

Direct Calls to the MFS

Certain MFS codes do not have a corresponding BASIC+ or System Monitor command to call them. Examples include [RECORD.COUNT](#) and [INSTALL](#).

If a program requires access to these codes, it must execute a direct call to the first MFS for the file to be accessed. This is done as an external subroutine call to the first MFS for the file, passing the seven filing system arguments.

Under most circumstances, the program should not do a direct call to the BFS itself. There are two reasons for this. First, a direct call assumes that the programmer knows in advance what BFS will be used to access the file. This makes the program unusable for files accessed through any other BFS. Second, a direct call to the BFS bypasses calls to any MFS that might be installed for the file. While this might be desirable under certain limited conditions, in general it is a much better practice to access a file via the full sequence of MFSs. This helps preserve the file integrity represented by those MFSs. For example, direct access to the file via the BFS will defeat indexing, security, or audit trail MFSs that may be essential to an application.

The MFS list for a file is available in two places. The first is in the FILES file. The MFS list for a file is stored as the fourth field in the FILES entry for a file. The list is kept as a list delimited with @VM. In order to make use of the list, the value marks must be converted to @SVMs.

The handle returned by the BASIC+ [OPEN statement](#) also contains the MFS list for a file. In this situation, the MFS list is the first value (delimited with @VM) in the handle, and the true file handle is the second value. The MFS list returned as part of the BASIC+ OPEN file handle is already @SVM-delimited, so no conversion is required.

The following sample program illustrates how to execute a direct call to the first MFS for a file. The example program calls the filing system with a code of **25 (RECORD.COUNT)** to return the total number of records in the file.

```
DECLARE SUBROUTINE MSG , FMSG
EQU RECORD.COUNT$ TO 28
RESP = 'CUSTOMERS'

OPEN RESP TO FILE THEN
  FS.LIST = FILE<1,1>
  HANDLE = FILE<1,2>
  NEXT.FS = FS.LIST<1,1,1>
  NAME = "" ; FMC = 0 ; RECORD = "" ; STATUS = 0
  CALL @NEXT.FS(RECORD.COUNT$,FS.LIST,HANDLE,NAME,FMC,RECORD,STATUS)
  IF STATUS THEN
    MSG(@window, FMC:" record(s) counted.")
  END ELSE
    MSG(@window, "Can't count records in ":RESP)
  END
END ELSE
  FMSG()
END
```