# ERRORS

**COLLABORATORS**

| | *TITLE* :<br><br>ERRORS | | |
|---|---|---|---|
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | January 13, 2023 | |

**REVISION HISTORY**

| NUMBER | DATE | DESCRIPTION | NAME |
|---|---|---|---|
| | | | |

# Contents

# Chapter 1

# ERRORS

## 1.1   main

                        Guide to AmigaOS Fail Codes, Errors
                and Guru Mediation Numbers



                1.
                   Fail Codes
                                2.
                   AmigaDOS Errors
                                3.
                   Guru Meditation
                                4.
                   About the Author




## 1.2   Fail Codes

Fail Codes

When an Amiga program finishes, it returns an exit code to
tell AmigaOS whether that program has finished ok or not.
The number returned is based on 4 levels:

     0 - Program finished ok

     5 - Program finished with a warning

    10 - Program finished with an error

    20 - Program failed to finish

You can have numbers inbetween or greater if you wish but in
most cases it uses the above.

You can test for the warning using the IF statement in AmigaDOS

and the keyword WARN, ERROR or FAIL. For example,

```
PROG1
IF FAIL
  Echo "Program failed"
  Quit
ENDIF
IF ERROR
  Echo "Program finished with an error"
  Quit
ENDIF
IF WARN
  Echo "Program gave a warning"
  Quit
ENDIF
Echo "Program ok"
```

Please note that you must check higher failures before
lower failures.

If you run a script such as startup-sequence, you want to ensure all
commands are executed even when one program fails for one reason
or another. To do this, use the FAILAT command to specify the failure
level so if it reaches that level, execution stops. A level of 21
would ensure all commands are run whether or not any of them fail
during execution.


## 1.3  AmigaDOS Errors

  Most AmigaDOS programs run ok but if for some reason comes across a
condition when it will fail, it will return an AmigaDOS error and
display it on the screen and in most cases with an explination.

```
Error Code      Description

103             Not enough memory available
105             Process table full (too many programs loaded)
114             Bad Template
115             Bad Number
116             Required Argument missing
117             Value after keyword missing
118             Wrong number of arguments
119             Unmatched quotes
120             Argument line invalid or too long
121             File is not an executable
122             Invalid resident library during load
202             Object in use (running)
203             Object already exists
204             Directory not found
205             Object not found
206             Invalid Window description (Con:)
209             Packet request type unknown
210             Stream name component invalid
211             Invalid object lock
212             Object not of required type
```

```
213          Disk not validated
214          Disk write-protected
215          Rename across devices attempted
216          Directory not empty
217          Too many levels
218          Device (or volume) not mounted
219          Seek failure
220          Comment too big
221          Disk full
222          File is protected from deletion
223          File is write protected
224          File is read protected
225          Not a valid DOS disk
226          No disk in drive
232          No more entries in directory
233          Object is soft link
234          Object is linked
235          Bad loadfile hunk (corrupt file)
236          Function not implemented
240          Record not locked
241          Record lock collision
242          Record lock timeout
243          Record unlock error
```

If a command fails you can type WHY to explain why a command failed
if there is an explination. Also typing FAULT and one or more numbers
will display the error description.


## 1.4   Guru Meditation


A Guru Meditation or Software Failure alert message appears when a
program fails or tries to write or read an invalid area of memory
and thus causing the Amiga to Crash.

Typical faults can be programs writing to memory they do not own or
have access to. Programs that trash memory or corrupt memory lists.
Programs using up too much memory. Programs that use machine code
that is not valid on an Amiga or use the wrong version of program
taylored to a specific processor on a different processor etc.

Format of a Guru number:

```
     SSGGEEEE AAAAAAAA
```

Where:
  SS is a Subsystem ID
  GG is a General Error
  EEEE is a Specific Error

  AAAAAAAA is the Address of the Task

The numbers will be displayed in Hexadecimal (0-9,A-F).

