

# FREEEX\$ - Allocate 32-bit Memory

The FREEEX\$ routine can be used to allocate a Temporary 32-bit Data Page. All Data Pages allocated by FREEEX\$ will be automatically de-allocated when the next STOP RUN occurs.

## 1. Invocation

To allocate a 32-bit Data Page code:

```
CALL FREEEX$ USING fm
```

where fm is a control block of the following format:

```

01   FM
02   FMFUN          PIC 9 COMP      * Function required
                                       * 0 = Get work space (fixed name)
                                       * 1 = Free work space (fixed name)
                                       * 2 = Get Work space (variable name)
                                       *   (GSM SP-11, and later)
                                       * 3 = Free Work space (variable name)
                                       *   (GSM SP-11, and later)
02   FMSIZE         PIC 9(6) COMP   * Size of work space required, or 0
02   FMPTR          PIC PTR        * Pointer to the 1st byte allocated
02   FMNAME         PIC X(8)       * Page name (only if FMFUN = 2 or 3)
                                       *   (GSM SP-11, and later)
02   FMESIZ        PIC 9(9) COMP   * Extended size, if FMSIZE zero
                                       *   (GSM SP-36, and later)

```

To de-allocate a 32-bit Data Page code:

```
CALL FREEEX$
```

The zero-parameter call of FREEEX\$ is equivalent to a call with FMFUN=1.

## 2. STOP Codes and Exception Conditions

The following STOP codes may be generated by FREEEX\$:

STOP code	Description
3603	The value of FMFUN is invalid (i.e. not in the range 0 to 3).
3604	The value of FMSIZE is invalid (i.e. not in the range 1 to 8,388,607).
3605	The value of FMESIZ is invalid (i.e. not in the range 1 to 2,147,483,647).

The following EXIT codes may be returned by FREEEX\$:

EXIT code	\$\$COND	Description
-----------	----------	-------------

3601	1	Insufficient memory or Page Table entries to allocate the 32-bit Data Page.
------	---	---

### 3. Programming Notes

The 32-bit FREEX\$ sub-routine is **broadly** compatible with the 16-bit equivalent. The main differences are:

- The format of the FMSIZE field has been extended from PIC 9(4) COMP to PIC 9(6) COMP. For GSM SP-36, and later, this has been further extended to PIC 9(9) COMP by the addition of FMESIZ;
- The size of the FMPTR PIC PTR has been extended from 16-bits to 32-bits;
- For functions 2 and 3 **only**, the FM block has been extended by the FMNAME PIC X(8) field.

For functions 0 and 1, the name of the 32-bit Data Page is set to \$\$FREE\$. The default name may be overridden by replacing functions 0 and 1 with 2 and 3, respectively. The option to override the default Page Name is only available with GSM Service Pack 11 (GSM SP-11), and later.

When allocating a Data Page (i.e. FMFUN = 0 or 2), if a Data Page of the same name is already present, it will be extended by allocating a Linked Data Page. **All** Linked Data Pages are de-allocated when any one page is de-allocated.

The Data Page allocated by FREEX\$ are not pre-initialised. **Do not assume the contents of any freshly allocated Data Pages.**

Each Data Page allocated by FREEX\$ will be contained within a separate 32-bit page. **DO NOT ASSUME THAT TWO CALLS TO FREEX\$ WILL ALLOCATE CONSECUTIVE MEMORY BLOCKS.**

Any attempt to allocate memory outside the block allocated by FREEX\$ will normally crash the Global Client.

The following fields in the 32-bit System Area may make it unnecessary to use FREEX\$ to create Data Pages under some circumstances:

\$\$AR32	PIC X(256)	* 256-byte supplement to 16-bit PIC X(16) \$\$AREA
\$\$3000	PIC X(1024)	* 1024-byte work area reserved for use by
		* Global-3000

Some highly-specialised 16-bit code may assume that the data established in 16-bit Free Memory (i.e. after a call to FREE\$) is preserved after a subsequent de-allocate and re-allocate. While this assumption may be true, **under some conditions**, for 16-bit FREE\$ it should **never** be assumed for 32-bit FREEX\$.

### 4. Examples

[EXAMPLES REQUIRED]

### 5. Copy-Books

See copy-book "f\$" in copy-library S.SYS32. Note that this copy-book **MUST** be expanded using a SUBSTITUTING clause. For example:

COPY "f\$" USING "FM"

## 6. See Also

SDATA\$     Allocate Temporary 32-bit Data Page  
XDATA\$     Extended SDATA\$  
UNLO\$      De-allocate Data Page  
FREE2\$     Allocate 32-bit Memory