



LUND UNIVERSITY

Implementation Procedures for INTRAC

Schöntal, Tomas

1980

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Schöntal, T. (1980). *Implementation Procedures for INTRAC*. (Technical Reports TFRT-7187). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

IMPLEMENTATION PROCEDURES FOR INTRAC

TOMAS SCHÖNTAL

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
FEBRUARY 1980

Dokumentutgivare

Lund Institute of Technology
Handläggare Dept of Automatic Control

Författare

Tomas Schönthal

Dokumentnamn

REPORT LUTFD2/(TFRT-7187)/1-26/(1980)

Utgivningsdatum

Feb 1980

Dokumentbeteckning

06T6

Ärendebeteckning

06T6

10T4

Dokumenttitel och undertitel

18T0
Implementation Procedures for INTRAC

Referat (sammandrag)

26T0
The purpose of this document is to describe how to install the program package INTRAC in an arbitrary computer system. It will neither discuss how to use INTRAC, nor how its individual component subroutines function.

Referat skrivet av

42T0
Author

Förslag till ytterligare nyckelord

44T0
Computer aided design, Interactive programs

Klassifikationssystem och -klass(er)

50T0

Indextermer (ange källa)

52T0
Computer software (Theaurus of Engineering and Scientific Terms, Engineers Joint Council, N.Y., USA).

Omfång

26T0 pages

Övriga bibliografiska uppgifter

56T2

Språk

English

Sekretessuppgifter

60T0

ISSN

60T4

ISBN

60T6

Dokumentet kan erhållas från

62T0
Department of Automatic Control
Lund Institute of Technology
Box 725, S-220 07 LUND, Sweden

Mottagarens uppgifter

62T4

Pris

66T0

DOKUMENTATABLAD enligt SIS 62 10 12

SIS-DB 1

Implementation Procedures for INTRAC

The purpose of this document is to describe how to install the program package INTRAC in an arbitrary computer system. It will neither discuss how to use INTRAC, nor how its individual component subroutines function.

Contents

- 1) Starting point
- 2) Multiple Entry Points
- 3) Building a Binary Library File
- 4) Segmenting INTRAC
- 5) Test Programs for SUBR. INTRAC and RIFF
- 6) Subroutine Summaries
- 7) Implementation Order
- 8) Subroutine Calls, Top-Down
- 9) Subroutine Calls, Bottom-Up
- 10) COMMON Block References
- 11) Referenced COMMON Blocks

1) Starting Point

=====

The reader is assumed to have implemented INTRAC's system dependent functions, i.e. file and text handling. See the documents:

"File Handling in Program Packages", by Tommy Essebo and
"Character and String Handling in INTRAC", by Tommy Essebo.

2) Multiple Entry Points

=====

Provided that your FORTRAN-compiler accepts subroutines with multiple entry-points, INTRAC can be compiled at once. If not, the following routines have to be split up:

Collective name	Entry points
-----	-----
DISHDL	EJECT IWRITE TRHDL WRHDL
LOGARG	LDELIM LFORML LHNAME LINT LNUMB LREAL LTERM

```
C1B  SUBROUTINE MULENT(A,B)
C    SAMPLE MULTIPLE ENTRY POINT ROUTINE
C1E
C2B  LOGICAL ....
      ***
      DIMENSION ...
      ***
      COMMON / ... /
      ***
      DATA ...
      ***
C2E
C3B  A=B
      RETURN
C3E  ENTRY E1(A1,B1)
      ***
      IF(J1.NE.0) GOTO 500
      ***
      RETURN
C4B  ENTRY E2(A2,B2)
      ***
      IF(J2.NE.0) GOTO 510
      ***
      RETURN
C4E
      ***
      500 IERR=1
          RETURN
C5B  510 IERR=2
          RETURN
C5E  END
```

To extract the entry E1, the code sections C1B-C1E, C3B-C3E, C4B-C4E and C5B-C5E plus the data in C2B-C2E that are not referenced by E1 should be deleted. The statement 'ENTRY E1(A1,B1)' should be replaced by 'SUBROUTINE E1(A1,B1)', placed above C2B.

Note: The above multiple entry point routine is not claimed to be general, it merely exemplifies the worst case situation to be found in INTRAC. E.g. different entry points do not reference each other's formal arguments in INTRAC.

