

# Wifi

vendeg

Agora-Vendeg

IDE

# XCode - telepítés



# Visual Studio 2012 Express

<http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-products>

The screenshot shows a web browser window displaying the Microsoft Visual Studio Express 2012 for Windows Desktop product page. The browser's address bar shows the URL: [www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-windows-desktop](http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-windows-desktop). The page layout includes a left-hand navigation menu and a main content area. The navigation menu lists '2012 editions' with a sub-section 'Express 2012 products' containing links for 'Express for Web', 'Express for Windows 8', 'Express for Windows Desktop' (which is highlighted), 'Express for Windows Phone', and 'Team Foundation Server Express'. Below this is a link for 'Team Foundation Service' and a section for '2010 Express products'. The main content area features a large background image of a woman working at a computer. Overlaid on this image is the text 'Visual Studio Express 2012 for Windows Desktop'. Below the title, a paragraph reads: 'Sometimes your vision requires the full power of the Windows desktop. Create the next great application for Windows 8 and more—and instantly find millions of potential customers in the Windows Store.' At the bottom of the main content area are two buttons: 'See product details' and 'Download', both with circular arrows indicating a dropdown menu. The footer of the page is a dark purple bar containing the Visual Studio logo and the text 'Visual Studio', followed by navigation links for 'Products', 'ALM', 'Download', and 'Buy', and a search icon.

2012 editions

Express 2012 products

- Express for Web
- Express for Windows 8
- Express for Windows Desktop
- Express for Windows Phone
- Team Foundation Server Express

Team Foundation Service

2010 Express products

## Visual Studio Express 2012 for Windows Desktop

Sometimes your vision requires the full power of the Windows desktop. Create the next great application for Windows 8 and more—and instantly find millions of potential customers in the Windows Store.

[See product details](#)

[Download](#)

Visual Studio

Products ALM Download Buy

[www.microsoft.com/visualstudio/eng/downloads#d-express-windows-desktop](http://www.microsoft.com/visualstudio/eng/downloads#d-express-windows-desktop)



## Express 2012 for Windows Desktop

Setup requires 4.53 GB in:

C:\Program Files (x86)\Microsoft Visual Studio 11.0



You must agree to the [License Terms](#) before you can install the product.

Visual Studio 2012 automatically sends information to Microsoft about your installation experience and notifies you when an update is available and connects you to content online. To learn more, see the [Privacy Statement](#).

- ☒ I agree to the License terms and conditions.
- ☐ Join the [Customer Experience Improvement Program](#) to help improve the quality, reliability and performance of Visual Studio.

**INSTALL**

Install

# Qt Creator

# Verziókezelők

# Miért jó?

- Biztonságban van a kód
- Kód megosztás
- Látszik, hogy mi változott és miért változott
- Tegnap még működött, mi változott azóta?
- Ezt miért így írtam meg?
- Ezt ki írta?
- Ez egy stabil változat, tegyük el!
- Leágaztatás



# Perforce

- nagyvállalati felhasználás (pl.: LogMeIn)
- központosított
- jó eszközök (p4v, p4merge)
- remek konfliktuskezelés
- ingyenes változat! (max 20 felhasználó)
- egyszerű beüzemelés (Windows)
- furcsa fogalmak (shelf, check out, server workspace)
- git integráció
- sandbox

# Nagyobb open source verziókövetők

- CVS
- SVN
- GIT

# Git

Elosztott verziókövető.

Nem feltétlenül kell központi repository.

Gyors.

Éppen ez a menő.

Igen jó tutorial a githubon:

<http://try.github.com>

# Git parancsok

init

clone

add

commit

push

pull

# Git parancsok

init

clone

add

commit

push

pull

Helyi parancsok.

Érdemes saját fejlesztésnél is használni.

# Ingyenes git tárhelyek

github

bitbucket(ingyen privát repo !)

code.google.com

sourceforge

...

# Git kliensek

Van plugin minden IDE-hez (<http://www.eclipse.org/egit/>)

Windows:

tortoise git, github kliens

Mac:

XCode, <http://www.sourcetreeapp.com/>

Linux: gitk, giggle, gitg

# SSH

ssh / putty

ubuntu@192.168.4.165  
123456

Mindenki saját csináljon saját könyvtárat.  
(mkdir)  
szerkesztő: nano



```
git clone https://github.com/lmicourse/2013q1.git
```

Fent van a kurzus githubos oldalán

# Projectek

# Project - általánosságban

Hogyan állnak össze a forrásfájlok futtatható programmá.

Mit fordítunk? Hol vannak a headerök? Mi legyen a kimenet? Hol vannak a libek? Milyen libekkel linkelünk? Ezekhez a műveletekhez milyen opciókat használunk? Esetleg milyen képek kellenek?

# Debug / Release

Elsőre egyszerű kérdés, másodjára nem az.

Debug: Nincs optimalizálva. Vannak benne/mellé *debug szimbólumok*.

Release: Optimalizálva van. Nincsenek benne debug szimbólumok, mellé lehetnek.

Gyakori hiba: Debug builddel mérünk vagy optimalizálunk

# Általános beállítások

- Mi legyen a kimenet
- Makrók definiálása (-D)
- Include path (-I)
- Link path (-L)
- Linkelt libek (-l)
- Debug/release (-g)
- Optimizálás (-O)
- *Build eventek*

# Make

# Make

```
test: test.o test1.o  
    g++ -o test test.o test1.o
```

# Make

```
test: test.o test1.o  
    g++ -o test test.o test1.o
```

Ezt akarjuk létrehozni



# Make

```
test: test.o test1.o  
    g++ -o test test.o test1.o
```

Ezek kellenek hozzá

# Make

```
test: test.o test1.o  
    g++ -o test test.o test1.o
```

Így hozzuk létre

# Make

```
test.o: test.cpp header.h header2.h  
    g++ -c -o test.o test.cpp
```

```
test: test.o  
    g++ -o test test.o
```

A függőségnek is lehetnek függőségei

# Make

```
test.o: test.cpp header.h header2.h  
    g++ -c -o test.o test.cpp
```

```
test: test.o  
    g++ -o test test.o
```

A létező fájlókig megyünk le.

