

 toddewilliams docs: update legality of vscode marketplace's usage (#1092) ✓

14 contributors

More Info

Table of Contents

- Getting all the Telemetry Out
 - Replacements to Microsoft Online Services
- Extensions + Marketplace
 - How to use the VS Code Marketplace
 - Proprietary Debugging Tools
 - Proprietary Extensions
- Migrating from Visual Studio Code to VSCodium
- Sign in with GitHub
- How do I run VSCodium in portable mode?
- How do I fix the default file manager?
- How do I press and hold a key and have it repeat in VSCodium?
- How do I open VSCodium from the terminal?
 - From Linux .tar.gz
- How to build VSCodium

Getting all the Telemetry Out

Even though we do not pass the telemetry build flags (and go out of our way to cripple the baked-in telemetry), Microsoft will still track usage by default.

We do however set the default `telemetry.enableCrashReporter` and `telemetry.enableTelemetry` values to false. You can see those by viewing your VSCode settings.json and searching for `telemetry`.

The instructions [here](#) and [here](#) help with explaining and toggling telemetry.

It is also highly recommended that you review all the settings that "use online services" by following [these instructions](#). The `@tag:usesOnlineServices` filter on the settings page will show that by default:

- Extensions auto check for updates and auto install updates
- Searches within the app are sent to an online service for "natural language processing"
- Updates to the app are fetched in the background

These can all be disabled.

Please note that some extensions send telemetry data to Microsoft as well. We have no control over this and can only recommend removing the extension. (For example, the C# extension `ms-vscode.csharp` sends tracking data to Microsoft.)

Replacements to Microsoft Online Services

When searching the `@tag:usesOnlineServices` filter, note that while the "Update: Mode" setting description still says "The updates are fetched from a Microsoft online service", VSCode's build script [sets the `updateUrl` field](#) in `product.json` to that of VSCode's own small [update server](#), so enabling that setting won't actually result in any calls to Microsoft servers.

Likewise, while the descriptions for "Extensions: Auto Check Updates" and "Extensions: Auto Update" include the same phrase, VSCode [replaces](#) the Visual Studio Marketplace with Open VSX, so these settings won't call Microsoft, either.

Extensions + Marketplace

The `product.json` file is set up to use [open-vsx.org](#) as extension gallery, which has an [adapter](#) to the Marketplace API used by VS Code. Since that is a rather new project, you will likely miss some extensions you know from the VS Code Marketplace. You have the following options to obtain such missing extensions:

- Ask the extension maintainers to publish to [open-vsx.org](#) in addition to the VS Code Marketplace. The publishing process is documented in the [Open VSX Wiki](#).

- Создайте запрос на извлечение в [этот репозиторий](#), чтобы учетная запись службы [@open-vsx](#) опубликовала расширения для вас.
- Загрузите и [установите файлы vsix](#)

☰ 188 строк (129 строк) | 11.4 КБ

...

VSX.

Как использовать VS Code Marketplace

Вы можете переключиться и использовать VS Code marketplace, используя следующие решения. Однако обратите внимание, что [это незаконно](#).

Со следующими переменными среды:

- `VSCODE_GALLERY_SERVICE_URL='https://marketplace.visualstudio.com/_apis/public/gallery'`
- `VSCODE_GALLERY_CACHE_URL='https://vscode.blob.core.windows.net/gallery/index'`
- `VSCODE_GALLERY_ITEM_URL='https://marketplace.visualstudio.com/items'`
- `VSCODE_GALLERY_CONTROL_URL=''`
- `VSCODE_GALLERY_RECOMMENDATIONS_URL=''`

Или путем создания пользовательского `product.json` в следующем расположении:

- Windows: `%APPDATA%\VSCodium` или `%USERPROFILE%\AppData\Roaming\VSCodium`
- macOS: `~/Library/Application Support/VSCodium`
- Linux: `$XDG_CONFIG_HOME/VSCodium` или `~/.config/VSCodium`