3) Building a Binary Library File

=====

For a system with a linker, that sequentially scans a library file, that (part of the) library file, which contains INTRAC's routines, should be sorted in the following order:

DIDENT	DINT	DREAL	FIDENT	FINT	FNUMB	FREAL
LFINAM	LTLONG	LHOLLS	LHOLL	LOGARG	LARG	
INTTAB	INTINI	INTERR				
INTRAC	INTR	RESEX				
FORMAC	FORNXT	FREE	IFGOLA	LET	MACEND	READX
RESUME	SWITCH	WRITEX				
FR2	READIN	WRT	WRT2			
MACHDL	COMLIN	RECLIN	SUBST	GET	PUT	CRENAM
LFIND						
LPCOM	LOGBUF	TDATE	ECHBUF	ERRPOS	WRLINE	
PFLOAT	PINT	PHOLL	PTERM			
RIFF	RABC	RFP	DIGITS	SIGN1		
DISHDL	...					
File Handling	...					
PSPACE	LENGTH					
Character and String Handling	...					
IMACON	RMACON					

4) Segmenting INTRAC

=====

To save memory, INTRAC may be segmented according to the diagram below. The structure is by no means optimal, but works reasonably well. The vertical dotted lines represent boarder lines between segments that may overlay each other, e.g. the segment (FORNXT+FREE+FR2) may execute in the same memory space as the segment (FORMAC+READ+READIN).

The source code does in no way reflect the overlay structure, so it is assumed that your system has a linker that accepts a description of the overlay structure as part of the command input.

Should the transfer of control between segments not be administered automatically by your computer system's overlay handler, but rather, which often is the case, e.g. require explicit calls to a system supplied subroutine, such as CALL LINK('SEGNAM'), then you must adjust the source code of INTRAC accordingly, without any assistance from our side.

Note: In the diagram below a number of COMMON-blocks are allocated to the resident area. This is a necessary condition for INTRAC to function properly, no matter for which application it is used.

Resident Supervisory MAIN PROGRAM
area SUBR. INTRAC, DISHDL, RIFF, FCLOSE, FENTER, FSEEK
 BLOCK DATA INTTAB
 COMMON /GLOBAL/

Level 1 . INTINI . FORMAC . FORNXT . IFGOLA . WRITEX .
 . INTERR . READX . FREE . LET . WRT .
 . READIN . FR2 . MACEND . WRT2 .
 . SWITCH . RESUME .

Level 2 . INTR .
 . COMLIN .
 . SUBST .
 . RECLIN .

Level 3 . RESEX .
 . MACHDL .
 . LPCOM .
 . WRLINE .

5) Test Programs for SUBR. INTRAC and RIFF

=====

PROGR. INMAIN can be used to exercise INTRAC.
It provides the dummy application command 'TEST'.
By typing TEST followed by a number of arguments, these will
be verified by a printout.

Below is a sample dialouge from INTRAC's test program.
Input is marked by '>'.
<=

** INTRAC TEST PROGRAM **

```
>TEST MONKEY 1234 123456.78E+12 * 'VERY LONG STRING' (
** TEST COMMAND **
NRL, NRR = 7, 0
ITYP ARG
 1 TEST
 1 MONKEY
 2      1234
 3      1.23457E+17
 4 *
 9      35
 4 (
```

PROGR. RIMAIN can be used to test RIFF.

In answer to the prompting message TYPE A LINE, a free format line is typed, and RIFF will extract its individual arguments. Below is a sample dialouge from RIFF's test program. Input is marked by '>'. Note that double quote, " in addition to carriage return acts as line terminator, i.e. no arguments to the right of it will be decoded.

TYPE A LINE

>TEST MONKEY QWERTYUIOP 123 3.1415 .56E-09 * ** : " (N

TEST MONKEY QWERTYUIOP 123 3.1415 .56E-09 * ** : " (N

54 CHARACTERS

IPOINT	IND	RRES	IRES
5	1	TEST	
12	1	MONKEY	
23	2	QWERTYUI	
27	3	123.00	123
34	4	3.1415	3
42	4	5.60000E-10	0
44	6	*	
46	6	*	
47	6	*	
49	6	:	
50	7		

6) Subroutine Summaries

=====

System dependent routines are marked by an asterisc.