Ha valami újabb, akkor buildelődik a függőségi lánc. Újabb = később lett módosítva

# Make makrók

```
test: test.o test1.o  
    g++ -o $@ test.o test1.o
```

A programozó lusta

**\$@ :** A targetet helyettesíti

# Make makrók

```
test: test.o test1.o  
      g++ -o $@ $^
```

A programozó lusta

**\$^** : A legelső függőséget helyettesíti

# Make makrók

```
test: test.o test1.o  
    $(CXX) -o $@ $^
```

A programozó lusta

**\$(CXX)** Az alapértelmezett cpp fordító

**\$(CC)** Az alapértelmezett c fordító

sokat tud még

# Make makrók

```
test.o: test.cpp header.h header2.h  
$(CXX) -c -o $@ $<
```

A programozó lusta

**\$<** A legelső függőség. fordításnál hasznos.



# Make változók

```
HEADERS=header.h header2.h
```

```
test.o: test.cpp $(HEADERS)  
    $(CXX) -c -o $@ $<
```

Lehet változókat is létrehozni

# Make változók

```
HEADERS = header.h header2.h
```

```
OBJS = test.o test1.o
```

```
test.o: test.cpp $(HEADERS)
```

```
$(CXX) -c -o $@ $<
```

```
...
```

```
test: $(OBJS)
```

```
$(CXX) -o $@ $^
```

# Varázs %

```
HEADERS = header.h header2.h
```

```
OBJS = test.o test1.o
```

```
% .o: % .cpp $(HEADERS)  
    $(CXX) -c -o $@ $<
```

?: mintaillesztés

# Gyakori változók

**CFLAGS , CXXFLAGS**

**LDFLAGS**

**CXX**

A beépítetteket felül lehet írni

# Példa

**CXX=g++-4.7**

**CXXFLAGS = -std=c++11 -O3 -I\$(LIBDIR)**

**LDFLAGS=**

# Példa

**CXX=g++-4.7**

**CXXFLAGS = -std=c++11 -O3 -I\$(LIBDIR)**

**LDFLAGS=**

Más legyen a fordító

# Példa

CXX=g++-4.7

CXXFLAGS = -std=c++11 -O3 -I\$(LIBDIR)

LDFLAGS=

C++11 szabvány

# Példa

CXX=g++-4.7

CXXFLAGS = -std=c++11 -O3 -I\$(LIBDIR)

LDFLAGS=

## Optimalizálás



# Példa

CXX=g++-4.7

CXXFLAGS = -std=c++11 -O3 -I\$ (LIBDIR)

LDFLAGS=

Include path

# Példa

CXX=g++-4.7

CXXFLAGS = -std=c++11 -O3 -I\$(LIBDIR) -g

LDFLAGS= -g

Debug build

# Ninja - a gyors Make

- <http://martine.github.com/ninja/>
- elsőnek Chromium
- nagyon gyors (no-op 1s vs. 10s)
- minimalista eszközkészlet
- olvasható, de inkább ne írjuk (cmake)
- alapértelmezett párhuzamosítás

# Ninja - példa

```
build.ninja:
```

```
cflags = -Wall
```

```
rule cc
```

```
    command = gcc $cflags -c $in -o $out
```

```
build foo.o: cc foo.c
```

# cmake

# CMake

- <http://www.cmake.org/>
- meta-build rendszer
- rengeteg kimenet (generator)  
(Makefile, VS, Eclipse, Xcode, Ninja)
- fordítás forráson kívül
- egyszerű (könnyű és alacsony szintű)
- fordítás vezérlés: header, define, option
- további eszközök: CTest, CPack

# CMake - CMakeLists.txt

```
cmake_minimum_required (VERSION 2.8)
project(SoundStuff)
```

```
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -Wall -std=c++11")
```

```
set(SRCDIR ${PROJECT_SOURCE_DIR}/src)
set(LIBDIR ${PROJECT_SOURCE_DIR}/lib)
```

```
include_directories( ${LIBDIR} )
```

```
add_library(morse SHARED
    ${SRCDIR}/morsetext.cpp
    ${SRCDIR}/SoundSample.cpp
    ${LIBDIR}/fft/fft.cpp
    ${LIBDIR}/fft/complex.cpp
)
add_executable(SoundStuff ${SRCDIR}/main.cpp)
target_link_libraries(SoundStuff morse)
```

# CMake - Build

- CMakeLists.txt kerüljön a projekt gyökérbe
- Hozzunk létre egy build (al)könyvtárat
- `cd build`
- `cmake /a/CMakeLists.txt/helye`
- `make`

vagy

- `cmake -G "Visual Studio 11" /a/CMakeLists.txt/helye`
- VS



# Library-k

# Library-k

- Statikus: .a .lib  
Fordítás közben linkelődik  
Csak az linkelődik ami kell  
Minden program ami azt használja  
tartalmazni fogja a kódot
- Dinamikus: .so .dylib .dll  
Futás közben töltődik be.  
Több process is osztozhat ugyanazon.

# Lib buildelése - Linux

Object fájlok -> Egy archív fájl

ar: **archive**

```
ar rsv <libvalami.a> <valami1.o> <valami2.o>
```

...

r: hozzáad + felülír, ha kell

s: indexet csinál

v: verbose

# Makefile

```
$ (BINDIR) /libmorse.a: $(OBJS)  
    $(AR) rvs $@ $(OBJS)
```

# Makefile

```
$ (BINDIR) /libmorse.a: $(OBJS)  
    $(AR) rvs $@ $(OBJS)
```

Ezt is tudja beépíttetten

# Lib buildelése - Linux

.so buildelés:

Fordításhoz -fPIC

Position Independent Code

Linkeléshez:

-shared

# Makefile

```
$(BINDIR)/libmorse.so: CXXFLAGS+= -fPIC
$(BINDIR)/libmorse.so: $(OBJS)
    $(CXX) -shared -o $@ $^
```

# Makefile

```
$ (BINDIR) /libmorse.so: CXXFLAGS+= -fPIC  
$ (BINDIR) /libmorse.so: $ (OBJS)  
    $ (CXX) -shared -o $@ $^
```

