Stadium Plan and Seat Plan routines for DeFacto

This document describes a number highly-specialised sub-routines that have been implemented for the DeFacto product.

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. DFPIN$ Initialise Stadium Plan window

The initialise window call is used to provide the overall structure of the window and set up the fixed items on the window.

1.1 Invocation

To Initialise the Stadium Plan window code:

```
CALL DFPIN$ USING pin
```

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

```
01    PIN                          * DFPIN$ CONTROL BLOCK
  02  PINVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
                  VALUE 1          * MUST CONTAIN 1
  02  PINPID      PIC X(8)         * PROGRAM ID
  02  PINWID      PIC X(4)         * WINDOW ID
  02  PINLIN      PIC 9(4) COMP    * TOP LEFT LINE
  02  PINCOL      PIC 9(4) COMP    * TOP LEFT COLUMN
  02  PINWDT      PIC 9(4) COMP    * WINDOW WIDTH
  02  PINTDE      PIC 9(4) COMP    * TITLE AREA DEPTH
  02  PINBDE      PIC 9(4) COMP    * BLOCK AREA DEPTH
  02  PINUDE      PIC 9(4) COMP    * BUTTON AREA DEPTH
  02  PINL1       PIC 9(4) COMP    * LENGTH OF WINDOW CAPTION TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PINP1       PIC PTR          * POINTER TO WINDOW CAPTION TEXT
  02  PINL2       PIC 9(4) COMP    * LENGTH OF TITLE AREA TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PINP2       PIC PTR          * POINTER TO TITLE AREA TEXT
```

1.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPIN$:

STOP code	Description
16401	DFPIN$ has been called on a non GX screen.

16402	The PINVER field does not contain 1.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by DFPIN$:

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

DFPIN$ is only available when running on GX. Any attempt to use DFPIN$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

1.4 Examples

[EXAMPLE REQUIRED]

1.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN
PAK See DFPAK$
PAL See DFPAL$
PAG See DFPAG$
PAB See DFPAB$
PUB See DFPUB$
PUK See DFPUK$
PAC See DFPAC$

1.6 See Also

DFPAK$ Add Block to Stadium Plan Window
DFPAL$ Add Label to Stadium Plan Window
DFPAG$ Add Graphic to Stadium Plan Window
DFPAB$ Add Button to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPUK$ Update Block on Stadium Plan Window
DFPAC$ Accept Operation on Stadium Plan Window
DFPCL$ Close Stadium Plan Window

2. DFPAK$ Add Block to Stadium Plan Window

The add block call is used to add a block definition to the stadium plan window set up by the initialise window call.

2.1 Invocation

To add a block to the Stadium Plan window code:

```
CALL DFPAK$ USING pak
```

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

```
01    PAK                          * DFPAK$ CONTROL BLOCK
  02  PAKVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
                  VALUE 1          * MUST CONTAIN 1
  02  PAKBID      PIC X(4)         * BLOCK-ID
  02  PAKLIN      PIC 9(4) COMP    * TOP LEFT LINE
  02  PAKCOL      PIC 9(4) COMP    * TOP LEFT COLUMN
  02  PAKWID      PIC 9(4) COMP    * BLOCK WIDTH
  02  PAKDEP      PIC 9(4) COMP    * BLOCK DEPTH
  02  PAKORI      PIC 9(2) COMP    * CAPTION ORIENTATION
                                   * 0 = SIDE-BY-SIDE
                                   * N = ABOVE (THE VLAUE SPECIFIES THE
                                   * GAP BETWEEN THE LINES IN PIXELS)
  02  PAKATT      PIC 9(2) COMP    * EXTENDED BLOCK ATTRIBUTE (1 TO 64)
  02  PAKL1       PIC 9(4) COMP    * LENGTH OF 1ST BLOCK TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PAKP1       PIC PTR          * POINTER TO 1ST BLOCK TEXT
  02  PAKL2       PIC 9(4) COMP    * LENGTH OF 2ND BLOCK TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PAKP2       PIC PTR          * POINTER TO 2ND BLOCK TEXT
  02  PAKL3       PIC 9(4) COMP    * LENGTH OF HOVER TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PAKP3       PIC PTR          * POINTER TO HOVER TEXT
```

2.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPAK$:

STOP code	Description
16401	DFPAK$ has been called on a non GX screen.
16402	The PAKVER field does not contain 1.
16403	The PAKATT field does not contain a value between 1 and 64.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by DFPAK$:

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

DFPAK$ is only available when running on GX. Any attempt to use DFPAK$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The extended attribute refers to the extended attribute colour combinations configured in GX.

The caption defines the text to be displayed on the block. This is defined in two parts to allow it to be displayed according to the orientation flag specified in the add block operation. If side-by-side orientation is specified then Text1 and Text2 are concatenated and displayed as a single line. If above orientation is specified then Text1 and displayed above Text2.

The hover text defines the text to be displayed when the mouse remains stationary over a block.

2.4 Examples

[EXAMPLE REQUIRED]

2.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN See DFPIN$
PAK
PAL See DFPAL$
PAG See DFPAG$
PAB See DFPAB$
PUB See DFPUB$
PUK See DFPUK$
PAC See DFPAC$

2.6 See Also

DFPIN$ Initialise Stadium Plan window
DFPAL$ Add Label to Stadium Plan Window
DFPAG$ Add Graphic to Stadium Plan Window
DFPAB$ Add Button to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPUK$ Update Block on Stadium Plan Window
DFPAC$ Accept Operation on Stadium Plan Window

DFPCL$ Close Stadium Plan Window

3. DFPAL$ Add Label to Stadium Plan Window

The add label call is used to add a label definition to the block display area.

3.1 Invocation

To add a label to the Stadium Plan window code:

```
CALL DFPAL$ USING pal
```

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

```
01    PAL                          * DFPAL$ CONTROL BLOCK
  02  PALVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
                  VALUE 1          * MUST CONTAIN 1
  02  PALLIN      PIC 9(4) COMP    * LINE NUMBER
  02  PALCOL      PIC 9(4) COMP    * COLUMN NUMBER
  02  PALATT      PIC 9(2) COMP    * EXTENDED BLOCK ATTRIBUTE (1 TO 64)
  02  PALL1       PIC 9(4) COMP    * LENGTH OF CAPTION TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PALP1       PIC PTR          * POINTER TO CAPTION TEXT
```

3.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPAL$:

STOP code	Description
16401	DFPAL$ has been called on a non GX screen.
16402	The PALVER field does not contain 1.
16403	The PALATT field does not contain a value between 1 and 64.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by DFPAL$:

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

DFPAL$ is only available when running on GX. Any attempt to use DFPAL$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The extended attribute refers to the extended attribute colour combinations configured in GX.

