To: Distribution

From: EJ Wallman

Date: July 8, 1974

Subject: Description and Specification for an On-Line Bug File Editor.

This document discusses the need for and implementation of a program which will maintain, on-line, a unified and consistent history of bugs in the Multics Software from the time they are discovered until they are declared "archaic".

BACKGROUND

The need for a unified and consistent bug history arose at PCO where Software Quality Assurance (SQA) requires that a history of all bugs found (whether fixed or not) during any of the testing phases (Exposure Test, System Test, and QA Test) be available on a daily basis in order that the "quality" of the product may be determined. The definition of the ephemeral term "quality" in this context consists primarily of the subjective evaluation of the answers to the two questions ...

1) Does the software, as implemented, properly perform those functions which are described in the documentation? and, if not, what are the functional restrictions and/or limitations?

2) What are the frequency and severity of functional aborts when operating within the restrictions of 1).

Based upon such an evaluation, a software release is classed "acceptable" or "unacceptable" by SQA and a "ship" or "hold" recommendation, respectively, is given to the Multics Program Office. There is no hard rule by which the reader may form his private subjective criteria for such an evaluation, but the following guide (one among several) has been applied ...

> "System fatal" bugs which cause more than 2-3 system crashes per service day must be fixed, but "system fatal" bugs which cause no more than 1 system crash per sevice week may be classed as "known errors".

A unified, on-line bug file will satisfy this requirement with no more effort on the part of Multics Development Engineering other than the proper maintenance of the file.

Moreover, the establishment of a unified and consistent bug file will serve the following purposes ...

1) It will provide a vehicle for customer release of the "known error" list.

2) It will provide the symptoms of fixed bugs in the event that the same bugs reappear under different guises.

3) It will provide the database for the first level screening of error reports to ascertain whether an error has already been reported (and, if so, the current status of the bug).

4) It will provide a mechanism for Multics Project Management to assign responsibility and monitor progress for the investigation and resolution of an error report.

5) It will provide a single source of correlative data among Error Reports (ISNs and CK97s), Multics Change Requests (MCRs), and Multics Installation Requests (MIRs).

6) It will provide the ability to gauge the "health" of any particular category or system function by data retrieval based upon a selection key. (See SPECIFICATION below for definition of "category".)

7) It will provide the vehicle by which Avoidance Techniques (Remedial Action) may be indicated to customers (and users).

SPECIFICATION

The program intended to maintain such a unified and consistent bug file shall ...

1) Execute as a special purpose editor (as opposed to an ASCII string editor) and shall provide full capability to add, delete, modify, and display any bug file data base entry and/or any field defined therein.

2) Operate upon a fixed but extensible set of bug file segments each of which shall define a "category" of bugs. This set shall include at least BOS, Salvager, Hardcore, Executable Libraries, Language Translators, and GCOS Environment.

3) Provide for mutual, concurrent perusal and/or update of a "category" segment by use of a suitable locking mechanism.

4) Provide for terminal type-out as well as accumulation for dprinting.

4) Provide for the unique identification of bugs within a "category".

6) Assist the inexperienced user with automatic prompting.

7) Operate as much as possible in a "command line" mode.

8) Provide for automatic update of a summary "info seg" every time a bug file entry is changed.

9) Permit selection of bug entries based on the content of any defined data field.

PROPOSED IMPLEMENTATION

The following is, the contents of the "help" file for the version of edit_bugfile currently residing in >auth_maint on System M. Reviewers are encouraged to experiment with the program by using the undocumented bug file >doc>bugs>tst. No real data is kept in this test bug file.

This procedure maintains the Multics bug files in a consistent manner. It operates as a special purpose editor with both prompting and brief modes.

The bug files are kept in the directory >doc>bugs and any number of them may be maintained. At the present writing, the following files are defined ...

hrd	hardcore operating system
pl1	PL1 Compiler
ftn	FORTRAN Compiler
bsc	BASIC Compiler
gcs	GCOS Environment and Daemon
cbl	COBOL Compiler
lib	System Libraries (>sss, >lang, >tools)
slv	Salvager
bos	BOS
ans	Answering Service
apl	APL Compiler

Bugs are identified by sequence number and development site suffix. At the present writing the following suffixes are recognized ...

p	Phoenix
c	Cambridge
b	Billerica

USAGE

The command is ...

```
        edit_bugfile <bugfile_name>
        eb <bugfile_name>
```

edit_bugfile will ask for input with the query "eb:" and will accept the following action codes. If data fields for a code are given following the code (as for a command line), the input will be treated as if it were a command line. If required data fields are missing, edit_bugfile will prompt the user for the required fields. A single "." on a line will signal the exit from any of the various input levels and a return to the prior level.

The action codes and their functions are ...

<u>code</u>	<u>description</u>
help,h	lists valid action codes.
help,h <code>	lists inputs for <code>.
bug,b	Enter a new bug into <bugfile_name> Required data are: <bug sequence number||site suffix> (1 to 4 digits||single char) <date reported> (YYMMDD) <Multics version> (20-2, 1.0.3) <bug symptoms> (<= 250 characters) optional inputs are: -st <bug status> (single character A = active,K = known,F = fixed X = fixed in exl compiler) -lg <ESN/ISN number> (C72814, etc.) -mcr <MCR number(s)> (n1,...n5) -mir <MIR number(s)> (n1,...n5) -fx <Multics version with fix> -fd <Date of fix> (YYMMDD) -rsp <responsible Person.Project> -k1,k2,k3 <keywords> (<= 32 chars) -D <bug description> (<= 250 chars) -R <remedial action> (<= 250 chars)

```
find,f                  Find and type-out bug(s) on key.

                        Valid keys are:
                        -a all
                        -b <bug sequence number||site suffix>
                        -s <site suffix>
                        -dr <date reported>
                        -dr> <date> (reported since)
                        -dr< <date> (reported before)
                        -v <Multics version where found>
                        -st <status>
                        -lg <ESN/ISN number>
                        -mcr <MCR number>
                        -mir <MIR number>
                        -rsp <responsible Person.Project>
                        -key <keyword>
                        -S <s> (string "s" within "S")
                        -D <s> (string "s" within "D")
                        -R <s> (string "s" within "R")
                        -fx <Multics version with fix>
                        -fd <date fixed>
                        -fd> <date> (fixed since)
                        -fd< <date> (fixed before)

modify,m                Modify bug entry data

                        Required input:
                        <bug sequence number||site suffix>
                        and a field selector from:

                        -b <bug sequence number||site>
                        -s <site suffix>
                        -dr <date reported>
                        -v <Multics version>
                        -st <status>
                        -lg <ESN/ISN number>
                        -mcr <MCR number>
                        -mir <MIR number>
                        -fx <Multics fix version>
                        -fd <date fixed>
                        -rsp <responsible Person.Project>
                        -k1,-k2,-k3 <keywords>
                        -S <bug symptoms>
                        -D <bug description>
                        -R <remedial action>

                           (A builtin text editor for -S, -D,
                            -R, -kn, and -rsp accepts ...
                            -r      replace
                            -d      delete
```

.s1.s2. string replacement where "." is any character not in s1 and s2.)

delete,d	Delete bug from file. Required: <bug sequence number||site>
list,l	Accumulate output for printing. Required; A search key from the list under "find".
count,c	Find and type count of items found. Required; A search key from the list under "find".
print,p	dprint the accumulated output. Optional input: Any combination of ... -cp <number of copies> (<= 4) -he <dprint heading> -ds <dprint destination>
switch,s	Switch to new <bugfile_name>. required: <new bugfile_name>
quit,q	Return to Multics command level. No arguments.