85-62043
Issue 6

OS4000 Facts Book
Volume 1:

TERMINAL USER'S FACTS BOOK

GPT COMPUTERS

The information presented herein gives only general indications of product capacity, performance and suitability, none of which shall form part of any contract. Reference should be made to GPT COMPUTERS LIMITED for information not herein defined. All products, materials and services are sold subject to GPT COMPUTERS LIMITED Conditions of Contract, copies of which are available on request.

Continuous development of GPT COMPUTERS LIMITED products may result in changes to the data herein and the Company reserves the right to add, delete or alter products without prior notice. You should ensure that the information contained herein has not been superseded.

GPT COMPUTERS LIMITED
Computer Products Division
Elstree Way
Borehamwood
Hertfordshire WD6 1RX
England

tel: 01-953 2030
fax: 01-207 1277
telex: 22777

A GEC PLESSEY TELECOMMUNICATIONS COMPANY

TERMINAL USER'S FACTS BOOK

CONTENTS

INTRODUCTION

This Facts Book is a summary of information required by a terminal user. It contains commands for use with job control, batch jobs, text editing and program development aids. Also included are error codes with brief explanations, and character codes.

Facts Book 1 is the first volume in a set covering OS4000 software. The other three available are the Processor's, the Programmer's and the System Manager and Operator Facts Books (85-64731,85-64732 and 85-62050 respectively).

This series of Facts Books is intended to supplement, and not replace, the manual set.

TYPOGRAPHIC CONVENTIONS

In the presentation of command syntax in this manual the following typographic conventions are used:

- If an item is in bold type (e.g. **DISPLAY**) it is to be entered exactly in the form given. Bold type is used chiefly for keywords (e.g. command and argument names).

- If an item is in italics (e.g. *file*) it is to be substituted by the name of an appropriate item of that class.

- If an item is enclosed in square brackets, its use is optional. The permitted number of selections from bracketted items is shown by numbers following the bracket. A subscript number gives the minimum and a superscript number gives the maximum permitted. The letter R in the superscript position indicates that the maximum is unlimited. If the numbers are omitted, the default values of 0 for minimum and 1 for maximum are implied.

For example:

$$\textbf{FREE}\qquad \left[\begin{array}{l}[\textit{virtual device name}]\\ \textbf{ALL}\end{array}\right]^{R}_{1}$$

The command FREE must be followed by either an unlimited number of virtual device names or the argument ALL.

In describing output from the computer, such as prompt and error messages, a similar distinction between fixed elements (in bold type) and variable elements (in italic type) is employed.

JOB CONTROL LANGUAGE

JCL input must not contain lower case alphabetic characters unless they occur within a string or the variable INLOW is set to perform conversions.

JCL COMMANDS

(System and Library Commands - see later section)

Conditional and Transfer Commands

IF *expression* **THEN** *command*
TEST *expression* **THEN** *command 1*
ELSE *command 2*

GOTO *label*
RETURN *expression*
STOP *expression*

These may occur only in macro commands or batch jobs. *expression* may contain integers (decimal or hex, the latter preceded by @) and defined integer-valued variables which may be combined by use of any of the following operators, in decreasing order of precedence:

NOT (invert) (unary)
! (bitwise or)
| (bitwise and)
***** (multiply) **/** (divide)
+ (plus) **-** (minus) (unary or binary)
GT GE EQ LE LT NE (relational)
AND (logical and)
OR (logical or)

Comments

//comment [**//**
NL]

For example: **EDIT //Using the current file**

Getting Input from the Terminal

[**/prompt*] *prompt* is optional

For example: **SET INPUTNAME = [*/Enter inputname]**

The prompt **Enter inputname** is output on the terminal. INPUTNAME is then set to the name given.

JCL SYSTEM CONSTANTS

ACCT	*acctid*
BATCH	TRUE if batch job
FALSE	0
MODE	String value giving environment mode user is working under (values are defined by the System Manager)
MODULE	String value identifying the jobid which the user is working under.
ONLINE	TRUE if online job
SYSTITLE	Title of system as defined at System Generation
TRANSACT	TRUE if transaction processing job
TRUE	-1
USER	*userid*
VERSION	Command process version number (in hexadecimal)

JCL SYSTEM VARIABLES

* = set by **SET/UNSET** commands.

Variable	**When Set**	**Default**	**Use**
ABEND	After a simple command	-	Set if the last user or library command ended abnormally
AIDAPROG	Each time that a library or user command is loaded into the AIDA shell	Null String	As USERPROG. Only available if a AIDA shell is present.
BADFILE	After any command	-	Set if filing error in last system command or if last user or library command returned a Type 3 error.
BVERIFY	*	Value of BATCH	If TRUE commands at outer-most level are listed
CONDCODE	After a simple or macro command	-	Set to condition code
DATETIME	When requested	-	String containing the current date and time. SET/UNSET not available
DIRECTMODE	Before and after macro commands	-	TRUE when obeying terminal commands. SET/UNSET not available

Variable	When Set	Default	Use
DUMP	*	FALSE	If TRUE, where AIDA is available, a formatted dump is produced if a user command ended abnormally
ERROR	After any command	-	Set if any, or all, of ABEND, BADFILE or CONDCODE are set.
ERRTYPE	After any command	-	Type of error: 1 = DM, 2 = JCL, 3 = FILING. Greater than 16 = any combination
ERRVAL	After any command	-	Value of error
EXPAND	*	FALSE	If TRUE arguments are listed before each command
INLOW	*	FALSE	If TRUE, command input has lower case converted to upper case
JOBCOMP	At end of each IEU	-	Computation time remaining for job (units of 1 000 000 instructions) SET/UNSET not available
JOBEXEC	Every 5 minutes	-	Elapsed time remaining for job (in minutes) SET/UNSET not available
LOG	*	FALSE	If TRUE all direct mode I/O is listed
NOERRS	*	FALSE	If TRUE error messages not output. Direct mode = FALSE
NOSTOP	*	FALSE	If TRUE do not stop if ERROR set.
OLDABEND	After any command	-	Set if an abnormal end has occurred since the last direct mode command
OLDCOND	After any command	-	Set if a non-zero CONDCODE since last direct mode command
OUTLOW	*	FALSE	If TRUE, command output has upper case converted to lower case
USERID	At login	0	Retained for compatibility
USERPROG	Each time that a library or user command is loaded into the USER shell	Null String	Name of the command loaded (the full filename - if the last command was EDIT, USERPROG will have the value SYS.EDIT).
VERIFY	*	FALSE	If TRUE lists all indirect mode commands

CONDITION CODE CONVENTIONS

By convention the following values of condition code are used:

Code	Type of Error	Interpretation
0	No errors	Command results valid
2	Warnings	Command results valid but should be checked
4	Errors	Command results probably invalid
8	Fatal errors	Command fails
16	Data Management error	Command fails

The contents of the A register after an error indicate the following:

```
 0                          15 16          23 24         31
[             INFO            |   ERRTYPE    |  CONDCODE  ]
```

IF ERRTYPE = 0   ERRVAL = CONDCODE
otherwise ERRVAL = INFO.

See the manual *OS4000 Job Control Language, 85-62007*.

ATTENTION MECHANISM

Used for on-line working. Commands of the form:

?A*command*

cause execution of *command* out of normal sequence.

If *command* is null the mechanism aborts the program running.

Commands available to the attention mechanism are:

ALTER	**CHANGE**	**CLAIM**	**CLAIMWAIT**	**CONTEXT**	**CONTINUE**	**CREATE**
DEFAULT	**DELETE**	**DISABLE**	**DISPLAY**	**EMPTY**	**EXPECT**	**EXTEND**
FREE	**JOB**	**LOG**	**LOSE**	**OUTPUT**	**PASSWORD**	**PROFORMA**
PROTECT	**QUERY**	**REFERENCE**	**RENAME**	**REPEAT**	**SAY**	**SEND**
SET	**UNSET**	**VALUE**	**WHEN**			

LIBRARY COMMAND FORMAT

A command name must consist of alphanumeric characters, starting with a letter.

[*prefix*] *command-name* [(*environment*)] [*arguments*]^{R}

Prefix may be **RUN** (simple commands), **EXEC**, **SWITCH** (macro commands) or **LOAD** (user commands).

Environment is of the form:

([*System Keyword*[=] *argument*]$^{R}_{1}$)

System keywords are:

COMP	or	C	Computation time (IEU)
TIME	or	T	Execution time (mins)
PROFORMA	or	P	Proforma to be used
STREAM *n*	or	S *n*	Stream number

Arguments are of the form:

Keyword[=][*argtype*][(*default*)]

Argtypes are:

FILE [*]	File name required
STREAM [integer][*]	File, filelist or virtual peripheral
STRING [*]	Text string - if spaces are to be included, the string must be surrounded by quotes
BOOL	Presence of keyword
VALUE [*]	Passes value to command
NONE	Remainder of line passed undecoded

If the proforma is not specified, the initial default proforma will be used:

```
FROM=STREAM 3(*)
TO=STREAM 4(*/NEW)
LIST=STREAM 6(LP/NEW)
WITH=STREAM 5(SINK)
GRAPH=STREAM 7(*)
IN=STREAM 1(*)
OUT=STREAM 2(*/NEW)
OPT=STRING()
```

FILE ATTRIBUTES

Specified after a file as:

$\textbf{FILENAME}[/\textit{attribute}]^{R}$

Size

$\begin{bmatrix}\textbf{SIZE}\\ \textbf{Z}\end{bmatrix}(n1[,n2[,n3]][,\begin{bmatrix}\textbf{C}\\ \textbf{D}\end{bmatrix}])$

where:

n1 = primary allocation (blocks)	(=<65535)
n2 = size of extension (blocks)	(=<16383)
n3 = number of extensions	(=<32767)
C = contiguous area of disc	
D = discrete area of disc	

Certain size attributes are already specified and are referred to as:

Name (synonym)	**Allocation**
TINY (T)	(1,1,10)
SMALL (S)	(2,2,10)
NORMAL (N)	(5,5,10) This is the initial default
MEDIUM (M)	(10,10,10)
LARGE (L)	(50,20,20)
GIGANTIC (G)	(100,50,20)
INFINITE (I)	(5,5,-1)

Format

Attribute

Default is /LST2

LST[*b*]
LSB[*b*]
LRT[*b*][(*r*)]
LRB[*b*][(*r*)]
PST[*b*]
PSB[*b*]
PRT[*b*]
PRB[*b*]
CAT[*b*]
LIS[*b*][(*i*[,*p*])]

L = Logical
P = Physical
I = Indexed
S = Sequential
B = Binary
T = Text
R = Random
CAT = Catalogue

b = blocksize (in 256 byte units)
r = record length (should include 4 bytes for D.M.)(r+4=<b)
i = key length (bytes) in an indexed sequential file
p = position of start of key (characters). Default = 0

Disposition

Attribute	**File exists**	**File does not exist**
ADD	No action	File created
CRE	Error	File created
MPT	File emptied	Error
NEW	File deleted & recreated	File created
OLD (default)	No action	Error
SCR	File emptied	File created

Protection

The attribute is written in the form:

[*owner access*] [.*user access*] [(*string*]

Owner and user access take the form:

RW[U[D[C[X]]]]

R = read access
W = write access
U = update access

D = delete access
C = change access (planning purposes only)
X = execute access (planning purposes only)

(*string*) = password of up to 8 characters

Access values are:

F = free access
P = password access (becomes F if no password specified in the string)
N = no access

Only the first two characters of an access specification need be present. If less than the full six characters is given, values are taken from the current local default FFFFFF.FNNNNF.

