

Создание собственного пакета Chocolatey

Экспорт в PDF 

Дата создания: 2024/08/19 14:06 (C) mihanik



Дано

Вышла новая платформа 1с.



Рассмотрим пример публикации платформы 1С на собственном сервере Chocolatey
Для определённости пусть версия платформы будет **8.3.24.1667**

Надо

Добавить версию платформы 1С на свой сервер Chocolatey чтобы в дальнейшем можно было быстро развернуть его на компьютерах пользователей.

Важно!!!

Chocolatey старается устанавливать на ПК программный пакет именно той разрядности, что и разрядность системы.



Для 1С это не всегда хорошо, т.к. если на ПК используются эквайринговые системы и некоторые другие «плюшки», работа с кассами и некоторым другим оборудованием может быть нарушена.

Мы используем обходной манёвр: «скажем» Chocolatey, что 32-хразрядная платформа отлично работает на Windows любой разрядности. 😊

Решение

Шаг 1

С сайта <https://releases.1c.ru/total> скачиваем архив с дистрибутивом программы.

Технологическая платформа 8.3, версия 8.3.24.1667

Контрольные суммы

Технологическая платформа 8.3. Версия 8.3.24.1667. Список изменений и порядок обновления

- Windows (64-bit)
- Windows (32-bit)
 - Тонкий клиент 1С:Предприятия для Windows
 - Технологическая платформа 1С:Предприятия для Windows + Тонкий клиент для Windows, Linux и MacOS для автоматического обновления клиентов через веб-сервер
 - Технологическая платформа 1С:Предприятия для Windows + Тонкий клиент для Windows и MacOS для автоматического обновления клиентов через веб-сервер
 - Технологическая платформа 1С:Предприятия для Windows ←

Шаг 2

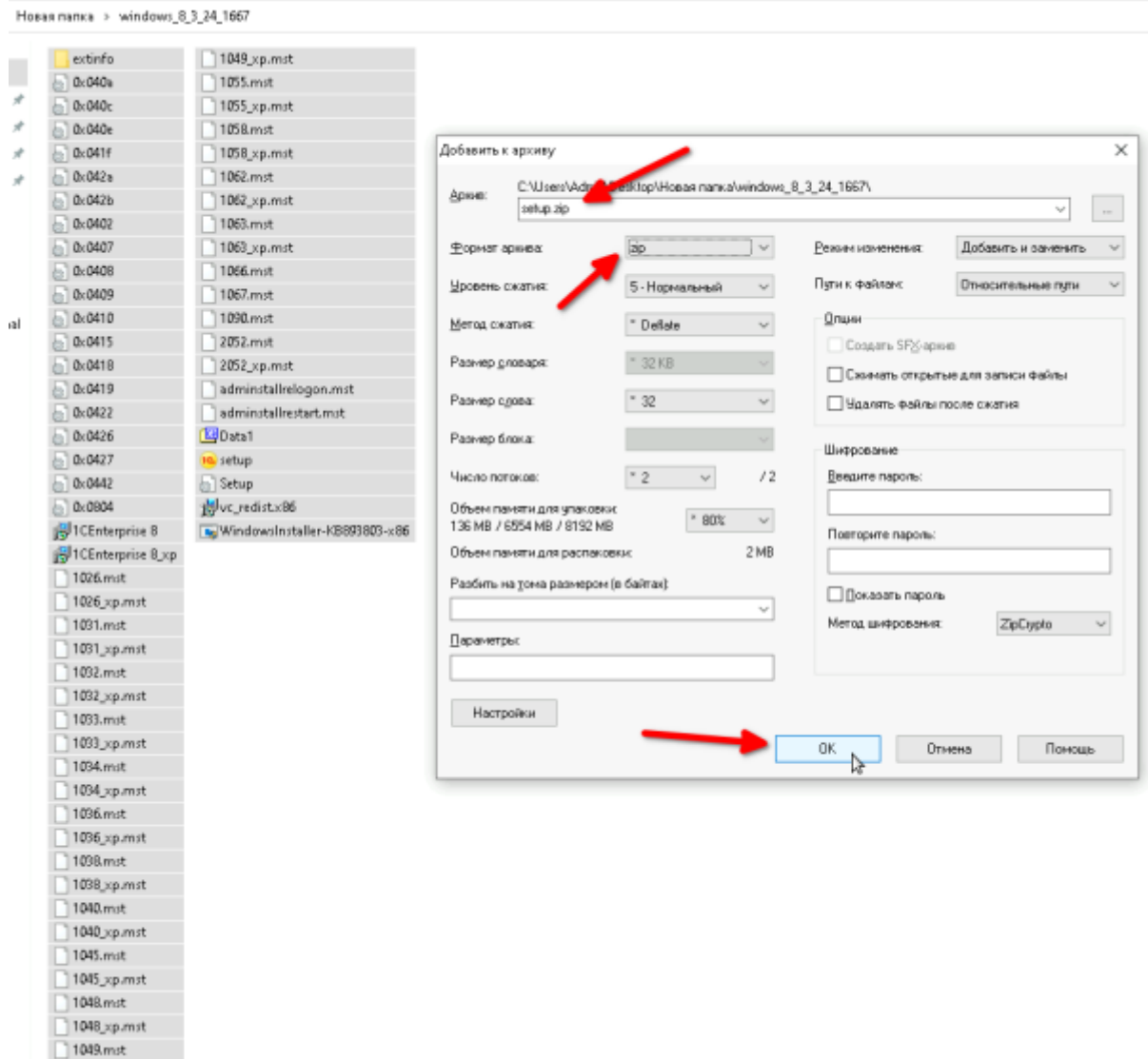
Скачанный **RAR**-архив с дистрибутивом распаковываем в какую-нибудь папку, а затем опять упаковываем в **ZIP**-архив с именем **setup.zip**



