PrettyC v1.5

The C source code printing utility

Jeremy Roussak
Whitecroft, 40 Cross Road
Bushey, Watford WD1 4DQ
England

Internet: jeremyr%dcs.qmw.ac.uk@nsfnet-relay
UUCP jeremyr@qmw-dcs.uucp

PrettyC is a utility to print C source code and to make it look nice. It accepts as input multiple files in multiple folders and will print their contents, highlighting C keywords and trying to ensure that the C functions in them are not split across pages. It will recognise **typedef**s and will add the keywords defined by them to its list of words to highlight: it will also recognise THINK C's class declarations and add the class name to its list of keywords. In general, v1.5 provides proper handling of THINK's object-oriented extensions to C. It does not attempt full handling of C++, although it copes with C++ syntax remarkably well. **PrettyC** will find and read **#include**d header files, noting new keywords defined by the **typedef**s they contain. It will optionally print in two columns and at the end of each run can print an index listing the functions printed, 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. If you are running MultiFinder, **PrettyC** will run happily, although more slowly, in the background (and with great consideration for the foreground process). **PrettyC** v1.5 will run under System 7 without crashing, but is not truly "System 7 friendly".

Files

PrettyC: the program

PrettyC dox: this file

PrettyC Options: created in your system folder the first time **PrettyC** is run, to store default settings

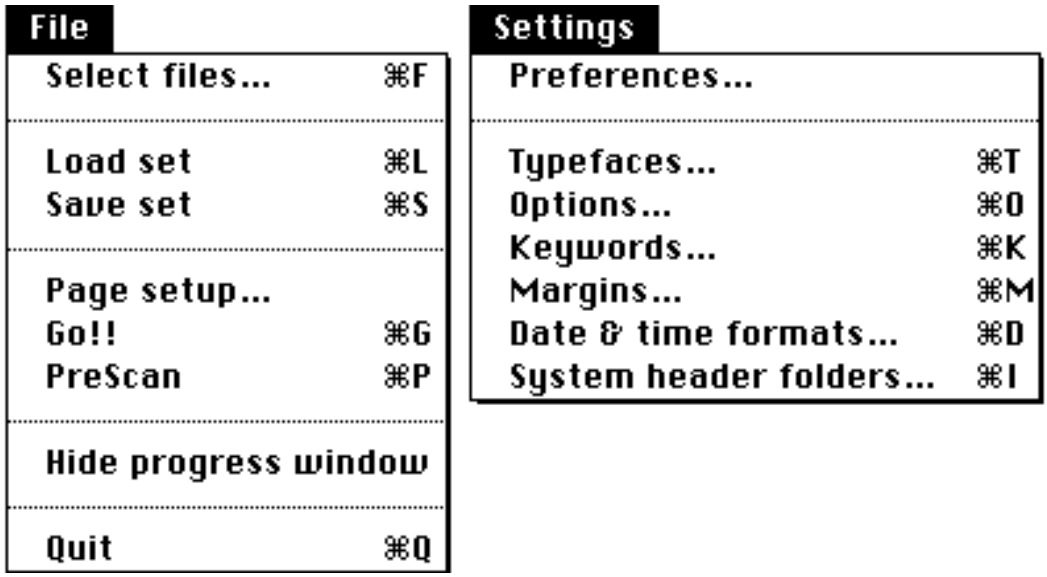
A **PrettyC** set

A **PrettyC** “load and go” set

(A note to upgraders: if you have been using PrettyC 1.04, you won’t see either of the set icons until you rebuild the desktop. Sorry about that.)