Blocksize

B*n*

Gives the required blocksize of a file where *n* is the size of blocks in 256 byte units.

File Allocation Type

$$\left[\begin{matrix}\textbf{CONT}\\\textbf{DISC}\end{matrix}\right]_1^1$$

This is an alternative to the size attribute form given above.
CONT = contiguous
DISC = discrete

File Security

$$\mathbf{L}\left[\begin{matrix}\mathrm{ON}\\ \mathrm{OFF}\end{matrix}\right]_{1}^{1}$$

The transaction logging service is available only to suitably privileged users of the CFS1 filing system.

Location

R (*discname.regionname*)

Initial default is determined by the working profile. Requires privilege to use non-default disc and region.

DATA FILES

Symbol	**Type of File**
.	Catalogue file (up to 8 characters)
&	Temporary file
%	System file (see table below)

System Files

%A	Object code file
%B	Process file
%C	Current file
%Hc	In-stream data. *c* is a terminating character
%Ic	
%L	Previous message file
%M	Message file
%O	Output file
%P	Previous current file
%W	Work file

MACRO LIBRARIES

Each macro in a macro library starts with the command:

MACRO *name* [*<cl c2>*] [*arglist*]

and is terminated by the next macro or end of file.

<cl c2> specifies opening and closing argument brackets

arglist gives the list of required arguments in the form:

name = [*argtype*] [*] [(*default*)]

argtype may be FILE, STREAM, STRING (default), BOOL, VALUE, NONE
default is the string to be used if none is given.

The macro is called thus:

$\left[\begin{array}{l}\textbf{RUN}\\ \textbf{EXEC}\\ \textbf{SWITCH}\end{array}\right]$ *library name* (*macro name*)

LIBRARY COMMANDS

NOTATION

The notation for specification of library commands in this section is as follows:

COMMAND [*KEY* = *argtype* (*default*)]$^{R}_{1}$

where: *COMMAND* is the command name
KEY is the keyword which may be omitted if the arguments entered are in the same order as specified
= is optional - a space may be used instead
argtype is the type of the argument to be entered. For further details refer to Library Command Format.

COMMANDS (Not all available. Some are available as optional packages.)

ACCOUNT IN=stream(*) OUT=stream(*/NEW) LIST=stream(LP/NEW) P=bool

AIDA OPT=string(TEST) PROC=string(USER) COMLIST=stream(%W/NEW)
PDUMP=stream(SINK/NEW/LSB) LIST=stream(LP/NEW) IN=stream(*)
OUT=stream(*/NEW)

ALGOL60 FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW)
OUT=stream(%M/NEW) OPT=string() ENTRY=string(ENTRYPOINT)

ALG60CLG FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
LIST=stream(SINK) LIB=stream(SYS.ALG60LIB) OPT=string()
ENTRY=string(ENTRYPOINT) WITH=stream(SINK) CONDC=value(2)
CONDL=value(0) ARGS=none()

ALGXREF FROM=stream(%C) TO=stream(%O/NEW) LIST=stream(LP/NEW)
OUT=stream(%M/NEW) OPT=string()

BABB FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW)
OUT=stream(%M/NEW) OPT=string*()

BABBCLG FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
LIST=stream(SINK) LIB=stream(SYS.LINKLIB) OPT=string()
WITH=stream(SINK) CONDC=value(0) CONDL=value(0) ARGS=none()

BARC FROM=stream(SINK) WITH=stream(SINK) LIST=stream(LP/NEW)
OUT=stream(*/NEW)

BASIC SPOOL=stream(LP/NEW) IN=stream(*) OUT=stream(*/NEW)
CAT=file(USER)

BASICT SPOOL=stream(LP/NEW) IN=stream(*) OUT=stream(*/NEW)
CAT=file(USER)

BEDIT FROM=file(&SAVE/ADD/L) TO=stream(%O/NEW) IN=stream(*)
OUT=stream(*/NEW) LIST=stream(LP/NEW)
BACKUP=stream(SINK/NEW) CAT=string(USER)

```
CBA      SEGS=value* OPT=string('TIMES PASSES') PROC=string(USER)
         LIST=stream(LP/NEW) OUT=stream(*/NEW)

CMBI     FROM=stream TO=stream(SINK) IN=stream(*) OUT=stream(*/NEW)

COBOL    FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW)
         OUT=stream(%M/NEW) OPT=string() PASS12=stream(%W1/NEW/LSB)
         PASS23=stream(%W2/NEW/LSB) XREF1=stream(%X1/NEW/LRB(124))
         XREF2=stream(%X2/NEW/LRB(124)) MODULESIZE=value(4)

COBOLCLG FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
         LIST=stream(SINK) LIB=stream(SYS.COBOLLIB) OPT=string()
         CONDC=value(1)

COLS     FROM=stream(%C) LIST=stream(LP/NEW) COLUMNS=value(2)
         WIDTH=value(60) SEP=value(2) LINES=value(65)

COMP     FROM=stream(%C) WITH=stream(%P) OUT=stream(%M/NEW)
         OPT=string(F*) ERRS=value(0) SKIP1=value(0) SKIP2=value(0)
         LENGTH=value(248)

CONTROL  IN=stream(%C) OUT=stream(*/NEW)

COPY     FROM=stream(%C) TO=stream(%O/NEW) OPT=string(TEXT)
         PHYSICAL=bool OUT=stream(*/NEW) LENGTH=value(0)

CORL     FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW)
         OUT=stream(%M/NEW) OPT=string*()

CORLCLG  FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
         LIST=stream(SINK) LIB=stream(SYS.CORLLIB) OPT=string()
         WITH=stream(SINK) CONDC=value(2) CONDL=value(0) ARGS=none()

DBFMT    FROM=stream(%C) LIST=stream(LP/NEW) OUT=stream(%M/NEW)

DBGEN    FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW)
         OUT=stream(%M/NEW)

DBRCV    FROM=stream(%C) LIST=stream(LP/NEW) OUT=stream(%M/NEW)

DEBFORT  PROC=stream(%B) FROM=stream(*) LIST=stream(LP/NEW)
         OUT=stream(*/NEW) MACROS=stream(%W/NEW/LRB (84)) ARGS=none()

EDIT     FROM=stream(%C) TO=stream(%O/NEW) WITH=steam(*)
         OUT=stream(*/NEW) LIST=stream(LP/NEW) BACKUP=stream(SINK/NEW)
         LENGTH=value(132)

EXAMINE  FILE=file(USER) OPT=string*() LIST=stream(*/NEW)
         OUT=stream(*/NEW) ACCTID=string() DAYS=value(0)
         FILENAME=string() FILETYPE=string() REGION=string()
         SINCE=string() USERID=string()

FCHK     FROM=stream LIST=stream(LP/NEW) OPT=string() IN=stream(*)
         OUT=stream(*/NEW)

FCOPY    FROM=string TO=string(SINK) OPT=string*() ANDFROM=string*()
         ANDTO=string*() DAYS=value(7) LIST=stream(SINK/NEW)
         OUT=stream(*/NEW) ACCESS=string() USERID=string()
         ACCTID=string() SINCE=string() BLOCKNO=value(32767)
         CLEANUP=bool IN=stream(*)
```

```
FOCUS    FROM=stream(*) LIST=stream(LP/NEW) OUT=stream(*/NEW)
         MACROS=steam(%W/NEW/LRB(84))

FORM     FROM=stream(%C) TO=stream(%O/NEW) LIST=stream(SINK/NEW)
         OUT=stream(%M/NEW) LENGTH=value(80)

FORT2    FROM=stream(%C) TO=stream(%A/LSB/NEW) LIST=stream(LP/NEW)
         OUT=stream(%M/NEW) OPT=string*()

FORT2CLG FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
         LIST=stream(SINK) LIB=stream(SYS.FORT2LIB) OPT=string()
         WITH=stream(SINK) CONDC=value(3) CONDL=value(0) ARGS=none()

FORT3    FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW)
         OUT=stream(%M/NEW) OPT=string*()

FORT3CL  FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
         LIST=stream(SINK) LIB=stream(SYS.FORT3LIB) OPT=string()
         WITH=stream(SINK) CONDC=value(3)

FORT3CLG FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
         LIST=stream(SINK) LIB=stream(SYS.FORT3LIB) OPT=string
         WITH=stream(SINK) CONDC=value(3) CONDL=value(0) ARGS=none

FORTV    FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW)
         OUT=stream(%M/NEW) OPT=string*()

FORTVCLG FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B)
         LIST=stream(SINK) LIB=stream(SYS.FORTVLIB) OPT=string()
         WITH=stream(SINK) CONDC=value(3) CONDL=value(0) ARGS none()

GEDIT    FROM=file(%C) TO=file(%O/NEW) WITH=stream(*)
         BACKUP=stream(SINK/NEW) LIST=stream(LP/NEW) OUT=stream(*/NEW)
         OPT=string*()

IBMC     FROM=stream(SINK/OLD) TO=stream(SINK/NEW) IN=stream(*)
         OUT=stream(*/NEW) OPT=string()

ISFC     NAME=file(SINK/NEW/LIS) FROM=stream(%C) OUT=stream(%M/NEW)
         FILLIND=value(100) FILLDAT=value(100) SORT=bool
         WORK=stream(%W/NEW/PRB16/Z(32,32,-1)

LGEN     FROM=stream(%A) TO=stream(%B/NEW/LRB(252)) WITH=stream(%C)
         LIST=stream(SINK/NEW) OUT=stream(%M/NEW)
         WORK=stream(SINK/NEW/LRB16(252))

LINK2    FROM=stream(%A) TO=stream(%B/NEW/LSB) WITH=stream(SINK)
         LIST=stream(SINK/NEW) LIB=stream(SYS.LINKLIB)
         OUT=stream(%M/NEW) WORK=stream(%W/NEW/LRB16(252)

MACG     FROM=stream(%C) TO=stream(%O/NEW) OUT=stream(%M/NEW)

MENU     FROM=stream(.MENUFILE) OPT=string() IN=stream(*)
         OUT=stream(*/NEW)

MERGE    FROM=stream(%C) TO=stream(%O/LST/NEW) KEY=string* BIN=bool
         PHYSICAL=bool PAD=bool STRIP=bool COLLATE=string()
         OUT=stream(%M/NEW)
```

