

DESCRIPTION OF NEW OPERATIONS TO CREATE CAPABILITIES OF SPECIFIED TYPE. WILL MODIFY USER MANUAL, PAGE 12 OR SO.

H. Create a Capability Creating Authorization

IP1 D: C-list index for returned authorization

A capability creating authorization is a special ~~form~~ type of capability. The second word of the capability contains the type of capability which may be manufactured under the authorization.

Possible errors while creating an authorization:

<u>Class</u>	<u>#</u>	<u>Description</u>
8	2	No more capability types are available*

I. Create a Capability of the Authorized Type

IP1 D: C-list index for returned capability

IP2 C: A capability creating authorization

IP3 D: Data for second word of returned capability

A capability of the type specified by IP2 ~~and~~, with all option bits on, and with second word equal to IP3, is returned at the specified index in the caller's C-list.

Only the entry/exit errors are possible.

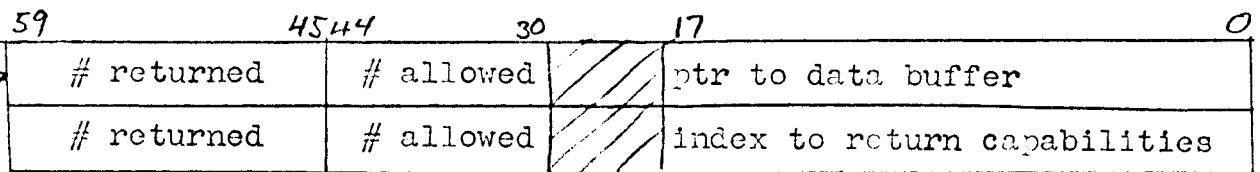
*Note that since there are only about 48000 different capability types, unrestricted use of this operation would allow one user to exhaust the supply, thus making those that wanted a special capability type later on very unhappy.

10 Aug 70

NEW FORM OF SYSTEM CALL. ALTERS PAGE 4 OR SO OF USER MANUAL.



When BIT = 0, no return of data or capabilities is authorized.
When BIT = 1, the word ~~fix~~ following the CEJ must contain
a pointer to a return authorization* as follows:



The # returned fields are both set by the system to the actual
number of data (caps) returned.

* USED WHEN THE RETURN IS VIA A "RETURN WITH
PARAMETERS" OPERATION; SEE ~P. 34.

DESCRIPTION OF NEW (AND OLD) PARAMETER TYPES. WILL GO ON PAGE 2 OR SO OF USER MANUAL.

Each entry in the input parameter (IP) list is one of the following types:

1) type D - a 60-bit data item

59		0
data		

capability

2) type C - a ~~capability~~ ~~index~~ specifier
direct

59	58		18	17	0
0	/ / / / /			index 1	

indirect

59		47		30		17	0
1	/	index 2		/	index 1		

the direct form specifies the capability at index 1 in the local C-list.

the indirect form requires that index 1 be the index ~~of~~ in the full C-list of a capability for a C-list; the specified capability is at index2 in the C-list given by index 1.

3) type BD - a block data specifier

47		30		17	0		
/ / / / /		num		/ / / / /		ptr	

num 60-bit data items starting at ptr are transferred.

4) type BC - a block capability specifier

47		30		17	0		
/ / / / /		num		/ / / / /		index	

num capabilities starting at index are transferred.

10 Aug 70

HOW TO SET UP BLOCK DATA AND BLOCK CAPABILITY PARAMETER TYPES
IN AN OPERATION. PAGE 57 OR SO OF USER MANUAL.

6. CHANGE PARAMETER FROM "NONE" to "block data"

IP1 C: Capability for an operation (OB.CHTYP)
IP2 D: ~~xxx~~ Index of parameter to be changed
IP3 D: Maximum size of block that will be passed

Sorry, no error list right now.

7. Change parameter from "none" to "block capability"

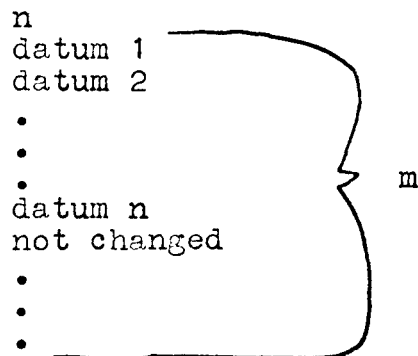
IP1 C: Capability for an operation (OB.CHTYP)
IP2 D: Index of parameter to be changed
IP3 D: Maximum number of capabilities that will be passed

Sorry, no error list right now.

DESCRIPTION OF HOW BLOCK DATA AND BLOCK CAPABILITY PARAMETERS LAND IN THE ADDRESS SPACE OF THE CALLED SUBPROCESS. PAGE 33 OR SO OF THE USER MANUAL NEEDS LOTS OF HELP IN THIS REGARD.

