The

codaboot

Tool Manual

Object-oriented programming in C

OC

Stephen M. Jones

Shell program: codaboot (Version 3.0 ***2020-06-01***)

www.coda-c.com

This document was typeset with GNU troff (***groff***) running on MAC OS X.

CREATING COMMAND LINE TOOLS

The App Store enforces security measures that prevents users from installing potentially unwanted programs, which prevents Apps from including unrestricted command line tools. The ***Coda-C App*** does not include any command line tools, but it does provide you with the resources to create the ***Coda-C*** tools for personal use on the computer where a licensed version of the ***Coda-C App*** is installed.

Before you bootstrap the command line tools you will need a copy of the **Xcode.app** installed and you will need to pick a folder to contain the command line tools. The`codaboot.sh` script defines three possble folders to pick or you can use a custom folder.

1. `/act/coda-c/bin`
2. `/usr/local/bin`
3. `$HOME/Library/Containers/com.actmac.ACTerm/Data/bin`

The first is simply a path name that I like. The second is standard location for **Unix** programs on the current machine. Even though the `/usr/local` folder may already exist, you may have difficulties creating the `/usr/local/bin` folder due to the enhanced security features of **OSX**. The third folder is a folder that will be created if you have the ***ACTerm.app*** and have used it. Finally if you choose to use a custom path, it should be writable by you and it is highly recommended that the last folder be named "`bin`", like `/Users/sally/bin`.

Once you have picked a path, it will normally need to be in you shell’s search path so you can run the new commands without typing a full path name. The`/usr/local/bin` is normally automatically added to paths on OSX, the special ***ACTerm*** path is added if using the ***ACTerm.app***, otherwise you may need to update a shell startup file like `~/.bash_profile` and set the **PATH** variable there.

A further consideration is that you may want to create a corresponding `.../lib` folder if you are planning to building static libraries. This is the default location of new libraries, for example `/usr/local/lib`.

To start the process, open Terminal.app or ACTerm.app and paste the following command in blue.

>>> `/Applications/Coda-C.app/Contents/Resources/tools/codaboot.sh` <<<

Once you have started the script and selected your path, you must type "**go**" to attempt to build the **codaboot** command line tool.

EXAMPLE SETTING PERMISSIONS

On my workstation, I was able to setup `/usr/local/bin` for login id steve with the following:

```
        sudo chownste ve /usr/local/bin
        sudo mkdir /usr/local/lib
        sudo chownste ve /usr/local/lib
```

USING THE CODABOOT PROGRAM

If you have successfully built the first program and it’s location is in your search path, just type **codaboot** to start the new program. The**codaboot** prog ram will show it’s version, the target folder to build the remaining programs, and show a table of programs that you can build. You can enter the number of the corresponding program to build a single program, enter ’ALL’ to build all programs, or type ’HELP’ to display this document. When you are finished, press <`ESc`> to exit.

The following tools are available for ***Coda-C*** version **3.0**:

#	Tool	Description
1.	`codaboot`	Coda-C bootstrap program
2.	`coda`	Coda-C Source Processor
3.	`codabuild`	Coda-C Build Settings Editor
4.	`codamake`	Coda-C Compiler Controller
5.	`codagrep`	Coda-C Source Searching Program
6.	`codall`	Coda-C Folder Listing Utility
7.	`codalist`	Coda-C Property List Utility
8.	`codacset`	Coda-C ACTerm Character Set Display

All of the ***Coda-C*** command line tools start with "`coda`" in order to limit the possibility of having name collisions when built in a location like `/usr/local/bin`. Some of these programs have recommended shorter names that can be implemented with simple shell scripts, for example ll contents: ’`exec codall "$@"`’. The following table lists the alternate names for tools.

Aka	Tool	Description
`cb`	`codabuild`	Coda-C Build Settings Editor
`cfar`	`codamake`	Coda-C Fix Archives
`cgrep`	`codagrep`	C Source "grep"
`ll`	`codall`	C Folder Listing Utility
`clist`	`codalist`	Coda-C Property List Utility
`cset`	`codacset`	Character Set Display - ACTerm