ML1	FROM=stream(%C) TO=stream*(%O/NEW) LIST=stream(LP/NEW) OUT=stream(%M/NEW) OPT=string()
MTUT	MAIN=stream(SINK) COPY=stream(SINK) IN=stream(*) OUT=stream(*/NEW) LIST=stream(LP/NEW)
PANDG	FROM=stream(PAG1) TO=stream(PAG1/NEW) TRACK=value(1) FILE=value(0) CHECK=bool B=bool P=bool S=bool I=bool C=bool WRITE=bool WITH=stream(SINK) OUT=stream(%M/NEW) H=bool SVACAD=value(0)
PASA	PCODE=stream(%B) OUT=file(%M) OBJ=file(%A) OPT=string()
PASAL	PCODE=stream(%B) LIST=file(SINK) OUT=file(%M) OPT=string() PROF=file(%T) PROC=file(%B) CONDL=value(0) ARGS=none()
PASC	SOURCE=file(%C) LIST=file(SINK) PCODE=file(%B) PROF=file(%T) OUT=file(%M) CONDC=value(2) OPTIM=bool
PASCA	SOURCE=file(%C) LIST=file(SINK) PCODE=file(%B) PROF=file(%T) OUT=file(%M) CONDC=value(2) OPTIM=bool OBJ=file(%A) OPT=string()
PASCAL	SOURCE=file(%C) LIST=file(SINK) PCODE=file(%B) PROF=file(%T) OUT=file(%M) CONDC=value(2) OPTIM=bool OBJ=file(%A) OPT=string() PROC=file(%B) CONDL=value(0) ARGS=none()
PERT	FROM=stream(SINK) TO=stream(SINK/NEW/LSB) WITH=stream(%C) LIST=stream(*/NEW) OUT=stream(*/NEW)
PERTED	FROM=stream(*) TO=stream(*/NEW) OUT=stream(*/NEW) OPT=string()
PERTPBC	FROM=stream(SINK) TO=stream(SINK) MODS=stream(%C) SPEC=stream OPT=string(1000) LIST=stream(*) OUT=stream(*)
PREPROC	FROM=stream(%C) TO=stream(%O/NEW) OUT=stream(%M/NEW) OPTS=string*(null)
REDPRINT	FROM=stream(%C) LIST=stream(*/NEW) OUT=stream(SINK/NEW)
RPG2	FROM=stream(%C) TO=stream(%A/NEW/LSB) LIST=stream(LP/NEW) OUT=stream(%M/NEW)
RPG2CLG	FROM=stream(%C) OBJ=stream(%A) PROC=stream(%B) LIST=stream(SINK) LIB=stream(SYS.LINKLIB) CONDC=value(0) CONDL=value(0) ARGS none()
RPG2LINK	FROM=stream(%A) PROC=stream(%B) LIST=stream(SINK) LIB=stream(SYS.LINKLIB)
RUNFORT	PROC=stream(%B) FROM=stream(SINK) LIST=stream(LP/NEW) OUT=stream(*/NEW) MACROS=stream(%W/NEW/LRB(84)) ARGS=none()
SALVE	FROM=stream(%C) TO=file(%O/NEW) WITH=stream(*) OUT=stream(*/NEW) BACKUP=file(.SEBACKUP/NEW/FFFF.FFFF/Z(1,1,-1) WORK=file(%W/NEW) LIST=stream(SINK/NEW) LENGTH=value(132) SECIN=stream SECOUT=file OPT=string*=() CM=bool WINDOW=none(MIN=5,MAX=19)

```
SORT     FROM=stream(%C) TO=stream(%O/NEW/LST) KEY=string* BIN=bool
         PHYSICAL=bool PAD=bool STRIP=bool COLLATE=string()
         OUT=stream(%M/NEW) WORK=stream*(%W/NEW/PRB16/Z(32,32,127))

TAIL     FROM=stream(%C) TO=stream(%O/NEW) OUT=stream(%M/NEW)

TEDIT    FROM=stream(%C) TO=stream(%O/NEW) WITH=stream(*)
         OUT=stream(*/NEW) LIST=stream(LP/NEW) BACKUP=stream(SINK/ADD)

TOTESTER IN=stream(*) OUT=stream(*/NEW)

TOTLBLNK IN=stream(*) OUT=stream(*/NEW)

TOTLCHCK IN=stream(*) OUT=stream(*/NEW) TO=stream(*/NEW)
         LIST=stream(SINK/NEW) WORK=stream(%W/NEW)

TOTLLIST FROM=stream(%C) OPT=string(WH) LIST=stream(LP/NEW)
         IN=stream(*) OUT=stream(*/NEW)

TOTLLOAD FROM=stream(%C) IN=stream(*) OUT=stream(*/NEW)

TOTLSTAT IN=stream(*) OUT=stream(*/NEW) LIST=stream(LP/NEW)

TOTLUNLD TO=stream(%O/NEW/LSB) IN=stream(*) OUT=stream (*/NEW)

VPLTCLG  FROM=file(%C/OLD) OBJECT=file(%A/LSB16) PROCESS=file(%B)
         IN=file(*/OLD) OUT=file(*/NEW) WITH=file(SINK/OLD)
         LIST=file(SINK/NEW) OPT=string(WHE) WORK1=file(%W1)
         WORK2=file(%W2) WORK3=file(%W3) WORK4=file(%W4)
         PLOT=file(VE/(SPOOLED)/NEW) CONDC=value(0) CONDL=value(0)
         MAPPED=bool LINKED=bool SORTED=bool GINO=bool
         VTOR=bool NORUN=bool

XREF     FROM=stream(%C) LIST=stream(LP/NEW) TITLE=string()
         OUT=stream(%M/NEW) OPT=string()
```

SYSTEM COMMANDS

Note: An asterisk preceding a command name indicates that use of the command (or part of it) is privileged.

```
                          R
  ALTER [file/attributes]
                          1

* CHANGE id-number class

                                                               R
  CLAIM[WAIT] [virtual-device-name (volume-descriptor)]
                                                               1

  CONTEXT name [file]

  CONTINUE

  CREATE  [file/attributes]
                           1

  CURRENT stream

* DEFAULT ┌┌ C  ┐                                 ┐
          ││COMP│      [value]                    │
          ││ T  │                                 │
          ││TIME│                                 │
          │└    ┘1                                │
          │                                       │
          │┌ P  ┐                                 │
          ││PROF│      [name]                     │
          │└    ┘1                                │
          │                                       │
          │┌  A   ┐                               │
          ││ACCTID│    [name]                     │
          ││  U   │                               │
          ││USERID│                               │
          │└      ┘1                              │
          │                                       │
          │┌      ┐1                              │
          ││  L   │                               │
          ││LOCAL │    [attribute-string]         │
          ││  G   │                      1        │
          ││GLOBAL│                               │
          └└      ┘                               ┘

                R
  DELETE [file]
                1

  DISABLE  ┌                        R                ┐
           │[system-command-name]                    │
           │                        1                │
           │                                         │
           │ATTENTION ┌[system-command-name]R┐       │
           │          │ALL                  1│       │
           │BREAK     └                      ┘       │
           │USER                                     │
           └                                         ┘
```

```
* DISPLAY [CLASS class        ]
          [                   ]
          [PRIORITY priority  ]

* DISPLAY [BATCH] [keyword     ]
          [BS   ] [JOB job-name]

* DISPLAY ISPL [[JOB name ] [ALL    ]]
               [[USER name] [ARRIVED]]
               [            [AWAITED]]
               [document-name-list   ]

* DISPLAY RESOURCES [job-name]

* DISPLAY USERS [initiator]
                [mode     ]

  DISPLAY WRITERS [class] [maxsize] [state]
  Availability of this command depends on system configuration.

               R
  EMPTY  [file]
               1

  END

  EXEC  [filename] [arguments]     Default: %C

  EXPECT ISPL (name[/attributes])

                                 R
  EXTEND  value 1  value 2  [file]
                                 1

  FILE  file

                                    R
  FREE  [[virtual-device-name]       ]
        [ALL                         ]

  INPUT  [file]

  INTERRUPT   writer-name
              id-number[record-number]
                                        1

* JOB  jobname command

  LOAD  [filename] [arguments]     Default: %B

  LOG  [stream]

  LOGOUT

* LOSE  [id-number  ]
        [writer-name]
                       1
```

```
* LOSE ISPL  [ [USER name] [ALL    ] ]
             [ [JOB name ] [ARRIVED] ]
             [             [AWAITED] ]
             [ document-name-list    ]

  OUTPUT  [stream]

* PASSWORD  [string]  [USER [D]]
                      [D [USER]]
                      [DISC no ]

                  R
  PRINT  [stream]

  PROFORMA  name [(environment)] [argname[=] [argtype] [(default)]]

  PROTECT   [file/attributes]
                             1

  QUERY  (See P19)

  REFERENCE file1 [file2]

  RENAME name file

* REPEAT id-number [count]

  RESTART  [writer-name                       ]
           [id-number[record-number]          ]
                                                  1

  RUN  [filename] [arguments]

  SAY line-of-text

  SEND name[.name2] line-of-text

  SET name [=] [value]

  SUBMIT filelist

  SWITCH  [filename] [arguments]          Default: %C

  SYNONYM [word [expression] string [text]  ]
          [                                 ]
          [command expression               ]
          [                                 ]
          [word                             ]

  TYPE  [stream]

  UNSET name

                         R
  VALUE  [expression]
```

```
WAIT  [expression]

WAITFORCLAIM

WHEN

                                                            R
QUERY A[CCTD]

      /
      ATTRIBUTES
      L[OCAL]

      G[LOBAL]
      C[OMP]

      COMMANDS
      SC
                 1
                                                        R
      E[RROR]    ACCO      [expression]
                                       R
                                       1
                 BLDR
                 BS
                 CFS
                 CHAP
                 CLASS
                 CMNI
                 DISC
                 DMAN
                 DSPL
                 EMA
                 ICI
                 ISPL
                 JCL
                 JD
                 PCON
                 PROC
                 RAL
                 STOR
                 TLOG
                                      R
      I[NSTRUCTION] [expression]
                                      1

      P[ROFORMA]    ALL
                                R
                    [proforma]
                                  0
                                    R
      SY[NONYM] [synonym-name]

      T[IME]

      U[SERID]

      V[ARIABLES]   [VALUE]
```

MODULES USED FOR BATCH JOBS

Input Spooler

ISPL handles the input and storage of documents which set up and run batch jobs. It also handles data input from peripheral devices.

