

```
C          SETUP          BEGSR
**          ~~~~~
C          MOVE *BLANKS   INIT      1
**
C          ENDSR
C*****
```

The next instalment in this series will add some additional commands, routines and a database that will allow messages to be pre-defined and scheduled for use on specific dates. Recurring messages will also be provided for.

Save files containing the source and objects shown in this article are available for download from the Astradyne (UK) website at www.astradyne-uk.com.

```

C          ENDIF          --+ |
C          ENDIF          -----+
C          COLOUR        WHEQ 'YLW'
C          REVIMG        IFEQ '0'          -----+
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'32'    TXT,IX          |
C          ELSE          --<          |
C          MOVE X'36'    TXT,IX          |
C          ENDIF        --+          |
C          ELSE          -----<
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'33'    TXT,IX          |
C          ELSE          --<          |
C          MOVE '↵'      TXT,IX          |
C          ENDIF        --+          |
C          ENDIF        -----+
C          OTHER
C          MOVE '↵'      TXT,IX
C          ENDSL
C          ENDIF
**
** Show the attribute on the example display...
**
C          MOVEATXT,001  MSG001
C          MOVEATXT,079  MSG002
C          MOVEATXT,157  MSG003
C          MOVEATXT,235  MSG004
C          MOVEATXT,313  MSG005
C          MOVEATXT,391  MSG006
C          MOVEATXT,469  MSG007
**
C          '↵'          LOKUPTXT,IX          81
C          ENDDO
**
C          ENDSR
*****
**
** UPDATE - UPDATE THE SUBFILE ENTRIES TO THE PHYSICAL FILE
**
*****
C          UPDATE      BEGSR
**          ~~~~~
C          MOVE MSG001  P$MSG1
C          MOVE MSG002  P$MSG2
C          MOVE MSG003  P$MSG3
C          MOVE MSG004  P$MSG4
C          MOVE MSG005  P$MSG5
C          MOVE MSG006  P$MSG6
C          MOVE MSG007  P$MSG7
**
C          ENDSR
*****
**
** ,  SETUP - SETUP INITIAL PROGRAM VALUES
**
*****

```

```

C          ENDIF          --+ |
C          ENDIF          -----+
C          COLOUR        WHEQ 'PNK'
C          REVIMG        IFEQ '0'          -----+
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'38'    TXT,IX           |
C          ELSE          --< |
C          MOVE X'3C'    TXT,IX           |
C          ENDIF        --+ |
C          ELSE          -----< |
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'39'    TXT,IX           |
C          ELSE          --< |
C          MOVE X'3D'    TXT,IX           |
C          ENDIF        --+ |
C          ENDIF        -----+
C          COLOUR        WHEQ 'RED'
C          REVIMG        IFEQ '0'          -----+
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'28'    TXT,IX           |
C          ELSE          --< |
C          MOVE X'2C'    TXT,IX           |
C          ENDIF        --+ |
C          ELSE          -----< |
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'2B'    TXT,IX           |
C          ELSE          --< |
C          MOVE X'2D'    TXT,IX           |
C          ENDIF        --+ |
C          ENDIF        -----+
C          COLOUR        WHEQ 'TRQ'
C          REVIMG        IFEQ '0'          -----+
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'30'    TXT,IX           |
C          ELSE          --< |
C          MOVE X'34'    TXT,IX           |
C          ENDIF        --+ |
C          ELSE          -----< |
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'31'    TXT,IX           |
C          ELSE          --< |
C          MOVE X'35'    TXT,IX           |
C          ENDIF        --+ |
C          ENDIF        -----+
C          COLOUR        WHEQ 'WHT'
C          REVIMG        IFEQ '0'          -----+
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'22'    TXT,IX           |
C          ELSE          --< |
C          MOVE X'26'    TXT,IX           |
C          ENDIF        --+ |
C          ELSE          -----< |
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'23'    TXT,IX           |
C          ELSE          --< |
C          MOVE '-'      TXT,IX           |

