

MCR10100
Fix Typos in Comments, Code, Documentation.
Add Errata for Two Manuals.
Revision 1.1

Eric Swenson

October 31, 2021

1 Problem Description

This MCR addresses many tickets:

- <https://multics-trac.swenson.org/ticket/15>
- <https://multics-trac.swenson.org/ticket/103>
- <https://multics-trac.swenson.org/ticket/161>
- <https://multics-trac.swenson.org/ticket/162>
- <https://multics-trac.swenson.org/ticket/179>
- <https://multics-trac.swenson.org/ticket/186>
- <https://multics-trac.swenson.org/ticket/201>
- <https://multics-trac.swenson.org/ticket/225>
- <https://multics-trac.swenson.org/ticket/231>
- <https://multics-trac.swenson.org/ticket/239>
- <https://multics-trac.swenson.org/ticket/245>
- <https://multics-trac.swenson.org/ticket/250>

To access the tickets, provide the username "anonymous" and the password "multics" if prompted to authenticate.

Most of these tickets are very minor and relate to fixing minor typos in comments, documentation, and in some cases, code. But because creating MCRs for each of these would result in tiny MCRs and a lot of work reviewing, approving, auditing, and installing these changes, they are grouped into this one MCR. If the MCR board prefers splitting these up, it can reject this MCR and request individual MCRs.

Each of these tickets will be discussed independently, in separate sections, however, tickets 103 and 239 both relate to typos in comments in `bootload_tape_label.alm` and will be treated as one in this MCR.

2 Ticket 15: Fix typo in `fim.alm`

2.1 Problem Description

A comment in the code reads “get the fault dataister”. “dataister” is not a word. The intended text should be “get the fault data word into a register”.

2.2 Proposed Fix

Change the comment to read “get the fault data word into a register”. The `compare_ascii` output is below:

```
cpa [lpn fim.alm] ==

A811          lda      bp|mc.scu.fault_data_word get the fault
      dataister
Changed by B to:
B811          lda      bp|mc.scu.fault_data_word get the fault data
      word into a register

Comparison finished: 1 difference, 2 lines.
r 19:44 0.832 183
```

3 Tickets 103 and 239: Fix errors in comments in `bootload_tape_label.alm`

3.1 Problem Description

The comment documenting the IOM mailbox contents at offset 10 reads: “18/004000”. It should read “18/040000”. Also the address of the IOM mailbox should be “Base*64” not “Base*6”.

3.2 Proposed Fix

Correct the two errors, above. The `compare_ascii` output provides the details of the change:

```
cpa [lpn bootload_tape_label.alm] ==

A101      " IOM Mailbox, at Base*6
A102      "   mbx + 07   12/Base, 6/02, 18/000002          Fault channel
      DCW
A103      "   mbx + 10   18/0, 18/004000                  Connect channel
      LPW -> PCW
Changed by B to:
B101      " IOM Mailbox, at Base*64
B102      "   mbx + 07   12/Base, 6/02, 18/000002          Fault channel
      DCW
```

```
B103      "    mbx + 10    18/0, 18/040000      Connect channel
          LPW -> PCW
```

```
Comparison finished: 1 difference, 6 lines.
r 19:46 1.319 36
```

4 Ticket 161: No info page for fim_meters

4.1 Problem Description

There is no info segment for the `fim_meters` program.

4.2 Proposed Fix

Add an info segment for this tool. The info segment contents follows:

```
:Info: fim_meters: 2021-10-28  fim_meters
```

```
Syntax as a command: fim_meters {FAULT_NAME} {-control_args}
```

Function: Prints the per-system count of faults by fault type. This data is accumulated in `wired_hardcore_data` by ring-0 fault processing (`fim`, `wired_fim`, and `page_fault`).

Arguments:

FAULT_NAME

is either a long or short name for a fault. See "List of fault names" for a list of acceptable values. If this argument is not provided, `fim_meters` reports on all faults.

