NEIL TRAFT | A Programmer

« Freedom's Just Another Word for Nothing Left to Work For

A Taxonomy of Toolchains »

Installing GDB on OS X Mavericks

http://ntraft.com/page/2/

January 21, 2014

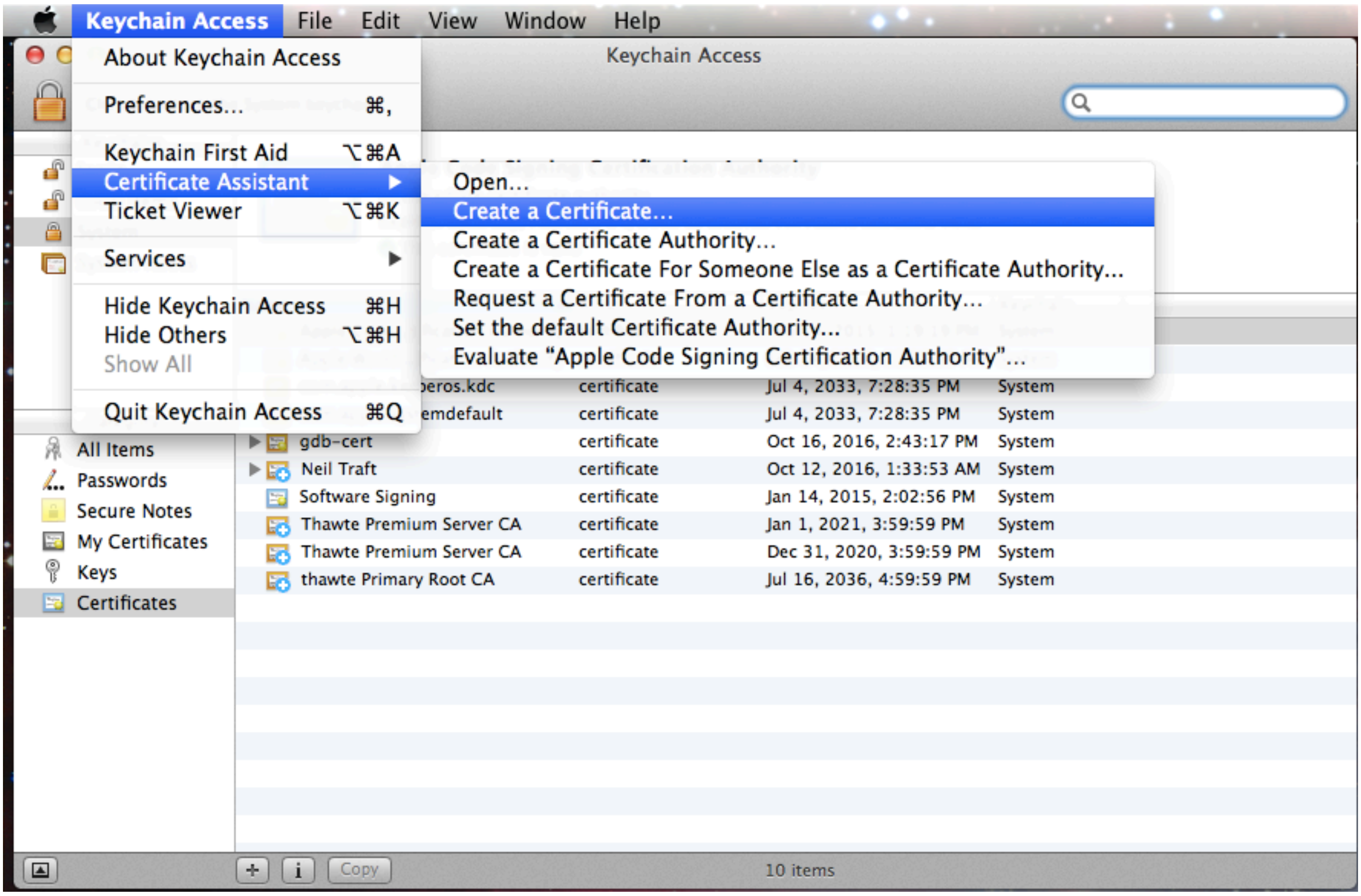
With its new OS release, Apple has discontinued the use of GDB in OS X. Since 2005 Apple has steadily been moving away from the GNU toolchain in favor of LLVM. This means that Xcode now uses LLDB instead.

LLDB looks to be a very nice replacement for GDB, and I hope to use it in the future, but currently Xcode is the only graphical front-end that supports its use; pretty much every other debugging GUI uses GDB under the hood, including Eclipse. So, if you want to debug C/C++ code in Eclipse CDT on the Mac, you *must* install GDB.[1]

Here is the procedure that worked for me. Others have reported issues with this, so please do let me know in the comments if it doesn't work for you.[2]