Subsystem ID Code

```
   00  CPU Trap
```

Libraries

```
   01  Exec
   02  Graphics
   03  Layers
   04  Intuition
   05  Math
   06  Clist
   07  DOS
   08  RAM
   09  Icon
   0A  Expansion
```

Devices

```
   10  Audio
   11  Console
   12  Gameport
   13  Keyboard
   14  Trackdisk
   15  Timer
```

Resource

```
   20  CIA
   21  Disk
   22  Misc
```

Other

```
   30  Bootstrap
   31  Workbench
   32  Diskcopy
```

If the first digit of the Subsystem ID is greater than 3 then it
is an unrecoverable error. Subtract 8 from ID to get Subsystem ID.

General Errors

```
   01  Insufficient memory
   02  MakeLibrary error
   03  OpenLibrary error
   04  OpenDevice error
   05  OpenResource error
   06  I/O Error
   07  No Signal
```

 Specific Alert Codes

Exec Library

```
 81000001  68000 exception vector checksum
 81000002  ExecBase checksum
 81000003  Library checksum error
 81000004  No memory to make library
```

```
81000005  Corrupted memory list
81000006  No memory for interrupt servers
81000007  InitAPtr
81000008  Semaphore corrupt
81000009  Free twice
8100000A  Bogus exception
```

Graphics Library

```
82010001  No memory for copper display list
82010002  No memory for copper instruction list
82000003  Copper list overload
82000004  Copper intermediate list overload
82010005  No memory for copper list head
82010006  Long frame, no memory
82010007  Short frame, no memory
82010008  No memory for flood fill
82010009  Text, no memory for TmpRas
8201000A  No memory for BltBitMap
8201000B  Region memory
82010030  MakeVPort
82011234  GfxNoLCM
```

Layers Library

```
83010001  Layers No Memory
```

Intuition Library

```
84000001  Unknown gadget type
04000001  Recoverable version of previous message
84010002  No memory to create port
84010003  Item plane allocation, no memory
84010004  Sub allocation, no memory
84010005  Plane allocation, no memory
84010006  Item box top less than RelZero
84010007  No memory to open screen
84010008  Open screen, raster allocation, no memory
84010009  Open sys screen, unknown type
8401000A  Add SW gadgets, no memory
8401000B  No memory to open window
8400000C  Bad State Return entering intuition
8400000D  Bad message received by IDCMP
8400000E  Weird message causing incomprehension
8400000F  Couldn't open the console device
```

DOS Library

```
07010001  No memory at startup
07000002  EndTask didn't
07000003  Opkt failure
07000004  Unexpected packet received
07000005  Freevec failed
07000006  Disk block sequence error
07000007  Bitmap corrupt
07000008  Key already free
07000009  Invalid checksum
```

```
 0700000A  Disk Error
 0700000B  Key out of range
 0700000C  Bad overlay
```

RAM Library

```
 08000001  Bad segment list
```

Expansion Library

```
 0A000001  Bad Expansion Free
```

Trackdisk Device

```
 14000001  Calibrate: seek error
 14000002  Delay: Error on time wait
```

Timer Device

```
 15000001  Bad request
 15000002  Bad supply
```

Disk Resource

```
 21000001  Get unit: already has disk
 21000002  Interrupt: no active unit
```

Bootstrap

```
 30000001  Boot code returned an error
```

* CPU traps (exceptions) are defined by the 680x0 CPU and not by
the Amiga OS. There are 256 vectors for CPU exceptions: the lower
64 are hardware defined, the upper 192 are user-definable interrupt
vectors.

Specific CPU traps you might see in a Guru are:

```
 00000002  Bus error
 00000003  Address error
 00000004  Illegal instruction
 00000005  Divide by zero
 00000006  CHK instruction
 00000007  TRAPV instruction
 00000008  Privilege violation
 00000009  Trace
 0000000A  Op Code 1010
 0000000B  Op Code 1111
```

Other CPU traps include autovectors for different level interrupts,
TRAP instruction vectors and reserved traps.

## 1.5  author

This guide was written by:

Peter Hutchison

Email: pjhutch@blizzard.u-net.com

Web: http://www.blizzard.u-net.com/