

# Xcode 9 Custom Templates



Dmytro Cheverda  
Mar 1, 2018 · 4 min read

We as a developers constantly trying to improve our development workflow. Whether it is related to writing code, automating some things or just staying more productive and focused. Here we will talk about how we can improve creation of files using Xcode custom templates. Many of us may not have heard about it because Apple is not promoting this functionality and there is not much documentation on this topic. But such functionality exists anyways and may drastically improve files or whole modules creation with some pre-populated boilerplate code.

As an example, we will setup `UIView` template that also will create a corresponding `.xib` file, which Xcode can't create by default.

## Location

These user-defined templates are located in `~/Library/Developer/Xcode/Templates/File Template`. If such folder is missing you can create it yourself and Xcode will be linked to it upon reopening. Also you can group them into subfolders inside that folder. For example, `~/Library/Developer/Xcode/Templates/File Template/Custom Templates` and Xcode will render those groups properly when creating a new file.

## Template Setup

Each template is a folder with `.xctemplate` extension. That folder contains file templates and resources that will be populated and added to a project and configuration `TemplateInfo.plist`.

All template settings and fields that user can fill in during template creation are defined in `TemplateInfo.plist`.

Key	Type	Value
▼ Information Property List	Dictionary	(3 items)
Icon	String	Templatelcon
Kind	String	Xcode.IDEKit.TextSubstitutionFileTemplateKind
▼ Options	Array	(2 items)
▼ Item 0	Dictionary	(6 items)
Identifier	String	subclassesFrom
Required	Boolean	YES
Name	String	Subclass of:
Description	String	Sublasses From
Type	String	text
Default	String	UIView
▼ Item 1	Dictionary	(6 items)
Identifier	String	productName

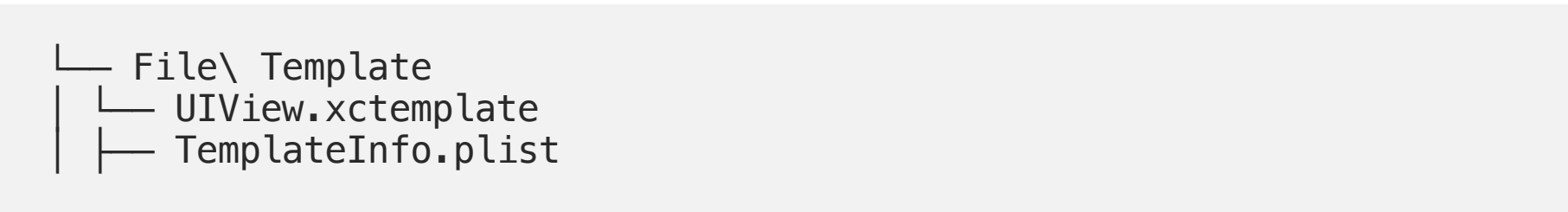
Required	Boolean	YES
Name	String	View:
Description	String	The name of the UIView to create
Type	String	text
Default	String	MyModel

