Chapter 4. Configuring Wine

**Table of Contents**

4.1. [Using Winecfg](https://www.winehq.org/docs/wineusr-guide/config-wine-main%22%20%5Cl%20%22USING-WINECFG)

4.2. [Using the Registry and Regedit](https://www.winehq.org/docs/wineusr-guide/using-regedit)

4.3. [Other Things to Configure](https://www.winehq.org/docs/wineusr-guide/misc-things-to-configure)

Most of the most common configuration changes can be done with the **winecfg** tool. We'll go through an easy, step-by-step introduction to **winecfg** and outline the options available. In the next section we'll go over more advanced changes you can make using **regedit** as well as provide a complete reference to all Wine configuration settings. Finally, some things you might want to configure fall out of the scope of **winecfg** and **regedit**, and we'll go over those.

4.1. Using Winecfg

In the past, Wine used a special configuration file that could be found in ~/.wine/config. If you are still using a version of Wine that references this file (older than June, 2005) you should upgrade before doing anything else. All settings are now stored directly in the registry and accessed by Wine when it starts.

Winecfg should have been installed on your computer along with the rest of the Wine programs. If you can't figure out how to start it, try running **winecfg**.

When the program starts you'll notice there are tabs along the top of the window for:

* Applications
* Libraries
* Graphics
* Desktop Integration
* Drives
* Audio
* About

Changing settings in the Applications and Libraries tabs will have the most impact on getting an application to run. The other settings focus on getting Wine itself to behave the way you want it to.

Note: The Applications, Libraries, and Graphics tabs are linked together! If you have "Default Settings" selected under Applications, all of the changes made within the Libraries and Graphics tabs will be changed for all applications. If you've configured a specific application under the Applications tab and have it selected, then any changes made in Libraries or Graphics will affect only that application. This allows for custom settings for specific applications.

4.1.1. Application Settings

Wine has the ability to mimic the behavior of different versions of Windows. In general, the biggest difference is whether Wine behaves as a Win9x version or an NT version. Some applications require a specific behavior in order to function and changing this setting may cause a buggy app to work. Wine default Windows version is Windows XP. Some applications may perform better if you choose Windows 98.

Within the tab you'll notice there is a *Default Settings* entry. If you select that you'll see the current default *Windows Version* for all applications. A troublesome application is best configured separately from the Default Settings. To do that:

1. Click on the Add application button.
2. Browse until you locate the executable.
3. After it's been added you can choose the specific Windows version Wine will emulate for that application.

4.1.2. Libraries Settings

Likewise, some applications require specific libraries in order to run. Wine reproduces the Windows system libraries (so-called native DLLs) with completely custom versions designed to function exactly the same way but without requiring licenses from Microsoft. Wine has many known deficiencies in its built-in versions, but in many instances the functionality is sufficient. Using only builtin DLLs ensures that your system is Microsoft-free. However, Wine has the ability to load native Windows DLLs.

4.1.2.1. DLL Overrides

It's not always possible to run an application on builtin DLLs, so sometimes native versions will be recommended as a workaround for a specific problem. Some may be directly copied to the directory configured as c:\windows\system32 (more on that in the drives section) while others may require an installer, see the next section on **winetricks**. Native versions of these DLLs do not work: kernel32.dll, gdi32.dll, user32.dll, and ntdll.dll. These libraries require low-level Windows kernel access that simply doesn't exist within Wine.

With that in mind, once you've copied the DLL you just need to tell Wine to try to use it. You can configure Wine to choose between native and builtin DLLs at two different levels. If you have *Default Settings* selected in the Applications tab, the changes you make will affect all applications. Or, you can override the global settings on a per-application level by adding and selecting an application in the Applications tab.

To add an override for FOO.DLL, enter FOO into the box labeled New override for library: and click on the Add button. To change how the DLL behaves, select it within the Existing overrides: box and choose Edit. By default the new load order will be native Windows libraries before Wine builtin ones (*Native then Builtin*). You can also choose native only, builtin only, or disable it altogether.

DLLs usually get loaded in the following order:

1. The directory the program was started from.
2. The current directory.
3. The Windows system directory.
4. The Windows directory.
5. The PATH variable directories.

