

## Technical Document

**Document No:** 04-01007  
**Document Title:** Modifying QDSIGNON and Getting the Message Across (Part 3)  
**Category:** Hints, Tips & FAQ  
**Functional Area:** Operating System  
**OS/400 Release:** v4r3 and above

### Document Description:

In the previous two articles in this series, we have looked at how the QDSIGNON AS/400 sign-on display can be modified to allow variable messages to be displayed, and provided a simple display program enabling the messages to be maintained.

It would be useful, though, if we were able to schedule particular “news” messages to be displayed on specific days – for example, a message that the system will be getting upgraded at the weekend could be scheduled in advance to be displayed for the three days prior to the upgrade taking place.

Another useful facility would be setting standard messages during particular system events, for example “backup is in progress” during the overnight saves. This article introduces a number of commands and programs that can be used to achieve this.

### **SGNP200A – Physical File**

To provide the functionality we need to store the messages that we want to schedule in a separate physical file, and file **SGNP200A** is provided for this purpose with the following layout:

```
.....A.....T.Name+++++RLen++TDpB.....Functions+++++
A          R SGNR200A
A          Q$TYPE          3          TEXT('MSG TYPE')
A          Q$FROM          8  0      TEXT('FROM DATE')
A          Q$TO            8  0      TEXT('TO DATE')
A          Q$FREQ          1          TEXT('FREQUENCY')
A          Q$SGN1          78          TEXT('MESSAGE 1')
A          Q$SGN2          78          TEXT('MESSAGE 2')
A          Q$SGN3          78          TEXT('MESSAGE 3')
A          Q$SGN4          78          TEXT('MESSAGE 4')
A          Q$SGN5          78          TEXT('MESSAGE 5')
A          Q$SGN6          78          TEXT('MESSAGE 6')
**
A          K Q$TYPE
A          K Q$FROM
```

A	K Q\$TO
A	K Q\$FREQ

Field **Q\$TYPE** identifies the type of message (record) in the file and has one of the following three values:

- +++** Control record – this identifies the last “Random” message used.
- ALR** Alert Message – this identifies notices that are to be scheduled on or between certain dates.
- RND** Random Message – this identifies “Random” messages. These are messages that are just used to provide a daily changing message for those days when there is no scheduled alert. These may be jokes, fascinating facts, quiz questions, etc. The idea is to provide a changing message so that the user gets used to reading the messages on the sign-on display.

Fields **Q\$FROM** and **Q\$TO** hold the from and to dates that an alert message is scheduled for. For the control record these values should be zero, whilst for random messages the values should hold a sequence number, e.g. 1, 2, 3, etc.

Field **Q\$FREQ** identifies the frequency, if any, at which messages are recycle. For example, in our shop we display a message about password confidentiality every 30 days and provide a “countdown to Christmas” as a reminder of how many shopping days are left to buy that must-have gift.

The following Frequency codes are defined, but others can be used by a simple modification to program **SGNR320**:

Code	Description
A	10 day intervals
B	20 day intervals
C	30 day intervals
M	Monthly intervals (e.g. 1 <sup>st</sup> of month, last day of month, etc)
Q	Quarterly intervals
S	Semi (Half) Yearly intervals
Y	Yearly intervals
0	Not to be cycled
1	7 days interval
2	14 days interval
3	21 days interval
4	28 days interval

Fields **Q\$SGN1** through **Q\$SGN6** hold the message to be displayed on the sign-on screen. Note, that we only store six lines while the sign-on display has space for seven. The reason

for this is that the top message line on the sign-on display will be used to show the date, leaving the remaining six lines free for use.

Logical file **SGNL200A** is based on this file and omits all records with a Frequency code of "0".

### ***CYCSGNMSG –Command Definition***

The **CYCSGNMSG** is a simple, no parameter, command front end for running program **SGNR320** and cycling the sign-on display messages.

