WineHQ Menu

- WineHQ
- AppDB
- Bugzilla
- Wine Wiki
- Wine Forums

Wiki Links

- FrontPage
- RecentChanges
- FindPage
- HelpContents
- FAQ

Page Tools

- Immutable Page
- Show Changes
- Get Info
- Show Raw Text
- Show Print View
- Attach File
- Delete Cache

User Tools

- Login
- Wiki Sand Box
- Syntax Reference

FAQ

This FAQ, or **F**requently **A**sked **Q**uestions, covers general topics about Wine. For questions related to Wine software development, see the developer FAQ.

Quick links: running a program in Wine, running Wine from a terminal, uninstalling an application, getting a debugging log.

Contents

1. General questions

1.1. Who is responsible for Wine?

Wine is available thanks to the work of many people around the world. Some companies that are or have been involved with Wine development are CodeWeavers, Bordeaux, TransGaming, Corel, Macadamian and Google. *See* Acknowledgements and WineHistory.

1.2. Does Wine hurt Linux?

Wine increases the usefulness of Linux, makes it easier for users to switch to Linux, and makes it easier for Windows developers to make applications that work on Linux. See the Debunking Wine Myths article for a fuller answer.

1.3. Is Wine an emulator? There seems to be disagreement.

There is a lot of confusion about this, particularly caused by people getting Wine's name wrong and calling it WINdows Emulator.

When users think of an emulator, they tend to think of things like game console emulators or virtualization software. However, Wine is a compatibility layer - it runs Windows applications in much the same way Windows does. There is no inherent loss of speed due to "emulation" when using Wine, nor is there a need to open Wine before running your application.

That said, Wine can be thought of as a Windows emulator in much the same way that Windows Vista can be thought of as a Windows XP emulator; both allow you to run the same applications by translating system calls in much the same way. Setting Wine to mimic Windows XP is not much different from setting Vista to launch an application in XP compatibility mode.

There are a few things that make Wine more than just an emulator.

- Sections of Wine can be used on Windows. Some virtual machines use Wine's OpenGL-based implementation of Direct3D on Windows rather than truly emulate 3D hardware.
- Winelib can be used for porting Windows application source code to other operating systems that Wine supports to run on any processor - even processors that Windows itself does not support.

"Wine is not just an emulator" is more accurate. Thinking of Wine as just an emulator is really forgetting about the other things it is. Wine's "emulator" is really just a binary loader that allows Windows applications to interface with the Wine API replacement.

1.4. What is the difference between Wine, CrossOver, and Cedega?

Wine is the base of the project, where most of the work is being done. Wine is not perfect, but tens of thousands of people nevertheless use "vanilla" Wine successfully to run a large number of Windows programs.

CrossOver (formerly CrossOver Office) is a product made by a company called CodeWeavers that is based directly on Wine with a few tweaks and proprietary add-ons. Unlike the biweekly Wine releases, CrossOver releases are rigorously tested for compatibility with CodeWeavers' supported applications in order to prevent "regressions". CodeWeavers employs a large proportion of the Wine developers and provides a great deal of leadership for the project. All improvements to Wine eventually work their way into CrossOver.

Cedega (formerly WineX) is a product from a company called TransGaming. TransGaming based their product on Wine back in 2002 when Wine had a different license, closed their source code, and rebranded their version as specialized for gamers. In the years since Cedega was originally created from Wine, development on Wine and Cedega have continued mostly independently. TransGaming currently gives back very little code to Wine. Cedega is **not** "Wine with more gaming support" - because Wine has had years of development since Cedega was made, and many games actually run better under Wine than under Cedega. Currently, Wine has more advanced Direct3D support than Cedega, but Cedega still has more advanced copy protection support due to TransGaming's licensing of (closed source) code from a handful of copy protection companies. Unlike CrossOver, most improvements to Wine *don't* get into Cedega due to the license differences between Cedega and Wine.

For more information, see Wine History.

1.5. Do I have to use the command line?

You do not have to use the command line to use Wine. You can use a graphical interface for most things, much like on Windows. In many cases you can right-click an installer and select "Open with Wine", or just double-click it. You can start installed programs using the shortcut icon or menu.

Even if you have a recent version, there are several situations when you might want to use the command line. The most common reason is to get debug output when your program does not run properly. You might also want to use utilities such as `regedit` that do not have menu shortcuts.

1.6. Will Wine work with my application?

Probably! If your application is even slightly well-known, you will probably find other user reports in the Application Database. If there aren't any reports using a recent version of Wine, however, your best bet is to simply try and see.

1.7. What applications run well with Wine?

Thousands of applications work well. As a general rule, simpler or older applications tend to work well, and the latest versions of complex applications or games tend to not work well yet. See the Wine Application Database for details on individual applications. If it rates your application Gold or Platinum, you're probably ok. (If it rates it Silver or Bronze, Wine isn't really ready to run the application for most users.)

For instance, here is the appdb entry for Microsoft Office. It rates the various versions of Office as Bronze or Silver, and has tips for how to get around some problems. (If you need to run Office now, consider the commercially-supported version of Wine offered by Codeweavers; it contains a recent Wine plus a few special workarounds to get Office working better.)

1.8. How do I run Wine?

Wine is not an application you run. Wine enables your computer to run Windows applications. Simply install and run your applications as you would in Windows. *See* How do I run an installer using Wine.

1.9. How can I help contribute to the Wine project, and in what ways?

You can contribute programming or documentation skills, or monetary or equipment donations, to aid the Wine developers in reaching their goals.

One area where every Wine user can contribute to this project is by sending high quality bug reports to our Bugzilla and helping the developers with any followup questions that they may have about your bug. It is impossible and impractical for a developer to have a copy of every program on the market, so we need your help even after your initial bug report. If a developer has a good idea what might be causing the bug, he or she may ask if you can try a patch and see if it fixes the problem. If the patch works and then makes its way into our main development tree, the bug report will be closed, your help will be appreciated by everyone and your problem will be fixed.

For a list of ideas of how you can help, please consult the helping Wine page.

2. Installing Wine

2.1. Which version of Wine should I use?

Short answer: Use the version that works best with the particular apps you want to run. In most cases, this will be the latest development version; however, in some cases it may take some experimenting to find it.

Longer answer: Wine development is rapid, with new releases in the development branch every two weeks or so. Functionality will usually be best with the most recent development version, however, there are cases where changes to Wine cause applications that worked well in older versions to not work in the new one; these are called regressions.

Agood rule of thumb is to start with the version of Wine installed with your distro and see if that works with the apps you want to use. If it does, good! If it doesn't, upgrade. In most cases the upgrade should be to the latest development version, but it is a good idea to check Bugzilla and the AppDB for any known regressions. If there are any, upgrade to the most recent version known to work for your app.

Note that while Wine does have a "stable" branch, the term "stable" refers to the branch as a whole, which is infrequently updated, and only with bugfixes promised not to cause regressions. The only apps promised to work in the current (1.0.1) stable release are Word Viewer 2003, Powerpoint Viewer 2003, Excel Viewer 2003, and the Photoshop CS2 tryout. Users of a development release can achieve the same degree of stability by simply not upgrading.

Ubuntu users: the "wine1.2" package in Ubuntu Karmic 9.10 (and up) is actually the most recent development version in the 1.1.x series. The official Wine version 1.2.0 (the next stable release) has not yet been released.