Resource Allocator

RAL deals with allocation of resources such as magnetic tape handlers, paper tape readers and non-spooled I/O.

Star

STAR is a pseudo-device. By default, all information logged by STAR is output to the line printer at the end of a batch job.

Batch Scheduler

BS controls the running of batch jobs. It holds the job queue and runs the jobs in order of their priorities.

CONTROL OF BATCH JOBS

SUBMIT [*filelist*]

Submits a batch job. By default the current file is used. The destination is the input writer (IWR).

The following commands are used in the control of batch jobs by a multi-access user and will only operate on jobs running under the requesting user's userid.

```
DISPLAY [BATCH]  username [HELD   ]
        [BS   ]1          [INPUT  ]
                          [READY  ]
                          [RUNNING]
```

Displays information about certain of the current batch jobs. By default all jobs are displayed.

DISPLAY BATCH JOB *job-name*

Displays information about specified job only. No information is given if job userid is not the same as the requesting user.

JOB *job-name* **REMOVE**

Removes the named job from batch job queue. If the job is already running it is killed. Job userid must be the same as the requesting user.

JOB DESCRIPTION STATEMENTS

Note: Spaces have no significance within a job description. Comments may be included; they must start with // and end with // or NL.

JOB (*userid*[/*title*])

Mandatory. Defines the userid. It must be the first command.

ACCOUNT *up to 6 alphanumeric characters*

Names the resource account. May be omitted if same as userid.

CPU *number*

CPU usage allowed (in IEUs).

DISCS *type number* (*volno/volname* [/**MOUNT**[(password)]])

Number of disc units required. With the inclusion of /MOUNT[(password)], the resource allocator mounts the disc on behalf of the user.

EXEC *number* $\begin{bmatrix}\textbf{SECS}\\ \textbf{MINS}\\ \textbf{HOURS}\end{bmatrix}$

Maximum time allowed for job. Default values are system dependent. Default unit is MINS.

INFORM [*userid*]

Informs named user when the job is completed or cancelled. By default, userid is as per JOB statement.

INPUT $\begin{bmatrix}\textit{name}[/\textit{attributes}]\\ \text{FOLLOWS }\textit{name}[/\textit{attributes}]\end{bmatrix}$

Defines the input documents. Given attributes override any previously defined. FOLLOWS names the following untitled document.

JCL *file list*

Defines initial source for the job's JCL commands and must not contain spaces. By default this is the following untitled document (ISPL(FOLLOWS)).

OPTION *up to 6 alphanumeric characters*

Defines the mode of working. By default NORM is used.

OUTPUT *file list*

The initial destination for the job's command output. By default this is STAR. The file list must not contain spaces and its components must already exist before the job is submitted.

PASSWORD *up to 8 alphanumeric characters*

Mandatory, unless no password exists or the userid for the batch job is also that of the submitter of the batch job.

PERIPHERALS [*up to 3 alphanumeric characters number*]^{R}

Defines class, and the number required of that class, of peripherals for exclusive usage.

QUEUE *up to 4 alphanumeric characters*

Enforces the order in which batch jobs are run; the lowest numbered job in the named queue is first.

ROUTE $\begin{bmatrix}\textbf{ALL}\ \textit{new-spooled-peripheral} \\ \textit{original}\ \ \textit{new-spooled-peripheral}\end{bmatrix}$

Changes the route of spooled output. The default routing of output depends on the input device or is system dependent.

$\begin{bmatrix}\textbf{STORELIM} \\ \textbf{CORELIM}\end{bmatrix}$ *number*[K]

Amount of store the job may allocate in units of 1024 bytes. Default value is system dependent.

TAPES *device-type + class n* $\left[(\begin{bmatrix}\textit{volno/volname} \\ \textit{new volname}\end{bmatrix} \begin{bmatrix}\textbf{/NOWPR} \\ \textbf{/WPR}\end{bmatrix}) \right]_{0}^{n}$

Number of magnetic tape (MT) units required as MT files (MTF) or MT virtual devices (MTD). The default of the write permit facility is site dependent.

TEMPSPACE *number*

Amount of temporary filing space the job may use (in Disc Space allocation Units). Default value is system dependent.

DOCUMENT TERMINATORS

Document terminators must stand alone and have the form:

warning-sequenceletter NL

warning-sequence is normally ******* but may be altered to any other non-alphanumeric characters by:

*****$**

which causes ******* to be replaced by **$$$**. This remains until changed again or the document in which it was changed is terminated.

The *letter* in the terminator determines the action taken by input spooling, either with the remainder of the file in which it occurs, or with the next file read by the same device. The available terminators, with their significances, are as follows:

*****Z** End of this document. More documents may follow as part of the same file.

*****U** End of this document. An untitled document follows as part of the same file and is to be read in text mode.

*****B** As above but for logical binary mode.

*****P** As above but for physical binary mode.

*****V** End of this document. No further documents are contained in this file. An untitled document follows in the next file to be read by this device, and is to be read in text mode.

*****C** As above but for logical binary mode.

*****Q** As above but for physical binary mode.

*****W** End of this file. The document currently being read is continued in the next file to be read by this device, and is to be read in text mode.

*****D** As above but for logical binary mode.

*****R** As above but for physical binary mode.

Documents read in logical mode (whether binary or text) may be terminated by end-of-file instead of one of the above terminators. This has the same effect as the terminator *****Z**.

Documents read in physical mode are terminated either by end-of-file, or by the end of the physical medium on which the file is held (for example, end of paper tape). Note however that physical mode input spooling is not supported in OS4000.

INPUT SPOOLING

Commands used with input spooling are:

EXPECT ISPL (*name*[/*attributes*])

> Causes input spooling module to expect to read a data document. Attributes (if specified) override any specified in the title line of the document. Title of expected document is *userid.name* where the userid is that of the job requesting it.

```
DISPLAY ISPL  [ [USERID name] [ARRIVED] ]
              [ [JOB name   ] [AWAITED] ]
              [               [ALL    ] ]
              [ document-name-list      ]
```

> Displays information about all or specified documents, or those awaited or received by a named job.

```
LOSE ISPL  [ [USER name]  [ARRIVED] ]
           [ [JOB name ]  [AWAITED] ]
           [              [ALL    ] ]
           [ document-name-list     ]
```

> Discards all or specified documents belonging to a job. Cancels any previous **EXPECT ISPL** commands.

ISPL(*document-name*)

> Refers to a document known to input spooling.

If an ISPL user error is detected, a message is output in the following form:

*****ISPL:***error-descr   device   stream-no*[*device-action*][*data-name*]

Definitions are given in the *Batch User Manual, DD1447.*

RESOURCE ALLOCATION

DISPLAY RESOURCES

Displays free resources and those allocated to the user's job.

CLAIM [*virtual-device-name* [(*volume-descriptor*)]^R]

Requests resources to be allocated to the job. *volume descriptor* requests the mounting of a specified volume.

CLAIMWAIT [*virtual-device-name* [(*volume-descriptor*)]^R]

Similar to CLAIM but anticipates that the resources will not be immediately available.

FREE $\begin{bmatrix}\textit{resource-list}\\ \textbf{ALL}\end{bmatrix}$

Renders all, or specified, resources available for re-allocation.

OUTPUT SPOOLING

Commands used with output spooling are:

DISPLAY [*Userid*] [**PRIORITY** *priority*] [**CLASS** *class*]

Displays information about documents being, or waiting to be, despooled.

DISPLAY WRITERS [*class*] [*max-size*] [*state*]

Displays information regarding the current state of despool writers. Availability of this command depends on the system configuration.

LOSE *document-id-number*

Causes the specified document to be removed from the despool queue. If it is currently being despooled it is aborted.

REPEAT *document-id-number* [count]

Causes *count* of further copies of the specified document to be produced. The default *count* is one.

AIDA COMMANDS

AT [*address*]$^{R}_{1}$

Specifies address of breakpoints.

BREAKS

Lists all breakpoints currently set.

BULK *stream*

Switches output from the **LIST** and **FDUMP** commands to the specified stream.

CALLS [**OFF**] $\left[\begin{array}{l}[\textit{route no}]^{R}_{1} \\ [\textit{symbolic name}]^{R}_{1}\end{array}\right]$

Monitors calls with any of the specified route numbers. *symbolic name* is the name of a route. Note that it must be preceded by an opening chevron.

When a monitored **CALL** is about to be executed, control is passed to AIDA at which point the following commands are also available:

B break execution of **CALL** in order to execute other commands.

D list all registers in hex.

C continue with execution of the **CALL**.

H continue but no **I/O** command or message is produced. Any state change is carried out.

R[[*data format*]*values*]$^{6}_{0}$

continue but do no state change. *values* are inserted into A, X, Y, Z, C, PAST 0 respectively. A null field leaves the register unaltered.

H and **R** used together result in no part of the **CALL** being executed.