Main Level

INTERR
prints INTRAC's error messages

INTINI
initializes INTRAC

INMAIN
INTRAC's test program

INTR
contains SUBR. INTRAC's heavy logic

INTRAC
main routine

INTTAB*
BLOCK data program defining all INTRAC's resident data

External Access to INTRAC's Data Base

DIDENT
 assigns an identifier (text string) to a global variable

DINT
 assigns an integer value to a global variable

DREAL
 assigns a real value to a global variable

FIDENT
 fetches an identifier (text string) from a global variable

FINT
 fetches the value of an integer global variable

FNUMB
 fetches a number from a real or integer global variable

FREAL
 fetches the value from a real global variable

LARG
 general argument decoding function

LHOLL
 compares a command argument with a given text string

LHOLLS
 compares a command argument with a list of text strings

LFINAM
 decodes a file name

LOGARG
 collection of argument decoding functions

LTLONG
 checks the length of a file name

Action Routines for INTRAC's Reserved Commands

RESEX
selects appropriate routine to execute
INTRAC's reserved commands

FORMAC
performs 'FORMAL' and 'MACRO'

FORNXT
performs 'FOR' and 'NEXT'

FREE, FR2
performs 'FREE'

IFGOLA
performs 'IF', 'GOTO' and 'LABEL'

LET
performs 'LET'

MACEND
performs 'END'

READX
performs 'READ'

RESUME
performs 'RESUME'

SWITCH
performs 'SWITCH'

WRITEX, WRT, WRT2
performs 'WRITE'

READIN
help routine to FORMAC and READ

Substitution Routines

GET

fetches the value of a global variable

PUT

assigns a value to a global variable

RECLIN

reconstructs the command line by replacing
symbolic arguments by their values

SUBST

replaces symbolic arguments in a command line
by their values

String Decoding Routines

COMLIN

command line decoder

LENGTH

computes the length of a text string

RIFF, RABC, RFP, DIGITS, SIGN1

decodes a free format argument in a line buffer

String Encoding Routines

CRENAM

concatenates a text string and an integer

PFLOAT

encodes a floating point number into a line buffer

PHOLL

packs a Hollerith string into a buffer

PINT

encodes an integer number into a line buffer

PSPACE

packs spaces into a buffer

PTERM

packs a line terminator into a line buffer

Input/Output

ECHBUF

echoes the currently executing command in a MACRO

ERRPOS*

positions terminal output

LOGBUF

writes a command line buffer with an optional
date/time message

LPCOM

logs the command line on the line printer

MACHDL

MACRO handler

WRLINE

writes a prompting sign, then reads a line buffer
on the same line, also used to write a line buffer

Miscellaneous

CLRSW*

resets an external break condition

IMACON*

returns integer system dependent constants
concerning data representations

ISENSW*

senses an external break condition

RMACON*

returns real system dependent constants
concerning data representations

TDATE*

returns the system date/time

7) Implementation order

=====

Start with INTTAB, IMACON and RMACON. Although their structures are system independent, they contain parameters, whose values may vary from one installation to another:

INTTAB: COMMONS /DEVICE/ and /DEVMAC/
contain logical I/O numbers.

IMACON, RMACON: parameters regarding data representations.

CLRSW, ISENSW and TDATE may be replaced by do-nothing routines, if the facilities associated with them are not desired.

ISENSW: do-nothing value = .FALSE.

TDATE : see the description of COMMON /DAT001/ in INTTAB.

Please refer to the source codes, then continue with:

a) Character and String Handling -> SIGN1 .. RIFF ->
RIFF's test program

b) File Handling and DISHDL ...

c) PSPACE, LENGTH -> PTERM .. PFLOAT -> SUBST, RECLIN ->
COMLIN -> MACHDL

d) reserved commands: WRT, WRT2 .. -> WRITEX .. ->
RESEX -> INTR -> INTRAC -> INTRAC's test program

