

# MENUX\$ - Extended Menu Display

The MENUX\$ routine is used to display various menu styles.

## 1. Invocation

To display a menu using MENUX\$ code:

```
CALL MENUX$ USING id style [me]
```

where *id* is a PIC X(2) unique identifier for the menu; *style* is a PIC (4) COMP field, or literal, that defines the menu style and *me* is the Menu Control Block.

The *style* parameter can be:

- 1 button menus
- 2 window menus
- N all other values are reserved for future use

The *me* Menu Control Block is defined as follows:

```
01 ME
02 MEVERS PIC 9(2) COMP * Block version
   VALUE 1 * must be set to 1
02 MELINE PIC 9(4) COMP * Position of top line of
   * menu or 0 for centred
02 MECOL PIC 9(4) COMP * Position of first col
   * of menu 0 for centred
02 MEDISP PIC 9 COMP * 1 Display only otherwise 0
02 MEFUNC OCCURS 4 PIC 9 COMP * Functions flags for F1-F4
   * If set to 1 then the
   * function will be processes
02 METIME PIC 9(3,1) COMP * time out on menu in
   * mm.s (s in tenths of minutes)
02 MEOPT PIC 9(2) COMP * 1-27 Returned menu
   * function number
   * 0 escape or exit line selected
   * -1 another type of return selected
02 MERFUNC PIC 9(4) COMP * If MEOPT = -1 this will contain
   * 1-4 Function 1-4 selected
   * 900 time out
02 METXL PIC 9(2) COMP * Accept text length
02 METEXT PIC X(100) * Any returned text
02 MEDIS PIC X(?) * Display lines of menu
```

A call without the *me* parameter will clear the current menu.

## 2. STOP Codes and Exception Conditions

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

STOP code	Description
19903	MENUX\$ has been called by an application that is not running on GX.

19904	Invalid style parameter (only values 1 or 2 are currently supported)
19905	An attempt has been made to call MENUX\$ on an incompatible version of GX. The version of GX must be V2.9, or later
19906	The total length of the GX command block has exceeded an internal limit.

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

EXIT code	\$\$COND	Description
19904	4	Unable to allocate memory for temporary work buffer

### 3. Programming Notes

MENUX\$ is only available when running on GX. Any attempt to use MENUX\$ on a non-GX terminal will result in a STOP code. The version of GX must be V2.9 or later. The version of GSM must be GSM SP-9, or later.

The Display Lines of the menu (MEDIS) contain a set of variable text strings, each with a terminating #00, in the following order:

- Application Title;
- Menu Title;
- Function Text;
- Menu lines (to a max of 127);
- #00 terminator.

The title and function strings may be empty and in this case only a #00 byte is required.

Each menu line should be defined as follows:

*a*-*sssstext* for a menu line that is enabled

or:

*a*\**sssstext* for a line which is disabled

where *a* is the accelerator character; *ssss* are four space characters.

#### 3.1 Menu Style 1

Menu style 1 is similar to the style of menus displayed by \$MENU32. In this style menus are automatically cleared so a call without the *me* parameter will be ignored.

### **MELINE, MECOL**

These fields are not required - the position of the menu on the screen is ignored.

### **MEDISP, METXL, METEXT**

These fields are not required - this menu style has no extra accept line and does not support the "display-only" mode.

### **METIME**

This field may be set and will cause the menu to time-out if nothing is keyed in the specified time.

### **MEOPT**

On return from MENUX\$ this will either be set to the function number selected; to 0 if escape or the menu exit line is selected; or -1 if a time-out has occurred.

### **MERFUNC**

On return from MENUX\$, if MEOPT is set to -1 then this field will be set to 900 indicating that the menu has timed out without the user keying a reply.

### **MEDIS**

These are the menu lines and should be set up as described above. The function text line will be ignored although the empty function text string **must** be present. The last menu line will be assumed to be the exit line and 0 will be returned in MEOPT if this line is selected.

## **3.2 Menu Style 2**

This style of menu has a look which is similar to a pop menu style. But also optionally allows a base line accept text to be entered.

### **MELINE, MECOL**

These fields specify the position of the top left hand corner of the menu on the screen. Setting either to 0 centres the menu with respect to the depth or width of the window.

### **MEDISP**

This is the "display only" flag. If this field is set to 1 the menu will be "display only" and nothing will be accepted; otherwise this field should be set to 0.

### **MEFUNC**

If MEDISP is 0, then you may want the function keys 1-4 to be an acceptable reply to the menu. If you do then the function flag should be set to 1, if not they must be set to 0.

### **METIME**

This should be set if you wish the menu to time-out if nothing is keyed in the indicated time.

### **MEOPT**

On return from MENUX\$ in accept mode, this field will either hold the function number selected, 0 if escape has been keyed or -1 if another function has been returned.

**MERFUNC**

On return from MENUX\$, in accept mode, if MEOPT is set to -1 then this field will contain the returned function which will either be 1-4 , for functions 1-4; or 900 if the menu has timed out.

**METXL**

If MEDISP should be set to 0 if a baseline is **not** required; otherwise METXL must be set to the length of the baseline accept to a maximum of 100.

The following combinations of MEDISP and METXTL are possible:

MEDISP	METXL	Comments
0	0	Normal Menu Accept (i.e. no Baseline Accept)
0	1	Normal Menu Accept <b>AND</b> Baseline Accept
1	0	Display-only menu
1	1	Display-only menu

**METEXT**

If MEDISP is set to 0, then on return from MENUX\$, this field will contain any text keyed to the baseline prompt by the operator.

**MEDIS**

The menu display lines should be set up as above. The function text will be the prompt for the baseline accept.

**4. Examples**

[EXAMPLE REQUIRED]

**5. Copy-Books**

None.

**6. See Also**

MH\$            Display "classical" menu