Control arguments:

-report

Causes a meter report to be generated and printed to the console. This is the default.

-reset, -rs

Reset the meters so the next report reflects meters from the time reset.

-report_reset, -rr

Generates a report and then resets the meters.

-long, -lg

Includes sub-faults as well as primary faults.

-total, -tt

Provides totals of primary faults only. This is the default.

-sort KIND

Sorts the output by the fault number (ordinal number of fault as

defined by the hardware) or by the number of times the fault has been signalled. KIND can be either "number" or "count". This control argument is incompatible with -cpu.

-cpu {CPU_TAGS}
where CPU_TAG gives the letter identifier for a CPU whose meters are to be displayed. If no CPU_TAGS are given, meters for all CPUs are displayed. This control argument is incompatible with -sort.

List of fault names:

The following fault names are supported. Note that some faults have long and short names. Others only have a single name. The short name, if any, follows the long name for the fault.

shutdown, sdf
store, str
mme1, mme
fault_tag_1, ft1
timer_runout, tro
command, cmd
derail, drl
lockup, luf
connect, con

parity, par
illegal_procedure, ipr
op_not_complete, onc
startup, suf
overflow, ovf
divide_check, div
execute, exf
segment_fault, df0
page_fault, df1
directed_fault_2, df2
directed_fault_3, df3
access_violation, acv
linkage_fault, ft2
fault_tag_3, ft3

trouble, trb
illegal_opcode
null_pointer
illegal_modifier
illegal_ring_order
not_in_execute_bracket
no_execute_permission
not_in_read_bracket
no_read_permission
not_in_write_bracket
no_write_permission

not_a_gate
not_in_call_bracket
outward_call

bad_outward_call
inward_return
cross_ring_transfer
ring_alarm_fault
am_fault
out_of_bounds
fixedoverflow
overflow
underflow
stringsize
illegal_procedure
stack_out_of_bounds
packed_pointer_fault
lot_fault
isot_fault

system_packed_pointer
size
neti
sus_
trm_
wkp_
undefined_pointer

Examples:

Typical output from this command with no arguments follows:

Total Metering Time: 362:02:19

Fault Type	Total Fault Count
timer_runout	20973050
connect	13305077
illegal_procedure	9
overflow	1
segment_fault	127201
page_fault	260845
access_violation	187563
linkage_fault	98447

5 Ticket 162: Fix `change_tuning_parameters` Documentation in AN52

5.1 Problem Description

The documentation in AN52 is out-of-date (incomplete) with respect to the `change_tuning_parameters` list of tuning parameters. It includes references to the obsolete (and removed) `max_abs_eligible` and `double_writes` parameters. It uses the short names only for `notify_timeout_interval` and `notify_timeout_severity`. It misses the short names for `write_limit`, and `process_initial_quantum`. And it leaves out the following tuning parameters, which are currently supported.

- `max_batch_eligible` (`maxabs`)
- `stack_truncation` (`stkt`)
- `stack_truncation_always` (`stta`)
- `stk_trunc_block_avg_factor` (`stk_baf`)
- `trap_invalid_masked`
- `meter_ast_locking`
- `realtime_io_priority` (`io_prior`)
- `realtime_io_deadline` (`io_deadline`)
- `realtime_io_quantum` (`io_quantum`)
- `checksum_filemap`
- `default_256K_enable`
- `dirlock_writebehind` (`dirw`)

5.2 Proposed Fix

The proposed fix is to add an errata info segment (as we have no source for the manuals). There are already many errata info segments, so this is just following the standard practices used in the old days.

The errata info segment will have these names (in keeping with conventions):

- `an52.errata.info`
- `errata.an52.info`
- `System_Metering.info`

The proposed contents of the info segment are:

:Info: `an52.errata`: `errata.an52`: `System_Metering`: 2021-10-29 AN52 Errata

Function: This info segment provides errata and addenda information for the Multics manual, AN52-01B titled:

Multics System Metering

change_tuning_parameters:

pg 4-4, under "Notes":

DELETE entry for max_abs_eligible (maxabs)

pg 4-5:

DELETE entry for double_writes (dblw)

CHANGE entry for nto_delta to:

notify_timeout_interval (nto_delta)
number of seconds
decimal number of seconds
30.0 sec.