С RAR-архивами Chocolatey не умеет работать «из коробки». Можно, конечно, и пропустить этот шаг, но тогда усложнится реализация алгоритма установки.



В работе я обычно использую архиватор 7Zip



Шаг 3

Получившийся архив **setup.zip** размещаем на своём сервере

choco.mihanik.net - /distr/1c/x32/8.3.24.1667/

[\[В родительский каталог\].](#)

25.08.2024 15:45 902528825 [setup.zip](#)



Важно!!!

Обязательно получите контрольную сумму архива.
Это пригодится при дальнейшей настройке!

Для вычисления контрольной суммы запускаем PowerShell и подаём несложную команду. Например, такую.

```
Get-FileHash C:\tools\chocolatey.server\distr\1c\x32\8.3.24.1667\setup.zip
-Algorithm SHA256
```



Можно получить контрольную сумму с использованием разных алгоритмов.

```
Выбрать Windows PowerShell
Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\Admin> Get-FileHash C:\tools\chocolatey.server\distr\1c\x32\8.3.24.1667\setup.zip -Algorithm MD5

Algorithm      Hash                                     Path
-----
MD5             7BEEF56EC49202A632A860FDA9C9F8E       C:\tools\chocolatey.server\di...

PS C:\Users\Admin> Get-FileHash C:\tools\chocolatey.server\distr\1c\x32\8.3.24.1667\setup.zip -Algorithm SHA256

Algorithm      Hash                                     Path
-----
SHA256         42654E8A9F8C50156F9C38B785687F2B378D3CE51741A8858422B95806249F26       C:\tools\chocolatey.server\di...

PS C:\Users\Admin>
```

Шаг 4

Теперь открываем командную строку и переходим в «песочницу», где мы «играем» с пакетами Chocolatey.

```
Администратор: Командная строка
Microsoft Windows [Version 10.0.19044.1288]
(с) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Windows\system32>cd C:\tools\sandbox

C:\tools\sandbox>
```

Шаг 5

Создаём шаблон для нового релиза платформы 1С.

```
choco new 1c --version=8.3.24.1667 --maintainer=mihanikus --force
```

Шаг 6

Теперь файл **1c.nuspec** приводим к следующему виду.



