

## AppendVersion\_X64

AppendVersion\_x64 is a simple program intended to expedite the systematic renaming of executable installation files, compressed installation files and Start Menu links that point to installed executable program files.

In this discussion, 'target executable file' can refer to an uncompressed installation executable, or to the compressed installation executable file inside a packaged compressed file (zip, 7z, rar or tar), or to the executable file that a Start Menu link points to. A 'rename target' can refer, again, to an uncompressed installation executable file, or to the compressed file containing a packaged target executable file, or to a Start Menu link pointing to a target executable file.

There are four scenarios for AppendVersion processing:

- An installation executable file is renamed and appended with that target executable file's own file version number.

- A compressed installation file (zip, 7z, tar, or rar) is renamed and appended with its compressed target executable file's own file version number.

- A Start Menu link pointing to its target executable file is renamed and appended with that target executable file's own version number.

- A folder is selected containing any assorted collection of rename target types.

Having an archived program installation file renamed with its target executable file's file properties version number appended to the installation file's name, in a systematic format, aids in quickly determining whether or not you have the latest version installed.

Having all the Start Menu program links (rename targets) appended with their target executable file's version numbers quickly aids in determining whether or not you have the most recent published upgrade or update of that program installed. Hovering your mouse over a Start Menu link will conveniently display the target executable file's version number.

Start Menu links exist in two system folders:

- @ProgramsDir (%appdata%\Microsoft\Windows\Start Menu\Programs)

- @ProgramsCommonDir (%programdata%\Microsoft\Windows\Start Menu\Programs)

Renaming Start Menu links in these two folders require Administrator privilege. You can either compile this AppendVersion.au3 script with the #RequireAdmin macro at the top of the script, or go to the compiled executable 'AppendVersion\_x64.exe' file's Properties Compatibility tab and check 'Run this program as an administrator'.

====

AppendVersion can also receive full file path argument parameters via Run or the Windows terminal. This allows for processing files from different directory locations in one AppendVersion session.

====

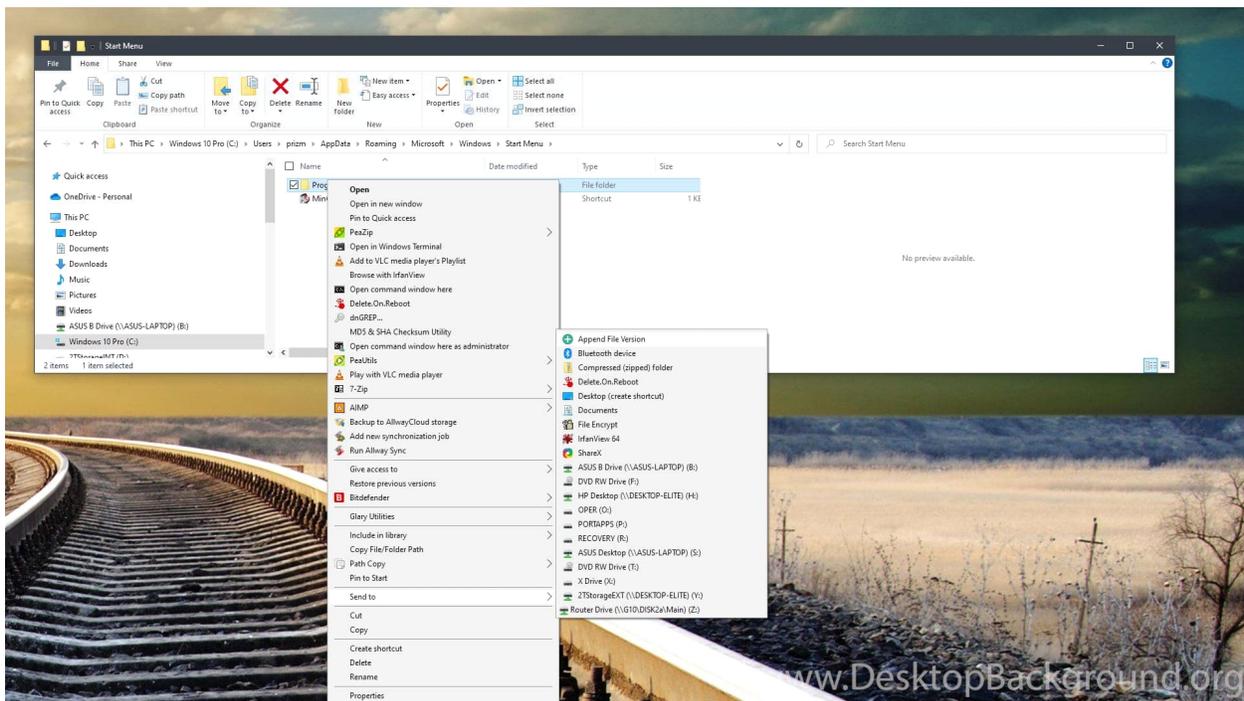
AppendVersion is initiated via a Windows SendTo 'AppendVersion' link. Running the compiled executable 'AppendVersion\_X64.exe' once will auto-create an 'AppendVersion' link in the SendTo subfolder if the SendTo link does not exist and will also create a default Settings.ini file in the @ScriptDir folder if the Settings.ini file does not exist. The Settings.ini file maintains basenames and other configuration data that AppendVersion requires.

