To:          Distribution

From:        James A. Bush

Date:        December 5, 1978

Subject:     New System Firmware Tape Support for MR7.0


        This MTB has been written to inform the Multics community of
pending changes in the format of the System Firmware Tape, and to
outline a plan for support of the new firmware tape for MR7.0.

        Comments on  the  material  provided  in  this  document  are
invited and may be mailed, via U. S. mail to:

                James A. Bush
                Honeywell Information Systems
                Mail Station B-84
                P. O. Box 6000
                Phoenix, Arizona   85005

or by Multics mail on System M to:

                Bush.MTD


        The reasons for  and the methodology  behind the decision to
go to a new  firmware tape are  detailed in  Appendix A attached,
which is a copy of a Technical Design Memo that describes the new
firmware tape format.

        Briefly,  the  new  firmware  tape  will  contain  not  only
Firmware, ITRs, and MDRs as  does the current tape, but also Polt
and Molt common slave mode  executives, and test pages. These new
slave mode  executives and test pages are  necessary to implement
the MR7.0 version of Polts and Molts, (which includes support for
the MSS500  disks and MTU610  tapes). The  MR7.0 version of Polts
and Molts will  consist of a  PL1 coded  Multics interface to the
common GCOS  slave object modules  known as Slave  Mode Polt and
Slave Mode  Molt. The Multics  interface  module will look like a
mini GCOS  simulator,  providing user  interface  and  service
functions to the slave mode execs.

        The new firmware  tape will also  contain the processor test
pages  needed for the  recently  completed Multics  Processor T&D
subsystem, ISOLTS.
------------------------------------------------------------------

A new utility program has been developed to load the contents of the new firmware tape onto the Multics storage system. A command description of this new utility, called load_tand_library, is attached to this MTB.

The load_tand_library command differs from it's predecessor, load_firmware_file, in several ways. These differences are listed below:

o The load_firmware_file command, loads only firmware, ITRs and MDRs.

   The load_tand_library command, will load Firmware, ITRs and MDRs as well as the previously mentioned T&D test pages.

o The load_firmware_file command, loads GCOS object modules from the firmware tape into individual segments in core loadable image. These individual segments are later combined into an archive segment (firmware.archive) using the archive command.

   The load_tand_library command, loads GCOS object modules from the firmware tape into a keyed sequential file (named tand_deck_file), in object deck format. Firmware object modules are also loaded into individual segments in core loadable image, for compatibility with BOS tape generation. This will allow MPC firmware to be written onto the BOS tape as is done today, with no changes necessary to the current procedures.

o The load_firmware_file command, reads the firmware tape using subroutines that are an integral part of the unbundled GCOS simulator software. This means that a costumer who otherwise does not wish to buy the GCOS simulator software, must do so in order to support loading firmware tapes.

   The load_tand_library command, uses the iox_ compatible tape io module, tape_nstd_ to read the firmware tape, removing the above restriction.

A keyed sequential file was chosen over an archive segment for two reasons:

1. archive segments cannot grow to multi-segment-files and with the addition of the new T&D test pages, all necessary objects from the firmware tape would not fit into an archive segment.

2. Finding the desired entry in an archive segment requires linear searching which, with many entries, is very time consuming. With a keyed sequential file, searching for the

desired key involves an indexed search using the vfile_ io module.

Loading the object modules in object deck format instead of core loadable format is being done for the following reasons:

1. Firmware, ITRs and MDRs have a load origin address of 0. That is, the first data word of the object deck is loaded into location 0 of the appropriate segment. However, most of the T&D test pages will have a loading origin address greater than 0. In fact most of the ISOLTS processor test pages load at location 0+32k. Without coming up with a load compression scheme, a lot of space in the storage system would be wasted if these decks were loaded in core image format. Instead, a general purpose GCOS card image loader was developed to load the core image of these decks when desired. This general loader, the gload_ subroutine, was installed in MR6.5.

2. Each object card has a checksum word associated with it. When an object deck is loaded by gload_, the checksum is regenerated and compared against the card checksum. This provides greater reliability and runtime detectability of decks which have been "munged" or made inconsistent by storage system errors.

The currently installed version of Molts uses a subroutine, firmware_util_, to search the firmware archive for Firmware, ITRs and MDRs. The functions performed by the firmware_util_ subroutine will be replaced by an internal tolts subroutine (tolts_util_$search) that will search the tand_deck_file for a given key and return a pointer to the appropriate record within the tand_deck_file. This new search facility has already been coded and checked out to support the ISOLTS subsystem.

