

Теперь доступна версия 1.78 (/updates)! Прочитайте о новых функциях и исправлениях от апреля.



Темы

Linux



В этой статье

Установка

<https://vscode.dev/github/microsoft/vscode-docs/blob/main/docs/setup/linux.md>

Код Visual Studio в Linux

Установка

Смотрите страницу Загрузки кода Visual Studio (/download) для получения полного списка доступных вариантов установки.

Загружая и используя Visual Studio Code, вы соглашаетесь с условиями лицензии (<https://code.visualstudio.com/license>) и заявлением о конфиденциальности (<https://go.microsoft.com/fwlink/?LinkID=528096&clid=0x409>).

Дистрибутивы на основе Debian и Ubuntu

Самый простой способ установить код Visual Studio для дистрибутивов на базе Debian / Ubuntu - загрузить и установить пакет .deb (64-разрядный) (<https://go.microsoft.com/fwlink/?LinkID=760868>) либо через центр графического программного обеспечения, если он доступен, либо через командную строку с помощью:

```
sudo apt install ./<file>.deb
```

```
# If you're on an older Linux distribution, you will need to run this instead:  
# sudo dpkg -i <file>.deb  
# sudo apt-get install -f # Install dependencies
```

Обратите внимание, что другие двоичные файлы также доступны на странице загрузки VS Code (/Download).

Установка пакета .deb автоматически установит репозиторий apt и ключ подписи, чтобы включить автоматическое обновление с помощью системного менеджера пакетов. В качестве альтернативы репозиторий и ключ также можно установить вручную с помощью следующего скрипта:

```
sudo apt-get install wget gpg
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > packages.microsoft.gpg
sudo install -D -o root -g root -m 644 packages.microsoft.gpg /etc/apt/keyrings/packages.microsoft.gpg
sudo sh -c 'echo "deb [arch=amd64,arm64,armhf signed-by=/etc/apt/keyrings/packages.microsoft.gpg] https://packages.microsoft.com/repos/code stable main" > /etc/apt/sources.list.d/vscode.list'
rm -f packages.microsoft.gpg
```

Затем обновите кэш пакета и установите пакет с помощью:

```
sudo apt install apt-transport-https
sudo apt update
sudo apt install code # or code-insiders
```

Дистрибутивы на основе RHEL, Fedora и CentOS

В настоящее время мы отправляем стабильную 64-разрядную версию VS Code в репозиторий yum, следующий скрипт установит ключ и репозиторий:

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
sudo sh -c 'echo -e "[code]\nname=Visual Studio Code\nbaseurl=https://packages.microsoft.com/yumrepos/vscode\nenabled=1\nngpgcheck=1\nngpgkey=https://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/vscode.repo'
```

Затем обновите кэш пакета и установите пакет с помощью dnf (Fedora 22 и выше):

```
dnf check-update
sudo dnf install code
```

Или в более старых версиях с использованием yum :

```
yum check-update
sudo yum install code
```

Из-за ручного процесса подписания и системы, которую мы используем для публикации, репозиторий yum может отставать и не сразу получать последнюю версию VS Code.

Привязка

Код Visual Studio официально распространяется в виде пакета Snap в Snap Store (<https://snapcraft.io/store>):



(<https://snapcraft.io/code>)

Вы можете установить его, выполнив:

```
sudo snap install --classic code # or code-insiders
```

После установки демон Snap позаботится об автоматическом обновлении VS Code в фоновом режиме. Вы будете получать уведомления об обновлении продукта всякий раз, когда доступно новое обновление.

Примечание: Если snap он недоступен в вашем дистрибутиве Linux, пожалуйста, ознакомьтесь со следующим руководством по установке snapd (<https://docs.snapcraft.io/installing-snapd>), которое поможет вам настроить его.

Узнайте больше о snaps из официальной документации Snap (<https://docs.snapcraft.io>) .

Дистрибутивы на основе openSUSE и SLE

Репозиторий yum, приведенный выше, также работает для систем на базе openSUSE и SLE, следующий скрипт установит ключ и репозиторий:

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
sudo sh -c 'echo -e "[code]\nname=Visual Studio Code\nbaseurl=https://packages.micr\nosoftware.com/yumrepos/vscode\nenabled=1\ntype=rpm-md\nngpgcheck=1\nngpgkey=https://packa\nges.microsoft.com/keys/microsoft.asc" > /etc/zypp/repos.d/vscode.repo'
```

Затем обновите кэш пакета и установите пакет с помощью:

```
sudo zypper refresh
sudo zypper install code
```

Пакет AUR для Arch Linux

Для VS Code существует поддерживаемый сообществом пакет пользовательского репозитория Arch (<https://aur.archlinux.org/packages/visual-studio-code-bin>).