2.2. How do I install Wine?

The best option is to use precompiled binary package(s) for your particular distribution. See the Wine download page for links and additional information. If you can't find packages with a recent Wine version for your distro, you can compile Wine from source.

2.3. Does Wine run on 64-bit?

Yes. Normally, installation should be the same as with 32-bit: simply install the Wine package for your distribution. Check the Downloads page. If you need to build Wine from source, see WineOn64bit.

Note that Wine for 64-bit actually runs in 32-bit mode. This is necessary, as virtually all Windows applications are 32-bit. Support for 64-bit Windows applications is currently experimental (see Wine64).

- Wine is currently offered in 32-bit. 16-bit and 32-bit Windows applications work with it. It runs on both 32-bit and 64-bit Linux/Unix installations.
- Wine is also experimentally offered in 64-bit. 32-bit and 64-bit Windows applications (should) work with it. It runs only on 64-bit Linux installations.

2.4. Installing on Apple

2.4.1. How do I install Wine on my Mac?

If you are running Mac OS X, please see the Mac OS X FAQ. If this is too complicated, there is also a version of Codeweavers' Crossover for Mac.

We are looking for someone to help us build Mac packages so that Wine will be as easy to install on Mac as it is on Linux. If you can help, please contact the developers' mailing list.

If you are running Linux on your Mac, installing Wine is as simple as installing it under Linux on a PC. Simply visit the downloads page.

2.4.2. Can I use Wine on an older Mac without an Intel chip?

No, not even in Linux. Older Macs used PowerPC processors, which are incompatible with code compiled for x86 (Intel and AMD) processors unless the code is run under CPU emulation. Wine Is Not a (CPU) Emulator, nor does it include one. The Darwine project, however, is an effort to do just that.

2.5. How do I install Wine on my netbook (eeePC, Acer Aspire One, etc.)?

If you have replaced the customized distro that came preinstalled on your netbook (Xandros, Linpus, etc.) with one of the mainstream distros that provide up-to-date Wine packages, you should be able to install Wine as normal for that distro.

If you are still using Xandros (eeePC), Linpus (Acer Aspire One) or any other customized distro that came preinstalled on your netbook, you will have to ask on your netbook's support forum. Only other users of those distros can advise you on what, if any, binary packages will work on your system, where to find them, and how to install them.

You can also try building Wine from source following the instructions in the Wine User Guide, but you will still need to consult your netbook's support forum regarding satisfying dependencies on your particular system.

2.6. How do I compile Wine from source?

- Make sure you have the recommended packages installed.
- Get Wine using Git, or download the latest tarball and unpack it.
- If you want to apply a patch, do it with a command like `patch -p1 < foo.patch` in the Wine source directory. You may have to experiment before you get this right. See the manual for patch for details.
- Run `configure`, `make depend` and `make` to compile Wine, and `sudo make install` to install it.

See any of the many tutorials about building packages from source, e.g. http://www.tuxfiles.org/linuxhelp/softinstall.html or https://help.ubuntu.com/community/CompilingSoftware for more tips, or see the README file in the source tarball for more detailed instructions.

You don't actually need to install Wine to use it. Even if you do install it, you might want to pass `--prefix=/usr/local/wine` to configure so it gets installed in `/usr/local/wine/bin/wine`; that way it won't conflict with the system's copy of Wine. But then you have to remember to run that version rather than the system's.

2.7. How do I apply a patch?

You have to build Wine from source; see above.

2.8. How do I uninstall Wine?

If you want to remove Wine entirely after you installed it with your distribution's package manager, you can generally uninstall in the same way. If you installed Wine from source, then use `make uninstall` in the source directory to remove it.

Note, however, that uninstalling Wine will *not* eliminate the virtual Windows installation - the equivalent of "reinstalling Windows." To do that you must follow the instructions below.

3. Installing applications

3.1. I have lots of applications already installed in Windows. How do I run them in Wine?

Short answer: You have to install them in Wine **just like you did in Windows**. Applications usually have a setup or installer program.

Long answer: Some applications can be copied from Windows to Wine and still work, but don't try this unless you like tinkering under the hood of your car while it's running.

Wine is not designed to interact with an existing Windows installation. If you have any data you need from a Windows installation, browse your Windows filesystems in your normal file manager and copy the data to another location.

WARNING: Do not try to configure Wine to point to your actual Windows `C:\` drive. **This will break Windows and require a Windows reinstall.** We have tried to make this hard to do, so you probably cannot do it by accident. If you do manage this, Wine may or may not continue to operate, but your Windows install will be 100% dead due to critical parts of it being overwritten. The only way to fix Windows after this has happened is to reinstall it.

3.2. How do I run an installer using Wine?

Double-click on the installer, just like in Windows!

This may sometimes open the file in the wrong program - if this happens, check the filetype associations for the file.

You can also right-click on it, choose "Run with", and choose "Wine".

Or, if all else fails, open a terminal window, "change directory" to the folder containing the installer's .exe file, and run the installer with Wine by typing `wine` followed by the installer's filename. For example:

```
cd ~/Desktop
wine FluffyBunnySetup.exe
```

If the installer's name doesn't end in .exe, you have to say "wine start" and then the filename:

```
cd ~/Desktop
wine start FluffyBunnySetup.msi
```

Instead of typing the entire path and filename, you can usually type just the first few letters and then press Tab, and the computer will complete the filename you were typing for you. If there are more then one file that start with the same letters, press TAB second time to see all files that match.

If the program comes on multiple CD-ROM discs, you don't want to start Wine from the CD-ROM's directory or you won't be able to eject discs without exiting the installer. First, verify that your CD-ROM drive is assigned a drive letter in the "Drives" tab of `winecfg` (e.g. D: -> `/media/cdrom`). Then run the installer this way:

```
wine start 'D:\setup.exe'
```

or

```
wine start /unix /media/cdrom/setup.exe
```

3.3. How can I install applications to be shared by multiple users?

Wine does not currently allow sharing its configuration ("prefixes") between users, due to the risk of registry corruption from running multiple wineservers simultaneously (bug #11112). At present, applications must be installed separately for each user.

However, you can copy Wine prefixes; you can install everything to one prefix, then make a copy of it in each user's home directory. This saves running installers repeatedly.

3.4. How can I prevent Wine from changing the filetype associations on my system or adding unwanted menu entries/desktop links when I install a Windows program?