In conclusion, I would like to make the following recommendations:

1. Since the firmware tape no longer contains only Firmware, ITRs and MDRs, I would like to propose that the name of the directory where these items are currently stored, be changed from >firmware to >system_library_tand. The name firmware should be retained as an addname for at least 1 release period.

2. As can be seen from the above discussion, the load_firmware_file command and the firmware_util_ subroutine will be obsoleted by the load_tand_library command and the internal tand_deck_file search facility. I would like to propose that the load_firmware_file command and the firmware_util_ subroutine be moved to the system_library_obsolete directory before MR7.0 is shipped.

Name: load_tand_library, ltdl

    The load_tand_library command loads MPC firmware, ITRs,
MDRs, and T&D test pages from the Firmware and T & D Library Tape
(formally the Firmware Tape) into a keyed sequential vfile_, with
the name of "tand_deck_file" (referred to as "the deckfile" in
the remainder of this document). In addition, MPC firmware
modules are loaded into individual segments in a form acceptable
to the generate_mst command, so they may be written onto the BOS
tape. This command also generates a printable ASCII segment that
contains a directory of those modules loaded from the tape.


Usage:

        load_tand_library tape_name {-control_args}


where:


1.    tape_name
            specifies the name for the Firmware and T & D Library
            Tape to be used

2.    control_args
            are optional control arguments that determine which
            modules are to be selected from the tape and where
            they are to be stored. The control arguments may be
            selected from the following list:

      -pathname path, -pn path
            specifies the name of the directory in which the
            deckfile and the firmware module segments are to be
            created. If omitted, the users current working
            directory is used.

      -config
            allows extensive tailoring of the deckfile by
            loading only those modules from the tape that conform
            to the current configuration.

      -list, -ls
            specifies that no modules are to be loaded from the
            tape; only a listing of the contents of the Firmware
            and T & D Library Tape is generated.

-firmware, -fw
    specifies that only firmware modules are to be loaded
    into individual segments; no deckfile or listing file
    will be produced. This argument is mutually exclusive
    with the "-deckfile" and "-list" arguments.

-deckfile, -dkf
    specifies that  no firmware  modules are to be loaded
    into  individual  segments,  only a  deckfile will be
    produced.  This argument  is mutually  exclusive with
    the "-firmware" argument.


Notes

    The default action of this  command, if no control arguments
are  specified, is to  load  all  modules on  the  tape that  are
applicable to Multics into the  deckfile and all  firmware modules
into individual segments.

    Firmware segments are created (in memory image format) using
a three component entry name, in the following format:

    fw.name.ident

where name is the name of the  device and ident is taken from the
MPC-assembler IDENT pseudo-op card.

    Each module  that is read  from the tape  and written to the
deckfile is copied as is in  GCOS object deck format. Each object
deck occupies one  record of the keyed  sequential deckfile.  The
record search key is formed  from information obtained from the $
OBJECT card.

    If the deckfile  already exists in the  specified directory,
this file is  "updated" with the new  modules read from the tape.
This is  done by  first  deleting records  for  which the record
search  key  already  exists  and  rewriting  the  contents of the
record,  leaving  records  in  the  file  that do  not  have  a
corresponding entry read from tape, unchanged.

Catalog records are built for all ITR and MDR tape files while the respective tape files are being read in. These catalog records are merely a list of all of the search keys associated with each individual ITR or MDR tape file and are used by the respective ITR and MDR T&D drivers to determine test sequencing. Each catalog record is written to the deckfile immediately following each ITR or MDR tape file, and have record search keys in the form:

    cata.itr.<mpc_name> - for an ITR catalog or
    cata.mdr.<device_name> - for an MDR catalog

A listing segment is created in the users working directory. This listing segment contains a directory of ITR, MDR, Firmware, T&D test pages and catalog records loaded from the tape and a corresponding entry for each deckfile entry. The listing segment has the name "tape_name.list".

APPENDIX A

# Honeywell

# HONEYWELL PROPRIETARY

No. TDM– ITD - 77

Title: FORMAT CHANGES IN THE SYSTEM
FIRMWARE TAPE

Author: G. A. HANEY

Date: 11-8-78     Rev. 0

CONTENTS

## 1.0   Introduction

This memo describes the format and contents of the new
System Firmware Tape and the requirements for modifying
or expanding said contents.  Although the primary users
of the new tape features will be the Test & Diagnostic
software modules, the differences between the old and
new firmware tape formats carry implications for all
modules which read any portion of the tape.  The final
sections of this memo contain a detailed description of
the various implementation tasks involved in moving to
the new format.