с содержанием:

```
{
  "extensionsGallery": {
    "serviceUrl": "https://marketplace.visualstudio.com/_apis/public/gallery ",
    "cacheUrl": "https://vscode.blob.core.windows.net/gallery/index ",
    "itemUrl": "https://marketplace.visualstudio.com/items ",
    "controlUrl": "",
    "recommendationsUrl": ""
  }
}
```



Собственные средства отладки

Отладчик, поставляемый с [расширением Microsoft C #](#), а также отладчик (Windows), поставляемый с [расширением C ++](#), имеют очень ограниченную лицензию на работу только с официальной сборкой кода Visual Studio. Смотрите [этот комментарий в репозитории расширений C #](#) и [этот комментарий в репозитории расширений C ++](#).

Существует обходной путь для обеспечения работы отладки в проектах C # с использованием пакета netcoredbg от Samsung с открытым исходным кодом. Смотрите [Этот комментарий](#) для получения инструкций о том, как это настроить.

Собственные расширения

Как и упомянутые выше отладчики, некоторые расширения, которые вы можете найти в marketplace (например, [расширения для удаленной разработки](#)), работают только с официальной сборкой кода Visual Studio. Вы можете обойти это, добавив внутренний идентификатор расширения (найденный на странице расширения) в `extensionAllowedProposedApi` свойство `product.json` в вашей установке VSCodium. Например:

```
"extensionAllowedProposedApi": [  
  // ...  
  "ms-vscode-remote.vscode-remote-extensionpack",  
  "ms-vscode-remote.remote-wsl",  
  // ...  
],
```

В некоторых случаях вышеуказанное изменение не поможет, поскольку расширение жестко запрограммировано для работы только с официальным продуктом Visual Studio Code.

Переход с кода Visual Studio на VSCodium

VSCodium (и свежая клонированная копия `vscode`, созданная из исходного кода) хранит свои файлы расширений `~/.vscode-oss`. Поэтому, если у вас в настоящее время установлен Visual Studio Code, ваши расширения не будут заполняться автоматически. Вы можете переустановить свои расширения из Marketplace в VSCodium или скопировать `extensions` из `~/.vscode/extensions` в `~/.vscode-oss/extensions`.

Visual Studio Code хранит свои `keybindings.json` `settings.json` файлы и в этих расположениях:

- **Windows:** `%APPDATA%\Code\User`
- **macOS:** `$HOME/Library/Application Support/Code/User`

- **Linux:** `$HOME/.config/Code/User`

Вы можете скопировать эти файлы в папку настроек пользователя VSCodium:

- **Windows:** `%APPDATA%\VSCodium\User`
- **macOS:** `$HOME/Library/Application Support/VSCodium/User`
- **Linux:** `$HOME/.config/VSCodium/User`

Чтобы скопировать настройки вручную:

- В коде Visual Studio перейдите в Настройки (Command+, если на Mac).
- Щелкните три точки ... и выберите "Открыть settings.json"
- Скопируйте содержимое settings.json в то же место в VSCodium

Войдите в систему с помощью GitHub

В VSCodium sign in with GitHub используется личный токен доступа.

Следуйте документации <https://docs.github.com/en/github/authenticating-to-github/creating-a-personal-access-token> чтобы создать свой токен.

Выберите области, зависящие от расширения, которым требуется доступ к GitHub. (Для GitLens требуется repo область действия.)

Linux

Если вы получаете сообщение об ошибке `writing login information to the keychain failed with error 'The name org.freedesktop.secrets was not provided by any .service files'.`, вам необходимо установить пакет `gnome-keyring`.

Как мне запустить VSCodium в переносимом режиме?

Вы можете следовать [инструкциям по переносному режиму](#) с веб-сайта Visual Studio Code.

