

GNU APL
on Macintosh

A brief installation guide

by

Paul Rockwell & Peter Teeson

Version 2.0
25 October 2026

Copyright © 2026 Paul Rockwell and Peter Teeson

Version 2.0 Updated 25 Oct 2026

- Updated for macOS 12 and later versions of macOS
- Updated URLs
- Added instructions for keeping GNU APL updated for bug fixes, new features, etc.

Copyright © 2014, 2015, 2016, 2017 Peter Teeson

Version 1.06 Added information on how to use Doxygen 8 Aug 2017

Version 1.05 Minor clarification 13 Dec 2016

Version 1.04 Page 10 revised on 10 Oct 2015

- formatting of the text quoted from Dyalog's site fixed
- various updates based on Yosemite 10.10.5 as the current OS X

Version 1.03 Page 10 and 11 revised on 14 August 2014

- to use new URL for the Dyalog fonts and MacAplAlt.keylayout provided with the release

Version 1.02 Pages 4 and 5 revised on 2 May 2014

- to point to latest release URL

Version 1.01 Page 11 items (2) and (3) revised on 22 March 2014

- better terminal instructions

We, the creators of this work, hereby grant the permission to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation. The GFDL license can be viewed at https://www.gnu.org/licenses/fdl-1.3.html.

Apple, Mac, Macintosh, and macOS are trademarks of Apple Inc., registered in the U.S. and other countries or regions.

All other trademarks, service marks, and company names are the property of their respective owners.

Contents

Acknowledgements iv

Introduction v

Introduction to version 1 vi

1. Quick Start 1

2. Building GNU APL 2
- 2.1. Install pre-requisite software . . . 2
- 2.2. Download the GNU APL source distribution . . . 4
- 2.3. Build and install GNU APL . . . 4
- 2.4. Install the APL keyboard layout . . . 6

3. Using GNU APL 11

4. Updating GNU APL 13

Appendices

A. Optional Tasks 14
- A.1. Install other APL fonts . . . 14
- A.2. Rebuild the GNU APL source code documentation . . . 16

B. Build command summary 18

C. Resources 19

Acknowledgements

Thanks go out to:

- Peter Teeson for graciously allowing me to update his original guide.
- Mike Hall for reviewing the intermediate drafts and providing valuable suggestions on the document flow.

Introduction

I first used APL during my undergraduate work and had always been impressed by the elegance and compact nature of the language. Alas, my career took me in a direction toward more traditional programming languages, and APL became a distant memory. But a fond memory nonetheless.

Fast forward. I started to have some spare time on my hands and I decided to reacquaint myself with APL. That led to a search for a free APL implementation that would run on both the Macintosh and Linux. Like Peter Teeson, I also discovered Dr. Jürgen Sauermann's implementation of GNU APL https://www.gnu.org/software/apl.

It took Peter's *GNU APL on Macintosh. A brief installation guide* (thankfully bundled with the GNU APL distribution) to help fill in the gaps on getting GNU APL up and running on the Mac. But I did have to modify Peter's instructions somewhat to get things working. The latest version of his guide at the time was published in 2017, and macOS and GNU APL had both evolved since that time.

This new version of the guide should make it easier to install GNU APL on macOS 12 Monterey and later on both Intel and Apple Silicon Macs (and yes, it runs natively on both architectures!). As Peter said in the introduction to version 1: YMMV.

Enjoy!

Paul Rockwell
25 October 2026

Introduction to version 1

For a long time I've been looking for a way to run APL on my Mac. There were and are a number of implementations, both commercial and free, but the ones I tried didn't satisfy me. Not their fault - just my pickiness.

In the last little while I had a programming problem that was going to be fairly easily solved using APL as opposed to C++, C, or Obj-C. So I revisited what solutions there might be for APL. Like so much in life, timing is important and this time I lucked out by finding GNU APL at https://www.gnu.org/software/apl.

Dr. Jürgen Sauermann has made a recent implementation of the ISO standard 13751 *[aka. Programming Language APL, Extended]*. By recent I mean October 10 2013, the earliest date shown on the above website. It only took me a few scattered hours, over a couple of days, to get it installed and working like a charm. And it's scriptable!!! Essentially there were only 2 issues - *building and installing it and setting up the keyboard*. Below are instructions based on my experience. YMMV.

Peter Teeson

1. Quick Start

If you aren't familiar with building software from an open-source distribution on macOS (including using the C++ compiler), skip to chapter 2: Building GNU APL on page 2.