## 2.0   Revisions

Revision 0, 11-14-78

3.0   <u>References</u>

1.   TDM-ITD-74, CP-6 MOLTS  Concept Design.

2.   TDM-ITD-76, CP-6 System Deckfile and Deckfile
         Utility Programs.

3.   Spec. No. 58007104, 6000 Firmware Index.

4.   GCOS DM #610, "Handling of MPC Firmware".

5.   TDM-ITD-79, "Library Editor For Firmware T&D".

## 4.0  Reasons for the Changes

There are 3 major reasons for adopting the new firmware tape format:

1.) Common T&D Package - L66 Systems Maintainability Design, faced with providing on line diagnostic (TOLTS) capability for GCOS-3, GCOS-66, CP-6, and MULTICS, has produced a common set of tests and Slave Mode Executives for all of the above operating systems. This is accomplished by placing all operating system dependent functionality in interface modules which are unique to each operating system. The firmware tape provides a single release mechanism for providing all software systems with these common modules.

2.) Timely Update Capability - The relative frequency of firmware tape releases, and the coupling of these releases to hardware FCO's, provides a means for keeping test programs consistent with changing hardware and firmware. Such changes may be invisible to operating system software but have adverse effects on test programs

3.) Separately Priced Software - The placing of T&D's on the firmware tape provides the site FER with a method of withholding the T&D's from those customers who do not purchase maintainability software, simply by copying only firmware from the Phoenix-provided master tape onto those tapes provided to the customers.

5.0    Modules on the New Firmware Tape

The following modules will be on the new System Firmware
Tape:

1)   Firmware for all supported firmware devices;
2)   ITR's for all supported MPC's;
3)   MDR's for all supported MPC devices;
4)   Before - After (BAFT) Routines for all MDR packages;
5)   TOLTS Slave Mode Subexecutives (POLTS, MOLTS, SOLTS,
     and COLTS);
6)   TOLTS Common Test Pages;
7)   HEALS and Subexecutive Interface Modules for all
     required GCOS and ACOS releases.
8)   T&D Utility Programs (to be defined).


Items (1) - (4) appear on today's firmware tape;
(5) - (8) are new.  The $OBJECT cards for (1) - (4)
will retain the "HMPCJ1" designation starting in column 16;
For (5) - (8), the designation will be "HTND".

6.0 <u>Physical Description of the Firmware Tape</u>

The new firmware tape will retain most of the characteristics
of the old - it will be unlabeled, multi-file, system
standard format, with 2 back-to-back  EOF's marking the
end of the last file.

The difference will be in the number of files and the
data on each file.

Files 1-n will consist of the ITR's and Application
Firmware for each of the n MPC controller models supported
by the firmware tape release; the next m files will contain
the MDR packages for each supported MPC device type, with
each set of MDR's preceded by its BAFT routine; the next
4 files will each contain a common slave mode executive
and its test pages (one each for POLT, MOLT, SOLT, and
COLT); the remaining files will contain the required
versions of HEALS and various T&D utility modules.

The difference in file contents is that the MDR's are now
in their own files, rather than being in the various
controller files along with ITR's and application firmware.