CHANGE entry for nto_severity to:

notify_timeout_severity (nto_severity)
number of the severity
integer
3

CHANGE entry for write_limit to:

write_limit (wlim)
number of pages
integer
1/8 of pageable memory

CHANGE entry for process_initial_quantum to:

process_initial_quantum (piq)
number of seconds
decimal number of seconds
2.0 sec.

ADD (after gv_integration entry) entries for:

max_batch_elig (maxabs)
number of processes
integer
0

stack_truncation (stkt)
whether to truncate stack on blocking on events if advantageous
boolean value (on or off)
on

`stack_truncation_always (stta)`
 whether to always truncate stack on blocking on events
 boolean value (on or off)
 off

`stk_trunc_block_avg_factor (stk_baf)`
 factor used in deciding whether to truncate stacks
 decimal number
 0.9375 (15/16)

`trap_invalid_masked`
 whether to crash system when invalid mask found when in
 user-ring
 boolean value (on or off)
 off

`meter_ast_locking`
 whether to meter locking the AST.
 boolean value (on or off)
 off

`realtime_io_priority (io_prior)`
 whether IOI interrupts make process high priority by scheduler
 boolean value (on or off)
 off

`realtime_io_deadline (io_deadline)`
 time interval after IOI interrupt process considered high
 priority by scheduler
 decimal number of seconds
 0. seconds

`realtime_io_quantum (io_quantum)`
 length of high priority quantum before normal priority
 scheduling
 decimal number of virtual CPU seconds
 0.005 seconds

`checksum_filemap`
 whether to checksum the file map to detect integrity issues
 boolean value (on or off)
 on

`default_256K_enable`
 whether 256K segments are allowed for all processes by default
 boolean value (on or off)
 off

`dirlock_writebehind (dirw)`

causes modified directory pages to be written out
boolean value (on or off)
off

pg 4-7.2, at end, add:

The max_batch_elig (maxabs) tuning parameter is the maximum number of absentee processes that can be eligible at one time. A value of zero causes this restriction to be ignored. This parameter is also ignored if the schedule is in deadline mode.

The realtime_io_priority, realtime_io_deadling, and realtime_io_quantum tuning parameters jointly control the traffic control priority of a process which has received an interrupt from an IOI device (usually a tape drive). If realtime_io_priority is off, the parameters realtime_io_deadline and realtime_io_burst have no effect. The effect of tuning realtime_io_priority on is improved throughput for tape I/O-bound processes (such as the hierarchy and volume dumpers). This is accomplished by treating a process which has just received an interrupt for an IOI device as a realtime process for one eligibility quantum after receipt of the interrupt. The tuning parameters realtime_io_deadline and realtime_io_quantum control that eligibility time.

dirlock_writebehind (dirw) causes modified directory pages to be written from main memory whenever a directory is unlocked. Specifying this parameters increases safety of the system at some cost in increased paging. Process directory pages are not written.

See the Multics System Maintenance Procedures Manual, Order No. AM81 for more details on system tuning and tuning parameters.

:hcom:

/****^ HISTORY COMMENTS:

END HISTORY COMMENTS */

6 Ticket 179: Fix errors in Storage System PLM

6.1 Problem Description

The Storage System PLM (AN61A) documentation is incorrect in its statement of where the core map is located. It says that the core map is in the "SST" segment, when, in fact, it is in the core_map segment.

6.2 Proposed Fix

Add an errata info segment for AN61, correcting the text.

