

Windows (WINE)

WINE will play Windows games, both old and recent games, both 32-bit and 64-bit. Some open-source games are available in the free Batocera content store (from the EmulationStation menu, **UPDATES & DOWNLOADS → CONTENT DOWNLOADER**)



The following links contain info about some games' compatibility with WINE:

- [List liberodark](#)
- The [Windows-games-on-Batocera spreadsheet](#). [Direct link to Google sheets document](#).
- [WineHQ AppDB](#)
- [ProtonDB](#)
- [supertuxland](#)



It is recommended to use **btrfs** or **ext4** for /userdata/ as WINE [explains on their FAQ](#) that some games won't run on a non-Linux native file system. Steam games are [notoriously known to be unusable under WINE when using ntfs](#).



In Batocera **v32** and higher there is a bug where WINE cannot run applications/games stored on a NAS. This can be worked around by not using a NAS for the saves folder.

This system scrapes metadata for the “pc” group and loads the windows set from the currently selected theme, if available.

Quick reference

- **Emulator:** [wine](#)
- **Cores available:** [wine: lutris](#), [wine: proton](#)
- **Folder:** /userdata/roms/windows/
- **Accepted ROM formats:** .pc, .exe, .wine, .wsquashfs, .wtgz

BIOS

No Windows emulator in Batocera needs a BIOS file to run.

Program files

- /userdata/roms/windows/ : installed windows game (both 32 and 64-bit).
- /userdata/roms/windows_installers/ : .iso, .msi or setup.exe used to install games into /userdata/roms/windows/.

Games coming with an installer

There are 2 types of formats for a Windows game installer:

- CD-ROM images: An *.iso file created from a game on a CD-ROM
- Executable Installers: A *.exe or *.msi file, that usually contains “setup” or “install” in the name (in some cases, other files may be bundled with them)

In both cases, those files need to be put into the /userdata/roms/windows_installers/ folder. Once this is done, refresh the gamelist in EmulationStation, then go into the “Windows” system, select “Install a new game”, and run the installer.

After a screen announcing that the configuration is being made, the installer should open up. It'll look the same way it does inside Windows.

Proceed to install the game in the normal way, as for the installation path, leave it to the default: The virtual C:\ Drive will be located at /userdata/roms/windows/<game_name>.wine/drive_c/.

Once the installation is done, head over to the /userdata/roms/windows/<game_name>.wine/ folder, and edit the autorun.cmd accordingly.



The file must use Linux line terminators! Do not use Notepad or other similar Windows-oriented text editors; use Notepad++ or Atom and select Unix (LF) as its line termination format.

Example:

```
DIR=drive_c/Program Files/xmoto 0.6.1  
CMD=xmoto.exe --fullscreen
```

or if there are some spaces in the name of the game:

```
DIR=drive_c/Program Files (x86)/Ubisoft/Rayman Origins  
CMD="Rayman Origins.exe"
```

This <game name>.wine folder is called the wineprefix.



If the path isn't working, try putting `./` before the path. Eg. `DIR=./drive_c/Program Files/xmoto 0.6.1`

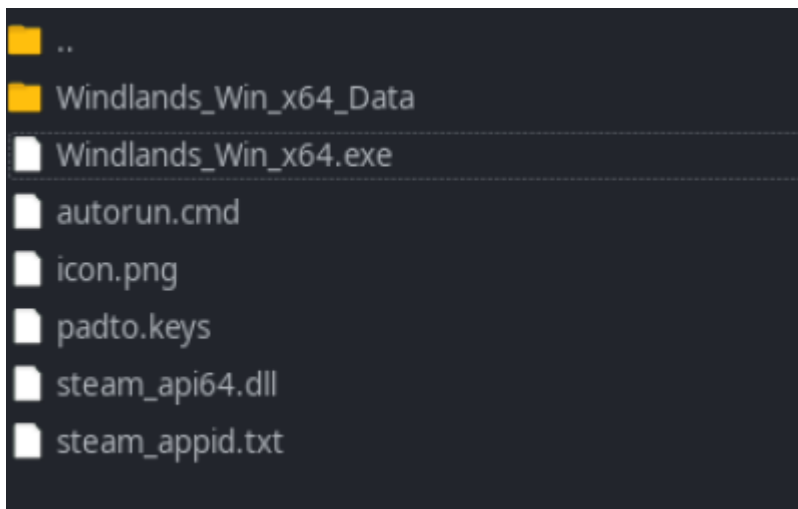
Games that don't need any installation

Those games usually come as a game folder containing a `*.exe` file (or several) as well as other files and subfolders. They may also come as a `*.zip` file, if that is the case, unzip them first.