DEFAULT [*print opts*]$^{R}_{1}$

Sets default values of print options for **FDUMP**.

END	Terminates AIDA.
$\textbf{ENDTEST}\left[\begin{array}{l}\textbf{ALL}\\ {[}\textit{process name}]^{R}_{1}\end{array}\right]$	Ends the testing of the specified process(es). This is a privileged command.
EVAL *integer expression*	Evaluates expressions, then gives result in hexadecimal and decimal.
$\textbf{FDUMP}\left[\begin{array}{l}\textbf{ALLSEGS}\\ \textbf{REALS}\\ \textit{print opts}\end{array}\right]\left[\begin{array}{l}\textbf{ALL}\\ {[}\textit{process name}]^{R}_{1}\end{array}\right]$	Dumps all relevant process information in hexadecimal and character (and floating point) form. The use of *process name* is privileged.
$\textbf{FREETEST}\left[\begin{array}{l}\textbf{ALL}\\ {[}\textit{process name}]^{R}_{1}\end{array}\right]$	Identical to **END** except that each process specified is left in the **going** state. Use of arguments is privileged.
$\textbf{GO}\left[\begin{array}{l}\textbf{ALL}\\ {[}\textit{process name}]^{R}_{1}\end{array}\right]$	The process runs until a program event occurs. Use of the arguments is privileged.
HISTORY [*integer*]	Lists the last *integer* items in STEP mode.
$\textbf{LIST}\ [[\textit{data format}]\textit{address range}]^{R}_{1}$	Lists data from specified address and in specified format.
MODE	Lists current operation mode.
$\textbf{MONITOR [OFF]}\left[\begin{array}{l}\textit{address range }[\textit{value}]\\ \textit{register name}\end{array}\right]^{R}_{1}$	Monitors a change in value or the attainment of the value. Control then returns to AIDA.
$\textbf{OFF}\left[\begin{array}{l}\textit{process name}\\ \textbf{ALL}\\ {[}\textit{address}]^{R}_{1}\end{array}\right]$	Removes all, or specified, breakpoints. Use of the *process name* argument is privileged.
PATCH *address*	Specifies the start address for patching code. Terminated by **E**.
$\textbf{PDUMP[ALLSEGS]}\left[\begin{array}{l}\textbf{ALL}\\ {[}\textit{process name}]^{R}_{1}\end{array}\right]$	Dumps the process's own (or all) segments in a reloadable binary process module format. *process name* is privileged.
REGS	Displays contents of all registers (except **RD** and **RE**) in hexadecimal format.
register name [*data format*]	Displays content of the register in specified format.

RESET	Resets the process to its initial state, except that patches to code or data are not removed. Additional access permission is required.
SET [*data format*]*address*[*value*]$^{R}_{1}$ *register namevalue*	Sets values into registers or store locations.
STEP [*integer*] *integer*	The process obeys *integer* instructions and then returns control to AIDA.
STEPMODE	Sets **STEP** mode.
STOP **ALL** R [*process name*] 1	Sets each process pseudo state to STOPPED until set running by an explicit **GO** command. This is a privileged command.
TEST [*process name*]$^{R}_{1}$	Places the process(es) named under test or defines a new current process. This is a privileged command.
UNSTEP	Reverts to fast mode, if possible.

For definitions of print options, data format and address, see the manual *OS4000 Program Development Aids, DD1392*

TERMINAL CONTROLS

There is a set of commands available at the terminal to control its operation. These commands consist of a single letter preceded by a warning character which is initially the query character (?). Throughout this section the query character is used but it may be replaced by whichever warning character is required. NL represents the new line sequence - normally carriage return, sp represents a space. Items within square brackets are optional.

Control Command	**Meaning**	**Format**
?A	Attention	**?A** *line* NL
	Interrupt	**?A** NL
?B	Buffered Mode	**?B** NL
?C	Delete Previous Character	*text* **?C** *text*
?D	Delete Line	*line* **?D** NL
?E	Enable Muted Prompt	[*name*]**?E** NL
?F	Reinstate Physical Mode	**?F** NL
?G	Buffered Prompt Mode	**?G** NL
?H	Hexadecimal Character	*text* **?Hnn** *text*
?I	Set Initial Length (0-4)	**?In** NL
?J	Enable Screen Mode Paging	**?Jn** NL Turn on, screen length=*n* System default if *n* unspecified **?J0** NL Turn off **?JPn** NL Set automatic paging for length=*n* or less **?JR** NL Output to run on # **?JTn** NL Change timeout to *n* # **?JB** NL Suppress blank lines # **?JB0** NL Enable blank lines # **?JBn** NL Suppress blank lines for # length=*n* or less #
?K	Control Shift	*text* **?K** *text*
?L	Lose Input Buffer	**?L** NL
?M	Mute Process	[*name*]**?M** NL
?N	Normal Shift	*text ?N text*
?O	Opposite Shift	*text* **?O** *text*
?P	Prompt Mode	**?P** NL
?Q	Reinstate Last Line	**?Q** NL

?R	Display Terminal Requests	**?R** NL	
?S	Select Destination	[*name*]**?S** *line* NL	
?T	Output Buffering On	**?T** NL	
?U	Output Buffering Off	**?U** NL	
?V	Verify Line	*line* **?V** NL	
?W	Set Screen Paging in Current Window only	**?W** NL Turn on **?WO** NL Turn off **?Wn** NL page length=*n* **?WR** NL Output to run on **?WTn** NL Change timeout to *n* **?WB** NL Suppress blank lines **?WBO** NL Enable blank lines	 # # # # # #
?X	Exchange Control Symbol	**?X**$\left[\begin{array}{l} ?\$ \\ C\$ \\ R \end{array}\right]$ NL where **$** is the new or replacement symbol *C* is the control symbol to be replaced *R* is the additional symbol to be removed	
?Y	Terminal Reset	**?Y** NL	
?Z	End of File	[*destn*]**?Z** NL	
?sp	Ignored	*text* ?sp *text*	
?NL	Continuation on next line	*text* ? NL	
?*PAD	Set Terminal In PAD Mode	**?*PAD** NL	#
?*PAGE	Set Terminal In PAGE mode	**?*PAGE** NL	#
?*TTY	Set Terminal In 'Old' TTY mode	**?*TTY** NL	# #

Note: # Denotes application to TF/TC only.

ERRORS

LOGIN ERRORS

account file too complex	Report error to system manager
accounts record type out of range	Report error to system manager
add module fail	EMA error adding user module
command process load fail	Report error to system manager
illegal option	Option not available to user
illegal user of account	User not listed in account
inconsistent accounts file	Report error to system manager
incorrect password	Wrong password quoted
initiator limits exceeded	No modules available for this particular initiator
insufficient computation time	User account overspent on computation time
insufficient elapsed time	User's elapsed time overspent
invalid mode	Report error to system manager
invalid request parameters	Report error to system manager
invalid virtual PNO in user profile	Report error to system manager
job exceeds maxtimes	Login has attempted to start a job whose resource limits exceed those available
job name already in use	Userid already logged in
login not allowed at this terminal	Self-explanatory
login not permitted at this time	Self-explanatory
login not permitted today	Self-explanatory
mode limit exceeded	No modules available for this particular mode
no core space	Report error to system manager
no disc record for user	Report error to system manager
no resource account	Report error to system manager
no temporary file space	Report error to system manager

RAL claim fail	Login has attempted to claim a resource which is not available
system full	No modules available
system not available - resetting	Self-explanatory
Too many tries	Wrong password entered 3 times
unknown account	Unknown account entered at login
unknown user	Unknown user entered at login
User Profile too complex	Report error to system manager
user too long - terminated	User has delayed logging in
Working profile not found	Report error to system manager

JCL ERROR MESSAGES

These are errors detected by the command process and are reported in the following format:

error *error no message* [*symbol concerned*]

If not in direct mode, the command line in error is also printed. The errors are listed in the table below.

Error No.	**Messages**	**Notes**
1	missing file title	No file title supplied for a system command which requires one.
2	unknown proforma	Proforma named in environment is not known.
3	illegal GOTO	GOTO may not occur in direct mode. No more than 255 consecutive GOTOs may occur.
4	no label	GOTO not followed by label.
5	illegal RETURN	RETURN may not occur in direct mode.
6	illegal STOP	STOP may not occur in direct mode.
7	label missing	The label specified for a GOTO does not exist in the current macro command or any outer macro commands.

Error No.	Messages	Notes
8	expression too complex	Overflow of stacks used in expression evaluation. Must be simplified.
9	member not found	The member specified by a macro library is not present in the library.
10	unknown command	An unknown command has been given.
11	too deep in macros	Only a limited number of levels of nested macros is allowed. This number is fixed at system generation and is normally 4.
12	missing THEN or ELSE	Conditional command has faulty structure.
13	bad nesting	More ELSEs than TESTs while skipping a command.
14	bad nesting	More close chevrons than open chevrons while skipping a command.
15	bad structure	End of file met while skipping a command.
16	unexpected word	THEN, MACRO or ARGUMENT unexpected
17	illegal end of file	Following continuation mark (-).
18	illegal string	Unterminated at end of line.
19	missing variable	In SET or UNSET command.
20	illegal filelist	Filelist contains illegal characters.
21	deleting system proforma	Not permitted to change proforma specified in system initial JCL.
22	missing expression	In VALUE command, EXTEND command etc.
23	setting system constant	Attempt to set ONLINE etc.
24	excess information	At end of system command.
25	missing condition	In IF..THEN or TEST..THEN construction.
26	unknown keyword	In environment of command.
27	environment faulty	Arguments incorrect in command environment.

Error No.	Messages	Notes
28	missing name	For example, a missing proforma has been found in a PROFORMA command.
29	missing keyword	In PROFORMA command, for example.
30	unknown attribute	In PROFORMA command, for example.
31	bad default string	End of line encountered in middle of default string.
32	dictionaries full	No more room for user variables, proformas etc.
33	surplus arguments	Extra positional arguments after last one allowed by proforma.
34	bad argument	Actual argument does not correspond to expected type.
35	repeated argument	Argument has been specified more than once when proforma does not permit it.
36	missing argument	Undefaulted argument has not been supplied.
37	bad default expn	Expression provided as default in proforma cannot be evaluated.
38	undetermined in-stream data file	Met end of file before terminator.
39	error in expression	Closing bracket missing, for example.
40	command unavailable	Not available in this version of the command process, or not available under attention mechanism or DISABLED.
41	load fail	Failed to load process.
42	delete fail	Failed to delete process.
43	unknown destination	For SEND command.
44	argument too long	Exceeds 252 characters.
45	value out of range	In an EXTEND command, for example.
46	password update failed	PASSWORD USER command failed. Password is unchanged.
47	name faulty	Argument for DISABLE not a word.
48	unknown name	Argument for DISABLE not known.