Users who do not want the installer for a Windows app to change filetype associations, add menu items, or create desktop links, can disable `winemenubuilder.exe` . There are several ways to do this:

1. *In winecfg:* before running the installer, run `winecfg`. Go to the Libraries tab and type `winemenubuilder.exe` into the "New overrides" box (it is not in the dropdown list). Click add, then select it from the "Existing overrides" box. Click "Edit" and select "Disable" from the list, then click "Apply."
2. *Registry file:* If you need to apply the setting many times (e.g., every time you recreate the Wine prefix), this approach may be more convenient. Create a text file named with extension .reg (e.g., `disable-winemenubuilder.reg`) containing the following:

   ```
   [HKEY_CURRENT_USER\Software\Wine\DllOverrides]
   "winemenubuilder.exe"=""
   ```

   To apply the setting, run:

   ```
   regedit disable-winemenubuilder.reg
   ```

3. *Environment variable:* set the WINEDLLOVERRIDES environment variable when you run the installer, e.g.,

   ```
   WINEDLLOVERRIDES=winemenubuilder.exe=d wine setup.exe
   ```

Disabling winemenubuilder.exe will cause wine programs to print errors that they cannot find it (though the errors do not seem to interfere with program operation). An alternative approach that avoids the errors is to replace it with the do-nothing program. Compile the following with MinGW's gcc under wine and place the executable at `C:\Windows\System32\winemenubuilder.exe` :

```
#include <stdio.h>
int main()
{
/* Do nothing */
return 0;
}
```

Then follow the same procedure as above, but set winemenubuilder.exe to "native" instead of "disable" in winecfg, or use the following registry file or environment variable:

```
[HKEY_CURRENT_USER\Software\Wine\DllOverrides]
"winemenubuilder.exe"="native"
```

```
WINEDLLOVERRIDES=winemenubuilder.exe=n wine setup.exe
```

3.5. I have an MSI file, how do I install/run it?

MSI files cannot be run directly, you need to use the ["msiexec"] program. MSI files can be installed in Wine from the terminal like this

```
wine msiexec /i whatever.msi
```

Alternatively:

```
wine start whatever.msi
```

That will then run the MSI program the same as if you had double-clicked it in Windows.

3.6. Can I use Wine to install drivers for my hardware?

No. With the possible future exception of some printer drivers, Wine requires your hardware to already be working on your operating system. The technical reason for this is that Wine, like most applications, runs in user mode and not kernel mode.

3.7. My installer tells me I don't have enough free disk space

Usually, you really don't have enough free disk space. Wine's C: drive is located in your home directory. Whatever partition contains /home must have enough free space for your program. You can check by running:

```
df -h $HOME
```

If the amount of space you need is less than 1 GB, and df reports you have more than 1 GB available, try setting the Windows version to Windows 98. This will work around bugs in some old (Windows 98 era) installers that could not cope with large drives.

If you want to use a different partition that has more free space, use a wineprefix that is located on the other partition. Note that your other partition must be a Unix filesystem; FAT and NTFS partitions will not work. See How can I run two programs as if they were on different computers for instructions.

4. Running applications

4.1. How do I run an application once I've installed it?

After you install an application with Wine, it will probably have an entry in the your computer's Applications->Wine->Programs menu, and/or an icon on the desktop, just as it would under Windows. You should be able to use them just as you would on Windows.

Alternately, browse to the .wine/drive_c/Program Files folder in your home directory (it is hidden, so you might need to do View->Show Hidden Files in your file manager), look for the application's main .exe file (you may have to guess), and double-click it.

Or, if all else fails, open a terminal window, and navigate down to the application's directory. For example:

First, navigate to your home directory in the command line:

```
cd
```

.wine/drive_c in your home directory is equivalent to C:\ for your Windows programs, so to look in C:\Program Files you enter .wine/drive_c/Program Files:

```
cd ~/.wine/drive_c/Program\ Files
ls
Adobe     Microsoft   FluffyBunny
cd FluffyBunny
```

Then look for the application's main .exe file (you may have to guess), and run it using the wine command:

```
ls *.exe
fluff.exe  uninstall.exe  ereg.exe
wine fluff.exe
```

4.2. I want to pass command line arguments to my program

If you're using a program with switches on Windows, for instance:

```
quake.exe -map e1m1
```

Then you can do the equivalent in Wine by running:

```
wine quake.exe -map e1m1
```

That is, the command line is identical, except with wine in front. Note, however, that you may need to escape certain special characters with backslashes due to the way they're handled in the Linux shell. For instance:

```
quake.exe -map C:\Quake\e1m1.bsp
```

Becomes:

```
wine quake.exe -map C:\\Quake\\e1m1.bsp
```

For more information on using backslashes, see: http://www.tuxfiles.org/linuxhelp/weirdchars.html

4.3. How should I start Windows programs from the command line?

This will allow you to see messages from Wine that may help troubleshoot problems.

Because Windows programs will often look for files in the location they were started from, when using the command line you should start them in a very specific way: "change directory" to the folder where the program is located and run the .exe file using **only** its filename. For example:

```
cd '.wine/drive_c/Games/Tron'
wine tron.exe
```

In some cases you may wish to specify the full path to the program's .exe file. For example, if you need to install a program from multiple CDs, the previous method won't work (entering the directory in the terminal will prevent you from removing the CD). You can provide Wine with a DOS or Windows style path inside single quotes like so:

```
wine start 'C:\Games\Tron\tron.exe'
```

You need to use `wine start` if you specify a full path, because that allows Wine to set the working directory for the program if it needs it. You can also use double quotes, but you need two backslashes instead of one:

```
wine start "C:\\Games\\Tron\\tron.exe"
```

If you need to use a Unix style pathname, use the /Unix option to start, e.g.

```
wine start /Unix "$HOME/installers/TronSetup.exe"
```

For current Wine, once a program is installed, you can safely use any shortcuts that the installer has created.

4.4. When I double-click on a .exe file in my file manager, nothing happens.

Note: If you can, start applications by clicking on the application's icon in the Applications / Wine menu or desktop instead. Double-clicking `.exe` files is typically only needed for applications that aren't installed yet, e.g. to run the `setup.exe` on a CD-ROM game or a downloaded installer.

If double-clicking doesn't work (*e.g.*, recent Ubuntu), you might need to right-click the file and choose "Run with Wine". It depends on your file manager. If that also doesn't work, contact whoever built your Wine packages and let them know there's a problem.

You can work around this by using the command line instead of your file manager (see previous question). If you see a line like:

```
err:module:import_dll Library MFC42.DLL (which is needed by L"C:\\Program
Files\\Yoyodyne\\Overthruster.DLL") not found
```

it means you need to install a missing runtime library; see e.g. "My application won't run, and says it needs MFC42.DLL" below.

5. Uninstalling applications

5.1. How do I uninstall Windows applications?

Wine has its own built-in uninstaller - the equivalent of Windows' "Add/Remove Programs" function for running standardized uninstallers.

Note that Wine does not fully implement everything required to cleanly uninstall all applications. Some uninstallers might not function at all. To remove **all programs installed under Wine**, remove the wineprefix (usually the `~/.wine` directory):

Please note that in the following commands there should be no spaces in the path, particularly between `$HOME/` and `.whatever`.

```
rm -rf $HOME/.wine
```

The uninstaller should remove menu and desktop entries. If the application was installed with an old version of Wine, it may not remove them. To remove **all** Wine-created menu entries run the following commands

```
rm -f $HOME/.config/menus/applications-merged/wine*
rm -rf $HOME/.local/share/applications/wine
rm -f $HOME/.local/share/desktop-directories/wine*
rm -f $HOME/.local/share/icons/????_*.xpm
```

6. Troubleshooting

6.1. How can I get a debugging log?

Run your application from the command line (see How should I start Windows programs from the command line?)

You can then copy the log from the screen and paste it into a file if it's short; otherwise redirect the output of wine to a file using a Unix shell redirection, e.g.

```
cd ~/.wine/drive_c/Games/Tron
wine tron.exe &> log.txt
```

See also How do I get a debug trace? for information on how to get more verbose debug logs.

6.2. How do I disable the GUI crash dialog?

As of 1.1.20, Wine includes a GUI crash dialog that is turned on by default. Users of apps that work despite a background crash may find the GUI dialog annoying, and in some cases the dialog itself has been reported to prevent an app from working.

