

# Call of Duty 4 on Ubuntu 8.04 – WINE 1.1.1

Written by Rob Whalley – V1.3 - 29/08/2008

**DISCLAIMER:** This is a guide based on personal experience. It may not work for you, and I accept no responsibility for anything whatsoever that happens as a result of reading this document. Basically, the stuff below is a bit technical. If you mess up your computer, it's not my fault! I have tried to document any and all links whereby I obtained the information below. This guide is provided free of charge and may be modified / copied / etc. as if governed by a Creative Commons license. In other words, use it as you like.

The basis of this guide was derived from:

<http://www.linuxquestions.org/questions/linux-games-33/call-of-duty-4-on-linux-597725/page3.html>

I originally wrote this guide as a sort of Ubuntu / Fedora guide. However, since upgrading my PC I've been happily Ubuntu-ing away with my 64-bit version of 8.04 Hardy. And I like it that way... I'll leave the 32 bit and old Fedora stuff in here, but don't assume it still works... FC9 is now out so things may have moved on... If your experiences are different, or if you have anything to add to the guide, feel free to create a more definitive version using this as a template.

The 64-bit stuff is after the 32-bit instructions.

## Introduction

It is quite possible (though a learning curve for beginners) to install and play Call of Duty 4 on Linux using Wine. This guide focuses on the Single Player game only. You will need:

- The development version of WINE (WINE-Git) that can be patched before compiling
- The patch to fix the COD4 "video card or driver does not support separate alpha blend" error
- DirectX 9 Setup file
- A No-CD patch (possibly)
- Some DLL files
- A shell script to satisfy dependencies for Ubuntu

## Recommended

If you haven't already, refer to the following setup guides for Ubuntu / Fedora:

<http://ubuntuguide.org/wiki/Ubuntu:Hardy>

<http://www.mjmwired.net/resources/mjm-fedora-f8.html>

Invaluable starting guides for new Linux users (e.g. Configuring sudo on Fedora 8)

Normally, you would install WINE through yum (Fedora) or apt (Ubuntu) as a pre-compiled binary. However, as we need to patch WINE we'll need to install the development (or Git) version of WINE:

## Installing Git version of WINE – see also <http://wiki.winehq.org/GitWine>:

32-bit version (old guide, may be outdated):

- 1) Install GCC version 3.4 used to compile WINE:

<http://www.winehq.org/?issue=333#Compiling%20Wine:%20Don't%20use%20GCC%204.0>

<http://wiki.winehq.org/GccVersions>

(version 4.1 breaks Securom support, may be fixed in 4.2)

Ubuntu: `sudo apt-get install gcc-3.4`

Fedora: `sudo yum install gcc34`

- 2) Install patch, git, and git-core:

Ubuntu: `sudo apt-get install patch git git-core`

Fedora: `sudo yum install patch git git-core`

- 3) Satisfy dependencies for WINE-Git as described here:

[http://wiki.winehq.org/Recommended\\_Packages](http://wiki.winehq.org/Recommended_Packages)

Ubuntu: Download shell script for Hardy Heron release at above website and do `./hardy.sh`

Fedora:

```
sudo yum install alsa-lib-devel audiofile-devel esound-devel glib2-devel
libogg-devel libvorbis-devel pkgconfig qt-devel fontconfig-devel freetype-
devel libICE-devel libSM-devel libX11-devel libXau-devel libXcursor-devel
libXdmp-devel libXext-devel libXft-devel libXinerama-devel libXrandr-devel
libXrender-devel libXt-devel libjpeg-devel libmng-devel libpng-devel mesa-
libGL-devel xorg-x11-proto-devel libXi-devel sane-backends-devel libieee1284-
devel libusb-devel libXmu-devel ncurses-devel gnutls-devel libgcrypt-devel
libgpg-error-devel libXv-devel opencdk-devel expat-devel cups-devel
e2fsprogs-devel krb5-devel openssl-devel giflib-devel openldap-devel lcms-
devel cyrus-sasl-devel libxml-devel libxml2-devel libxslt-devel libxml
libXxf86dga-devel libXxf86vm-devel isdn4k-utils-devel libao-devel hal-devel
dbus-devel libXcomposite-devel
```

Note: I'm pretty sure there was a font package as well that was required when I tried to configure WINE-Git, but I can't remember which one ;) - If you encounter this, just `sudo yum install` the package.