Menus

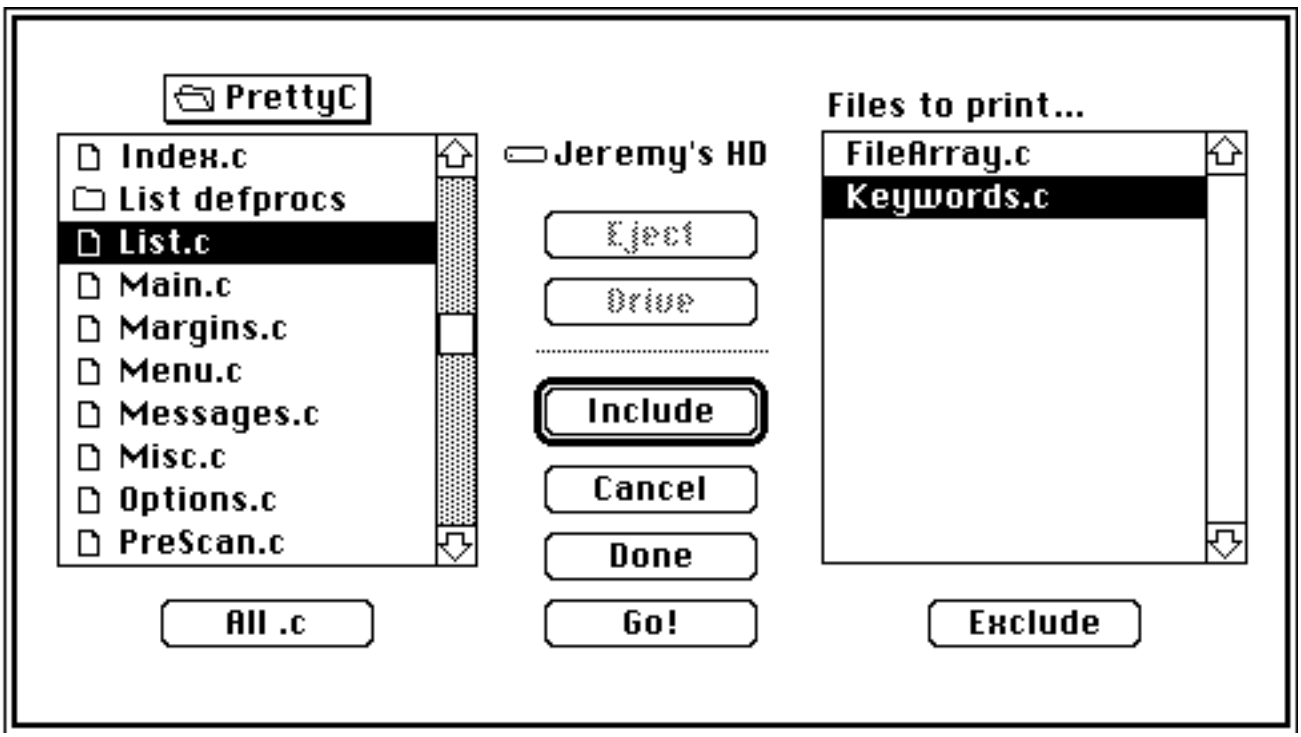
PrettyC has three menus, of which one, **Edit**, contains nothing of much interest: just cut, copy, paste and undo, for desk accessories.



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, suprisingly enough.

Select files…



Files available to be printed (of type ‘TEXT’) are shown in the left hand list, which can be manipulated in the usual Standard File way. The **Eject**, **Drive** and **Cancel**

buttons work as expected.

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** for printing, click on the **All .c** button or type command-A. If you hold down the option key while you do this, 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 (greyed out) if no files are selected.

PreScan

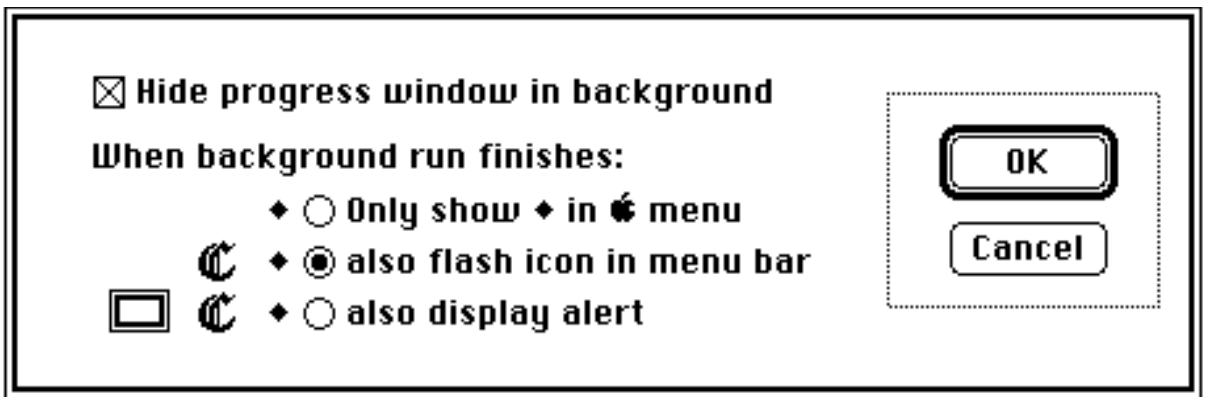
This starts a prescan, described in detail later.

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**.