Error No.	Messages	Notes
49	cannot disable LOGOUT	LOGOUT command cannot be DISABLED.
50	wrong type of variable	Tried to set integer system variable to string or vice-versa,
51	string too long	Exceeds 252 characters, or expected length (e.g. 8 for password).
52	break unavailable	Attention mechanism is DISABLED.
53	facility unavailable	Command cannot be performed.
54	invalid command file type	The file specified as a command is not suitable. Only logical sequential text or binary files are allowed.
55	redefining system command	Attempt to use a system command name as a synonym.
56	ambiguous synonym	Attempt to create a synonym by shortening a command such that the set of shortened versions includes the name of a system command or another synonym.

If the command process has a Data Management error, this is reported in the following manner:

Data Management error *error no* **on stream** *stream no message*

If the command process receives an error from the catalogue filing system or the Data Management connect process, this is reported in the following format:

Filing Error *error no message* [*file involved*]

Further explanation of the errors is given under CATALOGUE FILING ERROR CODES below.

USER PROCESS ERRORS

If a user process ends abnormally, the following message is output:

Abnormal End Error *no*: *message* **S**=*no* **PAST**=*no*

showing the error that occurred, the value in the S-register and the current code segment.

Error No.	Messages	Notes
0	protection violation	Attempt to write to, or read segment to which the process has insufficient access.
1	undefined instruction	Attempt to obey an illegal instruction.
2	QCOUNT error	Too many messages sent but not received.
3	CALL instruction error	Error on SEND-type instruction.
4	SEG instruction error	Error on LOADSEG-type instruction.
5	CALL I/O instruction error	Error on SEND or LWCB-type instruction.
7	SEM instruction error	Error on CLAIM-type instruction.
8 or 14	computation time expired	Limit on CPU usage, set implicitly or explicitly in environment declaration, has expired.
9 or 50	segment break	Attempt to load unallocated segment.
10	route trap	Attempt to use invalid route.
11	elapsed time expired	Limit on elapsed time, set implicitly or explicitly in environment declaration, has expired.
12	user interrupt	Attention mechanism (?A) has been used to interrupt program.
20	overflow	Arithmetic overflow has occurred with the overflow trap set.
21	array bound error	Attempt to access outside the limit of an array with the array-bound trap set.

Error No.	Messages	Notes
22	paged address space violation	Attempt to use Paged Address Space when not entitled (i.e. PASFLAG in the master segment is not set).
50	segment break	See error 9.
51	overlay segment transfer failure	Probably hardware error, or failure to fetch segment.
52	unable to fit segment into main store	Self-explanatory.

If the process ends normally but with a non-zero condition code, this is reported in the format:

condition code *no* **S=** *no* **past=** *no*

BATCH ERRORS

JOB DESCRIPTION ERRORS

Error No.	Explanation
1	Job statement missing or faulty
2	Name expected
3	Number expected
4	Symbol out of context
5	Password missing or incorrect
6	Unknown user
7	Data management error reading job description
8	Job description feature not implemented
9	Job description too complex
10	Maximum CFSSPACE exceeded
11	Maximum core limit exceeded
12	Bad volume number
13	Number negative or too large

RESOURCE ALLOCATOR ERRORS

Error No.	Explanation
-1	Unknown user
-2	Unknown operation
-3	Bad segment access or size
-4	Unknown resource
-5	Reserve too large
-6	Claim not reserved
-7	Claim exceeds reserve
-8	Claim fails
-9	Volume not available
-10	CONX error (error code in AM)
-11	User not allowed in this operation
-12	Claim already in progress for this user
-13	Not enough list space to log user in (i.e. maximum number of users already logged in
-14	Operator has not acknowledged mount request
-15	Volume already allocated
-16	Resource has been AVOIDed
-17	Too many characters in volume number

DATA MANAGEMENT ERRORS

ERROR REPORTING

In OS4000 errors which are detected by Data Management are classified into two categories:

- Inferior Errors (Bit 4 of the ERROR CODE is unset)

  These are errors which are not caused by the user process doing anything illegal.

- Superior Errors (Bit 4 of the ERROR CODE is set)

  These are errors caused by the user process attempting something illegal.

In addition to these two categories there are eight different classes of error (0-7) and the error class is set in bits 5, 6 and 7 of the ERROR CODE.

- Class 0 - Normal Status

  These are conditions that the program normally expects to encounter while performing Data Management transactions. The most important member of this group is End-of-File.

- Class 1 - Parameter Errors

  A parameter error occurs when a program using the Input/Output system passes parameters that are illegal. The Input/Output system performs sufficient checking to ensure:

  - That the system is protected against misuse

  - That users are protected against each other

- Class 2 - Device Off-Line

- Class 3 - Device Not Available

  A device not available condition arises when a program attempts to use a stream and the device software is unable to allow the process access to the device.

  The possible reasons for this condition are as follows:

  - A process is already using the device, which may only be used by one process at a time

  - The device may be used by several processes, but the capacity for shared use is saturated

  - The device does not support the required modes of operation (e.g. an input stream is connected to a line printer).

- Class 4 - Status Indication

  Normal Device Status information (e.g. File-marks on magnetic tape).

- Class 5 - Device Failure

  This class contains various device failure situations as specified by the device controller.

- Class 6 - Data Errors

  A data error may arise when a program is transferring information to or from a medium. Both the software and the hardware may perform checks on input data. Detailed lists of data errors are to be found in the description of device-specific software.

- Class 7 - Buffer Irregularities

  This class is used for errors concerning buffer lengths that are processable but where some part of the record may be lost.

Due to the large number of possible error indications errors are grouped into two types:

- Type 0 Errors - Mutually exclusive (Bit 1 of the ERROR CODE is unset)

  Errors are identified by a unique error number in bits 8 to 15 of the error code. A type 0 error excludes the possibility of any other error on the same request as the error is reported immediately it is detected.

- Type 1 Errors - Non-mutually exclusive (Bit 1 of the ERROR CODE is set)

  This type of error is capable of being generated in the presence of others of the same type. Errors are identified by unique bits in the error number field.

Where one or more such errors occur together the error class chosen is that with the lowest number. Because of the potential number of possibilities none of these variations are listed.

Thus error returns of the form 8XXX or 9XXX are unique errors whereas an error indication CXXX may be the result of more than one error condition.

ERROR CODE FORMAT

Data Management errors are described by a half word of information;

0	1	2 3	4	5 7	8 11	12 15
E	T		C	Q	R	S

E: determines the error code:
Unset = No error (bits 1 to 15 undefined)
Set = Errors as defined by bits 1 to 15 (the ERROR CODE)

T: determines the error type:
Unset = Mutually exclusive error type (type 0)
Set = Non-mutually exclusive error type (type 1)

C: determines the error code
Unset = Inferior error
Set = Superior error

Q: determines the error class

R:
holds the error number or bits
S:

The error tables that follow are in numerical order of the error numbers ignoring C and Q. There are two tables, one for each 'type' of error.

To use the tables proceed as follows:

If T = 0 (ERROR CODE starts with hex digit 8), look at the Error Type 0 table to find your error. If ERROR CODE has the form 93RS, the error is one returned to Data Management by the catalogue filing system.

If T = 1 (ERROR CODE starts with hex digit C), look at the Error Type 1 table. This contains a number of different types of errors but as the error code can contain one or more errors, they are found by the following method:

a) Consider R and S as a bit pattern.

b) If the bit pattern is as defined in the table then that is the error.

c) If the bit pattern is a summation of error codes then each is an error.

For example:

C610 Block sequencing error

C605 - C610 (Parity or cyclic redundancy check error)

C704 (Input record longer than user's buffer)

Note: The error class need only correspond with one of the error codes (6 in the example above)

DATA MANAGEMENT ERRORS (TYPE 0)

Error Code (hexadecimal)	Meaning
8000	End of File
8901	Illegal Stream Number
8B02	Insufficient access permission to open file (file protected)
8B03	Attempt to OPEN file in wrong mode
8B04	No disc file access or stream not connected to file
8905	Stream not open (other than for OPEN and CLOSE)
8906	Invalid Operation
8D07	IOP error (e.g. service failure - IOP overload)
8908	Buffer outside segment
8909	Activity already outstanding on this stream (work in progress)
830A	Device in use at OPEN time
830B	Multiplex capacity full at OPEN time
810C	Record too long for block size available
820D	Device off line
830E	Write inhibited by device
850F	Device has timed out (e.g. attempt to read blank magnetic tape)
8910	Stream Control Segment is corrupt
8911	Invalid Control Command
8612	Sumcheck error
8913	Insufficient access to user's transfer buffer
8614	Invalid Column in coded mode (Card Reader)
8615	Repeated Rejection of transfer (Synchronous Communications)
8616	Block is corrupt (Record header length is invalid)
8517	Power failed
8618	Contention on Synchronous Communications line
8D19	Command rejected
831A	Device Temporarily Busy
891B	Record does not start on half word boundary when it should
801C	Requested Service not yet complete (call of CWAITO, GETCIO, PUTCIO or REPLACECIO)
801D	Despooled control command
891E	No Request Outstanding (call of WAITO and CWAITO)
851F	Mis-transfer on Card Reader
8320	Unable to reconnect in multifile sequence
8521	Abandon Call
8622	Magnetic Tape File structure faulty
8623	Magnetic Tape File absent if reading, or already present if writing
8324	Wrong or unsuitable Magnetic Tape for options specified
8325	Unable to allocate space for Stream Control Segment
8B26	Attention Stream already in use
8927	Attempt to OPEN an already OPEN stream
8528	Line failure on non-dialled communications (comms.) line
8529	Line failure on dialled comms. line without auto-answer
852A	Line failure on dialled comms. line with auto-answer
842B	Terminal interrupt other than by the break key
8X2C	Reserved for comms. line status indication
812D	Invalid Control Sequence
832E	Facility unavailable on this device
8B2F	HDLC logical disconnect mode
8630	Defective disc track
8531	Disc not available
8532	Disc timed out
8633	Compare error on disc

