

# Accessing Different Files

If the MFS needs to update a file other than the current file, it is best to use BASIC+ file I/O commands ([OPEN](#), [READ](#), [WRITE](#), [DELETE](#)) rather than a direct call. If a direct call is used, the MFS will pass a call to the next filing system (MFS or BFS) for the current file. This may cause problems in two ways.

- First, the MFS that follows the current MFS may be installed for only the current file, and not for other files. A direct call might thus inadvertently "install" the next MFS onto the second file being accessed.
- Another problem is the converse of the first. By doing a direct call using the MFS list for the current file, an MFS may bypass the MFS list for the second file. For example, if an MFS uses a direct call to access a file that is indexed, SI.MFS will not be called unless by chance it appears in the BFS argument for the current file.
- A corollary to this is that the second file may use an entirely different BFS. Obviously, a direct call from the MFS can only use the BFS for the current file.

Some applications might present a hybrid of these problems. An MFS might use BASIC+ commands to access a second file, only to discover that the second file also uses the current MFS -- thereby resulting in a call to itself.