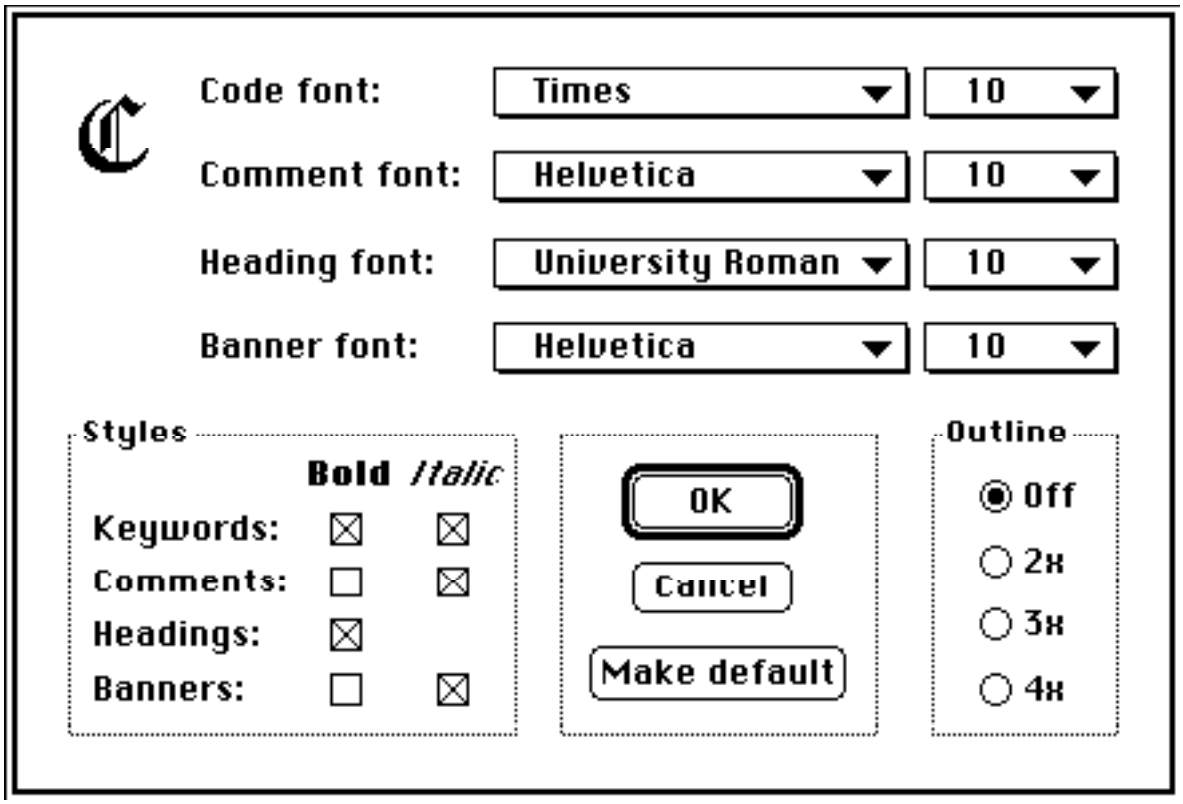
The Settings Menu

Preferences…



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

Typefaces…

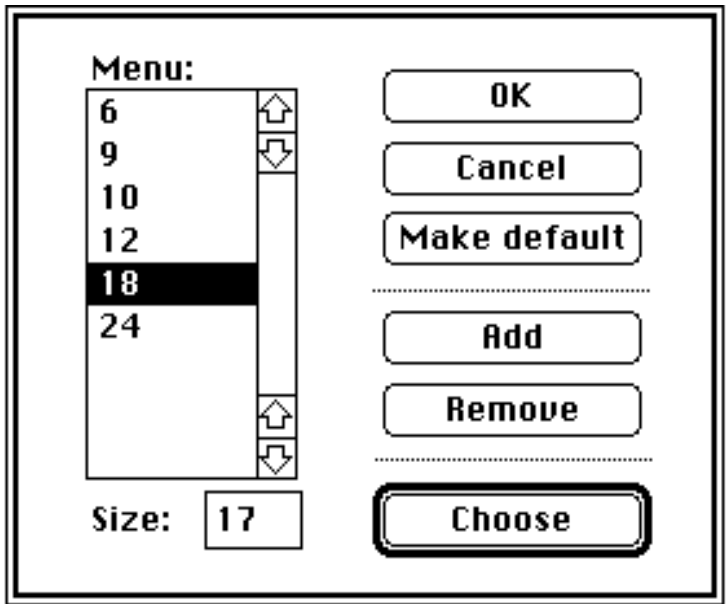


There are popup menus for the code, comment, page heading and function banner fonts and sizes (the code font is also used for the index). If any font is not available, the name will be greyed out. You can also select the style to be used for comments, keywords, headings and function banners. If you set plain text for keywords (not bold or italic), **PrettyC** will not bother to parse **#include** or **typedef** lines, since the only purpose of parsing these is to recognise new keywords to be printed in a particular style.

