

## CPTR\$ & CPTR0\$ - Perform Pointer Arithmetic

The CPTR\$ routine is used to perform arithmetic (addition and subtraction) on 32-bit pointers.

The entry-point CPTR0\$, which is functionally identical to CPTR\$, is reserved for use by GSM System DLMs.

**Important Note:** The CPTR\$ sub-routine should be used to perform all arithmetic on 32-bit pointers. CPTR\$ supports all the 32-bit pointer formats, including some that are currently reserved for future use.

### 1. Invocation

To perform arithmetic on a pointer code:

```
CALL CPTR$ USING ptr1 length ptr2
```

where *ptr1* is the PIC PTR source pointer; *length* is a PIC 9(4) COMP, or literal, containing the number of bytes to add to the pointer, *ptr2* is the PIC PTR destination pointer.

### 2. STOP Codes and Exception Conditions

No STOP codes are generated by CPTR\$.

No exceptions are returned by CPTR\$.

### 3. Programming Notes

The current 32-bit pointer format is:

77	PTR1	PIC PTR	
01	FILLER REDEFINES PTR1		
02	PAGE	PIC 9(4) COMP	* Page number
02	OFFSET	PIC 9(4) COMP	* Offset address

However, other pointer formats are reserved for future use.

CPTR\$ can be used to update a pointer variable or return the results to a different pointer variable. For example:

```
CALL CPTR$ USING P1 LEN P1          * ADD LEN TO P1
CALL CPTR$ USING P1 LEN P2          * ADD LEN TO P1 GIVING P2
```

If the value in length is strictly negative, CPTR\$ will subtract the value from a pointer. For example:

```
CALL CPTR$ USING P1 -2 P1          * SUBTRACT 2 FROM P1
```

### 4. Examples

### 5. Copy-Books

None.

### 6. See Also

None.