The caption defines the text to be displayed for the label.

3.4 Examples

[EXAMPLE REQUIRED]

3.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN See DFPIN$
PAK See DFPAK$
PAL
PAG See DFPAG$
PAB See DFPAB$
PUB See DFPUB$
PUK See DFPUK$
PAC See DFPAC$

3.6 See Also

DFPIN$ Initialise Stadium Plan window
DFPAK$ Add Block to Stadium Plan Window
DFPAG$ Add Graphic to Stadium Plan Window
DFPAB$ Add Button to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPUK$ Update Block on Stadium Plan Window
DFPAC$ Accept Operation on Stadium Plan Window
DFPCL$ Close Stadium Plan Window

4. DFPAG$ Add Bitmap Graphic to Stadium Plan Window

The add graphic call is used to add a bitmap to the block display area.

4.1 Invocation

To add a bitmap to the Stadium Plan window code:

```
CALL DFPAG$ USING pag
```

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

```
01    PAG                          * DFPAG$ CONTROL BLOCK
  02  PAGVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
                  VALUE 1          * MUST CONTAIN 1
  02  PAGLIN      PIC 9(4) COMP    * TOP LEFT LINE
```

```
02  PAGCOL      PIC 9(4) COMP     * TOP LEFT COLUMN
02  PAGWID      PIC 9(4) COMP     * GRAPHIC WIDTH
02  PAGDEP      PIC 9(4) COMP     * GRAPHIC DEPTH
02  PAGBMI      PIC 9(3)          * BITMAP INDEX
02  PAGFSF      PIC X             * FIXED SIZE FLAG
                                  * "F" = Display as fixed size
                                  * "S" = Stretch to fill area
```

4.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPAG$:

STOP code	Description
16401	DFPAG$ has been called on a non GX screen.
16402	The PAGVER field does not contain 1.
16404	The PAGFSF field does not contain either “F” or “S”.

The following EXIT codes may be returned by DFPAG$:

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

DFPAG$ is only available when running on GX. Any attempt to use DFPAG$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The Bitmap Index passed via PAGBMI **must** contain leading zeroes, where necessary (e.g. "001" rather than " 1"; "023" rather than " 23").

4.4 Examples

[EXAMPLE REQUIRED]

4.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN See DFPIN$
PAK See DFPAK$

PAL See DFPAL$
PAG
PAB See DFPAB$
PUB See DFPUB$
PUK See DFPUK$
PAC See DFPAC$

4.6 See Also

DFPIN$ Initialise Stadium Plan window
DFPAK$ Add Block to Stadium Plan Window
DFPAL$ Add Label to Stadium Plan Window
DFPAB$ Add Button to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPUK$ Update Block on Stadium Plan Window
DFPAC$ Accept Operation on Stadium Plan Window
DFPCL$ Close Stadium Plan Window

5. DFPAB$ Add Button to Stadium Plan Window

The add button call is used to add a button definition to the stadium plan window set up by the initialise window call.

5.1 Invocation

To add a button to the Stadium Plan window code:

```
CALL DFPAB$ USING pab
```

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

```
01    PAB                          * DFPAB$ CONTROL BLOCK
  02  PABVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
                  VALUE 1          * MUST CONTAIN 1
  02  PABBID      PIC 9(2) COMP    * BUTTON ID
  02  PABLIN      PIC 9(4) COMP    * TOP LEFT LINE
  02  PABCOL      PIC 9(4) COMP    * TOP LEFT COLUMN
  02  PABWID      PIC 9(4) COMP    * BUTTON WIDTH
  02  PABL1       PIC 9(4) COMP    * LENGTH OF BUTTON TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PABP1       PIC PTR          * POINTER TO BUTTON TEXT
```

5.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPAB$:

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

16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by DFPAB$:

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

DFPAB$ is only available when running on GX. Any attempt to use DFPAB$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The caption defines the text to be displayed on the button.

5.4 Examples

[EXAMPLE REQUIRED]

5.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN See DFPIN$
PAK See DFPAK$
PAL See DFPAL$
PAG See DFPAG$
PAB
PUB See DFPUB$
PUK See DFPUK$
PAC See DFPAC$

5.6 See Also

DFPIN$ Initialise Stadium Plan window
DFPAK$ Add Block to Stadium Plan Window
DFPAL$ Add Label to Stadium Plan Window
DFPAG$ Add Graphic to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPUK$ Update Block on Stadium Plan Window
DFPAC$ Accept Operation on Stadium Plan Window
DFPCL$ Close Stadium Plan Window

6. DFPUB$ Update Button on Stadium Plan Window

The update button call is used to update a button definition set up by the Add Button call.

6.1 Invocation

To update a button on the Stadium Plan window code:

```
CALL DFPUB$ USING pub
```

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

```
01    PUB                              * DFPUB$ CONTROL BLOCK
  02  PUBVER      PIC 9(4) COMP        * BLOCK VERSION NUMBER
                  VALUE 1              * MUST CONTAIN 1
  02  PUBBID      PIC 9(2) COMP        * BUTTON ID
  02  PUBFLG      PIC 9(2) COMP        * BUTTON FLAG
                                       * 0 = Button disabled
                                       * 1 = Button enabled
                                       * N = Reserved for future use
  02  PUBL1       PIC 9(4) COMP        * LENGTH OF BUTTON TEXT
                                       * 0 = ZERO-TERMINATED STRING
                                       * N = FIXED LENGTH STRING, LENGTH N
                                       * -1 = NO TEXT STRING DEFINED
  02  PUBP1       PIC PTR              * POINTER TO BUTTON TEXT
```

6.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPUB$:

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

The following EXIT codes may be returned by DFPUB$:

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

6.3 Programming Notes

DFPUB$ is only available when running on GX. Any attempt to use DFPUB$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

6.4 Examples

[EXAMPLE REQUIRED]

6.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN	See DFPIN$
PAK	See DFPAK$
PAL	See DFPAL$
PAG	See DFPAG$
PAB	See DFPAB$
PUB	
PUK	See DFPUK$
PAC	See DFPAC$

6.6 See Also

DFPIN$	Initialise Stadium Plan window
DFPAK$	Add Block to Stadium Plan Window
DFPAL$	Add Label to Stadium Plan Window
DFPAG$	Add Graphic to Stadium Plan Window
DFPAB$	Add Button to Stadium Plan Window
DFPUK$	Update Block on Stadium Plan Window
DFPAC$	Accept Operation on Stadium Plan Window
DFPCL$	Close Stadium Plan Window

7. DFPUK$ Update Block on Stadium Plan Window

The update block call is used to modify the current state and (optionally) text on a block.

