

Extension Entry

Extension functions are linked in by having a descriptor which, as with the keywords, must be in page 0. In this case, the BASIC system call LINKSYM is called with a pointer to the first byte of the entry. BASIC then calculates the hash number and links the entry into the built-in part of the symbol table. As with the keywords, the actual execution routine may be in a segment other than page 0.

The symbol table entry required for BASIC to link it in is as follows:

2 bytes	- used by BASIC in linking in.
1 byte	- flags
n bytes	- name text
2 bytes	- page 1 execution address
1 byte	- segment number of code (0 if in page 0)

For an extension function, the flags byte is always 8 (numeric) or 9 (string). See section 2.3 (Symbol Table) for more details.

The name text is the name of the function in upper case ASCII, preceded by the number of characters in the name. String function name text must end in a 'S'.

The function code, when called, has not had any parameters evaluated. To do this, it must perform the appropriate system calls such as P_NUM (see section 6 (System Calls) for more details).

A single result must always be left on BASIC's stack, the type of result matching the type of the function, of course.

Extension functions may use the X and Y floating point variables (see section 5 (Variables) for more details) but note that some of the transcendental functions that can be called as a system call will also use these. See section 6 (System Calls) for more details.