WinGet (Windows Package Manager) - пакетный менеджер для Windows

* Автор: Уваров А.С.

* 29.09.2023



Пакетный менеджер - хорошо знакомый любому Linux администратору вид ПО, который позволяет централизованно управлять программным обеспечением в системе и легко автоматизировать этот процесс. Пользователи Windows долгое время были лишены подобного инструмента, собственно, как вообще какого-то централизованного подхода к управлению ПО, но затем в Windows появился Магазин, а затем и менеджер пакетов, названный просто - Windows Package Manager или WinGet. В данной статье мы рассмотрим его особенности и примеры работы с ним.

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на [**углубленном курсе по администрированию MikroTik**](https://xn-----xlcfvffioc4g.xn--p1ai/lp-mikrotik-mtcna?utm_source=interface31&utm_medium=cpc&utm_campaign=92-1495). Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

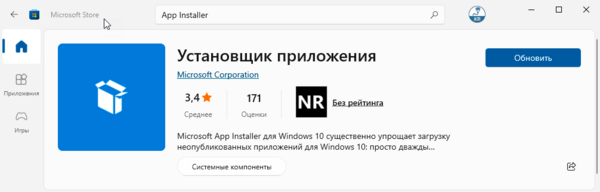
В целом, если вы имеете опыт работы с любым пакетным менеджером, то начать работу с WinGet будет несложно, но есть определенные отличия. Начнем с того, что в Linux системах репозиторий является доверенным источником приложений, который связан либо с дистрибутивом, либо с разработчиком ПО и мы всегда понимаем из какого источника мы получаем тот или иной пакет.

В WinGet имеется два источника пакетов: магазин Windows и собственный репозиторий, поддерживаемый Microsoft. Но данный репозиторий является лишь одним из каналов распространения ПО независимыми поставщиками. Грубо говоря репозиторий WinGet является просто каталогом ссылок на скачивание инсталляторов того или иного ПО из различных источников, предлагаемых их разработчиками.

Это, конечно же лучше, чем искать и скачивать пакеты в интернет, но серьезно отличается от принятой в Linux модели: если там, скачивая пакет из репозитория дистрибутива мы можем быть уверены, что пакет протестирован именно под вашу версию дистрибутива, то здесь все вопросы к разработчику, если пакет нестабильно работает или не работает под вашу версию ОС, то WinGet ничем вам тут не поможет.

Равно как вопрос доверия к дистрибутиву и репозиторию как единому целому меняется на вопрос доверия к каждой отдельной программе, поэтому будьте внимательны и осмотрительны в выборе ПО при использовании пакетного менеджера Windows.

Возможности WinGet доступны начиная с Windows 10 1709 и во всех актуальных выпусках он предустановлен по умолчанию, в противном случае его можно получить в Магазине Windows под названием **App Installer**:

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-001-18150.html)

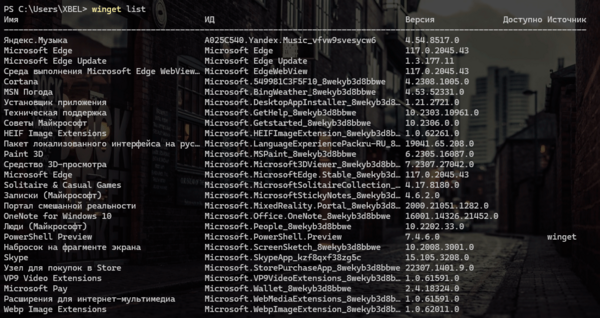
Либо скачать с официальной страницы на Github:

[**microsoft/winget-cli**](https://github.com/microsoft/winget-cli)

Начнем же работу с ним с получения списка уже установленных приложений:

winget list

Следует отметить, что WinGet поддерживает работу и с приложениями, установленными вручную или из магазина, поэтому вы всегда можете комбинировать подходы, это удобно и полностью оправдано в принятой в Windows модели управления ПО. Давайте рассмотрим вывод команды подробнее:

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-002-18153.html)Колонка **Имя** содержит понятное пользователю имя приложения на языке локализации системы, **ИД (ID)** - его идентификатор, уникальный для каждого приложения, колонки **Версия** и**Доступно** показывают установленную версию и последнюю доступную в репозиториях. И наконец колонка **Источник** показывает на репозиторий в котором доступен пакет. Посмотреть список подключенных репозиториев можно командой:

winget source list

Как мы уже говорили выше, по умолчанию подключено два репозитория: Магазин Windows и WinGet.

