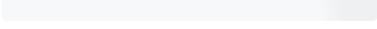
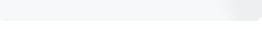
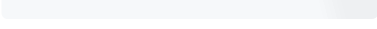
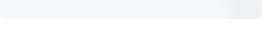
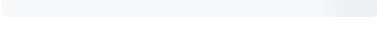
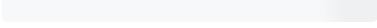
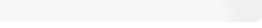
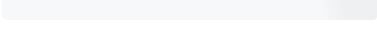
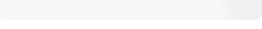
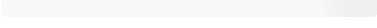
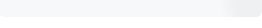
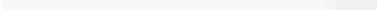
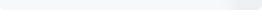
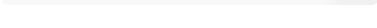
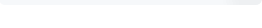
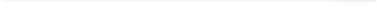
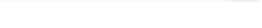
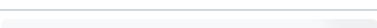
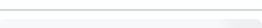
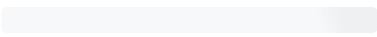
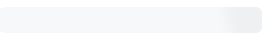
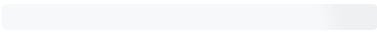
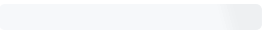
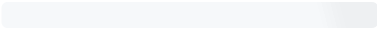
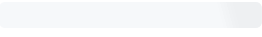
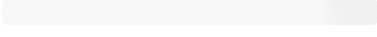
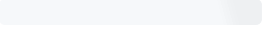
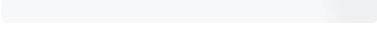
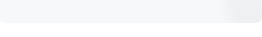
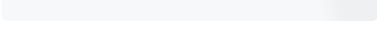
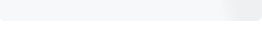
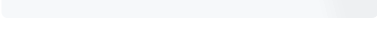
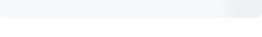
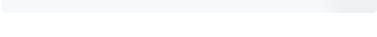
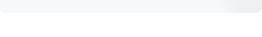
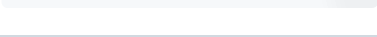
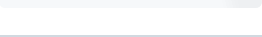
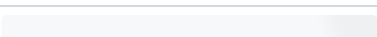
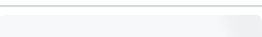
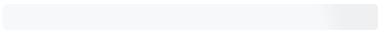
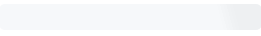
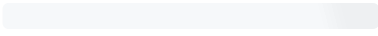
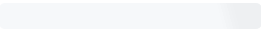
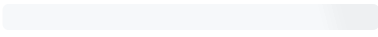
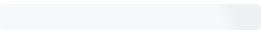
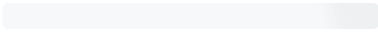
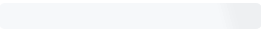
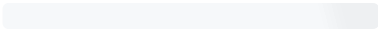
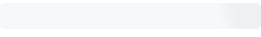
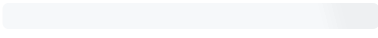
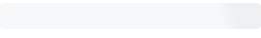
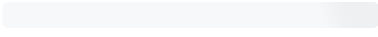
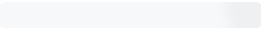
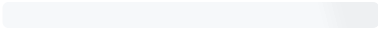
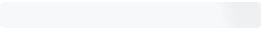
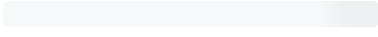
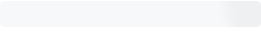
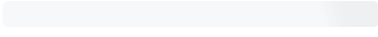
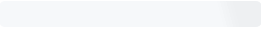
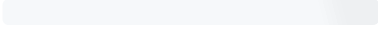
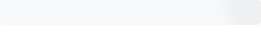
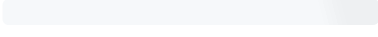
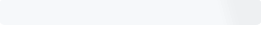
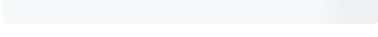
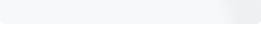
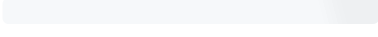
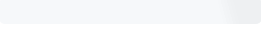
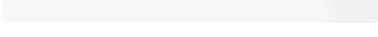
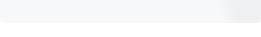
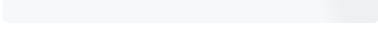
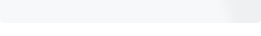
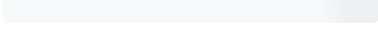
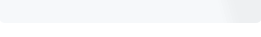
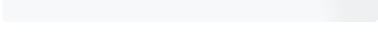
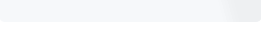
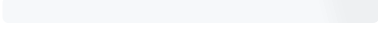
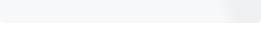
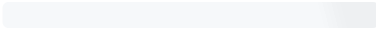
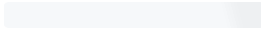
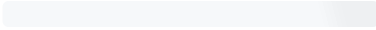
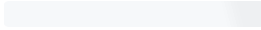
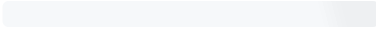
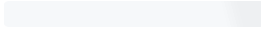
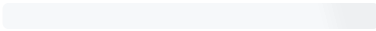
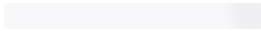
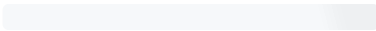
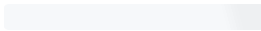
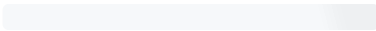
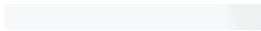
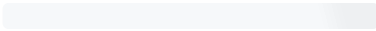
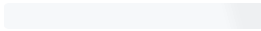


