Error messages

The following are the error messages printed by **csd**. All **Message**s are displayed in the history window. All **Fatal error** messages are displayed in the history window. Fatal errors force the history window to be displayed, and ask that you exit from **csd**.

usage: CSD -G -D -O[L][M] -Hhelppath -Ssourcepath -T file[.exe] args

Fatal error. You have specified improper arguments to **csd**. This message tells you the form of the **csd** command. If the source file is to be found other than on the current disk or directory, you need to specify the **-S** option to tell **csd** where to find it. If the **csd** disk is not in the current disk drive or directory, you need to tell **csd** where it is with the **-H** option so that it can find the help files. The **-G** option is used when running **csd** on a color system which is not in 80 x 25 character mode. The **file** is the name of the program that you are debugging. Finally, **[args]** is the list of arguments to your program. For a complete listing of the compile command arguments and their uses, see section 6, **csd** *Command reference.*

Reading tables...

Message. This means that the tables containing debugging information are being read in.

Loading *filename...*

Message. *filename* is the name of the program being debugged. This message is printed the first time the program is loaded.

Reloading *filename...*

Message. This message is printed when the program is reloaded after <Run><End>, <Run><Begin>, a machine fault or after the program exits under its own control.

Adding source: *sourcefile*

After the "Reading tables..." message, you will see one of these for each source file found in the tables.

file **.exe** : cannot open

Fatal error. The file that you have specified in the **csd** command cannot be found.

insufficient memory for csd to run

Fatal error. Not enough memory is available for **csd** to load the symbol table and allocate the number of kilobytes specified with the **-D** option. Reduce the number of bytes specified.

insufficient memory to load program

Fatal error. Not enough memory is left after loading the symbol table and the sources to run the program. Reduce the number of bytes specified with the **-D** option.

this source file is more recent than the program

Message. This message reports that the program probably needs to be recompiled to bring it up to date with the source.

blocks nested too deeply

Fatal error. The program has nested compound statement blocks deeper than **csd** can keep track of.

valid breakpoint traps are 0, 3,...12, and 15

Fatal error. This reports that the -Itrap# option has an invalid value. The default value is 12.

csd stack overflow

Message. This message appears when **csd** overflows its own stack.

junk in high byte of program counter: *addr*

Message. This reports that the traced program has a program counter with non-zero value in the high 8 bits. These bits are not used by the 68000 processor, but it's unusual for them to be set.

no room in **vblqueue** for break key handler

Message. This means that the break key will not be active. This should not appear in most circumstances because there are seven free slots available.

Can't read *file*.**prg** read *file*.**prg**'>=29

Fatal error. **csd** has encountered a read error trying to read your program.

No source!

Fatal error. No source files were compiled with the **-VCSD** option. You must use this option when compiling programs for debugging if you want to use them with **csd**.

out of memory

Fatal error. There is not enough room to hold the program's debug tables. Try using the **-D** option to increase the size of **csd**'s workspace. Otherwise, debug smaller portions of your program.

execute failed: *filename*

This means that the program cannot be executed.

sourcefile.**c**: cannot open

One of the source files used to compile your program cannot be found. The names of the source files are kept in the executable **.exe** file, so if you have moved them, **csd** cannot find them. Use the **-T** or **-S** command line option to tell **csd** where to find them.

Cannot Open

This means that the file name specified as the program to be debugged cannot be opened to read the debug tables.

file has a corrupted image, should be recompiled

Fatal error. In order to start tracing your program, **csd** must write a breakpoint instruction into the program file. When tracing begins, the program file is rewritten to its original value. This message means that something went wrong, and **csd** left the breakpoint in the program file. The program cannot be traced as it is, and you will have to relink from object files or recompile.

try Help

You have pressed a key that is not acceptable at this time. For example, if you are at the beginning of the source file and type **<↑>**, you will get this message.

at outermost frame

In exploring the stack, you have typed the **<←>** key, but your caller, if it exists, is not visible to **csd**.

at innermost frame

In exploring the stack, you have pressed the **<→>** key, but you are already positioned at the innermost frame; that is, you have undone all of the **<←>**'s.

not executable statement

You are trying to set a trace on a statement that is not executable. Declarations, comments, braces '{ }', and '#' statements are not executable. In addition, code for some statements is combined during optimization, so that statements such as **break**, **continue** and **do** may not have any executable code.

not traceable expression
The expression you are trying to trace cannot be traced because it has no value. This is true of functions that are of **void** type.

out of space
Remove some of the expressions, or delete some of the history window.

helpfile.**hlp**: cannot open
csd cannot find *helpfile*.**hlp** in the current path(s). You can use the **-H** option to correct this.

more space required
There is not enough room to add expressions to the evaluation window. Remove some expressions from the evaluation window and try again.

Can’t move boundary any further move boundary any further’>=29
You are trying to move the cursor so it is positioned before the beginning or after the end of the expression, history, source or evaluation windows. **csd** requires that the source window be at least three lines long, and that the evaluation window be at least two lines long.

exit status *nnn*
Message. this message reports the value returned by the program when it terminated.

Evaluation window error messages
The following messages refer to expressions in the evaluation window. They describe errors in entering expressions, and are self explanatory:

```
integer type required
integer or pointer type required
numeric type required
numeric or pointer type required
expression required
literal or variable in current scope required
struct or union member required
lvalue required
pointer required
lvalue or type required
terminal or prefix op required
function required
array or pointer required
pointer to struct or union required
struct or union required
infix operator required
matching '?' and ':' required
pointer to same type required
matching types required
')' required
']' required
```

Machine fault names

The following lists named machine faults.

bus error
Access to an address which does not exist, or which the hardware forbids in user mode.

Address error
Access to an odd address for a long or word operand.

Illegal instruction
Execution of an instruction which is not supported by the processor.

Divide by zero
The divide instruction was given a zero as a divisor.

Bounds trap
The 68000 chk instruction trapped.

Overflow trap
The 68000 trapv instruction trapped.