Target specifikus változó



# Makefile

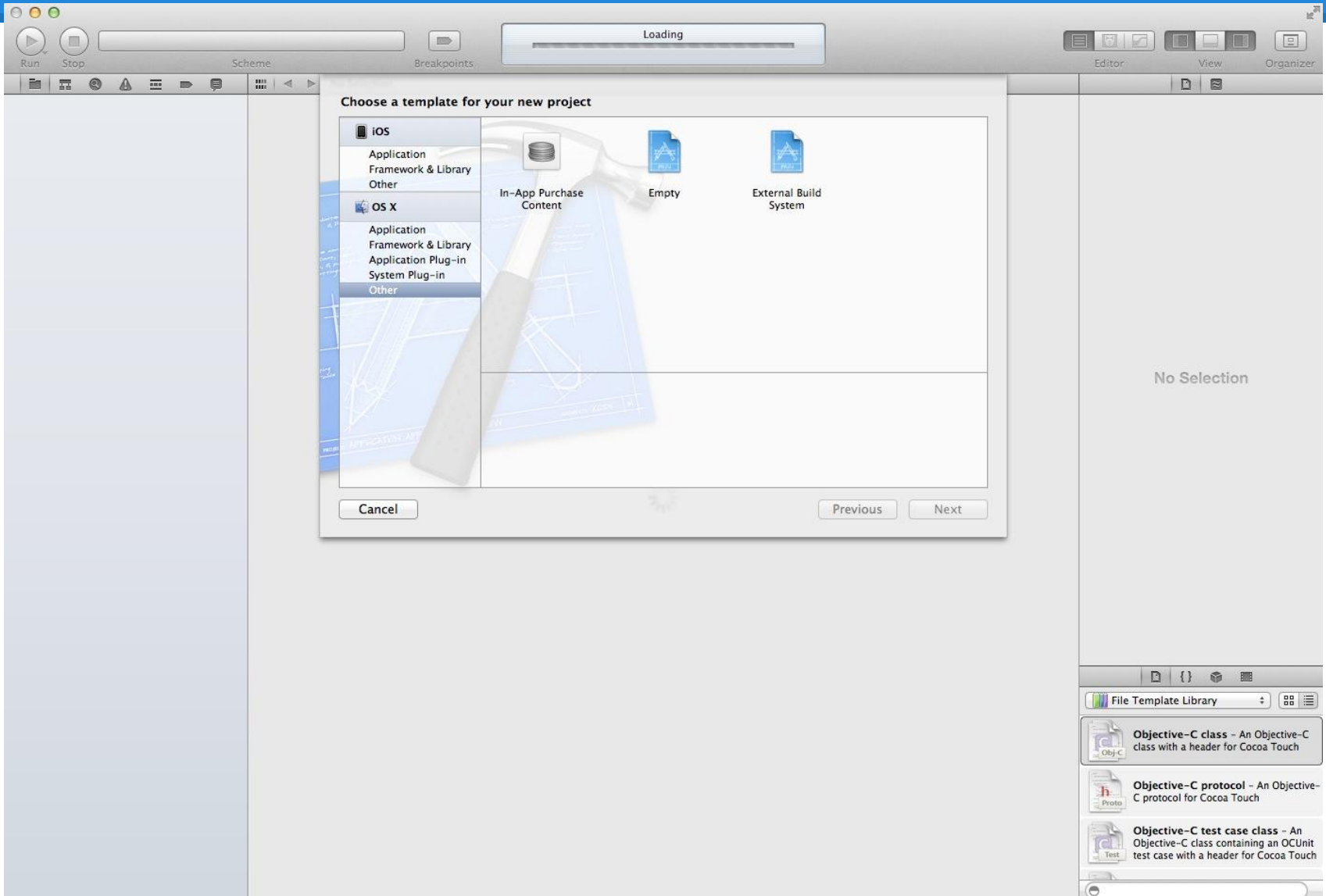
```
$ (BINDIR) /libmorse.so: CXXFLAGS+= -fPIC  
$ (BINDIR) /libmorse.so: $ (OBJS)  
    $ (CXX) -shared -o $@ $^
```