The **Outline** radio buttons control which sizes are outlined when the size menu is popped up. This is useful if you are using a printer such as an ImageWriter, Hewlett-Packard DeskWriter or LaserWriter SC, which requires large fonts to be present for top-quality printing.

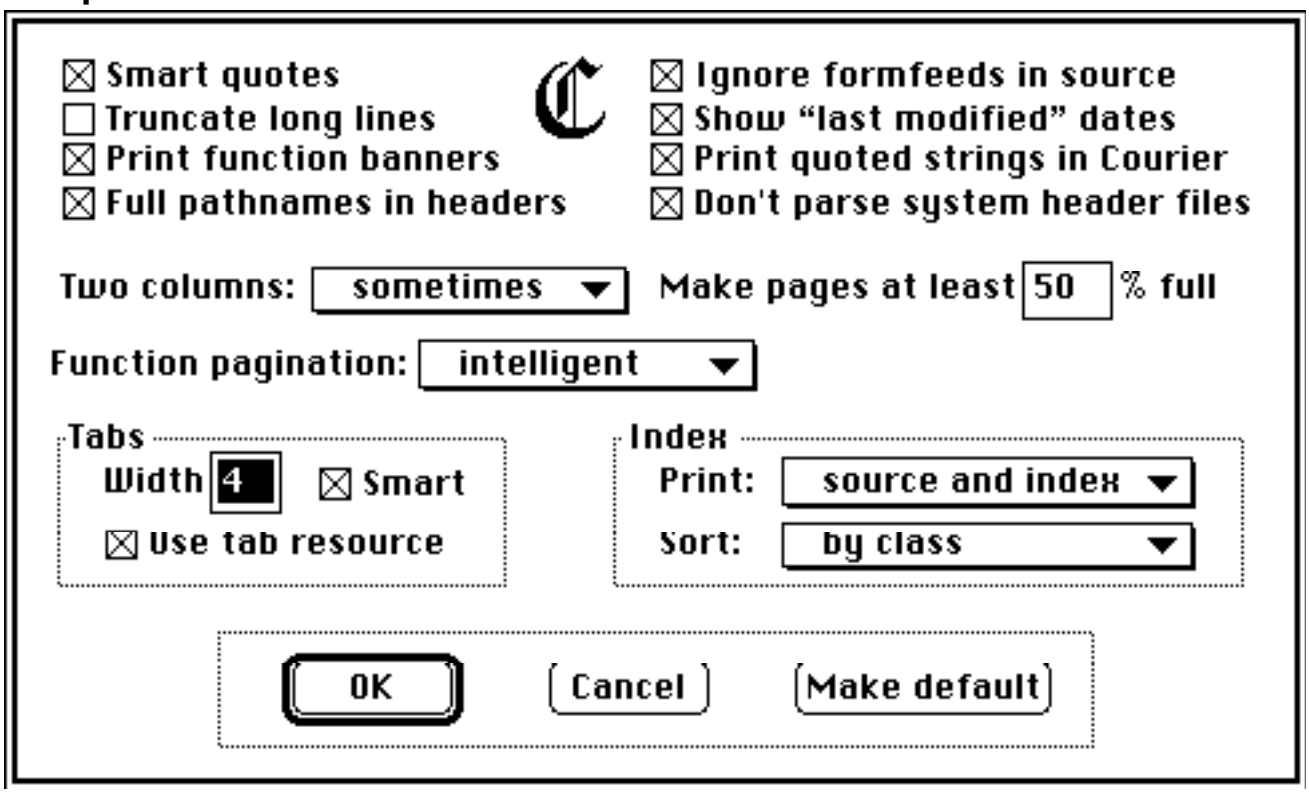
Clicking **OK** saves changes, **Cancel** discards them and **Make default** saves them in the **PrettyC Options** file.

The last option on the size popup is **Other…**. Selecting it produces this dialog:



You can use this font sizes dialog to configure the size menu permanently or for just one run of **PrettyC**. Any items added or removed using the **Add, Remove** or **Choose** buttons affect both the code and comment size popups, irrespective of which was used to activate the dialog. Changing the menus does not affect the size selected in the **Typefaces** dialog unless the **Choose** button is clicked.

Options…



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 (specifically, THINK C and MPW) 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: try it and see the difference!

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 **#include**d 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.

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.

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**, and only applies when printing files containing THINK C methods. It is useful if you want to have all a class's methods appear together in the index.