But if you are, then here's how to quickly get GNU APL up and running:

1. Open a Terminal window.
2. Install the **Command Line Tools for Xcode** from Apple.
3. Install the **`subversion`** package from either MacPorts or Homebrew.
4. Create a directory to hold the GNU APL sources.
5. **cd** to that folder, and get the GNU APL sources:

```
% svn checkout https://svn.savannah.gnu.org/svn/apl/trunk
```

6. Configure, build, and install GNU APL:

```
% cd trunk
% ./configure CXXFLAGS="-O2 -std=gnu++17"
% make
% sudo make install
```

7. Install the APL keyboard layout. See section 2.4: Install the APL keyboard layout on page 6.

2. Building GNU APL

This chapter assumes you are a somewhat experienced programmer[1] and know your way around *nix-like operating systems.

Since there's no installation package for GNU APL on macOS, you'll need to build it yourself. You'll need a bit of help to get GNU APL running on macOS, though. The GNU APL documentation is written for Linux and doesn't take into account the differences between macOS and Linux. This guide provides the help you'll need.

Here's a high level overview of what you'll need to do.

- Install software required to download and build GNU APL
- Download the GNU APL source distribution
- Configure, build and install GNU APL
- Install the APL keyboard layout so you can type APL symbols

We'll go into each of these steps in the following sections.

2.1. Install pre-requisite software

You'll need two software packages not provided "out of the box" in order to build GNU APL on macOS:

- The **Command Line Tools for Xcode**. This is software that's available from Apple that contains the C++ compiler and other development utilities you'll need to compile GNU APL
- The open-source **subversion** source code management utility. This is the recommended method to get the GNU APL sources you'll use for the build.[2]

[1] Otherwise why would you want an interpreter for A Programming Language?

[2] You'll find a tar.gz file of the GNU APL 2.0 source code available on the web site, but it is not recommended for use on the Mac. That file contains a version of GNU APL that is not current and has known bugs (including some that will prevent GNU APL from compiling on the Mac).

Install Command Line Tools for Xcode

You can verify if you have the **Command Line Tools for Xcode** installed by opening a Terminal window and typing the following command:

```
% xcode-select -p
```

- If the command returns **`xcode-select: error: unable to get active developer directory`**, you don't have the tools installed.
- If it returns a valid path name, the tools are already installed.

Here's how to install the **Command Line Tools for Xcode** if you don't have them:

1. Open a Terminal window.
2. Enter the following command to request installation of the tools:

   ```
   % xcode-select --install
   ```

3. A window will display asking to confirm the installation of the tools. Allow the installation.
4. A window will appear showing the progress of the installation. Wait for the tools to download and install.

Install subversion

The **subversion** utility used to get a copy of the GNU APL source code isn't provided by Apple. It is available from either of the two popular open-source package managers: Homebrew or MacPorts.

If you don't have either Homebrew or MacPorts installed already, pick one (it doesn't matter which one) and install it. Instructions for installation can be found:

- for Homebrew at https://brew.sh
- for MacPorts at https://www.macports.org/install.php

Once you have one of these installed, open a Terminal window and install **`subversion`**:

- for Homebrew:

  ```
  % brew install subversion
  ```

- for MacPorts:

```
% sudo port install subversion
```

2.2. Download the GNU APL source distribution

1. Open a Terminal window.
2. Create a build folder for GNU APL.

   I chose to create a folder named **`APL`** in my user account's Documents folder (**`~/Documents`**) to hold the GNU APL sources..

   ```
   % mkdir ~/Documents/APL
   ```

3. **`cd`** to the newly created folder.

   ```
   % cd ~/Documents/APL
   ```

4. Get the GNU APL sources using **subversion**'s **`svn`** command.

   ```
   % svn checkout https://svn.savannah.gnu.org/svn/apl/trunk
   ```

You'll now see a subfolder **`trunk`** that contains the GNU APL sources.

2.3. Build and install GNU APL

1. **`cd`** to the **`trunk`** folder.

   ```
   % cd ~/Documents/APL/trunk
   ```

2. Configure GNU APL for macOS.

   ```
   % ./configure CXXFLAGS="-O2 -std=gnu++17"
   ```

Note

The **CXXFLAGS=** argument to the **configure** command tells the Apple C++ compiler to process the GNU APL source code with the C++ 17 standard (with GNU extensions).

This argument is required. Omitting it can cause GNU APL build failures on some macOS releases.

