To: MTB Distribution

From: Richard Holmstedt (HVN 341-7248)

Date: 01/08/81

Subject: Software Technical Identifier

INTRODUCTION

This bulletin is intended to communicate the proposed standard for use in the identification of software products. A description of the Software Technical Identifier (STI) its meaning, use and proposed implementation plan for Multics is included in this topic.

SCOPE

The following is a synopsis of the proposed HIS standard for STI as it applies to Multics. At the present time there does not exist a common standard for the identification of software products. Each division has invented several different formats without any standardization. This has presented several problems, such as, HIS business systems can't deal with the volume of different formats. Other product line support responsiveness has been hampered due to lack of a means of identifying software status. It has inhibited DSE and product set integration. Our customers and outsiders view Honeywell as being fragmented.

In order to overcome these drawbacks the new standard will apply to systems and application software products which are either developed by HIS or Cii-HB. Third party software products that HIS or Cii-HB distributes will also be identified by the STI. In order to allow easy access remotely to current status of any code the STI will be internally embedded in all source, object archives and bound components as well as externally marked on program listings and media. The local design division (in our case LISD) will make the assignment of STIs to application software products and then communicate the assignment to the HIS - Architecture and Technology staff. The administration of the relationships between Marketing Identifiers and STIs will be maintained by HIS - Architecture and Technology in Minneapolis.

RELATIONSHIP TO EXISTING HIS STANDARDS

Internal Product Identifiers (IPIs)

This has been the standard method for the identification of hardware products. It identifies physically different products that support a common functionality. Its intended design is not suited for use with software needs as it is dependent on a hardware release cycle without any relevance to software releases. This was used as the model for the design of the STI format.

Marketing Identifiers (MI)

The Marketing Identifiers are used to produce a means of identifying product offering that are available for general release. When referred to internally, these marketing identifiers are commonly referred to as Level I to distinguish them from any other identifiers applied to the product (See Exhibit 2).

The problem with marketing identifiers as they apply to software is they do not allow for technical evolution; that they depend upon marketplace strategy. The STI is intended to provide a stable associative link between the Marketing and Development identification systems. The relationship between Marketing Identifiers and STIs can be as follows:

Marketing Identifier		Software Technical Identifier
One	to	One
One	to	Many
Many	to	One
Many	to	Many

These relationships are further compounded by the fact that engineering designs and documentation are not necessarily linked with the time cycle selected in Marketing strategy, for example, software unbundling or roll-up of functionality.

STI FORMAT

The STI consists of two components called the Software Product Identifier (SPI) and Software Version Index (SVI).

```
                    Software Technical Identifier
                    /                           \
   Software Product Identifier           Software Version Index
   /                        \            /            \
  D   C   F   F   F    L1   L2          R    S    M1     M2     E
  a   n   a   a   an   an   an          an   an   an     an     an
  ¦   ¦   \       /    \     /          ¦    ¦    \      /      ¦
  ¦   ¦     \   /    Local design       ¦    ¦      \  /        Early Release
  ¦   ¦       ¦      Division Use       ¦    ¦       ¦
  ¦   ¦       ¦                         ¦    ¦       Maintenance/Controlled
  ¦   ¦       Functional Code           ¦    ¦             Releases
  ¦   ¦                                 ¦    ¦
  ¦   Code Format                       ¦    Supplemental Release
  Prime Design Division                 Major Release
```

a - alphabetic character (excluding I, O, Q and Z)
an - alphanumeric character (excluding I, O, Q and Z)
n - numeric character

Example of an STI: W1MSSM090000

SOFTWARE PRODUCT IDENTIFIER (SPI)

DESCRIPTION

The Software Product Identifier (SPI) is the first seven characters of the STI which is used to identify a specific product. An SPI will live for the life of that functionality that is needed to satisfy a marketing identifier. The only exception to this would be the Prime Design Division code.

Prime Design Division (Character Code D)

This alpha character is used to identify uniquely the development facility responsible for supporting the product identified by the SPI. If support responsibility shifts from one division to another then the character will change at the next product release. For LISD, the code is "W".

Code Format (Character Code C)

This character is used to indicate the code format of the product i.e. source or executable code. The character must be numeric.

1 = Source code
2 = Object
3 = Executable
4 = Listings

Functional Code (Character Code FFF)

This three character code is used to identify a functionality rather then a product. Functionality is a commercially visible capability such as a language compiler or a sort. The characters identify the specific functionality being represented by the SPI. Where possible, these characters should have a mnemonic relationship to the software they describe and should be common throughout HIS and Cii-HB. The three characters comprising the functional code will always pertain to a single specific function, not a roll-up of functions. For example, the COBOL compiler is a single specific function. The sort feature for COBOL is also a specific function. In this case there would NOT be a separate function code for the COBOL compiler combined with the sort feature this would all be considered a part of the COBOL functionality. See Exhibit 1 for proposed functional codes.