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 phenomal 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 specifiy 100%, no pagination will occur[1]. 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.

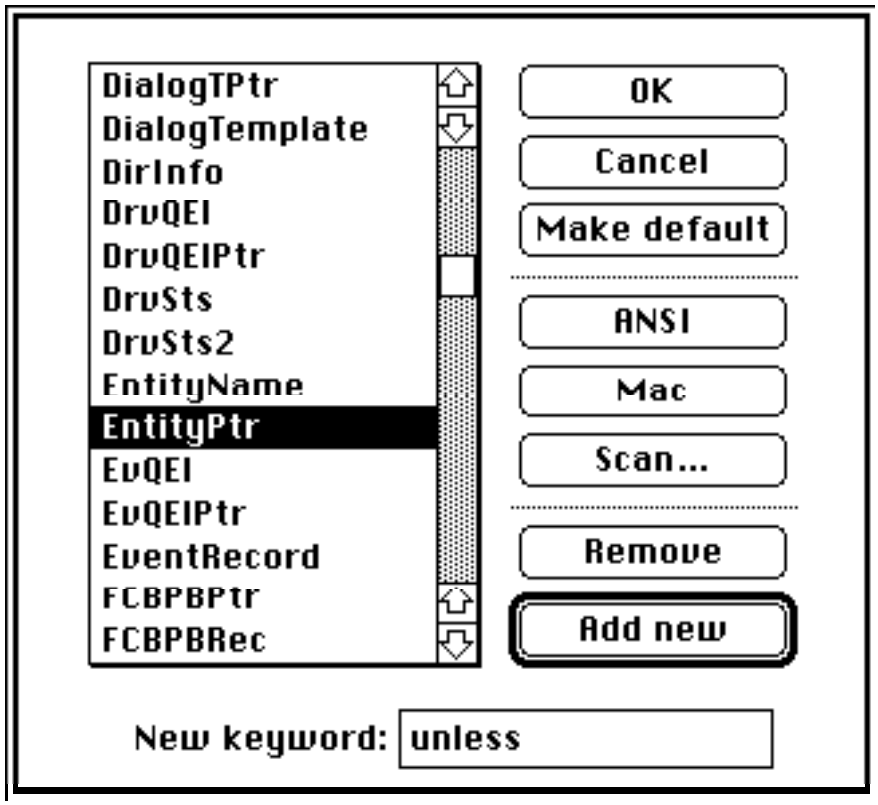
Clicking **OK** saves changes, **Cancel** discards them and **Make default** saves them in the **PrettyC Options** file.

[1]Or not much, anyway. When **PrettyC** nears the bottom of a page, it starts looking for a good place to start the next page. A "good place" is a blank line. Even if you want pages to be 100% full, **PrettyC** will still start a new page a few lines early if it thinks it should.

Keywords…

PrettyC comes with two sets of keywords defined: **ANSI**, which includes just those in the ANSI definition, and **Mac**, a much larger set comprising all the keywords from the Macintosh world[2]. However, you are free to edit these as you wish. The **Keywords…** option in the **Settings** menu brings up the following dialog box:

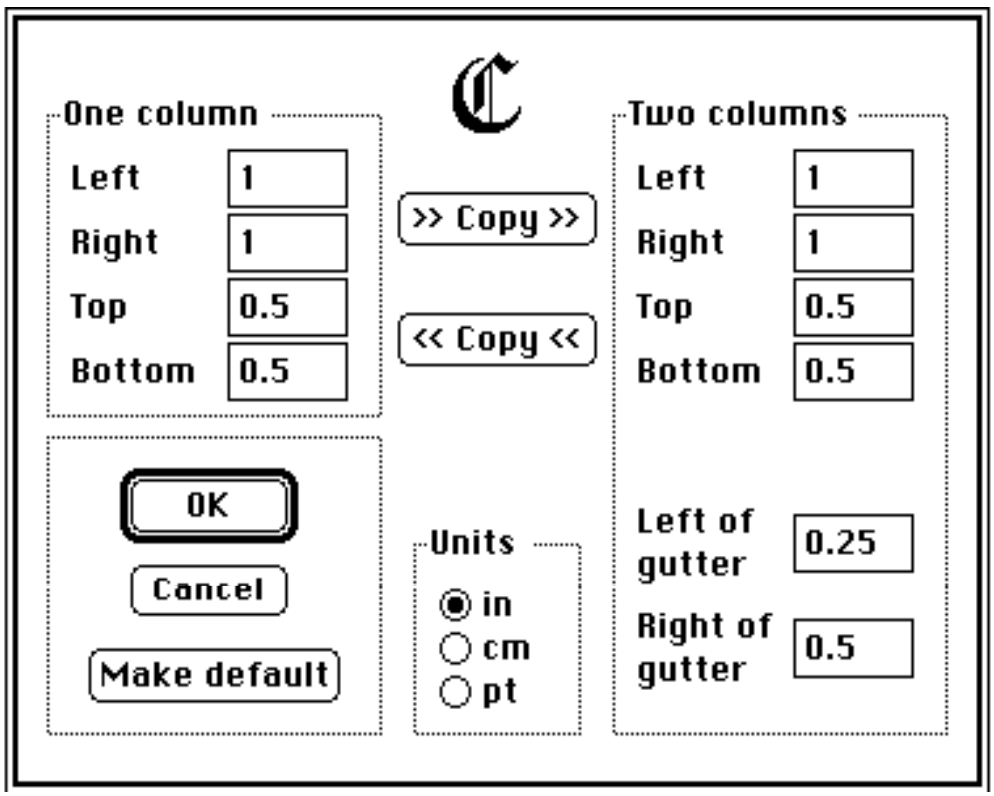
[2]To be precise, all the keywords **typedef**ed in all the header files supplied with THINK's Lightspeed C version 3.0. It doesn't include System 7 keywords, since I don't have the header files yet. Use the Scan button.