3 Compile and link GNU APL.

```
% make
```

4 Install the compiled GNU APL binaries and support files.

```
% sudo make install
```

GNU APL and its components are installed in the **/usr/local** folder.[3]

You should now be able to verify that the installation has completed by typing **apl --version** in a Terminal window. The version of GNU APL version you just installed plus some other build-related information should be displayed:

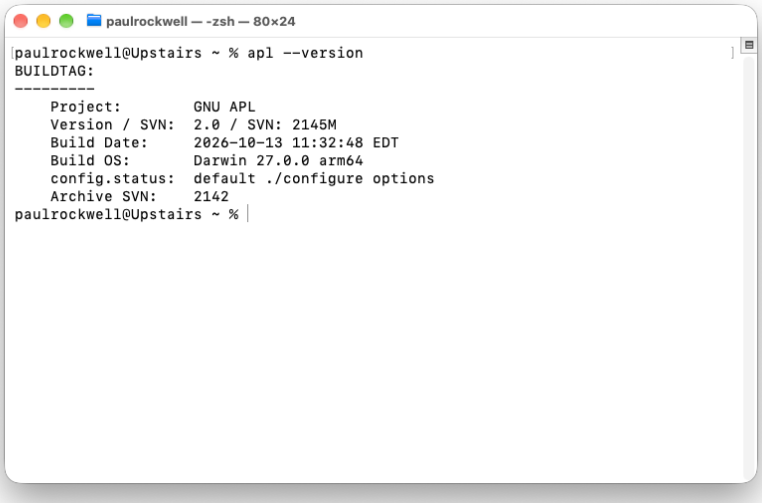


Figure 2.1: Verifying GNU APL installation

[3] See the GNU APL documentation if you would like to install it in a different location.

This completes the build and installation phase. You now need to install the keyboard layout needed for APL.

2.4. Install the APL keyboard layout

In order for you to type APL characters, macOS needs an APL keyboard layout. But the Apple-provided keyboard layouts don't know about the APL character set. Thankfully, a macOS keyboard layout file with the APL character set is provided by GNU APL.

Here's how to set it up:

Note

The following screen shots were taken from macOS 27 Golden Gate. The layout may differ in other macOS releases.

1. Open a Terminal window.
2. **cd** to the folder in the GNU APL sources that contains the layout file.

   ```
   % cd ~/Documents/APL/trunk/support-files/OS-X-keyboard
   ```

3. The layout file is named **MacAplAlt.keylayout**. Use one of the following to install the layout (and yes, the double quotes are required):

   - If you only want your user account to use the layout, copy the file to your user account's Library folder:

     ```
     % cp MacAplAlt.keylayout "~/Library/Keyboard Layouts"
     ```

   - If you want the layout to be available to all users, copy the file to the system-wide Library folder:

     ```
     % sudo cp MacAplAlt.keylayout "/Library/Keyboard Layouts"
     ```

   Most users will choose the former.

4. Log out of your Mac and log in again in order for these files to be recognized.
5. Enable the newly copied keyboard layout for use.

   - Open **System Settings** (or **System Preferences** for older macOS releases).
   - Select **Keyboard** in the left pane to open the Keyboard preference panel.

- Scroll to the **Text Input** section in the Keyboard preference panel and click on the **Edit...** button next to the **Input Sources**.

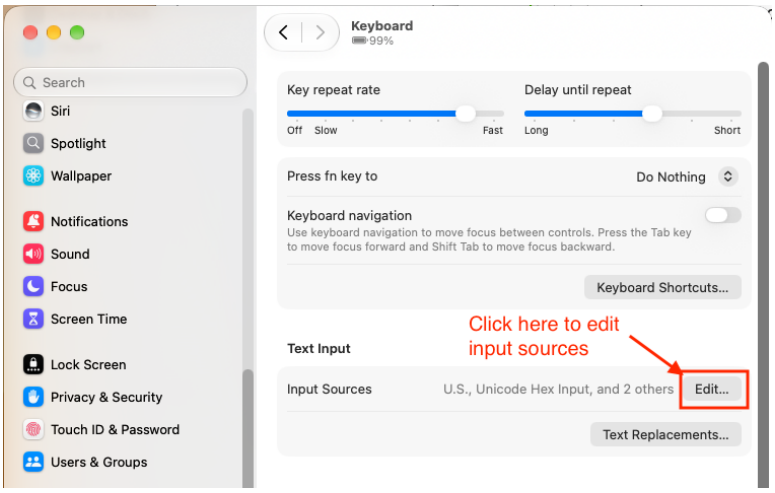


Figure 2.2: Keyboard preference panel

The **Input Sources** panel will open and will show the currently available sources.

