

This tutorial is made for the mount point `../..../` which is the default for Switch games. More info in the tutorial.

REQUIREMENTS (download these):

- "UnrealPakSwitch.zip" from the attachments, extract it after downloading
- Notepad++ (or any other text editor with Perl regular expressions)
- Python 2 or 3
- Unreal Engine 4.19.2 (for v4 *.pak)
- Unreal Engine 4.20.3 (for v5 *.pak)
- Unreal Engine 4.21.2 (for v7 *.pak)
- Unreal Engine 4.22.3 (for v8 *.pak)
- Unreal Engine 4.24.3 (for v8.23 *.pak)
- Unreal Engine 4.25.4 (for v9 *.pak)
- Unreal Engine 4.26.2 (for v11 *.pak).

There are instructions later on in the **CHECKING THE PACKAGE VERSION** tab on how to tell which version .pak file your game uses.

For a complete pack you need to download all seven of them.

My package lacks UnrealPak.exe which is licensed, but free to use. That's why I don't upload them.

After downloading each respective Unreal Engine, you can find UnrealPak.exe in:

1. `X:\Program Files\Epic Games\UE_4.19\Engine\Binaries\Win64\unrealpak.exe`
2. `X:\Program Files\Epic Games\UE_4.20\Engine\Binaries\Win64\unrealpak.exe`
3. `X:\Program Files\Epic Games\UE_4.21\Engine\Binaries\Win64\unrealpak.exe`
4. `X:\Program Files\Epic Games\UE_4.22\Engine\Binaries\Win64\unrealpak.exe`
5. `X:\Program Files\Epic Games\UE_4.24\Engine\Binaries\Win64\unrealpak.exe`
6. `X:\Program Files\Epic Games\UE_4.25\Engine\Binaries\Win64\unrealpak.exe`
7. `X:\Program Files\Epic Games\UE_4.26\Engine\Binaries\Win64\unrealpak.exe`

For number 3 (UE_4.21) you need to copy unrealpak.exe AND the DLLs and other files listed below (treat UnrealPak.exe and all files below as one if you need to use the **EXCEPTIONS** tab later in the tutorial below):

UnrealPak-BuildSettings.dll
 UnrealPak-Core.dll
 UnrealPak-Json.dll
 UnrealPak-PakFile.dll
 UnrealPak-PakFileUtilities.dll
 UnrealPak-Projects.dll
 UnrealPak.modules
 UnrealPak.exe

For number 4 (UE_4.22) you need these additional files:

UnrealPak-BuildSettings.dll
 UnrealPak-Core.dll
 UnrealPak-Json.dll
 UnrealPak-PakFile.dll
 UnrealPak-PakFileUtilities.dll
 UnrealPak-Projects.dll
 UnrealPak.modules
 UnrealPak-RSA.dll
 UnrealPak.exe

For number 5 (UE_4.24) and number 6 (UE_4.25) you need these additional files:

UnrealPak-BuildSettings.dll
 UnrealPak-Core.dll
 UnrealPak-DerivedDataCache.dll
 UnrealPak-Json.dll
 UnrealPak-PakFile.dll
 UnrealPak-PakFileUtilities.dll
 UnrealPak-Projects.dll
 UnrealPak.modules
 UnrealPak-RSA.dll
 UnrealPak-TraceLog.dll
 UnrealPak.exe

For number 7 (UE_4.26) you need these additional files:

UnrealPak-BuildSettings.dll
 UnrealPak-Core.dll
 UnrealPak-DerivedDataCache.dll
 UnrealPak-Json.dll
 UnrealPak-PakFile.dll
 UnrealPak-PakFileUtilities.dll
 UnrealPak-Projects.dll
 UnrealPak.modules
 UnrealPak-RSA.dll
 UnrealPak-SSL.dll
 UnrealPak-TraceLog.dll
 UnrealPak.exe

For each one version you need to copy UnrealPak.exe to the correct folder of my package:

