

Kunden Service

Bischoff Analysetechnik GmbH

Optichrom Advance

Technische Information

BASIC INTERPRETER FILE MOVE SYNTAX DIAGRAM

10/11/95

Syntax

The file move functions to be added to the BASIC interpreter and the table editor will conform to the following syntax:

```
MOVEF sloop, sunit, sslot, "sstring", dloop, dunit, dslot,  
      "dstring" [,mode[,V1[,V2...[,V10]]]...]
```

Where:

sloop is the source device loop number.

sunit is the source device unit number.

sslot is the source device slot number.

sstring is a variable format string describing the path and filename, within the source device, of the file data to be transferred. The format of the string is as above (in dstring).

dloop is the destination device loop number.

dunit is the destination device unit number.

dslot is the destination device slot number.

dstring is a variable format string describing the path and filename, within the destination device, which will receive the transferred file data. The format of the string is defined by the destination device operating system. The BASIC interpreter will scan this string for possible variable substitution options within the string. These options are discussed in detail later in this document.

mode is an optional destination file open mode. Legal values are "A" for append, "N" for new or "O" for overwrite. The default is "O". The "N" (new) mode will return an error if the destination file already exists.

V1-V10 are an optional list of variables which are used by the BASIC interpreter to build the dstring or sstring via the variable substitution options within the strings. The variable substitution options are discussed below.

Variable Substitutions

Variable substitution options may be entered in either the dstring or sstring to facilitate "on the fly" filename and path generation. Variable substitutions are used when some part of the filename or path will vary with each execution of an Event.

Example:

Assume: C(13) = "ANLZ1", C(21) = "A", C(23) = "BCDEFG", J = 1, and K=1234

Then the following Advance BASIC statement:

```
MOVEF 0, 0, 39, "C:\FTIR\SPEC?4.?3", 1, 1, 1, "C:\?8\SPEC?4.?3", "O",C(23), K
,C(13), C(21), J
```

Will yield the following dstring and sstring after substitution:

```
sstring = "C:\FTIR\SPECBCDE.234"
dstring = "C:\ANLZ1\SPEC.A.001"
```

Details:

A variable substitution consists of a question mark and a number (e.g. ?4). The question mark signals the BASIC interpreter to look at the end of the statement for an optional variable (V1...V10). The number is the field size.

The first question mark encountered will correspond to the first optional variable in the list (V1). The second corresponds to the second variable (V2) and so on. Up to 10 question marks and optional variables may be entered in one MOVEF statement. The number of question marks, **MUST** match the number of optional variables or syntax or run-time error will occur.

NOTE: The current Portable Service Panel Applications Overlay does not have a question mark ("?") character. To enter this character, press **ALT-HEX-3F**.

If an actual question mark is required in a string, use two question marks (??). This will cause the formatter to insert a question mark into the string, without using an optional variable.

The field size is required. It can vary from 0 to 9. A value of zero (?0) means that NO field fitting or justification is required. A value of one (?1) means that one and only one character will be substituted. Values from two (?2) to nine (?9) will be treated similarly, nine being the largest field size allowed.

Variable types:

Integers: If the variable is an integer (I, J, K, etc.), the absolute value will be used. The field will be right justified and zero filled. If the value is too large to fit the field, it will be truncated on the left, thus truncating the most significant digits of the value (see examples below).

Strings: If the variable is an alpha-numeric string (such as some control table entries) the field will be left justified. If the string is greater than the field size, the string will be truncated on the right (see examples below). If it is smaller than the field size, it will be left as is, and will NOT be padded to fit the field size. Any ASCII string containing **only** characters from "0" to "9" will be treated as an integer.

Floating point numbers: Floating point variables **WILL** be treated as strings. If it desired to use one as a number, it should be truncated and assigned to an integer variable BEFORE calling MOVEF.

Examples for a "?8" format:

DBA-Type	value	Assumed type	MOVEF Substitution
String	"SPECTROMETER"	String	"SPECTROM"
String	"AX1"	String	"AX1"
String	"0001"	Integer	"00000001"
Floating point	"123.5"	String	"123.5"
Integer	12	Integer	"00000012"
String	"123456789A"	String	"12345678"
String	"123456789"	Integer	"23456789"

Destination File Modes

The user may wish to specify how the destination file is opened. The mode options allowed to the user are:

- "N" for a new file,
- "O" for overwriting an existing file,
- "A" for appending information to an existing file.