- Click the **+** (plus sign) on the bottom left of the **Input Sources** panel to add a new input source.

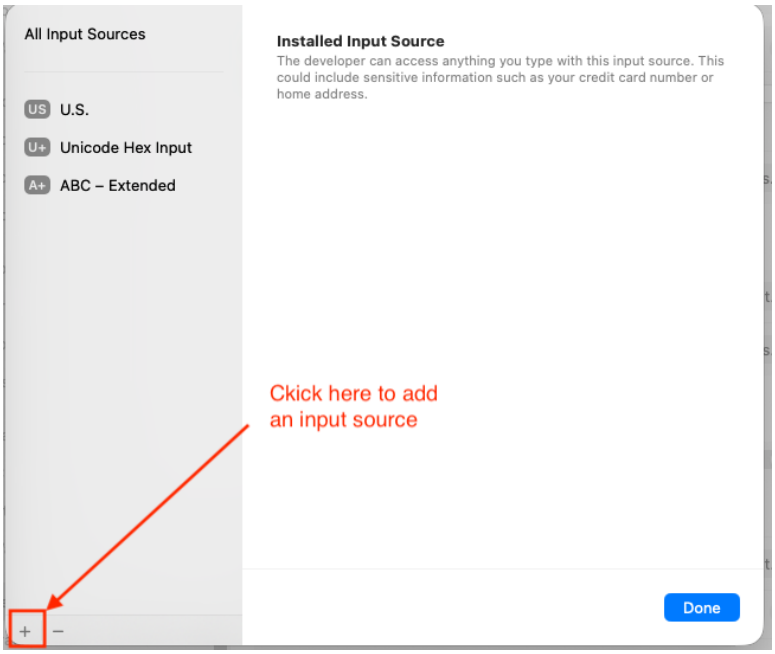


Figure 2.3: Currently available input sources

- A new panel will open and you should see all of the input sources available on your Mac. Type in **MacAplAlt** in the search area at the bottom left of the panel.

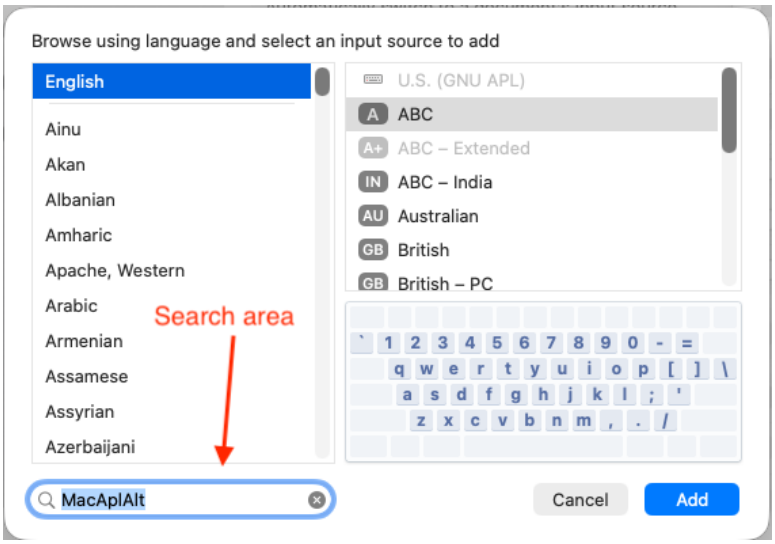


Figure 2.4: Search for MacAplAlt

- Highlight **MacAplAlt** once you have located it (it should appear under the **Others** category in the left-hand pane). Click on the **Add** button at the lower right of the panel to add it to the list of available input sources.