Error Code (hexadecimal)	Meaning
8634	CRC error on disc
8635	Address error on disc
8636	Seek error on disc
8637	Protected track
8B38	Disc not connected to controller
8B39	Disc index number too large
8B40	Change to file denied - logging inoperative
893A	Transfer request out of conceptual disc bounds
893B	Access to disc not permitted
8X3C to 8X3F	Reserved for disc errors
8X40	Reserved for future use
8541	Card Reader Motor Failure
8042	Backspace off start of file (file is left positioned at start)
8043	Key not found in indexed SEEK (positioned at next record in sequence beyond missing key)
8044	Key in indexed SEEK is shorter than key length of file (positioned at next record in sequence)
8145	Key already in file on indexed PUT (request ignored)
8146	Incorrect key in indexed REPLACE (request ignored)
8147	Record too short for key on indexed PUT or REPLACE (request ignored)
8148	Key sequence error on sequential PUT to indexed file (request ignored)
8949	Unable to use Physical Mode Buffer (user should supply a buffer segment with T access)
834A	Unable to allocate work segment
894B	Stream open when attempting connect
834C	Multiplex capacity full at connect time
8B4D	Remote connection failed
8B4E	Inter-computer link failure or closure
894F	Corrupt locate segment
8950	Locate buffer already free
8351	Driver connect parameter error - Parameter fixed
8352	Driver connect parameter error - String too long
8353	Driver connect parameter error - Value not supplied
8354	Driver connect parameter error - Numeric format error
8355	Driver connect parameter error - Name unknown
8356	Driver connect parameter error - Illegal separator
8357	Driver connect parameter error - No parameter required
8B58	Driver has been deleted from the system
9859	No free locate buffers available
895A	Too many (>120) or not enough room for locate buffers
835B	Unable to allocate locate segment
835C	Connect parameter out of range or inconsistent
895D	Command qualifier out of allowable range
855E	Timeout waiting for 'Ready for Sending'
835F	Parameter not variable in the hardware configuration
8360	Parameter may not be changed by user
8361	Parameter does not exist
8362	Compulsory parameter omitted
8363	Parameter has illegal value
8964	X25 misuse of the q bit
8965	X25 misuse of the m bit
8466	X25 Call has been reset

Error Code (hexadecimal)	Meaning
8467	X25 Interrupt packet has been missed
8968	Request not on the route used for specified stream
8069	X25 Remote Clear with data supplied
836A	Appropriate personality not current
836B	Command interrupted by other system activities (TF)
836C	Invalid controller response
816D	X25 Insufficient space for call statistics
896E	X25 Parameter modification not allowed
836F	X25 PVC Multiplex capacity full
8970	X25 PVC Id out of range
8471	Transport Service - Address received with error indication
8472	Transport Service - Reset received
8473	Transport Service - Address received
8474	Transport Service - Expedited data received
837F 8180 to 8187	CAMAC - Reserved for use by interface routines
8188	CAMAC - Illegal parameter list
8189	CAMAC - Invalid operation
818A	CAMAC - Module already booked
818B	CAMAC - Module already owned by you
818C	CAMAC - Module protection error
818D	CAMAC - Block transfer multiplexing limit exceeded
818E	CAMAC - No X or Q during LAM block transfers
818F	CAMAC - Cancel request for non-existent block transfer
8190	CAMAC - Block transfer ID already in use
8191	CAMAC - User segment error
8392	CAMAC - Branch offline
8393	CAMAC - Crate offline
8394	CAMAC - Station unoccupied
8395 to 8XFF	Reserved
93XX	Disc Filing error XX- See Catalogue Filing Errors. Note that XX is in hexadecimal.

DATA MANAGEMENT ERRORS (TYPE 1)

Error Code (hexadecimal)	Meaning
C601	Parity error on the last character in the buffer or cyclic redundancy check error
C602	Code conversion error
C704	Buffer is full (i.e. input record is longer than user's buffer)
C408	End of medium (includes blank header in binary file)
C610	Block sequencing error
C520	Device failure
C440	File mark found (MT) or wrong block terminator (Synchronous Communications)
CF80	Wrong length record supplied on PUT/REPLACE

Note: Combinations of the above errors are possible. See notes on Error Code Format.

ERROR MESSAGE CONTROL

At OPEN time the user may select whichever action he wishes the system to take for each class of error. If he does not wish to make such a specification, the system provides defaults suitable for most user processes.

The ERROR OPTION is part of the OPEN command. The default is obtained by leaving RX bit 0 as zero. If RX bit 0 is 1 then RY defines a halfword option. Class 0 errors are always returned to the user process and not to its owner.

	ERROR OPTION Bits Set (in RY)															
Meaning	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Suppress error message to owner for Error Class indicated	0		1		2		3		4		5		6		7	
Return error to user for Error Class indicated		0		1		2		3		4		5		6		7

Pairs of bits are used to control action on a given error class. The class number is controlled by a given effect. In this case:

00	Report error to owner but not to user process
01	Report error to owner and return error to user process
10	DO NOT USE
11	Return error to user process only

Default Error Handling

When bit 0 of RX is not set the system provides default error handling as follows:

- Logical Working

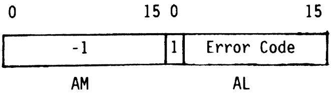
  The default is equivalent to error option @C000 so that class 0 errors are returned only to the user process while all other classes are reported only to the owner.

- Physical Working

  The default is equivalent to error option @00FF so that errors of classes 1 - 3 are reported only to the owner while errors of classes 0 and 4 - 7 are returned only to the user process.

CATALOGUE FILING ERROR CODES

Errors are reported by the routines with AM = -1, AL bit 0 = 1 and a code from the table below in AL bits 1 to 15:



When Data Management is reporting a CFS error this will appear as 93xx where xx is a hexadecimal number.

Error Code		Meaning
(decimal)	(hex)	
0	@8000	Serious error already reported to operator
1	@8001	Disc not mounted
2	@8002	Region not present or has wrong use
3	@8003	File not found
4	@8004	Too many files connected to this process
5	@8005	Attempt to disconnect an open stream
6	@8006	Reserved
7	@8007	File being deleted
8	@8008	File connected maximum number of times
9	@8009	Maximum connections exceeded
10	@800A	Disc being dismounted
11	@800B	Not enough room
12	@800C	Extensions capabilities exceeded
13	@800D	Disc already mounted or claimed
14	@800E	Name in use
15	@800F	Region hardware protected
16	@8010	Region directory full
17	@8011	Reserved
18	@8012	Insufficient access permission
19	@8013	Domain descriptor corrupt
20	@8014	Attempt to delete file without delete access (non-owner)
21	@8015	Password required to delete file
22	@8016	Stream already open (to simple device)
23	@8017	Illegal stream number
24-31	@8018-1F	Reserved
32	@8020	Illegal request parameters
33	@8021	Attempt to delete or empty a non-empty catalogue
34	@8022	Parent or destination of ODP is not a catalogue
35	@8023	Attempt to create a reference to temporary file
36	@8024	Request to delete reference or ODP used on a branch record
37	@8025	Request to delete branch record used on a reference or ODP
38	@8026	Request to change name of reference or ODP
39	@8027	Parent catalogue not found or base of genealogy is not master catalogue
40	@8028	Attempt to create or delete a master catalogue

Error Code		Meaning
(decimal)	(hex)	
41	@8029	Segment sent does not wholly contain the buffer or has insufficient access or the buffer is incorrectly aligned.
42	@802A	Disc for file create request not same as disc containing parent catalogue
43	@802B	Illegal filetype
44	@802C	Blocksize =<0/>16 Kbytes
45	@802D	Number of blocks for initial allocation not positive
46	@802E	Record size =<4 or >=blocksize -4
47	@802F	Illegal character in file name or context (e.g. a dot)
48	@8030	Context contains more than two references
49	@8031	Key specification invalid
50	@8032	Attempt to mix local and remote files in reference
51	@8033	Reference or ODP contains a reference or ODP
62-63	@8034-39	Reserved
64	@8040	Directory record cannot be accessed
65	@8041	Software protected directory entry
66	@8042	Array too small to hold genealogy
67-95	@8043-5F	Reserved
96	@8060	Failed to mount temporary file region
97	@8061	Temporary file region directory full
98	@8062	Despooler's master file full
99	@8063	Indexed sequential filing not available
100	@8064	Spooling not started
101	@8065	Filing system not started
102-127	@8066-7F	Reserved
128	@8080	File already open for exclusive use
129	@8081	File already open - cannot give exclusive use
130	@8082	Cannot open - insufficient access
131	@8083	File cannot be extended - no more room
132	@8084	Attempt to open a stream already open
133	@8085	File locked
134	@8086	End of medium
135	@8087	Reserved
136	@8088	Maximum opens exceeded
137-149	@8089-95	Reserved
150	@8096	Filing system queues full
151	@8097	Privileged operation not permitted
152	@8098	Not allowed to specify non-default region
153	@8099	Not allowed to connect this process
154	@809A	Module number illegal
155	@809B	Reserved
156	@809C	CFS busy
157-159	@809D-9F	Reserved
160	@80A1	Operation only permitted on catalogues
161-191	@80A2-BF	Reserved
192	@80C0	Userid not known
193	@80C1	Too many passwords
194	@80C2	Password not found
195	@80C3	Array is too small to contain details of all contexts requested
196-199	@80C4-C7	Reserved
200	@80C8	Attempt to redefine an initial context
201	@80C9	Too many context pointers

Error Code		Meaning
(decimal)	(hex)	
202	@80CA	Attempt to delete initial context pointer
203	@80CB	Context pointer or temporary file not found
204	@80CC	Connection through remote context not allowed
205	@80CD	Reserved
206	@80CE	Unknown account
207	@80CF	Accounting switched off - cannot allocate space
208	@80D0	Unexpected ACCO error - cannot allocate space
209	@80D1	Reserved
210	@80D2	Temporary filing space allocation would be exceeded
211	@80D3	Permanent filing space allocation would be exceeded
212-214	@80D4-D6	Reserved
215	@80D7	Job identifier already in use
216	@80D8	Too many users
217	@80D9	Users still logged on with initial context on this disc
218	@80DA	Password segment allocation failure
219	@80DB	Attempt to overwrite temporary file name
220	@80DC	Context not owned by user
221	@80DD	Region already mounted/claimed
222	@80DE	Too many claims
223	@80DF	Disc/region not claimed
224	@80E0	Insufficient space to add temporary filing region
225	@80E1	No temporary filing regions available
226	@80E2	Temporary region not known to CFS
227	@80E3	Disc transfer failure
228	@80E4	Not initialised for OS4000 filing
229	@80E5	Disc not online
230	@80E6	Illegal disc number
231	@80E7	Unknown filing system
232	@80E8	Named disc not mounted on given device
233	@80E9	Claim rejected
234	@80EA	Release rejected
235	@80EB	Incorrect password for disc
236	@80EC	Too many tries
237	@80ED	Disc state incompatible with request
238	@80EE	Too many discs
239	@80EF	Comment too long
240	@80F0	Master catalogue cannot have a parent
241	@80F1	ODP or reference from a variable disc
242	@80F2	Request failed
243-245	@80F3-F4	Reserved
246	@80F6	Module already added
247	@80F7	USER index full
248	@80F8	Space allocation failure
249	@80F9	Threads exhausted
250-255	@80FA-FF	Reserved

Error codes greater than 255 (FF) can be returned. These are Data Management errors encountered by the catalogue filing system, for example, when attempting to read a catalogue.