8) Subroutine Calls, Top-Down

=====

E.g. SUBR. LPCOM calls upon SUBR. LOGBUF.

```

CLRSW :
COMLIN: FAC      HSTORV IFAC      LCOMPV RECLIN RFLOAT RIFF
CRENAM: HSTORV  LENGTH PINT
DIDENT: PUT
DIGITS: FAC
DINT   : PUT      RFLOAT
DISHDL: PLCURS  TPOS   TREAD   TWRITE
DREAL  : PUT
ECHBUF: ERRPOS  WRLINE
ERRPOS:
FAC    : GAC      HSTORV
FCLOSE: DELETE  FILCHK LUFIND  RCLOSE  WCLOSE
FENTER: ENTER   FILCHK HSTORV  LUFIND
FIDENT: GET     HSTORV
FILCHK: LCOMPV
FILTAB:
FINT   : GET     IFIXR
FNUMB  : GET     IFIXR
FORMAC: FENTER  HSTORV LCOMPV  LFIND   READIN  RFLOAT
FORNXT: COMLIN  HSTORV IFIXR   LCOMPV  MACHDL  RBUF   RFLOAT  SUBST
FREAL  : GET
FREE   : FR2     HSTORV LCOMPV
FR2    : HSTORV  LCOMPV
FSEEK  : FILCHK HSTORV  LUFIND  SEEK
GAC    :
GET    : HSTORV  IFIXR   LCOMPV
HSTORV:
IFAC   : HSTORV
IFGOLA: COMLIN  HSTORV  IFIXR   LCOMPV  LFIND   MACHDL  RBUF   SUBST
IFIXR  :
IMACON:
INMAIN: HSTORV  IFIXR   INTERR  INTINI  INTRAC  LPCOM
INTERR: ECHBUF  IWRITE
INTINI:
INTR   : COMLIN  HSTORV  LCOMPV  LFIND   LOGBUF  RECLIN  SUBST  WBUF
        WRLINE
INTRAC: CLRSW   ECHBUF  INTR    ISENSW  LPCOM   MACHDL  RESEX
INTTAB:
ISENSW:
LARG   : HSTORV  IFIXR
LCOMPV:
LENGTH: GAC     LCOMPV
LET    : CRENAM  HSTORV  IFIXR   LCOMPV  LFIND   MACHDL  RFLOAT  SUBST
LFINAM: HSTORV  LHNAME  LTLONG

```

```

LFIND : LCOMPV
LHOLL : LCOMPV LDELIM LHNAME
LHOLLS: LHOLL
LOGARG: LARG
LOGBUF: HSTORV PAC PINT WRLINE
LPCOM : LOGBUF
LTLONG: IMACON LENGTH
LUFIND:
MACEND: FCLOSE FENTER FSEEK HSTORV IFAC LCOMPV MACHDL RBUF
      RIFF WBUF
MACHDL: FCLOSE FSEEK HSTORV IFAC LCOMPV LFIND PUT RBUF
      RFLOAT RIFF
PAC :
PFLOAT: GAC IMACON LCOMPV LOG10 PAC PINT
PHOLL : GAC LCOMPV PAC
PINT : PAC
PSPACE: PAC
PTERM : PAC
PUT : HSTORV IFIXR LCOMPV RFLOAT
RABC : FAC PAC
RBUF : HSTORV LENGTH RDREAL
READIN: COMLIN HSTORV LCOMPV WRLINE
READX : HSTORV LCOMPV LFIND MACHDL PUT READIN RFLOAT
RECLIN: GAC HSTORV IFIXR LCOMPV PAC PFLOAT PHOLL PINT
      PSPACE PTERM
RESEX : FORMAC FORNXT FREE HSTORV IFGOLA LCOMPV LET LPCOM
      MACEND READX RESUME SWITCH WRITEX WRLINE
RESUME: LCOMPV MACHDL
RFLOAT:
RFP : DIGITS FAC GAC IMACON LCOMPV LOG10 RMACON SIGN1
RIFF : FAC HSTORV RABC RFP
RMACON:
SIGN1 : FAC LCOMPV
SUBST : GET HSTORV IFIXR LCOMPV
SWITCH: LFIND SUBST
WBUF : IWRITE LUFIND PSPACE WRHDL WRREAL
WRITEX: EJECT LCOMPV LPCOM SUBST WBUF WRT WRT2
WRLINE: GAC HSTORV PAC PSPACE PTERM RBUF TRHDL WBUF
      WRHDL
WRT : GAC HSTORV IFIXR LCOMPV PAC PFLOAT PHOLL PINT
      PSPACE PTERM SUBST
WRT2 : EJECT IFIXR IMACON PAC PFLOAT PHOLL PINT PSPACE
      PTERM WBUF

