

To: MTB Distribution
From: Joe M. Stansbury
Date: February 4, 1981
Subject: Multics Software Protection

Comments or questions on this MTB may be directed via Multics mail to Stansbury.Multics or by phone to HVN 341-7240. The proposed implementation discussed herein is subject to change due to re-interpretation or revision of HIS Policy 14 by HIS and/or LISD.

A. INTRODUCTION

This MTB documents procedures and tools used to provide approved legal means of protecting Multics software and firmware produced or distributed by Honeywell Information Systems, Inc. (HIS). The MTB is intended to provide a technical proposal to meet the requirements of HIS Policy 14, titled "Protection of Software and Firmware". MTB-492 (Software Technical Identifiers) is related to this MTB because of the planned use of the proposed `generate_copyright` command. Copyright tools that are in current use will be modified. Some responsibilities of installers will change. Developers need to become more aware of the HIS policy toward software protection. A Multics Administrative Bulletin (MAB) will describe project control procedure changes as well as provide a source of information for developers and installers.

B. OUTLINE

The following topics are discussed in this document:

Description of the current procedures and tools used to protect Multics software.

Definitions of the new software protection classifications.

Classification of Multics software based on the new policy.

Description of proposed new procedures and tools. The area of procedures will be more fully addressed in a Multics Administrative Bulletin.

Appendices are included at the back of the document to support and further help explain its content.

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

C. CURRENT PROCEDURES AND TOOLS

Multics software installation personnel are responsible for adding, and optionally, deleting copyright notices. There are no trade secret notices in the current scheme of things. Installers of software at System M check software for the copyright notices before installation time, and if necessary, will add the notice(s).

Tools currently used to manipulate copyright notices are:

1. `add_copyright` - User interface to the `copyright_notice_subroutine`.
2. `copyright_archive` - User interface for adding copyright notices to an archive of source programs.
3. `copyright_notice` - Module called by `add_copyright` to add a notice to program source, and optionally delete an old notice.
4. Copyright notices for each source language are located in `>ldd>include`.

Their naming is of the form `<suffix_name>.copyright`.

Copyright notices are inserted at the beginning of source code programs and are bracketed according to various source code comment delimiters. There is one copyright form currently in use. Its format (sans delimiters) is as follows:

Copyright (c) 1972 by Massachusetts Institute of
Technology and Honeywell Information Systems, Inc.

Software protection now encompasses only source code. There are no procedures now in place to legally protect object code and executable code or communications system source, object and executable code.

D. DEFINITION OF SOFTWARE PROTECTION CLASSIFICATIONS

New software protection classifications are defined in HIS Policy 14 (HIS-14). The following definitions are as described in that policy dated April 18, 1980. These are provided for information purposes only and are not subject to change within review of this MTB. There are four (4) software and firmware legal protection and release classifications that apply to source, object and executable code. These are:

NEW PROTECTION CLASSIFICATIONS:

	NOTICE	RELEASE CLASS	REQ'D APPROVALS
Class A	Copyright	General	Normal approval, i.e. PLN, Step IV IPR
Class B	Copyright	Limited	VP & GM releasing Component (delegatable)
Class C	Trade Secret	Limited	VP & GM releasing Component (delegatable)
Class D	Trade Secret	Restricted	VP & GM developing Component (^delegatable)

Note:

"Copyright" protects the form, contents or structure.

"Trade Secret" protects the know-how, algorithms, knowledge, etc. Trade secret carries with it some internal industrial security implications.

CLASS A - software required for a customer to operate the system. This is normally object code and executable code which would not readily reveal a trade secret.

CLASS B - software for which some limitation on distribution is desired, and which is not required for normal operation. This normally is source code, but there will be exceptions. Approval for release of must come from the Releasing Component (not LISD), and is delegatable one level.

CLASS C - software containing ideas and algorithms which constitute a trade secret. This may include system source code, FED T&D routines, and other similar software. This software is a product trade secret. Approval for release is identical to CLASS B.

