MULTICS TECHNICAL BULLETIN MTB-447

To: MTB Distribution

From: F.W. Martinson
Date: April 10, 1980

Subject: Packaging of Separately Priced Software

INTRODUCTION

In the near future we will be required to repackage Multics to include more Priced Software Product (PSP) software. This will require modification of existing tools and design of new tools for generating tapes to be distributed to customer sites.

This proposal describes methods necessary to accomplish this repackaging.

Since the trend in the industry today is marketing of software as well as hardware this concept is very important. Large groups of the software will become PSP software. A preliminary list will be included with this MTB. The proposal to be described is relatively simple and should not require much software modification or design.

NEW SYSTEM LIBRARIES

It is proposed that new system libraries be created to contain the full current software release. These new directories will exactly mirror existing libraries in structure. These libraries will not be changed and will be used for software distribution while still providing the capability to update standard hardcore and online libraries as we traditionally have in the past. Only current release software will be retained online. As new general releases are frozen the previous release software will be dumped to tape and deleted. The libraries to be created are:

```
>MR8.0>system_library_standard
>MR8.0>system_library_tools
>MR8.0>system_library_unbundled
>MR8.0>system_library_network
>MR8.0>system_library_obsolete
>MR8.0>system_library_tandd
>MR8.0>library_dir_dir
>MR8.0>documentation
```

These libraries will be maintained on a demountable private logical volume. This will provide added security by providing added control over access to these libraries and will free up a minimum of two disk drives by requiring they only be mounted when actually creating release tapes for sites.

TAPE GENERATION METHODOLOGY

The default acl for all segments that are PSP software in these directories will be "null *.SysDaemon". By manipulating acls on these segments, the proper distribution of PSP software to sites will be insured.

A database describing each PSP and maintained by lister will be used to generate exec_coms for manipulating acls prior to generating tapes for sites. For example, if site X is to get only fortran, the generated exec_com would look something like:

```
&command_line off
&go_to &ec_name
&label before_dump
sa >MR8.0>ldd>unb>lists>bound_fort_.list r *.SysDaemon
sa >MR8.0>ldd>unb>lists>bound_fortran_.list r *.SysDaemon
sa >MR8.0>ldd>unb>lists>bound_fortran_io_.list r *.SysDaemon
sa >MR8.0>ldd>unb>o>bound_fort_.1.archive r *.SysDaemon
sa >MR8.0>ldd>unb>o>bound_fort_.2.archive r *.SysDaemon
sa >MR8.0>ldd>unb>o>bound_fortran_.1.archive r *.SysDaemon
sa >MR8.0>ldd>unb>o>bound_fortran_.2.archive r *.SysDaemon
sa >MR8.0>ldd>unb>o>bound_fortran_io_.archive r *.SysDaemon
sa >MR8.0>ldd>unb>o>close_file r *.SysDaemon
sa >MR8.0>ldd>unb>o>convert_numeric_file r *.SysDaemon
sa >MR8.0>ldd>unb>o>fortran_error_messages_ r *.SysDaemon
sa >MR8.0>ldd>unb>o>general_format_parse_ r *.SysDaemon
sa >MR8.0>ldd>unb>s>bound_fort_.1.s.archive r *.SysDaemon
sa >MR8.0>ldd>unb>s>bound_fort_.2.s.archive r *.SysDaemon
sa >MR8.0>ldd>unb>s>bound_fortran_.1.s.archive r *.SysDaemon
sa >MR8.0>ldd>unb>s>bound_fortran_.2.s.archive r *.SysDaemon
sa >MR8.0>ldd>unb>s>bound_fortran_io_.s.archive r *.SysDaemon
sa >MR8.0>ldd>unb>s>close_file.pl1 r *.SysDaemon
sa >MR8.0>ldd>unb>s>convert_numeric_file.pl1 r *.SysDaemon
sa >MR8.0>ldd>unb>s>fortran_error_messages_.message r *.SysDaemon
sa >MR8.0>ldd>unb>s>general_format_parse_.pl1 r *.SysDaemon
sa >MR8.0>unb>bound_fort_ r *.SysDaemon
sa >MR8.0>unb>bound_fortran_ r *.SysDaemon
sa >MR8.0>unb>bound_fortran_io_ r *.SysDaemon
sa >MR8.0>unb>close_file r *.SysDaemon
sa >MR8.0>unb>convert_numeric_file r *.SysDaemon
sa >MR8.0>unb>fortran_error_messages_ r *.SysDaemon
sa >MR8.0>unb>general_format_parse_ r *.SysDaemon
&quit
&label after_dump
```

```
sa >ldd>unb>lists>bound_fort_.list null *.SysDaemon
sa >ldd>unb>lists>bound_fortran_.list null *.SysDaemon
sa >ldd>unb>lists>bound_fortran_io_.list null *.SysDaemon
sa >ldd>unb>o>bound_fort_.1.archive null *.SysDaemon
sa >ldd>unb>o>bound_fort_.2.archive null *.SysDaemon
sa >ldd>unb>o>bound_fortran_.1.archive null *.SysDaemon
sa >ldd>unb>o>bound_fortran_.2.archive null *.SysDaemon
sa >ldd>unb>o>bound_fortran_io_.archive null *.SysDaemon
sa >ldd>unb>o>close_file null *.SysDaemon
sa >ldd>unb>o>convert_numeric_file null *.SysDaemon
sa >ldd>unb>o>fortran_error_messages_ null *.SysDaemon
sa >ldd>unb>o>general_format_parse_ null *.SysDaemon
sa >ldd>unb>s>bound_fort_.1.s.archive null *.SysDaemon
sa >ldd>unb>s>bound_fort_.2.s.archive null *.SysDaemon
sa >ldd>unb>s>bound_fortran_.1.s.archive null *.SysDaemon
sa >ldd>unb>s>bound_fortran_.2.s.archive null *.SysDaemon
sa >ldd>unb>s>bound_fortran_io_.s.archive null *.SysDaemon
sa >ldd>unb>s>close_file.pl1 null *.SysDaemon
sa >ldd>unb>s>convert_numeric_file.pl1 null *.SysDaemon
sa >ldd>unb>s>fortran_error_messages_.message null *.SysDaemon
sa >ldd>unb>s>general_format_parse_.pl1 null *.SysDaemon
sa >unb>bound_fort_ null *.SysDaemon
sa >unb>bound_fortran_ null *.SysDaemon
sa >unb>bound_fortran_io_ null *.SysDaemon
sa >unb>close_file null *.SysDaemon
sa >unb>convert_numeric_file null *.SysDaemon
sa >unb>fortran_error_messages_ null *.SysDaemon
sa >unb>general_format_parse_ null *.SysDaemon
&quit
```