7.1 Invocation

To update a block on the Stadium Plan window code:

```
CALL DFPUK$ USING puk
```

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

```
01    PUK                          * DFPUK$ control block
  02  PUKVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
                  VALUE 1          * MUST CONTAIN 1
  02  PUKID       PIC X(4)         * BLOCK-ID
  02  PUKATT      PIC 9(2) COMP    * EXTENDED BLOCK ATTRIBUTE (1 TO 64)
  02  PUKL1       PIC 9(4) COMP    * LENGTH OF 1ST BLOCK TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PUKP1       PIC PTR          * POINTER TO 1ST BLOCK TEXT
  02  PUKL2       PIC 9(4) COMP    * LENGTH OF 2ND BLOCK TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PUKP2       PIC PTR          * POINTER TO 2ND BLOCK TEXT
  02  PUKL3       PIC 9(4) COMP    * LENGTH OF HOVER TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  PUKP3       PIC PTR          * POINTER TO HOVER TEXT
```

7.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPUK$:

STOP code	Description
16401	DFPUK$ has been called on a non GX screen.
16402	The PUKVER field does not contain 1.
16403	The PUKATT field does not contain a value between 1 and 64.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by DFPUK$:

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

7.3 Programming Notes

DFPUK$ is only available when running on GX. Any attempt to use DFPUK$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The extended attribute refers to the extended attribute colour combinations configured in GX.

The caption defines the text to be displayed on the block.

The hover text defines the text to be displayed when the mouse remains stationary over a block.

7.4 Examples

[EXAMPLE REQUIRED]

7.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN See DFPIN$
PAK See DFPAK$
PAL See DFPAL$
PAG See DFPAG$
PAB See DFPAB$
PUB See DFPUB$
PUK
PAC See DFPAC$

7.6 See Also

DFPIN$ Initialise Stadium Plan window
DFPAK$ Add Block to Stadium Plan Window
DFPAL$ Add Label to Stadium Plan Window
DFPAG$ Add Graphic to Stadium Plan Window
DFPAB$ Add Button to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPAC$ Accept Operation on Stadium Plan Window
DFPCL$ Close Stadium Plan Window

8. DFPAC$ Accept Operation on Stadium Plan Window

The accept call selects the current block, passes control to the stadium plan window and returns on the first keystroke or mouse click detected in the window.

8.1 Invocation

To issue a SINGLE CHARACTER accept on the Stadium Plan window code:

```
CALL DFPAC$ USING pac
```

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

```
01    PAC                              * DFPAC$ CONTROL BLOCK
  02  PACVER      PIC 9(4) COMP        * BLOCK VERSION NUMBER
                  VALUE 1              * MUST CONTAIN 1
  02  PACID       PIC X(4)             * BLOCK-ID
  02  PACTO       PIC 9(4) COMP        * ACCEPT TIME-OUT
  02  PACRID      PIC X(4)             * RETURNED SELECTED BLOCK-ID
  02  PACEXE      PIC 9(2) COMP        * RETURNED KEYBOARD/BUTTON-ID
  02  PACFLG      PIC 9(2) COMP        * RETURNED MOUSE CLICK FLAG
                                       * #01 = Left button
                                       * #02 = Right button
                                       * #04 = SHIFT key
                                       * #08 = CTRL key
                                       * #10 = Middle button
```

8.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPAC$:

STOP code	Description
16401	DFPAC$ has been called on a non GX screen.

16402	The PACVER field does not contain 1.

The following EXIT codes may be returned by DFPAC$:

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

8.3 Programming Notes

DFPAC$ is only available when running on GX. Any attempt to use DFPAC$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The PACRID, PACEXE and PACFLG fields are returned by GX.

The mouse click flag will always be set to 1 when clicking on a button since this event will only be triggered by a left mouse click. The full range of mouse click values will only be returned when clicking on a block. When the keyboard is used the mouse click flag will always be 0.

8.4 Examples

[EXAMPLE REQUIRED]

8.5 Copy-Books

The "$0" copy-book expands the following control blocks:

PIN See DFPIN$
PAK See DFPAK$
PAL See DFPAL$
PAG See DFPAG$
PAB See DFPAB$
PUB See DFPUB$
PUK See DFPUK$
PAC

8.6 See Also

DFPIN$ Initialise Stadium Plan window
DFPAK$ Add Block to Stadium Plan Window
DFPAL$ Add Label to Stadium Plan Window
DFPAG$ Add Graphic to Stadium Plan Window
DFPAB$ Add Button to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPUK$ Update Block on Stadium Plan Window

DFPCL$ Close Stadium Plan Window

9. DFPCL$ Close Stadium Plan Window

The close window call removes the stadium plan window from the screen.

9.1 Invocation

To close the Stadium Plan window code:

```
CALL DFPCL$
```

9.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFPCL$:

STOP code	Description
16401	DFPCL$ has been called on a non GX screen.

The following EXIT codes may be returned by DFPCL$:

EXIT code	$$COND	Description
16401	1	An exception was returned by GX.

9.3 Programming Notes

DFPCL$ is only available when running on GX. Any attempt to use DFPCL$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

9.4 Examples

[EXAMPLE REQUIRED]

9.5 Copy-Books

None.

9.6 See Also

DFPIN$ Initialise Stadium Plan window
DFPAK$ Add Block to Stadium Plan Window
DFPAL$ Add Label to Stadium Plan Window
DFPAG$ Add Graphic to Stadium Plan Window
DFPAB$ Add Button to Stadium Plan Window
DFPUB$ Update Button on Stadium Plan Window
DFPUK$ Update Block on Stadium Plan Window
DFPAC$ Accept Operation on Stadium Plan Window

10. DFSIN$ Initialise Seat Plan window

The initialise window call is used to provide the overall structure of the window and set up the fixed items on the window.