FIG 6.1 New Tape Format

← NO LABEL

ITR/Application Firmware files

MDR Files

TOLTS Common T&D Files

HEALS File

T&D Utility File

← Back to Back EOF'S

5
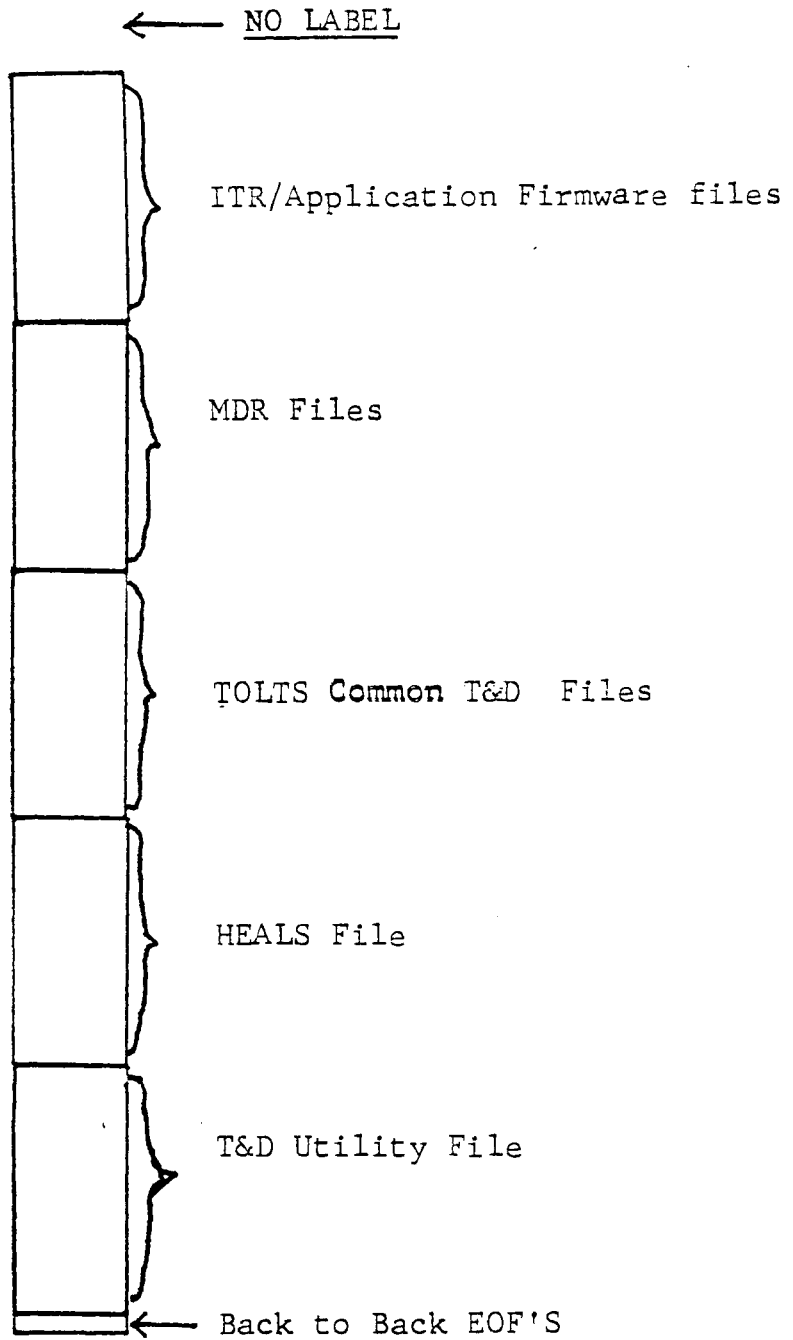
FIG 6.2   ITR/APPLICATION FIRMWARE FILE

| ITR #1 |
| ITR #2 |
| ITR #3 |

1 File For Each
   MPC Type

| ITR # X |
| APPLICATION FIRMWARE |
| EOF |

FIG 1.3     MDR File

| BAFT |
| MDR #1 |
| MDR #2 |
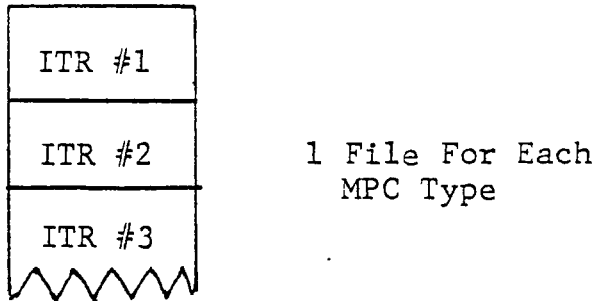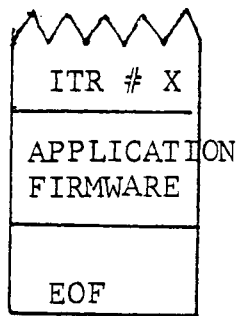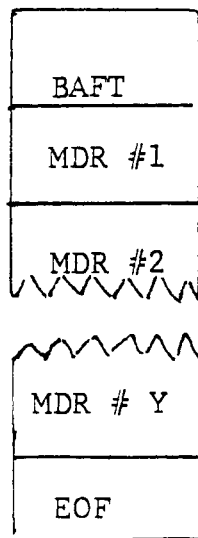
1 File For Each MDR package

| MDR # Y |
| EOF |

FIG 6.4   TOLTS COMMON T&D FILE

| |
|---|
| COMMON EXEC |
| TEST PROG. #1 |
| TEST PROG. #2 |
| TEST PROG. #3 |
| 〰〰〰〰〰〰 |
| TEST PROG. #2 |
| EOF |
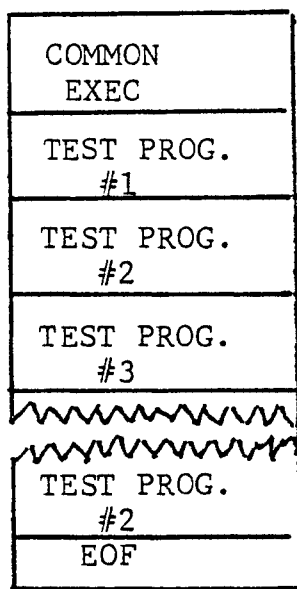
1 FILE FOR EACH TOLTS SUBEXECUTIVE.