### ***SGNR320 – Command Processing Program***

Program **SGNR320** is an RPG program that will read through logical file **SGNL200A**, identifying which alert messages are to be recycled and updating their scheduled dates. The source code for this program is shown below:

```
FSGNL200AUF  E          K          DISK
*****
E              FRQ      9  36  1  DYS      3  0
E              DPM      12  12  2  0
*****
I              DS
I              1      20CC
I              3      40YY
I              1      40YYYY
I              5      60MM
I              7      80DD
I              1      80DATE
I              DS
I              1      20HRS
I              3      40MIN
I              5      60SEC
I              1      6  SYTIME
I              7      8  DY
I              9     10  MT
I              11     12  YR
I              7     12  SYDATE
I              1     120TIMSTM
I              SDS
I              199  200 CT
*****
**
** Initialise the program work fields...
**
C              TIME          TIMSTM
C              MOVE DY      DD
C              MOVE MT      MM
C              MOVE CT      CC
C              MOVE YR      YY
C              Z-ADDDATE     TODAY  80
**
** Position to the start of the file...
**
```

```

C          *LOVAL      SETLLSGNL200A
**
** Read through the file, processing all records that do not have a
** frequency code of '0' (these are excluded by the Logical File).
**
C          READ SGNL200A          91
B01 C          *IN91      DOWNE*ON
**
** We only need to process records where both the scheduled from a
** scheduled to dates have both passed...
**
B02 C          TODAY      IFGT Q$FROM
C          TODAY      ANDGTQ$TO
**
** Lookup the number of days to be added when rescheduling the message...
**
C          Z-ADD1          IX          20
C          Q$FREQ      LOKUPFRQ,IX          92
**
C          Z-ADD*ZEROS      #DAYS      30
C          *IN92      IFEQ *ON
C          Z-ADDDYS,IX      #DAYS
C          ENDIF
**
** Reschedule the FROM date...
**
C          Z-ADDQ$FROM      #DATE      80
C          EXSR EXTDTE
C          Z-ADD#DATE      Q$FROM
**
** Reschedule the TO date...
**
C          Z-ADDQ$TO      #DATE
C          EXSR EXTDTE
C          Z-ADD#DATE      Q$TO
**
C          UPDATSGNR200A
C          ENDIF
**
** Read the next record and close the loop...
**
C          READ SGNL200A          91
C          ENDDO
**
** Exit the program...
**
C          MOVE *ON          *INLR
C          RETRN
*****
**
** EXTDTTE - Increase yyyymmdd date by a given number of days
**
*****

```

```

C          EXTDTE      BEGSR
**          -----
** Add the number of days to the date depending on the value passed
** Some values have special meanings, e.g. 365 = 1 year, 31 = 1 month,
** 90 = 3 months, 180 = 6 months...
**
C          MOVE #DATE      DATE
C          SELEC
C          #DAYS      WHEQ 000
**                      Do nothing
C          #DAYS      WHEQ 365
C          ADD 1          YYYY
C          #DAYS      WHEQ 180
C          ADD 6          MM
C          #DAYS      WHEQ 90
C          ADD 3          MM
C          #DAYS      WHEQ 31
C          ADD 1          MM
C          OTHER
C          ADD #DAYS      DD
C          ENDSL
**
** Now the days/months have been added, validate the date and ensure it
** remains as a valid date...
**
** First the month portion of the date...
**
C          MM          DOWGT12
C          SUB 12      MM
C          ADD 1          YYYY
C          ENDDO
**
** Now the day portion...
**
C          DD          IFGT DPM,MM
**
** If the #DAYS value was a 'special' then the day should be the
** closest working day of the new month - e.g. 31/Jan + 3mths
** becomes 30/Apr and not 1/May...
**
C          #DAYS      IFEQ 31
C          #DAYS      OREQ 90
C          #DAYS      OREQ 180
C          #DAYS      OREQ 365
C          Z-ADDDPM,MM      DD
**
** Otherwise, the date falls in the following month and the
** month and (possibly) the year should be moved on by 1...
** (possibly) the year should be moved on by 1...
**
C          ELSE
**
C          SUB DPM,MM      DD

```

```

C          ADD 1          MM
C          MM          IFGT 12          ---+          |
C          SUB 12        MM          |          |
C          ADD 1          YYYY          |          |
C          ENDIF          ---+          |
C          ENDIF          -----+          |
C          ENDIF          -----+
**
C          Z-ADDDATE      #DATE
**
C          ENDSR
*****
**
A010B020C030D000E000F000G000H000I000
J000K000L000M031N000O000P000Q090R000
S180T000U000V000W000X000Y365Z0000000
100720143021402850006000700080009000
**
312831303130313130313031

```

Now we have the mechanism for recycling scheduled messages, we need another mechanism for selecting the appropriate message to be displayed. For this we can use the **DFTSGNMSG**, Default Sign-on Message.