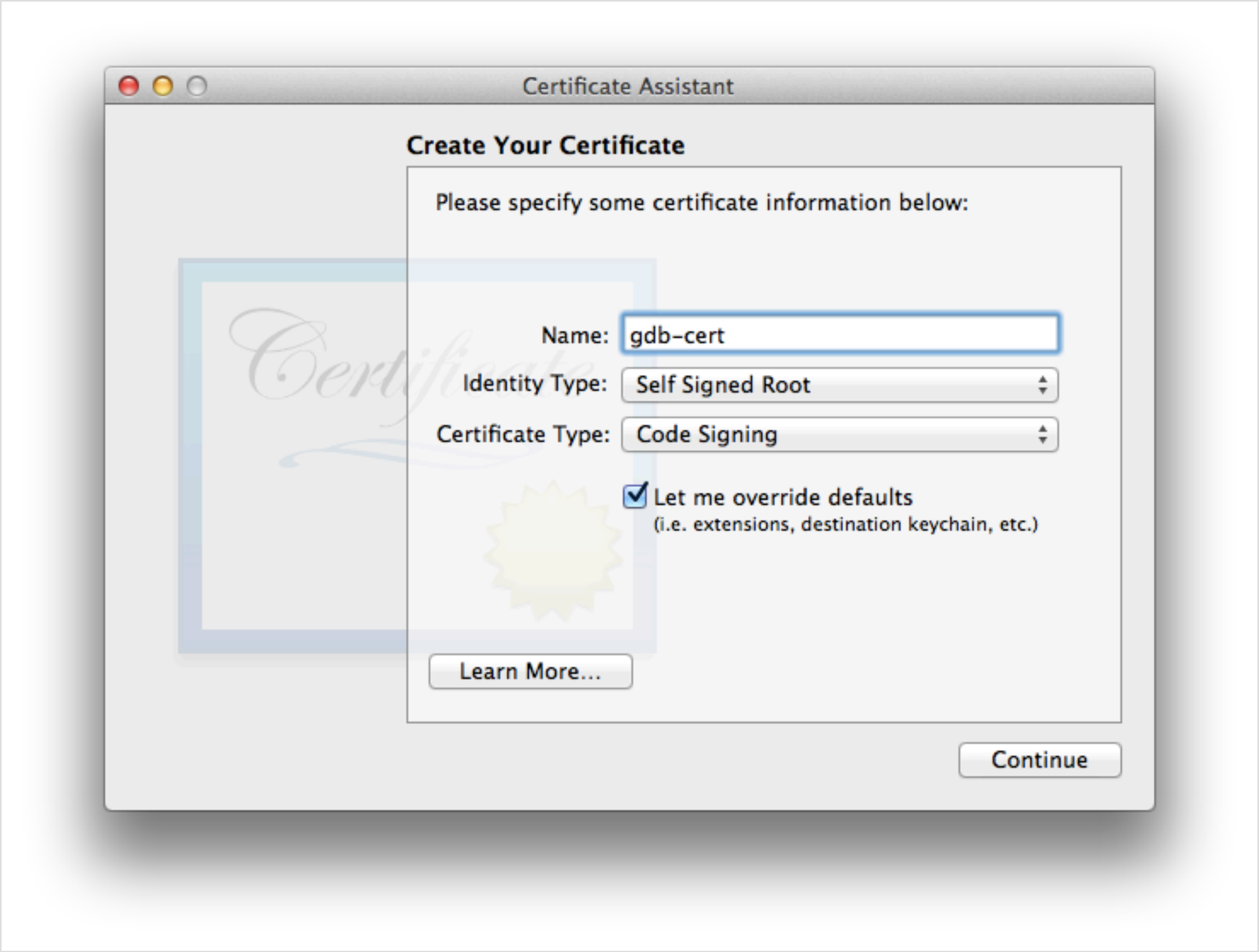
Installing & Certifying GDB

1. Install Xcode and Homebrew, if not already installed.
2. Now install GDB from Homebrew:
   ```
   $ brew tap homebrew/dupes
   $ brew install gdb
   ```
3. If that worked, then lucky you! Getting it compiled is where most people seem to have trouble. Now you just need to sign it to give it permission to control OS X processes. Open up the Keychain Access application (`/Applications/Utilities/Keychain Access.app`). Navigate via the menu to *Keychain Access* > *Certificate Assistant* > *Create Certificate...*



eros.kdc	certificate	Jul 4, 2033, 7:28:35 PM	System
emdefault	certificate	Jul 4, 2033, 7:28:35 PM	System
gdb-cert	certificate	Oct 16, 2016, 2:43:17 PM	System
Neil Traft	certificate	Oct 12, 2016, 1:33:53 AM	System
Software Signing	certificate	Jan 14, 2015, 2:02:56 PM	System
Thawte Premium Server CA	certificate	Jan 1, 2021, 3:59:59 PM	System
Thawte Premium Server CA	certificate	Dec 31, 2020, 3:59:59 PM	System
thawte Primary Root CA	certificate	Jul 16, 2036, 4:59:59 PM	System

Enter a name for the certificate. For this how-to, I'll call it "gdb-cert". Set the fields exactly as shown below.



Create Your Certificate

Please specify some certificate information below:

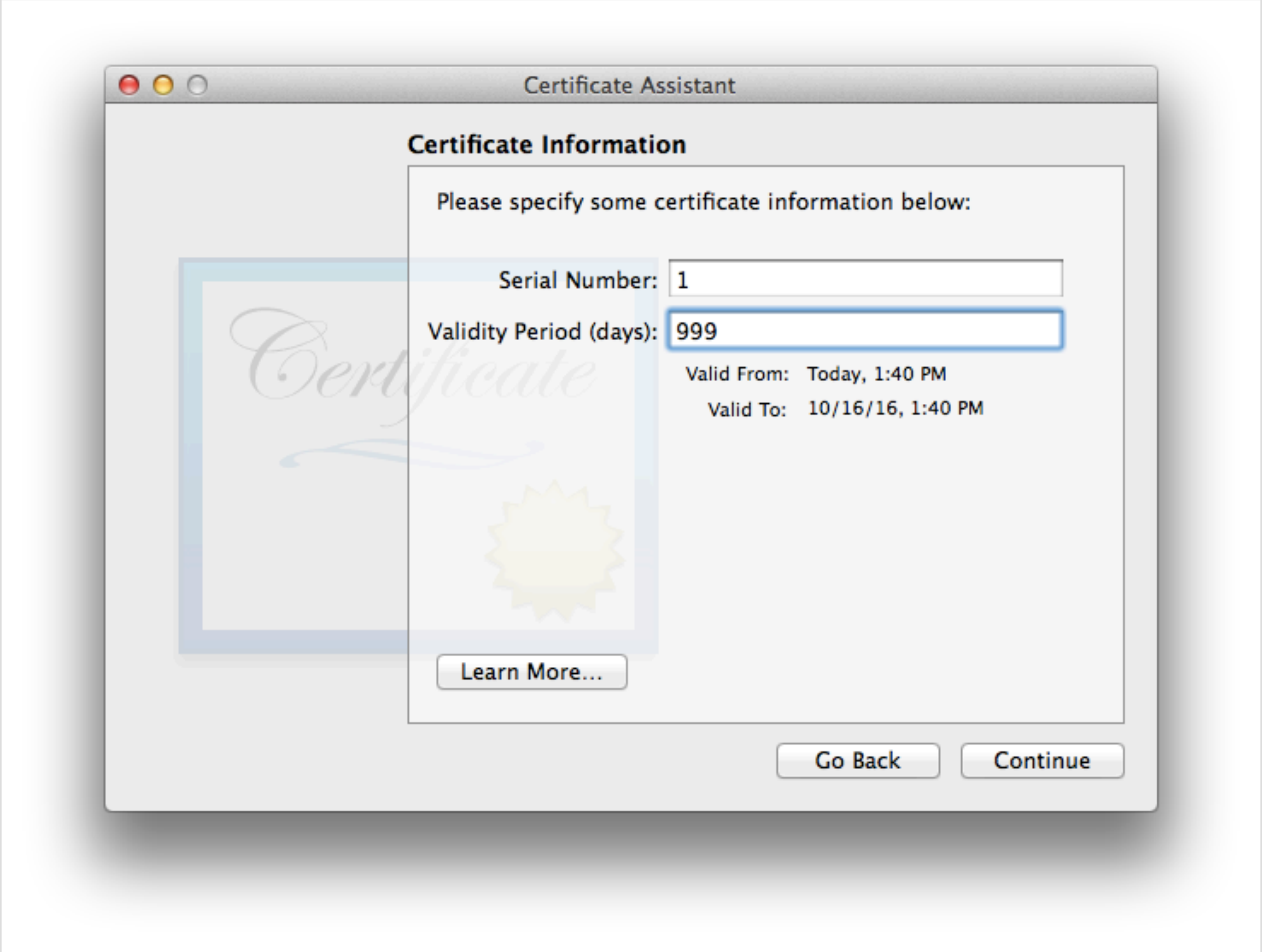
Name: gdb-cert

Identity Type: Self Signed Root

Certificate Type: Code Signing

☑ Let me override defaults
(i.e. extensions, destination keychain, etc.)

The maximum validity period is 999 days. I don't really want to deal with this again, so I'm going to max it out.



Certificate Information

Please specify some certificate information below:

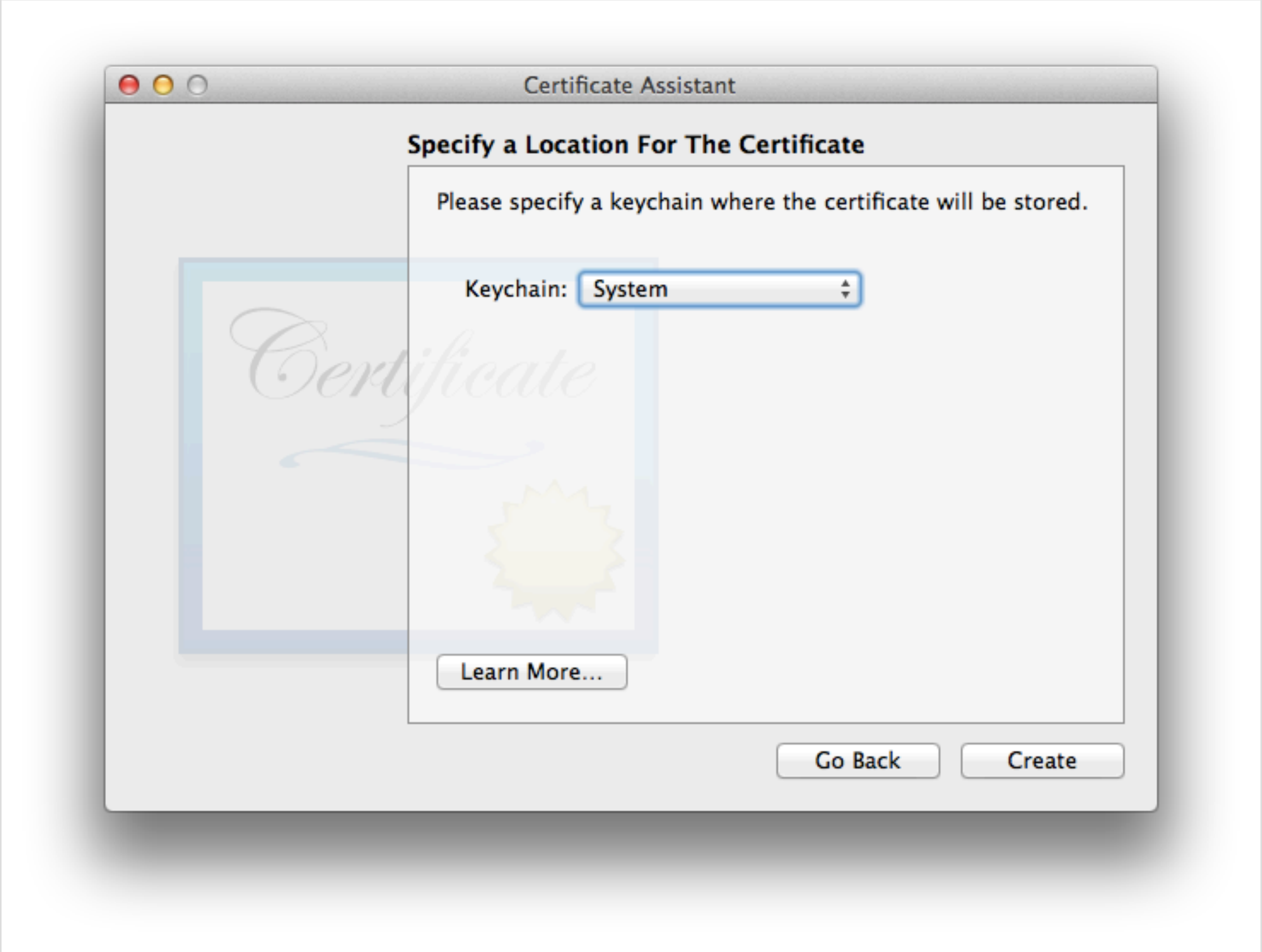
Serial Number: 1

Validity Period (days): 999

Valid From: Today, 1:40 PM

Valid To: 10/16/16, 1:40 PM

Keep clicking the "Continue" button until you are asked for a location. Set it to "System".[3]



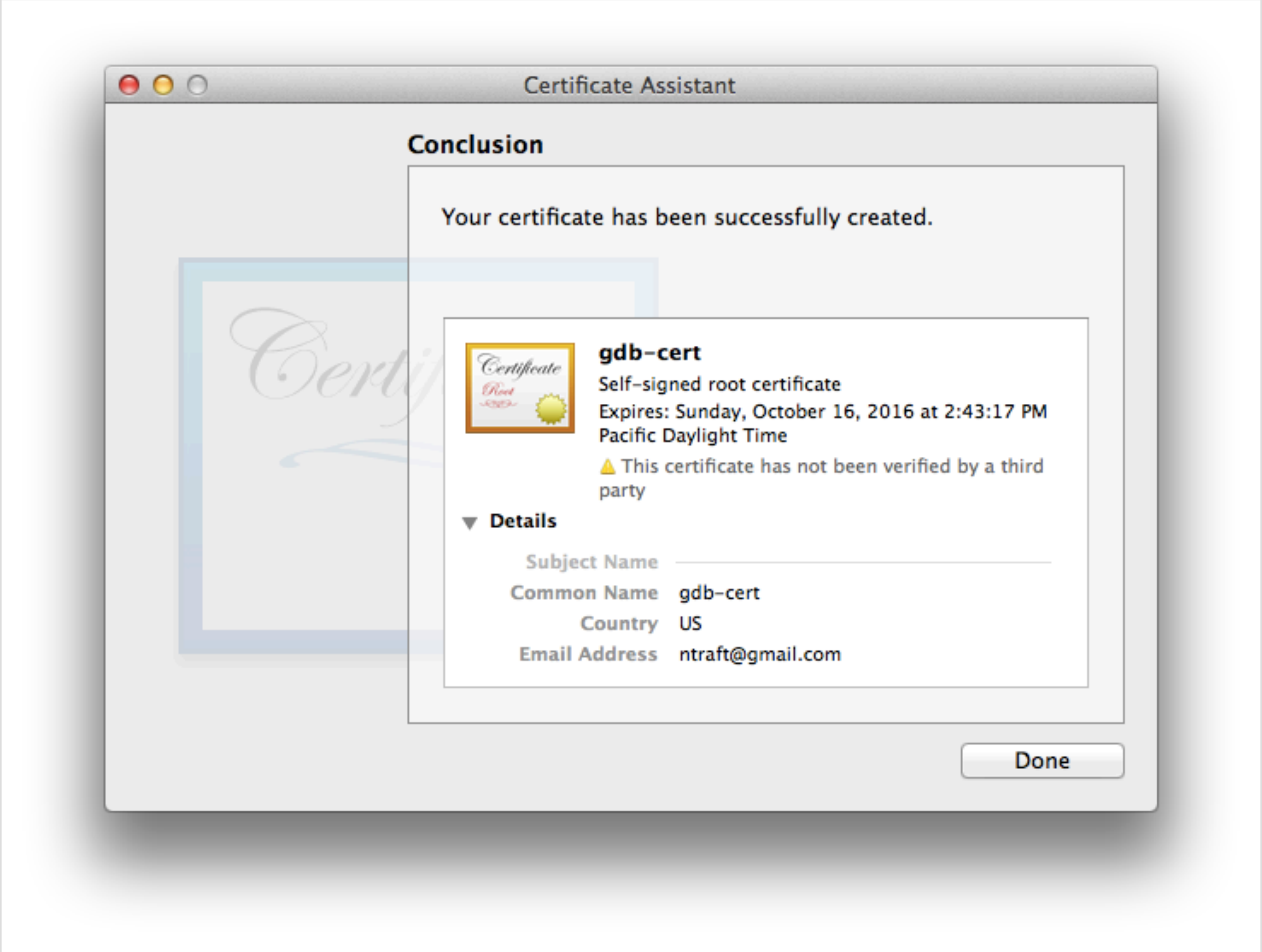
Certificate Assistant

Specify a Location For The Certificate

Please specify a keychain where the certificate will be stored.

Keychain: System

Success!