10.1 Invocation

To Initialise the Seat Plan window code:

```
CALL DFSIN$ USING sin
```

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

```
01    SIN                          * DFSIN$ CONTROL BLOCK
  02  SINVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
  02  SINPID      PIC X(8)         * PROGRAM ID
  02  SINWID      PIC X(4)         * WINDOW ID
  02  SINLIN      PIC 9(4) COMP    * TOP LEFT LINE
  02  SINCOL      PIC 9(4) COMP    * TOP LEFT COLUMN
  02  SINWDT      PIC 9(4) COMP    * WINDOW WIDTH
  02  SINTDE      PIC 9(4) COMP    * TITLE AREA DEPTH
  02  SINHDE      PIC 9(4) COMP    * HEADER AREA DEPTH
  02  SINSDE      PIC 9(4) COMP    * SEAT AREA DEPTH
  02  SINFDE      PIC 9(4) COMP    * FOOTER AREA DEPTH
  02  SINUDE      PIC 9(4) COMP    * BUTTON AREA DEPTH
  02  SINL1       PIC 9(4) COMP    * LENGTH OF CAPTION TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  SINP1       PIC PTR          * POINTER TO CAPTION TEXT
  02  SINL2       PIC 9(4) COMP    * LENGTH OF TITLE TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  SINP2       PIC PTR          * POINTER TO TITLE TEXT
```

10.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSIN$:

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

The following EXIT codes may be returned by DFSIN$:

EXIT code	$$COND	Description

16401	1	An exception was returned by GX.
16402	2	Insufficient memory to allocate a work buffer.

10.3 Programming Notes

DFSIN$ is only available when running on GX. Any attempt to use DFSIN$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

10.4 Examples

[EXAMPLE REQUIRED]

10.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN
SRN See DFSRN$
SSN See DFSSN$
SAD See DFSAD$
SAP See DFSAP$
SAR See DFSAR$
SAB See DFSAB$
SUB See DFSUB$
SAL See DFSAL$
SUS See DFSUS$
SUL See DFSUL$
SAC See DFSAC$

10.6 See Also

DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window
DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSAC$ Accept Operation on Seat Plan Window
DFSCL$ Close Seat Plan Window

11. DFSRN$ Add Row Note to Seat Plan window

The Add Row Note routine is used to associate a row-note index with a text-string.

11.1 Invocation

To add a Row Note to the Seat Plan window code:

```
CALL DFSRN$ USING srn
```

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

```
01    SRN                              * DFSRN$ CONTROL BLOCK
  02  SRNVER      PIC 9(4) COMP        * BLOCK VERSION NUMBER
  02  SRNINX      PIC 9(2) COMP        * ROW NOTE INDEX
  02  SRNL1       PIC 9(4) COMP        * LENGTH OF NOTE TEXT
                                       * 0 = ZERO-TERMINATED STRING
                                       * N = FIXED LENGTH STRING, LENGTH N
                                       * -1 = NO TEXT STRING DEFINED
  02  SRNP1       PIC PTR              * POINTER TO NOTE TEXT
```

11.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSRN$:

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

The following EXIT codes may be returned by DFSRN$:

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

11.3 Programming Notes

DFSRN$ is only available when running on GX. Any attempt to use DFSRN$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

11.4 Examples

[EXAMPLE REQUIRED]

11.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN	See DFSIN$
SRN	
SSN	See DFSSN$
SAD	See DFSAD$
SAP	See DFSAP$
SAR	See DFSAR$
SAB	See DFSAB$
SUB	See DFSUB$
SAL	See DFSAL$
SUS	See DFSUS$
SUL	See DFSUL$
SAC	See DFSAC$

11.6 See Also

DFSIN$	Initialise Seat Plan Window
DFSSN$	Add seat note to Seat Plan Window
DFSAD$	Add price description to Seat Plan Window
DFSAP$	Add price code list to Seat Plan Window
DFSAR$	Add row to Seat Plan Window
DFSAB$	Add button to Seat Plan Window
DFSUB$	Update button on Seat Plan Window
DFSAL$	Add label to Seat Plan Window
DFSUS$	Update seat on Seat Plan Window
DFSUL$	Update label on Seat Plan Window
DFSAC$	Accept Operation on Seat Plan Window
DFSCL$	Close Seat Plan Window

12. DFSSN$ Add Seat Note to Seat Plan window

The Add Seat Note routine is used to associate a seat-note index with a text-string.

12.1 Invocation

To add a Seat Note to the Seat Plan window code:

```
CALL DFSSN$ USING ssn
```

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

```
01    SSN                           * DFSSN$ CONTROL BLOCK
  02  SSNVER      PIC 9(4) COMP     * BLOCK VERSION NUMBER
  02  SSNINX      PIC 9(2) COMP     * SEAT NOTE INDEX
  02  SSNL1       PIC 9(4) COMP     * LENGTH OF NOTE TEXT
                                    * 0 = ZERO-TERMINATED STRING
                                    * N = FIXED LENGTH STRING, LENGTH N
                                    * -1 = NO TEXT STRING DEFINED
  02  SSNP1       PIC PTR           * POINTER TO NOTE TEXT
```

12.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSSN$:

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

The following EXIT codes may be returned by DFSSN$:

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

12.3 Programming Notes

DFSSN$ is only available when running on GX. Any attempt to use DFSSN$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

12.4 Examples

[EXAMPLE REQUIRED]

12.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN See DFSIN$
SRN See DFSRN$
SSN
SAD See DFSAD$
SAP See DFSAP$
SAR See DFSAR$
SAB See DFSAB$
SUB See DFSUB$
SAL See DFSAL$
SUS See DFSUS$
SUL See DFSUL$
SAC See DFSAC$

12.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSAC$ Accept Operation on Seat Plan Window
DFSCL$ Close Seat Plan Window

13. DFSAD$ Add Price Description to Seat Plan window

The Add Price Description routine is used to associate a Price Description index with a text-string.