The errata info segment will have these names (in keeping with conventions):

- `an61.errata.info`
- `errata.an61.info`
- `Storage_System_PLM.info`

The following is proposed contents of the new errata info segment.

```
:Info: an61.errata: errata.an61: Storage_System_PLM: 2021-10-29 AN61 Errata
```

```
Function: This info segment provides errata and addenda information for
the Multics manual, AN61A titled:
```

```
    Multics Storage System Program Logic Manual Addendum A
```

```
pg 5-7, paragraph 3, last sentence:
```

```
CHANGE:
```

```
    The core map is in the "SST" segment.
```

```
T0:
```

```
    The core map is in the "core_map" segment.
```

```
:hcom:
```

```
/*^^^ HISTORY COMMENTS:
```

```
END HISTORY COMMENTS */
```

7 Ticket 186: Fix incorrect declaration in `fs_util.info` for `set_bit_count`

7.1 Problem Description

In `fs_util.info`, the documentation on the `set_bit_count` entry has an incorrect declaration:

```
declare fs_util_$set_bit_count entry (char (*), char (*), fixed bin (41),
    fixed bin (35));
```

The fixed bin (41) should be fixed bin (24).

The actual declaration in `fs_util.pl1` is correct so this change makes the info segment match the implementation.

7.2 Proposed Fix

The trivial fix is shown in the `compare_ascii` output, below:

```
A718      :Entry:  set_bit_count:  1985-02-27  fs_util_$set_bit_count
Changed by B to:
B718      :Entry:  set_bit_count:  2021-10-28   fs_util_$set_bit_count

A726      declare fs_util_$set_bit_count entry (char (*), char (*), fixed
        bin
A727          (41), fixed bin (35));
Changed by B to:
B726      declare fs_util_$set_bit_count entry (char (*), char (*),
B727          fixed bin(24), fixed bin (35));

Comparison finished: 2 differences, 6 lines.
```

8 Ticket 201: Fix message documentation in `multiplexer_mgr.pl1`

8.1 Problem Description

Message documentation is a facility where various hardcore, ring-1, and answering service source files document messages that can appear on the system console or in the system logs, such as `syserr_log` and `>sc1>as_logs>log`. It is demarcated in source files by:

```
/* BEGIN MESSAGE DOCUMENTATION
```

and

```
END MESSAGE DOCUMENTATION */
```

Various tools know how to extract the message documentation from these source segments.

The message documentation for `multiplexer_mgr.pl1` is broken and unparseable by those tools because the ending marker is misspelled:

```
END MESSSAGE DOCUMENTATION */
```

8.2 Proposed Fix

The trivial fix is shown in the `compare_ascii` output below:

```
A2644      END MESSSAGE DOCUMENTATION */
Changed by B to:
```

B2644 END MESSAGE DOCUMENTATION */

Comparison finished: 1 difference, 2 lines.
r 08:18 0.915 86

9 Ticket 225: Fix typo in find_rpv_subsystem.pl1 query

9.1 Problem Description

The `find_rpv_subsystem` program is invoked during early initialization to locate the RPV among the available disk subsystems and to load firmware into the disk subsystem MPC, if needed. It is used both to boot an existing system, or during cold boot of a new system.

The prompt this program issues looks something like the one below:

```
bootload_0: Booting system MR12.8b generated 10/29/21 2213.5 pdt Fri.  
0131.1 announce_chwm: 428. pages used of 512. in wired environment.  
0131.1 announce_chwm: 706. words used of 1024. in int_unpaged_page_tables.  
find_rpv_subsystem: Enter RPV data: M->
```

The normal reply for booting an already installed system is something like this:

```
rpv a11 ipc 3381 0a
```

The “rpv” indicates that the expectation that the RPV has already been initialized (as opposed to “cold”, which is used to cold boot). The “a11” specifies that the RPV’s MPC is located on IOM A, channel 11 (decimal). The “ipc” says that the MPC type is “ipc”. The “3381” says that the disk type is a model 3381, and finally, the “0a” says that the RPC is located on device 0, subsystem “a”.