The easiest way to disable the crash dialog is with winetricks:

```
sh winetricks nocrashdialog
```

If you prefer to do it manually, copy the following key to a text editor:

```
[HKEY_CURRENT_USER\Software\Wine\WineDbg]
"ShowCrashDialog"=dword:00000000
```

Save the file with a .reg extension (e.g., `crashdialog.reg`), then apply it with regedit:

```
regedit crashdialog.reg
```

(You may need to specify the full path to the file, depending on where you saved it.)

To turn the GUI crash dialog back on, change 00000000 to 00000001 and reapply.

These changes can also be made by simply running `regedit` and adding/changing the appropriate key the point-and-click way.

6.3. My program froze up, how do I close it?

If you ran the program from a terminal window by typing wine (program).exe, you can usually just go back to that terminal window and press Ctrl+C. If you ran the application some other way, such as from a launcher shortcut, then you can open up a terminal and forcibly kill the process:

```
killall -9 Application.exe
```

If you want to kill all Wine programs at once, you can run:

```
wineserver -k
```

You can also open up a Wine version of the Windows task manager by running `wine taskmgr` in a terminal. This will allow you to kill individual Wine processes.

6.4. My whole computer freezes or reboots when I run my game in Wine!

If you are getting a complete deadlock and are unable to even use your mouse after running Wine, it's probably not a specific problem with the Wine software. Wine is a user-level process, and shouldn't be able to completely hang the operating system under any circumstances. Instead, Wine is likely exposing a deeper problem with the system, such as a defective hardware driver, a bad memory stick, or overclocking flakiness.

It's often a graphics driver problem, in which case non-Wine apps might also be affected. If running `glxgears` also crashes, it's definitely a graphics driver problem. The most common cause is upgrading to a new kernel without also updating the graphics drivers to match. Try reinstalling your graphics drivers.

If that doesn't help, ask for help on the wine-users forum. Be sure to mention the name of the app, the version of wine, the output of `cat /etc/issue`, `lspci | grep -i vga`, and, if you're using the proprietary NVidia drivers, `cat /proc/driver/nvidia/version`. Maybe someone can help.

6.5. My application says some DLL or font is missing. What do I do?

Applications *should* come with all the DLLs they need (except for core Windows DLLs). They sometimes forget to, and rely on you to already have the DLL or font installed. You can install the missing DLL or font in several ways:

1. downloading them from the original creators of the runtime (eg. Microsoft). The easiest way to do this is with winetricks (see below).
2. installing other applications which do include them
3. copying them from a licensed version of Windows installed on the same machine

⚠ Do not download DLLs or scripts from websites you do not know and trust! Fake or infected DLLs can cause you great pain, even on Wine.

Missing DLLs that can be installed with winetricks include:

- MFC40.DLL and friends: run winetricks and select `mfc40` (Microsoft Foundation Classes from Visual C++ 4).
- MFC42.DLL, MSVCP60.DLL and friends: run winetricks and select `vcrun6` (MS Visual C++ 6 sp4 libraries: mfc42, msvcp60, msvcrt).
- MFC71.DLL, MSVCP71.DLL and friends: run winetricks and select `vcrun2003` (MS Visual C++ 2003 libraries: mfc71,msvcp71,msvcr71).
- MFC80.DLL, MSVCP80.DLL and friends: run winetricks and select `vcrun2005` (MS Visual C++ 2005 libraries: mfc80,msvcp80,msvcr80).
- MFC90.DLL, MSVCP90.DLL and friends: run winetricks and select `vcrun2008` (MS Visual C++ 2008 libraries: mfc90,msvcp90,msvcr90).
- MDAC: run winetricks and select `mdac27` (MS MDAC 2.7).

See the winetricks wiki page for a complete list of winetricks commands.

6.6. My application won't run, and the log shows lots of OLE errors

There's some chance that Wine's COM implementation has a bug or missing feature that's hurting your app. If so, you can sometimes work around this by using native DCOM98. You can install OLE by running winetricks and selecting `dcom98`.

⚠ While DCOM98 may solve a problem with one app, installing it can break other apps in the same wineprefix. It is recommended that you install DCOM98 and any apps needing it to a separate wineprefix.

If DCOM98 helps, be sure to file a bug report. If it doesn't, you can undo the installation by running winecfg and removing the native library overrides for `rpcrt4`, `ole32` and `oleaut32`. See also NativeDcom.

6.7. My application won't run, and says it needs .NET

Wine still has many problems with .NET. You may be out of luck, but if you're brave, you can still try running it.

Install .net 2.0 by running winetricks and selecting `dotnet20`. This is not yet well-supported, but several .NET applications work. See the .NET AppDB pages for more information on installing these runtimes.

Alternately, you can install Mono, an open source implementation of .net 2.0, by running winetricks and selecting `mono19`. This is not well-supported, and very few .NET applications actually work with it under Wine. See the Mono Project AppDB page for more information.

6.8. My application won't run, and says it needs Internet Explorer

Wine uses the core of Firefox (called Gecko) to implement its own Internet Explorer.

In some cases, you can make applications which think they need IE happier by running winetricks and selecting `gecko` and/or `fakeie6`. You can also try to install the real Internet Explorer using winetricks: see "How do I install Internet Explorer in Wine?" below.

6.9. My application runs, but text areas act strangely (e.g. lines don't wrap, double-clicking doesn't select words)

You may have run into a bug in Wine's RICHED20.DLL. You can try using Microsoft's RICHED20.DLL by running winetricks and selecting `riched20`. This may let you work around the problem until the Wine developers fix the bug.

6.10. Graphics

6.10.1. My application complains about being unable to change the resolution or color depth.

You generally need to edit the Screen section of your `/etc/X11/xorg.conf` to support additional color depths and resolutions. There may also be a problem with Xrandr.

6.10.2. The application I am trying to run complains that it needs 256 colors but I have millions of colors.

The inability to switch from 24bpp mode to 8bpp mode is a limitation of X, not a bug in Wine. See 256ColorMode for some possible workarounds.

6.10.3. My X screen won't go back to its normal resolution after running a game fullscreen.

You can often work around this by changing the screen resolution and then changing it back again under the system preferences.

Alternately, you can run this terminal command to restore your X settings:

```
xrandr -s 0
```

6.10.4. I'm using Desktop Effects with Compiz, Fusion, or XGL and get poor performance/odd messages/broken applications

Using composite display managers in Linux tends to cripple OpenGL performance or break OpenGL entirely. We recommend that you disable them entirely before trying to use Wine. If you are using one and experiencing slow performance then please do not file bugs in Wine, as these are bugs in your window manager or your video drivers. Also, disabling the Composite extension within `/etc/X11/xorg.conf` will most certainly prevent any compositing from affecting Wine.

6.10.5. Graphics in games with good ratings in AppDB are scrambled.

- Check the game using the latest graphics drivers.
- Most AppDB entries are based on **NVIDIA/GeForce** hardware running the proprietary driver.
- **ATI/AMD/Radeon** cards running the proprietary **fglrx** driver have problems in Wine. As a rule of thumb, at least games that use shaders are broken. See this posting and Wine bug 7411 for details.
- **Other hardware** (Intel/S3/Matrox etc.) will probably run only old (non-shader) games. Compatibility is not well tested.
- Same for **open source** drivers as their 3D support is typically basic only.