13.1 Invocation

To add a Price Description to the Seat Plan window code:

```
CALL DFSAD$ USING sad
```

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

```
01    SAD                              * DFSAD$ CONTROL BLOCK
  02  SADVER      PIC 9(4) COMP        * BLOCK VERSION NUMBER
  02  SADINX      PIC 9(2) COMP        * PRICE DESC INDEX
  02  SADL1       PIC 9(4) COMP        * LENGTH OF PRICE DESC TEXT
                                       * 0 = ZERO-TERMINATED STRING
                                       * N = FIXED LENGTH STRING, LENGTH N
                                       * -1 = NO TEXT STRING DEFINED
  02  SADP1       PIC PTR              * POINTER TO PRICE DESC TEXT
```

13.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSAD$:

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

The following EXIT codes may be returned by DFSAD$:

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

13.3 Programming Notes

DFSAD$ is only available when running on GX. Any attempt to use DFSAD$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

13.4 Examples

[EXAMPLE REQUIRED]

13.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN See DFSIN$
SRN See DFSRN$
SSN See DFSSN$
SAD
SAP See DFSAP$
SAR See DFSAR$
SAB See DFSAB$
SUB See DFSUB$
SAL See DFSAL$
SUS See DFSUS$
SUL See DFSUL$
SAC See DFSAC$

13.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSAC$ Accept Operation on Seat Plan Window
DFSCL$ Close Seat Plan Window

14. DFSAP$ Add Price Code line to Seat Plan window

The add Price Code line call is used to add an extra price code line to a price code note which

is formed of multiple lines of text.

14.1 Invocation

To add Price Code line to the Seat Plan window code:

```
CALL DFSAP$ USING sap
```

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

```
01    SAP                          * DFSAP$ CONTROL BLOCK
  02  SAPVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
  02  SAPINX      PIC 9(2) COMP    * PRICE CODE LINE INDEX
  02  SAPDIX      PIC 9(2) COMP    * ASSOCIATED PRICE DESC. INDEX
  02  SAPL1       PIC 9(4) COMP    * LENGTH OF PRICE CODE TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  SAPP1       PIC PTR          * POINTER TO PRICE CODE TEXT
```

14.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSAP$:

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

The following EXIT codes may be returned by DFSAP$:

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

14.3 Programming Notes

DFSAP$ is only available when running on GX. Any attempt to use DFSAP$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

14.4 Examples

[EXAMPLE REQUIRED]

14.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN	See DFSIN$
SRN	See DFSRN$
SSN	See DFSSN$
SAD	See DFSAD$
SAP	
SAR	See DFSAR$
SAB	See DFSAB$
SUB	See DFSUB$
SAL	See DFSAL$
SUS	See DFSUS$
SUL	See DFSUL$
SAC	See DFSAC$

14.6 See Also

DFSIN$	Initialise Seat Plan Window
DFSRN$	Add row note to Seat Plan Window
DFSSN$	Add seat note to Seat Plan Window
DFSAD$	Add price description to Seat Plan Window
DFSAR$	Add row to Seat Plan Window
DFSAB$	Add button to Seat Plan Window
DFSUB$	Update button on Seat Plan Window
DFSAL$	Add label to Seat Plan Window
DFSUS$	Update seat on Seat Plan Window
DFSUL$	Update label on Seat Plan Window
DFSAC$	Accept Operation on Seat Plan Window
DFSCL$	Close Seat Plan Window

15. DFSAR$ Add Row to Seat Plan window

The add row call is used to add an entire row of seats to the seat plan window.

15.11 Invocation

To add a row of seats to the Seat Plan window code:

```
CALL DFSAR$ USING sar
```

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

```
01    SAR                          * DFSAR$ CONTROL BLOCK
02    SARVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
                  VALUE 2          * VERSION NUMBER
  02  SARNAM      PIC X(2)         * ROW NAME
  02  SARLIN      PIC 9(2) COMP    * LINE NUMBER
  02  SARCOL      PIC 9(2) COMP    * COLUMN NUMBER
  02  SARNOT      PIC 9(2) COMP    * ROW NOTE INDEX
  02  SARCNT      PIC 9(4) COMP    * COUNT OF NUMBER OF SEATS
  02  FILLER OCCURS 250
   03 SARATT      PIC 9(2) COMP    * EXTENDED BLOCK ATTRIBUTE (1 TO 64)
```

```
   03 SARFLG      PIC 9(2) COMP     * SPECIAL FLAGS
                                    * 0 = NORMAL SEAT
                                    * 1 = SOLD SEAT
                                    * 2 = NON-SEAT
                                    * 3 = RESERVED FOR FUTURE USE
                                    * >3 = INVALID
   03 SARSNI      PIC 9(2) COMP     * SEAT NOTE INDEX
   03 SARPCI      PIC 9(2) COMP     * PRICE CODE INDEX
*
*     The following fields have been added for version 2, and later,
*     of this control block.
*
   03 SARNUM      PIC 9(4) COMP     * SEAT NUMBER (NUMERIC)
   03 SARSUF      PIC X             * SEAT NUMBER SUFFIX
```

15.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSAR$:

STOP code	Description
16401	DFSAR$ has been called on a non GX screen.
16402	The SARVER field does not contain 1.
16403	The SARATT field does not contain a value between 1 and 64.
16405	A zero-terminated string is longer than 255 characters.
16406	The SARCNT field does not contain a value between 1 and 250.
16407	The SARFLG field does not contain a value between 0 and 3.

The following EXIT codes may be returned by DFSAR$:

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

15.3 Programming Notes

DFSAR$ is only available when running on GX. Any attempt to use DFSAR$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The extended attribute refers to the extended attribute colour combinations configured in GX.

15.4 Examples

[EXAMPLE REQUIRED]

15.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN	See DFSIN$
SRN	See DFSRN$
SSN	See DFSSN$
SAD	See DFSAD$
SAP	See DFSAP$
SAR	
SAB	See DFSAB$
SUB	See DFSUB$
SAL	See DFSAL$
SUS	See DFSUS$
SUL	See DFSUL$
SAC	See DFSAC$

15.6 See Also

DFSIN$	Initialise Seat Plan Window
DFSRN$	Add row note to Seat Plan Window
DFSSN$	Add seat note to Seat Plan Window
DFSAD$	Add price description to Seat Plan Window
DFSAP$	Add price code list to Seat Plan Window
DFSAB$	Add button to Seat Plan Window
DFSUB$	Update button on Seat Plan Window
DFSAL$	Add label to Seat Plan Window
DFSUS$	Update seat on Seat Plan Window
DFSUL$	Update label on Seat Plan Window
DFSAC$	Accept Operation on Seat Plan Window
DFSCL$	Close Seat Plan Window

16. DFSAB$ Add Button to Seat Plan window

The add button call is used to add a button definition to the seat plan widnow.