Az so buildeléshez használt kapcsolók

# Lib buildelése - Windows

# XCode

# XCode - SoundStuff



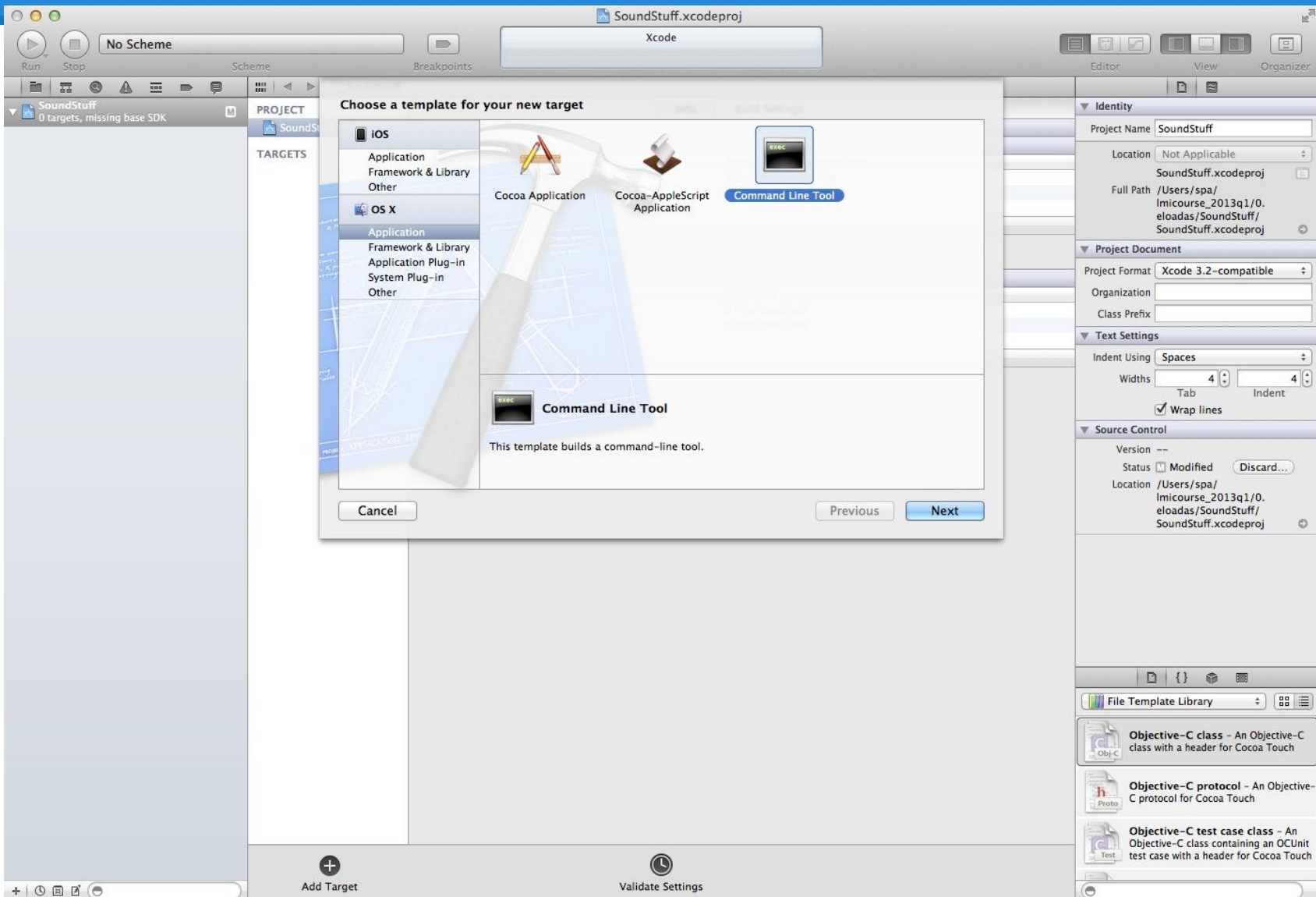
Choose options for your new project:

Product Name SoundStuff

Cancel

Previous

Next



**Choose options for adding these files**

Destination ☐ Copy items into destination group's folder (if needed)

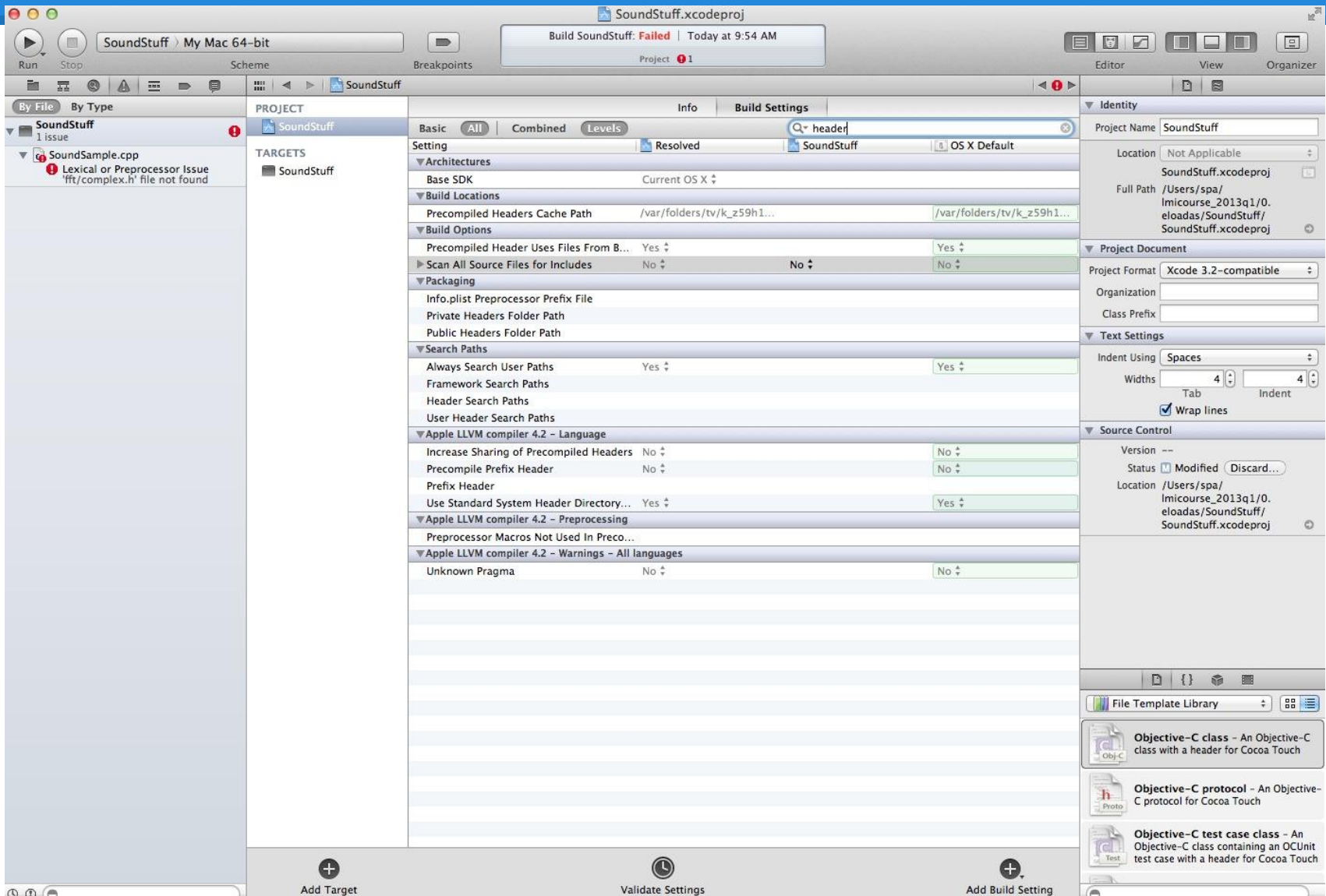
Folders ☒ Create groups for any added folders

