

PrettyC v2.0

The C and C++ source code printing utility  
Jeremy Roussak  
Madingley, 15 Woodhill Park  
Pembury, Kent TN2 4NW  
England  
Email: [JeremyR@dcs.qmw.ac.uk](mailto:JeremyR@dcs.qmw.ac.uk)

PrettyC is a utility to print C and C++ source code and to make it look nice. It accepts as input multiple files in multiple folders and will print their contents, highlighting keywords and trying to ensure that the functions in them are not split across pages. It will recognise typedefs (and, for C++, structs, enums, classes, and unions) and will add the keywords defined by them to its list of words to highlight. In general, v1.5 provides proper handling of the object-oriented extensions to C provided by THINK C versions 4 and 5. PrettyC will find and read #included header files, noting new keywords defined by the typedefs they contain. It will optionally print in two columns and at the end of each run can print an index listing the functions printed (sorted by name or by class), their page numbers, the files they were in and their types: if you want, you can print only the index (this is useful for locating functions in multi-file projects). A prescan option allows you to select just those files and functions you wish to be printed. PrettyC will run happily, although more slowly, in the background (and with great consideration for the foreground process).

PrettyC v2.0 requires system 7: it is AppleEvent-aware but handles the required class only. It is not recordable and only minimally scriptable. TEXT files, project files created by THINK C and CodeWarrior and PrettyC's own saved sets may be dropped onto PrettyC's icon in the Finder.

New in version 2

- requires system 7
- handles C++ properly
- takes THINK C and CodeWarrior projects as input and prints their source files
- snazzy-looking dialogs (some say)
- many internal changes and not a few bug fixes

Files

PrettyC: the program

PrettyC dox: this file

PrettyC Prefs: created in your system folder when required, to save default settings

A PrettyC set

A PrettyC “load and go” set

Menus

PrettyC has three menus, of which one, Edit, contains nothing of much interest: just cut, copy, paste and undo, for desk accessories.

The File Menu

The File menu contains eight options. Page setup... presents the standard page setup dialog for whichever printer is currently installed by the Chooser. Quit quits, surprisingly enough.  
Select files...

Files available to be printed are shown in the left hand list, which can be manipulated in the usual Standard File way. These files are of three types: plain TEXT files, containing source code; THINK C project files; and CodeWarrior project files. For details on the latter two, see below. The Eject, Desktop and Cancel buttons work as usual.

To choose a file to be included in the print list, either double-click its name or select it and click Include. The file will then appear in the print list on the right and will disappear from the list on the left. To select all files ending in .c, click on the All .c button. To select all files ending in .cc, .cp or .cpp, click on the All .cc/.cp button. If you hold down the option key while you click on either button, all files shown will be selected.

To remove files from the print list, either double-click on their names in the right-hand list or select them (by shift- and command-clicking) and click on Exclude. To move a file in the print list (files are processed in the order they appear), option-click on it and drag it to the appropriate position. You can only move one file at a time like this.

When you have finished setting up the print list, you can click on either Done or Go (command-G is the same as Go). Done saves the print list and PrettyC will then wait for further menu selections; Go starts the print run immediately.

Load set

Save set

These options allow you to save and restore all the current settings. See the section on sets below.

Go!!

Selecting this option is exactly equivalent to clicking on the Go button in the choose files dialog: it starts the print run. This option is disabled if no files are selected.

PreScan

This starts a prescan, described in detail later.

Hide progress window

When PrettyC is printing, it displays a window telling you how it's getting on. This option hides that window, and is greyed out when PrettyC is not printing (when it is printing, this is the only menu option available). If you hide the window, the text of this item changes to Show progress window.

The Settings Menu

Preferences...

There's nothing here that merits much in the way of explanation, really.

Typefaces...

There are popup menus for the code, comment, page heading and function banner fonts and sizes (the code font is also used for the index). You can also enter sizes directly into the boxes. If any font is not available, the name will be greyed out. You can also select the style to be used for comments, keywords, headings and function banners. If you set plain text for keywords (not bold or italic), PrettyC will not bother to parse #include directives, nor will it look for new, user-defined keywords as it prints.

Clicking OK saves changes, Cancel discards them and Make default saves them in the PrettyC Prefs file for later use.

Options...

If Smart quotes is selected, PrettyC will use matching opening and closing single and double quotes for character and string constants. Occasionally it is useful to use a monospaced font for these constants, so Print quoted strings in Courier lets you do that. (This option is disabled if a font called Courier can't be found).