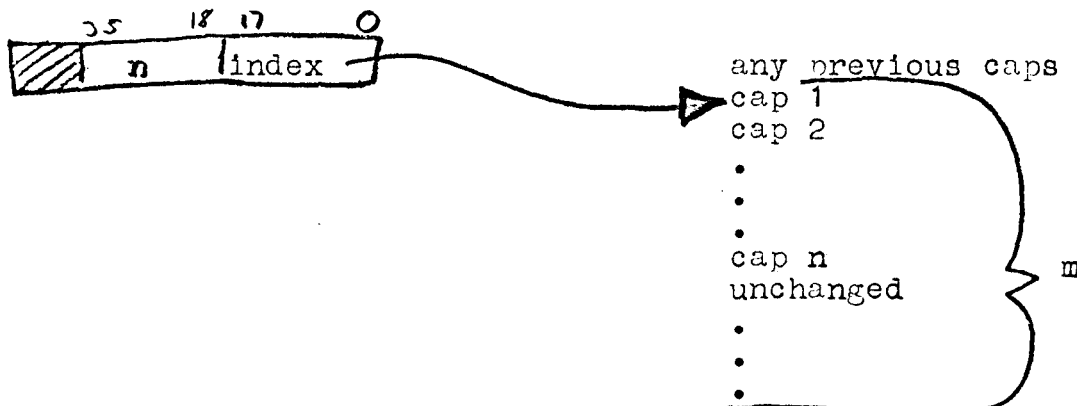
Block data

Let m be the ~~xxxxxxx~~ maximum number of data that can be passed, as specified by the operation. Then $m+1$ cells are reserved for passing the block of data (the cells immediately follow any previous information passed to the subprocess as part of the call (or start at cell 6 if there isn't any previous info)). The actual number of data passed is set in the first of the $m+1$ words, immediately followed by the n data in order. Unused cells at the end are unchanged when n is less than m .



Block capabilities

Let m be the maximum number that can be passed, as specified by the operation. Then m slots in the full C-list are reserved for passing the block of capabilities (the slots immediately follow any previous capabilities passed, or start at 0 if there were none) and the next available cell of the address space is reserved for an indicator. The number of capabilities actually passed, n , and the index of the first one are stored in the indicator. The n capabilities are copied into the C-list starting at the indicated index. When n is less than m , unused C-list slots are unchanged.

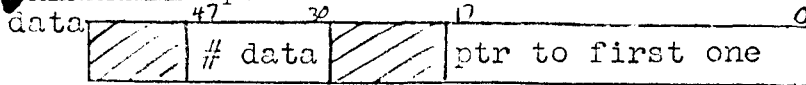


DESCRIPTION OF RETURN WITH PARAMETERS OPERATION. GOES ON PAGE 34 OR 30 OF USER MANUAL.

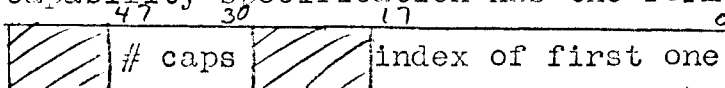
M. Return with Parameters

IP1 D: ~~XXXXXXXXXXXXXXXXXXXX~~ Data specification
 IP2 D: Capability specification

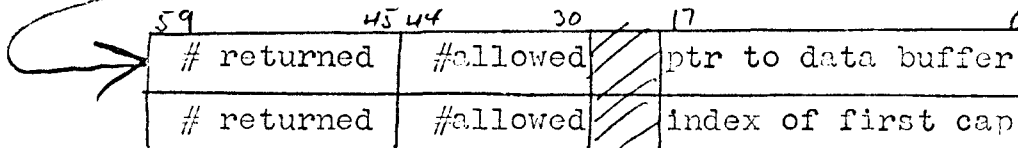
The ~~parameter~~ specification has the form



and the capability specification has the form



Data from the full address space and capabilities from the full C-list of the returning subprocess are returned to the full address space and full C-list of the subprocess determined by the top of the stack. Provided that all pointers, indices, counts, etc. in both subprocesses are legal. In particular, the P-counter in the stack must point to a CEJ which indicates a return authorization as described on page 4. Just so you won't have to turn the page, the PTR after the CEJ points to



The # returned fields are set by the system to the actual number returned; the other fields are prescribed by the subprocess which points to them and are not disturbed.

The error list is not in yet, but trying to return more data or caps than the authorization allows is not an error; the operation merely returns the max number and says nothing. If the CEJ doesn't specify a return authorization, that isn't an error either; needless to say, nothing is returned in that case.