6.10.6. Wine displays corrupted or missing text.

This may be bug 16146, caused by the `nvidia-96xx` legacy driver, or bug 18120, which affects QT 4.5.0 applications. It could also be caused by missing fonts, font conflicts, or adding new fonts to Wine.

Try using a fresh Wine prefix (by moving or deleting `~/.wine`, or changing the `$WINEPREFIX` environment variable). If you still have this problem, try setting the following in the Wine registry:

```
[HKEY_CURRENT_USER\Software\Wine\X11 Driver]
"ClientSideWithRender"="N"
```

Place above in a text file called `norender.txt` and it can be inserted into the registry with the command `regedit norender.txt`. Please apply only as required. (This was reported as being required of OS X on the 1 Dec 2007, and more recently on other platforms, such as Ubuntu.)

6.10.7. Wine's windows and fonts are extremely large, and the winecfg window doesn't fit on the

screen.

Sometimes you can use the ALT key and the mouse to move the `winecfg` window so you can reach the "Screen Resolution" slider on the Graphics tab; slide it down. Changes will not effect the `winecfg` window until it's restarted.

If that doesn't work, you can use this one line registry change (all one line):

```
echo -e "[HKEY_LOCAL_MACHINE\\System\\CurrentControlSet\\Hardware
Profiles\\Current\\Software\\Fonts]\n\"LogPixels\"=dword:60\n" | wine regedit -
```

If all fails, you could remove your `~/.wine` directory and reinstall your Windows applications.

6.11. Sound

6.11.1. I'm using OSS in Wine and I have no sound.

Unless you're using a sound card with an emu10k1/2 chipset (SB Live!/Audigy) or another type of card that supports "hardware mixing" then you can only ever have one application using sound at the same time. If you have aRts (KDE sound system), ESD (Gnome sound system) or any music applications open then you simply won't hear anything from Wine. You need to kill or close all these programs before being able to get sound from Wine. To work out if anything has locked your sound-card already, you can do `cat /dev/urandom >> /dev/dsp`. If you get an error message it means that something is using it. If you get loud static this means that your sound card is clear and ready for use.

6.11.2. Running winecfg seems to hang or complain about files when I click the audio tab

The hang is caused by the NAS sound driver. This causes `winecfg` to pause for a while but it should respond eventually. The only way to get around this is to remove NAS from your system and/or build Wine without NAS support in the first place.

If you see messages about JACK in the terminal, they can be ignored unless you intend to use the JACK driver. If you wish to use the JACK driver, you need to install JACK's libraries onto your machine before JACK will work.

6.11.3. Audio in games sounds different from running in Windows

A number of DirectSound features are not implemented (correctly) in Wine yet.

- Sound effects (echo etc.) are not implemented (Bug 10603)
- Multi-channel surround audio is not implemented (Bug 11764)
- 3D sound positioning is not completely correct (Bug 11499)
- Doppler effect is not implemented (source)

6.11.4. MP3s do not play in Windows Media Player or applications that depend on it

For MP3 sound to play out of the box in apps that use the WMP engine and codecs, you must have 32-bit `libmpg123` installed on your system and Wine must have been compiled with MP3 support. Not all distro packages provide this; openSUSE packages in particular are known to be compiled without MP3 support.

The workaround for lack of libmpg123 and/or winemp3.acm is to use the codec installed by WMP9, l3codeca.acm. Copy l3codeca.acm to the wineprefix's /windows/system32 directory (or use winetricks to install WMP9), then create a symlink to it named winemp3.acm in the same directory. Wine will then use the native codec to play MP3s.

This only affects WMP and apps that rely on it for MP3 playback (including Powerpoint). Apps that install their own MP3 codec, such as Winamp or VLC Player, should be able to play MP3s without this workaround.

6.12. My CD or DVD disc won't eject

Try `[:eject:wine eject]` . It is a function to free up, unlock, and eject the drive. Make sure that the drive is mapped as a CD-ROM in `winecfg` and specify the drive letter in the command line, *e.g.*:

```
wine eject d:
```

6.13. My program fails to do networking, but Linux can get online

Note: These instructions are for older Wine installs. If you're using Wine 1.x and your application still fails to do networking, you can give this a try as well. If you're running Wine 1.x and below instructions work for you, **file a bug** so we can fix Wine to improve other people's experience.

You need to make sure that your hostname resolves to the IP address of your network interface. To verify if you have this problem run **hostname -i**. If it returned IP address starting from "127." then read on.

To set this up correctly, you can type the following from a terminal:

```
hostname
```

This will return your hostname the way your computer sees it. Now, you need to open an editor with system administrator privileges, how you do this will depend on the distribution you are using. Open the file /etc/hosts and see if there is an entry for your hostname. Assuming your hostname is "yourhost" and your network IP address is 192.168.0.23, the entry might look like this:

```
127.0.0.1       yourhost.yourdomain.com      yourhost
```

Change this to (or add, if there is no such line):

```
192.168.0.23    yourhost.yourdomain.com      yourhost
```

For most Windows games with networking problems, this is all you need to get networking to work.

6.14. Why doesn't DNS resolve in 64-bit operating systems?

Many distributions don't provide all the 32-bit compatibility libraries that wine needs. In this case, wine needs 32-bit DNS libraries. On

Ubuntu/Debian, this package is `lib32nss-mdns`. To install it on those OSes, use:

```
sudo apt-get install lib32nss-mdns
```

For other operating systems, the package name and installation method may differ. Consult your distribution's support channels.

6.15. I deleted my Wine menu, and now I can't get it back.

Rather than actually delete anything, menu editors on Linux simply mark menus as "Deleted" so that they don't show up in the menu. First, see if the directory is in your menu editor and can be re-enabled.

This information is stored in `~/.config/menus/applications.menu`. Edit `~/.config/menus/applications.menu` and you should find a section near the end that looks similar to this:

```
        <Menu>
                <Name>wine-wine</Name>
                <Deleted/>
        </Menu>
```

or perhaps this:

```
        <Menu>
                <Name>wine-wine</Name>
                <Menu>
                        <Name>wine-Programs</Name>
                        <Menu>
                                <Name>wine-Programs-AutoHotkey</Name>
                                <DirectoryDir>/home/user/.local/share/desktop-
directories</DirectoryDir>
                        </Menu>
                </Menu>
                <Deleted/>
        </Menu>
```

Remove the <Deleted/> line and your Wine menu will reappear.

6.16. Some key combinations in my application do not work.

Even in full screen mode, window managers typically capture some keys. For example, in KDE and GNOME, Alt+Left Click is used to move the whole application window by default. Thus, this key combination is not available to applications in Wine. You have to disable the colliding combinations in your window manager. For KDE, see Control Center/Window Behaviour or (better) Window-specific settings/Workarounds/Block global shortcuts. For GNOME, see System/Preferences/Windows and change the "Movement Key" setting. Also see System/Preferences/Keyboard Shortcuts for specific keyboard combinations.

(Keywords: Keyboard, Shortcut, Modifier, Alt, Ctrl, Control.)

6.17. ISSUE: "preloader: Warning: failed to reserve range 00000000-60000000"

or

6.18. winevdm: unable to exec '<APP NAME>': DOS memory range unavailable

This issue is being followed in bug 12516.