.github		
Global		
community		
AL.gitignore		
Actionscript.gitignore		
Ada.gitignore		
AdventureGameStud...		
Agda.gitignore		
Android.gitignore		
Angular.gitignore		
AppEngine.gitignore		
AppceleratorTitanu...		
ArchLinuxPackages....		
Autotools.gitignore		
Ballerina.gitignore		
C++.gitignore		
C.gitignore		
CFWheels.gitignore		

 CMake.gitignore		
 CONTRIBUTING.md		
 CUDA.gitignore		
 CakePHP.gitignore		
 ChefCookbook.gitign...		
 Clojure.gitignore		
 CodeIgniter.gitignore		
 CommonLisp.gitignore		
 Composer.gitignore		
 Concrete5.gitignore		
 Coq.gitignore		
 CraftCMS.gitignore		
 D.gitignore		
 DM.gitignore		
 Dart.gitignore		
 Delphi.gitignore		
 Deno.gitignore		
 Dotnet.gitignore		
 Drupal.gitignore		
 EPiServer.gitignore		
 Eagle.gitignore		
 Elisp.gitignore		
 Elixir.gitignore		
 Elm.gitignore		

 Erlang.gitignore		
 ExpressionEngine.git...		
 ExtJs.gitignore		
 Fancy.gitignore		
 Finale.gitignore		
 Firebase.gitignore		
 FlaxEngine.gitignore		
 Flutter.gitignore		
 ForceDotCom.gitignore		
 Fortran.gitignore		
 FuelPHP.gitignore		
 GWT.gitignore		
 Gcov.gitignore		
 GitBook.gitignore		
 GitHubPages.gitignore		
 Gleam.gitignore		
 Go.gitignore		
 Godot.gitignore		
 Gradle.gitignore		
 Grails.gitignore		
 HIP.gitignore		
 Haskell.gitignore		
 Haxe.gitignore		
 IAR.gitignore		

 IGORPro.gitignore		
 Idris.gitignore		
 JBoss.gitignore		
 JENKINS_HOME.giti...		
 Java.gitignore		
 Jekyll.gitignore		
 Joomla.gitignore		
 Julia.gitignore		
 Katalon.gitignore		
 KiCad.gitignore		
 Kohana.gitignore		
 Kotlin.gitignore		
 LICENSE		
 LabVIEW.gitignore		
 LangChain.gitignore		
 Laravel.gitignore		
 Lasal.gitignore		
 Lean.gitignore		
 Leiningen.gitignore		
 LemonStand.gitignore		
 Lilypond.gitignore		
 Lithium.gitignore		
 Lua.gitignore		
 Luau.gitignore		

 Magento.gitignore		
 Maven.gitignore		
 Mercury.gitignore		
 MetaProgrammingSy...		
 ModelSim.gitignore		
 Modelica.gitignore		
 MoonBit.gitignore		
 Nanoc.gitignore		
 Nestjs.gitignore		
 Nextjs.gitignore		

[README](#)[Code of conduct](#)[Contributing](#)[CC0-1.0 license](#)[Security](#)

A collection of `.gitignore` templates

This is GitHub's collection of `.gitignore` file templates. We use this list to populate the `.gitignore` template choosers available in the GitHub.com interface when creating new repositories and files.