### ***DFTSGNONMSG – Command Definition***

The **DFTSGNONMSG** command has a single parameter, **MODE**, which identifies which message is to be displayed on the sign-on display. The source code for the command is shown below:

```

DFTSGNMSG:  CMD          PROMPT('Set Default Sign-On Messages')

          PARM          KWD(MODE) TYPE(*CHAR) LEN(10) RSTD(*YES) +
                      DFT(*DEFAULT) VALUES(*RESTORE *BACKUP +
                      *LOCKOUT *DELAYED *DATAMART *DEFAULT) +
                      PROMPT('Sign-on Message Mode')

```

The default value for the **MODE** parameter is **\*DEFAULT** which will cause the command processing program to determine whether or not there is a scheduled alert message to be displayed, and if not to pick up the next random message for use. Random messages are automatically rescheduled following use.

The remaining allowed values for the **MODE** parameter have the following meanings:

- \*BACKUP** Provides a standard message that the backups are in process and that production applications are unavailable.
- \*DELAYED** Provides a standard message that the overnight processing is still in progress.
- \*DATAMART** Provides a standard message that the datamart extract is still running and that data cubes can still be accessed using the previous days data.

- \*LOCKOUT** Provides a standard message that users cannot sign on due to system maintenance. This parameter option ties in with a separate **LOCKOUT** command that was developed to provide a means of preventing users from logging on to the system without a) disabling their profiles and b) closing down the QINTER subsystem.
- \*RESTORE** This parameter option will restore the Sign-on Display messages to their original values prior to the **\*BACKUP**, **\*DELAYED**, **\*DATAMART** and **\*LOCKOUT** options being processed. Data areas are used to store the saved message details.

Additional parameters can be added quite simply by modifying the command definition and the command processing programs **SGNC310** and **SGNR310**. The source code for these two programs is shown below:

### ***SGNC310 – Command Processing Program***

```

PGM          PARM(&MODE)

/* 0000 - Variable Declarations...                                     */

      DCL          VAR(&MODE)      TYPE(*CHAR) LEN(10)

      /* Valid Values: *RESTORE    - Restore to Saved Message */
      /*                  *BACKUP    - Backups in Process      */
      /*                  *LOCKOUT    - Lockout Command Enabled */
      /*                  *DELAYED    - Overnight Jobs Delayed  */
      /*                  *DATAMART    - Datamart Extract Delayed */
      /*                  *DEFAULT    - Default Messages        */

      DCL          VAR(&SAV001) TYPE(*CHAR) LEN(78)
      DCL          VAR(&SAV002) TYPE(*CHAR) LEN(78)
      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(&DATFMT) TYPE(*CHAR) LEN(3)

      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)

      RTVSYSVAL   SYSVAL(QDATFMT) RTNVAR(&DATFMT)

```

```

/* 0015 - If we are in *DEFAULT mode and the current message is      */
/*          one of the ALER messages, then don't change it - simply  */
/*          redisplay the message with the current date...          */

      IF          COND(&MODE *EQ '*DEFAULT') THEN(DO)

      IF          COND(%SST(&SAV002 8 18) *EQ 'SYSTEM +
                        UNAVAILABLE') THEN(DO)
      CHGVAR      VAR(&MODE) VALUE('*LOCKOUT')
      GOTO        CMDLBL(TAG0040)
      ENDDO

      IF          COND(%SST(&SAV002 27 13) *EQ 'SYSTEM +
                        BACKUP') THEN(DO)
      CHGVAR      VAR(&MODE) VALUE('*BACKUP')
      GOTO        CMDLBL(TAG0040)
      ENDDO

      IF          COND(%SST(&SAV002 9 9) *EQ 'OVERNIGHT') +
                        THEN(DO)
      CHGVAR      VAR(&MODE) VALUE('*DELAYED')
      GOTO        CMDLBL(TAG0040)
      ENDDO

      IF          COND(%SST(&SAV002 8 8) *EQ 'DATAMART') +
                        THEN(DO)
      CHGVAR      VAR(&MODE) VALUE('*DATAMART')
      GOTO        CMDLBL(TAG0040)
      ENDDO

      ENDDO

/* 0020 - Store the messages in an array so they can be restored    */
/*          later (but not if we are re-issuing an override without  */
/*          the original saved message having been restored first...  */

      IF          COND(&MODE *NE '*RESTORE') THEN(DO)
      IF          COND((%SST(&SAV002 08 18) *NE 'SYSTEM UNAVAILAB
                        *AND (%SST(&SAV002 27 13) *NE 'SYSTEM BACKUP')
                        *AND (%SST(&SAV002 09 09) *NE 'OVERNIGHT')
                        *AND (%SST(&SAV002 08 08) *NE 'DATAMART'))
                        THEN(DO)

      CHGDTAARA   DTAARA(SGNMSGLIN1) VALUE(&SAV001)
      CHGDTAARA   DTAARA(SGNMSGLIN2) VALUE(&SAV002)
      CHGDTAARA   DTAARA(SGNMSGLIN3) VALUE(&SAV003)
      CHGDTAARA   DTAARA(SGNMSGLIN4) VALUE(&SAV004)
      CHGDTAARA   DTAARA(SGNMSGLIN5) VALUE(&SAV005)
      CHGDTAARA   DTAARA(SGNMSGLIN6) VALUE(&SAV006)
      CHGDTAARA   DTAARA(SGNMSGLIN7) VALUE(&SAV007)
      ENDDO
      ENDDO