The cause is a Linux kernel setting. Run `cat /proc/sys/vm/mmap_min_addr`"as root, if it does not equal 0 then running `sysctl -w vm.mmap_min_addr=0` as root can be used to temporarily fix the issue; to fix it permanently, add the line `vm.mmap_min_addr=0` to `/etc/sysctl.conf`. Please record if you do this alteration, as the area Wine needs may change.

See PreloaderPageZeroProblem for more information.

6.19. Where can I get further help?

In addition to this wiki, the Wine HQ Documentation and mailing lists. You'll most likely want the wine-users mailing list which also has a forum interface. If you're an ISV looking at porting an application with Winelib you can also try wine-devel.

For help with a specific application, search the Application Database, a place where users share their experiences by submitting test data, tips and tricks, and asking questions.

The IRC channel: #WineHQ on irc.freenode.net. Knowledgeable Wine users hang out there, and often developers will lurk there too. See IRC for more important information.

There's also the comp.emulators.ms-windows.wine Usenet newsgroup. You're more likely to get help on the mailing lists though, as these are what the developers use and monitor.

6.20. I think I've found a bug. How do I report this bug to the Wine programming team?

Bug reports should be submitted to our online Bugzilla system (http://bugs.winehq.org/). To increase developer productivity and facilitate a resolution to submitted bugs, **please read the Wiki article on Bugs**. A poor bug report may be marked INVALID and closed, leaving you no closer to resolving your problem. High quality bug reports are an essential part of making Wine better.

Please note that you should *generally* avoid submitting bug reports if you have used any third party applications or native DLL overrides.

7. Using Wine

7.1. How do I wipe the virtual Windows installation?

You can remove your virtual Windows installation and start from scratch by either removing the hidden `.wine` directory in your home folder.

Please note that in the following commands there should be no spaces in the path, particularly between `$HOME/` and `.whatever`.

```
rm -rf $HOME/.wine
```

or renaming it (in case you want to keep it for a backup):

```
mv $HOME/.wine $HOME/.wine-old
```

This will leave dangling MIME entries and desktop entries in `~/.local/share`. To remove these, see How do I uninstall Windows applications? Alternatively, you can stop Wine from creating them in the first place.

7.2. Can I store the virtual Windows installation somewhere other than ~/.wine?

Yes: `~/.wine` is just the default Wine "prefix" (a.k.a. "configuration directory" or "bottle").

You can change which prefix Wine uses by changing the `WINEPREFIX` environment variable (outside Wine). To do this, run something like the following in a terminal:

```
export WINEPREFIX=~/.wine-new
wine winecfg
```

Wine will then create a new prefix in `~/.wine-new`.

To use the default prefix, use the command `unset WINEPREFIX`. Or just set `WINEPREFIX` to `~/.wine`.

You can rename, move, copy and delete prefixes without affecting others, and each prefix has its own `wineserver` instance.

Wherever you see "`~/.wine`" or "`$HOME/.wine`" in this Wiki, you can usually replace it with "`$WINEPREFIX`".

7.3. How can I run two programs as if they were on different computers?

Example: You have server and client programs. One won't run in the presence of the other.

Using different Wine prefixes will help you here, since they simulate two Windows computers, in essence.

Run the first program as normal:

```
wine first-program.exe
```

The second needs to be run in a different prefix, so we need to change the WINEPREFIX variable:

```
WINEPREFIX="$HOME/.wine-second" wine second-program.exe
```

The `first-program.exe` and `second-program.exe` can be two copies of the same program.

7.4. I want my Windows application to be able to launch a Linux application to open documents

You can start Linux applications directly from Wine only if you specify the full path or use the shell:

```
/usr/bin/glxgears
# or
/bin/sh -c glxgears
```

You might also need winepath to translate the filename from Windows format to Linux format (see next question).

7.5. I want to associate some Linux program with a file type in Wine

There are two ways using which you can associate a Linux program with a file type. The first method is to use winebrowser and an alternative would be to write a shell script.

The example below uses winebrowser to launch the default pdf handler on your system (on Linux it uses xdg-openg). Save the lines below to a file `pdf.reg`.

```
[HKEY_CLASSES_ROOT\.pdf]
@="PDFfile"
"Content Type"="application/pdf"
[HKEY_CLASSES_ROOT\PDFfile\Shell\Open\command]
@="winebrowser \"%1\""
```

Import the .reg file into the registry using the command '`regedit $HOME/pdf.reg`'.

Another option is to use a shell script to call a Linux application. Save it as `run_linx_program` under `$HOME/bin`:

```
#!/bin/sh
$1 "`wine winepath -u "$2"`"
```

Don't forget to run '`chmod a+x $HOME/bin/run_linx_program`' to make it executable. Also make sure that `$HOME/bin` directory is in your `$PATH`. Otherwise it will not work.

To associate (say) **.pdf** files with the `acroread` Linux program save the following as "`$HOME/pdf.reg`" and then import it with the command '`regedit $HOME/pdf.reg`':

```
[HKEY_CLASSES_ROOT\.pdf]
@="PDFfile"
"Content Type"="application/pdf"
[HKEY_CLASSES_ROOT\PDFfile\Shell\Open\command]
```

```
@="/bin/sh run_linx_program acroread \"%1\""
```

You can reuse this script and just edit the registry file. For example to associate .doc documents with OpenOffice (ooffice):

```
[HKEY_CLASSES_ROOT\.doc]
@="DOCfile"
"Content Type"="application/msword"
[HKEY_CLASSES_ROOT\DOCfile\Shell\Open\command]
@="/bin/sh run_linx_program ooffice \"%1\""
```

7.6. What is this "winetricks" thing? Where do I get it?

winetricks is a shell script that downloads missing DLLs and fonts for you from the best known source. It was written to help Wine developers, but it turned out to be useful for end users, too. See the wiki page about winetricks for more info.

7.7. How do I install Internet Explorer in Wine?

If you just want an application to think you have IE installed, see My application won't run, and says it needs Internet Explorer above.

The Wine project does not support installing the real Internet Explorer, as it requires a huge number of native DLLs, which is hard to configure.

If you really need the real IE use winetricks. Set your fake Windows version to win2k and then select 'ie6' or 'ie7'. IE installed from winetricks is far from fully functional, but works well enough to, *e.g.*, test web page rendering. Please do not ask the Wine project for help if you run in to problems.

You should put IE in a separate wineprefix, with whatever app really needs it.

You may also try commercial solutions, such as CrossOver and Bordeaux.

7.8. How do I get Wine to launch an application in a virtual desktop?

You can do this with winecfg. Add the application in the Applications tab and then, in the Graphics tab, enable "Emulate a virtual desktop".

You can also use the following command (for wine-0.9.60 and older, this is the only method available):

```
wine explorer /desktop=name,1024x768 program.exe
```

Replace `program.exe` with the name of your program, and change the resolution to the size of the virtual desktop you want. Changing `name` allows you to open several desktops simultaneously.

7.9. How do I know what version of Wine I have?

Open up a terminal and run `wine --version`. It will say something like "wine-1.1.25"; if you are using Git then you will have a version along the lines of "wine-1.1.25-311-g3d6bb38".

TIP: You can find out what the latest release of Wine is from WineHQ's main page. Currently, Wine development releases come out every two weeks. Your operating system may ship with an out of date (obsolete) version of Wine. Depending on what OS you use, you may be able to add an update source to your package management system to keep up to date. Check the downloads page for details.

