



## Generic Record Lock Program

Generic Record Lock Handling Program .....	2
Calling the lock handler .....	3
Test lock.....	4
Interactive mode.....	5
Batch Mode.....	6
RPG Code .....	9
DDS.....	12



## Generic Record Lock Handling Program

In order to avoid having a record lock condition hold up interactive and batch processes a program has been developed to manage simple record lock conditions. The program name is SWRCLKRI and the associated display file is SWRCLKDI.

For interactive jobs, the program will overlay the existing application screen with a pop-up style window informing the user of the error and listing the record number and the name of the job holding the lock. In addition the screen display will list the job number user name and job name of the application holding the lock. In addition the panel will show the name of the reporting program and the name of the file it is attempting to lock for update.

```

Work with Members Using PDM                                     MISDEVO

File . . . . . : .....
Library . . . . : RCLKR001 Record Access Error Screen . . : .
Type options, pres : A record is in use by another job on the :
2=Edit          3= : system. This denies access to a critical :
8=Display descrip : data file your program needs. :
Opt Member      T : The job which controls the above file: :
  ___ RPGLOCK    R : Record 1 in use by job 445812/SCROY/QPAD :
  C_  RPGLOCK2   R : EV0008. :
  ___ RPGLOCK3   R : :
  ___ RTVXML     R : Program requesting data: RPGLOCK2 :
  ___ SCFMSGPR   R : File name.....: MYNAMES :
  ___ SCFPROTOS  R : :
  ___ SCFSVR_PR  R : Press ENTER to attempt to get the record :
  ___ SCFTTEST   R : again. If the problem persists, call the :
Parameters or comm : user that has the record you need to :
==>                : determine when you can get access. :
F3=Exit           F : F3=Exit and bypass record update. :
F9=Retrieve        F : ..... :
  
```



## Calling the lock handler

The interactive portion of the program will wait for 60 seconds then return to the calling program, if no user input is detected. Otherwise the application will return on the ENTER key, or F3 being pressed.

```

*=====
* The message text comes from the program status DS as does
* the program name. ERRCMD will be returned with a '1' if
* the user pressed EXIT.
*---
C      ERPARM      PLIST
C              PARM              MSGTXT
C              PARM              PRGNAM
C              PARM              ERREXT
C              PARM              ERRCMD
C              PARM              ERRNAM
*
C              DOU      *IN12 = *OFF
C      1      CHAIN      MYNAMES              1112
*-----
* 11 on, no record found
* 12 on, record locked, the MSGID will be 'CPF5027'
* 11 + 12 off, record found and locked by this program
* If error in getting record, call error handling program
* If the user is allowed to bypass the update, set the
* ERREXT to 'Y' to allow an exit to the loop.
*-----
C              IF      *IN12 = *ON
C              MOVE    'Y'      ERREXT
C              MOVE    ERRFIL    ERRNAM
C              CALL    'SWRCLKRI'  ERPARM
*-----
* If user requested EXIT, do not retry access to record and
* record again. (#TRUE = '1') Set on *IN11 (record not lock)
* Set off *IN12 to fall through loop.
*-----
C              IF      ERRCMD = #TRUE
C              MOVE    *ON      #ERR
C              EVAL    *IN12 = *OFF
C              EVAL    *IN11 = *ON
C              ENDIF
C              ENDIF
C              ENDDO
*