```



```

/* 0030 - If this is a restore then retrieve the data area contents */
/*          and store them in the &SAVxxx fields...                */

          IF          COND(&MODE *EQ '*RESTORE') THEN(DO)
RTVDTAARA  DTAARA(SGNMSGLIN1) RTNVAR(&SAV001)
RTVDTAARA  DTAARA(SGNMSGLIN2) RTNVAR(&SAV002)
RTVDTAARA  DTAARA(SGNMSGLIN3) RTNVAR(&SAV003)
RTVDTAARA  DTAARA(SGNMSGLIN4) RTNVAR(&SAV004)
RTVDTAARA  DTAARA(SGNMSGLIN5) RTNVAR(&SAV005)
RTVDTAARA  DTAARA(SGNMSGLIN6) RTNVAR(&SAV006)
RTVDTAARA  DTAARA(SGNMSGLIN7) RTNVAR(&SAV007)
          ENDDO

/* 0040 - Call the program to set the message defaults, depending */
/*          on the mode the program was called in...              */

TAG0040:    CALL      PGM(SG NR310) PARM(&MODE &SAV001 &SAV002 +
                                     &SAV003 &SAV004 &SAV005 &SAV006 &SAV007 +
                                     &DATFMT)

/* 0050 - Update the descriptions...                               */

          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)

/* 0060 - If we are in *DEFAULT mode then run the process to move */
/*          dates on for recurring messages...                      */

          IF          COND(&MODE *EQ '*DEFAULT') THEN(DO)
          CALL      PGM(SG NR320)
          ENDDO

TAG9999:    ENDPGM

```

### SGNR320 – Command Processing Program

```

FSGNP200AUF  E          K          DISK
*****
E              MTH      1  12 78          Date Headings
E              BUP      1   5 78          Mode: *BACKUP
E              LCK      1   5 78          Mode: *LOCKOUT
E              DLY      1   5 78          Mode: *DELAYED
E              DMT      1   5 78          Mode: *DATAMART
E*
E              TXT              78  1
*****
I              DS
I              1      20HRS

```

```

I          3  40MIN
I          5  60SEC
I          1  6 SYTIME
I          7  8 DD
I          9 10 MM
I          9 100MX
I         11 12 YY
I          7 12 SYDATE
I          1 120TIMSTM
I          DS
I          1  2 CCC
I          3  4 YYY
I          5  6 MMM
I          7  8 DDD
I          1  80DATE8
I          DS
I          5 120LSTRND
I          1  78 CTL
I          SDS
I          199 200 CC
**
** Named constants for parameter checking...
**
I          '*RESTORE'      C      C$RST
I          '*BACKUP'       C      C$BKUP
I          '*LOCKOUT'      C      C$LOCK
I          '*DELAYED'      C      C$DLYD
I          '*DATAMART'     C      C$DATA
*****
C          *ENTRY      PLIST
C                      PARM          P$MODE 10
C                      PARM          P$MSG1 78
C                      PARM          P$MSG2 78
C                      PARM          P$MSG3 78
C                      PARM          P$MSG4 78
C                      PARM          P$MSG5 78
C                      PARM          P$MSG6 78
C                      PARM          P$MSG7 78
C                      PARM          P$DFMT  3
**
** Define work fields for Key Lists to the Sign-on Message file...
**
C          *LIKE      DEFN Q$TYPE      K$TYPE
C          *LIKE      DEFN Q$FROM      K$FROM
C          *LIKE      DEFN Q$TO        K$TO
C          *LIKE      DEFN Q$FREQ      K$FREQ
**
** Define the key lists for the different types of messages...
**
C          K$RND      KLIST
C                      KFLD          K$TYPE
C                      KFLD          K$FROM
C                      KFLD          K$TO