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-003-18156.html)Отсутствие репозитория в колонке источник означает, что пакета нет в репозиториях WinGet, но вы можете управлять им с его помощью, хотя список доступных возможностей здесь невелик и сводится к удалению пакета, это может быть удобно в сценариях, когда вы хотите автоматически удалить часть предустановленного ПО. Например, мы можем легко удалить Яндекс.Музыку командой:

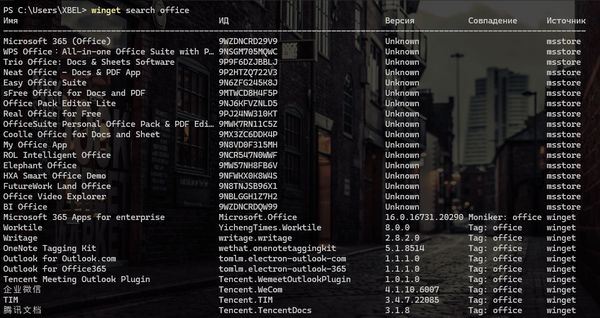
winget uninstall Яндекс.Музыка

Но установить назад мы ее уже не сможем, так как программы нет в доступных источниках пакетов.

Для установки ПО нам потребуется указать название пакета или его идентификатор, чтобы найти их следует использовать команду поиска, например:

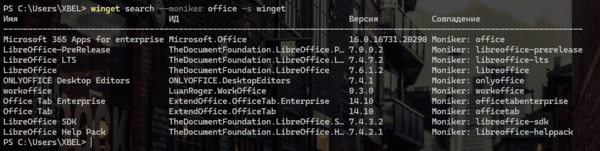
winget search office

В результате получим все возможные варианты, найденные по заданной строке:

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-004-18159.html)Если не указан фильтр, то поиск производится по всем доступным полям: имя, идентификатор, тег и моникер (прозвище), последний параметр представляет короткое имя, связанное именно с этим пакетом, если строка запроса содержит пробелы, то ее нужно обернуть в кавычки. Также мы можем указать источник, в котором производим поиск. Например, выполним поиск по прозвищу "office" в репозитории Winget:

winget search --moniker office -s winget

Результат показывает все пакеты, моникер которых содержит искомое вхождение:

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-005-18162.html)Если нам нужно строгое вхождение, то к строке запроса нужно добавить ключ**-e**, в этом случае будет искаться полное совпадение с учетом регистра. Например, команда:

winget search --moniker office -s winget -e

Вернет только первую строку, так как она единственная полностью подходит под запрос.

Всего мы можем использовать следующие фильтры:

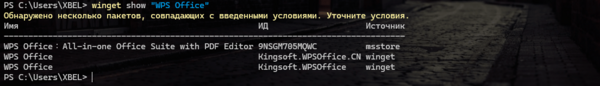
* **--name** - имя пакета
* **--id** - идентификатор пакета
* **--tag** - тег пакета
* **--moniker** - прозвище пакета

Если вы забыли какие-то ключи или опции команды не беда, просто запустите ее без аргументов и вы получите короткую справку по команде.

После того, как мы нашли интересующий нас пакет, мы можем посмотреть информацию о нем, например:

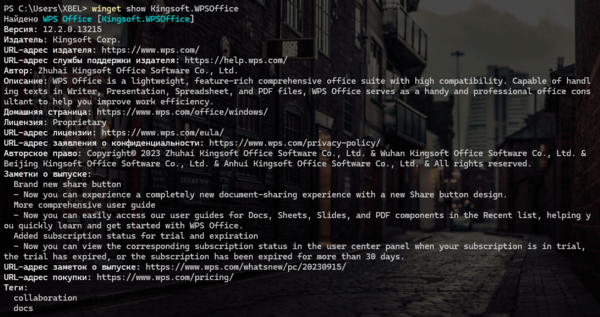
winget show "WPS Office"

Если под запрос попадают сразу несколько пакетов, то пакетный менеджер попросит нас уточнить запрос:

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-006-18165.html)В большинстве случаев будет достаточно добавить ключ **-e** для строгого совпадения, но не в этот раз, потому что китайская и международная версия имеют одинаковое имя, поэтому в данном случае правильно будет использовать идентификатор:

winget show Kingsoft.WPSOffice

После чего мы получим всю информацию о пакете, включая теги и прозвище, если есть.

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-007-18168.html)Для получения списка версий доступных в репозитории выполните:

winget show Kingsoft.WPSOffice --versions

Теперь можно установить пакет, для это используется команда**install** и она точно также производит поиск по строке запроса. Будет установлено самое точное вхождение, при нахождении нескольких результатов вас попросят уточнить запрос.

winget install Kingsoft.WPSOffice

Мы также можем добавить фильтр поиска, скажем, только по ID:

winget install --id Kingsoft.WPSOffice

После чего пакетный менеджер скачает и запустит инсталлятор, при этом пользователь будет видеть ход установки, но его участие в ней не потребуется, а само приложение будет установлено с параметрами по умолчанию.

