

MP/M IITM Operating System

PROGRAMMER'S GUIDE

MP/M II V2.1 Compatibility Attributes

Addendum #1 to the First Printing - 1981

Copyright © 1982 by Digital Research, Inc.

CP/M is a registered trademark of Digital Research

MP/M and MP/M II are trademarks of Digital Research

Compiled February 4, 1982

The MP/M IITM file system introduced some new restrictions relating to file operations that were not present in MP/MTM 1.1 or CP/M[®]. For example, if a process opens a file in the default mode (locked), MP/M II does not allow other processes on the system to open, delete, or rename the file until the process opening the file either closes the file or terminates. In addition, MP/M II does not allow a process to perform file operations with an FCB that has not been activated by a successful open or make operation, or with an FCB that has been deactivated by a close operation. These restrictions protect an MP/M II user from interference from other users on his open files. To illustrate, it is this protection that enables an MP/M II user to edit a file with the assurance that another user cannot delete or modify his file during his edit session.

The new restrictions added to MP/M II were required to provide file security when multiple users are running the system. The above example describes restrictions required to prevent collisions on file activity between independent processes. Another new MP/M II restriction sets limits on how a process can modify open FCBs. These limits are enforced by checksum verification of open FCBs and protect the integrity of the MP/M II file system from corrupted FCBs. Note that the new MP/M II restrictions are not intended to protect a user from his own actions. Instead, they ensure that the activity of one user does not adversely affect other users on the system.

In general, the new MP/M II file system restrictions create little difficulty for new application development. In fact, they enforce what is generally accepted as good programming practice. However, because of these new restrictions, some CP/M and MP/M software written prior to MP/M II's release does not run on MP/M II. In addition, multiple copies of some software do not run because the default open mode for MP/M II is a locked mode in which only one process can open a file.

To address these problems, Digital Research has added compatibility attributes to MP/M II, Version 2.1. The compatibility attributes are defined as attributes F1' through F4' of program

All Information Presented Here is Proprietary to Digital Research

files. A new GENSYS option determines whether the attributes are to be activated. If activated, the Command Line Interpreter (CLI) interrogates these attributes during program loading and modifies the MP/M II ground rules for the loaded program as described below. Note that the compatibility attributes should not be used with new software. They are intended for use with working software developed for CP/M and MP/M 1.1. This especially applies to compatibility attribute F4' which disables FCB checksum verification on read and write operations. Use this attribute sparingly and only with programs that are known to work.

COMPATIBILITY ATTRIBUTE DEFINITIONS

- F1'** MP/M 1.1 Default Open. Processes running with this attribute have all files opened in locked mode marked as Read/Only in the System Lock List. This allows all processes with this attribute set to read and write to common files with no restrictions. There is, however, no record locking provided. In addition, this attribute also allows a process to write to a file opened by another process in Read/Only mode. To be safe, all static files such as program and help files should be made Read/Only when this compatibility attribute is used.
- F2'** Partial Close default. Processes running with this attribute have their default close mode changed from permanent close to partial close. This attribute is intended for programs that close a file to update the directory but continue to use the file. Note that MP/M II assumes a process has finished with a file when the number of closes issued to the file equals the number of opens. A side effect of this attribute is that files opened by a process are not released until the process terminates. It might be necessary to set the System Lock List parameters to high values when using this attribute.
- F3'** Ignore Close Checksum Errors. This attribute changes the way Close Checksum errors are handled for a process. Normally, a message is printed on the console and the process is terminated. When this attribute is set and a checksum error is detected during a close operation, the file is closed if a lock list item exists for the file. Otherwise, an unsuccessful close error code is returned to the calling process.
- F4'** Disable FCB Checksum verification for read and write operations. Setting this attribute also sets attributes F2' and F3'. This attribute should be used carefully because it effectively disables MP/M II's file security. Use this attribute only with software that is known to work.

PROCEDURE FOR USING THE COMPATIBILITY ATTRIBUTES

- 1) Answer yes to the GENSYS question "Enable Compatibility Attributes (N) ?".
- 2) Use the MP/M II Utility SET to set the desired combination of compatibility attributes in the program name.

EXAMPLES:

```
0A>SET filespec [F1=on]
0A>SET filespec [F1=on,F3=on]
0A>SET filespec [F4=on]
```

If you have a program that runs under CP/M or MP/M 1.1 but does not run properly under MP/M II, use the following guidelines to select the compatibility attributes to set for the program.

- 1) If the program terminates with the message, "File Currently Opened" when multiple copies of the program are run, set compatibility attribute F1'. As an alternative, you might consider placing all common static files under User 0 with the SYS and R/O attributes set.
- 2) If the program terminates with the message, "Close Checksum Error", set compatibility attribute F3'.
- 3) If the program terminates with an I/O error, try running the program with attribute F2' set. If the problem still occurs, try attributes F2' and F3'. If the problem still persists, then try attribute F4'. Use attribute F4' only as a last resort.