```

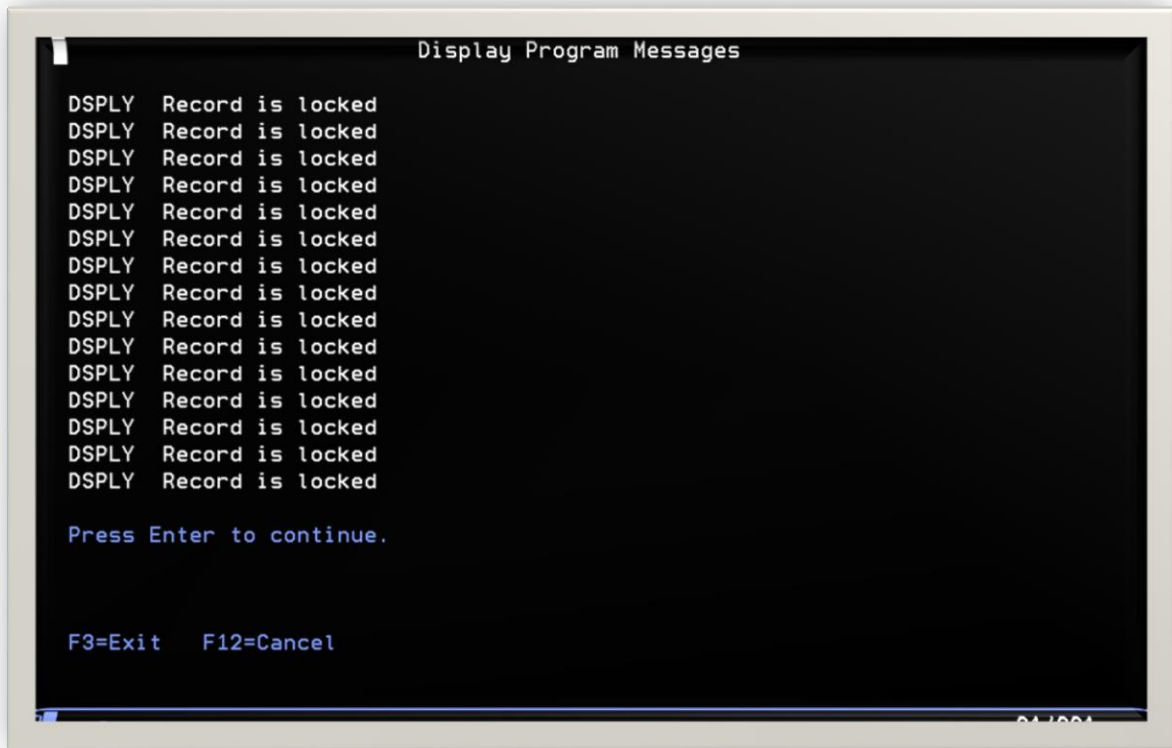
If the error exit parameter (ERREXT) is set to 'Y' F3 will be available to the user and allow the return of '1' for the ERRCMD parameter. If the value of 'N' is passed to the lock handler, F3 will not be available, and the only recourse for the user is to press the ENTER key, which returns a value of '0' for the ERRCMD parameter.

When the value of '0' is returned for the ERRCMD parameter, \*IN12 is not set off and the loop logic returns to the top of the loop and attempts to lock the record for update again. If the record lock has been resolved, the record is retrieved and the program resumes execution. Otherwise the lock handler is called again.



## Test lock

A unit test of the program has been created to test the lock handling process. RPGLOCK2 was created to lock a record in a test file and display a message on the screen to that effect, using the DISPLAY op code.



This session now has a record lock on the file MYNAMES. When the application is called from another session, the record lock is attempted on the same record of the file MYNAMES.



## Interactive mode

The lock is not granted to the second session, so the lock handler is invoked. This display is set to time-out with the WAITRCD parameter of the display set to 60 seconds. Whether the user actually presses the ENTER key or not, the program display will time out returning to the calling program. If the logic loop for the lock handler is in place, another attempt will be made to lock the record for update.

```

Work with Members Using PDM                                     MISDEVO

File . . . . . : .....
Library . . . . : RCLKR001 Record Access Error Screen . . : .
Type options, pres : A record is in use by another job on the :
2=Edit          3= : system. This denies access to a critical :
8=Display descrip : data file your program needs. :
Opt  Member      T : The job which controls the above file: :
  ___ RPGLOCK     R : Record 1 in use by job 445812/SCROY/QPAD :
  ___ RPGLOCK2    R : EV0008. :
  ___ RPGLOCK3    R : :
  ___ RTVXML      R : Program requesting data: RPGLOCK2 :
  ___ SCFMSGPR    R : File name.....: MYNAMES :
  ___ SCFPROTOS   R : :
  ___ SCFSVR_PR   R : Press ENTER to attempt to get the record :
  ___ SCFTTEST    R : again. If the problem persists, call the :
Parameters or comm : user that has the record you need to :
===> : determine when you can get access. :
F3=Exit          F : F3=Exit and bypass record update. :
F9=Retrieve      F : ..... :
  
```

If the record lock has been resolved, the calling program will automatically resume.