```

```

C          ADD 78          IX
C          MOVEAMSG004    TXT,IX
C          ADD 78          IX
C          MOVEAMSG005    TXT,IX
C          ADD 78          IX
C          MOVEAMSG006    TXT,IX
C          ADD 78          IX
C          MOVEAMSG007    TXT,IX
**
** Look for the first attribute tag...
**
C          MOVE 'NO '      F$TAGS 3
C          Z-ADD001        IX
C          ' '            LOKUPTXT,IX          81
**
** If an attribute tag was found then we need to display SCREEN2
and search
** for more tags...
**
C          *IN81          DOWEQ*ON
C          *IN03          OREQ *ON
**
C          MOVE 'YES'      F$TAGS
C          EXFMTSCREEN2
C          *IN03          IFNE *ON
**
** Replace the attribute tag with the appropriate Hex code...
**
C          SELEC
C          COLOUR        WHEQ 'BLU'
C          REVIMG        IFEQ '0'          -----+
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'3A'     TXT,IX          | |
C          ELSE          --< |
C          MOVE X'3E'     TXT,IX          | |
C          ENDIF        --+ |
C          ELSE          -----<
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'3B'     TXT,IX          | |
C          ELSE          --< |
C          MOVE ' '      TXT,IX          | | <-
leave as is
C          ENDIF        --+ |
C          ENDIF        -----+
C          COLOUR        WHEQ 'GRN'
C          REVIMG        IFEQ '0'          -----+
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'20'     TXT,IX          | |
C          ELSE          --< |
C          MOVE X'24'     TXT,IX          | |
C          ENDIF        --+ |
C          ELSE          -----<
C          UNDLIN        IFEQ '0'          --+ |
C          MOVE X'21'     TXT,IX          | |
C          ELSE          --< |
C          MOVE X'25'     TXT,IX          | |

```

```

C          ENTER      OREQ *ON
C          F$TAGS     ANDEQ'NO '
**
** If screen initialisation is required...
**
C          INIT       IFEQ *ON
**
** Populate the screen fields with the input parameters...
**
C          MOVE P$MSG1   MSG001
C          MOVE P$MSG2   MSG002
C          MOVE P$MSG3   MSG003
C          MOVE P$MSG4   MSG004
C          MOVE P$MSG5   MSG005
C          MOVE P$MSG6   MSG006
C          MOVE P$MSG7   MSG007
C          MOVE *OFF     INIT
C          ENDF
**
** Display the screen...
**
C          EXFMTSCREEN1
**
** Process reponse from display...
**
C          *IN03       IFNE *ON
**
** If Help was requested...
**
**
** If Refresh was requested...
**
C          *IN05       IFEQ *ON
C          MOVE *ON     INIT
C          ELSE
C          EXSR PRCDTL
C          ENDF
C          ENDF
C          ENDDO
**
C          ENDSR
*****
**
**      PRCDTL - PROCESS THE DETAIL SCREEN
**
*****
C          PRCDTL     BEGSR
**
**      ~~~~~
** Prepare to search for any attribute tags in the message text...
**
C          Z-ADD001     IX      30
C          MOVEAMSG001  TXT,IX
C          ADD 78       IX
C          MOVEAMSG002  TXT,IX
C          ADD 78       IX
C          MOVEAMSG003  TXT,IX