Full pathnames are used in the headers (but not in the index). Show "last modified" dates similarly affects the page headers.

If Ignore formfeeds in source is not selected, a formfeed (ASCII FF) will cause a new column to be started (a new page, in single-column mode). Truncate long lines means that lines too long for the column will be truncated just before the word which would have overflowed, rather than being split with the remainder printed at the left margin, preceded by an ellipsis. I don't like this option, but it was requested by a few people.

If you check the Don't parse system header files option, PrettyC won't try to read any files which are #included in angle brackets. This can save a lot of time if you already have Mac keywords defined (see Keywords).

Print function banners will print a shaded box with the function name in it before each function.

PrettyC automatically enables special parsing of C++ syntax if the filename ends in .cc, .cp or .cpp. If you have C++ in files with other suffixes, checking Enable C++ for all files would be a good idea.

The Two columns popup has three choices: always, never and sometimes. These are self-explanatory, apart from sometimes. If you have sometimes selected, PrettyC doesn't use a random number generator, but looks to see if your paper is wider than it is high (landscape mode). If it is, PrettyC uses two columns: if not, it only uses one.

Normally, PrettyC will start a new column if the next function to be printed will not fit in the current column (but would fit on the next). This can lead to phenomenal wastage of paper, so to preserve the rain forests, you can specify that all pages are to be at least as full as you wish. If you specify 100%, no pagination will occur. At the other extreme, the options new column and new page in the Function pagination popup allow you to waste vast quantities of paper very easily.

You can set the tab width to any reasonable number. If your code font is proportionally spaced (and most fonts are), PrettyC

uses the width of the letter x to calculate tab stops. Some programs allow you to set the tab width for a particular file and store it in a resource: if you check Use tab resource, PrettyC will override its current tab setting and use the resource, if it's there, for that file only. Smart tabs is an attempt to make neatly aligned comments stay neatly aligned if, like me, you use a fixed-width font for screen work and a proportionally-spaced font for neat printing. It's a little difficult to explain how it works (indeed, it doesn't work all the time): try it and see the difference!

The Index Print popup also has three choices: source only, index only and source and index (the default). If you pick index only, PrettyC will print only the index of functions, their types and the files they were in. All the other options in this dialog will be disabled (greyed out), except for the two-column and sort options. It doesn't make much sense to use this option with PreScan, but you can if you really want to.

Index Sort may be by function or by class. It is useful if you want to have all a class's methods appear together in the index. Clicking OK saves changes, Cancel discards them and Make default saves them in the PrettyC Prefs file.

Keywords...

PrettyC comes with three sets of keywords defined: ANSI, which includes just those in the ANSI definition, Mac (partial) and Mac (full), much larger sets comprising keywords from the Macintosh world. There are 44 keywords in the ANSI set, 852 in the Mac (partial) set and 1850 in the Mac (full) set. Mac (full) is a superset of Mac (partial) and both the Mac sets contain all the ANSI keywords. There is, obviously, a speed penalty when using the larger sets; however, PrettyC uses a balanced binary tree algorithm for keyword lookup, which is reasonably efficient. You are free to edit these sets as you wish. The Keywords... option in the Settings menu brings up the following dialog box:

Clicking on the ANSI or either of the Mac buttons throws away the current keyword list and reverts to the appropriate standard list. Remove deletes the selected keyword from the list (as does double-clicking on the word). Add new adds a new keyword to the list: if it's already there, PrettyC just beeps. Note that since C is a case-sensitive language, the list is sorted in a case-sensitive fashion, so the standard, lower-case C keywords appear at the end. The keyword list is a standard scrollable list, with the advantage that if you type a command-letter combination, it scrolls to the first keyword beginning with that letter (the shift key works here). Scan... scans a TEXT file and adds the keywords typedefed in it to the current keyword list. #include directives are always parsed during a scan.

Clicking OK saves changes, Cancel discards them and Make default saves them in the PrettyC Prefs file.

Margins...

You can specify your margins in inches, centimetres or points, either by using the Units radio buttons or by placing "in", "cm" or "pt" after the numbers. You can specify different margins for one and two-column printing. Left of gutter is the margin for the right-hand edge of the left column, and Right of gutter is the margin for the left-hand edge of the right column. The two Copy buttons copy the top four margins in the appropriate direction: command-leftarrow and command-rightarrow are keyboard shortcuts.