В моём случае имя файла оказалось таким: **C:\tools\sandbox\1c\1c.nuspec**

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Read this before creating packages:
https://docs.chocolatey.org/en-us/create/create-packages -->
<!-- It is especially important to read the above link to understand
additional requirements when publishing packages to the community feed aka
dot org (https://community.chocolatey.org/packages). -->

<!-- Test your packages in a test environment:
https://github.com/chocolatey/chocolatey-test-environment -->

<!--
This is a nuspec. It mostly adheres to
https://docs.nuget.org/create/Nuspec-Reference. Chocolatey uses a special
version of NuGet.Core that allows us to do more than was initially possible.
As such there are certain things to be aware of:

* the package xmlns schema url may cause issues with nuget.exe
* Any of the following elements can ONLY be used by choco tools -
projectSourceUrl, docsUrl, mailingListUrl, bugTrackerUrl, packageSourceUrl,
provides, conflicts, replaces
* nuget.exe can still install packages with those elements but they are
ignored. Any authoring tools or commands will error on those elements
-->

<!-- You can embed software files directly into packages, as long as you are
not bound by distribution rights. -->
<!-- * If you are an organization making private packages, you probably have
no issues here -->
<!-- * If you are releasing to the community feed, you need to consider
distribution rights. -->
<!-- Do not remove this test for UTF-8: if "Ω" doesn't appear as greek
uppercase omega letter enclosed in quotation marks, you should use an editor
```

```

that supports UTF-8, not this one. -->
<package xmlns="http://schemas.microsoft.com/packaging/2015/06/nuspec.xsd">
  <metadata>
    <!-- == PACKAGE SPECIFIC SECTION == -->
    <!-- This section is about this package, although id and version have
ties back to the software -->
    <!-- id is lowercase and if you want a good separator for words, use '-
', not '.'. Dots are only acceptable as suffixes for certain types of
packages, e.g. .install, .portable, .extension, .template -->
    <!-- If the software is cross-platform, attempt to use the same id as
the debian/rpm package(s) if possible. -->
    <id>lc</id>
    <!-- version should MATCH as closely as possible with the underlying
software -->
    <!-- Is the version a prerelease of a version?
https://docs.nuget.org/create/versioning#creating-prerelease-packages -->
    <!-- Note that unstable versions like 0.0.1 can be considered a released
version, but it's possible that one can release a 0.0.1-beta before you
release a 0.0.1 version. If the version number is final, that is considered
a released version and not a prerelease. -->
    <version>8.3.24.1667</version>
    <!-- <packageSourceUrl>Where is this Chocolatey package located (think
GitHub)? packageSourceUrl is highly recommended for the community
feed</packageSourceUrl>-->
    <!-- owners is a poor name for maintainers of the package. It sticks
around by this name for compatibility reasons. It basically means you. -->
    <!--<owners>mihanikus</owners>-->
    <!-- ===== -->

    <!-- == SOFTWARE SPECIFIC SECTION == -->
    <!-- This section is about the software itself -->
    <title>lc (Install)</title>
    <authors>lc</authors>
    <!-- projectUrl is required for the community feed -->
    <projectUrl>https://lc.ru</projectUrl>
    <!-- There are a number of CDN Services that can be used for hosting the
Icon for a package. More information can be found here:
https://docs.chocolatey.org/en-us/create/create-packages#package-icon-guidel
ines -->
    <!-- Here is an example using Githack -->
    <!--
<iconUrl>http://rawcdn.githack.com/___REPLACE_YOUR_REPO___/master/icons/lc.png
</iconUrl>-->
    <!-- <copyright>Year Software Vendor</copyright> -->
    <!-- If there is a license Url available, it is required for the
community feed -->
    <!-- <licenseUrl>Software License Location
___REMOVE_OR_FILL_OUT___</licenseUrl>
    <requireLicenseAcceptance>true</requireLicenseAcceptance>-->
    <!--<projectSourceUrl>Software Source Location - is the software FOSS
somewhere? Link to it with this</projectSourceUrl>-->

```

```

<!--<docsUrl>At what url are the software docs located?</docsUrl>-->
<!--<mailingListUrl></mailingListUrl>-->
<!--<bugTrackerUrl></bugTrackerUrl>-->
<tags>lc lcx32</tags>
<summary></summary>
<description>Установка платформы lc через choco</description>
<!-- <releaseNotes>_REPLACE_OR_REMOVE_MarkDown_Okay</releaseNotes> -->
<!-- ===== -->

