

# Appendix I - Error Message Index

---

The printable errors a running Intuition Engine produces. Grouped by source. Each entry has the exact text the machine prints, the numeric code the runtime uses internally, and a one-line explanation.

## I.1 IE64 BASIC runtime errors

---

The format on the screen is the message text, a space, the literal word ERROR, then (when raised from a running program rather than from direct mode) IN followed by the line number.

Code	Printed text	Meaning
1	SYNTAX	The tokeniser or parser cannot make sense of the statement.
2	DIVISION BY ZERO	A /, \, or MOD operator saw a zero denominator.
3	UNDEFINED LINE	GOTO, GOSUB, THEN, or RESTORE referenced a line that does not exist.
4	NEXT WITHOUT FOR	NEXT did not match a pending FOR.
5	RETURN WITHOUT GOSUB	RETURN did not match a pending GOSUB.
6	OUT OF MEMORY	The program, variable, array, or string area is full.
7	ILLEGAL QUANTITY	A function received an argument outside its domain (negative SQR, non-positive LOG, out-of-range CHR\$).
8	OVERFLOW	An arithmetic operation overflowed the double-precision numeric range.
9	TYPE MISMATCH	A numeric expression saw a string, or vice versa.
10	FC	Illegal function call. Raised by POKE16, POKE32, or POKE64 to an unaligned address, by a failed HOST action, and by other built-in helpers when they reject their argument shape.
11	REDIM	DIM named an array that already exists.

Additional message strings produced by specific verbs:

Verb context	Printed text
LOAD	?FILE NOT FOUND.
SAVE	?FILE ERROR (printed as a soft error; SAVE does not raise into the runtime).
RUN AOT	Compiling to native code... before compilation begins.
RUN AOT / COMPILE / TRANSPILE	?NO CODE TO COMPILE when the stored programme is empty.
RUN AOT / COMPILE / TRANSPILE	?COMPILE ERROR IN <line>: <reason> when a stored line cannot become native IE64 code.
RUN AOT / COMPILE / TRANSPILE / ASSEMBLE	?OUT OF MEMORY ERROR IN <line-or-0>: <reason> when the native-code arena, generated source, or output image is too large.

Verb context	Printed text
COMPILE / TRANSPILE / ASSEMBLE	?FC ERROR IN 0 for a bad output name.
COMPILE / TRANSPILE	?FILE ERROR IN 0 when the output image or generated source cannot be written.
ASSEMBLE	?FILE ERROR IN 0 when the matching assembly source is missing, unreadable, or too large.
ASSEMBLE	?COMPILE ERROR IN 0 when the IE64 source cannot be assembled.
TYPE	?SYNTAX ERROR IN 0 when the quoted filename is missing or malformed.
TYPE	?FILE NOT FOUND, ?FILE TOO LARGE, or ?FILE ERROR when the text file cannot be read.
TYPE	?NOT A TEXT FILE when the file contains binary control bytes.
RUN AOT / COMPILE / TRANSPILE	?COMPILE ERROR IN <line>: TYPE is direct-only when a stored line tries to compile TYPE.

## I.2 Machine monitor (IE Mon)

The monitor prints short, lowercase-prefixed messages. Every unrecognised command prints `unknown command`. The common ones:

Printed text	Meaning
?	Generic syntax error in a monitor command line.
unknown command	The first token did not name a command.
bad address	An address argument was outside the addressable range.
bad range	An end address was lower than its start address.
bad value	A value argument did not parse as a number.
no breakpoint at <addr>	bc was given an address with no breakpoint.
breakpoint list full	The breakpoint table cannot hold another entry.
cpu frozen	An operation tried to step a frozen CPU; thaw it first.
file error	A <code>save</code> or <code>load</code> sidecar command failed.

The monitor's `h` (hunt) and `c` (compare) commands print addresses of matches rather than diagnostic messages. The disassembler prints `???` for an unknown opcode.

## I.3 File I/O block

When the File I/O block (Chapter 35) fails, `FILE_STATUS` reads 1 and `FILE_ERROR_CODE` is one of:

Code	Meaning
0	OK (paired with <code>FILE_STATUS = 0</code> ).
1	Not found.
2	Permission.
3	Path traversal.

Code	Meaning
4	Range error: the staged data span would reach \$FFFF0000, wrap the 32-bit pointer, or exceed active RAM.