Clicking OK saves changes, Cancel discards them and Make default saves them in the PrettyC Prefs file.

Date & Time formats...

This option lets you choose the way the date and time will appear in the page headers. It affects both print date and "last modified" date, if printed. The line at the bottom shows you how the option you have chosen will look.

Clicking OK saves changes, Cancel discards them and Make default saves them in the PrettyC Prefs file.

System header folders...

When PrettyC is looking for a header file, it follows the same rules as THINK C and CodeWarrior. Files included with #include "filename" are looked for in the same folder as the file being printed and in all sub-folders it contains. Files included with #include <filename> are looked for in the folders specified by this option and in all sub-folders they contain. Only folders are shown. Simply highlight the appropriate folder and click on the Select button. Double-clicking, or clicking on Open, opens the folder as usual – you can then use the Select this folder: button. The chosen folders are always stored in the PrettyC Prefs file when you click OK, becoming the future defaults.

You can select as many folders as you like. They and their sub-folders are searched in the order in which they appear in the search list. To remove a folder from the list, double-click on it or click on it and click on the Remove button. To alter the order of folders in the search list, option-click on the name and drag to where you want it to be.

Project files

Version 2.0 of PrettyC can accept project files created by Symantec's THINK C (version 6 or later) and by MetroWerks' CodeWarrior. When you choose Go!! (or click the Go button in the file selection dialog), PrettyC will launch the appropriate creator and ask it for a list of source files from the project. The first time you do this, PrettyC will ask you to locate the creator application, but it will remember its location subsequently. You must have enough memory available for

the creating application to be loaded and AppleScript must be running.

## Running

When PrettyC is running, you'll see a modeless dialog box like this one, just to let you know how it's getting on.

The window can be closed by clicking its close box or by choosing Hide progress window from the File menu, and can be moved around (the new position will then be forgotten – maybe in the next version...). Clicking Stop (or typing Command-.) will stop the print run fairly quickly.

## PreScan

Sometimes, you just want to print part of a program. If you start a PrettyC run with PreScan rather than Go!!, PrettyC will read your selected files and then present you with a dialog box which allows you to choose just those files and functions you wish to print.

You can select from the lists by clicking and shift-clicking. If only one function is selected, its type and the file it is in are displayed at the bottom of the dialog.

If you select a file, or a group of files, all functions contained in those files are selected and all those not contained in them are deselected. If you hold down the option key when selecting files, functions already selected are not deselected, even if their file is no longer selected. Selecting files is simply a quicker way of selecting groups of functions: a file is scanned if any of its functions is chosen and is not scanned if none of its functions is chosen, regardless of whether its name is selected or not. The “preamble” of a file—the starting comment, initial preprocessor directives and declarations before the first function—is always printed if any function in that file is selected, as is anything apart from comments which is found in the inter-function gaps.

When the PreScan dialog is showing, typing a letter (without holding down the command key) makes PrettyC scroll the function list so the first function beginning with that letter is shown. The case of the letter is important here. If you just have one function selected, PrettyC will tell you its type and which file it's in in the space at the bottom of the dialog.

## Sets

You may have a particular group of files which you always want to print together, or particular sets of options you use for special occasions. To help you, PrettyC will let you save and restore settings in files called sets. A set contains the options, typefaces, margins and system header folder list in effect when it was saved, together with the list of files selected at the time.

Sets are of two types: normal and “load and go”. If you start PrettyC by double-clicking a normal set, it will read all the settings in the set and wait for further menu commands. If you double-click a “load and go” set, PrettyC will read it, print the files specified and quit without further intervention. There's no difference between the two types of set if they're loaded with the Load set... menu command or by dropping them onto PrettyC's icon when it's already running.

Choosing Save set... from the File menu brings up an ordinary “Save” dialog, with a “load & go” check box. The “load & go” check box is disabled if no files are selected when Save set... is chosen.

## Arguments

Arguments may be supplied to PrettyC from the MPW command line, by selecting files in the Finder and double-clicking or by dropping them on PrettyC's icon in the Finder. Suitable arguments are TEXT files containing source code, project files created by THINK C or by CodeWarrior (the creating application must be available) and PrettyC's own saved sets. PrettyC treats arguments of different types as follows:

A TEXT file or

a project file    Print it, using the default settings, and quit

A normal set    Load it and wait for menu commands

A “load & go” set    Load it, print its files using its settings, and quit

Many TEXT files    Print them all, using the default settings, and quit

Many sets    Load them all, merging their file lists. Apply the settings of the last set loaded. If any set was a “load & go” set, print the files and quit. Otherwise, wait for menu commands.