For those, there are two easy ways to handle them:

- The EXE method: Just put the game folder into the `/userdata/roms/windows`, and refresh the gamelist in EmulationStation
- The PC folder method (recommended): Rename the game folder as `<game name>.pc` (eg. `Windlands.pc`), put it into the `/userdata/roms/windows/`, create a file named `autorun.cmd` directly into it, edit it so it points to the game executable.

Example:



If the `*.exe` of the game is directly in the `<game name>.pc/` folder, and is called `Windlands_Win_x64.exe`, the `autorun.cmd` file should contain:

```
CMD=Windlands_Win_x64.exe
```

If the name of the executable contains special characters or spaces however, be sure to put quotes around it, like so:

```
CMD="launch game.exe"
```

If the EXE is instead inside of a subfolder, the `DIR` command needs to be added first. For example, if the game is at `Game name.pc/64bit/bin/folder` with space in its name/`Game name 64-bit.exe` and requires special launch parameters:

```
DIR=64bit/bin/folder with space in its name
```

```
CMD="Game name 64-bit.exe" --fullscreen
```

The first time a game is run, a <game name>.wine folder will be created at /userdata/saves/windows/proton/ or /userdata/saves/windows/.

This <game name>.wine folder is called the wineprefix.







Emulators

WINE

WINE configuration

Standardized features available to all cores of this emulator: windows.videomode, windows.padokeyboard

ES setting name batocera.conf_key	Description ⇒ ES option key_value
DXVK windows.dxvk	Converts the DirectX 9/10/11/12 calls to Vulkan. Should only be used on hardware that natively supports Vulkan. ⇒ Off 0, On 1.
DXVK HUD windows.dxvk_hud	See your FPS and version of Vulkan API/driver. ⇒ Off 0, On 1.
ESYNC windows.esync	Removes wineserver overhead for synchronization objects. Can increase performance for games that stress the CPU. ⇒ Off 0, On 1.
FSYNC windows.fsync	Enables stricter scheduling policies to improve frame rates and responsiveness. Can cause issues in certain apps. ⇒ Off 0, On 1.
PBA windows.pba	Implements a GL-free GPU heap allocator, vastly improving the speed of buffer maps. ⇒ Off 0, On 1.
FSR windows.fsr	AMD FidelityFX Super Resolution uses advanced scaling technologies to boost FPS by rendering the game at a lower resolution and then upscaling it. ⇒ Off 0, On 1.
NVAPI windows.nvapi	NVAPI is NVIDIA's core software that allows direct access to NVIDIA GPU. ⇒ Off 0, On 1.
FPS LIMITER windows.fps_limit	Sets the FPS limiter to 60. Required by some older titles. ⇒ Off 0, On 1.
MF windows.mf	Installs Media Foundation, require by some apps which use the Microsoft-specific codecs for audio and video to run. ⇒ Off 0, On 1.

ES setting name batocera.conf_key	Description ⇒ ES option key_value
VIRTUAL DESKTOP windows.virtual_desktop	Define the resolution and a new dedicated window. ⇒ Off 0, On 1.
ENABLE MOUSE windows.force_mouse	Show the cursor. ⇒ Off 0, On 1.
ENABLE XIM windows.allow_xim	Enable XIM support. ( Fix Me! what's XIM?) ⇒ Off 0, On 1.
NO WRITE WATCH windows.no_write_watch	Disable support for memory write watches in ntdll. ( Fix Me! what is this used for?) ⇒ Off 0, On 1.
FORCE LARGE ADDRESS windows.force_large_adress	Force WINE to run games with large address. ( Fix Me! what is this used for?) ⇒ Off 0, On 1.
HEAP DELAY FREE windows.heap_delay_free	Delay freeing some memory. ( Fix Me! what is this used for?) ⇒ Off 0, On 1.
HIDE NVIDIA GPU windows.hide_nvidia_gpu	Force Nvidia GPUs to always be reported as AMD GPUs. ( Fix Me! what is this used for?) ⇒ Off 0, On 1.
ENABLE DEBUG windows.wine_debug	Enable wine debug. ( Fix Me! be more descriptive) ⇒ Off 0, On 1.