## Batch Mode

The record lock handler will also function in the batch mode. Since the program is bound to the Common Utility Service Program (UTCMMNRI), it uses the facilities of the service program to determine if the job is in the batch mode.

```

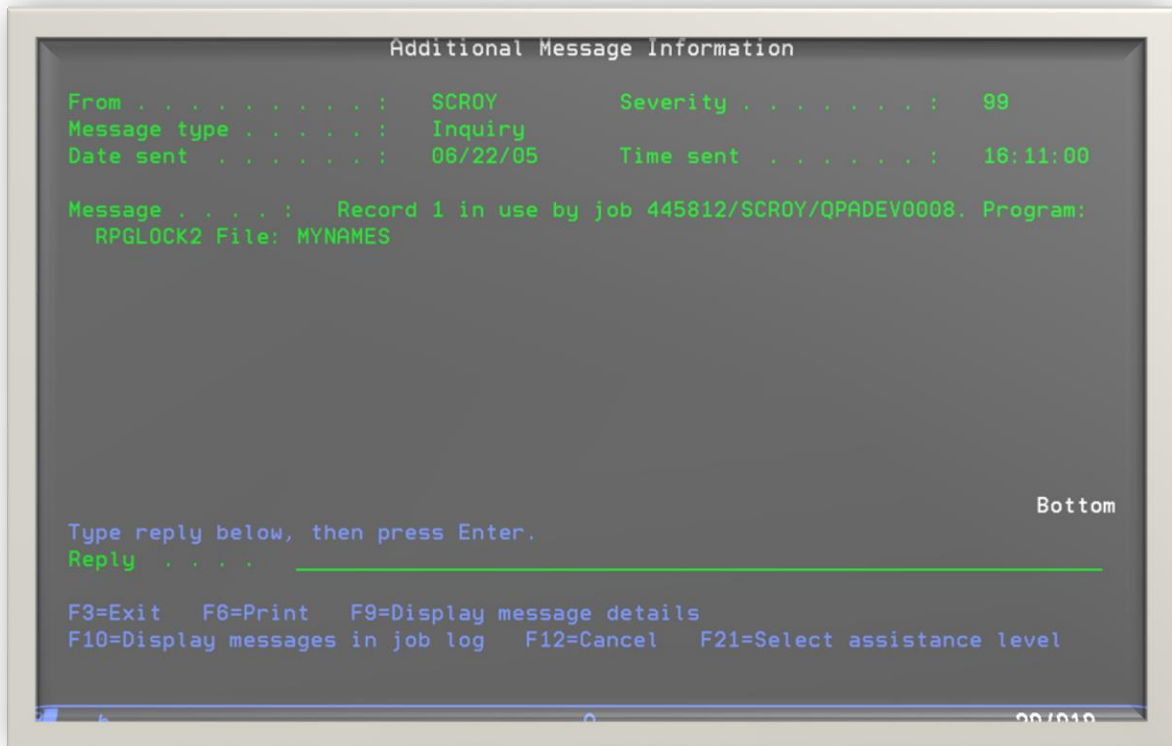
Work with Members Using PDM                                     MISDEV0
File . . . . . QRPGLSRC_
Library . . . . . SCROY          Position to . . . . .
Type options, press Enter.
 2=Edit      3=Copy  4=Delete 5=Display  6=Print  7=Rename
 8=Display description  9=Save 13=Change text 14=Compile 15=Create module...

Opt  Member      Type      Text
---  ---
___  RPGLCK        RPGLE    example of record lock
___  RPGLCK2       RPGLE    example of record lock
___  RPGLCK3       RPGLE    example of record lock
___  RTVXML        RPGLE
___  SCFMSGPR      RPGLE    SCF: Message prototypes
___  SCFPROTOS     RPGLE    SoftCode prototypes
___  SCFSVR_PR     RPGLE    SoftCode service prototypes
___  SCFTEST       RPGLE
More...

Parameters or command
==> SBMJOB CMD(CALL PGM(RPGLCK2)) JOB(LOCKTEST)
F3=Exit      F4=Prompt  F5=Refresh  F6=Create
F9=Retrieve   F10=Command entry  F23=More options  F24=More keys
  