TEXT files and sets    Same as Many sets, but include the TEXT files in the list of files to be printed.

I'm not entirely convinced that this is the best solution to the problem. If anyone has any better ideas, I'd be delighted to hear them.

## Dialogs in general

PrettyC positions its dialogs (including the print dialogs) in the place approved by the Apple Human Interface Thought Police: on the monitor containing the mouse, centred horizontally and a third of the way down the screen. Some keys have the same effect in all PrettyC's dialogs:

Return activates the default button, as usual

Command-return or Enter    activates the button which closes the dialog and accepts the changes (may be the same as

Return)

Command-Enter      activates “Make Default”, if present. Otherwise, the same as Enter

Command-. or esc    activates the Cancel button

Clear    activates the Remove or Exclude button

Some dialogs have special key shortcuts: these are described with the dialog.

## Problems

There a couple of known deficiencies in PrettyC, which I may correct in the future.

First, it doesn’t parse #ifdef/#else/#endif lines, so if a function, as a result of having some of these, appears to contain more left braces than right braces, PrettyC will get confused and remain so for the rest of the file. To give an example,

```
void Wotan()
{
...
#ifdef Alberich
    {      Brunhilde = 4;
#else
    {      Siegfried = 8;
#endif
    ...
}
```

will not be handled correctly.

Second, it will occasionally crash under low memory conditions. Normally it handles running out of memory well, but if it has encountered a very large number of new keywords during a print run (if it has parsed many header files, for example), it can crash while freeing the keyword tree. The crash will occur at the end of the print run, just as the print dialogs disappear. The solution is to increase PrettyC’s memory allocation. One megabyte seems to suffice for even the most complex situations.

## Upgrading from version 1.x

The PrettyC prefs file which version 2 of PrettyC uses is not compatible with the PrettyC Options file used by version 1.x.

Sorry.

Sets which were saved using PrettyC 1.x can be read by version 2. The file lists will be read, but any saved settings will be ignored (and deleted). Sorry again.

Registered users of version 1.x (there aren’t very many of you, I regret to say) should contact me for a registration key for version 2.0 (see Registration, below). The upgrade is free.

## Copyright

PrettyC is copyright © Jeremy Roussak 1990–95, and I retain all rights to it. It was written using CodeWarrior v5.5, so parts may be copyright © MetroWerks Inc. PrettyC is released as shareware. It may be freely distributed, given to friends, placed on bulletin boards and eaten by the cat, but it may not be distributed as part of commercial software without my written permission and the program and documentation files must be kept together.

## Registration

PrettyC is shareware. It is distributed as a fully-functional version with no disabled features. If, after a reasonable trial period, you decide that you like it and want to keep it, you should send me a cheque, postal order, international money order or Eurocheque for £10 sterling, a check for \$US25 drawn on a US bank (yes, it’s a little more: it costs a lot to cash a dollar cheque in this country!) or the equivalent of \$US25 in your local currency, along with the registration form which you can print from within PrettyC (see the apple menu). I will then send you, by email if possible, a registration key, derived from your name, which you can enter into PrettyC’s registration dialog. If you haven’t registered after a week, PrettyC will start to remind you, gently, that you should. It won’t stop working or do anything nasty, though. After you enter your name and key, you’ll never see that irritating alert again.

If you want to register multiple copies of PrettyC, please contact me: I expect we can work out a reduced fee.

## Coming soon

The response, in terms of registrations, to PrettyC 1.5 was rather disappointing, which is why it has taken nearly four years for another version to appear. If more people register this time, I’ll produce a new version rather sooner than 1999! I might incorporate a full C beautifier (if I think I can stand the endless arguments about the proper way to lay out C source code: I certainly don’t use the One True Brace Style) and it might be nice to be able to produce formatted text, which could be imported into a word processor, as an option instead of sending output straight to a printer.

## Acknowledgements

Thanks are due to (in alphabetical order!) Alex Chaffee, Eric Clements, Andrew Flavell, Wally Wedel and Gijsbert Wiesenekker. Also to MetroWerks, Inc., for CodeWarrior and in particular for agreeing to distribute PrettyC on the CD-ROM for CodeWarrior 6. This gave me the spur to update PrettyC to handle C++, amongst other things.