Local Design Division Use (Character Code L1 L2)

This alphanumeric field is for use by the local design division. A zero will be placed in each unused character position.

SOFTWARE VERSION INDEX (SVI)

DESCRIPTION

The Software Version Index (SVI) consists of the last five characters of the STI and is used to describe a hierarchical release mechanism. The SVI is that portion used for the identification of release information including successive major, supplemental, maintenance and early releases of the same software. It will be the responsibility of Multics Software Support to control the assignment of the SVI.

Major Release Indicator (Character Code R)

This alphanumeric character is the products major release indicator. It is assigned in ascending order 1 through 9, then A through Y. The first major release to use the STI will be MR9, the major release indicator will initially be set to 9.

Supplemental Release Indicator (Character Code S)

This alphanumeric character is used to identify a release with minor functionality modifications. It is assigned in the same manner as Major Release Indicators.

Maintenance/Controlled Release (Character Code M1 M2)

The alphanumeric characters M1 and M2 will be set to 00 for all major and supplemental releases. Any subsequent maintenance releases (i.e. bugfix) will result in an increment by one from 1 to 99. Any releases needed beyond 99 the alphanumeric character M1 becomes A and M2 increments from 1 to 9. M1 will continue to be incremented A through Z while M2 will recycle from 1 to 9 as necessary releases are made.
For controlled releases (pre-release) of new major and supplemental software, the character M2 will be alphabetic starting with the letter A to indicate the first controlled release and character M1 will be zero. Any subsequent controlled releases before the general release will result in M2 incremented alphabetically.

Early Release (Character Code E)

For partially qualified releases an alphabetic character starting with A will be placed in this normally zero character. Subsequent corrections prior to a fully qualified release will be incremented alphabetically.

PROPOSED IMPLEMENTATION PLAN FOR MULTICS

In order to comply with the MR9 PFS, a facility will be needed by which a user can login to a system and obtain a list of Priced Software Products, their STI numbers and titles installed on that site. This facility will be accessible via the DPU remote connection or via terminal connection thru the FNP. The intention of STIs is not to replace the tools or methods now used for the identification of sortware, i.e. compare_ascii or print_bind_map. It is intended to provide a method for tracking distributed software with a standardized format reguardless of product lines. The following is proposed to meet this specification.

o Identification of Online Software.

The first step in implementation of this facility will be to establish a procedure for embedding the STI in the distributed online software. Since all source, object and executable software must be protected by a copyright notice (see MTB 478), the segment which contains this copyright will be expanded to include information about the products STI. Since the copyright notice record must become the first component of the total segment, locating the identifier is simplified greatly.

A descriptor segment will be created which will be used by a new command that will be known as print_psp (See Exhibit 3). This descriptor will be used to identify the principal segment of a Priced Separate Product from which the source, object and executable STI will be extracted. The syntax of this descriptor segment will be similar to that used by other segment input driven command, i.e. library fetch or bind. An example of this format follows:

```
/* Descriptor segment for use by the print_psp command. */

Define: AGS6801;

product name: Timesharing Library;

path: >system_library_unbundled>bound_gcos_tss_;
path: >library_dir_dir>unbundled>source>bound_gcos_tss_.s.archive;
path: >library_dir_dir>unbundled>object>bound_gcos_tss_.archive;

End:;
```

1. Example of a descriptor entry.

Since release cycles and emergency fixes have been unpredictable in the past, and will probably remain so in the future, the copyright component will be updated with a new STI applied only when software is prepared for delivery to the customer. Multics Software Support now has the responsibility for preparation of software for delivery to a site. The additional responsibility for maintaining the STI and the STI descriptor segment will also be included.

o Identification of Communication Software.

Since the communications software consists of free standing segments this creates a special problem due to its non-standard format with regard to other libraries. In order to permit the use of the copyright segment and STI information contained in this segment, it will be necessary to archive the source and object segments under common product functions.

bound_mcs_.s.archive	W1MCSM090000	Multics Communications System
bound_autocall.s.archive	W1ATCM090000	Autocall Support Option
bound_mcs_ext_.s.archive	W1IBMM090000	3270 Support Option
bound_bisync_.s.archive	W1BISM090000	Bisync Support Option
bound_ges_.s.archive	W1GESM090000	G115 Support Option
bound_filetrans_.s.archive	W1FTSM090000	File Transfer Facility
bound_network_.s.archive	W1NETM090000	Network Tools

