Outline/notes for

Getting Apps Working on Wine

Penguicon 2015

Jeff D. Hanson

(CC BY-SA 4.0)

Overview

- Not an emulator. Compatibility layer and Windows API - Linux API call translator.
- x86/x86-64 only but there have been some attempts to support PPC and ARM CPUs.
- Targets functions and utilities that are a core part of the Windows OS, especially those that are common dependencies of Windows programs.
- Functionality can lag Windows releases by years, especially with complex components (DirectX).
- Native (Windows) components can be installed to supplement or replace Wine's built-in functions but this is not helpful for improving Wine itself.
- Drivers not included. It's not Windows and has limited direct hardware access. These would need kernel support. It is possible, e.g. NDISwrapper.
- Very strict on code quality and preventing regressions. Can frustrate contributors, leaving patches wasting away in the bug database (from bit rot).
- Forks, derivatives, and parallel developments: CrossOver, Cedega/GameTree, Wine Staging. Distros may include additional patches (PulseAudio support in Ubuntu packages).
- Many developers but only a few full-time and some work only on select APIs. Alexandre Julliard is the project leader and is employed by CodeWeavers.

Components and Commands

- Core dynamic link library (DLL) functions and utilities get priority (explorer vs. clock).
- Others are lower priority due to a lack of developer resources, not normally included with a Windows installation, or because native versions are readily available (included with apps or legally available via download). Visual Basic runtimes for example.
- Primary configuration tool is winecfg.
- Clones of Windows utilities vary in completeness. Those that are dependencies of many apps or their installers tend to be closest to their Windows equivalents. Examples:

  Registry Editor (regedit) is used for editing the Windows registry. Wine also stores its configuration in the registry. Wine's registry is stored as text files (system.reg, user.reg, userdef.reg). They can also be edited with any text editor. Each line is timestamped with the epoch but this is not required (probably used for debugging).

  Wine File Manager (winefile) is a File Manager clone. It can browse directories and launch executables but does not have any file manipulation ability.

  Wine Explorer (explorer.exe) is a Windows Explorer clone but is only slightly more complete than winefile (can delete files).

Wine's Program Manager (progman.exe) is a Windows Program Manager clone but only has enough functionality to support installers.

- Command line tools. Examples:

  start.exe is used launch the default application associated with a particular file and is often used in Linux desktop menu entries (XDG/freedesktop.org menu files).

  MSI install files are supported by Wine's msiexec.

  DLLs can be registred by Wine's regsvr32 just like in Windows.

  Control Panel applets (*.cpl) can be launched by control.exe but Wine only includes limited versions of Internet Settings (inetcpl.cpl), Add/Remove Programs (appwiz.cpl), and Game Controllers (joy.cpl). They can also be accessed via “Desktop/My Computer” in Wine Explorer.

  The registry can be modified via the command line with the Registry Editor, “reg”, or any file manipulation tool (sed).

- Some complex Windows components originate from third-parties. Examples:

  Mono for NET Framework (incomplete Windows Forms support)

  Mozilla's Gecko engine for iexplorer (no ActiveX support).

  DOSBox for DOS/CMD shells

- Some frivolous: clock
- Third party: winetricks, PlayOnLinux

Compatibility

- win32 best, win64 good, win16 bad (limited debug tools available)
- Can identify itself as any Windows version via the setting in winecfg but this does not change Wine's behavior.
- Not all functions are supported, only those required to get apps working. The rest are "stubs" that return a status but do nothing. Many apps work with stubs either because they don't actually depend on the result or have safe defaults. If a stub is inadequate then the function needs to be implemented but not necessarily every option is required.
- Lags supporting new Windows features, like major DirectX releases, but so do most other programs.
- Lags new Linux features, e.g. PulseAudio
- App workarounds for Windows bugs may cause errors if bugs are not present. Different reported Windows versions may change app behavior which triggers different bugs.
- Enforces some Windows limits such as font name/description length.
- System bugs/limits:

Kernel fix for a x86_64 security problem causes win16/win32 to break (bug #36664).

X.org video driver bugs are often triggered by Wine and cause lockups.

Z-axis dialog problem: app windows appear to freeze due to a dialog stuck behind them that is waiting for user input.

- Hardware limits: Unsupported display resolutions and bit depths, CPU bugs, speed regulation issues, large memory/drive space calculation errors (mostly affects installers)
- No direct access to hardware so no driver support. Affects some input devices with proprietary configuration tools (elite gaming mice).
- Case-insensitive filesystem support on a case-sensitive filesystem can result in duplicate files when applying app patches from outside of Wine (e.g., from a *.zip archive).
- A app may fail to function if it depends on another app that is broken under Wine (QuickTime, Flash)
- DRM/Copy Protection code may false trigger a failure due to Wine anomalies or lack of direct hardware access. Some hardware dongles may be supported.

Performance

- Usually slower. Sometimes much slower. Rarely faster. A slow Linux driver does not help (fglrx).
- Future improvements: CSMT (command stream multi-threading) . Was supposed to be the defining feature of 1.7 but may be postponed.
- Network stalls. Can cause indirect problems via gvfs (Gnome Virtual Filesystem) with mounted network shares.
- Input device problems (cursor lag)
- Race conditions: intermittent hangs (thread timeouts)
- Case-sensitivity look-ups with many files (Baldur's Gate)

Wine tools and settings

- WINEARCH
- WINEPREFIX
- WINEDEBUG: massive info, massive disk space (GBs per minute)
- winecfg
- winefile, progman, explorer
- msiexec (old format only)
- regedit and reg
- winetricks (dlls)

Other tools

- Dependency Walker
- ProtectionID (6.4.0 or newer version available in 2015)
- Double Commander 32-bit (or other native file manager)
- jsMSIx (MSI file extraction)
- RAD Video Tools (Smacker and Bink video conversion)
- Filmerit and GSpot (DirectShow info)

Logs and monitoring

- .xsession-errors
- .cache/upstart/startxfce4.log
- /var/log/Xorg.#.log, syslog, kern.log
- htop -u $USER
-

Scenarios

Baldur's Gate: case-sensitivity lookups, WeiDU lower-case requirement
The Fairly OddParents: Breakin' Da Rules!: Misleading error message
18 Wheels of Steel: Big City Rigs: Unsupported hta shortcuts
Double Commander: 32-bit works, 64-bit doesn't
DraftSight: 32-bit WMI (Win9x), no 64-bit (no public source)
MechCommander (non-fatal CPUID.EXE crash) vs. MechCommander 2: Different engines, command-line workarounds
Microsoft Agent: no built-in tools for testing, use MAPV
Wars and Warriors: Joan of Arc: SecuROM failure, cracked exe and non-CD versions available
The One Ring 3D Screensaver: Faulty menu entry. Use command line.
Quantum Gate: 16-bit QuickTime 1.0

Support

Bug reports
AppDB (submission, ratings, rating system limits)
Wine forums (also those of your distro)
Reddit: /r/winehq /r/wine_gaming
IRC: #winehq on FreeNode

Contributing

Bug reports, attaching logs
Publically available vs. your copy only (you get to be gatekeeper)
Compiling, bisecting, and patching

GameRanger bug reports: good (#38076) vs. bad (#32204)
DraftSight 64-bit immediate fix: bug #38330 (commit)
Copy protection: Wars and Warriors: Joan of Arc

Alternatives

Re-releases (Steam, GOG); less bugs
Commercial ports (Loki, Icculus, Humble Bundle, GOG)
Open sourced (MechCommander 2, Id Software), content vs. code
VM or DOSBox/Win1-3.11
Reversed-engineered clones (Double Agent, GemRB, OpenTTD)