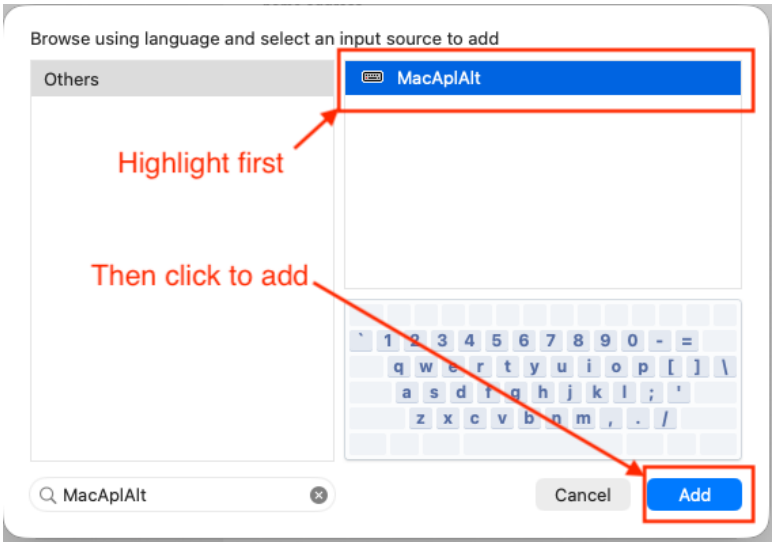


Figure 2.5: Add MacAplAlt input source

- You will be returned to the **Input Sources** panel. You will now find **MacAplAlt** under the list of **All Input Sources**.

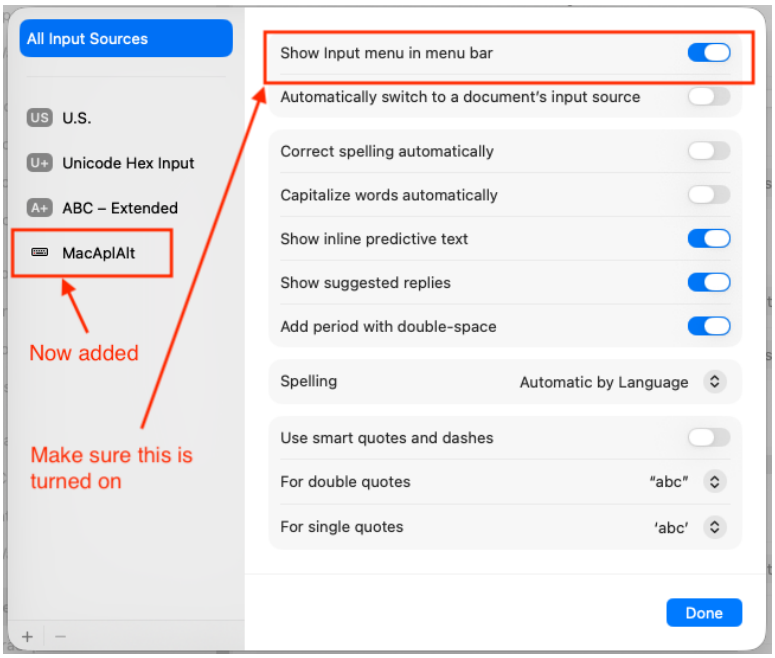


Figure 2.6: List of input sources after adding MacAplAlt

- Ensure that the **Show input menu in menu bar** item is enabled in the **Input Sources** panel. That will add the Input Source icon to the Mac's menu bar that will allow you to switch to and from the keyboard layout for the APL character set.
- Click **Done** in the **Input Sources** panel. You'll be returned to the Keyboard preference panel.
- Close **System Settings** (or **System Preferences**).

You're now ready to start using APL on macOS.

3. Using GNU APL

To use GNU APL, first open a Terminal window.

Then switch the input source to **MacAplAlt** so you can type the APL character set. That's done using the **Input Source** menu found on the macOS menu bar. Here's how:

- Locate the Input Source menu in the macOS menu bar. It will usually be a rounded box with a 2 character abbreviation of the current input source. (We'll refer to this as Input Source since the actual menu icon may not be same on all Macs.)

