

C++ (Release E) Installation Guide - November 1984

To make and install a C++ system you should

- (1) make and install a cfront (C++ compiler front-end)
- (2) install a CC command
- (3) install the C++ standard header files
- (4) install the C++ standard library
- (5) install the manual page
- (6) test the system

(0) First check that your cc can be used by CC (UNIX System V release 2 or higher and BSD4.1 or higher are OK):

- (1) You need long names and structure assignment. The following should compile.:

```
struct s { int a12345678, a12345679; };
main() { struct s a,b; a = b; }
```
- (2) Your stdio.h should be compatible with the stdio.h used to produce the ..c files in directory C++/c (compare the declarations of struct _jobuf, and the gets puts macros). If they are not you must convert the ..c files.

(1) If you have a CC installed already in directory C++:

```
CC -O *.c +E y.tab.c -o cfront
```

otherwise in directory C++/c:

```
cc -O *.c -o cfront
```

If you are in a hurry or you do not trust your optimizer leave the -O option out. The compilation may give a few warnings. Install cfront as /usr/bin/cfront

(2) If your cc understands .i (intermediate) files install CC as /usr/bin/CC otherwise install CCc as /usr/bin/CC.

(3) Install incl as /usr/include/CC. When time permits, look over these files to check that they do not contain declarations unsuited for your system.

(4) In directory libC:

```
sh < compile
ranlib xx
mv xx /lib/libC.a
```

If you do not have a "ranlib" you don't need it. The compilation may give a few warnings.

(5) Install the manual page CC.1 (actually two pages).

(6) Use the new CC to compile the C++ source for cfront as a test.

Finally, but most importantly, when you find bugs, confusing error messages, or any inconvenience you feel could have been avoided by a more suitable compiler action or by a language feature, please send a note about it to {ihnp4! or decvax! or research!} btlunix!c++; that way there is a chance others will not suffer in the same way.

Good luck.

P.S. C++/size_align is an example of the format of a "sizes and alignment" file to use with the +x cross compiler option.