```

```

I                                     261 270 FMTNAM
I                                     369 369 ENTER
I                                     B 370 3710CURPOS
I          SDS
I                                     *PROGRAM PGM
I                                     244 253 WRKSTN
I                                     254 263 USER
I                                     264 2690JOB#
*****
C          *ENTRY      PLIST
C                                     P$MSG1 78
C                                     P$MSG2 78
C                                     P$MSG3 78
C                                     P$MSG4 78
C                                     P$MSG5 78
C                                     P$MSG6 78
C                                     P$MSG7 78
C                                     P$EXIT 1
**
** Initialise all relevant fields.
**
C          EXSR SETUP
**
** Perform the main-line code until F3 is pressed...
**
C          *IN03      DOUEQ*ON
C          ENTER      OREQ *ON
C          F$TAGS     ANDEQ'NO '
**
C          EXSR MNTDTL
**
** If *IN03 wasn't pressed then update the file...
**
C          *IN03      IFNE *ON
C                                     EXSR UPDATE
C                                     ENDIF
C                                     ENDDO
**
** Exit the routine, closing all files...
**
C          MOVE *IN03      P$EXIT
C          MOVE '1'        *INLR
C          RETRN
*****
**
**      MNTDTL - MAINTAIN DETAIL SCREEN (SUBFILE)
**
*****
C          MNTDTL      BEGSR
**
          ~~~~~
** Repeat this section until either F3 (Exit) is pressed, or the
** Enter key is pressed and no errors are detected...
**
C          MOVE *ON      INIT
**
C          *IN03      DOUEQ*ON

```

```

A              7  4'Sign-on messages . . . . .
. . . . .':
A              17  5'Colour . . . . .
. . . . .':
A          COLOUR      3  B 17 39VALUES('BLU' 'GRN' 'PNK'
'RED' 'TRQ-
A              ' 'WHT' 'YLW')
A              19  5'Reverse Image . . . . .
. . . . .':
A              21  5'Underline . . . . .
. . . . .':
A          REVIMG      1  B 19 39VALUES('0' '1')
A          UNDLIN      1  B 21 39VALUES('0' '1')
A              19  42'1=Yes, 0=No'
A              21  42'1=Yes, 0=No'
A              19  56'Note: Both Reverse Image'
A              20  56'and Underline cannot be '
A              21  56'set for'
A              21  64'BLU,'
A              COLOR(BLU)
A              21  69'WHT'
A              COLOR(WHT)
A              21  75'YLW'
A              COLOR(YLW)
A              21  73'&'
A              17  43'BLU'
A              COLOR(BLU)
A              17  47'GRN'
A              17  51'PNK'
A              COLOR(PNK)
A              17  55'RED'
A              COLOR(RED)
A              17  59'TRQ'
A              COLOR(TRQ)
A              17  63'WHT'
A              COLOR(WHT)
A              17  67'YLW'
A              COLOR(YLW)

```

SGNR300 - Message Maintenance Program

SGNR300 is an RPG program that enables the users to use display file SGNS300 to manually maintain the sign-on display messages. The program accepts seven input parameters, containing the current sign-on display messages and passes back the same seven parameters plus an exit flag to identify whether or not the messages need to be updated.

The source is very self-explanatory, and is shown below:

```

FSGNS300 CF  E          WORKSTN          KINFDS DSPFDS
F          SGNF300A          KRENAMESCREEN1
F          SGNF300B          KRENAMESCREEN2
*****
E          TXT          547  1
*****
IDSPFDS          DS
I          *FILE          FILNAM

```

```

      A                               18  2'point you want to start or
end tex-
      A                               t formatting.  The Apply
Attributes-
      A                               Screen  '
      A                               COLOR(BLU)
      A                               19  2'will be displayed in turn
for each-
      A                               tag mark entered.
-
      A                               '
      A                               COLOR(BLU)
      A                               R SGNF300B
      A                               CF03(03 'Exit')
      A                               CF05(05 'Refresh')
      A                               CF08(08 'Next')
      A                               CF09(09 'WRKMSGF')
      A                               1 32'ASTRADYNE (UK) LTD'
      A                               COLOR(WHT)
      A                               DSPATR(HI)
      A                               DSPATR(UL)
      A                               2 27'Edit Sign-on Display
Messages'
      A                               4  2'Select the attributes you
wish to -
      A                               use at the point marked in
the text-
      A                               below.  '
      A                               COLOR(BLU)
      A                               23  2' F3Exit'
      A                               COLOR(BLU)
      A                               23 11'F5=Refresh'
      A                               COLOR(BLU)
      A                               1  2'SGNS300'
      A                               COLOR(BLU)
      A                               2  2'ATTRIBUTES'
      A                               COLOR(BLU)
      A                               2 73TIME
      A                               COLOR(BLU)
      A                               1 73DATE
      A                               EDTCDE(Y)
      A                               COLOR(BLU)
      A                               MSG001          78A  O  9  2DSPATR(HI)
      A                               MSG002          78A  O 10  2
      A                               MSG003          78A  O 11  2
      A                               MSG004          78A  O 12  2
      A                               MSG005          78A  O 13  2
      A                               MSG006          78A  O 14  2
      A                               MSG007          78A  O 15  2
      A                               5  2'Press Enter to accept the
attribut-
      A                               e and move on to the next
tag: -
      A                               '
      A                               COLOR(BLU)