```

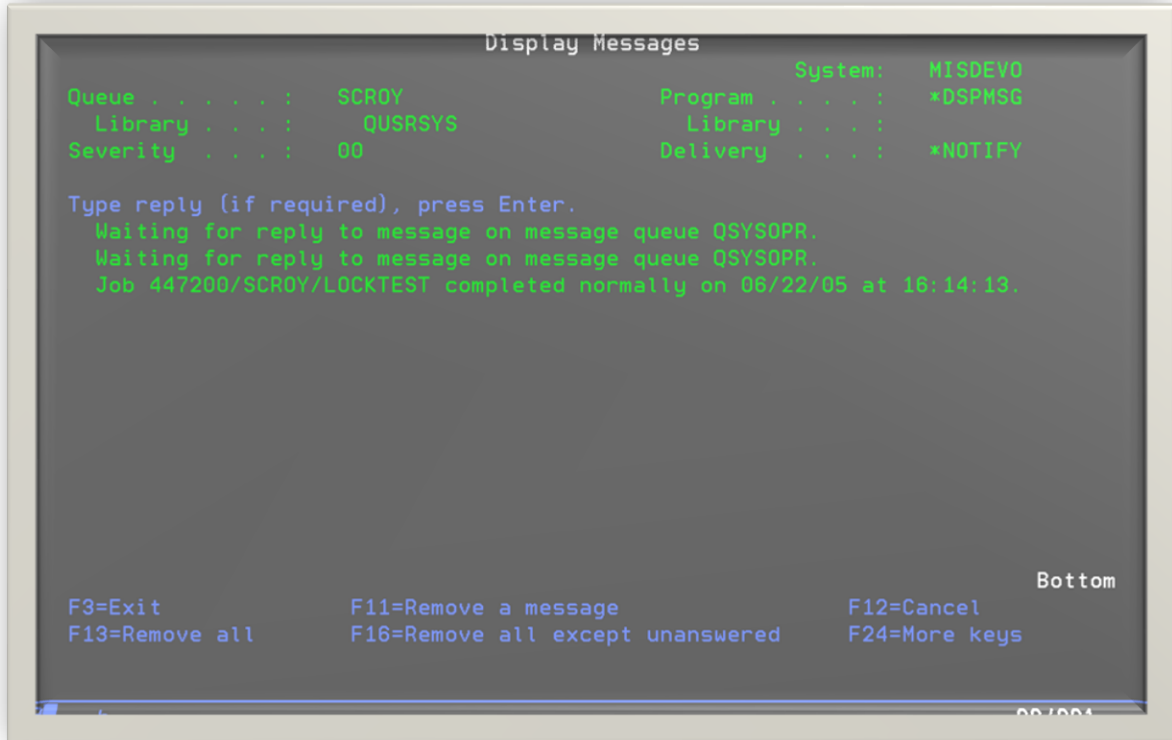


In the batch mode, the display file is not opened. Instead, using the system API, to send a message to the System Operator Message Queue, the program will wait for a response from the system operator. A response of 'C' will cause the lock handler to return a '1' for the ERRCMD parameter to bypass an update. Any other response will return '0' to attempt the record lock again.