- 4) Download the latest version of WINE-Git:

```
git clone git://source.winehq.org/git/wine.git wine-git
```

For me, this sometimes times out, but it seems to clear up if I visit <http://source.winehq.org/git/wine.git/> - though this could be my imagination! It's a useful page anyway, and worth visiting to see recent changes.

- 5) Download the following:

- 1.4 Official patch for COD4 from: <http://www.callofduty.com/>
- The patch for WINE-Git – the latest version I used could be downloaded here, but you have to sign up to the website to do so. Not a major chore, but I've included the patch source at the end of the document. Will probably go out of date soon! ;)  
<http://ubuntuforums.org/showpost.php?p=5286516&postcount=255>
- If you have problems with CD protection AND you own a LEGAL copy, you'll probably need a NOCD patch. You could try searching Google for "Call of Duty 4 NOCD". This step is controversial, but I leave it up to the beliefs / conscience of the reader. Personally, I believe if you bought the game fair and square it's not a problem. But hey... what do I know? Be aware that most sites have popups / adverts / possible malware so be careful where you get your NOCD patches from (and it's probably worth scanning them with anti-virus software first to be sure). Lots of NOCD files are in RAR format, so you may need to install *unrar* to uncompress them.

- The following dll files (e.g. from <http://www.dlldump.com/>)
    - ddrawex.dll*
    - mscoree.dll*
    - streamci.dll*
  - DirectX 9.0c from [http://filehippo.com/download\\_directx/](http://filehippo.com/download_directx/)
- 6) The patch should go into the wine-git folder. Next do the following:  
`patch -p1 < wine-1.1.0-3dmark.patch`
- 7) Now to configure, make and install WINE:  
 Ubuntu: `./configure CC=gcc-3.4`  
 Fedora: `./configure CC=gcc34`  
`make depend && make`  
`sudo make install`
- Note - if you keep the wine-git folder you can later remove wine with: `sudo make uninstall` or update wine-git with changes made, etc.
- 8) Setup WINE and prepare for DirectX installation. The majority of this step is detailed much more completely here:  
<http://wine-review.blogspot.com/2007/11/directx-90c-on-linux-with-wine.html>
- Install DirectX using guide here:

<http://www.wine-reviews.net/microsoft/directx-90c-march-2008-redistributable-on-linux-with-wine.html>

But using more up to date (June 2008) version of DirectX redistributables from Microsoft:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=822640ab-0983-4c41-9c70-632f6f42c557&DisplayLang=en>

- From a terminal, do: `winecfg`
- I recommend setting sound to *OSS*, *full hardware acceleration*, *16bits per sample*, *44100 Sample Rate*, and *no emulation* though this will vary with hardware / your requirements.
- You can turn off *desktop integration* to fix problems if you have problems with Gnome desktop and fullscreen modes – not required for KDE.
- Copy the dll files from step 5 to the `.wine/drive_c/windows/system32` folder
- Under libraries set a dll as native and exit winecfg.
- Open the file `.wine/user.reg` and search for `dlloverrides`, paste the following over the dll you added above:

```
[Software\\Wine\\DllOverrides] 1206264929
"d3d8"="builtin"
"d3d9"="builtin"
"d3dim"="native"
"d3drm"="native"
"d3dx8"="native"
"d3dx9_24"="native"
"d3dx9_25"="native"
"d3dx9_26"="native"
"d3dx9_27"="native"
"d3dx9_28"="native"
"d3dx9_29"="native"
```

```
"d3dx9_30"="native"
"d3dx9_31"="native"
"d3dx9_32"="native"
"d3dx9_33"="native"
"d3dx9_34"="native"
"d3dx9_35"="native"
"d3dx9_36"="native"
"d3dxof"="native"
"dciman32"="native"
"ddrawex"="native"
"devenum"="native"
"dinput"="builtin"
"dinput8"="builtin"
"dmband"="native"
"dmcompos"="native"
"dmime"="native"
"dmloader"="native"
"dmscript"="native"
"dmstyle"="native"
"dmsynth"="native"
"dmusic"="native"
"dmusic32"="native"
"dnsapi"="native"
"dplay"="native"
"dplayx"="native"
"dpnaddr"="native"
"dpnet"="native"
"dpnhpast"="native"
"dpnlobby"="native"
"dsound"="builtin"
"dswave"="native"
"dxdiag"="native"
"mscoree"="native"
"msdmo"="native"
"qcap"="native"
"quartz"="native"
"streamci"="native"
```