☐ Create folder references for any added folders

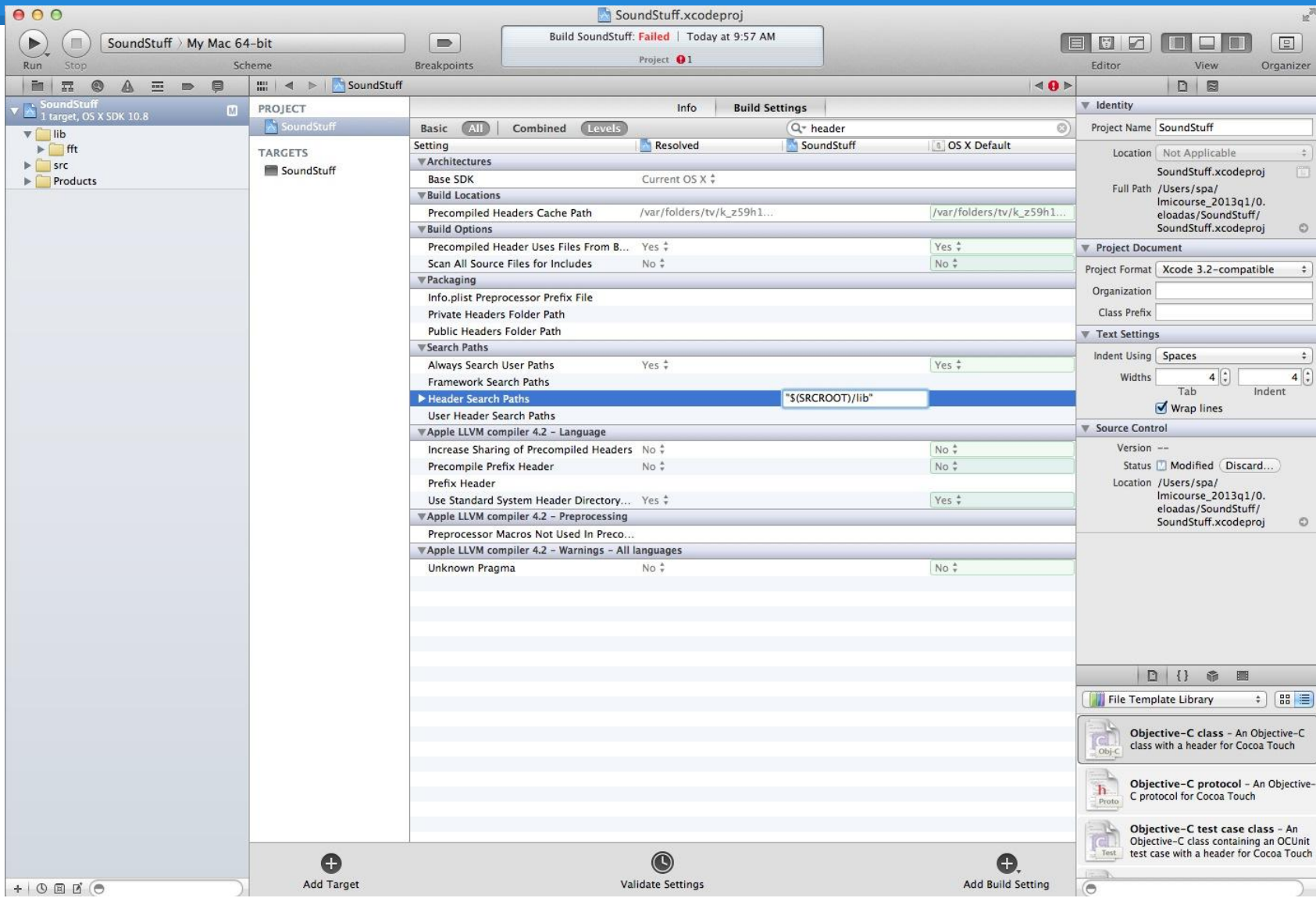
Add to targets ☒ SoundStuff

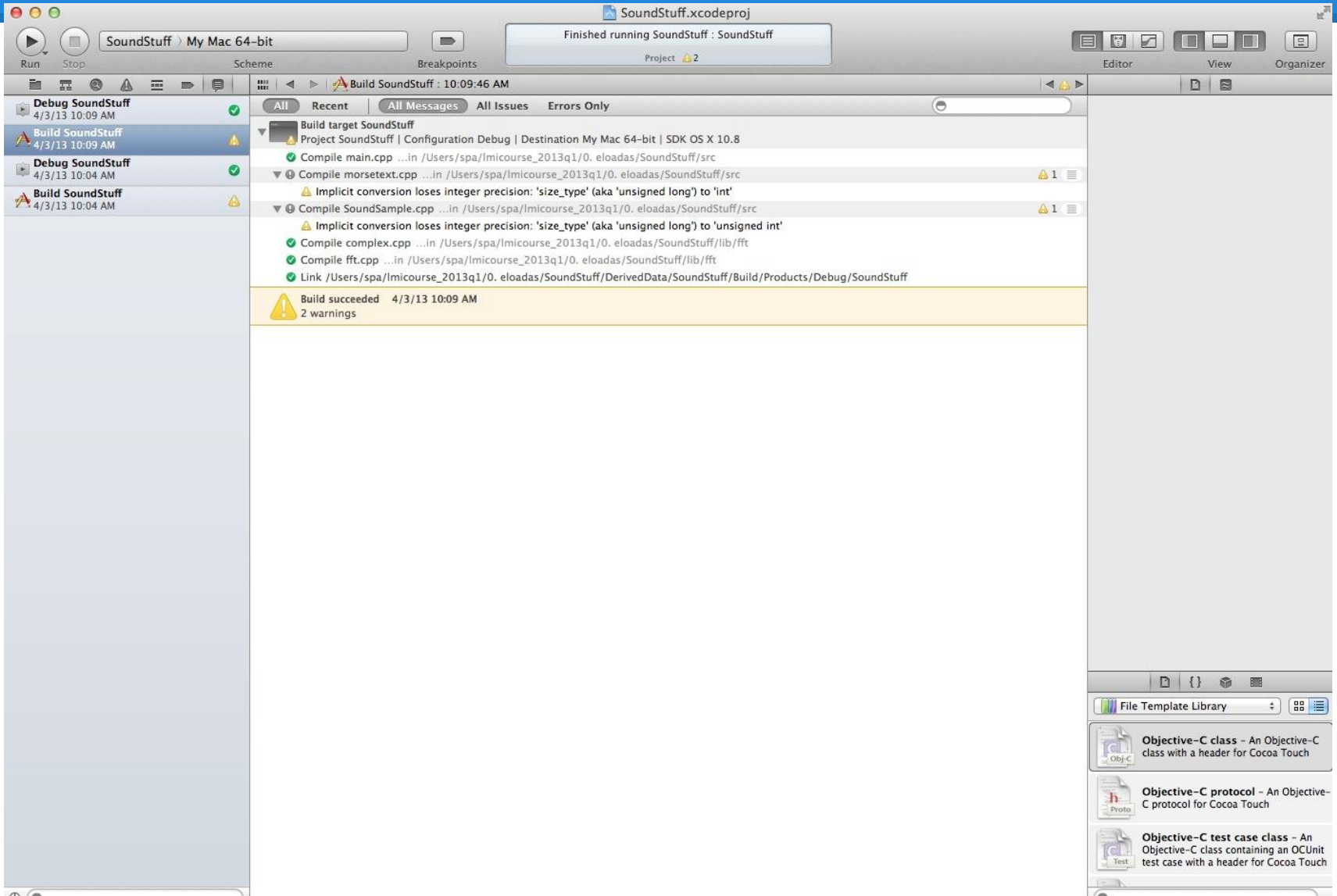
Cancel

Finish







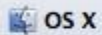