16.1 Invocation

To add a button to the Seat Plan window code:

```
CALL DFSAB$ USING sab
```

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

```
01    SAB                          * DFSAB$ CONTROL BLOCK
  02  SABVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
```

```
02  SABBID      PIC 9(2) COMP     * BUTTON ID
02  SABLIN      PIC 9(4) COMP     * TOP LEFT LINE
02  SABCOL      PIC 9(4) COMP     * TOP LEFT COLUMN
02  SABWID      PIC 9(4) COMP     * BUTTON WIDTH
02  SABL1       PIC 9(4) COMP     * LENGTH OF BUTTON TEXT
                                  * 0 = ZERO-TERMINATED STRING
                                  * N = FIXED LENGTH STRING, LENGTH N
                                  * -1 = NO TEXT STRING DEFINED
02  SABP1       PIC PTR           * POINTER TO BUTTON TEXT
```

16.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSAB$:

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

The following EXIT codes may be returned by DFSAB$:

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

16.3 Programming Notes

DFSAB$ is only available when running on GX. Any attempt to use DFSAB$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

16.4 Examples

[EXAMPLE REQUIRED]

16.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN See DFSIN$
SRN See DFSRN$
SSN See DFSSN$

SAD See DFSAD$
SAP See DFSAP$
SAR See DFSAR$
SAB
SUB See DFSUB$
SAL See DFSAL$
SUS See DFSUS$
SUL See DFSUL$
SAC See DFSAC$

16.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window
DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSAC$ Accept Operation on Seat Plan Window
DFSCL$ Close Seat Plan Window

17. DFSUB$ Update Button on Seat Plan window

The update button call is used to update a button definition set up by the Add Button call.

17.1 Invocation

To update a button on the Seat Plan window code:

```
CALL DFSUB$ USING sub
```

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

```
01    SUB                              * DFSUB$ CONTROL BLOCK
  02  SUBVER      PIC 9(4) COMP        * BLOCK VERSION NUMBER
                  VALUE 1              * MUST CONTAIN 1
  02  SUBBID      PIC 9(2) COMP        * BUTTON ID
  02  SUBFLG      PIC 9(2) COMP        * BUTTON FLAG
                                       * 0 = Button disabled
                                       * 1 = Button enabled
                                       * N = Reserved for future use
02    SUBL1       PIC 9(4) COMP        * LENGTH OF BUTTON TEXT
                                       * 0 = ZERO-TERMINATED STRING
                                       * N = FIXED LENGTH STRING, LENGTH N
                                       * -1 = NO TEXT STRING DEFINED
  02  SUBP1       PIC PTR              * POINTER TO BUTTON TEXT
```

17.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSUB$:

STOP code	Description

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

The following EXIT codes may be returned by DFSUB$:

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

17.3 Programming Notes

DFSUB$ is only available when running on GX. Any attempt to use DFSUB$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

17.4 Examples

[EXAMPLE REQUIRED]

17.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN See DFSIN$
SRN See DFSRN$
SSN See DFSSN$
SAD See DFSAD$
SAP See DFSAP$
SAR See DFSAR$
SAB See DFSAB$
SUB
SAL See DFSAL$
SUS See DFSUS$
SUL See DFSUL$
SAC See DFSAC$

17.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window

DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSAC$ Accept Operation on Seat Plan Window
DFSCL$ Close Seat Plan Window

18. DFSAL$ Add Label to Seat Plan window

The add label call is used to add a label definition to the seat plan window.

18.1 Invocation

To add a label to the Seat Plan window code:

```
CALL DFSAL$ USING sal
```

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

```
01    SAL                          * DFSAL$ CONTROL BLOCK
  02  SALVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
  02  SALHF       PIC 9(2) COMP    * AREA TO PLACE LABEL
*                                  * 1 = HEADER AREA
*                                  * 2 = FOOTER AREA
  02  SALID       PIC 9(2) COMP    * LABEL ID
  02  SALLIN      PIC 9(4) COMP    * LINE NUMBER
  02  SALCOL      PIC 9(4) COMP    * COLUMN NUMBER
  02  SALWID      PIC 9(4) COMP    * LABEL WIDTH
  02  SALATT      PIC 9(2) COMP    * EXTENDED BLOCK ATTRIBUTE (1 TO 64)
  02  SALL1       PIC 9(4) COMP    * LENGTH OF CAPTION TEXT
                                   * 0 = ZERO-TERMINATED STRING
                                   * N = FIXED LENGTH STRING, LENGTH N
                                   * -1 = NO TEXT STRING DEFINED
  02  SALP1       PIC PTR          * POINTER TO CAPTION TEXT
```

18.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSAL$:

STOP code	Description
16401	DFSAL$ has been called on a non GX screen.
16402	The SALVER field does not contain 1.
16403	The SALATT field does not contain a value between 1 and 64.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by DFSAL$:

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

18.3 Programming Notes

DFSAL$ is only available when running on GX. Any attempt to use DFSAL$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The extended attribute refers to the extended attribute colour combinations configured in GX.

18.4 Examples

[EXAMPLE REQUIRED]

18.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN See DFSIN$
SRN See DFSRN$
SSN See DFSSN$
SAD See DFSAD$
SAP See DFSAP$
SAR See DFSAR$
SAB See DFSAB$
SUB See DFSUB$
SAL
SUS See DFSUS$
SUL See DFSUL$
SAC See DFSAC$

18.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window
DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window

DFSAC$ Accept Operation on Seat Plan Window
DFSCL$ Close Seat Plan Window

19. DFSUS$ Update Seat on Seat Plan window

The update seat call is used to modify the current attribute of a seat on the Seat Plan window.

