Windows (WINE)

WINE allows you to play Windows games. Old and even recent games, 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 **v33** 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 your 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 `.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 telling you 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 as you normally would, 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.

Example:

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

or if you have some spaces in the name of the game

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

This `.wine` folder is called the wineprefix.

If you can't get the path 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:

![Game Folder Example](image)

If the *.exe of the game is directly in the .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 specials characters or spaces however, be sure to put quotes around it, like so:

`CMD="launch game.exe"`

The first time you run the game, a .wine folder will be created at /userdata/saves/windows.

This .wine folder is called the wineprefix.

**Emulators**

**WINE**
WINE configuration

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

<table>
<thead>
<tr>
<th>ES setting name</th>
<th>Description ⇒ ES option key_value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>batocera.conf_key</code></td>
<td></td>
</tr>
<tr>
<td><code>dxvk windows.dxvk</code></td>
<td>Converts the DirectX 9/10/11/12 calls to Vulkan ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>dxvk_hud windows.dxvk_hud</code></td>
<td>See your FPS and version of Vulkan API/driver ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>esync windows.esync</code></td>
<td>Can increase performance for games that stress the CPU ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>fsync windows.fsync</code></td>
<td>Can improve frame rate and responsiveness ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>pba windows.pba</code></td>
<td>Vastly improves the speed of buffer maps ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>fsr windows.fsr</code></td>
<td>AMD FidelityFX Super Resolution uses advanced scaling technologies to boost your fps ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>nvapi windows.nvapi</code></td>
<td>NVAPI is NVIDIA's core software that allows direct access to NVIDIA GPU ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>fps_limit windows.fps_limit</code></td>
<td>Allows you to limit fps to 60 ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>mf windows.mf</code></td>
<td>Install Media Foundation ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>virtual_desktop windows.virtual_desktop</code></td>
<td>Define the resolution and a new dedicated window ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>force_mouse windows.force_mouse</code></td>
<td>Force mouse in games ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>allow_xim windows.allow_xim</code></td>
<td>Enable XIM support ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>no_write_watch windows.no_write_watch</code></td>
<td>Disable support for memory write watches in ntdll ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>force_large_adress windows.force_large_adress</code></td>
<td>Force WINE to run games with large address ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>heap_delay_free windows.heap_delay_free</code></td>
<td>Delay freeing some memory ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>hide_nvidia_gpu windows.hide_nvidia_gpu</code></td>
<td>Force Nvidia GPUs to always be reported as AMD GPUs ⇒ Off 0, On 1.</td>
</tr>
<tr>
<td><code>wine_debug windows.wine_debug</code></td>
<td>Enable wine debug ⇒ Off 0, On 1.</td>
</tr>
</tbody>
</table>
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 too (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>/` 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>.wine/` folder when it wants to.

To reiterate, you would want 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 your save directory as well!

Launching a Windows game from SSH

For debugging purposes, if you want 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 your game

Recommended options

Recommended options for most games:
These options need AMD/NVIDIA/INTEL GPU with Vulkan support!

- DXVK = TRUE
- DXVK HUD = FALSE
- ESYNC = TRUE
- FSYNC = TRUE
- PBA = FALSE
- MF = AUTO
- VIRTUAL DESKTOP = FALSE

* DXVK** - converts the D3D (DirectX) calls from 9 to 12 in Vulkan.
* DXVK HUD** - to see your FPS and version of Vulkan API and Driver.
* ESYNC** - removes wineserver overhead for synchronization objects. This can increase performance for some games, especially ones that rely heavily on the CPU.
* FSYNC** - improving frame rates and responsiveness with scheduling policies.
* PBA** - implements a GL-free GPU heap allocator, vastly improving the speed of buffer maps.
* MF** - install Media Fondation, that help somes games for run.
* VIRTUAL DESKTOP** - is an option to define the resolution and a new dedicated window.

**Advanced**

**WINEtricks**

Some games require tricks to run. For example, Age of King requires directplay if you want to play on network ([https://appdb.winehq.org/objectManager.php?sClass=version&iId=147](https://appdb.winehq.org/objectManager.php?sClass=version&iId=147)). It can be installed automatically thanks to the tricks command.

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

You can find a list of WINEt tricks commands and also read a bit more about winetricks and Batocera 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, if you are on a 32-bits arch, install the 32-bits version, on a 64-bits arch, install both the 32 and 64 bits versions, for each one, a download SSH command will be given, the related licences for those is [https://webpifeed.blob.core.windows.net/webpifeed/eula/aspnetcomponent_rtw_ENU.htm](https://webpifeed.blob.core.windows.net/webpifeed/eula/aspnetcomponent_rtw_ENU.htm):

- **DirectX3D**: [https://batocera.org/users/liberodark/wine/directx.7z](https://batocera.org/users/liberodark/wine/directx.7z) (you can extract it)
  
  ○ `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**:
    - wget -P /userdata/system/wine/exe
  - 64-bit: https://batocera.org/users/liberodark/wine/vcredist_x64_2015_2019.exe
    - wget -P /userdata/system/wine/exe

- **2019 Visual C++ Redistributable**: for missing files ???. ???. or files ending with vc16.dll.
    - wget -P /userdata/system/wine/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`.
  - 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`.
    - 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`.
    - 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

Once you have put the required .exe for the game, you can launch the game, and the dependencies will install themselves silently. Once you closed the game, you need to rename the `/userdata/system/wine/exe` folder into something else, otherwise Batocera will try to reinstall them each time you start a game.
Manually put a registry file

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

Troubleshooting

My game isn't working!

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 will never work no matter what you try. You could also check the WineHQ application database.

Next, try launching the game via SSH, as that will give you some output in regards to the error it's encountering. This could lead you to discovering which WINETrick you might need to employ.

If you're still having issues, it might be worth checking to see if you could install the Steam version instead, as that allows the automatic use of Proton which may grant different results.

My application isn't working!

As above, but here you occasionally have 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, you should not be using SquashFS as they 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.

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: 2022/03/06 09:39