Clicking on the **ANSI** or **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 (usually a C header file) and adds the keywords **typedef**ed in it to the current keyword list.

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

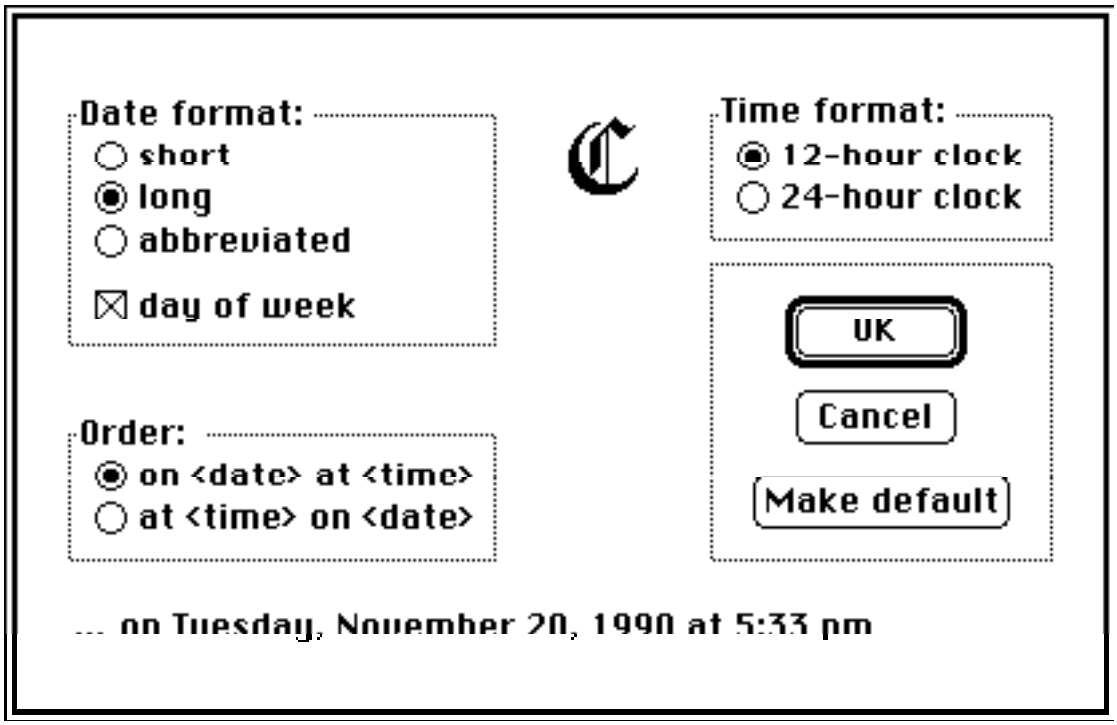
Margins…



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.