19.1 Invocation

To update a seat on the Seat Plan window code:

```
CALL DFSUS$ USING sus
```

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

```
01    SUS                              * DFSUS$ CONTROL BLOCK
  02  SUSVER      PIC 9(4) COMP        * BLOCK VERSION NUMBER
  02  SUSLIN      PIC 9(2) COMP        * LINE NUMBER
  02  SUSCOL      PIC 9(2) COMP        * COLUMN NUMBER
  02  SUSATT      PIC 9(2) COMP        * EXTENDED BLOCK ATTRIBUTE (1 TO 64)
  02  SUSFLG      PIC 9(2) COMP        * SPECIAL FLAGS
                                       * 0 = NORMAL SEAT
                                       * 1 = SOLD SEAT
                                       * 2 = NON-SEAT
                                       * 3 = RESERVED FOR FUTURE USE
                                       * >3 = INVALID
  02  SUSSNI      PIC 9(2) COMP        * SEAT NOTE INDEX
  02  SUSPCI      PIC 9(2) COMP        * PRICE CODE INDEX
```

19.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSUS$:

STOP code	Description
16401	DFSUS$ has been called on a non GX screen.
16402	The SUSVER field does not contain 1.
16403	The SUSATT field does not contain a value between 1 and 64.
16405	A zero-terminated string is longer than 255 characters.
16407	The SUSFLG field does not contain a value between 0 and 3.

The following EXIT codes may be returned by DFSUS$:

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

19.3 Programming Notes

DFSUS$ is only available when running on GX. Any attempt to use DFSUS$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The extended attribute refers to the extended attribute colour combinations configured in GX.

19.4 Examples

[EXAMPLE REQUIRED]

19.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN See DFSIN$
SRN See DFSRN$
SSN See DFSSN$
SAD See DFSAD$
SAP See DFSAP$
SAR See DFSAR$
SAB See DFSAB$
SUB See DFSUB$
SAL See DFSAL$
SUS
SUL See DFSUL$
SAC See DFSAC$

19.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window
DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSAC$ Accept Operation on Seat Plan Window
DFSCL$ Close Seat Plan Window

20. DFSUL$ Update Label on Seat Plan window

The update label call is used to modify the current attribute of a label on the seat plan window.

20.1 Invocation

To update a label on the Seat Plan window code:

```
CALL DFSUL$ USING sul
```

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

```
01    SUL                              * DFSUL$ CONTROL BLOCK
  02  SULVER      PIC 9(4) COMP        * BLOCK VERSION NUMBER
  02  SULHF       PIC 9(2) COMP        * AREA TO PLACE LABEL
*                                      * 1 = HEADER AREA
*                                      * 2 = FOOTER AREA
  02  SULID       PIC 9(2) COMP        * LABEL ID
  02  SULATT      PIC 9(2) COMP        * EXTENDED BLOCK ATTRIBUTE (1 TO 64)
  02  SULL1       PIC 9(4) COMP        * LENGTH OF CAPTION TEXT
                                       * 0 = ZERO-TERMINATED STRING
                                       * N = FIXED LENGTH STRING, LENGTH N
                                       * -1 = NO TEXT STRING DEFINED
  02  SULP1       PIC PTR              * POINTER TO CAPTION TEXT
```

20.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSUL$:

STOP code	Description
16401	DFSUL$ has been called on a non GX screen.
16402	The SULVER field does not contain 1.
16403	The SULATT field does not contain a value between 1 and 64.
16405	A zero-terminated string is longer than 255 characters.

The following EXIT codes may be returned by DFSUL$:

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

20.3 Programming Notes

DFSUL$ is only available when running on GX. Any attempt to use DFSUL$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The extended attribute refers to the extended attribute colour combinations configured in GX.

20.4 Examples

[EXAMPLE REQUIRED]

20.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN	See DFSIN$
SRN	See DFSRN$
SSN	See DFSSN$
SAD	See DFSAD$
SAP	See DFSAP$
SAR	See DFSAR$
SAB	See DFSAB$
SUB	See DFSUB$
SAL	See DFSAL$
SUS	See DFSUS$
SUL	
SAC	See DFSAC$

20.6 See Also

DFSIN$	Initialise Seat Plan Window
DFSRN$	Add row note to Seat Plan Window
DFSSN$	Add seat note to Seat Plan Window
DFSAD$	Add price description to Seat Plan Window
DFSAP$	Add price code list to Seat Plan Window
DFSAR$	Add row to Seat Plan Window
DFSAB$	Add button to Seat Plan Window
DFSUB$	Update button on Seat Plan Window
DFSAL$	Add label to Seat Plan Window
DFSUS$	Update seat on Seat Plan Window
DFSAC$	Accept Operation on Seat Plan Window
DFSCL$	Close Seat Plan Window

21. DFSAC$ Accept Operation on Seat Plan window

The Accept operation selects the current seat, passes control to the seat plan window and returns on the first keystroke or mouse click detected in the window.

21.1 Invocation

To issue an accept on the Seat Plan window code:

```
CALL DFSAC$ USING sac
```

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

```
01    SAC                          * DFSAC$ CONTROL BLOCK
  02  SACVER      PIC 9(4) COMP    * BLOCK VERSION NUMBER
  02  SACLIN      PIC 9(2) COMP    * SEAT LINE NUMBER
  02  SACCOL      PIC 9(2) COMP    * SEAT COLUMN NUMBER
  02  SACTIM      PIC 9(4) COMP    * ACCEPT TIMEOUT
  02  SACRLI      PIC 9(2) COMP    * SELECTED SEAT LINE NUMBER
  02  SACRCO      PIC 9(2) COMP    * SELECTED SEAT COLUMN NUMBER
  02  SACEXE      PIC 9(2) COMP    * KEYBOARD/BUTTON-ID
  02  SACFLG      PIC 9(2) COMP    * MOUSE CLICK FLAG
```

21.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSAC$:

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

The following EXIT codes may be returned by DFSAC$:

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

21.3 Programming Notes

DFSAC$ is only available when running on GX. Any attempt to use DFSAC$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

The SACRLI, SACRCO, SACEXE and SACFLG fields are returned by GX.

21.4 Examples

[EXAMPLE REQUIRED]

21.5 Copy-Books

The "$1 copy-book expands the following control blocks:

SIN See DFSIN$
SRN See DFSRN$
SSN See DFSSN$
SAD See DFSAD$
SAP See DFSAP$
SAR See DFSAR$
SAB See DFSAB$
SUB See DFSUB$
SAL See DFSAL$
SUS See DFSUS$
SUL See DFSUL$
SAC

21.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window
DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSCL$ Close Seat Plan Window

22. DFSCL$ Close Seat Plan window

The close window call removes the Seat Plan window from the screen.

22.1 Invocation

To close the Seat Plan window code:

```
CALL DFSCL$
```

22.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFSCL$:

STOP code	Description
16401	DFSCL$ has been called on a non GX screen.

