

How to Demo Hamilton C shell

To set up the Hamilton C shell demo, first install it just as described in the manual on pages 11 and 12, including setting up the environmental variables (HOME, PATH, COLORS, TABS, ADDITIONS, etc.) in the control panel as shown in the manual.

A good demonstration might go as follows. What you type is shown in Courier Bold, machine responses are in Courier and my comments are in italics.

```
1 C% cd \hamilton Go to the Hamilton C shell directory
2 C% ls -l List the contents of the directory. Notice how subdirectories are shown
           highlighted. The C shell makes extensive use of color and all colors are
           completely user-customizable.
D---- Apr 14  5:00      -   bin
D---- Apr 14  5:00      -   samples
----- Apr 14  5:00    1712 login.csh
----- Apr 14  5:00    5215 readme
---A- May 20  2:00   31410 readme.too
----- Apr 14  5:00    4369 startup.csh
3 C% ls bin List the bin directory, showing all the utilities. Notice that all the usual
           favorites such as grep, fgrep, head, tail, etc., are included.
binedit.exe    dskread.exe    hrm.exe        split.exe      uniq.exe
cat.exe        dskwrite.exe   ls.exe         strings.exe    ver.csh
chmod.exe      dt.exe         mkdir.exe      sum.exe        vl.exe
cp.exe         du.exe         mv.exe         tabs.exe       wc.exe
csh.exe        fgrep.exe      newer.exe      tail.exe       whereis.csh
cut.exe        grep.exe       older.exe      tar.exe        xd.exe
des.exe        head.exe       pwd.exe        tee.exe
diff.exe       hlabel.exe    rmdir.exe     touch.exe
dim.exe        hmore.exe     sed.exe        tr.exe
4 C% alias mi It has aliases -- here's an alias for the more filter, starting it up in interactive
           mode (so it'll first clear the screen and stay around even if it's less than a
           screenful.)
mi             more -i
5 C% grep -h | mi Everything always has online help with the -h option.
           Regular expression pattern search of text files,
           Release 2.1

Usage:  grep [-hclnqsv-] [-f ptrnfile] [ pattern ] [ file1 file2 ...]

grep uses special patterns called regular expressions to filter
what it reads from stdin or from any files you specify.
:
--- more --- (Press H for Help)  _ Notice we can scroll up and down with the arrow keys
(Do it).
           And the more filter itself has online help by pressing H. (Press H to show it,
           then press any key to exit the help screen, followed by Q or ESC to get out.)
6 C% time factor 12341234 The C shell is a powerful scripting language. Here's an
           example, factoring a large number. Notice that we can time any command to see how
           long it takes.

2
73
137
617
0:00:01.37
```

7 C% whereis factor *This is actually a C shell script. We can find it with the whereis command.*

c:\hamilton\samples\factor.csh

8 C% mi `!!` *(Note that this is typed with BACKQUOTES, located next to the numeric one.) Let's go browse it. Here, the !! (pronounced bang-bang) part means pick up the text of the previous command (the whereis factor). The backquotes mean run what's inside there and paste the output (the c:\hamilton\samples\whereis.csh result) back onto the command line. That's called command substitution, passing the filename as an argument to mi, which we saw was the alias for more. So what we'll do is browse that script.*

proc factor(n) *Notice that this C shell has procedures,*

 if (n > 3) then

 for i = 2 to floor(sqrt(n)) do *a genuine numeric for loop, built-in functions like floor and square root,*

 if (n % i == 0) then

 echo \$i

 return factor(n//i) *and procedures can even be recursive. Not surprisingly,*

a lot off customers are using Hamilton C shell with thousands of lines of scripts. For example, Sybase uses it to run the build for SQL Server, Microsoft's languages group uses it to run nightly regression tests, etc.

 end

 end

 end

 return n

end

:

9 C% echo \<Ctrl-D> *Finally, the C shell has a lot of "creature comforts" such as filename completion and command line editing. Here, e.g., we'll expand in-place all the names of the files in the root. So for folks coming from a UNIX background -- or anyone looking simply for a more powerful scripting or development environment, Hamilton C shell can be very attractive, offering instant productivity gains.*