Advanced options will only be applied on the first launch of the game. Adjusting these values later will not do anything.

If discovered that the wrong settings were used, delete the game's installation from the saves/windows/<game> folder, set up the correct settings, and try launching it again.

Folder compression

Installed games can be converted into two other formats: TGZ and SquashFS. Batocera requires changing the extensions to *.wtgz and *.wsquashfs respectively. This conversion is doable via the command line. For TGZ:

```
batocera-wine windows wine2winetgz xmoto.wine
```

or for SquashFS:

```
batocera-wine windows wine2squashfs xmoto.wine
```

TGZ and SquashFS files cannot be written to (they are read-only). To work around this:

- When a TGZ is detected, it is fully unzipped to the saves directory and a copy is created
- If a SquashFS is detected instead, Batocera will create a new directory in `saves/windows/<game name>/` which saves the difference in files between the read-only SquashFS and what the game writes to its directory during execution.

Ordinarily, Batocera would just let the game directly write data to the `roms/windows/<game name>.wine/` folder when it wants to.

To reiterate, it is preferable to use:

- `wtgz` for small games that make large writes (at least proportionally) where it makes more sense to just quickly repackage the game in order to keep things portable
- `wsquashfs` for larger games that make small writes (such as save files only) where it doesn't make sense to rapidly duplicate it, as that would take too long/wear down the storage hardware too quickly

This makes it easier to remove and restore a game, as it won't be modified with every launch. Remember to keep track of the save directory as well!

Launching a Windows game from SSH

For debugging purposes, to launch a Windows game from SSH, use the following commands:

```
export DISPLAY=:0.0
batocera-wine windows play /userdata/roms/windows/Windlands.pc
```

Where `/userdata/roms/windows/Windlands.pc` is the path to the game

Advanced

WINEtricks

Some games require tricks to run. For example, Age of King requires directplay for network play (<https://appdb.winehq.org/objectManager.php?sClass=version&iid=147>). It can be installed automatically thanks to the `tricks` command.

```
export DISPLAY=:0.0
batocera-wine windows tricks /userdata/roms/windows/aok.wine directplay
```

A list of WINEtricks commands and more detail about using WINEtricks with Batocera is available [right here](#).

Specify required dependencies

The other way to install dependencies inside a WINEprefix is by creating a folder in `/userdata/system/wine/exe`, and then putting the required executables inside. For 32-bit

architectures, install the 32-bit version, and for 64-bit architectures, install both the 32 and 64-bit versions. For each one, a download SSH command will be given. The related licenses for those is https://webpifed.blob.core.windows.net/webpifed/eula/aspnetcomponent_rtw_ENU.htm :

- **DirectX3D** : <https://batocera.org/users/liberodark/wine/directx.7z>
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/directx.7z && 7zr x /userdata/system/wine/exe/directx.7z -o/userdata/system/wine/exe && rm /userdata/system/wine/exe/directx.7z`
- **2015-2019 Visual C++ Redistributable** :
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2015_2019.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x86_2015_2019.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2015_2019.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x64_2015_2019.exe`
- **2019 Visual C++ Redistributable** : for missing files ???, ???, or files ending with vc16.dll.
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2019.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x86_2019.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2019.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x64_2019.exe`
- **2017 Visual C++ Redistributable** : for missing files mfc140ud.dll, msvcp150.dll, or files ending with vc15.dll.
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2017.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x86_2017.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2017.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x64_2017.exe`
- **2015 Visual C++ Redistributable** : for missing files mfc140u.dll, msvcp140.dll, or files ending with vc14.dll.
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2015.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x86_2015.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2015.exe
 - `wget -P /userdata/system/wine/exe https://batocera.org/users/liberodark/wine/vcredist_x64_2015.exe`
- **2013 Visual C++ Redistributable** : for missing files mfc120u.dll, msvcp120.dll, or files

ending with vc12.dll.