Other errors detected within CONX or the filing routines themselves are:

AL = -1	Process has no stream
-2	Stream open
-3	Illegal stream number
-4	Syntax error in file list
-5	Invalid process number
-6	Illegal userid
-7	Invalid buffer segment
-8	Device not claimed
-9 -10 -11	Buffers full (temporary overflow)
-13	Illegal module number specified (by privileged user)
-14	Illegal virtual peripheral name
-15	Buffers full
-16	CONX busy
-17	Peripheral unavailable
-18	File name faulty
-19	Syntax error in attribute string
-20	Operation unavailable
-21	Filename too long
-22	Required attributes not present
-23	Buffer too small
-24	Reserved
-25	INIT_FILE_LIB has not been called
-26	Attributes array faulty
-27 to -31	Reserved
-32	Document already present
-33	Document already expected
-34	Document abandoned (timeout)
-35	Unknown document
-36	Document abandoned (operator action)
-37	Unknown job
-38	Unknown user
-39	Too many users in the system
-40	Too many documents in the system
-41	ISPL master file full
-42	Document not yet present

Note that errors -32 to -42 are produced through having accessed ISPL.

CHARACTER CODES

1. 7 Bit code (hexadecimal)
2. With even parity (hexadecimal)
3. Character
4. Card rows punched
5. Coded byte value (hexadecimal)

1	2	3	4	5
00	00	NUL	12-0-9-8-1	B9
01	81	SOH	12-9-1	31
02	82	STX	12-9-2	32
03	03	ETX	12-9-3	33
04	84	EOT	9-7	17
05	05	ENQ	0-9-8-5	9D
06	06	ACK	0-9-8-6	9E
07	87	BEL	0-9-8-7	9F
08	88	BS	11-9-6	56
09	09	HT	12-9-5	35
0A	0A	LF	0-9-5	95
0B	8B	VT	12-9-8-3	3B
0C	0C	FF	12-9-8-4	3C
0D	8D	CR	12-9-8-5	3D
0E	8E	SO	12-9-8-6	3E
0F	0F	SI	12-9-8-7	3F
10	90	DLE	12-11-9-8-1	79
11	11	DC1	11-9-1	51
12	12	DC2	11-9-2	52
13	93	DC3	11-9-3	53
14	14	DC4	9-8-4	1C
15	95	NAK	9-8-5	1D
16	96	SYN	9-2	12
17	17	ETB	0-9-6	96
18	18	CAN	11-9-8	58
19	99	EM	11-9-8-1	59
1A	9A	SUB	9-8-7	1F
1B	1B	ESC	0-9-7	97
1C	9C	FS	11-9-8-4	5C
1D	1D	GS	11-9-8-5	5D
1E	1E	RS	11-9-8-6	5E
1F	9F	US	11-9-8-7	5F

1	2	3	4	5
20	A0	space	no punch	00
21	21	!	12-8-7	2F
22	22	"	8-7	0F
23	A3	vary	8-3	0B
24	24	vary	11-8-3	4B
25	A5	%	0-8-4	8C
26	A6	&	12	20
27	27	acute	8-5	0D
28	28	(	12-8-5	2D
29	A9	)	11-8-5	4D
2A	AA	*	11-8-4	4C
2B	2B	+	12-8-6	2E
2C	AC	,	0-8-3	8B
2D	2D	-	11	40
2E	2E	.	12-8-3	2B
2F	AF	/	0-1	81
30	30	0	0	80
31	B1	1	1	01
32	B2	2	2	02
33	33	3	3	03
34	B4	4	4	04
35	35	5	5	05
36	36	6	6	06
37	B7	7	7	07
38	B8	8	8	08
39	39	9	9	10
3A	3A	:	8-2	0A
3B	BB	;	11-8-6	4E
3C	3C	<	12-8-4	2C
3D	BD	=	8-6	0E
3E	BE	>	0-8-6	8E
3F	3F	?	0-8-7	8F

CHARACTER CODES (Continued)

1	2	3	4	5
40	C0	@	8-4	0C
41	41	A	12-1	21
42	42	B	12-2	22
43	C3	C	12-3	23
44	44	D	12-4	24
45	C5	E	12-5	25
46	C6	F	12-6	26
47	47	G	12-7	27
48	48	H	12-8	28
49	C9	I	12-9	30
4A	CA	J	11-1	41
4B	4B	K	11-2	42
4C	CC	L	11-3	43
4D	4D	M	11-4	44
4E	4E	N	11-5	45
4F	CF	O	11-6	46
50	50	P	11-7	47
51	D1	Q	11-8	48
52	D2	R	11-9	50
53	53	S	0-2	82
54	D4	T	0-3	83
55	55	U	0-4	84
56	56	V	0-5	85
57	D7	W	0-6	86
58	D8	X	0-7	87
59	59	Y	0-8	88
5A	5A	Z	0-9	90
5B	DB	[	12-8-2	2A
5C	5C	\	0-8-2	8A
5D	DD	]	11-8-2	4A
5E	DE	up arrow	11-8-7	4F
5F	5F	underline	0-8-5	8D

1	2	3	4	5
60	60	grave	8-1	09
61	E1	a	12-0-1	A1
62	E2	b	12-0-2	A2
63	63	c	12-0-3	A3
64	E4	d	12-0-4	A4
65	65	e	12-0-5	A5
66	66	f	12-0-6	A6
67	E7	g	12-0-7	A7
68	E8	h	12-0-8	A8
69	69	i	12-0-9	B0
6A	6A	j	12-11-1	61
6B	EB	k	12-11-2	62
6C	6C	l	12-11-3	63
6D	ED	m	12-11-4	64
6E	EE	n	12-11-5	65
6F	6F	o	12-11-6	66
70	F0	p	12-11-7	67
71	71	q	12-11-8	68
72	72	r	12-11-9	70
73	F3	s	11-0-2	C2
74	74	t	11-0-3	C3
75	F5	u	11-0-4	C4
76	F6	v	11-0-5	C5
77	77	w	11-0-6	C6
78	78	x	11-0-7	C7
79	F9	y	11-0-8	C8
7A	FA	z	11-0-9	D0
7B	7B	{	12-0	A0
7C	FC	vertical	12-11	60
7D	7D	}	11-0	C0
7E	7E	overline	11-0-1	C1
7F	FF	DEL	12-9-7	37

CONVERSION TABLES

BINARY CONVERSION

DEC	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	0001 0000

POWERS OF 16 TABLE

16**n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 848 976	15

POWERS OF 2

2^n					n	2^{-n}							
				2	1	.5							
				4	2	.25							
				8	3	.125							
				16	4	.062	5						
				32	5	.031	25						
				64	6	.015	625						
				128	7	.007	812	5					
				256	8	.003	906	25					
				512	9	.001	953	125					
			1	024	10	.000	976	562	5				
			2	048	11	.000	488	281	25				
			4	096	12	.000	244	140	625				
			8	192	13	.000	122	070	312	5			
			16	384	14	.000	061	035	156	25			
			32	768	15	.000	030	517	578	125			
			65	536	16	.000	015	258	789	062	5		
			131	072	17	.000	007	629	394	531	25		
			262	144	18	.000	003	814	697	265	625	5	
			524	288	19	.000	001	907	348	632	812	25	
		1	048	576	20	.000	000	953	674	316	406	625	
		2	097	152	21	.000	000	476	837	158	203	125	
		4	194	304	22	.000	000	238	418	579	101	562	
		8	388	608	23	.000	000	119	209	289	550	781	5
		16	777	216	24	.000	000	059	604	644	775	390	25
		33	554	432	25	.000	000	029	802	322	387	695	313
		67	108	864	26	.000	000	014	901	161	193	847	656
		134	217	728	27	.000	000	007	450	580	596	923	828
		268	435	456	28	.000	000	003	725	290	298	461	914
		536	870	912	29	.000	000	001	862	645	149	230	957
	1	073	741	824	30	.000	000	000	931	322	574	615	479
	2	147	483	648	31	.000	000	000	465	661	287	307	739
	4	294	967	296	32	.000	000	000	232	830	643	653	870
	8	589	934	592	33	.000	000	000	116	415	321	826	935
	17	179	869	184	34	.000	000	000	058	207	660	913	467
	34	359	738	368	35	.000	000	000	029	103	830	456	734
	68	719	476	736	36	.000	000	000	014	551	915	228	367
	137	438	953	472	37	.000	000	000	007	275	957	614	183
	274	877	906	944	38	.000	000	000	003	637	978	807	092
	549	755	813	888	39	.000	000	000	001	818	989	403	546
1	099	511	627	776	40	.000	000	000	000	909	494	701	773

HEXADECIMAL AND DECIMAL CONVERSION

HEXADECIMAL COLUMNS											
6		5		4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
0123		4567		0123		4567		0123		4567	
BYTE				BYTE				BYTE			