FIG 6.5   HEALS FILE

| |
|---|
| HEALS FOR RLS 'A' |
| HEALS FOR RLS 'B' |
| 〰〰〰〰〰〰 |
| HEALS FOR RLS 'N-1' |
| HEALS FOR RLS 'N' |
| EOF |
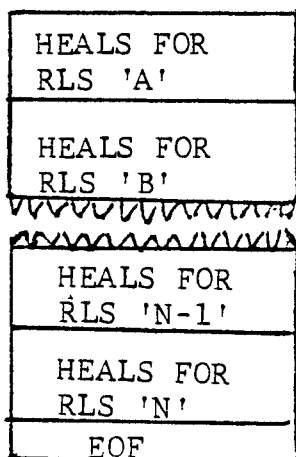
1 FILE, CONTAINING AS MANY VERSIONS OF HEALS AS REQUIRED.

FIG 6.6   T&D UTILITY FILE

| |
|---|
| UTILITY PROG. #1 |
| UTILITY PROG. #2 |
| 〰〰〰〰〰〰 |
| UTILITY PROG. #N |
| EOF |
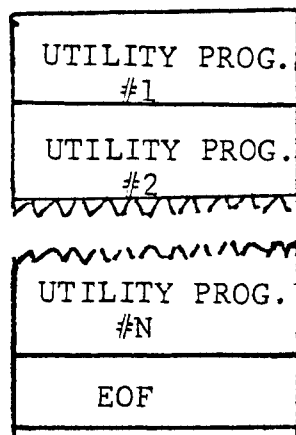
1 FILE, CONTAINING ALL UTILITY PROGRAMS.

17

7.0    Implications on Firmware Tape Users

7.1  T&D -  Test and Diagnostics Systems (TOLTS) for the various
operating systems will need to provide a means of getting
the common T&D modules from the firmware tape to the system
T&D deckfile.  Similarly, a utility routine will be re-
quired to retrieve these modules from the deckfile whenever
TOLTS is to execute.  This utility routine will be called
in place of the operating system call which presently fetches
the modules from the o.s. library.  (See references 1 and 2
for further information)  Also, Offline T&D programs that
build firmware tapes or read MDR's will require modification.

7.2  Others  -  Other system software which accesses the firmware
tape may have to be cognizant of one or more of the following
traits:
1)  MDR's are no longer in the ITR/Firmware files;
2)  The last HMPCJ1 deck (this will be an MDR) is not followed
by a double EOF, but by a single EOF and the first HTND
deck.
3)  The last application firmware deck is followed by a
single EOF and the first MDR file deck ( a BAFT routine.
This deck is distinguishable  from ITR/firmware decks by
information on the $OBJECT card.  These implications are
explored in more detail with specific regard for particular
operating systems and releases, in section 9.

## 8.0    Module Applicability

8.1    Definition and Usage  -  Module applicability may be
       defined as that set of operating systems and their
       releases for which a  particular version of any given
       module is valid.  With the increased number of modules
       on the firmware deck, the possibility exists that a
       system deckfile may become unnecessarily large (and
       therefore a customer annoyance) by containing data
       which the operating system, including TOLTS, will never
       use.  Such unused data could be HMPCJ1 or HTND, decks
       or both, due to one or more of the following conditions:
       1)  A given device or controller is not yet supported
           by this o.s.;
       2)  A given device or controller is no longer (or was
           never) supported by th o.s.; or
       3)  Several versions of a module exist, each for a
           particular o.s. or a subset of its releases.

       Examples of the above abound, i.e., DSS180 T&D's for
       MULTICS (case 1), CR 214 MDR's for CP-6 (case 2), and
       URC MPC firmware (case 3, GCOS vs. WWMCS).  Because of
       these and the many existing and potential cases like
       them, it is necessary to design a means of defining the
       applicability of each deck on the firmware tape.

8.2    Methodology

       The method for defining applicability will be via special
       fields on the $OBJECT card images for each deck on the tape.
       The original $OBJECT card data as produced by the various
       assemblers will be modified and merged with applicability
       data to form the new $OBJECT image on the tape at the time
       that the tape is made.  Figures 8.1 - 8.3 show $OBJECT images
       both in the original forms as they will appear on the
       firmware tape.