Чтобы получить дополнительную информацию об установке из AUR, пожалуйста, обратитесь к следующей вики-записи: Установите пакеты AUR (https://wiki.archlinux.org/index.php/Arch_User_Repository#Build_and_install_the_package).

Пакет Nix для NixOS (или любой дистрибутив Linux, использующий Nix package manager)

В репозитории nixpkgs имеется поддерживаемый сообществом пакет Nix для VS Code (<https://github.com/NixOS/nixpkgs/blob/master/pkgs/applications/editors/vscode/vscode.nix>). Чтобы установить его с помощью Nix, установите для allowUnfree параметра значение true в вашем config.nix и выполните:

```
nix-env -i vscode
```

Установка пакета .rpm вручную

Пакет VS Code .rpm (64-разрядный) (<https://go.microsoft.com/fwlink/?LinkID=760867>) также можно загрузить и установить вручную, однако автоматическое обновление не будет работать, если не установлен указанный выше репозиторий. После загрузки его можно установить с помощью вашего менеджера пакетов, например с помощью `dnf` :

```
sudo dnf install <file>.rpm
```

Обратите внимание, что другие двоичные файлы также доступны на странице загрузки VS Code (/Download).

Обновления

VS Code поставляется ежемесячно, и вы можете узнать, когда будет доступен новый выпуск, проверив примечания к выпуску (/updates). Если репозиторий VS Code был установлен правильно, то ваш системный менеджер пакетов должен обрабатывать автоматическое обновление таким же образом, как и другие пакеты в системе.

Примечание: Обновления выполняются автоматически и для пакета Snap выполняются в фоновом режиме.

Node.js

Node.js это популярная платформа и среда выполнения для простого создания и запуска приложений JavaScript. Он также включает в себя npm (<https://www.npmjs.com/>), менеджер пакетов для Node.js модули. Вы увидите Node.js и npm часто упоминаются в нашей документации, а также требуются некоторые дополнительные инструменты VS Code Node.js (например, генератор (/api/get-started/your-first-extension) расширений VS Code).

Если вы хотите установить Node.js в Linux см. раздел Установка Node.js через диспетчер пакетов (<https://nodejs.org/en/download/package-manager>), чтобы найти Node.js инструкции по упаковке и установке, адаптированные к вашему дистрибутиву Linux. Вы также можете установить и поддерживать несколько версий Node.js с помощью диспетчера версий узла (<https://github.com/creationix/nvm>).

Чтобы узнать больше о JavaScript и Node.js смотрите наш Node.js учебное пособие (/docs/nodejs/nodejs-tutorial), в котором вы узнаете о запуске и отладке Node.js приложения с VS Code.

Установка VS Code в качестве текстового редактора по умолчанию

xdg-открыть

Вы можете установить текстовый редактор по умолчанию для текстовых файлов (`text/plain`), который используется `xdg-open` , с помощью следующей команды:

```
xdg-mime default code.desktop text/plain
```

Альтернативная система Debian

Дистрибутивы на основе Debian позволяют устанавливать **редактор** по умолчанию с использованием системы альтернатив Debian (<https://wiki.debian.org/DebianAlternatives>), не заботясь о типе MIME. Вы можете установить это, выполнив следующее и выбрав code:

```
sudo update-alternatives --set editor /usr/bin/code
```

Если код Visual Studio не отображается в качестве альтернативы `editor`, вам необходимо зарегистрировать его:

```
sudo update-alternatives --install /usr/bin/editor editor $(which code) 10
```

Windows как машина разработчика Linux

Другой вариант разработки Linux с использованием VS Code - использовать компьютер под управлением Windows с подсистемой Windows для Linux (<https://learn.microsoft.com/windows/wsl/install>) (WSL).

Подсистема Windows для Linux

С помощью WSL вы можете устанавливать и запускать дистрибутивы Linux в Windows. Это позволяет разрабатывать и тестировать исходный код в Linux, продолжая работать локально на компьютере с Windows. WSL поддерживает дистрибутивы Linux, такие как Ubuntu, Debian, SUSE и Alpine, доступные в Microsoft Store.