Conclusion

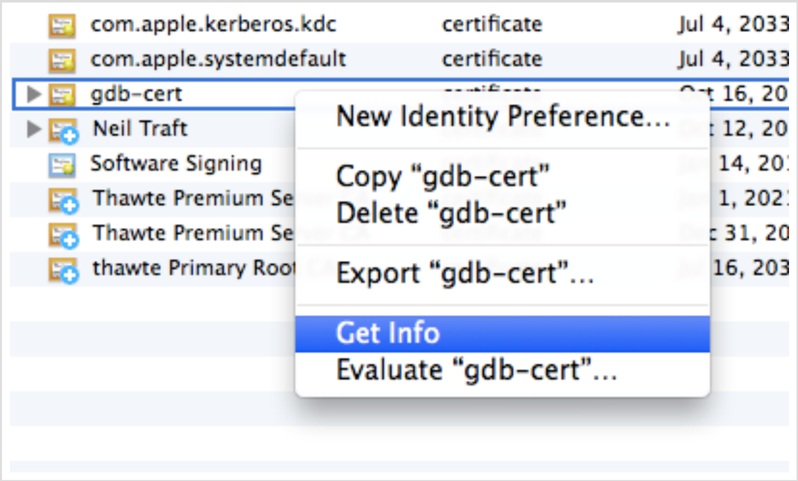
Your certificate has been successfully created.

gdb-cert
Self-signed root certificate
Expires: Sunday, October 16, 2016 at 2:43:17 PM Pacific Daylight Time
⚠ This certificate has not been verified by a third party

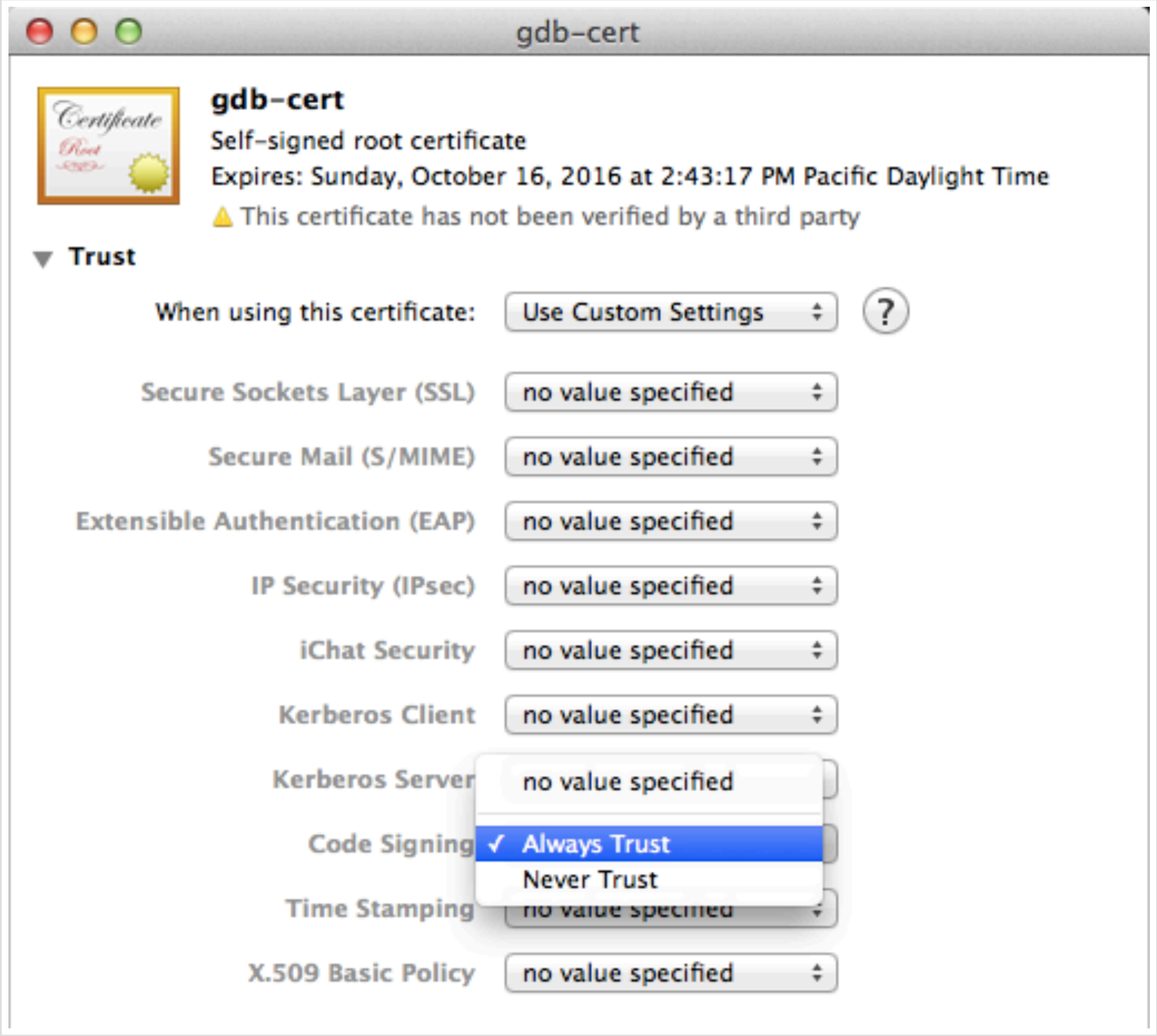
Details

Subject Name
Common Name gdb-cert
Country US
Email Address ntraft@gmail.com

Now make sure the cert is always trusted. Right-click the new certificate and select *Get Info*. Under the *Trust* section, set *Code Signing* to *Always Trust*.