TemplateInfo.plist

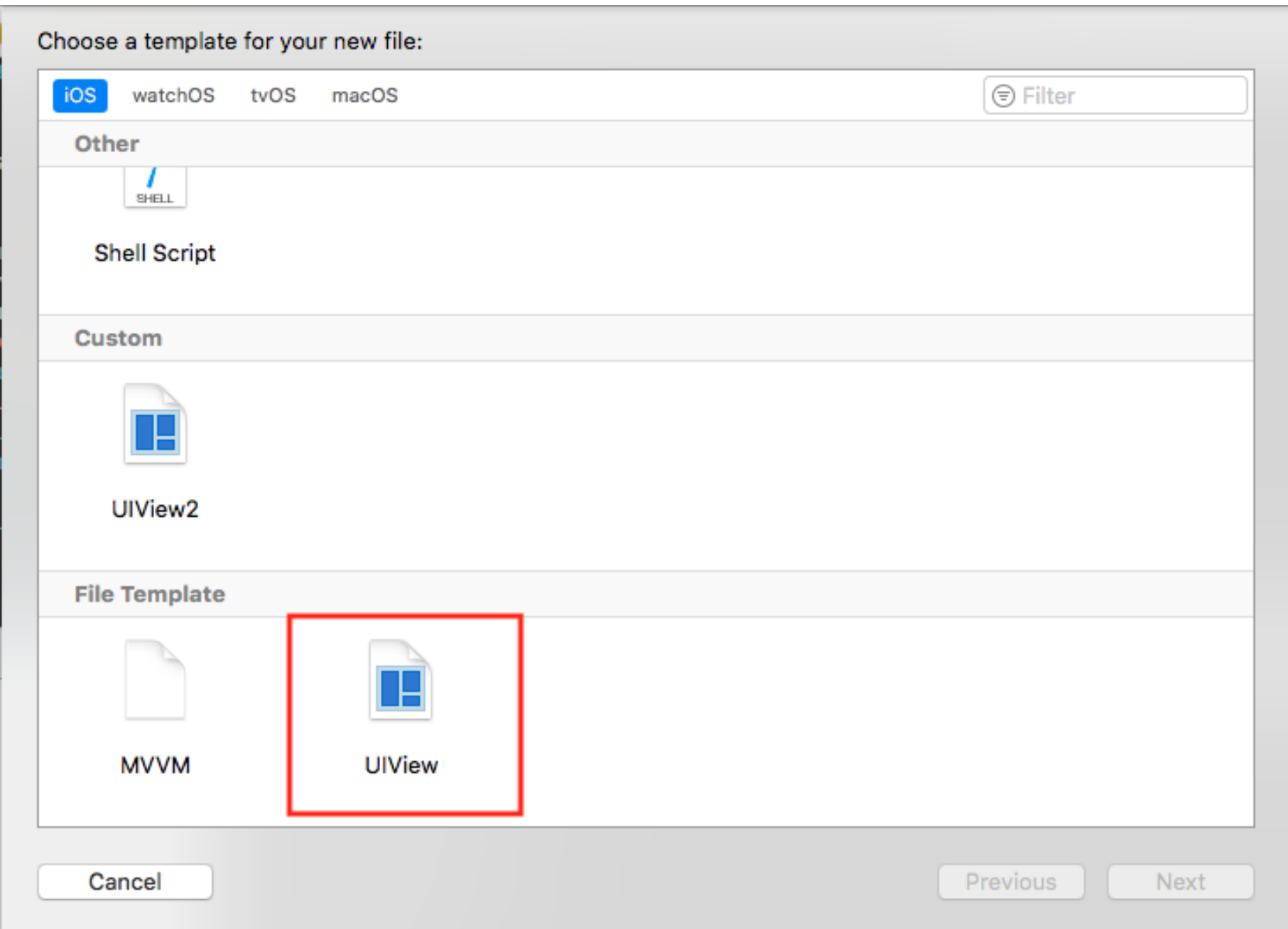
In this plist file you can define Icon that will be used for template selection dialog. Kind of template can have two values, but most of times it will be `Xcode.IDEKit.TextSubstitutionFileTemplateKind`, another one is `Xcode.IDECoreDataModeler.ManagedObjectTemplateKind` and we won't cover it's usage here.

Next item is `Options` array which can contain various input fields, pickers, checkboxes, etc. Each field is pretty self-explanatory, just note that `Identifier` field is used for user-provided variables that can be substituted in source code template. The `Type` defines a type of input field. Can be `text`, `static`, `checkbox`, `combo`, `popup`.

