

What follows is a somewhat technical explanation of what common system error codes are really describing.

A Macintosh usually crashes with a numeric system error code while running under System 6.0.x. System 7 changed the way errors are displayed – instead of displaying an error code, it translates the code to the appropriate words, like "Address Error".

#### ID=01 Bus Error

This means the computer tried to access memory that doesn't exist. You can get this error on almost any Macintosh. If the computer tried to access one or more bytes beyond the total number of bytes in RAM, you see a bus error. You should never see this error on a Macintosh Plus or SE, because address references that are out of bounds "roll over". This means if one of these computers tries to access one byte beyond the total bytes in RAM, it actually accesses the first byte in memory. Essentially going in a big loop. If you see this error on a Macintosh Plus or SE, it's reporting the wrong error or having hardware problems.

#### ID=02 Address Error

The Motorola 68000 microprocessor can access memory in increments of one byte (8 bits), one word (16 bits), or one long word (32 bits). The microprocessor can access a byte of information at an odd or even memory address. But it must access a word or long word at an even memory address. So, when the microprocessor attempts to read or write a word or long word at an odd address, you see this error. Since that's a 50/50 proposition when running random code, this one shows up quite often.

#### ID=03 Illegal Instruction

The computer has a specific vocabulary of machine language instructions it can understand. If a computer tries to execute an instruction that isn't in its vocabulary, you see this error code. It's less likely than error 02, but still very common.

#### ID=04 Zero Divide Error

This error results if the microprocessor divides two numbers, and the divisor is zero. Sometimes a programmer puts these in as debugging aids, and then forgets to take them out.

#### ID=05 Range Check Error

Programmers can use an instruction in the Motorola 68000 to check if a number is within a certain range. This error indicates that the number tested isn't in the specified range.

#### ID=06 Overflow Error

Each number stored in a computer is given a certain amount of space. The larger the number, the more space is needed to represent the number. An overflow condition results if

a generated number is too big for its allotted space. A Motorola 68000 instruction tests for an overflow condition, and displays this error if it detects an overflow.

#### [ID=07 Privilege Violation](#)

The Motorola 68000 runs in Supervisor or User mode. The Macintosh should always be in Supervisor mode, but sometimes is placed in User mode. Some of the instructions can only be executed in Supervisor mode. If the computer attempts one of these instructions while in User mode, a Privilege Violation error results.

#### [ID=08 Trace Mode Error](#)

A programmer can use a runtime debugger while in Trace mode. This allows tracing through a program one instruction at a time. You see this error if a debugger isn't installed and the 68000 is accidentally placed in Trace mode.

#### [ID=09 Line 1010 Trap](#)

#### [ID=10 Line 1111 Trap](#)

There are many routines in the Macintosh ROM that can be called by placing instructions in a program that aren't in the 68000's vocabulary. When the 68000 encounters such an instruction, it looks it up in the instruction table. This table gives the location of routines paired with each instruction. If it finds an entry in the table for the instruction, it branches to the routine. If there's no entry for the instruction, you see one of these errors.

#### [ID=12 Unimplemented Core Routine](#)

A programmer might set breakpoints in parts of a program to inspect for errors. This requires using a debugger. If a debugger isn't installed when a breakpoint occurs, you see this error code.

#### [ID=13 Uninstalled Interrupt](#)

The Macintosh uses an interrupt to identify when devices like keyboards and disk drives need service. Routines must be available in memory to tell the computer how to service the device. If those routines aren't available, you see this error.

#### [ID=15 Segment Loader Error](#)

Macintosh programs are broken up into segments, and each program will always have at least one segment. Multiple segments allow loading parts of the program into memory to provide more room for data in internal RAM. The segment loader is responsible for loading a needed segment into RAM. If the segment loader can't do this, you'll see this error.

#### [ID=17 through ID=24 Missing Packages 0-7](#)

The Macintosh uses packages to do specific tasks. Some of the packages are International Utilities, Binary-Decimal Conversion, Standard File Utilities, and Disk Initialization. These packages are located in the System file. If you get these errors, you probably have a damaged System file. [Error codes 15, 16, 26, 27, 30, and 31](#) also come up when the System file is damaged. Try replacing the System file.

### ID=25 Memory Full Error

You've probably run out of RAM. But you can get this error when an earlier error causes the Macintosh to falsely detect an out-of-memory condition.

### ID=26 Bad Program Launch

The Macintosh couldn't execute the application opened.

### ID=28 Stack Ran into Heap

This is similar to the Memory Full error. It's a good idea to save your work frequently, and keep current backups of your hard disk data. When a system crash does occur, you'll lose less data if you've taken these precautions.