$s/^-$/\c
.fl\c
.sp 2\c
.in 15/

1,$s/^S://Stream://
1,$s/^T://Time://
1,$s/^M://Meaning://
1,$s/^A://Action://
1,$s/^Stream://.un 10\c
Stream://
1,$s/^Time://.un 10\c
```

```
Time:/
1,$s/~Meaning:/ .un 10\c
Meaning:/
1,$s/~Action:/ .un 10\c
Action:/
```

```
1,$s/$crash$/BOS Typewriter (Crashes system)/
1,$s/$info$/BOS Typewriter./
1,$s/$beep$/BOS Typewriter (Sounds beeper)/
1,$s/$term$/BOS Typewriter (Terminates user process)/
1,$s/$log$/Logged in SYSERR log./
```

```
1,$s/$response$/In response to an operator command./
1,$s/$inlt$/System initialization./
1,$s/$run$/While system is running./
1,$s/$shut$/System shutdown./
```

```
1,$s/$err$/This indicates a logic error in the supervisor, or
CPU or memory hardware problems./
```

```
1,$s/$crashes$/This error crashes the system./
```

```
1,$s/$recovers$/Follow normal recovery procedures./
1,$s/$ignore$/No operator action is required./
1,$s/$contact$/Contact the system programming staff./
1,$s/$contact_sa$/Contact the system administrator./
1,$s/$tryagain$/Enter a corrected command line./
```

```
w error_msgs.runoff
q
dl temp
canon error_msgs.runoff
```

The include file "headline" takes care of ejecting a page when the program name changes. It also sets a variable which is used in a page footer to provide an easy reference index.

```
.* headline.runoff - macro for error message doc
.in 0
.sp 3
.ur .sr x "%Parameter%"(1,1)
.ur .ex ZOJPATH>runoff_fcn %FILENAME% .sr xcolon [index
"%Parameter%" "1"]
.ur .if %FILENAME%.fcn
.ur .ts (%xcolon>0)
.ur .sr x "%Parameter%"(1,%xcolon%-1)
.ur .ts ("%x%"!="%footletter%")
.bp
.ne 15
.ur .sr footletter "%x%"
```