However, because this syntax is a bit obscure, an acceptable answer to the above prompt is “help”. This causes the following output to be emitted:

```
For an ordinary boot, enter:  
  rpv lcc MPC_Model Drive_Model Drive_Number  
Where: l is the iom tag, cc is the base channel in decimal,  
       MPC_Model is MPC Model,  
       Drive_Model is RPV drive model number, and  
       Drive_Number is the RPV device number.  
  
For a cold boot, enter:  
  cold lcc MPC_Model Drive_Model Drive_Number  
  
To skip the firmware loading of the MPC, enter  
  skip_load or skip  
  
If this still is not clear, enter "query" for a setp-by-step query  
find_rpv_subsystem: Enter RPV data: M->
```

The help message, in its last line before reprompting, misspells “step-by-step”.

9.2 Proposed Fix

The trivial fix is shown in the `compare_ascii` output, below:

```
A311          call ioa_ ("~/If this still is not clear, enter
  ""query"" for a setp-by-step query");
Changed by B to:
B311          call ioa_ ("~/If this still is not clear, enter
  ""query"" for a step-by-step query");

Comparison finished: 1 difference, 2 lines.
r 08:28 1.104 62
```

Ticket 231: Add more info on RPD, RPL, and RPT instructions in AL39

10.1 Problem Description

The RPD, RPL, and RPT instructions are documented in the “Multics Processor Manual”, AL39. These instructions are quite involved and the documentation leaves out some details that are found in the GCOS documentation, “DPS 8 & DPS 88 Assembly Instructions”, DH53-01.

Rather than attempt to summarize all the missing information, this MCR simply provides the additional text by copying it from the DH53-01 manual to an updated AL39 errata info segment. This info segment already exists, so new errata instructions will be added to the top of the info segment.

See the proposed fix section for the details.

10.2 Proposed Fix

The proposed fix is to make the changes shown in the `compare_ascii` output, below:

```
A1          10/31/92  AL39 (Multics Processor Manual)
A2          Errata Information for MR12.5
Changed by B to:
B1          2021-10-29  AL39 (Multics Processor Manual)
B2          Errata Information for MR12.8
```

Inserted in B:

```
B11          pg 4-147: at end:
B12          ADD:
B13
B14          Coding Format:
B15
B16          RPD N, I, k1, k2, ... , k7. (A=B=C=1.)
B17
```

B18 The command generated by the assembler from this format will
 cause the
 B19 two instructions immediately following the RPD instruction to be
 B20 iterated times and the effective addresses of those two
 instructions to
 B21 be incremented by the value I for each of N iterations.
 B22
 B23 The meaning of k1, k2, ..., k7 is the same as for the RPT
 instruction.
 B24 (See item below). Since the repeat double must fall in an odd
 location,
 B25 precede this instruction with the odd pseudo-op.
 B26
 B27
 B28 RPDX N, I (A=B=C=0.)
 B29
 B30 This instruction operates just as the RPD instruction with the
 B31 exception that A, B, N and the conditions for termination are
 loaded
 B32 by the user into index register X0 as an octal value.
 B33
 B34 RPDA N, I, k1, k2, ..., k7. (A=C=1. B=0.)
 B35
 B36 This instruction operates just as the RPD instruction with the
 B37 exception that only the effective address of the first instruction
 B38 following the RPDA instruction will be incremented by the value
 of I
 B39 for each of N iterations.
 B40
 B41
 B42 RPDB N, I, k1, k2, ..., k7. (A=0. B=C=1.)
 B43
 B44 This instruction operates just as the RPD instruction with the
 B45 exception that only the effective address of the second
 instruction
 B46 following the RPDB instruction will be incremented by the value I
 B47 for each of N iterations.
 B48
 B49
 B50 The following discussion of RPT also applies to RPD and RPL
 B51 instructions. It explains how the k1, k2, ..., k7 bits of the Term
 B52 Cond. field can be specified in those instructions.
 B53
 B54 RPT N, I, k1, k2, ..., k7. (Bit C=1.)
 B55
 B56 The command generated by the assembler from this format will
 cause the
 B57 instruction immediately following the RPT instruction to be
 iterated N
 B58 times and that instruction's effective address to be incremented
 by the
 B59 value I for each of N iterations. The range for N is 0-255. If N