When coupled with the WSL (<https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-wsl>) extension, you get full VS Code editing and debugging support while running in the context of a Linux distro on WSL.

See the Developing in WSL (</docs/remote/wsl>) documentation to learn more or try the Working in WSL (</docs/remote/wsl-tutorial>) introductory tutorial.

Следующие шаги

После установки VS Code эти разделы помогут вам узнать о нем больше:

- Дополнительные компоненты (</docs/setup/additional-components>) - Узнайте, как установить Git, Node.js, TypeScript и таких инструментов, как Yeoman.
- Пользовательский интерфейс (</docs/getstarted/userinterface>) - быстрая ориентация в VS Code.
- Настройки пользователя / рабочей области (</docs/getstarted/settings>) - Узнайте, как настроить VS Code в соответствии с вашими предпочтениями с помощью настроек.

Общие вопросы

Проблемы с виртуальной машиной Azure

Я получаю сообщение об ошибке "Запуск без песочницы SUID"?

Вы можете спокойно игнорировать эту ошибку.

Debian и перемещение файлов в корзину

Если вы видите ошибку при удалении файлов из VS Code Explorer в операционной системе Debian, это может быть связано с тем, что реализация корзины, которую использует VS Code, отсутствует.

Выполните эти команды, чтобы решить эту проблему:

```
sudo apt-get install gvfs-bin
```

Конфликтует с пакетами VS Code из других репозиториев

Некоторые дистрибутивы, например Pop!_OS (<https://pop.system76.com>) предоставляют свой собственный code пакет. Чтобы убедиться, что используется официальный репозиторий VS Code, создайте файл с именем `/etc/apt/preferences.d/code` со следующим содержимым:

```
Package: code
Pin: origin "packages.microsoft.com"
Pin-Priority: 9999
```

"Visual Studio Code не может отслеживать изменения файлов в этой большой рабочей области" (ошибка ENOSPC)

Когда вы видите это уведомление, это указывает на то, что у наблюдателя за файлами VS Code заканчиваются дескрипторы, поскольку рабочая область большая и содержит много файлов. Перед настройкой ограничений платформы убедитесь, что в настройку добавлены потенциально большие папки, такие как `Python .venv files.watcherExclude` (более подробная информация приведена ниже). Текущее ограничение можно просмотреть, выполнив:

```
cat /proc/sys/fs/inotify/max_user_watches
```

Ограничение может быть увеличено до максимума путем редактирования `/etc/sysctl.conf` (за исключением Arch Linux, читайте ниже) и добавления этой строки в конец файла:

```
fs.inotify.max_user_watches=524288
```

Затем новое значение можно загрузить, выполнив `sudo sysctl -p`.

Хотя 524 288 - это максимальное количество файлов, которые можно просматривать, если вы работаете в среде с особенно ограниченным объемом памяти, вы можете уменьшить это число. Каждый просмотр файла занимает 1080 байт (<https://stackoverflow.com/a/7091897/1156119>), поэтому, если предположить, что используются все 524 288 просмотров, это приводит к верхней границе около 540 Мбайт.

Дистрибутивы на основе Arch (<https://www.archlinux.org/>) (включая Manjaro) требуют, чтобы вы изменили другой файл; вместо этого выполните эти шаги

(<https://gist.github.com/tbjgolden/c53ca37f3bc2fab8c930183310918c8c>).

Другой вариант - исключить определенные каталоги рабочей области из средства просмотра файлов VS Code с помощью `files.watcherExclude` параметра (/docs/getstarted/settings). Значение по умолчанию для `files.watcherExclude` исключает `node_modules` и некоторые папки под `.git`, но вы можете добавить другие каталоги, которые вы не хотите, чтобы VS Code отслеживал.

```
"files.watcherExclude": {
  "**/.git/objects/**": true,
  "**/.git/subtree-cache/**": true,
  "**/node_modules/**/*.**": true
}
```

Я не вижу китайских символов в Ubuntu

Мы работаем над исправлением. Тем временем откройте меню приложения, затем выберите **Файл > Настройки > Настройки**. В разделе **Текстовый редактор > Шрифт** установите для "Семейства шрифтов" значение Droid Sans Mono, Droid Sans Fallback. Если вы предпочитаете редактировать `settings.json` файл напрямую, установите `editor.fontFamily`, как показано:

```
"editor.fontFamily": "Droid Sans Mono, Droid Sans Fallback"
```