- 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2013.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x86_2013.exe`
- 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2013.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x64_2013.exe`
- **2012 Visual C++ Redistributable** : for missing files mfc110u.dll, msvcp110.dll, or files ending with vc11.dll.
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2012.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x86_2012.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2012.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x64_2012.exe`
- **2010 Visual C++ Redistributable** : for missing files mfc100u.dll, msvcp100.dll, or files ending with vc10.dll.
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2010.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x86_2010.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2010.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x64_2010.exe`
- **2008 Visual C++ Redistributable** : for missing files mfc90u.dll, msvcp90.dll, or files ending with vc9.dll.
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2008.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x86_2008.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2008.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x64_2008.exe`
- **2005 Visual C++ Redistributable** : for missing files mfc80u.dll, msvcp80.dll, or files ending with vc8.dll.
 - 32-bit: https://batocera.org/users/liberodark/wine/vcredist_x86_2005.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x86_2005.exe`
 - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2005.exe
 - `wget -P /userdata/system/wine/exe`
`https://batocera.org/users/liberodark/wine/vcredist_x64_2005.exe`

e

Once placing the required *.exe for the game as described previously, launch the game, and the dependencies will install themselves silently. After closing the game, rename the /userdata/system/wine/exe/ folder to something else, otherwise Batocera will try to reinstall the game each time it is started.

Manually put a registry file

To apply specific registry values for a given game, create a folder in /userdata/system/wine/regs/ and put the registry files (*.reg) inside. Once the game is launched, it will import the values contained in this file. Once the game is closed, rename the /userdata/system/wine/regs/ folder to something else, otherwise Batocera will try to re-apply those changes each time the game is started.

Troubleshooting

My game isn't working!

It can be very hard to troubleshoot what exactly is going wrong for games that are failing to run in WINE. First, check that it's on the spreadsheet as confirmed working before continuing troubleshooting, it might just be “borked” and cannot be fixed. Also, check the [WineHQ application database](#).

Next, try launching the game via SSH, as that will display some output in regards to the error it's encountering. This could lead to discovering which WINEtrick is required.

After discovering which [advanced system setting](#) adjustment got the game working, be sure to delete its saves/windows/<game> folder before trying again.

If there are still issues, it might be worth checking to see if the [Steam](#) version can be used instead, as that allows the automatic use of Proton which may grant different results.

My application isn't working!

As above, but here there may be more options. Check if the distributor/creator of the program offers a “portable” version, one that doesn't require writing files to the system directories and keeps everything self-contained. Some programs can be told to behave in a portable-like manner by having a portable.txt file present in their directory. Of course, for programs set up like this, do not use SquashFS, as games set up in this way require being able to write into their installed directory instead.

Sophisticated applications such as ones that require low-level access to hardware (such as drivers, patchers, kernels, complicated professional production software like compilers, etc.) may just not work due to the nature of WINE.

My game/app was working before but now is no longer!

Try deleting the /userdata/system/Wine/ folder, then try running the game again. Wine may need to reinstall, so be patient.

Which WINE versions are installed on Batocera?

Knowing which versions of WINE for Lutris and Proton (yes, they are different!) are actually installed on Batocera is essential for troubleshooting. To display the WINE versions for Lutris and Proton, execute the following commands from a remote PC via SSH:

- Get WINE version for Lutris:

```
/usr/wine/lutris/bin/wine64 --version
```

- Get WINE version for Proton:

```
/usr/wine/proton/bin/wine64 --version
```

Further troubleshooting

For further troubleshooting, refer to the [generic support pages](#).

From:

<https://wiki.batocera.org/> - **Batocera.linux** - Wiki

Permanent link:

<https://wiki.batocera.org/systems:windows>

Last update: **2023/05/22 23:49**