CLASS D - software protected because of inherent "know-how" techniques or processes. This is a company trade secret. Examples of this are firmware source code, design automation programs and internal tools. Approval for release must come from the Developing Component (LISD, i.e., R. R. Douglas) and is not delegatable.

E. NEW CLASSES OF MULTICS SOFTWARE

There are eight (8) broad categories of Multics PSP (Priced Software Products). These are included as APPENDIX A. As of MR8.0, all Multics object and executable code will be (is), classified as Class A. As of MR8.0, all Multics source code will be (is), classified as Class B. It is possible however, that future releases will include software categorized as Class C or D.

Generally, Multics software will be single-party software, ie. only Honeywell will have the copyright. Provisions must be made however, for the protection of third-party software, i.e., code in which Honeywell and some other party have a joint interest. This possibility will be provided for source code in new control arguments added to the add_copyright command. Similar means will be provided to allow protection of object code.

F. NEW PROCEDURES AND TOOLS

As mentioned earlier, installers of software will assume new responsibility for software protection. The installer will be responsible for adding notices to source code, and will screen all changes before actual installation. There are two notable exceptions to the proposed standard software protection methods. These are communications software and hardcore/BOS software.

1. Proposal for protection of communications object and executable code

This proposed procedure would provide protection to Multics PSP's SGC6800 through SGC6806, designated as 'COMMUNICATIONS SOFTWARE'. A copyright notice will be inserted at the front of each source code module as is the case with other software. Another method of protection must be used with regard to executable code because of space limitations of FNP memory. It is proposed therefore, that a shorter, 26 character copyright notice be inserted into an unused portion of the system communications region of FNP memory. This method would allow the copyright notice to be resident in FNP memory while it is running. It is not possible to place the notice at the beginning of the FNP object code, since the interrupt vectors are there. The format of the notice would be:

"COPR. (C) H.I.S. INC. 1980"

It is further proposed that the bind_fnp command be modified to put the notice into the correct place in the code. This way, it can't be omitted, and it will go into otherwise unused low memory and won't use buffer space. The protection of object code would follow the procedure described later in this topic. That procedure will work provided that the archival of communications software is undertaken as described in MTB-492.

2. Proposal for protection of object and executable hardcore and BOS.

This proposed procedure would provide protection to PSP SGS6800. Since the toehold is available to BOS and Multics, it is proposed that it be modified to contain a copyright notice. This would allow protection of both with a single notice. It is proposed that the flagbox within the toehold be modified to contain a copyright notice, and that the BOS init sequence (specifically FWLOAD) be modified to place the notice into the flagbox. In this way, a dump of BOS prior to boot would contain a notice. The `get_flagbox` command could be modified to print the copyright notice in some way, perhaps with a control argument. With the copyright notice in the flagbox (toehold), any dump of BOS or the hardcore system would include a copyright notice. It is also proposed to place a copyright notice into `active_hardcore_data` to provide protection to the contents of the Multics System Tape itself, lest its contents be dumped into another segment.

3. Proposed changes to copyright tools.

The tools `add_copyright` and `copyright_archive` will be modified so that up to three control arguments may be given with them when adding or changing a copyright notice for source code. The `copyright_notice_subroutine` will be updated to provide this functionality. The proposed `add_copyright` command description is provided in APPENDIX B. The format of copyright notices will also change to allow addition of third-party information if required and to comply with HIS-14. Examples of the new notices are included in APPENDIX C. The examples are shown without third-party information since that is the most prevalent form.

4. Proposed procedure for copyright of source software.

If the software is a modified version of existing code, there is only one question to be answered. That is, will the copyright notice require new or changed third-party information? The installer will have the information on the MSCR form to make this determination at installation time. If no change is necessary, then the installation is performed without needing any of the copyright tools. If the code is new, then the copyright tools must be used to insert a proper notice into the code.