1. *UnrealPakSwitch\v4\2\3\UnrealPak.exe*
2. *UnrealPakSwitch\v5\2\3\UnrealPak.exe*
3. *UnrealPakSwitch\v7\2\3\UnrealPak.exe* (+ other files)
4. *UnrealPakSwitch\v8\2\3\UnrealPak.exe* (+ other files)
5. *UnrealPakSwitch\v8.23\2\3\UnrealPak.exe* (+ other files)
6. *UnrealPakSwitch\v9\2\3\UnrealPak.exe* (+ other files)
7. *UnrealPakSwitch\v11\2\3\UnrealPak.exe* (+ other files)

You can now uninstall Unreal Engine(s) - they are no longer needed.

WARNING!

For some .paks, UnrealPak shows an error about not detecting some plugin. In most cases it's not needed, just ignore it and it will continue.

CHECKING THE PACKAGE (.PAK) VERSION

Archives are usually inside *romfs/*****/Content/Paks* as *.pak files

***** is the project name made by the devs. It's different per-game (for example Octopath has "Kingship"), but most games have one folder in *romfs*

1. Copy the .pak file you want to unpack into the UnrealPakSwitch folder.
2. Drag and drop your file to "1. check.cmd"

It will show Version: X, Where X is a version of your package.

UNPACKING

If you know what version you have, then drag and drop your .pak file to the correct unpack.cmd:

version 4 - "unpack-v4.cmd"

version 5 - "unpack-v5.cmd"

version 7 - "unpack-v7.cmd"

version 8 - "unpack-v8.cmd"

version 8.23 - "unpack-v8-23.cmd"

version 9 - "unpack-v9.cmd"

version 11 - "unpack-v9.cmd"

This creates a "lista.txt" in the UnrealPakSwitch folder which contains info about all the files inside the .pak. It will be needed for re-packing the archive.

DO NOT press a key to continue once it is finished.

First, open "lista.txt" and check if there is a line like this:

LogPakFile: Display: Mount point ../../..

If it looks like the line above, press any key to continue. If it does not (something different after *Mount point* or a lack of it), close the window and go to the Exceptions tab.

After some time it will unpack all the files, in most cases to 2 folders:

- Engine

- ***** (yes, the same ***** as from previous tab)

***** contains the assets. **Engine** should not be modified unless you know what you are doing.

PACKING THE FILES

First, we need to edit "lista.txt" with Notepad++

1. Delete the following lines (don't worry if one or more lines are missing)

LogPaths: Warning: No paths for game localization data were specified in the game configuration.

LogInit: Warning: No paths for engine localization data were specified in the engine configuration.

LogPakFile: Display: Using command line for crypto configuration

LogPakFile: Display: Added 0 entries to add to pak file.

LogPakFile: Display: Mount point ../../..

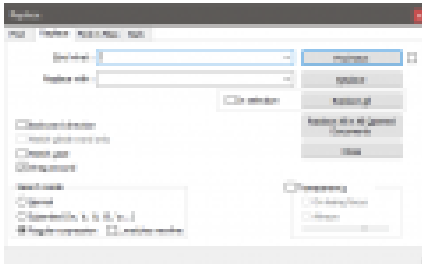
2. Delete the last three lines. It will look something like this (quantity of files, size and seconds may differ):

LogPakFile: Display: 103018 files (1944681023 bytes), (1944681023 filtered bytes).

LogPakFile: Display: Unreal pak executed in 2.605882 seconds

The third line is empty, you need to delete those three lines INCLUDING the empty third line

3. Run "Find & Replace" (Ctrl+H), Check "Wrap Around" and "Regular Expression"



4.1. For **Find what:** paste

LogPakFile: Display: "

Replace with:

"../..../"

Yes - our mount point is the quotation mark at the beginning.

Click **Replace All**

4.2. For **Find what:** paste

" offset: (.*)\$

Replace with: "

Click **Replace All**

And save.

Now we have a file ready to use for our re-packing.