4.1.2.2. Notes About System or Missing DLLs

There are of course DLLs that Wine does not currently implement very well (or at all).

In case Wine complains about a missing DLL, you should check whether this file is a publicly available DLL or a custom DLL belonging to your program. In the latter case, check that you have installed your program correctly.

Most often applications will assume that a required redistributable package has already been installed and subsequently fail to run when the required dependencies are not met. For example:

err:module:import\_dll Library MFC42.DLL (which is needed by L"C:\\Program Files\\Foo\\Bar.dll") not found

Redistributable packages which install the necessary runtimes can be obtained through the use of [winetricks](http://wiki.winehq.org/winetricks). Note these components are subject to their own license and are not part of the Wine project. You should refer to the application's [AppDB](http://appdb.winehq.org/) entry for advice on what is required.

4.1.3. Graphics Settings

There are basically five different graphics settings you can configure. For most people the defaults are fine.

The first setting primarily affect games and is somewhat self-explanatory. You can prevent the mouse from leaving the window of a full-screen program (e.g. a game) and the default is to not have that box checked. That is mostly needed when using a virtual desktop.

You may find it helpful to tick Emulate a virtual desktop. In this case, all programs will run in a separate window. You may find this useful as a way to test buggy games that change (possibly unsuccessfully) the screen resolution. Confining them to a window can allow for more control over them at the possible expense of decreased usability. Sizes you might want to try are 800x600 (the default) or 1024x768.

4.1.4. Drive Settings

Windows requires a fairly rigid drive configuration that Wine imitates. Most people are familiar with the standard notation of the A: drive representing the floppy disk, the C: drive representing the primary system disk, etc. Wine uses the same concept and maps those drives to the underlying native filesystem.

Wine drive configuration is relatively simple. In **winecfg** under the Drives tab you'll see buttons to add and remove available drives. When you choose to add a drive, a new entry will be made and a default drive mapping will appear. You can change where this drive points to by changing what's in the Path: box. If you're unsure of the exact path you can choose Browse to search for it. Removing a drive is as easy as selecting the drive and clicking Remove.

**winecfg** has the ability to automatically detect the drives available on your system. It's recommended you try this before attempting to configure drives manually. Simply click on the Autodetect button to have Wine search for drives on your system.

You may be interested in configuring your drive settings outside of **winecfg**, in which case you're in luck because it's quite easy. All of the drive settings reside in a special directory: ~/.wine/dosdevices. Each "drive" is simply a link to where it actually resides. Wine automatically sets up two drives the first time you run Wine:

$ ls -la ~/.wine/dosdevices/

lrwxrwxrwx 1 *wineuser* *wineuser* 10 Jul 23 15:12 c: -> ../drive\_c

lrwxrwxrwx 1 *wineuser* *wineuser* 1 Jul 23 15:12 z: -> /

To add another drive, for example your CD-ROM, just create a new link pointing to it:

$ ln -s /mnt/cdrom ~/.wine/dosdevices/d:

Take note of the DOS-style naming convention used for links - the format is a letter followed by a colon, such as a:. So, if the link to your c: drive points to ~/.wine/drive\_c, you can interpret references to c:\windows\system32 to mean ~/.wine/drive\_c/windows/system32.

4.1.5. Audio Settings

Wine can work with quite a few different audio subsystems. You can see the selected driver that Wine figures out for you under the Audio tab.

You can manually select which device will be used for Output, Input, Voice output and Voice input. For example you can choose the digital output of your sound device instead of the analog one.

4.1.6. Desktop Integration

Wine can load Windows themes if you have them available. While this certainly isn't necessary in order to use Wine or applications, it does allow you to customize the look and feel of a program. Wine supports the newer MSStyles type of themes. Unlike the older Microsoft Plus! style themes, the uxtheme engine supports special .msstyles files that can retheme all of the Windows controls. This is more or less the same kind of theming that modern Linux desktops have supported for years. If you'd like to try this out:

1. Download a Windows XP theme. Be sure it contains a .msstyles file.
2. Use the Desktop Integration tab of **winecfg** to install and select the new theme.