"N" - this mode allows the user to specify that if the file exists, do NOT overwrite it. An error will be returned via the M variable if the file exists. If the file does not exist then create the file and begin the data transfer.

"O" - this mode is the default mode. It specifies create the file if it does not exist OR overwrite the file if it does exist.

"A" - this mode allows the user to append data to an existing file, instead of overwriting it. The file will be append to if it exists, or will be created if the file if it does not exist.

Example:

"A" - append mode:

```
MOVEF 0, 0, 39, "C:\FTIR\SPEC?4.?3", 1, 1, 1, "C:\SPEC.TXT", "A", C(13), J
```

Example:

Default mode "O": (notice the empty comma after txt" and before c(13)

```
MOVEF 0, 0, 39, "C:\FTIR\SPEC?4.?3", 1, 1, 1, "C:\SPEC.TXT", , C(13), J
```

Effect on Event Execution

After parsing and building the source and destination strings, the BASIC interpreter will call the file transfer system. The file transfer system will return a status code via the "M" integer variable.

Upon calling MOVEF the parent event will be suspended from further execution. Execution will be resumed upon completion of the file move, or upon time-out.

M Variable Return Codes:

The status codes returned will be defined in a later document.

Special File Move Functions

The MOVEF statement gives the user the ability to transfer portions of ACB2 memory which can be thought of as logical files. Each of these logical files has a fixed name and an address slot associated with it to be used to specify it in the MOVEF command..

Below is a list of Logical Files:

NAME	Address Slot	Description
CR1 thru CR5	1	Chromatogram tables 1 thru 5
DT1 thru DT31	2	Data tables 1 thru 31
TABLESET	3	Analyzer's Table Set
EEPROM	4	Analyzer's Firmware (reprogram EEPROMS)

Logical File Syntax

The Syntax varies depending on the type of logical file.

For Chromatogram Records:

"CRx[:start_point[:end_point]][\format_name]

Where:

CRx= CR1 through CR5, Chrom records.
start_point= Chrom data point to start with.[optional]
end_point= Chrom data point to end with.[optional]
format name= Name of non-default format to change data to. [optional]
Formats available:
Default: Standard Chrom record file (IBM PC format).
/b - Transfer data in raw binary format instead of
Standard (not recommended)

For Data Table Records:

"DTx[:start_point[:end_point]][\format_name]

Where:

DTx= DT1 through DT31, Data tables
start_point= Data Table record to start with.[optional]
end_point= Data Table record to end with.[optional]
format_name= Name of non-default format to change data
to.[optional]
Formats available:
Default: Data only ASCII format. Field
Delimiter = ","
/t - Add Time/Date Stamp and status to
report.
/b - Transfer data in binary format
instead of ASCII
/dx - Make the Field Delimiter = x.

Examples of File Move Syntax for Chromatogram records and Data Tables:

Example 1:

Move Chromatogram record 1 to the file c:\anlz1.cr1 on the APC at Loop 1, Unit 1, Slot 1.

MOVEF 0, 0, 1, "CR1", 1, 1, 1, "C:\ANLZ1.CR1"

The entire chrom record would be moved. The resulting file would be exactly the same as from the Save Chrom record option in Service Panel Emulation. The mode defaults to mode "O", overwrite.

Example 2:

Move chromatogram record 1, data points 1 through 50 to the file C:\ANLZ1.CR1

MOVEF 0, 0, 1, "CR1:1:50", 1, 1, 1, "C:\ANLZ1.CR1"

Example 3:

Move chromatogram record 1, starting with data point 50 thru the end of the chrom record, to the file C:\ANLZ1.CR1

```
MOVEF 0, 0, 1, "CR1:50", 1, 1, 1, "C:\ANLZ1.CR1", ,
```

Example 3:

Move the data table for stream 31, data points 500 through 65530 to the file C:\DATA.D31

```
MOVEF 0, 0, 2, "DT31:500:65530/td ", 1, 1, 1, "C:\DATA.D31", ,
```

The resulting file would include all value field data, along with that data point's time, date, and status. Each field would be **delimited by a space** ("d ") and NOT the default comma.