It might be necessary to increase the GENSYS parameters that set the maximum number of files a process can open and the size of the System Lock List when using compatibility attributes F2' and F4'. This might be required because both default to partial closes. As a result, system lock list entries consumed by opening files are not released until the process terminates. In general, if a process terminates with the message "No Room in System Lock List" or "Open File Limit Exceeded", it usually indicates that the above GENSYS parameters need to be increased. Another option is to patch in a BDOS Free Drive call at a point in the program where no files are active. Note that a Free Drive call specifying all drives, purges all file system lock entries belonging to the calling process.

When GENSYS activates compatibility attributes, the Command Line Interpreter copies the settings for attributes F1' through F4' of the filename of the loaded program into byte 1DH of the process descriptor as shown below:

PROCESS DESCRIPTOR BYTE 1DH

(Bits defined 7-0 high order to low order)

Bit 7 set = F1
Bit 6 set = F2
Bit 5 set = F3
Bit 4 set = F4

MP/M IITM Operating System

PROGRAMMER'S GUIDE

MP/M II V2.1 Extended File Locking

Addendum #2 to the First Printing - 1981
Copyright © 1982 by Digital Research, Inc.
MP/M II is a trademark of Digital Research
Compiled January 15, 1982

Extended file locking is a new facility implemented in release 2.1 of MP/M IITM. Extended file locking enables a process to maintain a lock on a file even after the file is closed. This facility allows a process to rename, set the attributes, or delete a file without having to contend with the possibility of interference from other processes after the file is closed. In addition, a process can reopen a file with an extended lock and continue normal file processing. For example, a process can open a file, perform file operations on the file, close the file, rename the file, reopen the file under its new name, and proceed with file operations, without ever losing the file's lock list item and control over the file.

Extended file locking is only available to files that are opened in the default open mode (locked mode). To extend a file's lock, set interface attribute F6' when closing the file. Note that this attribute is only interrogated by the Close function when it is closing a file permanently. Thus, interface attribute F5' must be reset when the close call is made. In addition, if a file has been opened N times (more than once), this attribute is only interrogated when the file is closed for the Nth time.

To maintain an extended file lock through a Rename File call or a Set File Attributes call, set interface attribute F5' of the referenced FCB when making the call. Note that this attribute is only honored for extended file locks, not normal locks. Setting attribute F5' also maintains an extended file lock for the Delete File function, but setting this attribute also changes the nature of the Delete function to an XFCB-Only delete. If successful, all three of these functions delete a file's extended lock item when with attribute F5' reset. On the other hand, if they return with an error code, the extended lock item is not deleted.

A standard open call can be made to resume file operations on a file with an extended lock. The open mode, however, is restricted to the default locked mode. The following list illustrates uses of extended locks.

All Information Presented Here is Proprietary to Digital Research

- Open file EXLOCK.TST in locked mode
- Perform file operations on the file EXLOCK.TST using the open FCB.
- Close file EXLOCK.TST with interface attribute F6' set to retain the file's lock item.
- Use the Rename File function to change the name of the file to EXLOCK.NEW with interface attribute F5' set to retain the file's extended lock item.
- Open the file EXLOCK.NEW in locked mode.
- Perform file operations on the file EXLOCK.NEW using the opened FCB.
- Close file EXLOCK.NEW with interface attribute F6' set to retain the file's lock item.
- Set the Read/Only attribute and release the file's lock item by using the Set File Attributes function with interface attribute F5' reset. At this point, the file EXLOCK.NEW becomes available for access by another process.

MP/M IITM Operating System

PROGRAMMER'S GUIDE

MP/M II V2.1 Programming Guidelines

Addendum #3 to the First Printing - 1981
Copyright © 1982 by Digital Research, Inc.
CP/M is a registered trademark of Digital Research
MP/M and MP/M II are trademarks of Digital Research
Compiled January 15, 1982

This guideline provides additional discussion on the information presented in the MP/M IITM Programmer's Guide. In particular, this document emphasizes those areas of MP/M II where restrictions exist that did not exist in version 1 of MP/MTM or versions 1 and 2 of CP/M[®]. The intent is to enable the MP/M II application programmer to avoid potential problems with new software. As a prerequisite, the reader should be familiar with the material presented in the MP/M II Programmer's Guide.

- 1) Always use the following sequence when performing file operations that require an open file. Under MP/M II, these operations are the BDOS read, write, lock and unlock record commands.
 - Activate a file's FCB with a BDOS Open or Make function call before using the FCB in a file operation. Verify that the Open or Make operation was successful. MP/M II only accepts FCBs activated by a successful Open or Make call for open file operations. If an FCB that has not been activated is used, MP/M II returns a checksum error.
 - Perform all file operations using activated FCBs. Note that MP/M II does not deactivate an activated FCB when it returns error codes for file operations. In general, only the current record and random record fields of an activated FCB should be modified. In addition, all file operations with an activated FCB must be made under the user number that was in effect when the FCB was activated. A similar restriction applies to activated FCBs that specify the default drive. All file operations specifying such an FCB must be made under the current drive that was in effect when the FCB was activated. The complete rules regarding activated FCB modification are covered in item 3.