```

```

A          DSPSIZ(24 80 *DS3)
A          R SGNF300A
A          CF03(03 'Exit')
A          CF05(05 'Refresh')
A          CF08(08 'Next')
A          CF09(09 'WRKMSGF')
A          1 32'ASTRADYNE (UK) LTD'
A          COLOR(WHT)
A          DSPATR(HI)
A          DSPATR(UL)
A          2 27'Edit Sign-on Display
Messages'
A          4 2'Type the details you wish
to appea-
A          r on the sign-on display in
the spa-
A          ce below.'
A          COLOR(BLU)
A          23 2'F3=Exit'
A          COLOR(BLU)
A          23 11'F5=Refresh'
A          COLOR(BLU)
A          1 2'SGNS300'
A          COLOR(BLU)
A          2 2'SYSNEWS'
A          COLOR(BLU)
A          2 73TIME
A          COLOR(BLU)
A          1 73DATE
A          EDTCDE(Y)
A          COLOR(BLU)
A          MSG001          78A  B  9 2DSPATR(HI)
A          COLOR(WHT)
A          CHECK(LC)
A          MSG002          78A  B 10 2CHECK(LC)
A          MSG003          78A  B 11 2CHECK(LC)
A          MSG004          78A  B 12 2CHECK(LC)
A          MSG005          78A  B 13 2CHECK(LC)
A          MSG006          78A  B 14 2CHECK(LC)
A          MSG007          78A  B 15 2CHECK(LC)
A          5 2'The details will appear
instantly -
A          once the Enter key has been
pressed-
A          .'
A          COLOR(BLU)
A          7 4'Sign-on messages . . . . .
. . . .'
A          17 2'Display attributes,
colours, etc, -
A          can be changed by placing
a'
A          COLOR(BLU)
A          17 64'-'
A          17 66'mark at each '
A          COLOR(BLU)