====

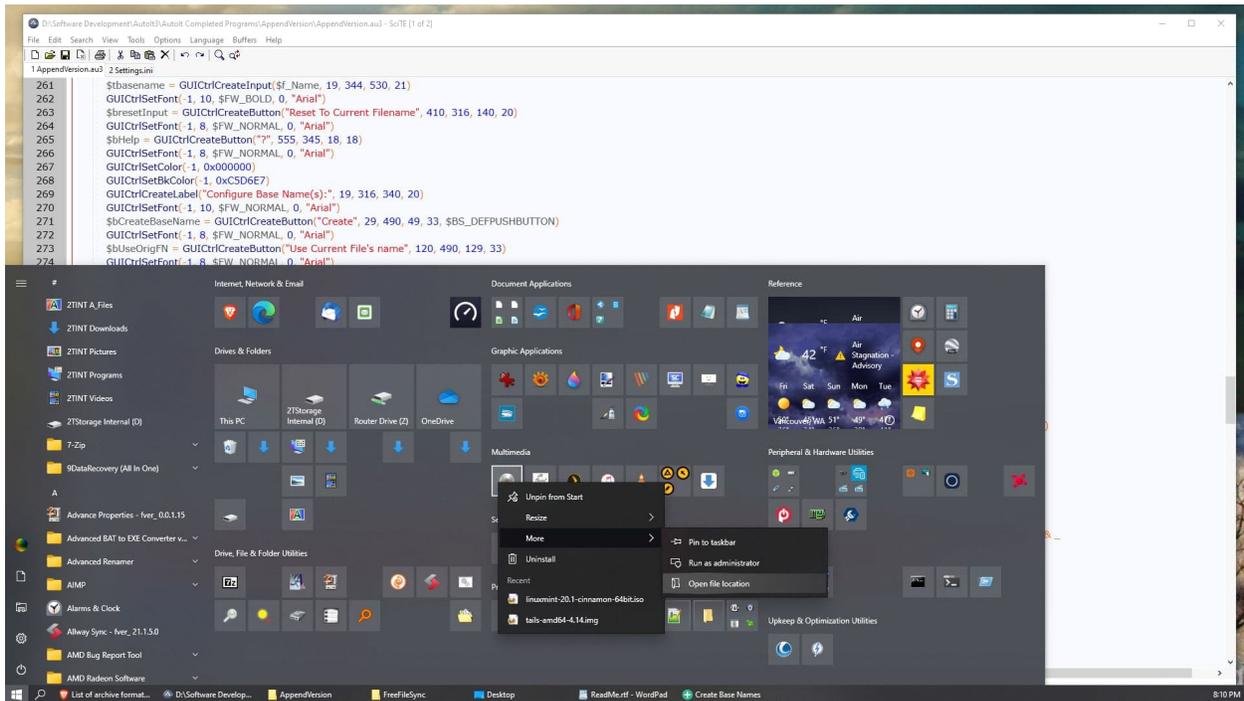
Icon @ScripDir subfolder location directive for compiling AppendVersion\_x64.au3 is:  
#Autolt3Wrapper\_Icon=.\\Icon\\add2.ico

====

To use AppendVersion, in Windows Explorer select one or more rename target files, rename target links and any folders containing rename target files or links, then select the SendTo 'AppendVersion' link.



You can also right-click on a Start Menu link and select Open Location:



After Windows Explorer opens, right-click on the highlighted rename target link and then select 'AppendVersion' from the SendTo menu.

