

# Modifying MFS Arguments

The arguments for filing system calls come from the various elements of BASIC+ statements. A programmer can thus pass information or keys to the MFS by simply passing them along with data.

One example is the [Create\\_Table command](#). All parameters following the filename are passed intact to the filing system for its own use. A developer could pass any number of parameters to an MFS by simply adding them onto the ones already being passed. The MFS would examine the parameter string, extract its own parameters, and pass the remaining ones on for further processing.

Programmers can even use BASIC+ [READ](#) and [WRITE](#) statements to pass information, using the key or record variables. As in the last example, the programmer adds the MFS-specific information into the key or record. The MFS strips the information for its own use, and passes clean data on for further processing.

An example is an encryption MFS that relies on an encryption key. The developer can concatenate the encryption key onto the record key using a specific delimiter, then simply read or write the data. The MFS can strip the encryption key from the record key, and then pass the key on to the filing systems that follow.

Because this requires that the encryption key be passed with the record, it provides a much higher level of security than a single-key encryption MFS. The developer can create windows in which a pre-read and pre-write process tacks a window-specific encryption key onto the record key. This limits access to the data to that window.

While this method is very useful, it also requires caution. As a rule, if an MFS is modifying basic data (records and keys), then the position of the MFS in the BFS list is important. For instance, it will be critical to install the MFS correctly with respect to [SI.MFS](#), so that data is indexed properly -- not with control information in it.

Programmers should also consider the multiple ways that data can be accessed, and assure themselves that users cannot access or update files in such a way that the MFS is skipped or is confused about data conversion it must do. For instance, if an MFS has been coded to convert data during reads and writes, the developer must guarantee that it will never be called with data that is not to be converted.