```
EEEEEEEEEE  XX      XX    AAAAAA    MM      MM  PPPPPPPP    LL          EEEEEEEEEE    SSSSSSSS
EEEEEEEEEE  XX      XX    AAAAAA    MM      MM  PPPPPPPP    LL          EEEEEEEEEE    SSSSSSSS
EEEEEEEEEE  XX      XX    AAAAAA    MM      MM  PPPPPPPP    LL          EEEEEEEEEE    SSSSSSSS
EE          XX      XX   AA    AA   MMMM  MMMM  PP      PP  LL          EE          SS
EE          XX      XX   AA    AA   MMMM  MMMM  PP      PP  LL          EE          SS
EE          XX      XX   AA    AA   MMMMMMMMMM  PP      PP  LL          EE          SS
EE            XX  XX     AA    AA   MM  MM  MM  PP      PP  LL          EE          SS
EE            XX  XX     AA    AA   MM  MM  MM  PP      PP  LL          EE          SS
EE            XX  XX     AA    AA   MM  MM  MM  PP      PP  LL          EE          SS
EEEEEEEE        XX       AA    AA   MM      MM  PPPPPPPP    LL          EEEEEEEE      SSSSSS
EEEEEEEE        XX       AA    AA   MM      MM  PPPPPPPP    LL          EEEEEEEE      SSSSSS
EEEEEEEE        XX       AA    AA   MM      MM  PPPPPPPP    LL          EEEEEEEE      SSSSSS
EE            XX  XX     AAAAAAAAAA   MM      MM  PP          LL          EE                  SS
EE            XX  XX     AAAAAAAAAA   MM      MM  PP          LL          EE                  SS
EE            XX  XX     AAAAAAAAAA   MM      MM  PP          LL          EE                  SS
EE          XX      XX   AA    AA   MM      MM  PP          LL          EE                  SS
EE          XX      XX   AA    AA   MM      MM  PP          LL          EE                  SS
EE          XX      XX   AA    AA   MM      MM  PP          LL          EE                  SS
EEEEEEEEEE  XX      XX   AA    AA   MM      MM  PP          LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS
EEEEEEEEEE  XX      XX   AA    AA   MM      MM  PP          LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS
EEEEEEEEEE  XX      XX   AA    AA   MM      MM  PP          LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS
```

```
LL              AAAAAA    BBBBBBBB      IIIIII      OOOOOO      CCCCCCCC    OOOOOO    MM        MM
LL              AAAAAA    BBBBBBBB      IIIIII      OOOOOO      CCCCCCCC    OOOOOO    MM        MM
LL            AA      AA  BB      BB      II      OO      OO  CC          OO      OO  MMMM    MMMM
LL            AA      AA  BB      BB      II      OO      OO  CC          OO      OO  MMMM    MMMM
LL            AA      AA  BB      BB      II      OO      OO  CC          OO      OO  MM  MM  MM
LL            AA      AA  BB      BB      II      OO      OO  CC          OO      OO  MM  MM  MM
LL            AA      AA  BBBBBBBB        II      OO      OO  CC          OO      OO  MM      MM
LL            AA      AA  BBBBBBBB        II      OO      OO  CC          OO      OO  MM      MM
LL            AAAAAAAAAA  BB      BB      II      OO      OO  CC          OO      OO  MM      MM
LL            AAAAAAAAAA  BB      BB      II      OO      OO  CC          OO      OO  MM      MM
LL            AA      AA  BB      BB      II      OO      OO  CC          OO      OO  MM      MM
LL            AA      AA  BB      BB      II      OO      OO  CC          OO      OO  MM      MM
LLLLLLLLLL    AA      AA  BBBBBBBB      IIIIII      OOOOOO      CCCCCCCC    OOOOOO    MM      MM
LLLLLLLLLL    AA      AA  BBBBBBBB      IIIIII      OOOOOO      CCCCCCCC    OOOOOO    MM      MM

FFFFFFFFFF      OOOOOO    RRRRRRRR
FFFFFFFFFF      OOOOOO    RRRRRRRR
FF            OO      OO  RR      RR
FF            OO      OO  RR      RR
FF            OO      OO  RR      RR
FF            OO      OO  RR      RR
FFFFFFFF      OO      OO  RRRRRRRR
FFFFFFFF      OO      OO  RRRRRRRR
FF            OO      OO  RR  RR
FF            OO      OO  RR  RR
FF            OO      OO  RR    RR
FF            OO      OO  RR    RR
FF              OOOOOO    RR      RR
FF              OOOOOO    RR      RR
```

```
File:  LABIOCOM.FOR
       Version 'V04-000'
```



```
      Logical Function LABIO_INIT( PRIVILEGE )