You can also process Start menu rename target links enmasse in succession by selecting one of the two Programs folders where Windows maintains most all the Start Menu links:

%appdata%\Microsoft\Windows\Start Menu

%programdata%\Microsoft\Windows\Start Menu

Select one of the two Programs folders and then select 'AppendVersion' from the SendTo menu

You can process both Programs folders in one session by running the compiled executable 'AppendVersion Start Menu Links\_x64.exe'.

====

The script maintains processed file identification data in the Settings.ini [BaseNames] section. This section is used in identifying new updated versions of earlier files previously processed by AppendVersion. This file identification data (key basenames and fingerprint value) is used to compare and systematically rename later matching updated versions of the identified file.

```
Settings.ini - Notepad
File Edit Format View Help
[QuotedPreFix]
fver=" - fver_"
pver=" - pver_"
uver=" - uver_"

[BaseNames]
balenaEtcher-Portable=balenaEtcherBalenaInc.FlashOSImagestoSDcardsandUSBdrives,safelyandeasily.040904E4
CDBurnerXP=cdbxpp.exeCDBurnerXPCanneverbeLimitedCDBurnerXPAnapplicationtoburnaudioanddatadiscs000004B0
DataRecovery_all_in_one=9DataRecovery(AllInOne)9DataRecovery.com9DataRecovery(AllInOne)ThisinstallationwasbuiltwithInnoSetup.000004B0
Find.Same.Images.OK.Portable_x64=
FreeFileSync_[Donation_Edition]=FreeFileSync[DonationEdition]FreeFileSync.orgFreeFileSync[DonationEdition]SetupThisinstallationwasbuiltwithInnoSetup.000004B0
h2testw=h2testw.exeH2testw.exeH2testw.cMagazinFürComputertechnikIntegritätstestFürUSB-SticksundandereDatenträger040704E4
imageusb=imageUSBimageUSB.exeimageUSBApplicationPassMarkSoftwareimageUSB040904B0
innosetup=innosetupjrssoftware.orgInnoSetupSetupThisinstallationwasbuiltwithInnoSetup.
Intel-Driver-and-Support-Assistant-Installer=setupIntel-Driver-and-Support-Assistant-Installer.exeIntel®Driver&SupportAssistantIntelIntel®Driver&SupportAssistant040904E4
MSERT=MSERT.exeMSERT.exeMicrosoftMalwareProtectionMicrosoftCorporationMicrosoftSupportEmergencyResponseTool
notepad++.portable.x64=notepad++.exenotepad++.exenotepad++DonH0don.h@free.frNotepad++:afree(GPL)sourcecodeeditor040904B0
paint.net=SetupSfxSetupSfx.exePaint.netPaint.netSetup040904B0
.....
```

'BaseName(s)' is one or two substrings in the installation file that usually does not change in the rename target file name from one file version to the next.

Except for file size and file/product version properties, which can change with every new file version, the remaining available target file properties are concatenated together to create a target file properties 'fingerprint'. The fingerprint consists of file properties that usually will not change from one target file version to the next, i.e., 'File Description', 'Product Name', 'Copyright', 'Language', 'Original File name', 'Legal Trademarks', 'Company Name' and 'DefaultLangCodePage'.

The basename(s) and fingerprint together constitute the 'identification' of a candidate file. This ID is used to identify later downloaded versions of the file that can be automatically renamed without user interaction.

If the target executable file's properties does not have a file version number, the file's product version number will be used instead in the version appending and renaming of the file.

If the target executable has neither a file version nor product version number, the user will be prompted to enter the published file version number acquired from the provider's web site.