<!-- Specifying dependencies and version ranges?
https://docs.nuget.org/create/versioning#specifying-version-ranges-in-.nuspe
c-files -->
<!--<dependencies>
  <dependency id="" version="__MINIMUM_VERSION__" />
  <dependency id="" version="[__EXACT_VERSION__]" />
  <dependency id="" version="[_MIN_VERSION_INCLUSIVE,
MAX_VERSION_INCLUSIVE]" />
  <dependency id="" version="[_MIN_VERSION_INCLUSIVE,
MAX_VERSION_EXCLUSIVE)" />
  <dependency id="" />
  <dependency id="chocolatey-core.extension" version="1.1.0" />
</dependencies>-->
<!-- chocolatey-core.extension -
https://community.chocolatey.org/packages/chocolatey-core.extension -->

<!--<provides>NOT YET IMPLEMENTED</provides>-->
<!--<conflicts>NOT YET IMPLEMENTED</conflicts>-->
<!--<replaces>NOT YET IMPLEMENTED</replaces>-->
</metadata>
<files>
  <!-- this section controls what actually gets packaged into the
Chocolatey package -->
  <file src="tools\*" target="tools" />
</files>
</package>

```

Собственно говоря, указываем некоторые данные. Остальное можно не трогать.

```

<id>lc</id>
...
<version>8.3.24.1667</version>
...
<title>lc (Install)</title>
...
<projectUrl>https://lc.ru</projectUrl>
...
<tags>lc lcx32</tags>
...
<description>Установка платформы lc через choco</description>

```

Этого будет достаточно.

Шаг 7

Теперь настраиваем алгоритм установки 1C

Алгоритм будет состоять из 2-х частей:

- скачивание и распаковка архива с дистрибутивом
- непосредственно установка платформы 1C



В моём случае алгоритм установки находится в файле с именем **C:\tools\sandbox\1c\tools\chocolateyinstall.ps1**

Файл **chocolateyinstall.ps1** привожу к следующему виду

```
$ErrorActionPreference = 'Stop' # stop on all errors
$toolsDir = "$(Split-Path -parent $MyInvocation.MyCommand.Definition)"

$packageZipArgs = @{
    packageName     = $env:ChocolateyPackageName
    unzipLocation  = $toolsDir
    Url             =
'http://choco.mihanik.net/distr/1c/x32/8.3.24.1667/setup.zip'
    Checksum       =
'42654E8A9F8C50156F9C38B785687F2B378D3CE51741ABB58422B95B06249F26'
    ChecksumType   = 'SHA256'
    Url64          =
'http://choco.mihanik.net/distr/1c/x32/8.3.24.1667/setup.zip'
    Checksum64     =
'42654E8A9F8C50156F9C38B785687F2B378D3CE51741ABB58422B95B06249F26'
    ChecksumType64 = 'SHA256'
    softwareName   = '1C*'
    validExitCodes = @(0, 3010, 1641)
}

$packageArgs = @{
    packageName     = $env:ChocolateyPackageName
    file            = $toolsDir + '\1CEnterprise 8.msi'
    file64         = $toolsDir + '\1CEnterprise 8.msi'
    fileType       = 'MSI'
    silentArgs     = '/qn TRANSFORMS=admininstallrelogon.mst;1049.mst
DESIGNERALLCLIENTS=1 THICKCLIENT=1 THINCLIENTFILE=1 THINCLIENT=1
WEBSERVEREXT=1 SERVER=0 CONFREPOSSERVER=0 CONVERTER77=0 SERVERCLIENT=1
LANGUAGES=RU'
    validExitCodes = @(0, 3010, 1641)
    softwareName   = '1C*'
}

Write-Output "Установка 1c."
```



```
Write-Output "Скачиваем архив с программой."

Install-ChocolateyZipPackage @packageZipArgs

Write-Output "Устанавливаем платформу."