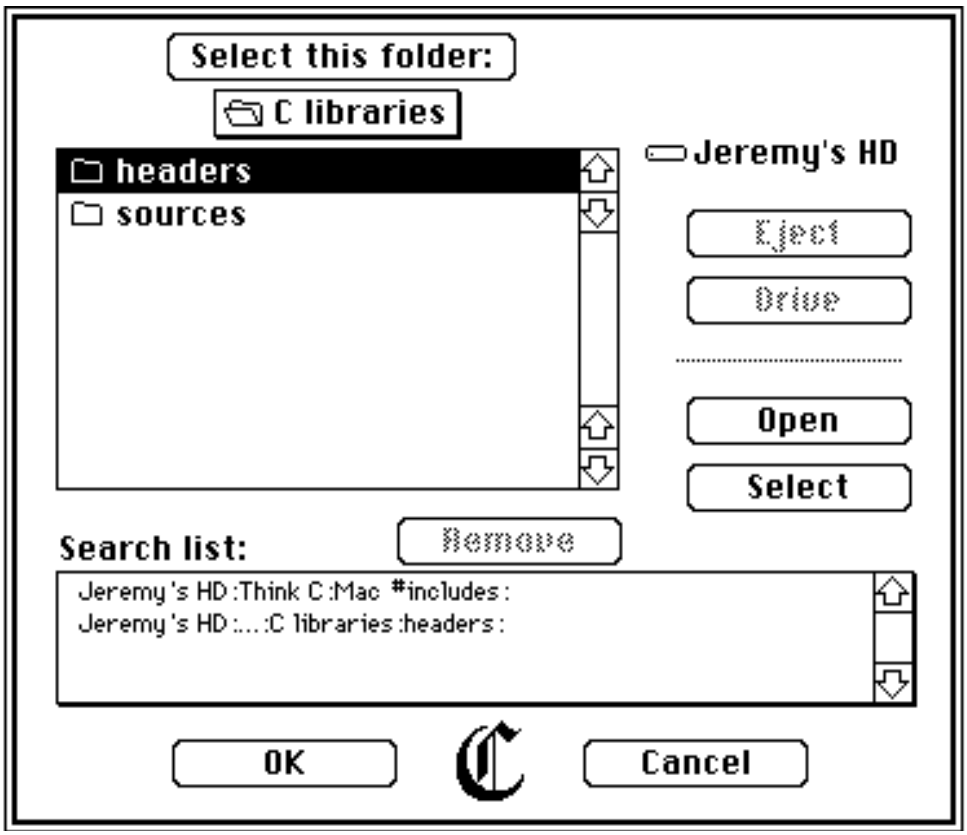
Clicking **OK** saves changes, **Cancel** discards them and **Make default** saves them in the **PrettyC Options** file.

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.

System header folders…



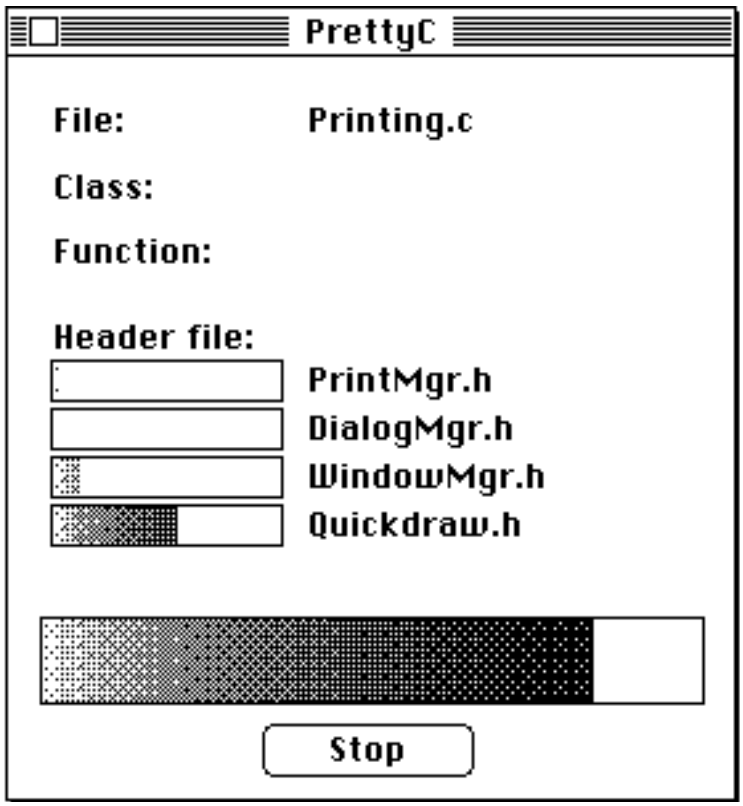
When **PrettyC** is looking for a header file, it follows the same rules as THINK C. Files included with **#include "filename"** are looked for in the same folder as the file being printed and in all sub-folders it contains. Files included with **#include <filename>** are looked for in the folders specified 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 Options** 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.