= 0,
 B60 the instruction will be iterated 256 times. If N is greater than
 256,
 B61 the instruction will cause an error flag (A) to be printed on the
 B62 assembly listing.
 B63
 B64
 B65 The fields k1, k2, ... k7 may or may not be present. They
 represent
 B66 conditions for termination which, when needed, are declared by the
 B67 conditional transfer instructions TMI, TNC, TNZ, TOV, TPL, TRC,
 and
 B68 TZE. These instructions affect the termination condition bits in
 B69 positions 11-17 of the Repeat instruction.
 B70
 B71 It is also possible to use an octal number rather than the
 transfer
 B72 instructions to denote termination conditions. Thus, if the field
 for
 B73 k1, k2,..., k7 is found to be numeric, it will be interpreted as
 octal
 B74 and the low-order 7 bits will be ORed into bit positions 11-17 of
 the
 B75 Repeat instruction. The variable-field scan will be terminated
 with the
 B76 octal field.
 B77
 B78
 B79 pg 4-150: at end of RPL section, before RPT:
 B80
 B81 ADD:
 B82
 B83 Coding Format:
 B84
 B85 RPL N, k1, k2, ... k7. (C = 1.)
 B86
 B87 This format causes the instruction immediately following the RPL
 B88 instruction to be repeated N times or until one of the conditions
 B89 specified in k1, ..., k7 is satisfied, or until the link address
 of
 B90 zero is detected. The range of N is 0-255. If N = 0, the
 instruction
 B91 will be iterated 256 times. If N is greater than 255, the
 instruction
 B92 will cause an error flags (A) to be printed on the assembly
 listing.
 B93
 B94
 B95 The fields k1, k2, ..., k7 may or may not be present. They
 represent
 B96 conditions for termination which, when needed, are declared by the
 B97 conditional transfer instructions TMI, TNC, TNZ, TOV, TPL, TRC,

and

B98 TZE. These instructions affect the termination condition bits in
 B99 position 11-17 of the Repeat instruction.

B100

B101 It is also possible to use an octal number rather than the
 transfer

B102 instructions to denote termination conditions. Thus, if the field
 for

B103 k1, k2, ..., k7 is found to be numeric, it will be interpreted as
 octal

B104 and the low-order 7 bits will be ORed into bit positions 11-17 of
 the

B105 Repeat instruction. The variable-field scan will be terminated
 with the

B106 octal field.

B107

B108

B109 RPLX (C = 0).

B110

B111 This instruction operations just as the RPL instruction except
 that N

B112 and the conditions for termination are loaded by the user into
 index

B113 register zero.

B114

B115

B116 pg 4-152: at end of RPT section:

B117

B118 ADD:

B119

B120 Coding Format:

B121

B122 RPT N, I, k1, k2, ..., k7. (Bit C=1.)

B123

B124 The command generated by the assembler from this format will
 cause the

B125 instruction immediately following the RPT instruction to be
 iterated N

B126 times and that instruction's effective address to be incremented
 by the

B127 value I for each of N iterations. The range for N is 0-255. If N
 = 0,

B128 the instruction will be iterated 256 times. If N is greater than
 256,

B129 the instruction will cause an error flag (A) to be printed on the
 B130 assembly listing.