```


9) Subroutine Calls, Bottom-Up

=====

E.g. the following routines call upon SUBR. WRLINE:
 ECHBUF, INTR, LOGBUF, READIN and RESEX.

```

CLRSW : INTRAC
COMLIN: FORNXT IFGOLA INTR  READIN
CRENAM: LET
DELETE: FCLOSE
DIGITS: RFP
ECHBUF: INTERR INTRAC
EJECT : WRITEX WRT2
ENTER  : FENTER
ERRPOS: ECHBUF
FAC    : COMLIN DIGITS RABC  RFP    RIFF  SIGN1
FCLOSE: MACEND MACHDL
FENTER: FORMAC MACEND
FILCHK: FCLOSE FENTER FSEEK
FORMAC: RESEX
FORNXT: RESEX
FREE   : RESEX
FR2    : FREE
FSEEK  : MACEND MACHDL
GAC    : FAC    LENGTH PFLOAT PHOLL  RECLIN RFP    WRLINE WRT
GET    : FIDENT FINT   FNUMB  FREAL  SUBST
HSTORV: COMLIN CRENAM FAC    FENTER FIDENT FORMAC FORNXT FREE
        FR2    FSEEK  GET    IFAC   IFGOLA INMAIN INTR  LARG
        LET   LFINAM LOGBUF MACEND MACHDL PUT   RBUFF READIN
        READX RECLIN RESEX  RIFF   SUBST  WRLINE WRT
IFAC   : COMLIN MACEND MACHDL
IFGOLA: RESEX
IFIXR  : FINT   FNUMB  FORNXT GET    IFGOLA INMAIN LARG  LET
        PUT   RECLIN SUBST  WRT    WRT2
IMACON: LTLONG PFLOAT RFP    WRT2
INTERR: INMAIN
INTINI: INMAIN
INTR   : INTRAC
INTRAC: INMAIN
ISENSW: INTRAC
IWRITE: INTERR WBUFF
LARG   : LOGARG
LCOMPV: COMLIN FILCHK FORMAC FORNXT FREE  FR2    GET    IFGOLA
        INTR  LENGTH LET    LFINAM LHOLL  MACEND MACHDL PFLOAT
        PHOLL PUT   READIN READX  RECLIN RESEX  RESUME RFP
        SIGN1 SUBST WRITEX WRT
LDELIM: LHOLL
LENGTH: CRENAM LTLONG RBUFF
LET    : RESEX

```

```

LFIND : FORMAC IFGOLA INTR LET MACHDL READX SWITCH
LHNAME: LFINAM LHOLL
LHOLL : LHOLLS
LOGBUF: INTR LPCOM
LOG10 : PFLOAT RFP
LPCOM : INMAIN INTRAC RESEX WRITEX
LTLONG: LFINAM
LUFIND: FCLOSE FENTER FSEEK WBUFF
MACEND: RESEX
MACHDL: FORNXT IFGOLA INTRAC LET MACEND READX RESUME
PAC : LOGBUF PFLOAT PHOLL PINT PSPACE PTERM RABC RECLIN
      WRLINE WRT WRT2
PFLOAT: RECLIN WRT WRT2
PHOLL : RECLIN WRT WRT2
PINT : CRENAM LOGBUF PFLOAT RECLIN WRT WRT2
PLCURS: DISHDL
PSPACE: RECLIN WBUFF WRLINE WRT WRT2
PTERM : RECLIN WRLINE WRT WRT2
PUT : DIDENT DINT DREAL MACHDL READX
RABC : RIFF
RBUF : FORNXT IFGOLA MACEND MACHDL WRLINE
RCLOSE: FCLOSE
RDREAL: RBUF
READIN: FORMAC READX
READX : RESEX
RECLIN: COMLIN INTR
RESEX : INTRAC
RESUME: RESEX
RFLOAT: COMLIN DINT FORMAC FORNXT LET MACHDL PUT READX
RFP : RIFF
RIFF : COMLIN MACEND MACHDL
RMACON: RFP
SEEK : FSEEK
SIGN1 : RFP
SUBST : FORNXT IFGOLA INTR LET SWITCH WRITEX WRT
SWITCH: RESEX
TPOS : DISHDL
TREAD : DISHDL
TRHDL : WRLINE
TWRITE: DISHDL
WBUFF : INTR MACEND WRITEX WRLINE WRT2
WCLOSE: FCLOSE
WRHDL : WBUFF WRLINE
WRITEX: RESEX
WRLINE: ECHBUF INTR LOGBUF READIN RESEX
WRREAL: WBUFF
WRT : WRITEX
WRT2 : WRITEX