Install-ChocolateyInstallPackage @packageArgs
```

Шаг 8

Настраиваем алгоритм удаления нашей платформы. Для этого редактируем файл **chocolateyuninstall.ps1**.



В моём случае файл имеет имя
C:\tools\sandbox\1c\tools\chocolateyuninstall.ps1

Файл **chocolateyuninstall.ps1** привожу к следующему виду

```
# IMPORTANT: Before releasing this package, copy/paste the next 2 lines into
PowerShell to remove all comments from this file:
# $f='c:\path\to\thisFile.ps1'
# gc $f | ? {$_ -notmatch "\s*#"} | % {$_ -replace
'(^.*?)\s*?[\`~]#.*','$1'} | Out-File $f+"~" -en utf8; mv -fo $f+"~" $f

## NOTE: In 80-90% of the cases (95% with licensed versions due to Package
Synchronizer and other enhancements),
## AutoUninstaller should be able to detect and handle registry uninstalls
without a chocolateyUninstall.ps1.
## See https://docs.chocolatey.org/en-us/choco/commands/uninstall
## and
https://docs.chocolatey.org/en-us/create/functions/uninstall-chocolateypacka
ge

## If this is an MSI, ensure 'softwareName' is appropriate, then clean up
comments and you are done.
## If this is an exe, change fileType, silentArgs, and validExitCodes

$errorActionPreference = 'Stop' # stop on all errors
$packageArgs = @{
    packageName     = $env:ChocolateyPackageName
    softwareName    = '1c*' #part or all of the Display Name as you see it in
Programs and Features. It should be enough to be unique
    fileType        = 'MSI' #only one of these: MSI or EXE (ignore MSU for now)
    # MSI
    silentArgs      = "/qn /norestart"
    validExitCodes= @(0, 3010, 1605, 1614, 1641) #
```

```

https://msdn.microsoft.com/en-us/library/aa376931(v=vs.85).aspx
# OTHERS
# Uncomment matching EXE type (sorted by most to least common)
#silentArgs = '/S' # NSIS
#silentArgs = '/VERYSILENT /SUPPRESSMSGBOXES /NORESTART /SP-' # Inno
Setup
#silentArgs = '/s' # InstallShield
#silentArgs = '/s /v"/qn"' # InstallShield with MSI
#silentArgs = '/s' # Wise InstallMaster
#silentArgs = '-s' # Squirrel
#silentArgs = '-q' # Install4j
#silentArgs = '-s -u' # Ghost
# Note that some installers, in addition to the silentArgs above, may also
need assistance of AHK to achieve silence.
#silentArgs = '' # none; make silent with input macro script
like AutoHotKey (AHK)
#
https://community.chocolatey.org/packages/autohotkey.portable
#validExitCodes= @(0) #please insert other valid exit codes here
}

[array]$key = Get-UninstallRegistryKey -SoftwareName
$packageArgs['softwareName']

if ($key.Count -eq 1) {
    $key | % {
        $packageArgs['file'] = "$($_.UninstallString)" #NOTE: You may need to
split this if it contains spaces, see below

        if ($packageArgs['fileType'] -eq 'MSI') {
            # The Product Code GUID is all that should be passed for MSI, and very
            # FIRST, because it comes directly after /x, which is already set in
the
            # Uninstall-ChocolateyPackage msiargs (facepalm).
            $packageArgs['silentArgs'] = "$($_.PSChildName)
$($packageArgs['silentArgs'])"

            # Don't pass anything for file, it is ignored for msi (facepalm number
2)
            # Alternatively if you need to pass a path to an msi, determine that
and
            # use it instead of the above in silentArgs, still very first
            $packageArgs['file'] = ''
        } else {
            # NOTES:
            # - You probably will need to sanitize $packageArgs['file'] as it
comes from the registry and could be in a variety of fun but unusable
formats
            # - Split args from exe in $packageArgs['file'] and pass those args
through $packageArgs['silentArgs'] or ignore them
            # - Ensure you don't pass double quotes in $file (aka

```

```
$packageArgs['file']) - otherwise you will get "Illegal characters in path
when you attempt to run this"
    # - Review the code for auto-uninstaller for all of the fun things it
does in sanitizing -
https://github.com/chocolatey/choco/blob/bfe351b7d10c798014efe4bfbb100b171db
25099/src/chocolatey/infrastructure.app/services/AutomaticUninstallerService
.cs#L142-L192
    }

