Advanced Message Box Routines

This document describes a number highly-specialised sub-routines that allow a windows Message Box to be created using a number of primitive routines. The routines described in this note allow more control over the message box displayed using the standard MBOX$ and MBOXX$ routines:

- Initialise Advanced Message Box (AMBIN$)
- Add Icon to Advanced Message Box (AMBAI$)
- Add Text to Advanced Message Box (AMBAT$)
- Add Button to Advanced Message Box (AMBAB$)
- Accept Operation on an Advanced Message Box (AMBAC$)

All the following routines are within the DF$001 DLM in the P.$DFDLM library. This library must be linked in explicitly using the LNK option in the $SDL32 dialogue. For example:

```
$A3 COMPILATION OPTION:LNK
$A3 LNK> LOAD-MODULE ID:P.$DFDLM UNIT:$DP
$A3 LNK> LOAD-MODULE ID:<CR>
```

Note that an entry for the DLM library P.$DFDLM must be present in the $$DLM Index File on $$D.

1.   AMBIN$ Initialise Advanced Message Box

The initialise Advanced Message Box call is used to provide the overall structure of the message box.

1.1   Invocation

To initialise an Advanced Message Box code:

```
CALL AMBIN$ USING bin
```

where *bin* is a control block of the following format:

```
01   BIN                              * AMBIN$ CONTROL BLOCK
  02  BINVER    PIC 9(4) COMP    * BLOCK VERSION NUMBER
                VALUE 1          * MUST CONTAIN 1
  02  BINPID    PIC X(8)         * PROGRAM ID
  02  BINWID    PIC X(4)         * WINDOW ID
  02  BINX      PIC 9(4) COMP    * TOP LEFT X-CO-ORD (PIXELS)
  02  BINY      PIC 9(4) COMP    * TOP LEFT Y-CO-ORD (PIXELS)
  02  BINWDT    PIC 9(4) COMP    * WINDOW WIDTH (PIXELS)
  02  BINHEI    PIC 9(4) COMP    * WINDOW HEIGHT (PIXELS)
  02  BINATT    PIC 9(2) COMP    * ATTRIBUTE
  02  BINFLG    PIC 9(4) COMP    * WINDOW CONTROL FLAGS
  02  BINL1     PIC 9(4) COMP    * LENGTH OF CAPTION TEXT
  02  BINP1     PIC PTR          * POINTER TO CAPTION TEXT
  02  BINTFS    PIC 9(2) COMP    * TEXT-FONT POINT SIZE
  02  BINTFY    PIC 9(2) COMP    * TEXT-FONT PRINT STYLE
  02  BINL2     PIC 9(4) COMP    * LENGTH OF TEXT-FONT NAME TEXT
  02  BINP2     PIC PTR          * POINTER TO TEXT-FONT NAME TEXT
  02  BINBFS    PIC 9(2) COMP    * DEFAULT BUTTON-FONT POINT SIZE
  02  BINBFY    PIC 9(2) COMP    * DEFAULT BUTTON-FONT PRINT STYLE
  02  BINL3     PIC 9(4) COMP    * LENGTH OF DEFAULT BUTTON-FONT NAME TEXT
  02  BINP3     PIC PTR          * POINTER TO DEFAULT BUTTON-FONT NAME TEXT
```

1.2   STOP Codes and Exception Conditions

The following STOP codes may be generated by AMBIN$:

STOP code	Description
16401	AMBIN$ has been called on a non GX screen.
16402	The BINVER field does not contain 1.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by AMBIN$:

EXIT code	$$COND	Description
16401	1	An exception was returned by GX.
16402	2	Insufficient memory to allocate a work buffer.

1.3   Programming Notes

AMBIN$ is only available when running on GX. Any attempt to use AMBIN$ on a non-GX terminal will result in a STOP code. The version of GX must be V3.3e, or later. The version of GSM must be GSM SP-15, or later.

AMBIN$ must be called before any of the other Advanced Message Box calls can be made.

1.4   Examples

[EXAMPLE REQUIRED]

1.5   Copy-Books

The "$2" copy-book, in copy-library S.DF, expands the following control blocks:

```
BIN
BAI    See AMBAI$
BAT    See AMBAT$
BAB    See AMBAB$
BAC    See AMBAC$
```

1.6   See Also

AMBAI$ Add icon to Advanced Message Box
AMBAT$ Add text to Advanced Message Box
AMBAB$ Add button to Advanced Message Box
AMBAC$ Accept operation on Advanced Message Box

2. AMBAI$ Add Icon to Advanced Message Box

The add icon call is used to add an icon to an Advanced Message Box. The parameters allow both the identity and position of the icon on the Advanced Message Box to be defined.