7.10. How does the Wine version numbering system work?

Each version of Wine has a release tag, in the following format:

> wine-*x*.*y*.*z*

This represents Wine version *x*.*y*.*z*. (For a list of tags, see here.)

- *x* is the major version number. This has changed precisely once since the current version numbering system was introduced, and is unlikely to change for a while.
- *y* is the minor version number. This changes every few years. If *y* is even, then this is a "stable" version: later releases are unlikely to break much. If *y* is odd, then this is a "development" version: later releases may cause regressions.
- *z* is the release number. For development releases, this is incremented every two weeks. For stable releases, this is incremented every few months.

If you are using Git, the tag will be generated by the `git-describe` command, and looks like:

> wine-*x*.*y*.*z*-*n*-g*ccccccc*

Where *n* is the number of patches/commits applied since *x*.*y*.*z* was released, and *ccccccc* is the first few hex digits of the most recent commit's id. Examples: wine-1.1.19-228-g1e256e4, wine-1.1.25-311-g3d6bb38, wine-1.1.32-584-g10b0b86.

Just before a stable release of Wine, there are so-called "release candidates", which have tags of the form:

> wine-*x*.*y*-rc*n*

Which is the *n*th release candidate for Wine *x*.*y*.

Examples:

Release tag	Date	Note
wine-0.9.60	18 Apr 2008	
wine-0.9.61	2 May 2008	Last beta
wine-1.0-rc1	9 May 2008	First release candidate
wine-1.0-rc2	23 May 2008	
wine-1.0-rc3	30 May 2008	
wine-1.0-rc4	6 Jun 2008	
wine-1.0-rc5	13 Jun 2008	
wine-1.0	17 Jun 2008	First "stable" version

wine-1.1.0	27 Jun 2008	First "development" version
wine-1.1.1	11 Jul 2008	
wine-1.1.2	25 Jul 2008	
...		

7.11. Should I run Wine as root?

⚠ ***NEVER run Wine as root!*** Doing so gives Windows programs (and viruses) full access to your computer and every piece of media attached to it. Running with sudo also has these same risks but with the added bonus of breaking the permissions on your ~/.wine folder in the process. If you have run Wine with sudo you need to fix the permission errors as described in the next question, and then run winecfg to set Wine up again. You should always run Wine as the normal user you use to login.

For Linux systems, all ideas that Wine needs root can be solved through Posix Capabilities (http://www.linuxjournal.com/article/5737) or Posix File Capabilities (http://www.ibm.com/developerworks/library/l-posixcap.html) or correcting other security settings.

As far as Windows programs are concerned, you are running with administrator privileges. If an application complains about a lack of administrator privileges, file a bug; running Wine as root probably won't help.

7.12. I ran wine with sudo or as root. How do I fix my permission errors?

You need to fix the permissions on your `~/.wine` directory, this is where all Wine state, configuration and any important data you might have such as installed programs, saved data within Wine programs, etc. are stored. Once you delete or fix the permssions on this directory, rerun Wine as a regular user always! Run the following to fix the permissions on your `~/.wine` directory if it now has root permissions.

```
cd ~
sudo chown -R $USER:$USER .wine
```

7.13. Which sound driver should I select in winecfg?

As of Wine 0.9.47 you shouldn't need to configure it any more, defaults should work just fine. For Wine 0.9.46 ALSA and OSS should work equally well. For Wine versions *before* 0.9.42 OSS works best, after that ALSA is slightly behind OSS since it might miss some voice chat capability.

7.14. Why isn't PulseAudio available?

The Wine project has decided not to pursue a Pulse driver for Wine at this time. We feel it is best to keep working on the more mature Wine ALSA driver. We are aware that some distributions use Pulse as the default, and this is unfortunate. PulseAudio is also known to be buggy when emulating ALSA/OSS and should be disabled when using Wine.

There is an unofficial PulseAudio driver for Wine, but it is unsupported. Do not submit bug reports while using it. Bug tracking this issue: bug 10495.

7.15. How can I make Wine fonts anti-aliased?

Support for subpixel font rendering was added to Wine in version 1.1.12, but it may not be enabled. Use winetricks and select one of the fontsmooth-gray, fontsmooth-rgb and fontsmooth-bgr options.

Bugs tracking these issues: bug 10342 and bug 16729.

7.16. How do I change the DPI (font size)?

First, you should try editing with winecfg. Go to the Graphics tab, and slide the "Screen Resolution" slider accordingly. Changes will not effect the winecfg window until you restart it.

If windows and fonts are so big you can't get to the controls in winecfg, see "Wine's windows and fonts are extremely large, and the winecfg window doesn't fit on the screen".

7.17. How do I edit the Wine registry?

The Wine registry is stored in the .reg files in `~/.wine`, however you should not edit these files by hand due to the encoding that they use. Always use the regedit program that comes with Wine. This can be run by typing `regedit` in the terminal. Wine's regedit is virtually identical to the Windows version of regedit and also supports importing and exporting of registry files. NEVER try and import your entire Windows registry, this will just break Wine.

See also: UsefulRegistryKeys

7.18. How do I configure a proxy?

If you want to use a proxy server for all HTTP connections, simply set the http_proxy environment variable. On many Linux distributions, configuring a network proxy, e.g. with the Network Proxy tool, does this for you automatically.

Alternatively, you can configure a proxy in the registry. There are separate locations for wininet.dll and winhttp.dll.

For wininet, use regedit to add the following values to the `[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings]` key:

```
"ProxyEnable"=dword:00000001
"ProxyServer"="proxy-server-address:port"
```

At present, Wine's `wininet.dll` does not support proxy autoconfiguration (PAC) scripts.

For winhttp, you'll need to use the `proxycfg.exe` utility to configure the registry. This utility is available in the system32 directory of a Windows installation, and MSDN describes its usage.

7.19. How do I get a debug trace?

When filing bugs, it is often necessary to get a trace (generally `+relay,+seh`, but you might be asked for specific debug channels). To retrieve a trace, run:

```
WINEDEBUG=+relay,+seh,+tid wine your_program.exe &> /tmp/output.txt
```

Then **attach** `/tmp/output.txt` to the bug. If the resulting file is larger than 1 MB, compress it with `bzip2` or `rzip -9` before attaching. There are some cases where the bug seems to disappear when WINEDEBUG is used with the right channel. Please mention that in the bug report. For a list of debug channels that are available in Wine, see DebugChannels

7.20. What applications/services conflict with Wine?

Many distributions are moving towards enabling Compiz/Beryl/other eye candy by default. The problem is that these programs conflict with Wine over the display driver. Disable these programs before using any Wine applications (especially games, or when noticing weird window problems). Also, disabling the Composite extension in `/etc/X11/xorg.conf` will most certainly prevent compositing from affecting Wine.

Pulseaudio is also known to be buggy when emulating Alsa/OSS and conflicts with Wine.

Keyboard switching tools such as `xneur` (`gxneur`, `kxneur`) and SCIM have been reported to conflict with keyboard input in Wine.

8. Applications

8.1. Does Wine support DirectX? Can I install Microsoft's DirectX under Wine?

Wine itself provides a DirectX implementation that, although it has a few bugs left, should run fine. Wine supports DirectX 9.0c at this time. Plans for DirectX 10 are underway.