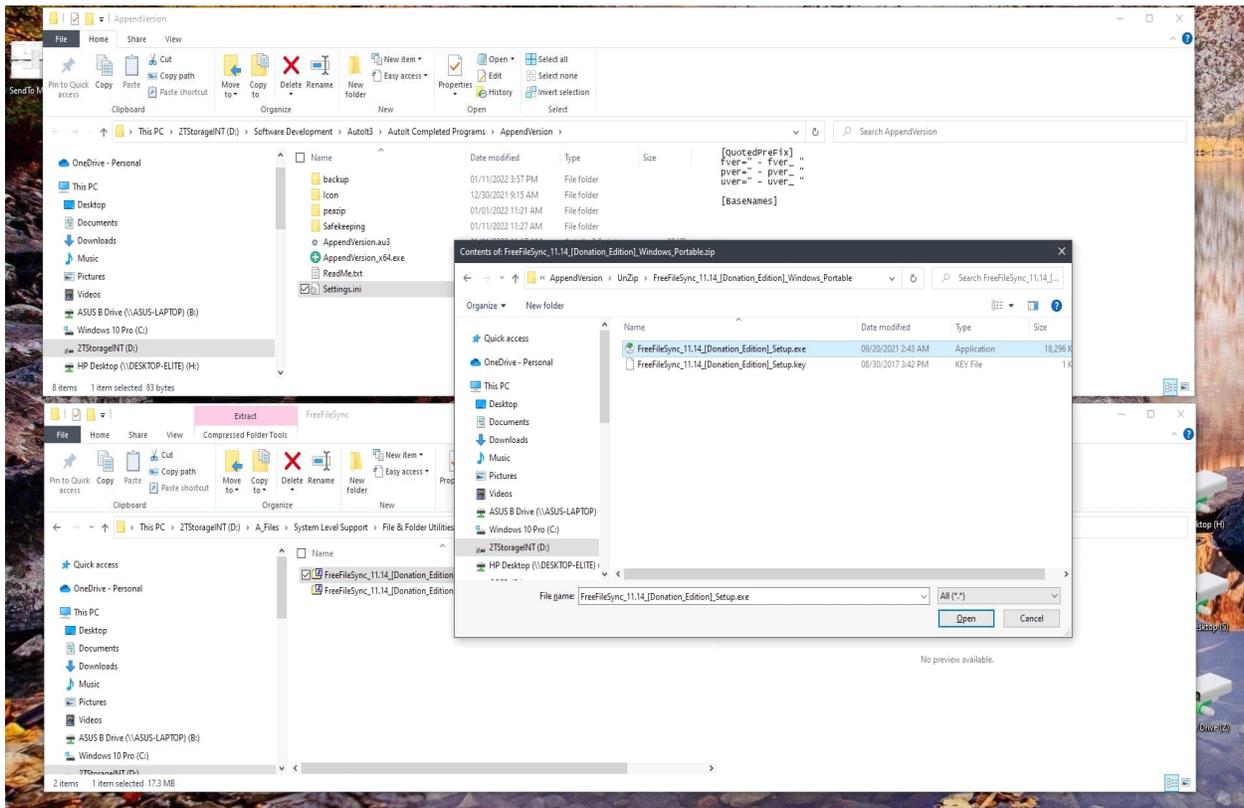
Both the basename(s) and version numbers will be used for the renaming of the file.

If the rename target file is in a popular compressed format, i.e., (\*.zip, \*.7z, \*.rar, and \*.tar), the script will use PeaZip-portable to decompress the zipped file into a temporary folder (@ScripDir & "\UnZip") and then prompt the user to select the target executable file with file properties used in the rename target 'identification'.

The freeware PeaZip portable can be downloaded from:

<https://peazip.github.io/peazip-portable.html>

Unpack PeaZip portable into the @ScripDir & "\peazip" subfolder.