For a read whose name passes path validation but whose file cannot be opened, FILE\_RESULT\_LEN is cleared to 0. A program should still test FILE\_STATUS first and then read FILE\_ERROR\_CODE. The same 0 length is reported when a read or directory listing is refused with range error 4.

## I.4 HOST appliance block

The status byte at \$F1408 (Chapter 36) reads one of:

Code	Meaning
0	Running.
1	OK (terminal).
2	Error (terminal).
3	Cancelled by user (terminal).
4	Disabled (the system-action bridge is off).
5	Idle (no command has been fired).

A non-zero exit code at \$F140C after a terminal status of 2 gives the underlying action's exit value; the meaning is action-specific and is not normalised across subverbs.

## I.5 RUN loader block

The RUN loader block (RUN "<name>", Chapter 35) reports status:

Code	Meaning
0	Idle.
1	Loading.
2	Running.
3	Error.

On error, the error register reports:

Code	Meaning
0	OK.
1	Not found.
2	Unsupported type.
3	Path invalid.
4	Load failed.

RUN translates a non-zero result into ?FILE\_ERROR for the file-error cases and ?FC\_ERROR for the unsupported cases. For .ie64 images, load failed includes an image too large to fit at PROG\_START; the image is rejected before it can partially overwrite memory.

## I.6 Media loader

The media loader (SOUND PLAY, Chapter 23) reports status:

Code	Meaning
0	Idle.
1	Loading.
2	Playing.
3	Error.

On error, MEDIA\_ERROR reports:

Code	Meaning
0	OK.
1	File not found.
2	Bad format or read failure.
3	Unsupported extension.
4	Invalid filename.
5	File too large for the staging buffer.

For MIDI/MUS, bad SMF headers, unsupported SMF type 2, SMPTE timing, bad MUS score ranges, and unsupported MUS event types all report as bad format.

## I.7 Coprocessor

COSTATUS (Chapter 32) reports:

Code	Constant	Meaning
0	COPROC_TICKET_PENDING	Queued, not yet started.
1	COPROC_TICKET_RUNNING	Worker is processing.
2	COPROC_TICKET_OK	Completed successfully.
3	COPROC_TICKET_ERROR	Worker returned an error.
4	COPROC_TICKET_TIMEOUT	Wait deadline expired.
5	COPROC_TICKET_WORKER_DOWN	Worker is no longer running.

COWAIT blocks until the ticket reaches a terminal state or the timeout expires; call COSTATUS(ticket) afterwards to read the final code.

Raw coprocessor commands report their last command result through COPROC\_CMD\_STATUS and COPROC\_CMD\_ERROR:

Code	Constant	Meaning
0	COPROC_ERR_NONE	No command error.
1	COPROC_ERR_INVALID_CPU	The selected CPU type is not valid for this command.

Code	Constant	Meaning
2	COPROC_ERR_NOT_FOUND	The named worker was not found, or COPROC_CMD_START_MEM was given an empty or out-of-range guest-RAM image.
3	COPROC_ERR_PATH_INVALID	The worker name is not an accepted path.
4	COPROC_ERR_LOAD_FAILED	The worker image could not be loaded or started.
5	COPROC_ERR_QUEUE_FULL	The request queue cannot accept another entry.
6	COPROC_ERR_NO_WORKER	The selected worker is not running.
7	COPROC_ERR_STALE_TICKET	The ticket no longer names a live request.

## I.8 Raised by the CPU itself

Per CPU, the chapter (Ch 25-30) lists the trap and exception vectors and their meanings. The monitor's `r` command displays the current trap source when a CPU has stopped at one. Common cross-CPU shapes:

- Division by zero raises the CPU's divide-by-zero exception on M68K (vector 5) and x86 (INT 0). IE32 stops with a division-by-zero error. IE64 integer divide and modulo by zero write 0 and do not trap; IE64 floating-point divide-by-zero is reported in the FPU status register.
- An unaligned 32-bit access on the IE64 raises an alignment fault; the address that caused it is in CR\_FAULT\_ADDR.
- An undefined opcode raises the CPU's illegal-instruction vector (M68K vector 4, x86 INT 6, Z80 silently re-executes on most undocumented prefixes, 6502 documents the undocumented opcodes - see Chapter 27).
- On IE64, MTCR to the read-only CR\_RAM\_SIZE\_BYTES control register raises FAULT\_ILLEGAL\_INSTRUCTION (cause 11).