```

C		KFLD	K\$FREQ
**			
C	K\$ALR	KLIST	
C		KFLD	K\$TYPE
C		KFLD	K\$FROM
**			
**	Set up the first and last message to hold the date. Note, the		
**	line will only hold the date in *RESTORE mode if the parameter		
**	has a plus sign in position 3...		
**			
C		TIME	TIMSTM
**			
**	If the date format isn't in DMY (hey, I'm English) then we need		
**	to make sure the right values are picked up for the date to be		
**	formatted on the message line...		
**			
C		SELEC	
C	P\$DFMT	WHEQ 'MDY'	
C		MOVE MM	W\$002A 2
C		MOVE DD	MM
C		MOVE W\$002A	DD
C	P\$DFMT	WHEQ 'YMD'	
C		MOVE YY	W\$002A
C		MOVE DD	YY
C		MOVE W\$002A	DD
C		OTHER	
C		ENDSL	
**			
C		MOVEAMTH,MX	TXT
C		SELEC	
C	MX	WHEQ 01	
C		MOVEADD	TXT,32
C		MOVEACC	TXT,44
C		MOVEAYY	TXT,46
C	MX	WHEQ 02	
C		MOVEADD	TXT,32
C		MOVEACC	TXT,44
C		MOVEAYY	TXT,46
C	MX	WHEQ 03	
C		MOVEADD	TXT,33
C		MOVEACC	TXT,43
C		MOVEAYY	TXT,45
C	MX	WHEQ 04	
C		MOVEADD	TXT,33
C		MOVEACC	TXT,43
C		MOVEAYY	TXT,45
C	MX	WHEQ 05	
C		MOVEADD	TXT,34
C		MOVEACC	TXT,42
C		MOVEAYY	TXT,44
C	MX	WHEQ 06	
C		MOVEADD	TXT,34
C		MOVEACC	TXT,42

```

C          MOVEAYY          TXT,44
C          MX              WHEQ 07
C          MOVEADD          TXT,34
C          MOVEACC          TXT,42
C          MOVEAYY          TXT,44
C          MX              WHEQ 08
C          MOVEADD          TXT,33
C          MOVEACC          TXT,43
C          MOVEAYY          TXT,45
C          MX              WHEQ 09
C          MOVEADD          TXT,31
C          MOVEACC          TXT,45
C          MOVEAYY          TXT,47
C          MX              WHEQ 10
C          MOVEADD          TXT,32
C          MOVEACC          TXT,44
C          MOVEAYY          TXT,46
C          MX              WHEQ 11
C          MOVEADD          TXT,32
C          MOVEACC          TXT,44
C          MOVEAYY          TXT,46
C          MX              WHEQ 12
C          MOVEADD          TXT,32
C          MOVEACC          TXT,44
C          MOVEAYY          TXT,46
C          OTHER
C          ENDSL
**
C          MOVEATXT,1      P$MSG1
**
** Process according to the mode passed...
**
C          SELEC
C          P$MODE          WHEQ C$RST
C          EXSR RSTDFT
C          P$MODE          WHEQ C$BKUP
C          EXSR BACKUP
C          P$MODE          WHEQ C$LOCK
C          EXSR LCKOUT
C          P$MODE          WHEQ C$DLYD
C          EXSR DELAYD
C          P$MODE          WHEQ C$DATA
C          EXSR DTAMRT
C          OTHER
C          EXSR NONEWS
C          ENDSL
**
** Exit the program
**
C          MOVE *ON          *INLR
C          RETRN
C*****
C**   RSTDFT - Restore Default Messages          **