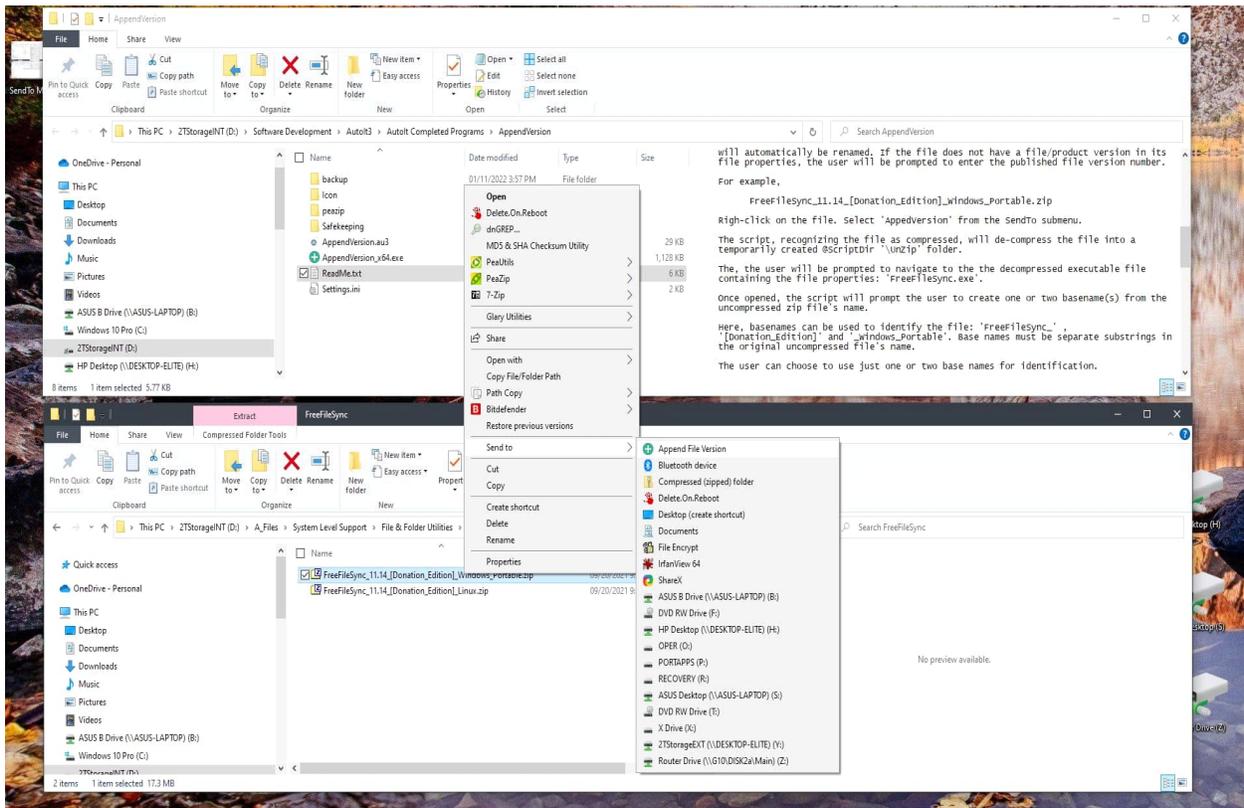


Once the target executable file properties are read from the target executable, the user will be prompted to configure the basename(s) from the rename target file name, unless an identification already exists in Settings.ini, in which case the rename target file will automatically be renamed by AppendVersion. The process is not always fully automatic. If the target executable file has neither a file nor product version in its file properties, the user will be prompted to manually enter the published file version number.

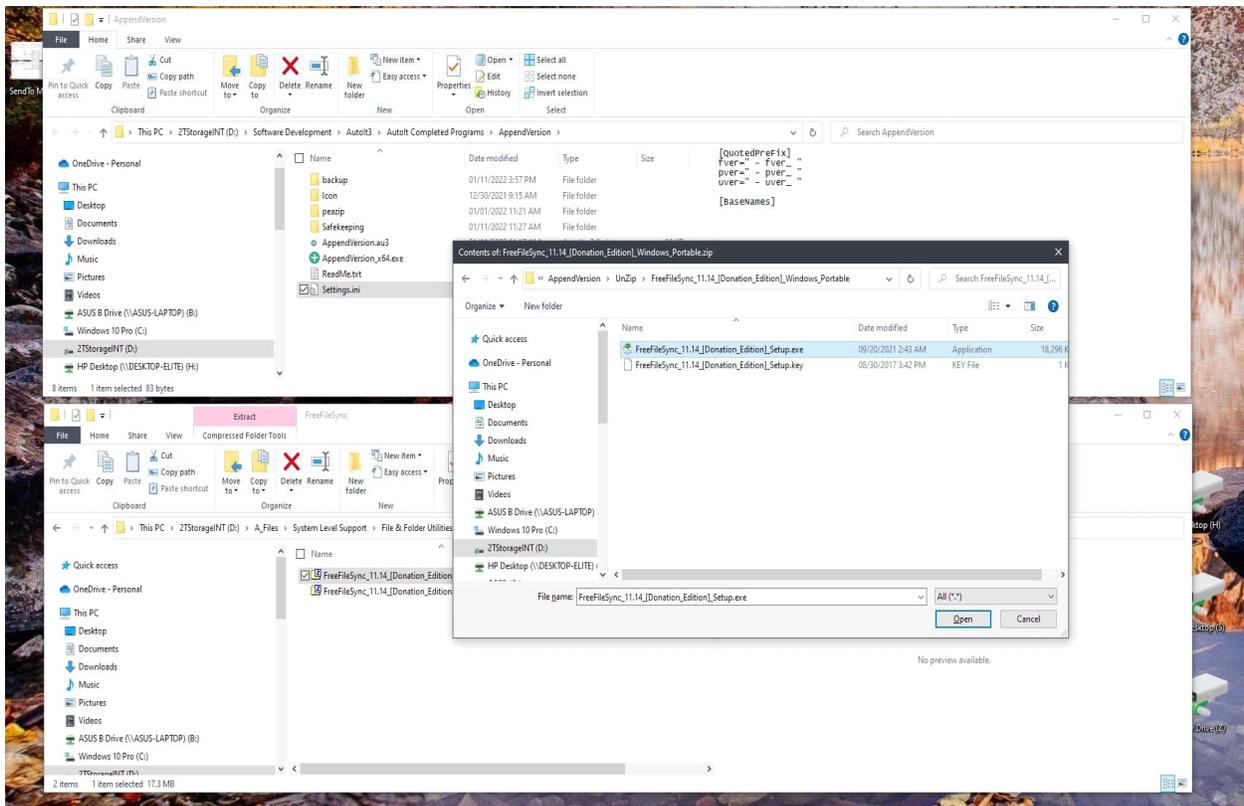
Process example, the target file for renaming is:

FreeFileSync\_11.14\_[Donation\_Edition]\_Windows\_Portable.zip

Right-click on the file. Select 'AppedVersion' from the SendTo submenu.



The script, recognizing the rename target file as a compressed file (\*.zip, \*.7z, \*.tar, and \*.rar), will de-compress the rename target file into a temporarily created @ScriptDir '\UnZip' folder.

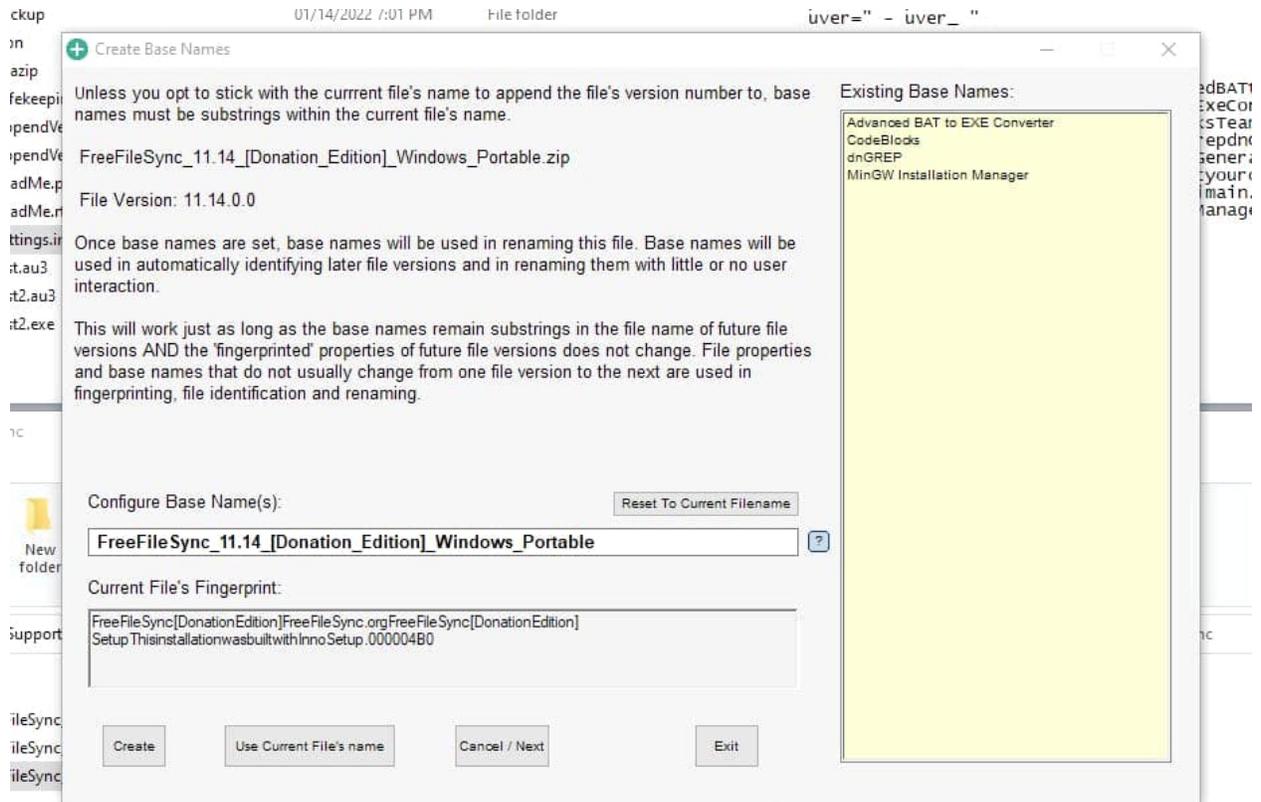


Then, the user will then be prompted to select from the folder or subfolder containing the target

executable file from which the file properties will be read:

'FreeFileSync\_11.14\_[Donation\_Edition]\_Setup.exe'.

