



**White Paper:
Windows Legacy
Application Support
Under Wine**



Windows Legacy Application Support Under Wine

Wine offers a viable alternative for keeping legacy Windows apps running in perpetuity.

Overview: With the constant pressure from Microsoft to migrate to newer versions of Windows, many users are left with unpalatable options for running their older applications. Wine offers a viable, inexpensive alternative for keeping these legacy applications functioning in perpetuity.

It is a truism that the needs of customers and vendors are often at cross-purposes. Nowhere has this been more vividly demonstrated than in the forced upgrade cycles of the Microsoft Windows platform, which have completely reversed the leverage customers normally wield over their suppliers. In many cases, end-users have seen little utility in switching to newer versions of Windows, but have been left with essentially zero choice in the matter.

Forced OS upgrades have, in turn, forced the migration of many legacy applications. These older Windows programs—which in many cases are still perfectly functional—must either be redeveloped for a newer Windows platform, or replaced. This places yet another cost on top of the cost of the OS upgrade.

However, this cycle, which once appeared unbreakable, now seems to be weakening. For one thing, Windows is no longer as dominant as it once was on the desktop. The rise of both Mac OS X and Linux have begun eating into the monopolistic power of Microsoft to dictate these cycles.

The MS OS forced-upgrade cycle finally seems to be weakening.

Second is the fact that users now have a realistic alternative for running their legacy apps. Utilizing the power of Wine, an open-source reimplementation of the Win32 OS, in many cases these legacy applications can be run in perpetuity on alternative hardware platforms. The remainder of this paper explains how this is possible, and how companies can harness the power of Wine to contain their legacy application migration costs.

Wine acts a compatibility layer between a Windows application and the underlying Unix-based operating system. In essence, the Windows app makes requests to Wine (i.e. “repaint the screen,” “provide a printer dialog box,” etc.) and Wine instructs the native Unix OS on how to fulfill these needs. The result is that Windows applications run as if they are native to the UNIX OS.

Wine in many cases offers better legacy support that newer versions of Windows running in “Compatibility Mode”

As Wine has progressed as a technology, it has accommodated newer versions of the Win32 API. Thus, support for Windows 3.1 applications was augmented by support for Win95, Win98, and so on. The beautiful thing about Wine, though, is that there has never been a pressure on the part of the Wine Project to intentionally obsolete the older portions of the codebase; quite the opposite. One of Wine’s inherent virtues is its ability to run older applications, and this ability continues to be enhanced. Indeed, Wine in many cases now offers better legacy support for older Windows applications than newer versions of native Windows running in “Compatibility Mode.”

CodeWeavers has taken this concept one step further with the introduction of a technology called “Bottles.” A Bottle is a portable, self-contained Windows environment. By using bottles, a user can establish exactly the “flavor” of Windows they want to run—Win95, WinXP, etc.—and then install Windows applications into the bottle to be run. This allows Wine users to establish multiple controlled Windows environments. This can be very useful for, say, Web developers, who want to see how a web site might look on IE 6.5 running under both Win2000 and WinXP. By running bottles, CrossOver users can successfully run Windows 3.1 applications in their corporate environments, a decade after official support for 3.1 was discontinued.

It must be noted, of course, that not all Windows applications currently run under Wine. In many cases, some work may be required to get the application to run under Wine. However, the upside to doing this work is that it’s a one-time cost. Once the application runs under a given version of Wine, the customer can continue running the application under that version of Wine *forever*, without anybody pressuring them to upgrade to the latest, greatest version of Wine. The choice of if and when to upgrade is *entirely* in the customer’s hands.

Not only that, but since Wine is open source, the end user retains the ability to use Wine forever. Wine isn’t controlled by a single developer; it’s controlled collectively by its licensing structure. This assures that Wine’s users always retain access to their codebase. They can modify and enhance Wine at any time, so long as they return their changes to the Project.

The net result is that end-users now have more options regarding their legacy applications than ever before. As Microsoft’s hold on the desktop has weakened, and Wine has matured, end-users can now realistically contemplate breaking out of the forced upgrade cycle. CodeWeavers, of course, welcomes the opportunity to work with anyone looking for assistance in using Wine in this manner. Please contact sales@codeweavers.com, or (1) 651-523-9300 to discuss these opportunities.