If the operator fails to respond to the record lock message, the program will automatically respond with a retry after 5 minutes (300 seconds). If the record lock has been resolved the application will resume execution. If not, another message will be sent to the system operator.







# RPG Code

```

H/TITLE ** Standard error handler to record locks **
H DEBUG(*YES) BNDDIR('DPORDP_BD')
H OPTION(*SRCSTMT : *NODEBUGIO) DFTACTGRP(*NO) ACTGRP('QILE')
*****
* PROGRAM NAME - SWRCLKRI *
*
* FUNCTION - This program will function as the standard *
* record lock monitor for interactive and batch *
* programs. Interactively, it will display a *
* pop-up window to the user, with lock information. *
* In batch mode, the program will send a message *
* to the system operator message queue. *
*
* PROGRAMMER - STEVE CROY 04/12/2006 *
*****
** INDICATOR USAGE **
* ** ON ** ** OFF **
* 50 -- ALLOW FUNCTION KEY TO BY-PASS UPDATE
* F3 -- EXIT
*
* ** ERROR INDICATORS **
*
*****
* MODIFICATION LOG
*
* DATE PROGRAMMER DESCRIPTION
*
*****
FSWRCLKDI CF E WORKSTN USROPN
F MAXDEV(*FILE)
F INFDS(DSPDS)
D CF E DS EXTNAME(SWKEYSP) qualified
D PGMDS ESDS EXTNAME(SWPSTSP)
D DSPDS E DS EXTNAME(SWDSFPF)
Function keys
Pgm status map
Display INFDS

/copy xpssrc/qrpglesrc,utcmnn_pr

D SWRCLKRI PR
D CPFMessage 80
D CallingPgm 10
D ExitAllowed 1
D BypassLock 1
D FileLocked 8

D SWRCLKRI PI
D CPFMessage 80
D CallingPgm 10
D ExitAllowed 1
D BypassLock 1
D FileLocked 8

D QMHSNDM PR ExtPgm('QMHSNDM')
D MsgID 7A const
D QualMsgF 20A const
D MsgTxt 32767A const options(*varsize)
D MsgTxtLen 10I 0 const
D MsgType 10A const
D MsgQueues 20A const dim(50) options(*varsize)
D NumQueues 10I 0 const
D RpyQueue 20A const
D MsgKey 4A const
D ErrorCode 8000A options(*varsize)
D CCSID 10I 0 const options(*nopass)

D QMHRCVPM PR ExtPgm('QMHRCVPM')
D MsgInfo 32767A options(*varsize)
D MsgInfoLen 10I 0 const
D Format 8A const
D StackEntry 10A const
D StackCount 10I 0 const
D MsgType 10A const
D MsgKey 4A const
D WaitTime 10I 0 const
D MsgAction 10A const
D ErrorCode 8000A options(*varsize)

D RCVMO100 DS qualified
D BytesRtn 10I 0
D BytesAvail 10I 0
D MsgSev 10I 0
D MsgID 7A
D MsgType 2A
D MsgKey 4A
D 7A
D CCSID_status 10I 0
D CCSID 10I 0
D MsgDtaLen 10I 0
D MsgDtaAvail 10I 0

