Help screens

If you have a question about how the debugger operates, press **<F6>**. This will let you invoke any of the following help screens to refresh your memory about how a given **csd** command works.

Note that the windows may appear slightly differently on your screen due to differences in formatting.

General help

Program <F7>
Enter program window. This is the window that is usually shown while the program is running.

Source <F8>
Enter source window, where the program source is shown.

Eval <F9>
Enter evaluation window, where C expressions are evaluated.

History <F10>
Enter the history window, where a log of traced statements and expressions is shown.

Help <F6>
Display helpful information, or return to debugger if already in the help window.

Exit <Shift-F1>
Exit the debugger.

The keys below are not accepted in the program window.

Beg<Home>
Go to beginning of display; in the source window, this is the first line of the first source file.

End <End>
Go to the end of the display; in the source window, this is the last line of the last source file.

Out <←>
In the source window, go to the statement from which the current function was called.

In <→>
Go back to the statement from which you went **Out**.

Move about with **<↑> <↓>** (up, down arrows).

For more on the following commands, press the corresponding key.

Trace <F3>	**Run <F4>**	**Eval <F9>**	**Find <F1>**
Insert <Ins>	**Delete <Del>**	**Select <F2>**	

Trace

In the **source window**, the current statement is marked traced. The statement is shown highlighted, and whenever it is executed it is first copied to the history log. If the statement is already traced, the trace is removed.

Only executable statements may be traced. This excludes all comments and declarations. Furthermore, the optimizer in the compiler deletes unreachable and duplicated code (usually break, continue or return statements); deleted code is not executable.

In the **evaluation window**, the current expression is traced. This means that before each source statement is executed, the expression is re-evaluated and is copied to the history log whenever it is found to have changed value. If the expression is already traced, the trace is removed.

This command is not accepted in any other window.

The program may also be run until a trace is encountered: see **Run <F4>**.

Press **Help <F6>** to return to the debugger.

Execution

Run <F4>

Execute program.

The next key must be one of the following:

Trace <F3>

Program executes until a traced statement is encountered or a traced expression changes value.

Return <return>

One statement is executed (i.e., program is single-stepped).

Down <↓>

One statement in the current function is executed. A function call is treated as an indivisible statement.

Beg <Home>

Program is reloaded and execution is reinitialized.

End <End>

Program executes through to the end, logging tracepoints in history window.

Out <←>

Program executes until the current function returns.

Cancel <F5>

Cancels the **<F4>** command.

The program executes at full speed if there are no traced expressions in the evaluation window (traced statements do not significantly impact the speed) and if you use **Run Trace** or **Run End**. Otherwise, it runs at single-step speed, which can be significantly slower.

Press **Help <F6>** to return to the debugger.

Evaluation window

Evaluating C Expressions

In the evaluation window, a C expression may be entered involving operators, literal strings, constants and variables in the current scope (the scope of the current line in the source window). An expression may **not** involve keywords (such as **if**, **for**) or labels or braces '{}'or semi-colons ';'. See your C reference manual for details.

Expressions are evaluated immediately, and whenever the program encounters a tracepoint. The resulting value is displayed in a format appropriate to the type. No value is displayed if the expression involves an automatic variable and the current function does not have an active stack frame. The debugger defines types **(oct)**, **(hex)** and **(str)** for use as casts to display values in base 8, base 16, or as character strings, respectively. Macro names (generated by **#define**) are not recognized.

Expressions with side effects (assignments, increments, decrements, functions performing I/O) will naturally cause these effects whenever evaluated; they should probably be removed after being evaluated the first time. Literal character strings may be used as arguments to functions (including **strcmp** and **strcpy**), but cannot be assigned to variables, since the space for any literal in a C program is allocated statically.

Press **Help <F6>** to return to the debugger.

Find

Find <F1>

Find the first line in the current window matching a pattern, which may be the Trace key or metacharacters.