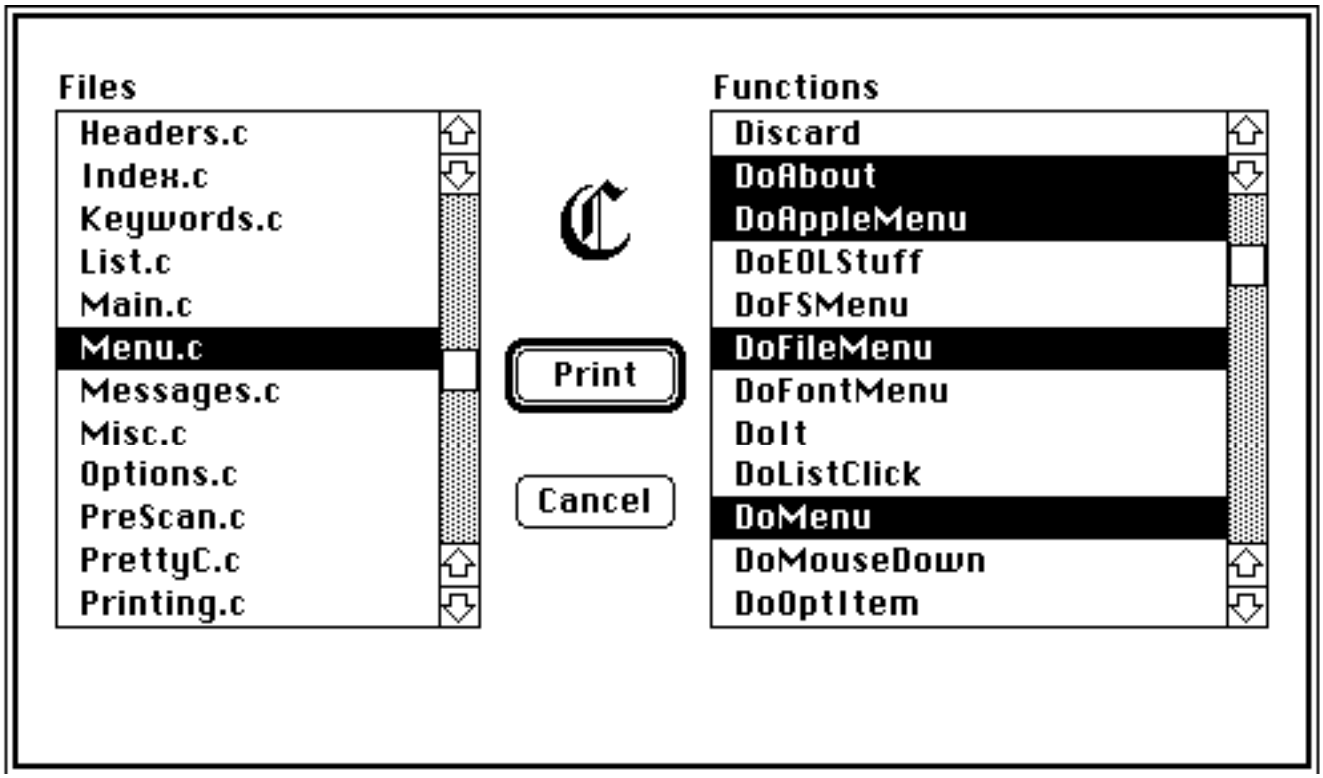
Running

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.

Selecting from the lists by clicking and shift-clicking works in exactly the same way as Font/DA mover. 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, keywords, 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.

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

Arguments

Arguments may be supplied to **PrettyC** from the MPW command line or by selecting files in the Finder and double-clicking. **PrettyC** treats arguments of different types as follows:

ATEXT 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 always 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.

Copyright

PrettyC is copyright © Jeremy Roussak 1990–91, and I retain all rights to it. It is written in THINK C v4.0, so parts are copyright © Symantec 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. If, after a reasonable trial period, you decide that you like it and want to keep it, please send me a cheque (or a check, if you're not British) or international money order for £10 sterling or $25 US (yes, it's a little more: it costs me over six pounds to cash a dollar cheque in this country!) along with the registration form which you can print from within **PrettyC** (see the apple menu). Alternatively, you can send a check for $20US to my brother in California (address on the registration form) and he'll see that it gets to me. If you want to register multiple copies of **PrettyC**, please contact me (not my brother): I expect we can work out a reduced fee.

Coming soon

If I think there's sufficient interest (that is, if enough people register), I will maintain and improve **PrettyC**. I'll include proper support for C++; I might incorporate a C beautifier, if I think I can stand the endless arguments about the best way to lay out C source, and I have a number of other improvements in mind…