```



```

D   MsgDta                8000A

D   ErrorCode             ds                qualified
D   BytesProv             10I 0 inz(0)
D   BytesAvail            10I 0 inz(0)
-----
*   Define constants
-----
D #YES                     C                CONST('Y')
D #NO                      C                CONST('N')
-----
*   START of work fields
-----
D action1                  s                25A   inz('Reply R, to retry lock')
D action2                  s                30A   inz('reply C, cancel (by-pass) lock')
D Function                  S                10
D JobAttr                  S                1A
D NbrSecs                  S                15   5 inz(180)
D Cmd                      S                256   inz('Dspmsg')
D Len                      S                15   5 inz(6)
D Message                  s                256A   varying
D MsgKey                   s                4A
D MsgQ                     s                20A   dim(1) inz('*SYSOPR')
D Reply                    s                100A
-----
*   END of work fields
-----
/free

JobAttr = GetJobAttr() ;
IF JobAttr = 'I' ;
    EXSR @Interactive ;
ELSE ;
    EXSR @BatchMode ;
ENDIF ;
*INLR = *ON ;
RETURN ;

BEGSR @BatchMode ;

//-----
// Create a message to send to the system operator.
// Send an *INQ message to QSYSOPR asking for a reply.
//-----

Reply = 'R' ;
Message = %trim(cpfmessage) + ' ' + %trim(action1) ;
IF ExitAllowed = #YES ;
    Message = %trim(Message) + ', ' + %trim(action2) + '.' ;
ELSE ;
    Message = %trim(Message) + '.' ;
ENDIF ;

QMHSNDM( *blanks ;
         *blanks ;
         Message ;
         %len(Message) ;
         '*INQ' ;
         MsgQ ;
         %elem(MsgQ) ;
         '*PGMQ' ;
         MsgKey ;
         ErrorCode ) ;

//-----
// Wait up to 5 minutes (300 seconds) for a reply to the
// above message. If you change the value of 300 below to
// a value of -1, it will wait indefinitely.
//-----

QMHRVCVPM( RCVM0100 ;
           %size(RCVM0100) ;
           'RCVM0100' ;
           '*' ;
           0 ;
           '*RPY' ;
           MsgKey ;
           300 ;
           '*REMOVE' ;
           ErrorCode ) ;

//-----
// The "Reply" Variable contains the operator's reply
// If the reply was C (cancel) by-pass update
//-----

IF RCVM0100.BytesRtn > 0 AND RCVM0100.MsgDta <> *blank ;
    Reply = %subst(RCVM0100.MsgDta: 1: RCVM0100.MsgDtaLen) ;
ENDIF ;
IF Reply = *blank ;
    Reply = 'R' ;
ENDIF ;

Reply = UpperCase(Reply:%size(Reply)) ;
IF %subst(Reply:1:1) = 'C' ;

```



```
BypassLock = *ON ;
ELSE ;
BypassLock = *OFF ;
ENDIF ;

ENDSR ;

BEGSR @Interactive ;

//-----
// Open the display file to show the lock window
//-----

IF not %open(SWRCLKDI) ;
OPEN SWRCLKDI ;
ENDIF ;
cf = DefineKeys() ;
BypassLock = *OFF ;
z$msgs1 = %subst(CPFMessage:1:40) ;
z$msgs2 = %subst(CPFMessage:41:40) ;
errpgm = CallingPgm ;
errnam = FileLocked ;
// Determine if user allowed to use ESCAPE
*IN50 = ExitAllowed = #YES ;

//-----
// Display record lock message until request to exit
// If F3 was used then by-pass update, if allowed
//-----

WRITE RCLKR001 ;
DOU Function = 'EXIT' ;
MONITOR ;
WRITE RCLKR002 ;
READ SWRCLKDI ;
ON-ERROR *FILE ;
KeyPressed = cf.ENTER ;
ENDMON ;
SELECT ;
WHEN KeyPressed = cf.F3 ;
IF *IN50 ;
BypassLock = *ON ;
ENDIF ;
Function = 'EXIT' ;
WHEN KeyPressed = cf.ENTER ;
Function = 'EXIT' ;
ENDSL ;
ENDDO ;

//-----
// Close lock warning display
//-----

IF %open (SWRCLKDI) ;
CLOSE SWRCLKDI ;
ENDIF ;
ENDSR ;
```