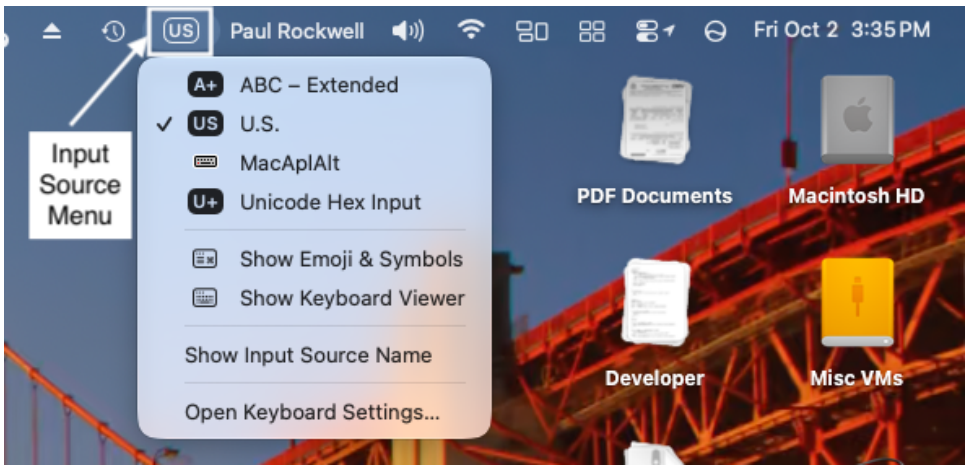


Figure 3.1: Input Source menu in macOS menu bar

- Select Input Source › MacAplAlt. That switches the keyboard layout so that you're now able to type the APL characters by using the Option and Option+Shift keys on your keyboard.
- Select Input Source › Show Input Source Name if it's available. That will make it easier to find the Input Source menu and give a better indication of which keyboard layout is currently in use.

Not sure where all the APL characters are on your keyboard? Go back to the macOS menu bar and select Input Source › Show Keyboard Viewer. The Keyboard Viewer opens and the current character mappings are now visible. Since you've already selected the **`MacAplAlt`** keyboard layout, press the Option key and... *voilà!*. Your familiar APL symbols should now appear. Try Option+Shift to get the rest.

Note

Selecting the **MacAplAlt** keyboard layout makes that layout active across all applications that you're using. That's usually not a problem. But if it does cause an issue with another application (for example generating accented characters), you can switch back to the original keyboard layout from the Input Source menu.

Type **`apl`** in the Terminal window and enjoy!

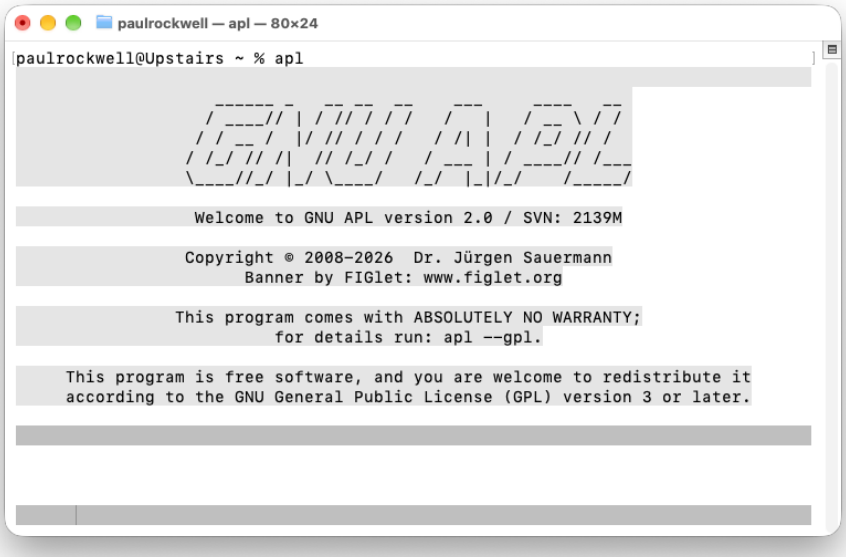


Figure 3.2: GNU APL running in Terminal window

4. Updating GNU APL

From time to time, GNU APL is updated to fix bugs, enhance performance, and add new features. Using **subversion** to get GNU APL source code lets you update your GNU APL installation with these new versions.

Here's how to keep GNU APL up-to-date:

1. Open a Terminal window.
2. **cd** to the GNU APL source folder.

   ```
   % cd ~/Documents/APL/trunk
   ```

3. Prepare the source folder for update.

   Enter these commands to clean up the build folder and reset any files that may have been changed during the build process to their original contents.

   Don't forget the period in the **svn revert** command - it's needed!

   ```
   % make clean
   % svn revert -R ./
   ```

4. Update the GNU APL sources to the latest available version.

   ```
   % svn update
   ```

5. Rebuild and reinstall GNU APL.

   Use the following commands to rebuild and install GNU APL (the same way you did for the initial installation):

   ```
   % ./configure CXXFLAGS="-O2 -std=gnu++17"
   % make
   % sudo make install
   ```

GNU APL is now updated to the latest version.

A. Optional Tasks

Note

Most users will not need to install additional APL fonts or use the documentation for the GNU APL source code. Unless you're curious, need to install other fonts for use with GNU APL, or are a GNU APL developer, you can safely ignore this chapter

A.1. Install other APL fonts

macOS contains fonts that will display the APL character set. If you would like to install other fonts to use with GNU APL, read on.