2.1 Invocation

To add an icon to an Advanced Message Box code:

```
CALL AMBAI$ USING bai
```

where *bai* is a control block of the following format:

```
01    BAI                            * AMBAI$ CONTROL BLOCK
  02  BAIVER     PIC 9(4) COMP       * BLOCK VERSION NUMBER
                 VALUE 1             * MUST CONTAIN 1
  02  BAIX       PIC 9(4) COMP       * TOP LEFT X-CO-ORD (PIXELS)
  02  BAIY       PIC 9(4) COMP       * TOP LEFT Y-CO-ORD (PIXELS)
  02  BAIWDT     PIC 9(4) COMP       * WINDOW WIDTH (PIXELS)
  02  BAIHEI     PIC 9(4) COMP       * WINDOW HEIGHT (PIXELS)
  02  BAISII     PIC 9(2) COMP       * STANDARD ICON INDEX
  02  BAIVII     PIC 9(4) COMP       * VARIABLE ICON INDEX
```

2.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by AMBAI$:

STOP code	Description
16401	AMBAI$ has been called on a non GX screen.
16402	The BAIVER field does not contain 1.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by AMBAI$:

EXIT code	$$COND	Description
16401	1	An exception was returned by GX.
16402	2	Insufficient memory to allocate a work buffer.

2.3 Programming Notes

AMBAI$ is only available when running on GX. Any attempt to use AMBAI$ on a non-GX terminal will result in a STOP code. The version of GX must be V3.3e, or later. The version of GSM must be GSM SP-15, or later.

2.4 Examples
[EXAMPLE REQUIRED]

2.5 Copy-Books
The "$2" copy-book, in copy-library S.DF, expands the following control blocks:

BIN   See AMBIN$
BAI
BAT   See AMBAT$
BAB   See AMBAB$
BAC   See AMBAC$

2.6 See Also
AMBIN$ Initialise Advanced Message Box
AMBAT$ Add text to Advanced Message Box
AMBAB$ Add button to Advanced Message Box
AMBAC$ Accept operation on Advanced Message Box

3.   AMBAT$ Add Text to Advanced Message Box
The add text call is used to add a text string to an Advanced Message Box. The parameters allow both the position and font of the text, as well as the text-string itself, to be defined.

3.1 Invocation
To add a text string to an Advanced Message Box code:

```
CALL AMBAT$ USING bat
```

where *bat* is a control block of the following format:

```
01    BAT                          * AMBAT$ CONTROL BLOCK
  02    BATVER     PIC 9(4) COMP    * BLOCK VERSION NUMBER
                   VALUE 1          * MUST CONTAIN 1
  02    BATX       PIC 9(4) COMP    * TOP LEFT X-CO-ORD (PIXELS)
  02    BATY       PIC 9(4) COMP    * TOP LEFT Y-CO-ORD (PIXELS)
  02    BATWDT     PIC 9(4) COMP    * TEXT BLOCK WIDTH (PIXELS)
  02    BATHEI     PIC 9(4) COMP    * TEXT BLOCK HEIGHT (PIXELS)
  02    BATATT     PIC 9(2) COMP    * ATTRIBUTE
  02    BATL1      PIC 9(4) COMP    * LENGTH OF TEXT STRING
  02    BATP1      PIC PTR          * POINTER TO TEXT STRING
  02    BATTFS     PIC 9(2) COMP    * TEXT-FONT POINT SIZE
  02    BATTFY     PIC 9(2) COMP    * TEXT-FONT PRINT STYLE
  02    BATL2      PIC 9(4) COMP    * LENGTH OF TEXT-FONT NAME TEXT
  02    BATP2      PIC PTR          * POINTER TO TEXT-FONT NAME TEXT
```

3.2 STOP Codes and Exception Conditions
The following STOP codes may be generated by AMBAT$:

STOP code	Description

16401	AMBAT$ has been called on a non GX screen.
16402	The BATVER field does not contain 1.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by AMBAT$:

EXIT code	$$COND	Description
16401	1	An exception was returned by GX.
16402	2	Insufficient memory to allocate a work buffer.

3.3   Programming Notes

AMBAT$ is only available when running on GX. Any attempt to use AMBAT$ on a non-GX terminal will result in a STOP code. The version of GX must be V3.3e, or later. The version of GSM must be GSM SP-15, or later.

3.4   Examples

[EXAMPLE REQUIRED]

3.5   Copy-Books

The "$2" copy-book, in copy-library S.DF, expands the following control blocks:

      BIN   See AMBIN$
      BAI   See AMBAI$
      BAT
      BAB   See AMBAB$
      BAC   See AMBAC$

3.6   See Also

AMBIN$ Initialise Advanced message box window
AMBAI$ Add icon to Advanced message box window
AMBAB$ Add button to Advanced message box
AMBAC$ Accept operation on Advanced message box

4.   AMBAB$ Add Button to Advanced Message Box

The add button call is used to add a button to an Advanced Message Box.