DDS

```

A*****
A* DISPLAY NAME - SWRCLKDI *
A* *
A* FUNCTION - This display shows a pop up screen to *
A* inform the application user of a record lock. *
A* *
A* PROGRAMMER - STEVE CROY 04/12/2006 *
A*****
A* FILE ATTRIBUTE SECTION *
A* *
A* RSTDSP : *YES *
A* MAXDEV : 1 *
A* WAITRCD : 60 *
A* *
A*****
A* MODIFICATION LOG *
A* *
A* DATE PROGRAMMER DESCRIPTION *
A* *
A*****
A* Include PRINT and INVITE at the file level to allow PRINT *
A* function and device file time out coding in applications *
A*****
A*
A*%%EC
A DSPSIZ(24 80 *DS3)
A INVITE
A PRINT
A INDARA
A CF01
A CF02
A CA03
A CF04
A CF05
A CF06
A CF07
A CF08
A CF09
A CF10
A CF11
A CF12
A CF13
A CF14
A CF15
A CF16
A CF17
A CF18
A CF19
A CF20
A CF21
A CF22
A CF23
A CF24
A ROLLUP(90)
A ROLLDOWN(91)
A HOME
A HELP
A R RCLKR001
A OVERLAY
A KEEP
A PUTOVR
A ASSUME
A 24 79' '
A OVRATR
A R RCLKR002
A*%%TS SD 20050622 104000 SCROY REL-V5R2M0 5722-WDS
A CLRL(*NO)
A BLINK
A OVERLAY
A ALARM
A 3 21'.....'
A .....
A DSPATR(HI)
A 4 21': '
A DSPATR(HI)
A FMTNAM 8A O 4 24COLOR(BLU)
A 4 33'Record Access Error Screen'
A DSPATR(HI)
A 4 64': '
A DSPATR(HI)
A 5 21': '
A : '
A DSPATR(HI)
A 6 21': '
A DSPATR(HI)
A 6 23'A record is in use by another job -
A on the'
A 6 64': '
A DSPATR(HI)
A 7 21': '

```



```

A          DSPATR(HI)
A          7 23'system. This denies access to a cr-
A          itical'
A          7 64':
A          DSPATR(HI)
A          8 21':
A          DSPATR(HI)
A          8 23'data file your program needs. -
A          '
A          8 64':
A          DSPATR(HI)
A          9 21': -
A          '
A          DSPATR(HI)
A          10 21':
A          DSPATR(HI)
A          10 23'The job which controls the above f-
A          ile: '
A          DSPATR(UL)
A          10 64':
A          DSPATR(HI)
A          11 21':
A          DSPATR(HI)
A          11 23DSPATR(HI)
A          Z$MSG1 40A O 11 64':
A          DSPATR(HI)
A          12 21':
A          DSPATR(HI)
A          Z$MSG2 40A O 12 23DSPATR(HI)
A          12 64':
A          DSPATR(HI)
A          13 21': -
A          '
A          DSPATR(HI)
A          14 21':
A          DSPATR(HI)
A          14 23'Program requesting data:'
A          ERRRFGM 10A O 14 48DSPATR(HI)
A          14 64':
A          DSPATR(HI)
A          15 21':
A          DSPATR(HI)
A          15 23'File name.....:'
A          ERRNAM 8A O 15 48DSPATR(HI)
A          15 64':
A          DSPATR(HI)
A          16 21': -
A          '
A          DSPATR(HI)
A          17 21':
A          DSPATR(HI)
A          17 23'Press ENTER to attempt to get the -
A          record'
A          17 64':
A          DSPATR(HI)
A          18 21':
A          DSPATR(HI)
A          18 23'again. If the problem persists, ca-
A          ll the'
A          18 64':
A          DSPATR(HI)
A          19 21':
A          DSPATR(HI)
A          19 23'user that has the record you need -
A          to '
A          19 64':
A          DSPATR(HI)
A          20 21':
A          DSPATR(HI)
A          20 23'determine when you can get access.-
A          '
A          20 64':
A          DSPATR(HI)
A          21 21': -
A          '
A          DSPATR(HI)
A          22 21':
A          DSPATR(HI)
A          22 23'F3=Exit and bypass record update. -
A          50 '
A          COLOR(BLU)
A          N50 22 23'Enter=Retry -
A          '
A          COLOR(BLU)
A          22 64':
A          DSPATR(HI)
A          23 21':.....-
A          .....:'
A          DSPATR(HI)

```