Пакет git не установлен

Эта ошибка может появиться во время установки и обычно вызвана тем, что списки менеджера пакетов устарели. Попробуйте обновить их и установить снова:

```
# For .deb
sudo apt-get update

# For .rpm (Fedora 21 and below)
sudo yum check-update

# For .rpm (Fedora 22 and above)
sudo dnf check-update
```

Команда `code bin` не выводит окно на передний план в Ubuntu

`code`. Запуск в Ubuntu, когда VS Code уже открыт в текущем каталоге, не выведет VS Code на передний план. Это функция операционной системы, которую можно отключить с помощью `ccsm`.

```
# Install
sudo apt-get update
sudo apt-get install compizconfig-settings-manager

# Run
ccsm
```

Под **общие** > **общие параметры** > **внимание и поднять поведения**, набор "фокус предупреждению уровень" на "off". Помните, что это параметр уровня операционной системы, который будет применяться ко всем приложениям, а не только к VS Code.

Не удастся установить .пакет deb из-за "/etc/apt/sources.list.d/vscode.list: нет такого файла или каталога"

Это может произойти, когда `sources.list.d` файл не существует или у вас нет доступа для создания файла. Чтобы исправить это, попробуйте вручную создать папку и пустой `vscode.list` файл:

```
sudo mkdir /etc/apt/sources.list.d
sudo touch /etc/apt/sources.list.d/vscode.list
```

Не удастся переместить или изменить размер окна во время X пересылки удаленного окна

Если вы используете X forwarding для удаленного использования VS Code, вам нужно будет использовать встроенную строку заголовка, чтобы убедиться, что вы можете правильно управлять окном. Вы можете переключиться на его использование, установив `window.titleBarStyle` значение `native`.

Использование пользовательской строки заголовка

Настраиваемая строка заголовка и меню были включены по умолчанию в Linux в течение нескольких месяцев. Пользовательская строка заголовка была успешной в Windows, но реакция клиента в Linux свидетельствует об обратном. Основываясь на отзывах, мы решили включить этот параметр в Linux и оставить встроенную строку заголовка по умолчанию.

Пользовательская строка заголовка предоставляет множество преимуществ, включая отличную поддержку тематизации и улучшенную доступность благодаря навигации с клавиатуры и программам чтения с экрана. К сожалению, эти преимущества не так хорошо переносятся на платформу Linux. В Linux есть множество окружений рабочего стола и оконных менеджеров, из-за которых тематизация VS Code может показаться пользователям непривычной. Пользователям, нуждающимся в улучшениях специальных возможностей, мы рекомендуем включать пользовательскую строку заголовка при запуске в режиме специальных возможностей с помощью программы чтения с экрана. Вы все еще можете вручную настроить строку заголовка с помощью параметра **Window: Стиль строки заголовка** (`window.titleBarStyle`).

Сломанный курсор в редакторе с включенным масштабированием дисплея

Из-за вышестоящей проблемы # 14787 (<https://github.com/electron/electron/issues/14787>) с Electron курсор мыши может отображаться неправильно при включенном масштабировании. Если вы заметили, что обычный текстовый курсор отображается в редакторе не так, как можно было бы ожидать, попробуйте вернуться к родной строке меню, настроив параметр `window.titleBarStyle` на `native`.

Репозиторий изменил свое исходное значение

Если вы получите сообщение об ошибке, подобное следующему:


```
E: Repository '...' changed its 'Origin' value from '...' to '...'
N: This must be accepted explicitly before updates for this repository can be applied. See apt-secure(8) manpage for details.
```

Используйте `apt` вместо `apt-get`, и вам будет предложено принять исходное изменение:

```
sudo apt update
```

Была ли эта документация полезной?

Да

5/3/2023

Hello from Seattle. Follow @code (<https://go.microsoft.com/fwlink/?LinkId=533687>)

Звезда 146,703

Support (<https://support.serviceshub.microsoft.com/supportforbusiness/create?sapId=d66407ed-3967-b000-4cfb-2c318cad363d>)

Privacy (<https://go.microsoft.com/fwlink/?LinkId=521839>)

Terms of Use (<https://www.microsoft.com/legal/terms-of-use>)

License (/License)

 Microsoft (<https://www.microsoft.com>)

© 2023 Microsoft