READL$ - Read Long Operation

The standard 32-bit READ verb only supports a 16-bit (64Kb) offset address. Both the FREEX$ and FREE2$ routines allow a dynamic memory page larger than 64Kb to be allocated. The READL$ sub-routine can be used to read directly into a memory page that is larger than 64Kb.

1. Invocation

To perform a Long Read operation code

```
CALL READL$ USING fd file_address page_number offset_address length
```

where *fd* is an FD of any organization, **which must be open before calling READL$**; *file_address* is a PIC 9(9) COMP field, **not a literal**, that specifies the offset of the first byte in the file to be read; *page_number* is a PIC 9(4) COMP field that contains the number of the dynamic memory page; *offset_address* is a PIC 9(9) COMP field, **not a literal**, that contains the offset address within the memory page; and length is a PIC 9(9) COMP field, **not a literal**, that contains the length of the read operation.

2. STOP Codes and Exception Conditions

No STOP codes are generated by READL$.

The following EXIT codes may be returned by READL$:

EXIT code	$$COND	Description
401	1	An irrecoverable I/O error has occurred.
402	2	A file boundary violation has occurred.

3. Programming Notes

READL$ is only available with GSM SP-41, or later; and GLOBAL.EXE V5.9, or later.

The open File Definition (FD) passed to READL$ can be of any organization but it only makes sense to used READL$ with either the Relative Sequential (RSAM) or Basic Direct (BDAM) Access Method. It is not recommended to mix Access Method functions (e.g. READ, READ NEXT, WRITE etc.) with READL$ calls on the same open FD.

The *page_number* passed to READL$ is normally the first 2 bytes of the 32-bit pointer FMPTR returned by the FREEX$ or FREE2$ routine. For example:

```
01      FM                          * FREEX$ CONTROL BLOCK
  02    FMOPC  PIC 9 COMP
  02    FMSIZE PIC 9(6) COMP
  02    FMPTR  PIC PTR
  02    FMNAME  PIC X(8)
01 FILLER REDEFINES FMPTR
  02 FMPAGE PIC 9(4) COMP
  02 FMOFFS PIC 9(4) COMP

01      F2                          * FREE2$ CONTROL BLOCK
```

```
    02    F2OPC  PIC 9 COMP
    02    F2SIZE PIC 9(9) COMP
    02    F2PTR  PIC PTR
    02    F2NAME  PIC X(8)
  01 FILLER REDEFINES F2PTR
    02 F2PAGE PIC 9(4) COMP
    02 F2OFFS PIC 9(4) COMP
```

Note that the size of the offset address, FMOFFS, in the FM block definition is a 2-byte PIC 9(4) COMP field. This is sufficiently large for FREEX$ and FREE2$, both of which return a 0 in this field. However, the maximum size of a memory block created by FREEX$ is 8,388,607 (or 2,147,483,647 for FREE2$) both of which far exceed the capacity of a PIC 9(4) COMP field. Consequently, the offset address passed to READL$ is a PIC 9(9) COMP field.

4. Examples

[EXAMPLES REQUIRED]

5. Copy-Books

None.

6. See Also

FREEX$ Allocate 32-bit Memory
FREE2$ Allocate 32-bit Memory