Trace <F3>

The **Trace** key causes the pattern to be highlighted and matches traced statements or expressions.

```
^      matches the beginning of a line.
$      matches the end of a line.
?      matches any character.
*      matches a string of zero or more of any characters.
[abc]  a class (in [ ]) matches any member of the class.
[i-n]  part of a class may be specified as a range.
```

If not given, the pattern defaults to the previous value.

The command is terminated by one of the following:

Up <↑>

Search up to the beginning of the current source file or top of the evaluation or history display.

Down <↓>

Search down to the end of the current source file or bottom of the evaluation or history display.

Beg <Home>

Search up to the beginning of the first source file or top of the evaluation or history display.

End <End>

Search down to the end of the last source file or bottom of the evaluation or history window.

Cancel <F5>

Cancel the **<F1>** command.

If the pattern is not found, this failure is reported.

Press **Help <F6>** to return to the debugger.

Insert/Delete

Insert<Ins>

Insert a blank line above the current line and make it current. This is only permitted in the evaluation window.

Delete <Del>

Delete a range of lines in the current window. This works in the evaluation and history windows only.

The command is terminated by one of the following:

Up <↑>

Delete the current line and move up one line.

Down <↓>

Delete the current line and move down one line.

Beg <Home>

Delete from the current line to the beginning of the display.

End <End>

Delete from the current line to the end of the display.

Cancel <F5>

Cancel the **<Del>** command.

If you get an "out of space" message in the evaluation window, it may be necessary to remove part of the display to free space for new expressions. If the debugger is rereading the disk for each new screen of source, removing several lines of history or evaluation may free enough space for more source buffers, eliminating the extra disk activity.

Press **Help <F6>** to return to the debugger.

Select

Select <F2>

Select various auxiliary options.

The command is terminated by one of the following:

Up <↑>

Move the boundary between the source and evaluation windows up one line.

Down <↓>

Move the boundary between the source and evaluation windows down one line.

L

Toggle the listing switch (initially off). When on, this copies the history log to the printer.

Program <F7>

Select the program window to be displayed when the program is running. When initially invoked, the debugger selects the program window by default.

Source <F8>

Select the program window to be displayed when the program is running. Any program output will be displayed in the source window.

Eval <F9>

Select the evaluation window to be displayed when the program is running. Any program output will be displayed in the evaluation window.

History <F10>

Select the history window to be displayed when the program is running. Any program output will be displayed in the history window.

Cancel <F5>

Cancel the **<F2>** command.

Press **Help <F6>** to return to the debugger.

History

History log

The history window stores messages about the program being debugged.

These messages begin with csd's sign on banner, the list of source files found in the debug tables, and an announcement that csd has loaded the program for execution.

If the program causes a machine exception, the details are reported. If the program terminates normally, the exit status is reported. If you **Run** the program after a normal or abnormal termination, the reload for execution is reported. If you **Run** the program to the **End**, **Out**, or **Down**, traced statements and changes in the values of traced expressions are logged to the history window as they occur. If csd encounters an error during the session, the details are reported.

You can use the cursor movement and **Find** commands to examine the history log, and you can **Delete** lines.

csd automatically discards history, from the beginning, when other commands for memory begin to fail.

Press **Help** to return to the debugger.

Cursor movement

FUNCTION	**KEY**	**ACTION**
Begin	<Home>	Go to beginning of display; in the source window this is the first line of the first source file.
End	<End>	Go to end of the display; in the source window this is the last line of the last source file.
Fbegin	<ctrl-A>	Go to beginning of current source file.
Fend	<ctrl-E>	Go to end of the current source file.
Up	<↑>	Move the cursor one line up.
Down	<↓>	Move the cursor one line down.
Previous	<ctrl-↑>	Move the cursor one page up.
Next	<ctrl-↓>	Move the cursor one page down.
Out	<←>	In the source window go to the statement from which current function was called.
In	<→>	Go back to the statement from which you went **Out**.
Locate	<Home>	Go to the source statement at which execution is stopped.

Cursor movement is possible in the source, evaluation, and history windows.

Press **Help** to return to the debugger.