com.apple.kerberos.kdc	certificate	Jul 4, 2033
com.apple.systemdefault	certificate	Jul 4, 2033
gdb-cert		Oct 16, 20
Neil Traft		12, 20
Software Signing		14, 20
Thawte Premium Se		1, 202
Thawte Premium Se		31, 20
thawte Primary Root		16, 203



gdb-cert

gdb-cert
Self-signed root certificate
Expires: Sunday, October 16, 2016 at 2:43:17 PM Pacific Daylight Time
⚠ This certificate has not been verified by a third party

▼ Trust

When using this certificate: Use Custom Settings

Secure Sockets Layer (SSL): no value specified
Secure Mail (S/MIME): no value specified
Extensible Authentication (EAP): no value specified
IP Security (IPsec): no value specified
iChat Security: no value specified
Kerberos Client: no value specified
Kerberos Server: no value specified
Code Signing: ✓ Always Trust
Time Stamping: no value specified
X.509 Basic Policy: no value specified

Now that we have a certificate, we need to use it to sign GDB. First, we'll restart the `taskgated` process to make sure it picks up the new certificate. Quit Keychain Access (you *must* quit Keychain Access!) and return to the Terminal for these final commands.

Find the taskgated process.

```
$ ps -e | grep taskgated
56822 ??         0:03.11 /usr/libexec/taskgated -s
60944 ttys002    0:00.00 grep --color=auto taskgated
```

The first number in the above output is the PID. Use this to kill the process (it will immediately restart itself).

```
$ sudo kill -9 56822
```

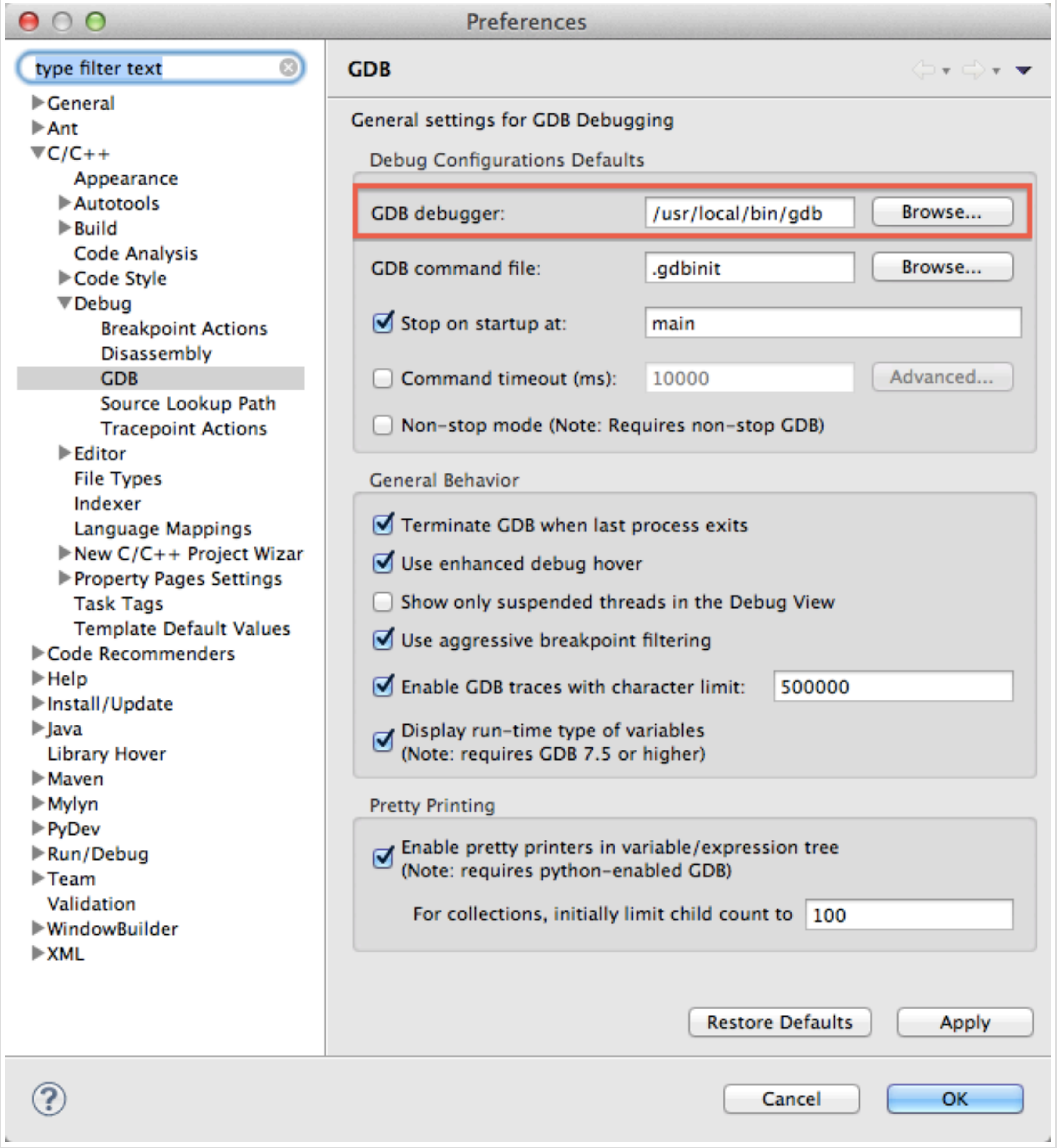
Now you can finally code sign GDB.

```
$ codesign -s gdb-cert $(which gdb)
```

Now you should be all set! The OS X Keychain may ask for your password the first time you attempt to debug a program, but it should work!