Once the target executable file is selected and opened, the script will prompt the user to create one or two basename(s) from the rename target zip file's name.



Base name(s) are created from the rename target zip file name (without extension):

Examples, 'FreeFileSync\_', '[Donation\_Edition]' and '\_Windows\_Portable'.

Base names must be separate substrings in the rename target zip file's name.

The user can choose to use just one or two base names for file identification.

The user can also choose to use the entirety of the original rename target zip file name: 'FreeFileSync\_11.14\_[Donation\_Edition]\_Windows\_Portable' as a base name in renaming later versions of FreeFileSync. This may not be advisable with this base name, since later versions of this rename target file name will not have version '11.14' in the name, but this naming convention is an option.

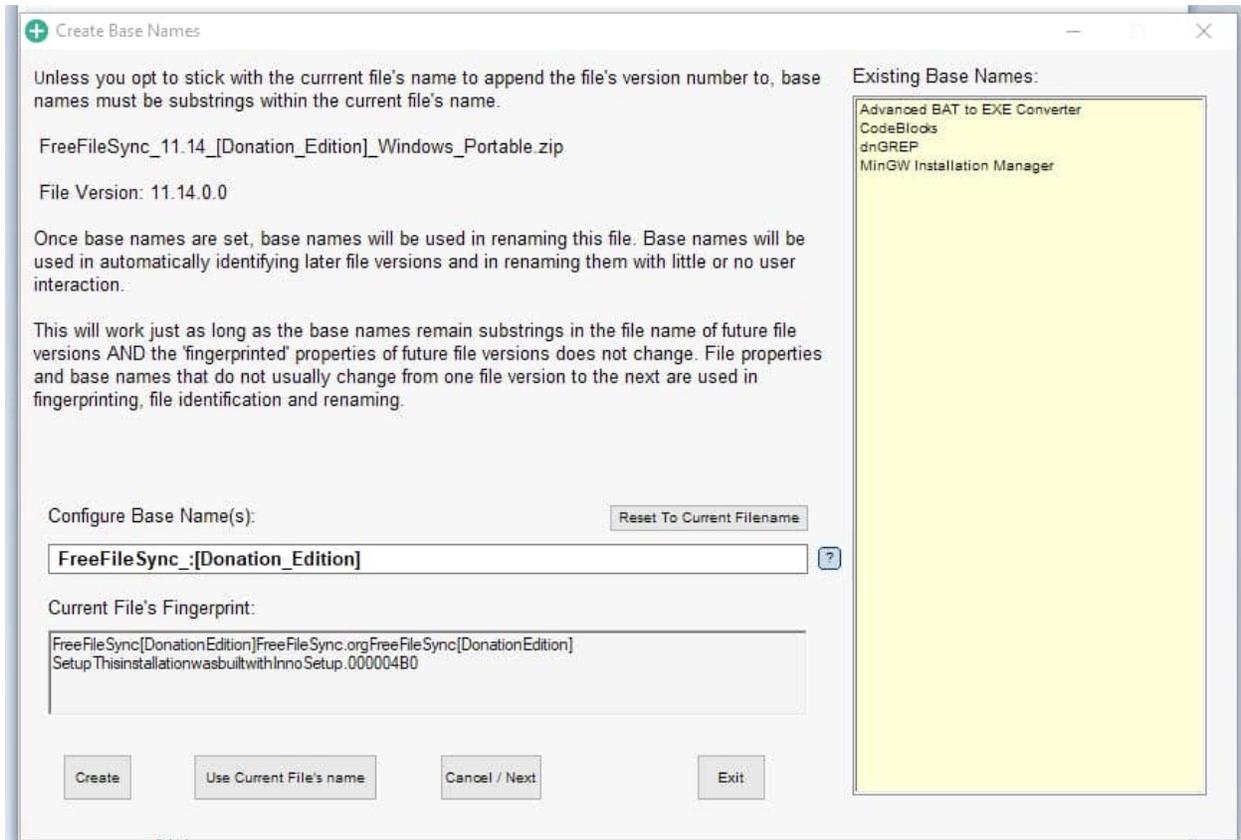
In this example, I will use the 'FreeFileSync\_' and '[Donation\_Edition]' substrings as base names, since these are file name structures that are unique to the file name and will probably not change in later published file version names.