Dumping of libraries following execution of the before_dump exec_com will dump only those products for which SysDaemon has access. Others will be recorded in an errorfile providing hardcopy documentation of software not sent to a site. Hardcopy documentation of software on a particular tape is provided in the normal dump map listing.

Following tape generation the after_dump exec_com will be executed, restoring the acls back to their default state of "null *.SysDaemon".

TAPE RELOADING METHODOLOGY

Everything described to this point can be provided under current software. The only changes, such as creation of new directories, are mechanical in nature and require no software modification. The tape reloading phase of this plan will require some minor changes to the hierarchy reloader. A capability to provide cross reloading, as cross retrieval is now provided, will be necessary. In addition removal of the equal level cross reload/retrieval restriction will be required. This restriction has already been removed from the carry facility. We must have the capability to cross reload from a lower level directory to a higher level directory. For example:

```
     >MR8.0>unb>***=>unb
```

must be permitted.

Finally, in order to complete this software packaging proposal the implementation of a "-new_release" control argument to the Initializer ring-1 commands "reload" and "reload_new_release" will be required. The function of this new control argument will be to search the hardcore database active_hardcore_data for a reload control file containing reload pathnames as shown in the example above. This control argument will also cause, in the case of "reload_system_release" only, all segments in the target directory to be deleted prior to starting of the reload. The addition of this information to the end of active_hardcore_data should be as follows:

```
     dcl 1 ahd aligned based (active_hardcore_datap),
        .
        .
        .
     2 n_reload_release fixed bin, /* number of pathnames */
     2 reload_release (32), /* allow up to 32 */
     3 name char (96); /* pathnames, some can be quite long */
```

Pathnames will be initialized as other variables in this database. For example:

```
ahd.reload_release(1).name = ">MR8.0>unb>***=>unb"
```

Both the "reload" and "reload_system_release" commands will continue to function as they do today when the "-new_release" control argument is not given.

When the "-new_release" control argument is given to the "reload" command, cross reloading will occur according to pathnames specified in active_hardcore_data. The feature deleting all segments in the target directory being reloaded will be turned off. Normal trimming will still occur as today.

This reloading philosophy will provide two important capabilities. First it will provide a method of unbundling many parts of the system and of maintaining an online static, frozen version of the current software release. Second, it will provide the beginnings of guaranteed syncronization of hardcore and online libraries. The use of the "-new_release" control argument will allow reloading only of those libraries described in active_hardcore_data.

SUMMARY

1. Create mirror images of system libraries in a directory directly off the root containing full system release software. This directory is named for the system release.

2. Generate release tapes only from these libraries. This will permit updating of standard libraries as in the past and will provide a means of controlling distribution of software under control of existing ACL mechanisms.

3. Load system releases at sites under control of system tape distributed with the release by specifying directories to be reloaded in a segment contained on the tape.

4. Insure PSP software not distributed with a new release, but existing at a site prior to the release, is deleted prior to reloading the new release.

APPENDIX A

MKTG ID	Product
SGS6800	Multics Operating System EXEC
SGS6801	GCOS (III) Timesharing Environment Facility
SGS6802	Transaction Processing Tools
SGS6803	FAST/DFAST (Fast Access System for Timesharing)
SGS6804	GCOS (III) Batch Environment Facility
SGX6802	Remote Job Entry Facility
SGE6800	Multics System Software Extensions
SGC6800	Multics Communications System (Multics CS)
SGC6801	Autocall Support Option to Multics CS
SGC6802	3270 Support Option to Multics CS
SGC6803	Basic Bisync Support Option to Multics CS
SGC6804	G115 Support Option to Multics CS
SGC6805	File Transfer Facility
SGC6806	Network Tools
SGL6801	FORTRAN Compiler and Runtime Facility
SGL6802	Basic Compiler and Runtime Facility
SGL6803	COBOL-74 Compiler and Runtime Facility
SGL6805	MRPG (Report Generator) Facility
SGL6806	APL (Version 2)
SGU6803	LISTER Facility
SGU6804	SPEEDTYPE Facility
SGU6805	Dictionary Tools
SGU6807	Extended Mail Facility
SGU6820	Compose Facility
SGU6821	Device Table Compiler Option to Compose
SGU6833	Multics Ted
SGU6834	Emacs (Editor Macros)
SGU6835	Multics Off-Line PPS
SGD6800	MRDS (Multics Relational Data Store) Facility
SGD6801	LINUS (Logical Inquiry and Update System)
AGS6801	Timesharing Library
AGS6802	ISTAT
AGT6801	Graphics Facility
SGU6807	Extended Mail