| PROBLEM TDM-ITD-77 | PROGRAMMER | REQUESTED BY | DATE | PAGE OF |

```
FIGURE 8.1       OLD AND NEW FIRMWARE TAPE $OBJECT IMAGE EXAMPLES

                 ITR'S AND APPLICATION FIRMWARE
OLD IMAGE:
$        OBJECT  HMPCJ1              DAT     MMMMMMMMM-MM        M       DDDDDEEEESSSS

NEW IMAGE:
$        OBJECT  HMPCJ1  6CGMWRR     DATA    MMMMMMMM-MM         M       DDDDDEEEESSSS

MMM.....M        -MPC MODEL INFO
M IN COL. 60     -ASSEMBLER CODE (M FOR MICROCODE-MPC)
DDDDDD           -ASSEMBLY DATE
EEEE             -MODULE EDIT NAME
SSSS             -SEQUENCE NO.

6CGMWRR          -MODULE APPLICABILITY INFO (SUPPLIED WHEN TAPE IS MADE)
                 "6" IN COL 24 MEANS NOT APPLICABLE FOR GCOS-66 (ACOS)
                 "C" IN COL 25 MEANS NOT APPLICABLE FOR CP-6
                 "G" IN COL 26 MEANS NOT APPLICABLE FOR GCOS-3
                 "M" IN COL 27 MEANS NOT APPLICABLE FOR MULTICS
                 "W" IN COL 28 MEANS NOT APPLICABLE FOR WWMCCS

                 COLS 29-34 WILL BE USED FOR FUTURE EXPANSION - BLANKS FOR NOW
```

C.E. 103-V-721

```
FIGURE 8.2  OLD AND NEW FIRMWARE TAPE #OBJECT IMAGE EXAMPLES

                 BAFT MODULES AND MDR'S
OLD IMAGE:
   OBJECT  HMPCJI  ******    DAT  MMMMMMMM        GTIITTTDDDDDEEEESSS

NEW IMAGE:
   OBJECT  HMPCJI  GCGMWRR   DATA MMMMMMMM        G******DDDDDEEEESSS

SEE FIGURE 8.1 FOR DEFINITION OF "DDDDDD","EEEE","SSSS",AND "GCGMWRR"
TTTTTT  ASSEMBLY TIME OF DAY (OVERLAID ON TAPE IMAGE)
MMM..M  DEVICE MODEL INFO FROM MODULE LABEL CARD
******  MODULE GECALL NAME FROM MODULE LABEL. WHEN FIRMWARE TAPE
        IS MADE, THIS FIELD IS MOVED FROM COLS 25-30 TO COLS 61-66
        AND APPLICABILITY INFO IS PLACED IN COLS 24-30.

G IN COL 60  ASSEMBLER CODE (G FOR GMAP)
```

TDM-ITD-77

TDM-ITD-77

FIGURE 8.3    OLD AND NEW FIRMWARE TAPE $OBJECT EXAMPLES

COMMON TED, HEALS, TED UTILITY

NEW IMAGE:

B    OBJECT  HTND  XY6CGMWRR           ZZZZZZZZZZZZ    G******* DDDDDEEEESSS

REFER TO FIGURES 8.1 AND 8.2 FOR DEFINITION OF FIELDS BELOW:

"6CGMWRR", "G", "DDDDD", "EEEE", "SSS"

******  MODULE GECALL NAME. WHEN TAPE IS MADE, THIS FIELD IS MOVED
FROM COLS 25-30 ON MODULE OBJECT CARD TO FIELD SHOWN ON IMAGE.
THIS OVERLAYS ASSEMBLY TOD INFO ORIGINALLY IN THAT FIELD.

ZZZ....Z.  MODULE INFO FROM MODULE LABEL CARD

X (COL. 22)  DECK LOAD TYPE:    (FOR FURTHER INFO, SEE TDM-ITD-76)

    IF X = "M", MASTER DECK (PAS2 TED) (MULTICS ONLY)
    IF X = "S", SLAVE DECK (PAS2 TED) (MULTICS ONLY)
    IF X = "P", PROGRAM (TOLTS TEST PAGE OR SLAVE MODE SUBEXEC.)
    IF X = "R", RELOC. (BAET ROUTINES.
    IF X = "D", DATA. (FIRMWARE, ITR'S, MDR'S)

Y (COL. 23)  TED SUBSYSTEM TYPE:

    "P" = POLTS        "S" = SOLTS (AND ISOLTS FOR MULTICS.)
    "M" = MOLTS        "H" = HEALS (HEALS, MPCD, ECFR)
    "C" = COLTS        "U" = UTILITY

NOTE: THE XY VALUES ARE INSERTED WITH THE APPLICABILITY DATA

Summary of $OBJECT Manipulation

1. The information in columns 25-30 of the module's $OBJECT
   card (as produced by the assembler) is moved to columns
   61-66. This information will be the module GECALL name
   for HTND modules and certain HMPCJ1 modules (BAFTS, MDR's),
   and "don't care" information (most likely blanks) for the
   remaining HMPCJ1 modules (ITR's, Application Firmware). As
   time permits, label cards for the "don't care" cases will
   be altered so that the field will always contain blanks.
   The information in columns 61-66 on the original $OBJECT
   card (time of day of assembly or blanks) will not appear on
   the firmware tape $OBJECT image.