⚠ **If you attempt to install Microsoft's DirectX, you will run into problems.** It is not recommended nor supported by Wine HQ to attempt this. You can install the runtime, but it will not run. The runtime needs access to the Windows drivers, and Wine cannot access them for obvious reasons. The only native Microsoft DLLs that could be useful anyway are the `d3dx9_xx.dll` type ones, and these require you to accept Microsoft's license. Additionally, versions of these DLLs are now part of the Wine tree. So, as Wine improves these DLLs will only become less relevant.

That said, there are some guides out there which describe how you can install Microsoft's DirectX. We reiterate: It is not recommended nor supported by Wine HQ to attempt this. Furthermore it is considered off topic in Wine HQ support mediums (such as the forums). Please use a clean Wine configuration folder before seeking help. (You may need to `rm -rf ~/.wine` and re-install your Windows applications.)

8.2. How come Wine doesn't work well with copy protection?

Copy protection uses several "dirty" methods to detect if discs are "real" or not. Some of these methods work in Wine but most do not, such as the extremely invasive StarForce system. Wine would need to be altered to allow for almost rootkit-like functionality of programs to get some of these copy protection schemes to work, so support is likely to be a long time off if ever. (Some people use illegally modified or "cracked" games with the copy protection removed, but the Wine project cannot advocate that.)

See also our page on copy protection.

8.3. I followed a howto I found on a website, and the application still doesn't work.

There are many unofficial howtos for various apps posted on blogs and forums. They are rarely maintained by their authors, and because of Wine's rapid rate of development, even ones that were correct at the time they were posted can quickly become outdated. Applying tweaks that are no longer needed in current Wine can actually prevent an app that now runs fine without tweaks from working at all (Office 2007 is a prime example). The only howtos supported here are the ones on this site. If you insist on following an outside one, ask its author for help.

If in doubt, start again installing the app in a fresh wineprefix and ask for help on the user forum.

8.4. I used a third party application (PlayOnLinux, WineBottler, etc.) to install or manage applications in Wine and need help.

Consult the support channels for whatever third party application you used. Links to some of these outside applications are provided as a convenience on the Third Party Applications page, but they are not supported here. Before seeking help on the forum/mailing list/IRC or filing bugs, reinstall your Windows application using plain Wine.

8.5. My application worked with an older version of Wine, but now it's broken!

These are called regressions. Please perform a regression test to identify which patch caused the regression, then file a bug report and add the regression keyword - we'll pay special attention to it, since they're a lot easier to fix when caught early.

8.6. I really need this particular application to work. How can I go about making this happen?

We're glad you asked. Please see our Helping Wine page for some suggestions.

Briefly, you can improve Wine yourself, convince someone else to do it for you, or try some complicated hacks involving native Windows DLLs rather than Wine's unfinished ones.

8.7. I'm writing a Windows app. How can it detect if it's running under Wine?

This is a bad idea. See the Developer FAQ.

9. Performance

9.1. What are the system requirements for Wine?

The rule of thumb is that if your application runs fine in Windows, it should run fine on the same system using Wine. Wine, along with the operating system you use to run it, generally requires less disk space and memory than Windows itself. If you're not currently running a Windows application, Wine won't consume any resources at all other than about 20 megabytes of disk space.

9.2. My 3D application/game is very slow (fps).

Usually a 3D performance issue, indicates that something is wrong with your OpenGL 3D drivers. See 3DDriverIssues for more information.

9.3. Is Wine slower than just using Windows?

Actually, Wine is sometimes faster. The speed of an application depends on a lot of factors: the available hardware and their drivers, the quality of the code in the APIs the application uses, and the quality of the code in the underlying operating system.

Driver code matters a lot. If you're running a graphics-heavy application using a video card with very poor drivers such as an ATI card under Linux, performance will degrade substantially. On the other hand, Linux has superior memory management, and comes out ahead of Windows in many CPU-related tasks; see benchmarks for more information.

Sometimes, bugs in Wine can make applications excessively slow; see Performance-related bugs.

That said, Wine works on correctness first, performance second.

9.4. I get lots of "fixme:" messages in the terminal and Wine runs a bit slow.

Ignore them. Generally speaking, a fixme message will not make much sense to someone unfamiliar with Wine development. They are for Wine developers. Also, it's important to understand that fixme messages often *do not* indicate a serious problem. Many applications will work fine even though Wine prints a few `fixme` messages. That said, they can still sometimes provide insight into how a particular application works (or doesn't work).

💡 If there are a very large number of these messages scrolling repeatedly, you can sometimes speed Wine up a little by turning them off altogether. You can do so by setting the `WINEDEBUG` environment variable to `-all`. For example, your command line could look something like:

```
WINEDEBUG=-all wine program.exe
```

More advanced users and programmers interested in debugging Wine should see DebugChannels and the Developer Documentation wiki pages. Here's an example to selectively turn off `fixme` messages from dsound and part of D3D only:

```
WINEDEBUG=fixme-dsound,fixme-d3d_surface wine program.exe
```

10. Risks

10.1. Wine is malware-compatible

Just because Wine runs on Linux doesn't mean you're protected from viruses, trojans, and other forms of malware.

There are several things you can do to protect yourself:

1. Never run executables from sites you don't trust. Infections have already happened.
2. In web browsers and mail clients, be suspicious of links to URLs you don't understand and trust.
3. Never run any GUI application (including Wine applications) as root. (See above.)
4. Use a virus scanner, e.g. ClamAV is a free virus scanner you might consider using if you are worried about an infection; see also Ubuntu's notes on how to use ClamAV. No virus scanner is 100% effective, though.
5. Consider removing the default Wine Z: drive, which maps to the unix root directory. This is only a weak defense, but it might help against some attacks. The downside to this is you **won't be able to run Windows applications that aren't reachable from a Wine drive (like C: or D:). This includes inability to install Wine-gecko (see bug 19873).** A work around is to copy/move/symlink downloaded installers to `~/.wine/drive_c` before you can run them.
6. If you're running applications that you suspect to be infected, run them as their own Linux user or in a virtual machine. (The ZeroWine malware analyzer works this way.)

10.2. How good is Wine at sandboxing Windows apps?

Wine does not sandbox in any way at all. When run under Wine, a Windows app can do anything your user can. Wine does not (and cannot) stop a Windows app directly calling Linux syscalls, messing with your files, altering your startup scripts, or doing other nasty things.

You need to use AppArmor, SELinux or some type of virtual machine if you want to properly sandbox Windows apps.

That said, winetricks does have a sandbox verb that does at least a partial job of isolating Wine programs from the rest of your system. It protects against errors rather than malice. It's useful for, *e.g.*, keeping games from saving their settings in random subdirectories of your home directory.

10.3. Nautilus can delete your home directory when you empty the trash!

In some versions of Gnome (2.21.90?), if you move ~/.wine to trash, and empty the trash, everything that you can write to in your root filesystem is then deleted (including the contents of your home folder).

This is said to be fixed in Gnome 2.21.92. See https://bugs.launchpad.net/ubuntu/+source/nautilus/+bug/188361 and https://bugs.launchpad.net/ubuntu/+source/gnome-applets/+bug/195207

If you are logged in as root, this bug is worse, it can delete your entire hard drive.

FAQ (last edited 2010-07-17 15:07:54 by DavidGerard)