- 9) Now run DX setup. Note that I had installed this first when using a prebuilt version of WINE. When I removed that and installed from Git-source, I have NO overrides in place, just the physical files from the previous install. And WINE don't seem to care either way...
- 10) Now install COD4 and the 1.4 update. Also check if you require the NOCD patch.
- 11) To run the game, change to your COD4 folder under your .wine folder and run **wine iw3sp.exe**
- 12) In game turn off Field Depth and Soften Smoke Edges. Soften smoke edges causes graphical problems and Field Depth causes slowness.

### Upgrading to the latest version of wine:

I created a folder under my home folder called **Compile**, e.g. /home/user/Compile. When running `git clone git://source.winehq.org/git/wine.git wine-git` from this folder, the process automatically creates the wine-git folder under Compile. I still get the weird timeouts unless I enter <http://source.winehq.org/git/wine.git/> in my browser first.

If you keep the Compile folder and its contents after your initial compilation and install of git-wine, then you can uninstall the current version by using the `sudo make uninstall` command from the `/home/user/Compile/wine-git` folder.

The easiest way to upgrade after this is to `cd ..` back to the Compile folder, and remove the entire wine-git folder (making sure you backup the cod4 patch file first). So in my case... (from the wine-git folder)

```
sudo make uninstall
mv wine-1.1.0-3dmark.patch /home/user
cd ..
rm -R -f wine-git
git clone git://source.winehq.org/git/wine.git wine-git
mv /home/user/ wine-1.1.0-3dmark.patch wine-git
cd wine-git
patch -p1 < wine-1.1.0-3dmark.patch
./configure CC=gcc34
make depend && make
sudo make install
```

I've found this to be the easiest way of upgrading to the latest version.

AND NOW...

The 64-bit version...

Oh the headaches I had doing this before I realised there was a wiki page for it. Sigh.

<http://wiki.winehq.org/WineOn64bit>

Setup the basics as above (e.g downloading Git, you'll need the DirectX files, etc.)

The dependencies I ended up installing:

```
sudo apt-get install libXxf86vm-dev libxcomposite-dev libhal-dev libncurses5-dev libsane-
dev libgphoto2-2-dev libcapi20-dev libcupsys2-dev libldap2-dev libxml2-dev libxslt-dev
ia32-libs lib32gcc1 lib32stdc++6 gcc-multilib flex bison
```

Ripped from the official guide linked to above:

*These instructions will help you compile WINE*

```
sudo apt-get build-dep wine
```

*On Debian you may need to add some more, or run the build-dep command after adding the AptRepository.*

*Wine's configure requires .so links, and for some reason Ubuntu 64 bit doesn't make .so links for the 32 bit libraries. Accordingly, to avoid missing libraries and compile smoothly you'll have to do some linking by hand.*