    Uninstall-ChocolateyPackage @packageArgs
}
} elseif ($key.Count -eq 0) {
    Write-Warning "$packageName has already been uninstalled by other means."
} elseif ($key.Count -gt 1) {
    Write-Warning "$($key.Count) matches found!"
    Write-Warning "To prevent accidental data loss, no programs will be
uninstalled."
    Write-Warning "Please alert package maintainer the following keys were
matched:"
    $key | % {Write-Warning "- $($_.DisplayName)"}
}

## OTHER POWERSHELL FUNCTIONS
## https://docs.chocolatey.org/en-us/create/functions
#Uninstall-ChocolateyZipPackage $packageName # Only necessary if you did not
unpack to package directory - see
https://docs.chocolatey.org/en-us/create/functions/uninstall-chocolateyzipa
ckage
#Uninstall-ChocolateyEnvironmentVariable -
https://docs.chocolatey.org/en-us/create/functions/uninstall-chocolateyenvir
onmentvariable
#Uninstall-BinFile # Only needed if you used Install-BinFile - see
https://docs.chocolatey.org/en-us/create/functions/uninstall-binfile
## Remove any shortcuts you added in the install script.
```

Собственно говоря, я тут отредактировал только одну строчку.

```
fileType      = 'MSI' #only one of these: MSI or EXE (ignore MSU for now)
```

Шаг 9

Теперь можем проверить правильность настройки нашего пакета.

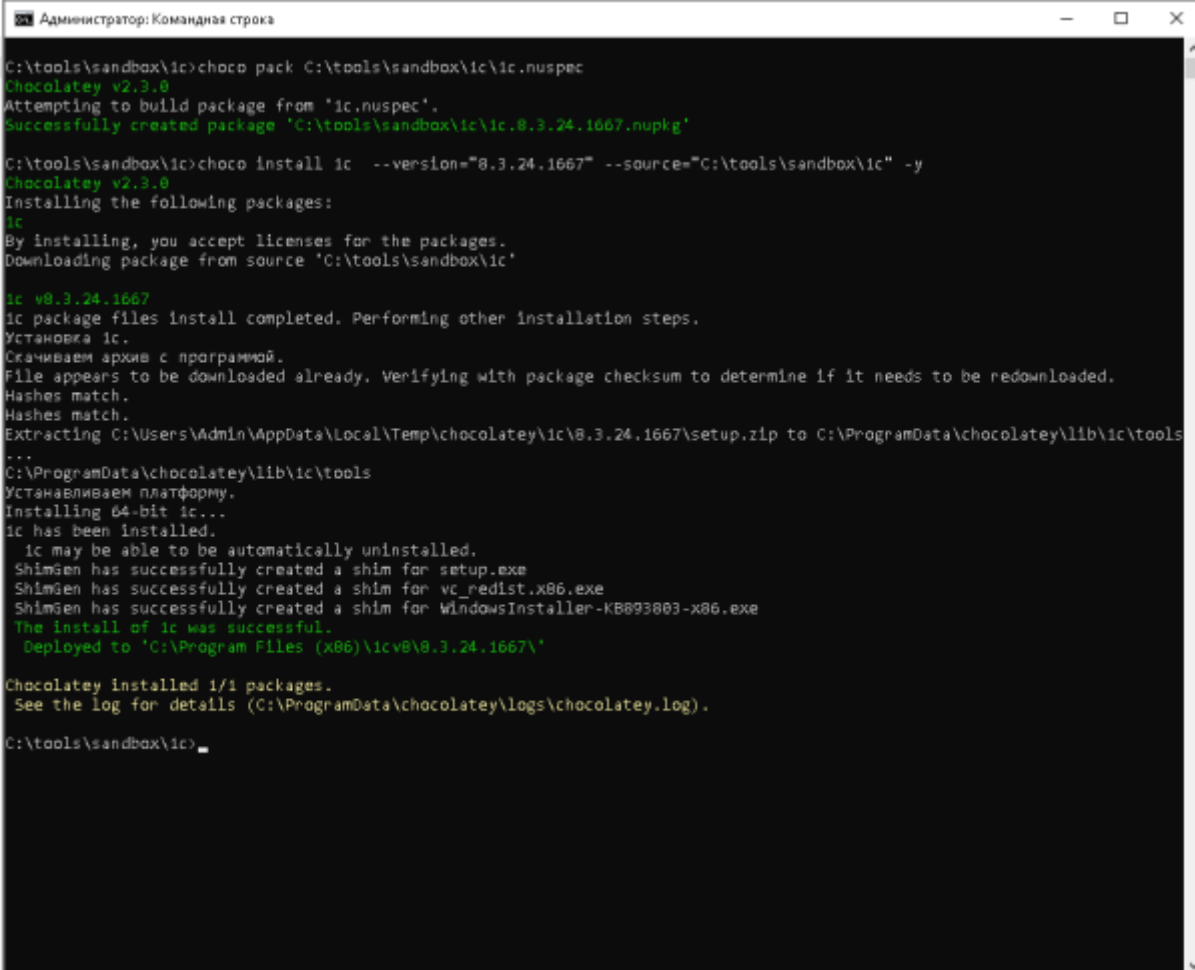
Сначала упаковываем наш пакет командой