```

```

DCL          VAR(&SAV003) TYPE(*CHAR) LEN(78)
DCL          VAR(&SAV004) TYPE(*CHAR) LEN(78)
DCL          VAR(&SAV005) TYPE(*CHAR) LEN(78)
DCL          VAR(&SAV006) TYPE(*CHAR) LEN(78)
DCL          VAR(&SAV007) TYPE(*CHAR) LEN(78)
DCL          VAR(&EXIT)   TYPE(*CHAR) LEN(1)

MONMSG      MSGID(CPF0000)

/* 0010 - Initialise Screen with values... */

RTVMSG      MSGID(SGN0001) MSGF(AUKMSGF) MSG(&SAV001)
RTVMSG      MSGID(SGN0002) MSGF(AUKMSGF) MSG(&SAV002)
RTVMSG      MSGID(SGN0003) MSGF(AUKMSGF) MSG(&SAV003)
RTVMSG      MSGID(SGN0004) MSGF(AUKMSGF) MSG(&SAV004)
RTVMSG      MSGID(SGN0005) MSGF(AUKMSGF) MSG(&SAV005)
RTVMSG      MSGID(SGN0006) MSGF(AUKMSGF) MSG(&SAV006)
RTVMSG      MSGID(SGN0007) MSGF(AUKMSGF) MSG(&SAV007)

/* 0020 - Display the screen... */

TAG0020:    CALL          PGM(SGNS300) PARM(&SAV001 &SAV002 &SAV003 +
&SAV004 &SAV005 &SAV006 &SAV007 &EXIT)

/* 0030 - Process the function keys... */

IF          COND(&EXIT *EQ '1') THEN(GOTO CMDLBL(TAG9999))

/* 0040 - Update the descriptions... */

TAG0040:    CHGMSGD      MSGID(SGN0001) MSGF(AUKMSGF) MSG(&SAV001)
CHGMSGD      MSGID(SGN0002) MSGF(AUKMSGF) MSG(&SAV002)
CHGMSGD      MSGID(SGN0003) MSGF(AUKMSGF) MSG(&SAV003)
CHGMSGD      MSGID(SGN0004) MSGF(AUKMSGF) MSG(&SAV004)
CHGMSGD      MSGID(SGN0005) MSGF(AUKMSGF) MSG(&SAV005)
CHGMSGD      MSGID(SGN0006) MSGF(AUKMSGF) MSG(&SAV006)
CHGMSGD      MSGID(SGN0007) MSGF(AUKMSGF) MSG(&SAV007)

TAG9999:    ENDPGM

```

SGNS300 - Display File

SGNS300 is the display file for the main maintenance program and consists of two screens. Format SGNF300A is the actual maintenance screen and provides seven lines of input for the sign-on display message, unfortunately there is no automatic word wrap on these lines, but there is the facility to specify some formatting attributes against the text.

To specify formatting attributes for colour, reverse image and underline, you simply need to flag the start and end of the text using the "-" character. When attributes are applied, they continue until either another attribute code is found, or the end of the field.

Format SGNF300B is the attribute definition screen. Colours are specified by supplying the standard three character colour code, and the reverse image and underline attributes are set by specifying a "1" in the appropriate field, and a "0" to reset them. The source for SGNS300 is shown below:



Figure 1: Edit Sign-On Display Message Command Display

The EDTSGNMSG command consists of four objects; the command definition, the command processing program (SGNC300), an RPG message maintenance program (SGNR300) and a display file (SGNS300).

EDTSGNMSG - Command Definition

The EDTSGNMSG command definition could not be any simpler as there are no parameters for this command. The source for the command is shown below:

```
EDTSGNMSG:  CMD          PROMPT('Edit Sign-on Display Messages')
```

When compiling the source, ensure that the "Program to Process Command" parameter (PGM) on the CRTCMD command is set to point to program SGNC300. For example:

```
CRTCMD CMD(library_name/EDTSGNMSG) PGM(library_name/SGNC300)
```

SGNC300 - Command Processing Program

SGNC300 is the CL program that retrieves the current message settings before calling the message maintenance program, SGNR300, and then updating the sign-on display messages with any changes made. The source for SGNC300 is shown below:

```
PGM

/* 0000 - Variable Declarations... */

          DCL          VAR(&SAV001) TYPE(*CHAR) LEN(78)
          DCL          VAR(&SAV002) TYPE(*CHAR) LEN(78)
```

Technical Document

Document No: 04-01006
Document Title: Modifying QDSIGNON and Getting the Message Across (Part 2)
Category: Hints, Tips & FAQ
Functional Area: Operating System
OS/400 Release: v4r3 and above

Document Description:

In part one of this series of articles, we modified the QDSIGNON sign-on display to provide space for system notices and announcements. In this, the second article in the series, we will create a new tool that will allow us to maintain the text of the notices and announcements displayed on the sign-on screen. Up until now, we have been able to modify the messages using the WRKMSGF, Work with Message File, command which is fine for general use.

The new EDTSGNMSG, Edit Sign-on Message, command that we will be creating will provide us with a WYSISWYG interpretation of the message and provide the ability to specify colour, reverse image and underline attributes. Figure 1 shows a screen shot of the new command interface.