## Choose a template for your new target



iOS

Application  
Framework & Library  
Other



OS X

Application  
Framework & Library  
Application Plug-in  
System Plug-in  
Other



Cocoa Framework



Cocoa Library



Bundle



XPC Service



C/C++ Library



STL C++ Library



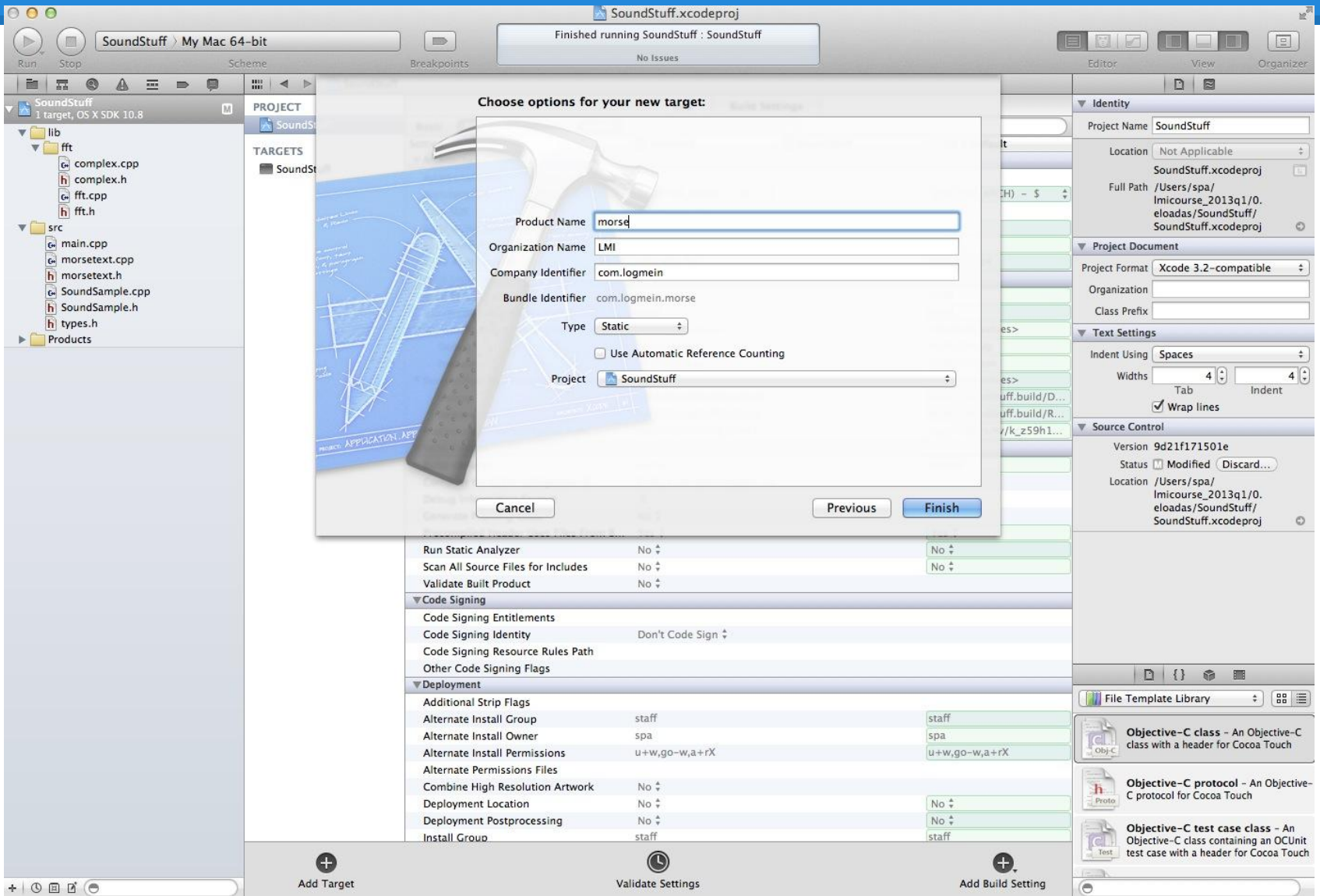
C/C++ Library

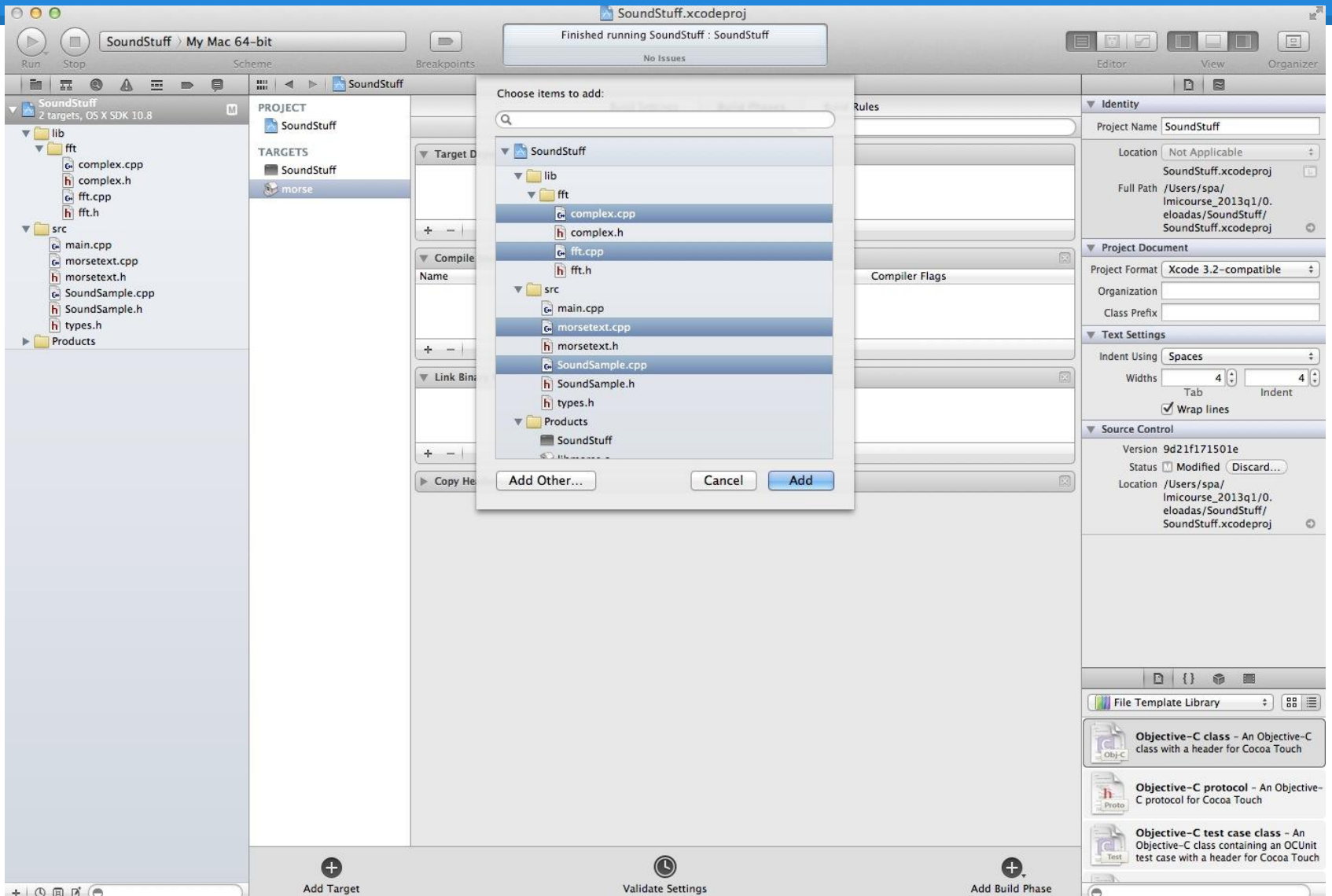
This template builds a library written in C or C++.