All Information Presented Here is Proprietary to Digital Research

- If a process has completed file operations on a file but still has a significant amount of processing left to do, the file should be closed. This applies even if the file was not modified. With some exceptions (See Section 2.2.9 of the MP/M II Programmer's Guide), the lock list entry associated with a file in the system lock list is not released until a file is permanently closed. MP/M II restricts access to a file by other processes while a lock list item for the file resides in the system lock list. It is not necessary to close input files if a process is about to terminate. At termination, all lock items belonging to a process are released. Output files, however, must always be closed or data may be lost. Note that a successful permanent close operation deactivates the FCB and removes the file's item from the system lock list. If the deactivated FCB is used in a subsequent open file operation, MP/M II returns a checksum error.
- 2) If a process opens the same file more than once, a matching number of close commands must be issued to the file to remove the file's lock list item from the system lock list. Thus, if a file has been opened N times, the first N-1 close operations issued to the file default to partial close operations. Only the last close, close operation N, is interpreted as a permanent close. By definition, a permanent close is a close operation that removes the referenced file's item from the system lock list. Note that only one lock list item is allocated in the system lock list for a file regardless of the number of FCBs a process has opened for the file.
 - 3) The following list specifies how an activated FCB can be changed without affecting the FCB checksum. MP/M II returns a checksum error code and does not perform the requested operation if an FCB with an invalid checksum is used in an open file operation.
 - FCB(0) cannot specify a new drive.
 - With the exception of interface attributes F5' and F6' for the BDOS Close function, FCB(1) through FCB(11) cannot be changed.
 - The high order 3 bits of FCB(12) cannot be changed. The low order 5 bits can be changed. Note that when a file is opened in the default open mode (locked mode), the high order 3 bits of this FCB field are set to zeros.
 - FCB(13) cannot be changed.
 - FCB(14) and FCB(15) can be changed.
 - FCB(16) through FCB(31) cannot be changed.

- o FCB(32) through FCB(35) can be changed.

If compatibility with future releases of MP/M and CP/M is a requirement, programs should restrict open FCB modification to the FCB fields 32 through 35. In particular, Digital Research does not support techniques that involve modifying fields 12, 14 and 15 of open FCBs.

- 4) Processes that access a printer must issue a Detach List device to free the printer before another process can use the printer. If the Detach List call is not made, a process that accesses a printer continues to own the printer until it terminates.
- 5) CP/M programs that create submit files for chaining must be modified to work under MP/M II. MP/M II requires a different filename for submit files, which includes the originating console number, and requires that a submit flag be set in the System Data Page. The technique for creating and executing submit files is described in MP/M II Application Note 07. Note that MP/M II also has a Program Chain (Function 47) command that provides an efficient mechanism for program chaining.
- 6) CP/M programs that make direct BIOS calls for disk I/O do not work under MP/M II. MP/M II does support direct XIOS calls for the console and printer, but not to the disk. If programs must make direct XIOS disk calls, a technique **strongly** discouraged in a multi-user environment, two levels of indirection must be used to obtain the real XIOS jump table address. The second level of indirection is required because an intercept table handles the console and printer.

The following two steps should be performed in a program before making direct XIOS calls to a disk. The first step is to make a BDOS Write Protect Disk (Function 28) call to the disk to ensure that no other process has open files on the disk. Secondly, the MXDisk mutual exclusion queue message should be read to prevent other programs from making BDOS disk function calls while your program is making direct XIOS calls. After completing your direct XIOS calls, write back the MXDisk message and then reset the drives you have set to Read/Only.

- 7) The following procedure is a protocol that multiple processes can use to coordinate record update and addition operations to a shared file. Each process must open the shared file in unlocked mode. This procedure also assumes that records containing binary zeros are null records.

- Attempt to lock the record.
 - If the lock attempt fails because another process has locked the record, delay and repeat the procedure.
 - If the lock attempt fails because the record does not exist in the file, add a record initialized to binary zeros to the file with the BDOS Write Random with Zero Fill command and repeat the procedure. Note that files opened in unlocked mode are extended in block units and not in record units as is the case for files opened in the default locked mode.
 - If the lock attempt succeeds, read the record, update it, and then unlock it.
- 8) Multiple FCB I/O is a technique that involves opening each extent for a file independently and maintaining them in a table in memory. Then random I/O is handled by selecting the proper FCB from the table, setting the current record field to the proper record number within the extent, and making a sequential Read or Write command. When processing is completed, each FCB is closed. The maximum file size that can be accessed with this technique is 512K bytes. This limits the maximum table size to 32 FCBs. Note that this technique provides a method of performing random I/O that is compatible with CP/M 1.4.

Multiple FCB I/O has to be performed carefully under MP/M II because of the restrictions MP/M II places on file operations to provide file security. In general, an FCB should not be used in file I/O unless it has been activated and it should not be modified while it is activated (see items 1 and 3). In addition, the number of opens and closes issued to a file is important (see item 2). Note that all 32 bytes of each extent's FCB should be maintained in the open FCB table. In addition, verify that interface attribute F8' is set to 1 in all FCBs if the first FCB has F8' set to 1. F8' set to 1 indicates the file was opened under user 0 although the current user number is non-zero (see Function 15 in the MP/M II Programmer's Guide).