

TO: MSPM Distribution
FROM: Karolyn Martin
DATE: July 7, 1969
SUBJECT: Standards for Library Procedures, BY.0

Section BY.0 is being re-issued to reflect Multics policy changes in the past two years.

1. Restricted EPL is to be used for all library coding.
2. Rules for initiating segments now include using the new file system facilities.
3. Error handling is no longer done by signalling.

Published: 07/10/69
(Supersedes: BY.0 01/06/67)

Identification

Standards for System Library Procedures
K.J. Martin

Purpose

System library procedures are designed to be used by the casual user of Multics. Many are called by system commands. Consequently library procedures must be easy to use and very reliable.

Rules

The following rules are to be followed when writing system library procedures.

1. They must be pure procedures.
2. They must return to the calling procedure under all circumstances.
3. They must be well-commented. They should be written in a high-level language. Explicit permission is required to use any language except EPL or PL/I. They must be written in a straightforward way. At the top of each procedure there must be a statement of its purpose, a description of the calling sequence, name of the original author and date, name of the author of the last modification and date. These rules are necessary because system library procedures go from hand to hand for improvement and modification.
4. All library procedures should use restricted EPL (REPL defined in BB.2.01) as far as possible. Whenever a user interface is not involved, REPL should be strictly adhered to. User interfaces may (often do) require the use of otherwise forbidden argument types, such as character strings.
5. Library procedures should not use write-arounds; they should call the desired primitive unless a very significant simplification results from use of a write-around.
6. Library procedures must be careful to leave data bases in a consistent state. In particular, they should set the bit count on an entry after changing the length of the associated segment. They should also release any free storage they utilize.

7. In general, they should not issue I/O attach calls but should use standard stream names. (Refer to section BF.1.01 for a discussion of stream names.)
8. Every effort must be made to do thorough validity checking on arguments. Incorrect arguments should not cause unexpected, random-type behavior. Any default conditions must be clearly stated in the writeup of the library procedure.
9. They should use other system library procedures when possible. In addition, a function which is needed by a library procedure and which seems generally useful should be coded separately so that it can be called by others.
10. Library procedures should not initiate segments by their entry name in order to avoid name conflicts in the process. They should use a unique call name only when a name is needed to refer to the segment; in general, they should initiate by null call name and make all subsequent accesses by pointer. Library procedures should terminate the segments they initiate.
11. They should handle errors as defined in BD.8.00. Every attempt should be made to let the highest-level procedure possible decide on the seriousness of errors and what to do about an error.
12. They must check all system options pertinent to their task, and should interpret them in a standard way. For example, they must check the brief option to determine whether or not all comments are wanted. (See BX.12.00 BX.12.02 on options).
13. When several related tasks are performed by a library procedure, multiple entry points are preferable to one entry point with a convoluted argument list. An easy-to-remember call results in fewer caller errors than a complex one.
14. All library procedures must be documented in a full MSPM section and in an abstract as described in BS.0.