*You can make these links in a temporary folder within the wine tree.*

```

mkdir -p `pwd`/lib32
ln -s /usr/lib32/libX11.so.6 `pwd`/lib32/libX11.so
ln -s /usr/lib32/libXext.so.6 `pwd`/lib32/libXext.so
ln -s /usr/lib32/libfontconfig.so.1 `pwd`/lib32/libfontconfig.so
ln -s /usr/lib32/libGL.so.1 `pwd`/lib32/libGL.so
ln -s /usr/lib32/libGLU.so.1 `pwd`/lib32/libGLU.so
ln -s /usr/lib32/libXrender.so.1 `pwd`/lib32/libXrender.so
ln -s /usr/lib32/libXinerama.so.1 `pwd`/lib32/libXinerama.so
ln -s /usr/lib32/libXxf86vm.so.1 `pwd`/lib32/libXxf86vm.so
ln -s /usr/lib32/libXi.so.6 `pwd`/lib32/libXi.so
ln -s /usr/lib32/libXrandr.so.2 `pwd`/lib32/libXrandr.so
ln -s /usr/lib32/liblcms.so.1 `pwd`/lib32/liblcms.so
ln -s /usr/lib32/libpng12.so.0 `pwd`/lib32/libpng.so
ln -s /usr/lib32/libcrypto.so.0.9.8 `pwd`/lib32/libcrypto.so
ln -s /usr/lib32/libssl.so.0.9.8 `pwd`/lib32/libssl.so
ln -s /usr/lib32/libxml2.so.2 `pwd`/lib32/libxml2.so
ln -s /usr/lib32/libjpeg.so.62 `pwd`/lib32/libjpeg.so
ln -s /usr/lib32/libXcomposite.so.1 `pwd`/lib32/libXcomposite.so
ln -s /usr/lib32/libcups.so.2 `pwd`/lib32/libcups.so
ln -s /usr/lib32/libXcursor.so.1 `pwd`/lib32/libXcursor.so
ln -s /usr/lib32/libdbus-1.so.3 `pwd`/lib32/libdbus-1.so
ln -s /usr/lib32/libhal.so.1 `pwd`/lib32/libhal.so
ln -s /usr/lib32/lib sane.so.1 `pwd`/lib32/lib sane.so
ln -s /usr/lib32/libgphoto2.so.2 `pwd`/lib32/libgphoto2.so
ln -s /usr/lib32/libgphoto2_port.so.0 `pwd`/lib32/libgphoto2_port.so
ln -s /usr/lib32/libldap-2.4.so.2 `pwd`/lib32/libldap.so
ln -s /usr/lib32/libldap_r-2.4.so.2 `pwd`/lib32/libldap_r.so
ln -s /usr/lib32/liblber-2.4.so.2 `pwd`/lib32/liblber.so
ln -s /usr/lib32/libxslt.so.1 `pwd`/lib32/libxslt.so
ln -s /usr/lib32/libcapi20.so.3 `pwd`/lib32/libcapi20.so
ln -s /usr/lib32/libjack.so.0 `pwd`/lib32/libjack.so
ln -s /usr/lib32/libodbc.so.1 `pwd`/lib32/libodbc.so

```

*It is ok if you delete this directory after compiling Wine (configure is linking to the real libs, not the symlinks we just made)*

Run...

```
./configure
```

...build and install with:

```

CC="gcc-4.2 -m32" LDFLAGS="-L/lib32 -L/usr/lib32 -L`pwd`/lib32 -Wl,-rpath,/lib32 -Wl,-rpath,/usr/lib32" ./configure -v

make depend && make

sudo make install

```

## Further notes:

- If you're having problems with the Git version of WINE, try here first: <http://wiki.winehq.org/GitWine>
- Remember, you can uninstall WINE with the `sudo make uninstall` command if run from the wine-git folder.
- To reset your git-wine version to the latest version and remove all patches, change to the wine-git folder and do:  

```
git reset --hard origin  
git fetch ; git rebase origin
```

Note that I don't believe this works if wine has actually moved a up a version (e.g. from 0.9.57 to 0.9.58)
- Do `git status` in the wine-git folder to see which files have been modified / which files in the wine-git folder are untracked.
- To learn about debugging WINE if you have problems and need help, go here first:  
<http://www.winehq.org/site/docs/winedev-guide/wine-debugger>
- Two debugging variants I've used in the past are:  

```
WINEDEBUG=+relay wine program_name.exe  
WINEDEBUG=+relay,+tid,+server,+seh wine program_name.exe
```
- To output a text file of the debug to the same folder as the exe file, add the following to the command  

```
> debug.log 2>&1
```

e.g.

```
WINEDEBUG=+relay wine program_name.exe > debug.log 2>&1
```

- If playing in Gnome, you may find it necessary to tun of GL effects (particularly if using Compiz).