Getting it to Work with Eclipse

There's one more step for Eclipse users. You need to specify where Eclipse can find the new GDB. Specify the path to GDB in *Preferences* > *C/C++* > *Debug* > *GDB*:



GDB

General settings for GDB Debugging

Debug Configurations Defaults

GDB debugger: /usr/local/bin/gdb

GDB command file: .gdbinit

☑ Stop on startup at: main

□ Command timeout (ms): 10000

□ Non-stop mode (Note: Requires non-stop GDB)

General Behavior

☑ Terminate GDB when last process exits

☑ Use enhanced debug hover

□ Show only suspended threads in the Debug View

☑ Use aggressive breakpoint filtering

☑ Enable GDB traces with character limit: 500000

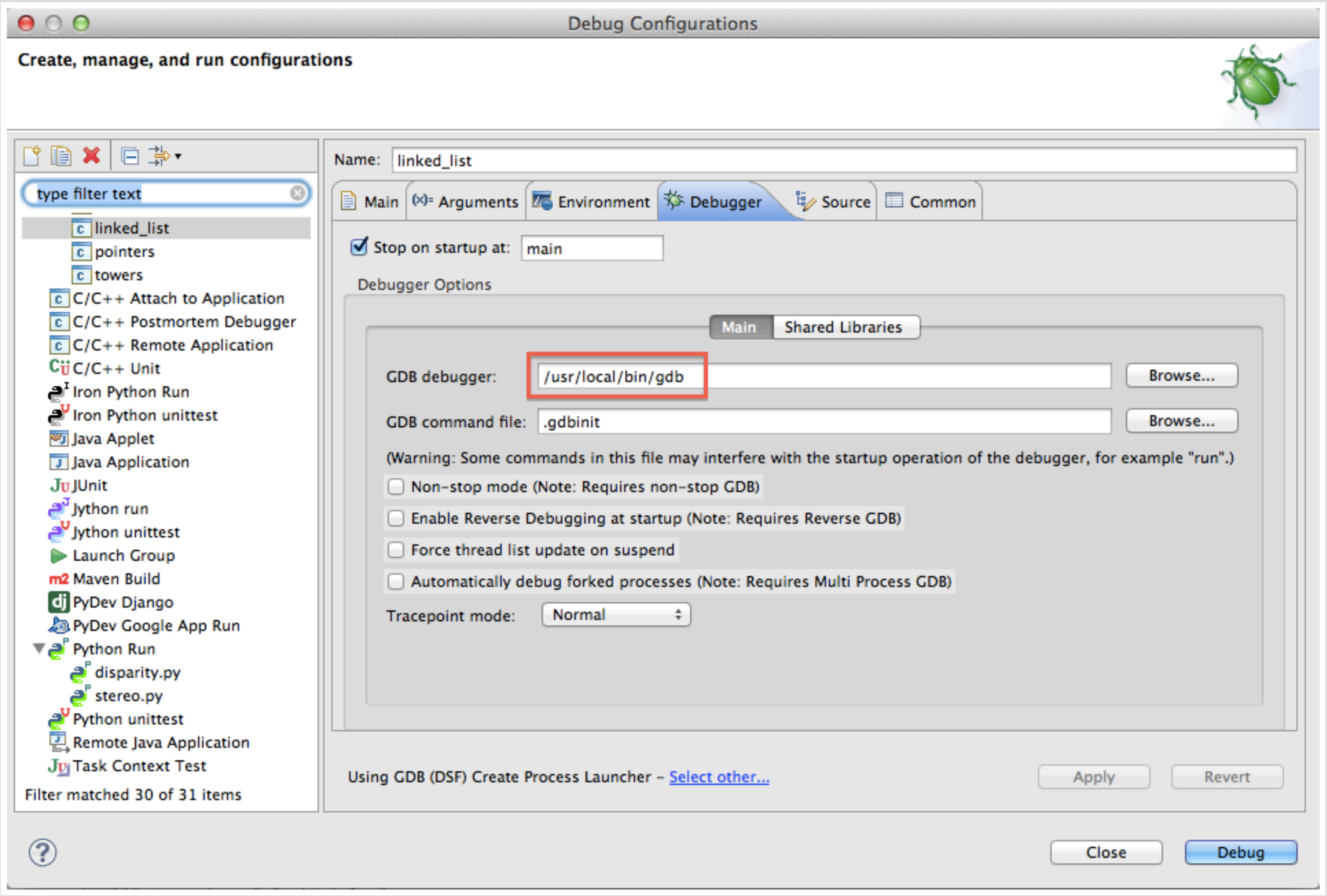
☑ Display run-time type of variables
(Note: requires GDB 7.5 or higher)

Pretty Printing

☑ Enable pretty printers in variable/expression tree
(Note: requires python-enabled GDB)

For collections, initially limit child count to 100

If you already have some debug configurations, you may need to edit them individually to point to the correct place (under *Run > Debug Configurations...*):



Name: linked_list

☑ Stop on startup at: main

GDB debugger: /usr/local/bin/gdb

GDB command file: .gdbinit

(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)

□ Non-stop mode (Note: Requires non-stop GDB)

□ Enable Reverse Debugging at startup (Note: Requires Reverse GDB)

□ Force thread list update on suspend

□ Automatically debug forked processes (Note: Requires Multi Process GDB)

Tracepoint mode: Normal

Using GDB (DSF) Create Process Launcher – Select other...

1. The CDT developers are planning to support LLDB, but they will have to write a whole new interface, and I think most of them only work on Eclipse in their spare time, so it will likely be at least some months before LLDB support is there.
2. The procedure is derived from this StackOverflow post and this GDB Wiki page.
3. If you are unable to save it to the `System` keychain, then save it to the `login` keychain. You can later *export* the cert, and then *import* it into the `System` keychain. I didn't have to do this, so comment if you have any problem.