```

```

C*****
CSR          RSTDFT      BEGSR
C*
CSR          MOVEAP$MSG7      TXT,1
CSR          TXT,3      IFEQ '+'
CSR          MOVE P$MSG1      P$MSG7
CSR          ENDIF
C*
CSR          ENDSR
C*****
C**    BACKUP - Set Backup in Progress Messages      **
C*****
CSR          BACKUP      BEGSR
C*
CSR          MOVE BUP,1      P$MSG2
CSR          MOVE BUP,2      P$MSG3
CSR          MOVE BUP,3      P$MSG4
CSR          MOVE BUP,4      P$MSG5
CSR          MOVE BUP,5      P$MSG6
CSR          MOVE P$MSG1      P$MSG7
C*
CSR          ENDSR
C*****
C**    LCKOUT - Set Lockout Message      **
C*****
CSR          LCKOUT      BEGSR
C*
CSR          MOVE LCK,1      P$MSG2
CSR          MOVE LCK,2      P$MSG3
CSR          MOVE LCK,3      P$MSG4
CSR          MOVE LCK,4      P$MSG5
CSR          MOVE LCK,5      P$MSG6
CSR          MOVE P$MSG1      P$MSG7
C*
CSR          ENDSR
C*****
C**    DELAYD - Set Overnight Jobs Delayed Message      **
C*****
CSR          DELAYD      BEGSR
C*
CSR          MOVE DLY,1      P$MSG2
CSR          MOVE DLY,2      P$MSG3
CSR          MOVE DLY,3      P$MSG4
CSR          MOVE DLY,4      P$MSG5
CSR          MOVE DLY,5      P$MSG6
CSR          MOVE P$MSG1      P$MSG7
C*
CSR          ENDSR
C*****
C**    DTAMRT - Set Datamart Message      **
C*****
CSR          DTAMRT      BEGSR
C*

```

```

CSR          MOVE DMT,1      P$MSG2
CSR          MOVE DMT,2      P$MSG3
CSR          MOVE DMT,3      P$MSG4
CSR          MOVE DMT,4      P$MSG5
CSR          MOVE DMT,5      P$MSG6
CSR          MOVE P$MSG1     P$MSG7
C*
CSR          ENDSR
C*****
C**   NONEWS - No News Defined                               **
C*****
CSR          NONEWS      BEGSR
C*
CSR          MOVE *BLANKS    P$MSG2
CSR          MOVE *BLANKS    P$MSG3
CSR          MOVE *BLANKS    P$MSG4
CSR          MOVE *BLANKS    P$MSG5
CSR          MOVE *BLANKS    P$MSG6
CSR          MOVE P$MSG1     P$MSG7
C*
C* Check the Signon message file to see if there is an Alert messa
C* displayed for the current date...
C*
CSR          MOVE 'ALR'      K$TYPE
CSR          MOVE DD         DDD
CSR          MOVE MM         MMM
CSR          MOVE CC         CCC
CSR          MOVE YY         YYY
CSR          Z-ADDDATE8      K$FROM
CSR          K$ALR          SETGTSGNP200A
CSR          'ALR'          REDPESGNP200A          99
CSR          *IN99          IFNE *ON              -----+
CSR          Q$FROM          IFLE DATE8          -----+ |
CSR          Q$TO            ANDGEDATE8          | |
C*                                                                    | |
C* If a record is found then display that message on the          | |
C* sign-on display...                                            | |
C*                                                                    | |
CSR          MOVE Q$SGN1     P$MSG2          | |
CSR          MOVE Q$SGN2     P$MSG3          | |
CSR          MOVE Q$SGN3     P$MSG4          | |
CSR          MOVE Q$SGN4     P$MSG5          | |
CSR          MOVE Q$SGN5     P$MSG6          | |
CSR          Q$SGN6          IFNE ' +++'      --+ | |
CSR          MOVE Q$SGN6     P$MSG7          | |
CSR          ENDIF          --+ | |
C*                                                                    | |
C* Otherwise set the EOF indicator so that we can use the        | |
C* next "Random" thoughts/jokes/fact message available...        | |
C*                                                                    | |
CSR          ELSE          -----< |
CSR          MOVE *ON        *IN99          | |
CSR          ENDIF          -----+ |