! This routine is used to attach a LABIO user program to the
! LABIO system. It associated the two event flag clusters and
! maps to the LABIO global data section.

! INPUT:        PR VILEGE - Privileged LABIO users can set this
!               to 1 to allow write access to the data base.
!               All others must set this to 0.

! OUTPUT:       None - Currently will always return with success code.
!               If an error occurs, FATALERR is called to display
!               the error messages and STOP THE PROCESS!


      Include 'LABCHNDEF.FOR'
      Logical*4 SYS$ASCEFC,SYS$MGBLSC,SUCCESS,SYS$WAITFR
      External SEC$M_WRT

!
! Create event flag cluster EF_NOTIFY and associate with event flags 64-95
! These are used to notify the Data Acquisition process.

      SUCCESS = SYS$ASCEFC( %VAL(EF_NOTIFY_1),EF_NOTIFY_CLSTR,,)
      If ( .not. SUCCESS)
     1      Call FATAL_ERROR( SUCCESS, 'CREATING EVENT FLAG CLUSTER')
!
! Create event flag cluster EF_STATUS and associate with event flags 96-127
! These are used to notify and report the status of the user buffers
```

```
!
        SUCCESS = SYS$ASCEFC( %VAL(EF_STATUS_1),EF_STATUS_CLSTR,,)
        If ( .not. SUCCESS)
        1       Call FATAL_ERROR( SUCCESS, 'CREATING EVENT FLAG CLUSTER')

!
! Map to the GLOBAL DATA section created by the I/O program
! Wait for event flag EF_CONNECT (non-privileged) or
! EF_DATA_ACQ  (privileged) before attempting mappimg.

        SECTION(1) = %Loc(LABIO_BUFFER_S)
        SECTION(2) = %Loc(LABIO_BUFFER_E) - 1

        SECTION_FLAGS = 0               !Default flags

        If( PRIVILEGE .ne. 0 ) Then
          SECTION_FLAGS=%Loc(SEC$M_WRT)
          Call SYS$WAITFR( %Val( EF_DATA_ACQ ) )
        Else
          Call SYS$WAITFR( %Val( EF_CONNECT ) )
        End If

        SUCCESS = SYS$MGBLSC( SECTION,,,%Val(SECTION_FLAGS),'LABIOCOMMON',,)
        If( .not. SUCCESS ) Call FATAL_ERROR(SUCCESS,'mapping data section')

        LABIO_INIT = SUCCESS

        Return

        End
```

```
!       FATAL_ERROR  -  FATAL ERROR HANDLER
!
!       This routine is used to report a fatal error and exit the image
!
!       INPUT:  ERROR_CODE - SYSTEM ERROR CODE TO REPORT
!               ERROR_MESSAGE - ERROR MESSAGE TO BE PRINTED
!
!       OUTPUT: NONE
!
!       >>>> THIS ROUTINE DOES NOT RETURN <<<<<
!
!       FUNCTION: TYPEs the message
!
!                 'process name-FATAL ERROR - error_message'
!
!                Then prints system message corresponding to  ERROR_CODE
!
!
!                Finally, exits image by calling LIB$STOP
!
        Subroutine FATAL_ERROR(error_code,error_message)

        Integer*4 ERROR_CODE
        Character ERROR_MESSAGE*(*)
        Logical*4 SUCCESS,SYS$CREMBX,SYS$GETJPI
        Integer*2 JPI2(8),PROCESS_NAME_L
        Integer*4 JPI4(4)
        Character*15 PROCESS_NAME
        Equivalence (JPI2,JPI4)
        Parameter JPI$_PRCNAM='31C'X

!       Get the process name

        JPI2(1) =  15                       !Number of elements in name
        JPI2(2) = JPI$_PRCNAM               !Want process name
        JPI4(2) = %Loc(PROCESS_NAME)        !Address of process name
        JPI4(3) = %Loc(PROCESS_NAME_L)      !Address of process name length
        JPI4(4) = 0                         !Terminate list

        Call SYS$GETJPI(,,,JPI4,,,)

!  Print the process name and error message
!
        Type 100, PROCESS_NAME(1:PROCESS_NAME_L),ERROR_MESSAGE

!  Print the error message corresponding to ERROR_CODE and exit
!
        Call LIB$STOP( %Val(ERROR_CODE) )

100     Format(' 'A,' - FATAL ERROR ',A)

        Stop

        END
![End of File]
```

XALINK
MAR

DRMASTER
FOR

LABIOLINK
COM

LPATEST
FOR

LABIOPEAK
FOR

LABIOSTRT
COM

XAMESSAGE
MAR

XATEST
FOR

LABIOCOM
FOR

LBRDEMO
COM

LABMBXDEF
FOR

LABIOSAMP
FOR

MAILCOMPRESS
COM

LABCHNDEF
FOR

CONNECT
COM

LABIOCON
FOR

LBRDEMO
FOR

PEAK
FOR

DRCOPYBLD
COM

XIDRIVER
MAR

LABIOSEC
FOR

DRSLAVE
FOR

LABIOACQ
FOR

LABIOCOMP
COM

LABIOSTAT
FOR

TESTLABIO
FOR