

clpr (1.0)

Harry R. Chesley
3/29/85

1. Introduction

clpr (pronounced "clipper") is an application which takes program source code and prints it so it looks nice and is more easily read. The name is chosen for historical (hysterical?) reasons (Unix people will understand).

Since I expect people who will find clpr useful will be more sophisticated users (the Mac is not a machine to be programmed by novices), and will have access to a resource editor of one sort or another, I have set it up so that clpr can be customized to a large degree by changing a few of the resources. This way, you can tailor it to your tastes, yet the program doesn't have to carry around a lot of code to do the tailoring process but which isn't used 99% of the time.

The next section describes clpr in its released configuration. The following section describes how to tailor it.

2. Using clpr

In its initial configuration, clpr will paginate and print C programs, with normal text in 9-point Geneva, key words in bold, and comments in New York; tabs are set to eight Geneva spaces. The page headings and footings will be in 12-point Geneva. The heading will contain the file name, the page number, and a user-setable title right-, center-, and left-justified respectively. The footing will be the date and time, right justified.

If the first thing in a comment is a string (possibly empty) surrounded by international quote marks («...»), a new page will be started and the string used as the user-setable title for that and subsequent pages. All of the comment, but none of the string or the quotes will be printed on the new page. (Note: the quotes must **immediately** follow the comment start; not even a space can be present.)

clpr is very simple to operate. There are two ways to invoke it: opening it by itself, and opening it along with one or more documents. In the first case, the File menu contains the usual two functions, Page Setup and Print. In the second case, the printer page setup and print dialogs are invoked only once and then all the selected files are printed using that format.

Caveats: Tabs do not always come out exactly the way you intended, due to line expansion because of font changes. The files given to clpr must be non-formatted text files. clpr does not recognize nested comments (I don't much like them either — if you want to "comment out" a block of code, try "#ifdef NEVER ... #endif"). clpr does not recognize "//" type commenting. If you start a user-setable title but don't provide the ending quote, it won't blow up, but you won't get a very nice page heading.

3. Customizing clpr

The fonts, sizes, and styles used by clpr, the key words, the comment beginning and ending sequences, and the user-setable title quote characters can all be changed by editing clpr's resource file, as well as the usual menus and dialogs.

The configuration is kept in three resources of type STR# (string lists), numbers 256, 257, and 258.

256 contains the formatting information:

<u>String Number</u>	<u>Meaning</u>
1	Header font
2	Header size
3	Header style
4-6	Normal text font, size, & style
7-9	Key word font, size, & style
10-12	Comment font, size, & style
13	Number of spaces in a tab

Fonts are font names (i.e., "New York" or "Geneva"). Sizes are simple decimal numbers in ASCII (i.e., "12"). Styles are the sum of the following constants, also in decimal:

<u>Style Constant</u>	<u>Meaning</u>
0	Normal
1	Bold
2	Italic
4	Underline
8	Outline
16	Shadow
32	Condensed
64	Extended

The strings in STR# 257 are the key words. They must either be: (1) a letter or underbar followed by letters, numbers, and underbars; or (2) single characters.

The four strings in STR# 258 define the following:

<u>String Number</u>	<u>Meaning</u>
1	Beginning of comment sequence
2	End of comment sequence
3	User-setable title beginning quote character
4	User setable title ending quote character

In this STR#, each string begins with a space, which is ignored (this is to get past a problem in rmaker, which I can't get to accept an asterisk in the first column — it thinks I mean a comment). Following the space, the comment sequences can be one or two characters long; the title quotes are single characters only.

Originally, I thought clpr could even be changed to handle a different programming language, and it mostly can be. The problem is quoted strings. I have had to build in a knowledge of C's backslash string escape convention. However, so long as the new language does not use backslashes before quotes (single or double), and so long as quotes are always paired in the language, it should work OK.

If you give other people copies of this program, please try to give them the original configuration, so that the first part of this documentation applies.

4. Conclusion

I hope everyone enjoys using clpr — it's a program I've written or ported for two versions of CP/M, PC-DOS, and now the Macintosh, and I've always gotten a lot of use out of it.

Already, I can think of several possible extensions/improvements to clpr. If you're wondering if you're own idea for a change has already been stuck in, the latest version will always be on-line on my BBS: 415-563-2491 (24 hrs, 300 or 1200, 8 bits, no parity). At least so long as the BBS itself is on-line...