The following EXIT codes may be returned by DFSCL$:

EXIT code	$$COND	Description
16401	1	An exception was returned by GX.

22.3 Programming Notes

DFSCL$ is only available when running on GX. Any attempt to use DFSCL$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

22.4 Examples

[EXAMPLE REQUIRED]

22.5 Copy-Books

None.

22.6 See Also

DFSIN$ Initialise Seat Plan Window
DFSRN$ Add row note to Seat Plan Window
DFSSN$ Add seat note to Seat Plan Window
DFSAD$ Add price description to Seat Plan Window
DFSAP$ Add price code list to Seat Plan Window
DFSAR$ Add row to Seat Plan Window
DFSAB$ Add button to Seat Plan Window
DFSUB$ Update button on Seat Plan Window
DFSAL$ Add label to Seat Plan Window
DFSUS$ Update seat on Seat Plan Window
DFSUL$ Update label on Seat Plan Window
DFSAC$ Accept Operation on Seat Plan Window

23. DFGCA$ Get GX Colour Attribute(s)

The Get Attribute operation displays a dialogue box containing the current set of 64 extended attribute colour combinations. A descriptive piece of text can be associated with each attribute. One (or more) of these combinations can then be updated using the standard Windows “Choose colour” dialogue. Finally the updated combinations will be returned. Note that the customisations made will remain in force until they are reset but are not saved to the GXCUST4.CUS file.

23.1 Invocation

To display the Colour Customisation dialogue box and get one, or more, customised colours code:

```
CALL DFGCA$ USING gcA gcb
```

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

```
01    GCA                                 * DFGCA$ CONTROL BLOCK
  02  GCAVER      PIC 9(4) COMP           * BLOCK VERSION NUMBER
  02  GCADEF      PIC 9(2) COMP           * DEFAULT ATTRIBUTE FOR UPDATE
  02  GCACNT      PIC 9(2) COMP           * NUMBER OF ENTRIES
  02  GCADAT OCCURS 64
```

```
    03 GCAIX       PIC 9(2) COMP          * EXTENDED ATTRIBUTE INDEX NO.
    03 GCALX       PIC 9(4) COMP          * LENGTH OF TEXT STRING
    03 GCAPX       PIC PTR                * POINTER TO TEXT STRING
```

and *gcb* is the returned control block of the following format:

```
01    GCB                                 * DFGCA$ RETURN BLOCK
  02  GCBCNT      PIC 9(2) COMP           * NUMBER OF ENTRIES UPDATED BY USER
  02  GCBDAT OCCURS 64
    03 GCBIX      PIC 9(2) COMP           * EXTENDED ATTRIBUTE INDEX NO.
    03 GCBINK     PIC 9(9) COMP           * INK COLOUR
    03 GCBPAP     PIC 9(9) COMP           * PAPER COLOUR
```

Important Note: The format of the Ink and paper colours are:
blue_ component *256**2 +*green_ component* *256+*red_component*.

23.2 STOP Codes and Exception Conditions

The following STOP codes may be generated by DFGCA$:

STOP code	Description
16401	DFGCA$ has been called on a non GX screen.
16402	The GCAVER field does not contain 1.
16408	The GCACNT field contains 0.
16409	The GCACNT field contains a value higher than 64.

The following EXIT codes may be returned by DFGCA$:

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

23.3 Programming Notes

DFGCA$ is only available when running on GX. Any attempt to use DFGCA$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

23.4 Examples

[EXAMPLE REQUIRED]

23.5 Copy-Books
None.

23.6 See Also
DFSCA$ Set GX Colour Attribute(s)
DFRCA$ Reset GX Colour Attributes

24. DFSCA$ Set GX Colour Attribute(s)
The Set Attribute operation allows one (or more) attributes to be updated in memory. As with the get attribute operation the changes are not saved in the GXCUST4.CUS file.

24.1 Invocation
To set one, or more, customised colours code:

```
CALL DFSCA$ USING sca
```

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

```
01    SCA                                  * DFSCA$ CONTROL BLOCK
  02  SCAVER      PIC 9(4) COMP            * BLOCK VERSION NUMBER
  02  SCACNT      PIC 9(2) COMP            * NUMBER OF ENTRIES
  02  SCADAT OCCURS 64
   03 SCAIX       PIC 9(2) COMP            * EXTENDED ATTRIBUTE INDEX NO.
   03 SCAINK      PIC 9(9) COMP            * INK COLOUR
   03 SCAPAP      PIC 9(9) COMP            * PAPER COLOUR
```

24.2 STOP Codes and Exception Conditions
The following STOP codes may be generated by DFSCA$:

STOP code	Description
16401	DFSCA$ has been called on a non GX screen.
16402	The GSAVER field does not contain 1.
16408	The GSACNT field contains 0.
16409	The GSACNT field contains a value higher than 64.

The following EXIT codes may be returned by DFSCA$:

EXIT code	$$COND	Description

16401	1	An exception was returned by GX.
16402	2	Insufficient memory to allocate a work buffer.

24.3 Programming Notes
DFSCA$ is only available when running on GX. Any attempt to use DFSCA$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

24.4 Examples
[EXAMPLE REQUIRED]

24.5 Copy-Books
None.

24.6 See Also
DFGCA$ Get GX Colour Attribute(s)
DFRCA$ Reset GX Colour Attributes

25. DFRCA$ Reset GX Colour Attributes
The Reset Attribute operation resets all extended attributes to the saved state (i.e. those settings made in GXCUST4.CUS).

25.1 Invocation
To reset the colour attributes to the GXCUST4.CUS defaults:

```
CALL DFRCA$
```

25.2 STOP Codes and Exception Conditions
The following STOP codes may be generated by DFRCA$:

STOP code	**Description**
16401	DFRCA$ has been called on a non GX screen.

The following EXIT codes may be returned by DFRCA$:

EXIT code	**$$COND**	**Description**
16401	1	An exception was returned by GX.

25.3 Programming Notes

DFRCA$ is only available when running on GX. Any attempt to use DFRCA$ on a non-GX terminal will result in a STOP code. The version of GX must be V?.?, or later. The version of GSM must be GSM SP-13, or later.

25.4 Examples

[EXAMPLE REQUIRED]

25.5 Copy-Books

None.

25.6 See Also

DFGCA$ Get GX Colour Attribute(s)
DFSCA$ Set GX Colour Attribute(s)