2. List of Communications Source Archives.

The following example would comprise the product SGC6804, (G115 support option to MCS).

```
>library_dir dir>communications>source>bound_ges_.s.archive
  ges_copyright_notice_.alm
  g115_tables.map355

>library_dir dir>communications>object>bound_ges_.archive
  ges_copyright_notice_
  g115_tables.objdk
```

3. Contents of Communications Archives by a Product.

The treatment of executable communications software has its own special problems, this being the code that is executed is not directly available online. Since the communications processor is a separately running system under control of the main system, a core image is downline loaded from the host system into this processor. This communications software has available now the module names and the dates on which the individual modules were assembled. This would appear the appropriate place to expand to include the STI information. The routines that are now used by the db_fnp command that make the modules name and date available should be expanded to include STI information. This information should then be available to the print_psp command for printing as well. The type of information which would be available is given in the example that follows:

```
! db_fnp
! fnp b
! map

sked       4540   12/03/79   W3MCSM090000
intp       7224   02/07/80
dia       13140   12/02/80
hsla      21660   09/09/80
util      31250   05/05/80
g115      36400   11/02/79   W3CGEM090000
bsc       42200   12/09/80   W3BISM090000
ctrl      46700   10/30/80
cons      52250   05/10/78
bkpt      53204   11/02/79
acu       53540   11/02/79   W3ATCM090000
hasp      53720   09/09/80   W3HSPM090000
pvip      55300   11/02/79
auto      57652   11/02/79
x25lap    60246   09/09/80   W3X25M090000
init      64706   11/02/79
```

4. Output from db_fnp command.

o Identification of SPS SGS6800 (Multics Operating System EXEC).

The identification of the Multics hardcore has traditionally been done through use of a value in active_all_rings_data. This value will be changed to comply with the standard using the 12 character STI for the Multics Operating System EXEC, (W3MOSM09000). This will allow printing of the executable operating system exec STI and give visability to the new standard.

Methods for tracking software shipped to sites.

Currently a database in lister format is maintained in the directory, >udd>sm>fwm>MR8.0dumps>ls. The information now available is limited to site name, marketing identifiers and product name. With slight modification the lister file could be enhanced to identify STI modules shipped as well. Access and maintenance of this database is controlled by Multics Software Development. Exec_coms are now available to access information about software currently installed at the sites.

```
                 Nomenclature Mappings
                 |       |         |
                 |       |         |
                 v       v         v
 Site BCO  =>   MI  =>  STI  =>   Modules
```

The following tools have been developed and are located in the database directory:

o Identification of software installed at a site.

```
ec proc_unb by_site <site name>
```

This exec_com will produce a segment called by_site.<site_name>. The format of which will contain a list of software installed at a site as follows:

```
Date: December 12, 1980
SITE: OU
System number: LR0005

W1TSSM090000  AGS6801  Timesharing Library
W1GRFM090000  AGS6803  Graphics Facility
W1MCSM090000  SGC6800  Multics Communications System (Multics CS)
W1GESM090000  SGC6804  G115 Support Option to Multics CS
W1FTSM090000  SGC6805  File Transfer Facility
W1RDSM090000  SGD6800  MRDS (Multics Relational Data Store) Facility
W1LIUM090000  SGD6801  LINUS (Logical Inquiry and Update System)
W1MSSM090000  SGE6800  Multics System Software Extensions
W1RJEM090000  SGE6802  Remote Job Entry Facility
W1FT7M090000  SGL6801  FORTRAN Compiler and Runtime Facility
W1BISM090000  SGL6802  Basic Compiler and Runtime Facility
W1CCDM090000  SGL6803  COBOL-74 Compiler and Runtime Facility
W1RPGM090000  SGL6805  MRPG (Report Generator) Facility
W1APLM090000  SGL6806  APL (Version 2)
W1MOSM090000  SGS6800  Multics Operating System EXEC
W1FASM090000  SGS6803  FAST/DFAST (Fast Access System for Timesharing)
W1GTEM090000  SGS6804  GCOS (III) Batch Environment Facility
W1SMGM090000  SGU6801  SORT/MERGE Facility
W1EMFM090000  SGU6807  Extended Mail Facility
```

o Identification of sites by software ordered.

```
ec by soft <marketing identifier>
```

This exec_com will produce a segment called by soft.<marketing identifier>. This segment will contain a list of sites which have ordered a given product as follows:

```
Date: December 22, 1980
Software code: SGD6800
Software: MRDS (Multics Relational Data Store) Facility

Site      System
Ident     Number    Account Name

AFDSC     LM0011    US Air Force (DC)
BCO       BJ8002    BCO
CNO       LR0002    Honeywell Corporate Tech. Ctr.
Ford      LR0007    Ford Motor Company
GM        LR0013    General Motors Corporation
NWGS      LR00XX    Naval War Games
OU        LR0005    Oakland University
UC        LR7002    University of Calgary
USGS-D    LM0005    US Geological Survey (Denver)
```