Other fonts containing the APL character set can be downloaded from Dyalog at https://dyalog.com/apl-font-keyboard.htm. Here is a quote from that page:

> "This page contains resources that are believed to be of general interest to the APL community. The fonts are freely available to everyone irrespective of whether you use Dyalog (or any other APL system for that matter)".

Indeed they are and we applaud Dyalog for making them available.

Here's how to install Dyalog's APL fonts for use by GNU APL:

1. Get the APL font from Dyalog.

   Open a browser and go to https://dyalog.com/apl-font-keyboard.htm. Select the desired APL font that you want to use and click on it to start the download. My personal recommendation is to use the APL 385 fixed width font.

   The font will be downloaded to your browser's downloads folder with a `.ttf` extension.

2. Use the Finder to locate the font file in your downloads folder and double click on it.

3. The **Font Book** utility will open and display the font in a new window. Click the **Install** button at the bottom of the window to install the font. The font will be installed for use by your user account[1].

[1] If you want all users to be able to use the font, check the Font Book documentation for instructions on changing the font installation location.

4 Close **Font Book**.

After installing the font, you need to create a custom profile in the Terminal so that windows can display using the Dyalog APL font instead of the default macOS Terminal font.

1 Open the Terminal settings from the Terminal's menu bar (either Terminal 〉 Settings or Terminal 〉 Preferences depending on your macOS version).

2 Select **Profiles**

3 Highlight the **Basic** profile (the default) and click the drop-down at the lower left of the window. (it looks like a circle with 3 dots inside it)

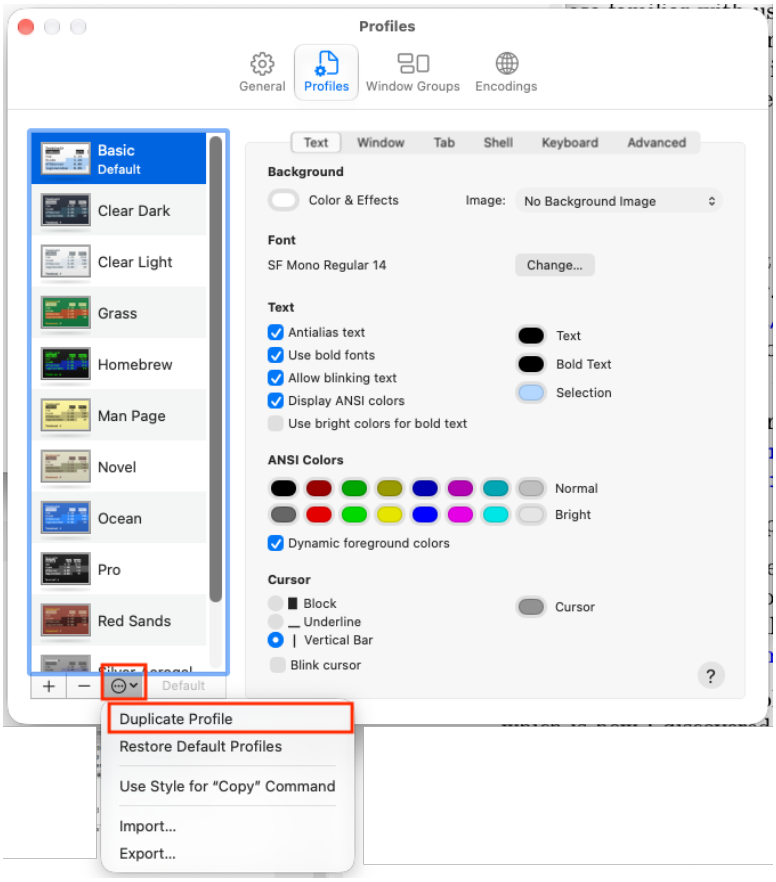


Figure A.1: Duplicating the Basic profile in Terminal settings

Select and click **Duplicate Profile**. A new profile will be create named **Basic 1**.

4 Rename this new profile from **Basic 1** to **APL**.

5 Highlight the **APL** profile, in the left hand pane and select the **Text** tab

6 Click the **Change...** button next to the default font that is displayed to change. A selection box will appear that will allow you to change the font used by this profile. Change the font to **APL 385 Unicode**. You can also change the default font size if you want.

7 Close the **Profiles** window.

To use this new profile, go to the Terminal menu bar and select Shell › New Window › APL. A new Terminal window will appear which will display its text using the APL 385 font.