2. A 7-character applicability code (5 currently used plus 2
   blanks) will be retrieved from a master file and placed in
   columns 24-30.

3. For HTND modules, an additional 2-character XY code will be
   retrieved from the master file and placed in columns 22-23.
   For HMPCJ1 modules, these columns will be set to blanks.

4. Columns 35-38 will be changed from "DATΦ" to "DATA" to
   indicate that (1) above has been done and should not be
   done again.

Implementation Note: Note that applicability codes as defined
in figure 8.1 are set up so as to flag which systems a module
is not applicable to; that is, an "M" in column 27 tells the
MULTICS deckfile builder to ignore this module, and an all-blank
applicability field means that the module should go on all deckfiles.
This implementation allows each program that builds deckfiles to
only be concerned with checking the particular column pertaining
to its operating system.

8.3     Procedural Considerations  -  The applicability in-
        formation for all firmware tape modules will be maintained
        by Firmware/T&D Processing & System Test (FTPST), the
        organization which produces and distributes firmware tapes.
        The initial set of applicability attributes will be jointly
        produced by FTPST and L66SMD.  FTPST must formally  define
        the procedures for either:
        1)   defining the applicability of a new module; or
        2)   Altering the applicability of an existing module.
        Further, FTPST will implement an automated procedure for
        reading the applicability information and producing the
        $OBJECT data for a firmware tape.  The timetable for these
        events is in section 10.

9.0    Operating System VS. Firmware Release Considerations.

The first firmware tape release in the new format will
be release AL scheduled for FW913.  Given this information,
one can now proceed to answer some questions for each L66
operating system:
1)  What release any new functionality should go in;
2)  What exactly that functionality is;
3)  What prior releases will use the new firmware tape
    and must be altered to accomodate it; and
4)  What exactly those alterations are.

The net effect of the new tape format on operating system
software will be beneficial because of a reduction or
simplification of required functionality.  The following
paragraphs answer the above four questions as they relate
to each operating system.  Most of the affected software
development organizations have already been notified of
the changes, with the result that much of the required
work is underway or done; the intent here is merely to
document all of the information in one place.

9.1    GCOS-3

9.1.1  Patches will be released to SR 3/I, SR 4/J and SR4/JS1
       Startup which will allow either the old or new format
       tape will be ignored.  These patches have already been
       designed by D. Augustine and given to the Startup developers.
       They can be tested when a firmware tape in the new format
       becomes available.

9.1.2  GCOS 5/K will contain the functionality to fully benefit
       from the new tape format.  5/K Startup will be  much simplified,
       since it will read and catalog application firmware only and
       no longer be concerned with ITR's, MDR's, and BAFTS.  All
       modules other than application firmware will be read and
       cataloged by HEALS via user option. The HEALS module released
       to 5/K will contain only the functionality decide whether or
       not the system deckfile has yet been built; if not, HEALS
       will call for the firmware tape and catalog and call the "real"
       5/K HEALS from the tape.  5/K HEALS can then complete the
       building of the system deckfile.  Firmware release AL or later
       will be required for GCOS 5/K.

9.1.3    By retrofitting (via an insert) this HEALS functionality
         into previous GCOS releases, full 5/K T&D functionality
         can be achieved for customers utilizing firmware release
         AL or later.  This will be done as time permits; it's
         not mandatory.


9.2      ACOS


9.2.1    The same procedure will be followed for ACOS as for GCOS-3,
         with the exception that the final retrofit as outlined in
         9.1.3 will not be done, since ACOS has not yet shipped.
         The ACOS release that will contain the full functionality
         (analagous to GCOS 5/K) is 4VX.

9.3    MULTICS

9.3.1  For Multics releases prior to MR 7.0, a utility program
       (Load_Firmware_File) is used to read the firmware tape
       and create a core loadable image segment of each firmware,
       ITR, and MDR object deck.  These individual segments are
       then combined into one large segment (an archive segment)
       which is used by MOLTS as a source for those decks.  The
       application firmware modules are extracted from the archive
       segment and are written on the BOS system startup tape by
       a system utility program that builds BOS tapes.  It must
       be determined whether the Load_Firmware_File utility can,
       with the proper options applied, handle the new tape format.
       If not, firmware release AL will be tied to MR7.0.