Proposed Software Technical Identifiers

	MARKETING	
STI	IDENTIFIER	FUNCTIONALITY
W1TSSM090000	AGS6801	Timesharing Library
W1STAM090000	AGS6802	ISTAT
W1GRFM090000	AGS6803	Graphics Facility
W1MCSM090000	SGC6800	Multics Communications System (Multics CS)
W1ATCM090000	SGC6801	Autocall Support Option to Multics CS
W1IBMM090000	SGC6802	3270 Support Option to Multics CS
W1BISM090000	SGC6803	Basic Bisync Support Option to Multics CS
W1GESM090000	SGC6804	G115 Support Option to Multics CS
W1FTSM090000	SGC6805	File Transfer Facility
W1NETM090000	SGC6806	Network Tools
W1RDSM090000	SGD6800	MRDS (Multics Relational Data Store) Facility
W1LIUM090000	SGD6801	LINUS (Logical Inquiry and Update System)
W1MSSM090000	SGE6800	Multics System Software Extensions
W1RJEM090000	SGE6802	Remote Job Entry Facility
W1FT7M090000	SGL6801	FORTRAN Compiler and Runtime Facility
W1BISM090000	SGL6802	Basic Compiler and Runtime Facility
W1CCDM090000	SGL6803	COBOL-74 Compiler and Runtime Facility
W1RPGM090000	SGL6805	MRPG (Report Generator) Facility
W1APLM090000	SGL6806	APL (Version 2)
W1MOSM090000	SGS6800	Multics Operating System EXEC
W1GTEM090000	SGS6801	GCOS (III) Timesharing Environment Facility
W1TPTM090000	SGS6802	Transaction Processing Tools
W1FASM090000	SGS6803	FAST/DFAST (Fast Access System for Timesharing)
W1GTEM090000	SGS6804	GCOS (III) Batch Environment Facility
W1WPCM090000	SGU6800	WORDPRO Comprehensive Facility
W1SMGM090000	SGU6801	SORT/MERGE Facility
W1LISM090000	SGU6803	LISTER Facility
W1SYPM090000	SGU6804	SPEEDTYPE Facility
W1DICM090000	SGU6805	Dictionary Tools
W1EMFM090000	SGU6807	Extended Mail Facility
W1COMM090000	SGU6820	Compose Facility
W1TEDM090000	SGU6833	TED (Text Editor) Facility
W1EMCM090000	SGU6834	Emacs Text Processing Facility
W1PPSM090000	SGU6835	Offline Page Printing System Support Facility

(Exhibit 1)

MARKETING IDENTIFIER FOR GENERAL SOFTWARE

```
                                             S  O  F  N  N  N  N
                                             |  |  |  ----------
                                             |  |  |       |
S = General Software_________________________|  |  |       |
A = Application Programs                        |  |       |
                                                |  |       |
Operating System:_______________________________|  |       |
  A = 58 (61 GCOS)                                 |       |
  B = 62 GCOS                                      |       |
  C = 64 GCOS                                      |       |
  E = 66/6000 GCOS                                 |       |
  F = 66 GCOS                                      |       |
  G = Level 68                                     |       |
  T = Terminal (Intelligent)                       |       |
                                                   |       |
                                                   |       |
Basic Function:____________________________________|       |
  C = Communications                                       |
  D = Data Management                                      |
  F = File System                                          |
  J = Job Management                                       |
  L = Language                                             |
  M = Program Mode                                         |
  P = Programming Tools                                    |
  S = Supervisor                                           |
  U = Utility                                              |
  V = Conversion Tools                                     |
(All other letters are reserved                            |
 and will be assigned as needed)                           |
                                                           |
Nonsignificant sequential number___________________________|
```

(Exhibit 2)

----
print_psp

Syntax as a command:

```
              print_psp {-control_args}
```

Function: displays selected information about distributed software found installed in online systems libraries. The information includes marketing identifier, software technical identifier, and titles for the software requested.

Arguments:

Control arguments:

-sti Marketing Identifier
   This argument will return selected information for a specific product if found installed in the systems library.

-all, -a
   Will return selected information of all products found installed in the systems libraries. This is the default.

-brief, -bf
   Only the software technical identifier will be printed. This will be the default.

-long, -lg
   will print the marketing identifier, software technical identifier, and titles selected.

The -brief and -long; arguments are mutually exclusive and only one argument can be given in a command.

(Exhibit 3)