Есть два способа изменить это поведение, для того чтобы полностью скрыть процесс установки используйте ключ:

-h, --silent

А для запуска установки в интерактивном режиме, когда вам, как обычно, придется отвечать на вопросы и выбирать варианты, добавьте ключ:

-i, --interactive

Чтобы установить пакет нужной версии используйте ключ:

-v, --version

Альтернативой установке может быть действие **download**, которое просто скачает в папку Загрузки инсталлятор пакета. Для обоих действий можно явно указать архитектуру при помощи ключа:

-a, --architecture

Доступными являются значения: X86 и X64.

А если пакет поддерживает разделение на языковые версии, то также можно указать нужный язык при помощи ключа:

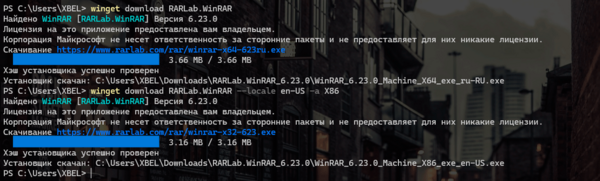
--locale

Значение языка следует указывать в формате **BCP47**, для русского это будет **ru-RU,**а для английского**en-US**. По умолчанию скачивается пакет с локалью и разрядностью совпадающей с параметрами системы. Например, команда:

winget download RARLab.WinRAR

Скачает русскую 64-разрядную версию приложения, а для скачивания 32-разрядной английской нам потребуется указать:

winget download RARLab.WinRAR --locale en-US -a X86

[](https://interface31.ru/tech_it/assets_c/2023/09/WinGet-Windows-Package-Manager-009-18171.html)С установкой и скачиванием понятно, но одно из самых ценных умений пакетного менеджера - это обновление системы и именно этого сильно не хватало пользователям и администраторам Windows. Чтобы обновить отдельный пакет используйте:

winget upgrade Microsoft.VCRedist.2015+.x64

В данном случае мы обновим распространяемый пакет Microsoft Visual C++ Redistributable

Для того, чтобы обновить все пакеты, имеющие доступные обновления, используйте:

winget upgrade --all

С обновлением пакета тесно связано еще одно действие - закрепление, которое позволяет зафиксировать текущую версию пакета и сделать его недоступным для обновления. В самом простом случае для закрепления используйте:

winget pin add RARLab.WinRAR

После чего пакет не будет обновлен при использовании команды **upgrade --all**, но может быть обновлен командой обновления пакета или добавлением к **upgrade --all** ключа **--include-pinned**.

Чтобы полностью заблокировать обновление пакета следует добавить дополнительный ключ:

winget pin add RARLab.WinRAR --blocking

Еще один вариант закрепления, это закрепление пакета в рамках определенной версии, например:

winget pin add LibreOffice --version 7.4.\*

Для просмотра закрепленных пакетов используйте

winget pin list

А для удаления закрепления:

winget pin remove LibreOffice

Чтобы удалить установленную программу используйте действие **uninstall**:

winget uninstall LibreOffice

Но на этом возможности пакетного менеджера не заканчиваются, вы можете экспортировать список установленного ПО командой:

winget export -o C:\ADM\myapp.json

Если требуется указать конкретные версии ПО, то добавьте ключ:

--include-versions

После чего в указанный файл будет выгружен список всего установленного на ПК ПО, которое есть в репозиториях, неважно как оно было установлено, через WinGet или вручную.

Для автоматической установки ПО из указанного списка используйте команду импорта:

winget import -i C:\ADM\myapp.json

Это позволяет быстро восстановить набор ПО после переустановки ОС или перенести его на другой ПК, а также развернуть нужный набор программ на партию компьютеров.

Если вы запускаете команду импорта на новом ПК, то добавьте ключ:

--accept-source-agreements

Который подавляет запрос на принятие исходного лицензионного соглашения на использование источника пакетов. Также можно добавить:

--accept-package-agreements

Для автоматического принятия лицензионного соглашения, которое может потребоваться для некоторых пакетов.

Чтобы избежать появления ошибок в случае недоступности пакета в источнике используйте:

--ignore-unavailable

Если в файле экспорта указаны определенные версии пакетов, то вы можете проигнорировать их ключом:

--ignore-versions

При его использовании будут установлены самые последние доступные версии пакетов.

Также, если вы не хотите обновлять уже существующие пакеты добавьте ключ:

--no-upgrade

В итоге команда импорта на новом ПК может выглядеть так:

winget import -i C:\ADM\myapp.json --accept-source-agreements --accept-package-agreements --ignore-unavailable --no-upgrade

Конечно, WinGet - молодой продукт и его возможности во многом уступают возможностям пакетных менеджеров Linux, также есть неразрешенные проблемы и неоднозначное поведение менеджера при попытках обновления уже запущенного приложения, если сам инсталлятор не умеет корректно обрабатывать данную ситуацию. Но он развивается и предлагает администратору простой, но в тоже время мощный инструмент для управления программным обеспечением с широкими возможностями автоматизации.