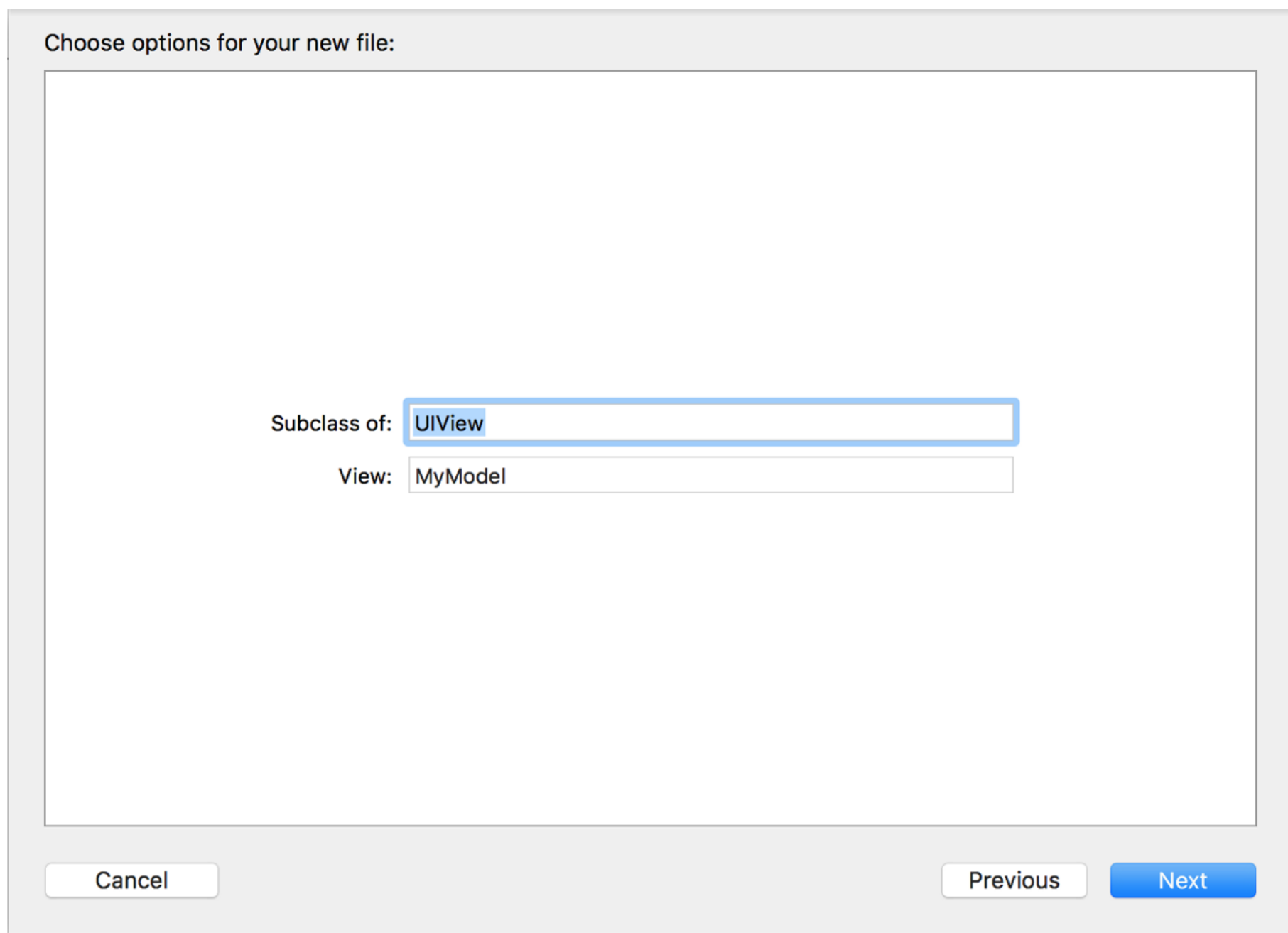
Let's place `TemplateInfo.plist` file into `UIView.xctemplate` folder under user-defined templates. Files structure should look like this:



And after restarting Xcode you should see your template listed like this:



And after selecting template menu should look like this:



Choose options for your new file:

Subclass of:

View:

Cancel Previous Next

Now, let's look into file templates. Basically, they are usual files that have predefined keywords that will be replaced with user-provided values during creation. The main keywords are:

- `___FILEBASENAME___` — The name of the current file without any extension.
- `___FILEHEADER___` — IDE predefined comments header for each source file.
- `___VARIABLE_variable_name___` — a custom variable that user inputs during template creation and can be used later in template files, as per example we have *Subclass of:* input field which used define inheritance for our `UIView`.

Let's create file named `___FILEBASENAME___.swift` and make it look like this:

```
// ___FILEHEADER___  
  
import UIKit  
  
class ___FILEBASENAME___: ___VARIABLE_subclassesFrom___ {  
}
```

Also, create `___FILEBASENAME___.xib` it's content can be pasted from general newly created `xib` file except make sure root view configuration line looks something like this,

main point is to set `customClass` keyword to link source file to `xib` file.

```
<view contentMode="scaleToFill" id="iN0-l3-epB"
customClass="__FILEBASENAME__" customModule="someModule"
customModuleProvider="target">
```

After this quick setup you should be able to create `UIView` source file along with `xib` file that are ready for use.

## One more thing

You can reverse engineer Xcode default templates and see how to improve custom templates for better usage. They are located in

`Xcode.app/Contents/Developer/Library/Xcode/Templates` or

`Xcode.app/Contents/Developer/Platforms/<platform_name>/Developer/Library/Xcode/Templates` . Also you can place your own templates directly into Xcode to avoid scrolling to the bottom, but those files may get overwritten with Xcode update.

## Conclusion

Xcode templates may be especially useful during early stages of a project and you have to create a lot of files and modules. The common templates may be used for MVVM or VIPER modules which require a lot of files to be created at once.

The final template can be found [here](#).

## Useful links

[Xcode Text macros reference](#)

[Customizing the file header comment and other text macros in Xcode 9 — Ole Begemann](#)

[Xcode 9 File Templates | They call me Jet](#)

<https://littlebitesofcocoa.com/90-custom-xcode-project-templates>

[Creating Custom Xcode Project Templates](#)

iOS

Xcode

Xcode 9

Template

Developer Tools

Get the Medium app