Tagged C++, Eclipse, gdb, GNU, how-to, IDEs, osx.

« Freedom's Just Another Word for Nothing Left to Work For

A Taxonomy of Toolchains »

17 Responses to *Installing GDB on OS X Mavericks*

Kevin February 11, 2014 at 13:59

This was a huge help! Thanks for the detailed procedure.

Kevin

digitaurus February 27, 2014 at 14:03

Thanks for the article. I can't get breaks set for a shared library function to work. Have you encountered this problem ?

ntraft February 27, 2014 at 14:57

Sorry, I haven't ever tried that, and am not in a position to at the moment. No one has mentioned that in the StackOverflow post either. http://stackoverflow.com/questions/19554439/gdb-missing-in-os-x-mavericks

Chul March 2, 2014 at 06:15

Thanks. This works perfectly for my Mavericks.

fab March 12, 2014 at 11:58

Thanks. I have GDB working now. However, Eclipse allows me to debug but it doesn't allow me to set breakpoints anywhere inside templates functions. Have you guys experienced this before?

ntraft March 12, 2014 at 13:49

Uh-oh. Yeah, I just checked, and it's happening to me too. That's a much more serious problem.

It happens in raw gdb, not just Eclipse, so it's an issue with gdb somehow. It can step into template functions just fine, but not set breakpoints.

I don't have a solution for this right now.

luxward March 15, 2014 at 23:12

Help!!I just followed the step you listed.But I just got the this.

```
ps -e | grep taskgated
1282 ?? 0:00.03 /usr/libexec/taskgated -sp
sudo kill -9 1282
codesign -s gdb-cert /usr/local/bin/gdb
gdb-cert: no identity found
```

That is why?

ntraft March 16, 2014 at 11:58

In the step where you created your certificate, what name did you give it? I named mine "gdb-cert", but you need to replace that with whatever name you chose.

jl March 24, 2014 at 08:32

Hi, I have follow the steps, but when I try to debug some project, eclipse says: "'Launching 'project' has encountered a problem. Error with command: /usr/local/bin/gdb -- version"
Do you have any idea why is this happening?
Thank you

ntraft March 24, 2014 at 14:06

@jl
It means it can't find your gdb. If you go to the terminal and run "/usr/local/bin/gdb --version" do you get

some output? What is the result of running "which gdb"?

Nidhi April 6, 2014 at 20:33

Hi,
I followed your instructions to the letter but there were two problems:
1) Under Preferences > C/C++ > Debug > there is no GDB option. Instead there is a GDB MI option, so I cant set the path there.
2) I set the path to the debugger under Debug Configurations, but now when I run it on the Device I get the error :
GdbServerReceiver: run-as: Package xxx is unknown
NDK: Gdbserver not started properly
NDK: Launch error: Failed to connect to remote VM

How do i fix this?

ntraft April 7, 2014 at 11:47

@Nidhi: It looks like you're trying to run a remote debugging session on an Android device? I have no experience with that. Seems like more of a problem with gdbserver than with gdb, though, judging from your error message.

joeb April 8, 2014 at 05:54

i have it working when logged in directly on the machine but when i ssh into the machine remotely it says:

Unable to find Mach task port for process-id 33493: (os/kern) failure (0x5).
(please check gdb is codesigned - see taskgated(8))

do i need to do something else for it to work on ssh/remote connections?

any help greatly appreciated.

ntraft April 8, 2014 at 11:24

@joeb: Whoa, that's pretty weird! Maybe it has something to do with the user you're logging in as... although I thought that codesigning should mean that any user can use it.

As a temporary workaround, doing "sudo gdb" may work. Other than that, I got nothing for you. Do post back if you find an answer!

Jordan April 11, 2014 at 09:58

I had the same problem as jl who posted:

"Hi, I have follow the steps, but when I try to debug some project, eclipse says: "'Launching 'project' has encountered a problem. Error with command: /usr/local/bin/gdb -- version"
Do you have any idea why is this happening?
Thank you"

So I followed your instructions:
Your advice to enter 'which gdb' into the terminal resulted in "/usr/local/bin/gdb"

When I run '/usr/local/bin/gdb --version', I get a lot of output related to gdb, which makes me think that I do have something there.

However, I'm still encountering the result: "'Launching 'project' has encountered a problem. Error with command: /usr/local/bin/gdb -- version".

What am I doing wrong?

ntraft April 11, 2014 at 12:18

@Jordan: And you followed my instructions in the "Getting it to work in Eclipse" section? If so, then I'm not sure why it can't be accessed. Maybe it's a permissions issue.

Oliver April 11, 2014 at 15:06

Thanks for this article! It helped me alot!
But I had some problems, too! After following all instructions I got this error message:

Unable to find Mach task port for process-id 33493: (os/kern) failure (0x5).
(please check gdb is codesigned - see taskgated(8))

The solution was NOT to change the settings from 'Getting it to Work with Eclipse'.
Maybe it makes sense to try running gdb with eclipse first befor changing the settings.
Thanks alot ntraft! :-)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

Post Comment

About Me

I'm a *programmer* who aspires to be a *computer scientist*. Those are two very different things. I'm no longer sure what the words *nerd* and *geek* actually mean.

More About Me

Complete Archives

RSS

Recent Posts

- A Taxonomy of Toolchains
- Installing GDB on OS X Mavericks
- Freedom's Just Another Word for Nothing Left to Work For
- (See the Exercises)
- New Grad Student Insights

Being Social

- Bitbucket
- Facebook
- Github
- Google+
- Last.fm
- LinkedIn
- Quora
- Spotify
- Stack Overflow
- Twitter

Neil Traft | Powered by Mantra & WordPress.