5. Proposed procedure for copyright of object and executable software.

This proposed method for protection of object and executable code makes use of a new term, herein called a "primary bound segment". There will be a primary bound segment for each PSP when more than

one bound segment exists within a PSP. The primary bound segment will contain as its first component a data structure containing the copyright (or trade secret) notice and STI information. For example, fortran contains bound_fort_ and bound_fortran_io_. Only this bound segment (ie, bound_fort_) will be required to have a copyright notice as its first component. In a similar manner, the source and object archives corresponding to the primary bound segment will have the data structure as their first components. The exceptions to this are MCM, hardcore, and BOS software, as were discussed earlier. Responsibility for the determination of the primary bound segment is that of Multics Project management. The guide used to determine the primary segment will be content of user-visible interfaces, ie. bound_cobol front_ contains the "cobol" command. APPENDIX A contains a cross reference for PSP products and their proposed primary bound segments. The data structures will be generated by a new command, generate_copyright, which will be used by installers. This command will build an ALM macro whose source will go into the 'primary' source archive as the first component, and whose object will go into the 'primary' object archive. This archive will be bound and ordered such that the data structure will be the first component in it. The proposed naming format for the data structures is:

"<marketing_id>_copyright_notice".

for the modules requiring copyright protection. If a trade secret notice is to be used, the naming convention proposed is:

"<marketing_id>_trade_secret_notice".

The ALM macro (for copyrights) should look something like this:

```
name    <MID>_copyright_notice
segdef  <MID>_copyright_notice
dec    1                                " version 1 structure
dec    55                               " 55 character notice
dec    1                                " No. of software tech. ids
aci    "Copyright, (C) Honeywell Information Systems Inc., 1980"
aci    "W1MSSM090000"                  " STI
end
```

If the source copyright notice was updated, the primary bound segment copyright must also change. The name of each copyright notice must be retained so that it, and the STI (or STIs) are accessible/printable by command (see MTB-492). The copyright notice must become the first ASCII data encountered, so that if the primary bound segment is dumped, the copyright (or trade secret) notice is seen first. An example is shown below:

```
dump_segment >udd>m>jms>bound_copyright_ 0 20 -ascii -nad
```

```
103157160171 162151147150 164054040050 103051040110 Copyright, (C) H  
157156145171 167145154154 040111156146 157162155141 oneywell Informa  
164151157156 040123171163 164145155163 040111156143 tion Systems Inc  
056073040061 071070060056 400000000000 141144144137 .; 1980.....add_
```

There will be only one type/format copyright notice for object code. It will not have delimiters as the notices in source code must have. Otherwise, it is identical to source copyrights. Likewise, the trade secret notice contains the text of the notice inserted into source code, but without asterisks and delimiters.

APPENDIX A

MULTICS PSP PRODUCTS

Multics Priced Software Products

PRODUCT NUMBER	TITLE	PROPOSED PRIMARY BOUND SEGMENT
SGS6800	Multics Operating System EXEC	See Section F.
<u>OPERATING SYSTEM SOFTWARE</u>		
SGS6801	GCOS (III) Timesharing Environment Facility	bound_gcoss_tss_
SGS6802	Transaction Processing Tools	bound_tp_runtime_
SGS6803	FAST/DFAST (Fast Access System for Timesharing)	bound fast_
SGS6804	GCOS (III) Batch Environment Facility	bound_gcoss_
<u>OPERATING SYSTEM SOFTWARE EXTENSIONS</u>		
SGE6800	Multics System Software Extensions	bound_copyright_
SGE6802	Remote Job Entry Facility	bound_g115_
<u>COMMUNICATIONS SOFTWARE</u>		
SGC6800	Multics Communications System (Multics CS)	See Section F.
SGC6801	Autocall Support Option to Multics CS	See Section F.
SGC6802	3270 Support Option to Multics CS	See Section F.
SGC6803	Basic Bisync Support Option to Multics CS	See Section F.
SGC6804	G115 Support Option to Multics CS	See Section F.
SGC6805	File Transfer Facility	See section F.
SGC6806	Network Tools	See Section F.
<u>PROGRAMMING LANGUAGE SOFTWARE</u>		
SGL6801	FORTTRAN Compiler and Runtime Facility	bound_fort_
SGL6802	Basic Compiler and Runtime Facility	bound_basic_
SGL6803	COBOL-74 Compiler and Runtime Facility	bound_cobol_front_
SGL6805	MRPG (Report Generator) Facility	bound mrpg_
SGL6806	APL (Version 2)	bound_v2apl