```
choco pack C:\tools\sandbox\1c\1c.nuspec
```

А потом пробуем его установить

```
choco install ic --version="8.3.24.1667" --source="C:\tools\sandbox\ic" -y
```

Как видим, всё проходит успешно



```
Администратор: Командная строка
C:\tools\sandbox\ic>choco pack C:\tools\sandbox\ic\ic.nuspec
Chocolatey v2.3.0
Attempting to build package from 'ic.nuspec'.
Successfully created package 'C:\tools\sandbox\ic\ic.8.3.24.1667.nupkg'

C:\tools\sandbox\ic>choco install ic --version="8.3.24.1667" --source="C:\tools\sandbox\ic" -y
Chocolatey v2.3.0
Installing the following packages:
ic
By installing, you accept licenses for the packages.
Downloading package from source 'C:\tools\sandbox\ic'

ic v8.3.24.1667
ic package files install completed. Performing other installation steps.
Установка ic.
Скачиваем архив с программой.
File appears to be downloaded already. Verifying with package checksum to determine if it needs to be redownloaded.
Hashes match.
Hashes match.
Extracting C:\Users\Admin\AppData\Local\Temp\chocolatey\ic\8.3.24.1667\setup.zip to C:\ProgramData\chocolatey\lib\ic\tools
...
C:\ProgramData\chocolatey\lib\ic\tools
Устанавливаем платформу.
Installing 64-bit ic...
ic has been installed.
ic may be able to be automatically uninstalled.
ShimGen has successfully created a shim for setup.exe
ShimGen has successfully created a shim for vc_redist.x86.exe
ShimGen has successfully created a shim for WindowsInstaller-KB893803-x86.exe
The install of ic was successful.
  Deployed to 'C:\Program Files (x86)\icv8\8.3.24.1667\'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

C:\tools\sandbox\ic>
```

Шаг 10

А теперь можем отправить наш пакет на сервер Chocolatey

```
choco push C:\tools\sandbox\ic\ic.8.3.24.1667.nupkg --source=http://choco.mihanik.net/chocolatey --apikey="бла-бла-бла" --force
```



```
Администратор: Командная строка
C:\tools\sandbox\ic>choco pack C:\tools\sandbox\ic\ic.8.3.24.1667.nupkg --source=http://choco.mihanik.net/chocolatey --apikey="бла-бла-бла" --force
Chocolatey v2.3.0
Attempting to push ic.8.3.24.1667.nupkg to http://choco.mihanik.net/chocolatey
[Debug] You are raising the "push" operation with an "HTTP" source, "http://choco.mihanik.net/chocolatey/". Non-HTTPS access will be removed in a future version. Consider migrating to an "HTTPS" source.
ic.8.3.24.1667.nupkg was pushed successfully to http://choco.mihanik.net/chocolatey

C:\tools\sandbox\ic>
```

Шаг 11

Теперь смело можем устанавливать нашу платформу на ПК пользователей командой

```
choco install lc --version="8.3.24.1667" -y
```

Источники

- <https://docs.chocolatey.org/en-us/create/create-packages-quick-start/>

↑ Наверх ↑



В моей WIKI постоянно ведётся какая-то работа со статьями.
Если у вас возникли вопросы или замечания,
можете их отправлять на почту **support@mihanik.net**