The file properties fingerprint collected from the FreeFileSync.exe executable is:

FreeFileSyncFreeFileSync.orgFreeFileSync-FolderComparisonandSynchronization040904B0

To use two base names for later automatic file version identification, the base names must be

separated with a colon ':', i.e., 'FreeFileSync\_[Donation\_Edition]'.



If a later file version of a candidate rename target file's name contains both of these substrings and has the same target executable file properties 'fingerprint', the script will automatically identify and rename the later-versioned file's name.

The combined file identification of [Key] base names + [Value] fingerprint is:

FreeFileSync\_[Donation\_Edition]=FreeFileSyncFreeFileSync.orgFreeFileSync-FolderComparisonandSynchronization040904B0

This identification is stored in the Settings.ini file under the [BaseNames] section.



When a new version of FreeFileSync\_XX.XX\_[Donation\_Edition]\_Windows\_Portable.zip is processed by

AppendVersion, if an entry in Settings.ini [BaseNames] section can match the rename target's file name and also match the candidate target executable's fingerprint, the script will use this to automatically rename the zip file.

Finally, the rename target file is then renamed with the base name(s) and with one of three version prefixes to the version number:

" - fver\_ "; for files with a properties file version number (preferred default)

" - pver\_ "; for files without file version numbers but with product version numbers(fallback)

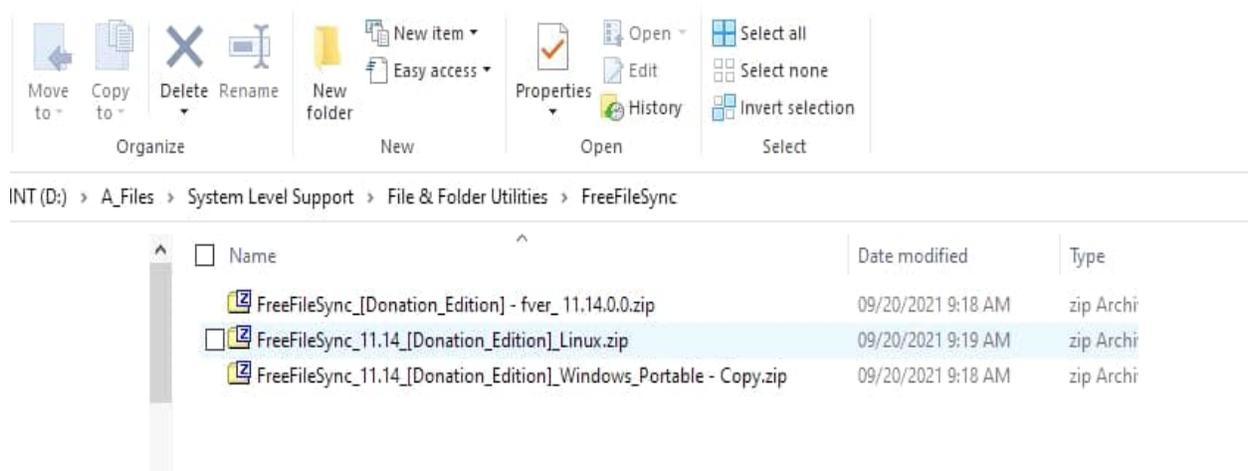
" - uver\_ "; for files with neither file nor product numbers. User manually entered published file version number.

NOTE: These version format prefixes can be changed in Settings.ini [QuotedPrefix] section to meet your prefix format preferences. The [QuotedPrefix] section value entries must be quoted.

In this example, FreeFileSync.exe contains a properties file version number of 11.14.0.0

The resultant AppendVersion renamed file is:

FreeFileSync\_[Donation\_Edition] - fver\_ 11.14.0.0.zip



This systematic approach works well with most file names, but files with short file names such as 'a.exe' and having no finger-printable file properties are problematic. In such case, perhaps renaming the downloaded file with a longer, unique name before running this script on it is preferable, although renaming the file before processing it with AppendVersion circumvents the purpose of the script. This would not be an issue if software creators would always include full file properties in their file compilations.