5. Drag and drop "lista.txt" to the same version of pack.cmd that the original pack used:

version 4 - "pack-v4.cmd"

version 5 - "pack-v5.cmd"

version 7 - "pack-v7.cmd"

version 8 - "pack-v8.cmd"

version 8.23 - "pack-v8-23.cmd"

version 9 - "pack-v9.cmd"

version 11 - "pack-v11.cmd"

It will create a "newpak.pak" inside the UnrealPakSwitch folder. Rename it to the name of original .pak file.

CREATING THE PATCH

You can create a .pak file that will be treated as a patch and doesn't need all files that were in the original .pak file.

1. You need have done steps 1 through 4.2 above, skip step 5.

2. Now, in "lista.txt", you need to delete every line that contains a path to the non-edited files (the files that you didn't edit) and save lista.txt.

3. Drag and drop "lista.txt" to the same version of pack.cmd that the original pack used:

version 4 - "pack-v4.cmd"

version 5 - "pack-v5.cmd"

version 7 - "pack-v7.cmd"

version 8 - "pack-v8.cmd"

version 8.23 - "pack-v8-23.cmd"

version 9 - "pack-v9.cmd"
 version 9 - "pack-v11.cmd"

It will create a "newpak.pak" inside the UnrealPakSwitch folder. Rename it to name of original .pak file, but add this to the end of filename:

_p

For example, Kingship-Switch.pak -> **Kingship-Switch_p.pak**

And put it to the same folder as your original .pak

Not every game accepts patches in this form. For now, I found only Octopath Traveler crashes when you try to put a .pak file without the "proper" name, it needs to be named like this to work:

Kingship-Switch_1_P.pak

Kingship-Switch_2_P.pak

Etc

EXCEPTIONS

Now we need to consider 2 cases: either ONLY your mount point is different or your .pak is encrypted. If in lista.txt there is ANYTHING about encryption, go to the "Decryption" tab below. Otherwise, continue in this tab.

Sometimes, the mount point can be different and my package needs to be slightly modified if you don't want to make a mess. In most cases it means that the .pak archive is built with other .pak file(s) and is "inside" one of them, so after unpacking it will look different than the normal Engine and ***** folders.

You can check it in "lista.txt"

The default mount point is ../../../../ which is "3 folders before", that's why I put UnrealPak.exe in such a weird combination of folders.

Mount point is saved in archive properties and if it's not preserved, the game will show an error.

For example:

If it is a version 4 package with mount point ../../King/Sample/:

1. You need to move UnrealPak.exe from v4\2\3 to v4\2\ (only two folders back)

2. Change in pack-v4.cmd:

v4\2\3\UnrealPak.exe ..\..\..\newpak.pak

to

v4\2\UnrealPak.exe ..\..\newpak.pak

and in unpack-v4.cmd:

v4\2\3\UnrealPak.exe

to

v4\2\UnrealPak.exe

Unpacked folders need to be moved to

UnrealPakSwitch\King\Sample

For step 4.1 in the **PACKING THE FILES** tab above, you will need to change **Replace with:** to

../../King/Sample/

Now you can go back to previous tabs and continue with the knowledge that some files are now different, but the steps are the same if you take the information from here and change the steps in previous tabs accordingly.

DECRYPTION

In some cases, devs will decide to encrypt the index of a .pak file and because of that we can't easily unpack all the files. Some multiplatform games are using the same key, so if your game also has a PC version, try to find the key on google (there are sites specializing in extracting keys from PC exes). But there is a method on how to get this decryption key from a Switch executable. For more go here:

<https://gbatemp.net/threads/how-to-extract-decryption-key-for-unreal-engine-4-pak-files.574916/>

If you have a key in 32 bytes hex format like this:

12113456789012344321098765432100363636363636363636363636363636363621

We need to convert it to Base64. There are many sites converting hex to Base64, like this one

<https://base64.guru/converter/encode/hex>

So in my case I got:

EhE0VniQEjRDIQmHZUMhADY2NjY2NjY2NjY2NjY2NiE=

Now go to the UnrealPakSwitch folder, edit "crypto.json" and replace **DUMMY** with our Base64 key and save it. This is how it will look in the end:

```
{
  "$types": {
    "UnrealBuildTool.EncryptionAndSigning+CryptoSettings, UnrealBuildTool, Version=4.0.0.0, Culture=neutral, PublicKeyToken=null": "1",
    "UnrealBuildTool.EncryptionAndSigning+EncryptionKey, UnrealBuildTool, Version=4.0.0.0, Culture=neutral, PublicKeyToken=null": "2"
  },
  "$type": "1",
  "EncryptionKey": {
    "$type": "2",
    "Name": null,
    "Guid": null,
    "Key": "EhE0VniQEjRDIQmHZUMhADY2NjY2NjY2NjY2NjY2NiE="
  },
  "SigningKey": null,
  "bEnablePakSigning": false,
  "bEnablePakIndexEncryption": true,
  "bEnablePakIniEncryption": true,
  "bEnablePakUAssetEncryption": false,
  "bEnablePakFullAssetEncryption": false,
  "bDataCryptoRequired": true,
  "SecondaryEncryptionKeys": []
}
```

Go back to the **UNPACKING** tab above, but instead of using "unpack-*****.cmd" we need to use "unpack-encrypted-*****.cmd".

UFONT

*.ufont file is a ttf font with a different extension

If your file is, for example:

Ariadna.ufont

change the name to

Ariadna.ttf

You can edit it without worry if you don't change the config of this font inside a font editor.

UEXP, UASSET, UBULK

If the asset contains only uexp and uasset:

*.uexp is an archive for data - by default it's not compressed nor encrypted, but it can be configured by the devs if they want to modify the way files are packed.

*.uasset has information about files inside .uexp - their filenames and references - and how big the *.uexp file is:

size in bytes-4 converted to Hex Little Endian

For example:

1966078-4 = FAFF1D

If the asset contains uexp, uasset and ubulk:

*.uasset has information about the files inside uexp - their filenames and references - and how big uexp is

*.uexp is an archive for Editor data, it contains data about the ubulk size and a thumbnail representing the asset

*.ubulk contains the raw asset without any additional info

If you change the size of the *.uexp file and you don't change the value inside *.uasset, the game will not load. If you want to serialize information included in them (for example, stats of equipment), I recommend John Wick Parse. It extracts supported data to json files.

<https://github.com/SirWaddles/JohnWickParse/releases>

LOCRES

*.locres - localization files. This is the default way of saving strings, but it's not often used by devs. They prefer to create their own format and put it in a *.uexp file (for example, Octopath Traveler).

OODLE COMPRESSION

Some games may use Oodle compression for .pak files (v8 can generate a filelist with info about the used compression method, if it's not a custom method).

Unrealpak is then not usable without a licensed, paid Oodle SDK. But for unpacking you can use QuickBMS

<https://aluigi.altervista.org/quickbms.htm>

With this script

<https://gbatemp.net/threads/unpacking-sao-fatal-bullet-pak.545478/#post-8748129>

You don't need Oodle for packing, the default compression (zlib) will be readable by the game, so you can follow this tutorial for packing files.

ZSTD COMPRESSION

Some games may use zstd compression for .pak files (v8 can generate A filelist with info about used the compression method, if it's not a custom method).

Unrealpak is then not usable without compiling the zstd plugin by yourself. But for unpacking, you can use QuickBMS

<https://alugi.altervista.org/quickbms.htm>

With this script

<https://gbatemp.net/threads/how-to-unpack-and-repack-unreal-engine-4-files.531784/page-10#post-9380618>

You don't need zstd for packing, the default compression (zlib) will be readable by the game, so you can follow this tutorial for packing files.

OTHER TUTORIALS

How to swap audio in Unreal Engine 4 games

<https://gbatemp.net/threads/how-to-swap-audio-in-unreal-engine-4-games.535516/>

Tool for extracting and making textures compatible with Switch

<https://gbatemp.net/threads/noesis-mod-tiling-texture-for-ue4.535571/>

THANKS

Thanks to **panzi** for u4pak.py (that I slightly modified for this package)

<https://github.com/panzi/u4pak>