```

10) COMMON Block References

=====

E.g. SUBR. WRLINE references COMMON /DEVICE/

```
DIGITS: CRANK  RFP003
DISHDL: DEVICE DISCOM
ECHBUF: COMINF DEVICE LINBUF MACINF
FAC    : CRANK  IFACOM
FCLOSE: FCTAB
FENTER: DEVICE FCTAB
FILCHK: FCTAB
FILTAB: FCTAB
FORMAC: COMINF LINBUF MACINF
FORNXT: COMINF LINBUF MACINF
FREAL  :
FREE   : COMINF GLOBAL LINBUF MACINF
FRZ    : GLOBAL
FSEEK  : DEVICE FCTAB
GAC    : CRANK
GET    : GLOBAL
IFAC   : IFACOM
IFGOLA: COMINF LINBUF MACINF SUS001
INMAIN: COMINF DEVICE GLOBAL LINBUF
INTERR: COMINF DEVICE
INTINI: COMINF MACINF SWI001
INTR   : COMINF DEVICE LINBUF MACINF SWI001
INTRAC: COMINF MACINF
INTTAB: COMINF DAT001 DEVICE DEVMAC DISCOM LINBUF MACINF SUS001
        SWI001
LARG   : COMINF LINBUF
LET    : COMINF LINBUF MACINF
LFINAM: COMINF
LOGBUF: DAT001 DAT002 LINBUF
LPCOM  : COMINF DEVICE MACINF SWI001
LUFIND: FCTAB
MACEND: COMINF LINBUF MACINF
MACHDL: COMINF LINBUF MACINF SUS001
PUT    : GLOBAL
READIN: DEVICE LINBUF
READX  : COMINF LINBUF MACINF
RECLIN: COMINF LINBUF
RESEX  : COMINF DEVICE LINBUF MACINF SWI001
RESUME: COMINF LINBUF MACINF SUS001
RFP    : CRANK  RFP003
SUBST  : COMINF LINBUF MACINF
SWITCH: COMINF DAT001 LINBUF MACINF SWI001
WBUF   : DEVICE FCTAB
WRITEX: COMINF DEVICE LINBUF MACINF
```

WRLINE: DEVICE
WRT : COMINF LINBUF MACINF
WRT2 : COMINF GLOBAL MACINF

11) Referenced COMMON Blocks

=====

E.g. COMMON /DISCOM/ is referenced by SUBR. DISHDL
and BLOCK DATA INTTAB.

```

COMINF: ECHBUF  FORMAC  FORNXT  FREE    IFGOLA  INMAIN  INTERR  INTINI
        INTR    INTRAC  INTTAB  LARG    LET     LFINAM  LPCOM   MACEND
        MACHDL  READX   RECLIN  RESEX   RESUME  SUBST   SWITCH  WRITEX
        WRT     WRT2

CRANK  : DIGITS  FAC     GAC     RFP

DAT001: INTTAB  LOGBUF  SWITCH

DAT002: LOGBUF

DEVICE: DISHDL  ECHBUF  FENTER  FSEEK   INMAIN  INTERR  INTR    INTTAB
        LPCOM   READIN  RESEX   WBUFF   WRITEX  WRLINE

DEVMAC: INTTAB

DISCOM: DISHDL  INTTAB

FCTAB  : FCLOSE  FENTER  FILCHK  FILTAB  FSEEK   LUFIND  WBUFF
GLOBAL: FREE    FR2     GET     INMAIN  PUT     WRT2

IFACOM: FAC     IFAC

LINBUF: ECHBUF  FORMAC  FORNXT  FREE    IFGOLA  INMAIN  INTR    INTTAB
        LARG    LET     LOGBUF  MACEND  MACHDL  READIN  READX   RECLIN
        RESEX   RESUME  SUBST   SWITCH  WRITEX  WRT

MACINF: ECHBUF  FORMAC  FORNXT  FREE    IFGOLA  INTINI  INTR    INTRAC
        INTTAB  LET     LPCOM   MACEND  MACHDL  READX   RESEX   RESUME
        SUBST   SWITCH  WRITEX  WRT     WRT2

RFPO03: DIGITS  RFP

SUS001: IFGOLA  INTTAB  MACHDL  RESUME

SWIO01: INTINI  INTR    INTTAB  LPCOM   RESEX   SWITCH

```