The 1.1.0 patch (a.k.a 'wine-1.1.0-3dmark.patch'):

```
diff -Naur wine.old/dlls/wined3d/directx.c wine/dlls/wined3d/directx.c
--- wine.old/dlls/wined3d/directx.c      2008-06-27 16:24:42.000000000 +0200
+++ wine/dlls/wined3d/directx.c      2008-06-29 21:13:36.000000000 +0200
@@ -847,7 +847,7 @@
     }
     if (gl_info->supported[ARB_MULTITEXTURE]) {
         glGetIntegerv(GL_MAX_TEXTURE_UNITS_ARB, &gl_max);
-        gl_info->max_textures = min(MAX_TEXTURES, gl_max);
+        gl_info->max_textures = 8;
         TRACE_(d3d_caps) ("Max textures: %d\n", gl_info->max_textures);

         if (gl_info->supported[NV_REGISTER_COMBINERS]) {
@@ -2952,8 +2952,12 @@
                                     WINED3DPMISCCAPS_CLIPTLVERTS |
                                     WINED3DPMISCCAPS_CLIPPLANESCALEDPOINTS |
                                     WINED3DPMISCCAPS_MASKZ |
+        +        WINED3DPMISCCAPS_TSSARGTEMP |
+        +        WINED3DPMISCCAPS_FOGANDSPECULARALPHA |
+        +        WINED3DPMISCCAPS_SEPARATEALPHABLEND |
                                     WINED3DPMISCCAPS_BLENDOP |
-        -        WINED3DPMISCCAPS_MRTPOSTPIXELSHADERBLENDING;
+        +        WINED3DPMISCCAPS_MRTPOSTPIXELSHADERBLENDING |
+        +        WINED3DPMISCCAPS_FOGVERTEXCLAMPED;
                                     /* TODO:
                                     WINED3DPMISCCAPS_NULLREFERENCE
                                     WINED3DPMISCCAPS_INDEPENDENTWRITEMASKS
@@ -3217,8 +3221,8 @@
     pCaps->MaxVertexBlendMatrixIndex = 0;

     pCaps->MaxAnisotropy = GL_LIMITS(anisotropy);
-    pCaps->MaxPointSize = GL_LIMITS(pointsize);
-
+    //pCaps->MaxPointSize = GL_LIMITS(pointsize);
+    pCaps->MaxPointSize = 64.0f;

     pCaps->VertexProcessingCaps = WINED3DVTXPCAPS_DIRECTIONALLIGHTS |
                                     WINED3DVTXPCAPS_MATERIALSOURCE7 |
@@ -3235,7 +3239,13 @@
     pCaps->MaxStreamStride = 1024;

     /* d3d9.dll sets D3DDEVCAPS2_CAN_STRETCHRECT_FROM_TEXTURES here because StretchRects
is implemented in d3d9 */
-    pCaps->DevCaps2 = WINED3DDEVCAPS2_STREAMOFFSET;
+
+    //pCaps->DevCaps2 = WINED3DDEVCAPS2_STREAMOFFSET;
+    pCaps->DevCaps2 = WINED3DDEVCAPS2_STREAMOFFSET |
+    +    WINED3DDEVCAPS2_CAN_STRETCHRECT_FROM_TEXTURES |
+    +    WINED3DDEVCAPS2_PRESAMPLEDDMAPNPATCH |
+    +    WINED3DDEVCAPS2_VERTEXELEMENTSCANSHARESTREAMOFFSET;
+
     /* TODO: VS3.0 needs at least D3DDEVCAPS2_VERTEXELEMENTSCANSHARESTREAMOFFSET */
     pCaps->MaxNpatchTessellationLevel = 0;
     pCaps->MasterAdapterOrdinal = 0;
```

Please be careful copying and pasting the text above! Word processors frequently change the 'straight' double quotes to the inverted / angled double quotes, so make sure you've got the straight quotes or you'll have difficulties. Basically, probably best to copy the lot to a text file and replace all the quotes.