- **Windows / Linux:** инструкции можно выполнять в том виде, в каком они написаны.
- **macOS :** режим переносимости включается при наличии папки со специальным именем. Для кода Visual Studio это имя папки `code-portable-data`. Для VSCodium это имя папки `codium-portable-data`. Итак, чтобы включить портативный режим для VSCodium в Mac OS, следуйте инструкциям, изложенным по [ссылке выше](#), но создайте папку с именем `codium-portable-data` ВМЕСТО `code-portable-data`.

Как мне исправить файловый менеджер по умолчанию (Linux)?

В некоторых случаях VSCodium становится файловым менеджером, используемым для открытия каталогов (вместо приложений, таких как Dolphin или Nautilus).

Это связано с тем, что ни одно приложение не было определено в качестве файлового менеджера по умолчанию, и поэтому система использует последнее работоспособное приложение.

Чтобы установить приложение по умолчанию, создайте файл `~/.config/mimeapps.list` с содержимым, подобным:

```
[Default Applications]
inode/directory=org.gnome.Nautilus.desktop;
```

Вы можете найти свой обычный файловый менеджер с помощью команды:

```
> grep directory /usr/share/applications/mimeinfo.cache
inode/directory=codium.desktop;org.gnome.Nautilus.desktop;
```

Как мне нажать и удерживать клавишу и повторить ее в VSCodium (Mac)?

Это общий вопрос для кода Visual Studio, и процедура в VSCodium немного отличается, потому что `defaults` что путь отличается.

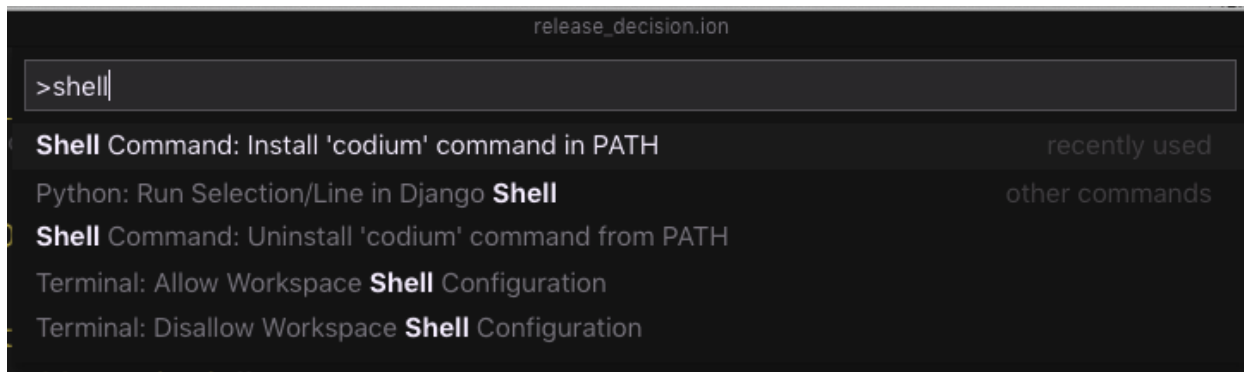
```
$ defaults написать com.visualstudio.code.oss ApplePressAndHoldEnabled -bool
```



Как мне открыть VSCodium с терминала?

Для macOS и Windows:

- Перейдите в палитру команд (Вид | Палитра команд ...)
- Выбирай `Shell command: Install 'codium' command in PATH`.



Это позволяет открывать файлы или каталоги в VSCodium непосредственно с вашего терминала:

```
~/ в моем проекте $ codium . # открыть этот каталог
~/в моем проекте $ codium file.txt # открыть этот файл
```

Не стесняйтесь использовать псевдоним этой команды для чего-то более простого для ввода в вашем профиле оболочки (например `alias code=codium`).

В Linux, при установке с помощью менеджера пакетов, `codium` был установлен в вашем `PATH`.

Из Linux.tar.gz

Когда архив `VSCodium-linux-<arch>-<version>.tar.gz` извлекается, основной точкой входа для VSCodium является `./bin/codium`.