```

```

CSR                                ENDIF                                -----+
C*
C* If the EOF indicator is set then pick up the next available "random"
C* message...
C*
CSR          *IN99          IFEQ *ON                                -----+
C*
C* First pick up the number of the last Random message
C* shown...
C*
CSR          '+++'          CHAINSGNP200A                          91
CSR                                MOVE Q$SGN1          CTL
C*
C* Now pick up the next Random message...
C*
CSR                                MOVE 'RND'          K$TYPE
CSR          LSTRND          ADD 1          K$FROM
CSR                                Z-ADDK$FROM          K$TO
CSR                                MOVE '0'          K$FREQ
C*
CSR          K$RND          CHAINSGNP200A                          91
C*
C* If no record was found then recycle from the first one
C* again...
C*
CSR          *IN91          IFEQ *ON                                ---+
CSR                                Z-ADD1          K$FROM
CSR                                Z-ADD1          K$TO
CSR          K$RND          CHAINSGNP200A                          91
CSR                                ENDIF                                ---+
C*
C* Update the display with the message picked up...
C*
CSR                                MOVE Q$SGN1          P$MSG2
CSR                                MOVE Q$SGN2          P$MSG3
CSR                                MOVE Q$SGN3          P$MSG4
CSR                                MOVE Q$SGN4          P$MSG5
CSR                                MOVE Q$SGN5          P$MSG6
CSR          Q$SGN6          IFNE '+++'                                ---+
CSR                                MOVE Q$SGN6          P$MSG7
CSR                                ENDIF                                ---+
C*
C* Now update the last Random message used control
C* Record...
C*
CSR                                Z-ADDK$FROM          LSTRND
CSR          '+++'          CHAINSGNP200A                          91
CSR                                MOVE CTL          Q$SGN1
CSR                                UPDATSGNR200A
C*
CSR                                ENDIF                                -----+
C*
CSR                                ENDSR

```

```

C*****
** MTH - MONTHLY HEADER (LINE 1 AND LINE 7)
++++++ AS/400 SYSTEM NEWS +++ xx JANUARY, xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx FEBRUARY xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx MARCH, xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx APRIL, xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx MAY, xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx JUNE xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx JULY xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx AUGUST xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx SEPTEMBER, xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx OCTOBER, xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx NOVEMBER xxxx +++ AS/400 SYSTEM NEWS ++++++
++++++ AS/400 SYSTEM NEWS +++ xx DECEMBER xxxx +++ AS/400 SYSTEM NEWS ++++++
** BUP - BACKUP MESSAGE DISPLAY
          *** SYSTEM BACKUPS IN PROGRESS ***
Please be aware that the system backups are in progress. As a result, the
production application systems are currently not available for use and you
should not attempt to access them.
          *** SYSTEM BACKUPS IN PROGRESS ***
** LCK - LOCKOUT MESSAGES
          *** SYSTEM UNAVAILABLE *** SYSTEM UNAVAILABLE *** SYSTEM UNAVAILABLE ***
Please be aware that the system is currently unavailable for use due to some
system maintenance that is taking place. We shall endeavour to return to
normal service as soon as possible. Thank you.
          *** SYSTEM UNAVAILABLE *** SYSTEM UNAVAILABLE *** SYSTEM UNAVAILABLE ***
** DLY - OVERNIGHT JOBS RUNNING LATE
          *** OVERNIGHT JOBS RUNNING LATE! **** OVERNIGHT JOBS RUNNING LATE! ***
Please be aware that the overnight processing had not reached a pre-defined
check point at 7:00am (UK) this morning and there may still be some overnight
processes waiting to run. IT are aware and are investigating the cause.
          *** OVERNIGHT JOBS RUNNING LATE! **** OVERNIGHT JOBS RUNNING LATE! ***
** DMT - DATAMART MESSAGES
          *** DATAMART EXTRACT NOT COMPLETE **** DATAMART EXTRACT NOT COMPLETE ***
Please be aware that the Datamart extract from the AS/400 did not complete in
time for the build process to run on the Datamart server. This means that
some of the data shown on the server will not include "yesterdays" sales.
          *** DATAMART EXTRACT NOT COMPLETE **** DATAMART EXTRACT NOT COMPLETE ***

```

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](http://www.astradyne-uk.com). Physical file SGNP200A contained within the save file includes existing alert and random message entries.