A.2. Rebuild the GNU APL source code documentation

The source code for the GNU APL interpreter has been documented using the **doxygen** conventions. Although the GNU APL home page states in the Documentation section that you should use **`make DOXY`** to build this documentation, it does not work on macOS.

Why not?

It's because Apple does not provide the **doxygen** command line utility required by the GNU APL build system to re-generate the documentation. Fortunately the doxygen developers provide a **Doxygen** app for macOS so you don't have to build it yourself [2].

To install the Doxygen app:

- Open a browser.
- Download the disk image for your Mac's CPU type (ARM or Intel) from the Doxygen web site: https://doxygen.nl/download.html.
- Open the disk image (it should be in your Downloads folder) by double-clicking it.
- Drag the **Doxygen** app to your Applications folder.
- Add the **Doxygen** app to your Dock.

You should now be able to launch **Doxygen** from the Dock.

The **Doxygen** app is is a very nice GUI front end to the command line **doxygen** utility that's embedded within the app. The GUI has two modes, Wizard and Expert. Once you have configured it to your liking you just press the "run" tab and the embedded **doxygen** command line utility will be run.

[2] The current version of **doxygen** available from the developer web site supports macOS 13 Ventura and later. Older versions that support macOS 12 Monterey or earlier can be found on GitHub or SourceForge as noted on the web site.

If you want to use **`make DOXY`** to rebuild the source code documentation (per the GNU APL home page), you will have to either

- create a symbolic link to the embedded **doxygen** command line utility (found in **`/Applications/Doxygen.app/Contents/Resources/doxygen`**) in a directory that's already in your **`$PATH`** variable (such as **`/usr/local/bin`**), or
- add **`/Applications/Doxygen.app/Contents/Resources`** to your **`$PATH`** variable.

If you intend to rebuild the source documentation, you should also should also make sure you have installed **Graphviz**. The source code documentation depends on the **dot** utility found in **Graphviz**. See https://graphviz.org/download – they indicate that there are pre-built packages of **GraphViz** available from both MacPorts and Homebrew.

B. Build command summary

Configure/build commands

`configure`	– Configure the GNU APL build for the current operating system platform. Additional arguments are needed as noted in this guide for GNU APL to build under macOS.
`make`	– Compile and link GNU APL with the configuration provided by the **`configure`** command. Note: this command does not install the product.
`make check`	– (Optional) Run self-checks against the uninstalled GNU APL binaries in the build folder.
`sudo make install`	– Install GNU APL and associated support files from the build folder to the **/usr/local** directory.
`make installcheck`	– (Optional) Run self-checks against the installed GNU APL binaries.
`make clean`	– Remove the program binaries and object files produced by the **`make`** command from the build folder. Does not remove product configuration or the installed GNU APL application.
`make distclean`	– Remove product configuration files produced by the **`configure`** command. Does not remove the installed GNU APL application.

Subversion commands

`svn checkout`	– Check out a working copy (local copy) of a **subversion** repository's contents. Used to get the first copy of GNU APL sources from https://svn.savannah.gnu.org/svn/apl/trunk.
`svn revert`	– Restore the working copy to a pristine state (removing any local changes). Used to prepare the GNU APL sources for updating.
`svn update`	– Bring changes from the repository into the working copy. Used to update GNU APL sources to a later version.
`svn help`	– Get help for **subversion** and any of its subcommands.

C. Resources

There are several files in the GNU APL source **trunk** subfolder which may be of interest to advanced users for the configuration and use of GNU APL.

INSTALL – Generic documentation on the tools used to build GNU APL. Covers generic options to the **configure** command. Useful if you want to configure GNU APL to be installed in a directory other than **/usr/local**.

README-2-configure – Documentation for options to the configure command that are specific to GNU APL and impact features that are provided.

README-9-post-installation – Documentation on additional tasks you might want to perform after installation of GNU APL. This file contains a list of the configurable features (or preferences) that are available for GNU APL and how to change them. For example, this file contains information on configuring the location of the libraries used by the APL commands **)LIB,)LOAD,)SAVE,)COPY**, etc.

doc – A folder containing user-focused documentation.

doc/apl.html – The GNU APL Reference Manual in HTML format. Documentation on GNU APL extensions to the ISO APL standards can be found here.

HOWTOs – A folder containing other documentation that might be of interest to the advanced user or GNU APL developer. Strangely enough, this guide can be found in this folder.