4.1   Invocation

To add a button to an Advanced Message Box code:

```
        CALL AMBAB$ USING bab
```

where *bab* is a control block of the following format:

```
01    BAB                            * AMBAB$ CONTROL BLOCK
  02  BABVER    PIC 9(4) COMP        * BLOCK VERSION NUMBER
                VALUE 1              * MUST CONTAIN 1
  02  BABX      PIC 9(4) COMP        * TOP LEFT X-CO-ORD (PIXELS)
  02  BABY      PIC 9(4) COMP        * TOP LEFT Y-CO-ORD (PIXELS)
  02  BABWDT    PIC 9(4) COMP        * BUTTON WIDTH (PIXELS)
  02  BABHEI    PIC 9(4) COMP        * BUTTON HEIGHT (PIXELS)
  02  BABBID    PIC 9(4) COMP        * BUTTON-ID (50 TO 99)
  02  BABL1     PIC 9(4) COMP        * LENGTH OF TEXT STRING
  02  BABP1     PIC PTR              * POINTER TO TEXT STRING
  02  BABTFS    PIC 9(2) COMP        * BUTTON-FONT POINT SIZE
  02  BABTFY    PIC 9(2) COMP        * BUTTON-FONT PRINT STYLE
  02  BABL2     PIC 9(4) COMP        * LENGTH OF BUTTON-FONT NAME TEXT
  02  BABP2     PIC PTR              * POINTER TO BUTTON-FONT NAME TEXT
```

4.2   STOP Codes and Exception Conditions

The following STOP codes may be generated by AMBAB$:

STOP code	Description
16401	AMBAB$ has been called on a non GX screen.
16402	The BABVER field does not contain 1.

The following EXIT codes may be returned by AMBAB$:

EXIT code	$$COND	Description
16401	1	An exception was returned by GX.
16402	2	Insufficient memory to allocate a work buffer.

4.3   Programming Notes

AMBAB$ is only available when running on GX. Any attempt to use AMBAB$ on a non-GX terminal will result in a STOP code. The version of GX must be V3.3e, or later. The version of GSM must be GSM SP-15, or later.

The Button-id should normally be in the range 50 to 99 (for compatibility with buttons drawn via the "BTN" option).

The Accelerator character on a button is indicated by prefixing it with a "&" character.

4.4   Examples
[EXAMPLE REQUIRED]

4.5   Copy-Books
The "$2" copy-book, in copy-library S.DF, expands the following control blocks:

BIN   See AMBIN$
BAI   See AMBAI$
BAT   See AMBAT$
BAB
BAC   See AMBAC$

4.6   See Also
AMBIN$ Initialise Advanced message box window
AMBAI$ Add icon to Advanced message box window
AMBAT$ Add text to Advanced message box
AMBAC$ Accept operation on Advanced message box

5.   AMBAC$ Accept Operation on an Advanced Message Box
The accept call is used to accept a button selection from an Advanced Message Box. The accept call finalises the Advanced Message Box contents and displays it on the screen.

5.1   Invocation
To issue an accept on an Advanced Message Box code:

```
CALL AMBAC$ USING bac
```

where *bac* is a control block of the following format:

```
01    BAC                             * AMBAC$ CONTROL BLOCK
  02    BACVER     PIC 9(4) COMP      * BLOCK VERSION NUMBER
                   VALUE 1            * MUST CONTAIN 1
  02    BACDEF     PIC 9(4) COMP      * DEFAULT BUTTON
  02    BAC-ID     PIC 9(4) COMP      * RETURNED BUTTON ID
```

5.2   STOP Codes and Exception Conditions
The following STOP codes may be generated by AMBAC$:

STOP code	Description
16401	AMBAC$ has been called on a non GX screen.
16402	The BACVER field does not contain 1.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by AMBAC$:

EXIT code	$$COND	Description
16401	1	An exception was returned by GX.
16402	2	Insufficient memory to allocate a work buffer.

5.3   Programming Notes

AMBAC$ is only available when running on GX. Any attempt to use AMBAC$ on a non-GX terminal will result in a STOP code. The version of GX must be V3.3e, or later. The version of GSM must be GSM SP-15, or later.

The id of the button selected by the operator is returned in BAC-ID and will normally be in the range 50 to 99. If the operator keys <ESC>, or closes the Advanced Message Box by clicking on the Close Box, a value of 11 will be returned in BAC-ID.

To display a typical Advanced Message Box:

```
CALL AMBIN$ USING BIN        * Call Initialise routine once only
[CALL AMBAI$ USING BAI]      * Call Add Icon 0, 1 or N times
CALL AMBAT$ USING BAT        * Call Add Text 1 or N times
CALL AMBAB$ USING BAB        * Call Add Button 1 or N times
CALL AMBAC$ USING BAC        * Call Accept routine once only
```

Completion of the accept operation will close the Advanced message Box automatically. There is no requirement for an "Clear Advanced Message Box" sub-routine.

5.4   Examples
[EXAMPLE REQUIRED]

5.5   Copy-Books
The "$2" copy-book, in copy-library S.DF, expands the following control blocks:

```
BIN    See AMBIN$
BAI    See AMBAI$
BAT    See AMBAT$
BAB    See AMBAB$
BAC
```

5.6   See Also
AMBIN$ Initialise Advanced message box window
AMBAI$ Add icon to Advanced message box window
AMBAT$ Add text to Advanced message box
AMBAB$ Add button to Advanced message box