B131

B132

B133 The fields k1, k2, ... k7 may or may not be present. They
 represent

B134 conditions for termination which, when needed, are declared by the

B135 conditional transfer instructions TMI, TNC, TNZ, TOV, TPL, TRC,

```

        and
B136      TZE. These instructions affect the termination condition bits in
B137      positions 11-17 of the Repeat instruction.
B138
B139      It is also possible to use an octal number rather than the
        transfer
B140      instructions to denote termination conditions. Thus, if the field
        for
B141      k1, k2,..., k7 is found to be numeric, it will be interpreted as
        octal
B142      and the low-order 7 bits will be ORed into bit positions 11-17 of
        the
B143      Repeat instruction. The variable-field scan will be terminated
        with the
B144      octal field.
B145
B146
B147      RPTX N, I (Bit C=0).
B148
B149      This instruction operations just as the RPT instruction with the
B150      exception that N and the conditions for termination are loaded by
        the
B151      user into bit positions 0-7 and 11-17, respectively, of index
        register
B152      zero (instead of being embedded in the instruction).
B153
B154
Preceding:
A11      pg 8-8:  for Figure 8-1, correct the typos in the decision block.
        |

```

```

Inserted in B:
B168
At end

```

```

Comparison finished: 3 differences, 149 lines.
r 08:34 2.266 29

```

11 Ticket 245: Fix incorrect grammar in prompt in `bce_get_to_command_line`

11.1 Problem Description

During a cold boot, the Bootload Command Environment (BCE) prompts the user with the following question:

```
Do you really wish to boot cold and there by destroy the system hierarchy?
```

The two words, “there by” should be one word – “thereby”.

The message is displayed by the program `bce_get_to_command_level.pl1`.

11.2 Proposed Fix

The trivial proposed fix is shown in the `compare_ascii` output, below:

```
A174                                call bce_query$yes_no (yes_no, "Do you
    really wish to boot cold and there by destroy the system
\chierarchy? ");
Changed by B to:
B174                                call bce_query$yes_no (yes_no, "Do you
    really wish to boot cold and thereby destroy the system h
\cierarchy? ");

Comparison finished: 1 difference, 2 lines.
r 08:36 0.872 40
```

12 Ticket 250: Add correct MTB number to comment in `mcs_timer.pl1`

12.1 Problem Description

In the comment section at the top of the source segment `mcs_timer.pl1` there appears the following text:

```
/* *      MCS_TIMER -- Ring zero MCS timer manager
 *
 *      This procedure implements timers for ring zero MCS. When a timer
comes due,
 *      a TIMER interrupt is delivered to the lucky channel. For all the
frankly
 *      fascinating details, see MTB-XXX.
 *
 *      Written 24 March 1982, W. Olin Sibert, for the ASEA Hyperchannel
project.
 *      Modification history:
 *      24 Mar 82, WOS: Initial coding, for the ASEA Hyperchannel project.
 *      30 May 82, WOS: Added subchan_idx parameters, changed timer_id to
bit (36).
 *      October 1982, CAH: Redesigned to use an hproc.
*/
```

The referenced to “MTB-XXX” was most likely intended to be updated by the author, but was never acted upon. For readers of this source code, knowing the MTB referenced herein would help understand how this code was supposed to work and what problems it was trying to address.

The MTB in question was “MTB-580”, entitled “Timers in Ring Zero MCS”.

12.2 Proposed Fix

The proposed fix is to fill in the MTB number and provide its title. The `compare_ascii` output below shows the change:

```
A19          *      fascinating details , see MTB-xxx .
Changed by B to:
B19          *      fascinating details , see MTB-580 ("Timers in Ring Zero
                MCS") .

Comparison finished: 1 difference , 2 lines .
r 08:38 0.696 36
```

13 Testing

Testing of these changes will involve making sure that all updated info segments, including errata info segments read correctly and pass the checks from `verify_info` (a recent replacement for `verify_info_segment`).

Testing of the two code changes that have any impact on a running system will be tested by provoking the corrected prompts and making sure that they read correctly and perform correctly.

Testing of changes in code where the changes are only in comments will involve ensuring the programs compile correctly and continue to work correctly.

14 Documentation

There is no additional documentation required for these changes. Most of these changes are in documentation, which is provided in the fixes.

15 Version History

2021-10-31	1.0	Initial version of the MCR.
2021-10-31	1.1	Fix typos and other minor mistakes.