OPERATING SYSTEM UTILITIES AND TOOLS

SGU6800	WORDPRO Comprehensive Facility	Protected via other PSP's
SGU6801	SORT/MERGE Facility	bound_sort_
SGU6803	LISTER Facility	bound_lister_
SGU6804	SPEEDTYPE Facility	bound_speedtype_
SGU6805	Dictionary Tools	bound_word_commands_
SGU6807	Extended Mail Facility	bound_mail_system_
SGU6820	Compose Facility	bound_compose_
SGU6833	TED (Text Editor) Facility	bound_ted_
SGU6834	Emacs Text Processing Facility	bound_multics_emacs_
SGU6835	Offline Page Printing System Support Facility	bound_pps_support_

DATA MANAGEMENT SOFTWARE

SGD6800	MRDS Facility	bound_mrds_
SGD6801	LINUS	bound linus_

APPLICATIONS SOFTWARE

AGS6801	Timesharing Library	bound_tss_fortran_
AGS6802	ISTAT	bound_istat_
AGS6803	Graphics Facility	bound_graphics_system_

MR9 SOFTWARE

SGU6821	Mergenthaler VIP Device Support Option to Compose
SGE6801	DPS 8/70 M Extension Feature
* TBD	DSA Session Control
* TBD	Unified File Transfer Facility
* TBD	Multics Comm. HASP Support
* TBD	Multics Menu Facility
* TBD	Format Document Facility
* TBD	Executive Mail Facility
* TBD	Level-6 Attached Processor

* indicates Marketing ID yet to be determined.

APPENDIX B

COPYRIGHT TOOL FOR SOURCE

```
-----
add_copyright
-----
```

SYNTAX AS A COMMAND:

```
add_copyright path {control_args}
```

FUNCTION:

Adds a copyright notice to a source program. Different notices are used for each different language type suffix, ie. pl1, fortran, alm, ec, etc. The date in the copyright notice will be the current year. The internal subroutine `copyright_notice_` is invoked to do the work.

ARGUMENTS:

path

Is the pathname of the source segment to be modified.

CONTROL ARGUMENTS:

-first_party, -fp <STR>

where STR is the name of the primary party holding the copyright. If this argument is not given, the default is "Honeywell Information Systems Inc., <year>", where <year> will be the current 4-digit year. Maximum length of STR may be 35 characters, and if blanks occur in STR, then it must be enclosed in quotes.

-third_party, -tp <STR>

where STR is the name of a secondary party holding an interest in the copyright. If this argument is not given, the default is a null string. Maximum length of STR may be 30 characters, and if blanks occur in STR, then it must be enclosed in quotes. If given, STR is prefaced by " and ".

-trade_secret, -ts

this control argument is to be used if the notice to be inserted is the Trade Secret notice. Use of this control argument conflicts with the previous two.

APPENDIX C

NEW SOURCE PROTECTION NOTICES

COPYRIGHT NOTICE

```

*****
*
*
*   Copyright, (C) Honeywell Information Systems Inc., 1980
*
*
*****

```

TRADE SECRET NOTICE

```

*****
*
*
*   HONEYWELL CONFIDENTIAL AND PROPRIETARY
*
*
*****

```

Appropriate source code delimiters will be inserted in front and back of the notice used in order to make it a comment.

APPENDIX D

COPYRIGHT TOOL FOR OBJECT

```
-----  
generate_copyright  
-----
```

SYNTAX AS A COMMAND:

```
generate_copyright marketing_id
```

FUNCTION:

Generates an ALM macro source segment and object data structure to allow protection of object and executable code in a Multics Priced Software Product (PSP). The data structure will contain a software protection notice and at least one Software Technical Identifier.

ARGUMENTS:

marketing_id

Is the PSP number (e.g. SGE6800). The command will prompt the user, requesting information on the protection notice to be used, and its contents if it is a copyright notice. The user will also be prompted for STI information. The user must specify at least one STI (as specified in MTB-492).