9.3.2  Two new utility routines will be developed for MR 7.0 by
       L66SMD.  The first will create a complete system T&D
       deckfile from the firmware tape, and the second will use
       the deckfile records to create files in a format acceptable
       to the (unchanged) module which builds BOS tapes.  MR 7.0
       will thus have full benefit of the new tape format.

9.4    <u>CP-5</u>

9.4.1   The firmware tape contains special application firmware
        and ITR's for use on MPC's coupled to SIGMA systems.
        The CP-5 module which reads the firmware tape is currently
        smart enough to find those files that it needs, so no
        changes are anticipated for the new tape format (this
        will be verified).  No increased functionality for CP-5
        will result from the new tape format.

9.4.2   The program which reads the system firmware tape and
        builds PO (startup) tapes for CP-6 runs on the CP-5
        system.  It must be determined what changes this program
        will need and what aligning of firmware and CP-5 releases,
        if any, is required.  Since the PO tape contains only
        application firmware, the format differences are likely
        to be invisible.

9.5    <u>CP-6</u>

9.5.1   The CP-6 system will not initially (year-end) utilize
        the system firmware tape.  Since MOLTS will not initially
        be available, ITR's and MDR's will not be required.  The
        remaining TOLTS functionality (POLTS, SOLTS, and
        their test programs) will be released as part of the CP-6
        operating system.

9.5.2   The date and method of adding MOLTS and the firmware tape/
        system deckfile utility to CP-6 are still to be determined.

## 9.6    Offline T&D

9.6.1  A new release of Offline T&D (release B3) will be
       required to support firmware release AL. (This
       does not imply a special release of any sort, since
       each release of the firmware tape is normally accompanied
       by a T&D release; AL and B3 are both scheduled releases).
       Changes will be made to program 08F, the offline program
       that builds firmware tapes, to produce $OBJECT card images
       on the tapes consistent with the new format.  Also, because
       of the new format of the $OBJECT cards on the tape, the
       offline MDR Driver programs must be changed to pick up
       the MDR GECALL module names from a different field on the
       card.  T&D release B3 and firmware release AL must therefore
       be tied together such that neither  one will be required to
       interface with a previous release of the other.

## 10.0   Master Implementation Plan

## 10.1   New OS Releases

The full usage functionality for all new o.s. releases
as described in section 9 must be implemented so as to
fit the various release schedules for these systems.
This includes GCOS 5/K, MULTICS 7.0, ACOS 4VX, and CP-6.
The reader is referred to the current schedules of each
of these events for exact dates, the point being that
the dates are based on considerations other than and
subsequent to, the schedule for firmware release AL.
It should again be emphasized that plans are in place,
and that the net effect will be a smaller requirement
on operating system software.

## 10.2   Fixes to Released OS versions

These include GCOS-3 (back to 3/I), ACOS 4V, and (possibly)
CP-5.  These fixes must be in place prior to the time
firmware release AL is made (913).

## 10.3   Prior MULTICS Releases

The determination of whether MULTICS releases prior to 7.0
can handle the new tape format must be made and documented
to FTPST by FW901 so that, if necessary, release AL can be
tied to MR 7.0.

## 10.4   Offline T&D

Offline T&D changes to support the new tape format must be
made by FW905, the comdk cutoff for offline release B3.

10.5   Other

10.5.1   Memo review   -   This memo should be reviewed and
comments returned by 11/30/78.

10.5.2   Firmware Tape Edit Program   -   Development is already
underway on the program to produce the new format
firmware tape.   Estimated completion date is 11/17/78,
including  documentation. (TDM-ITD-79)

10.5.3   New Modules   -   LBL card changes for HTND modules and
applicability code is scheduled for completion by FW905.

10.5.4   Preliminary Version of New Tape   -   A preliminary version
of the new firmware tape including HMPCJ1 modules in the
new file format and HTND modules will be available as of
11/17/78 for testing software.

10.5.5   Procedures for Applicability Changes   -   FTPST will have
defined procedures for defining and changing module
applicability by FW901.

## Milestones

### FW846

- Memo distributed        (10.5.1)
- Edit program complete  (10.5.2)
- Preliminary firmware tape available for software testing (10.5.4)

### FW848

- Memo comments due      (10.5.1)
- CP-5 Work requirements defined (10.2)

### FW901

- Applicability update procedures in place    (10.5.5)
- MULTICS relationship to release AL defined (10.3)

### FW905

- Offline T&D fixes complete  (10.4.)
- HTND modules transmitted      (10.5.3)

### FW912

- Fixes transmitted for current releases  (10.2)

### FW913

- Firmware release AL