Cancel

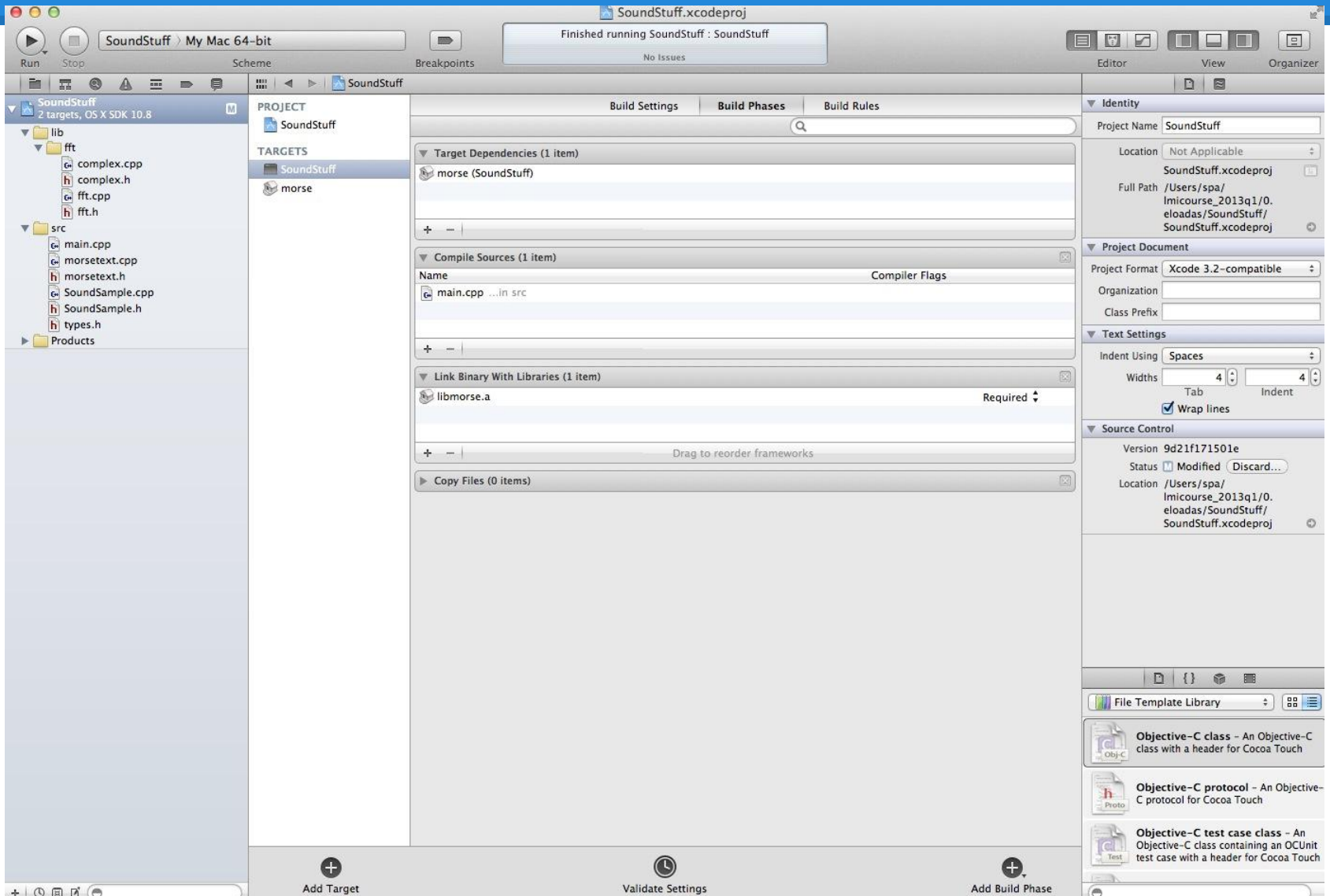
Previous

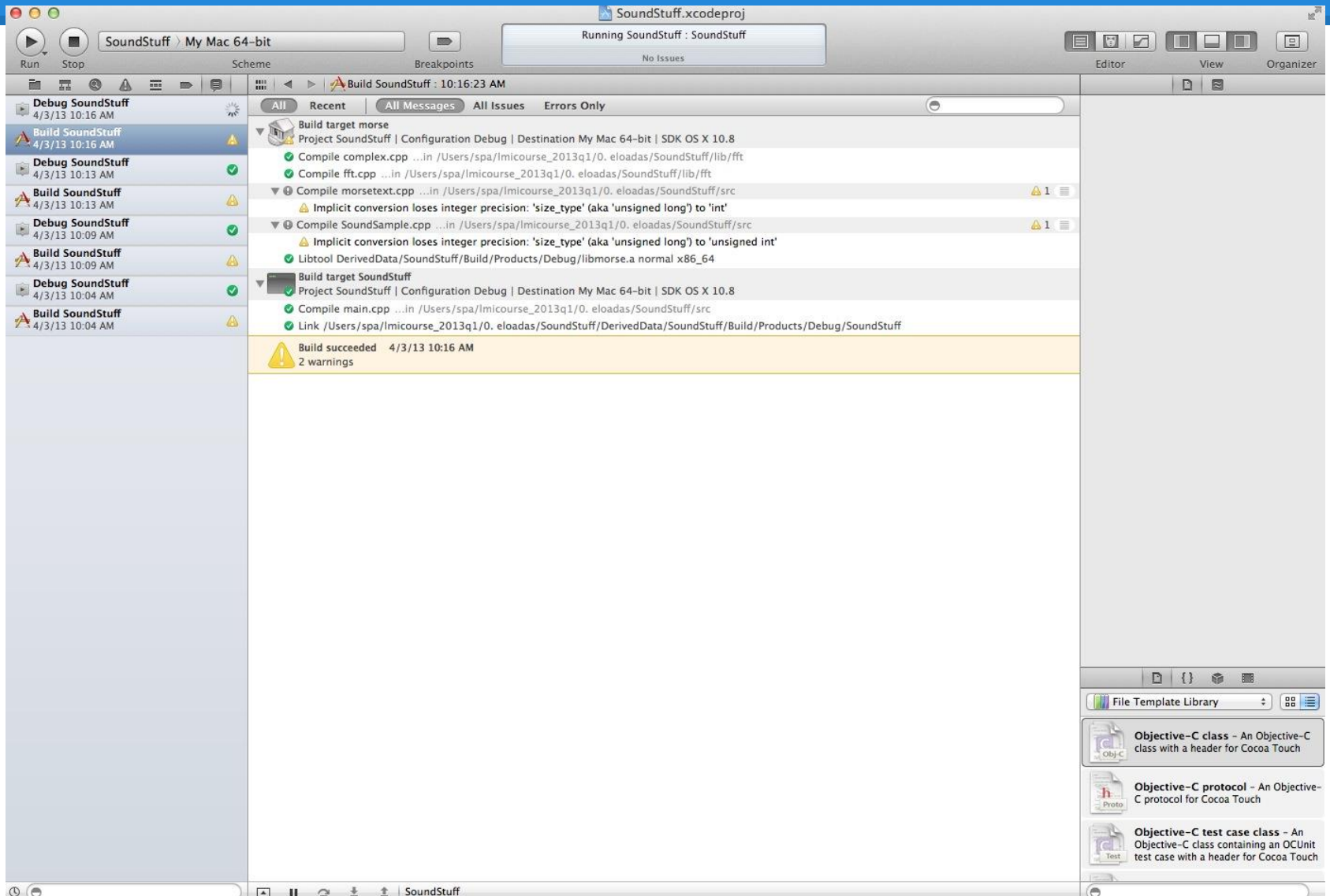
Next











# Eclipse windowson

MinGW:

<http://sourceforge.net/projects/mingw/>  
(latest catalog)

java:

<http://www.java.com/en/download/index.jsp>

Eclipse:

[http://www.eclipse.org/downloads/download.  
php?](http://www.eclipse.org/downloads/download.php?)

[file=/technology/enp/downloads/release/juno/S](http://www.eclipse.org/downloads/download.php?file=/technology/enp/downloads/release/juno/S)