For more information about how `.gitignore` files work, and how to use them, the following resources are a great place to start:

- The [Ignoring Files chapter](#) of the [Pro Git](#) book.
- The [Ignoring Files article](#) on the GitHub Help site.
- The [gitignore\(5\)](#) manual page.

Folder structure

We support a collection of templates, organized in this way:

- The root folder contains templates in common use, to help people get started with popular programming languages and technologies. These define a meaningful set of rules to help get started, and ensure you are not committing unimportant files into your repository.
- [Global](#) contains templates for various editors, tools and operating systems that can be used in different situations. It is recommended that you either [add these to your global template](#) or merge these rules into your project-specific templates if you want to use them permanently.
- [community](#) contains specialized templates for other popular languages, tools, and projects that don't currently belong in the mainstream templates. These should be added to your project-specific templates when you decide to adopt the framework or tool.

What makes a good template?

First and foremost, a template contribution must adhere to our [Contributing Guidelines](#).

A template should contain a set of rules to help Git repositories work with a specific programming language, framework, tool or environment.

If it's not possible to curate a small set of useful rules for this situation, then the template is not a good fit for this collection.

If a template is mostly a list of files installed by a particular version of some software (e.g. a PHP framework), it could live under the [community](#) directory. See [versioned templates](#) for more details.

If you have a small set of rules, or want to support a technology that is not widely in use, and still believe this will be helpful to others, please read the section about [specialized templates](#) for more details.

Include details when opening a pull request if the template is important and visible. We may not accept it immediately, but we can promote it to the root at a later date based on interest.

Please also understand that we can't list every tool that ever existed. Our aim is to curate a collection of the *most common and helpful* templates, not to make sure we cover every project possible. If we choose not to include your language, tool, or project, it's not because it's not awesome.

Contributing guidelines

Please see our [Contributing Guidelines](#).

Versioned templates

Some templates can change greatly between versions, and if you wish to contribute to this repository we need to follow this specific flow:

- the template at the root should be the current supported version
- the template at the root should not have a version in the filename (i.e. "evergreen")
- previous versions of templates should live under `community/`
- previous versions of the template should embed the version in the filename, for readability

This helps ensure users get the latest version (because they'll use whatever is at the root) but helps maintainers support older versions still in the wild.

Specialized templates

If you have a template that you would like to contribute, but it isn't quite mainstream, please consider adding this to the `community` directory under a folder that best suits where it belongs.

The rules in your specialized template should be specific to the framework or tool, and any additional templates should be mentioned in a comment in the header of the template.

For example, this template might live at `community/DotNet/InforCRM.gitignore` :

```
# gitignore template for InforCRM (formerly SalesLogix)
# website: https://www.infor.com/product-summary/cx/infor-crm/
#
# Recommended: VisualStudio.gitignore

# Ignore model files that are auto-generated
ModelIndex.xml
ExportedFiles.xml

# Ignore deployment files
[Mm]odel/[Dd]eployment

# Force include portal SupportFiles
```



```
!Model/Portal/*/SupportFiles/[Bb]in/  
!Model/Portal/PortalTemplates/*/SupportFiles/[Bb]in
```

Contributing workflow

Here's how we suggest you go about proposing a change to this project:

1. [Fork this project](#) to your account.
2. [Create a branch](#) for the change you intend to make.
3. Make your changes to your fork.
4. [Send a pull request](#) from your fork's branch to our `main` branch.

Using the web-based interface to make changes is fine too, and will help you by automatically forking the project and prompting to send a pull request too.

Releases

Packages

Used by

Contributors