

Second System

Version 2.2

Compatibility enhancement for Microsoft
Windows® operating systems

Document updated November 5, 2025

Copyright 2025 Legacy Software Corporation

Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation.

What does Second System do?

Second System is an operating system enhancement tool which improves the software compatibility of Windows Vista and Windows 7 systems by introducing new, custom versions of system modules. These modules satisfy the requirements of a variety of newer software applications and allow them to run.

How do I install Second System?

Run "InstallSecondSystem.cmd" from the Second System installation directory and press "Enter" when directed, then log out and log back in (or restart your computer).

How do I uninstall Second System?

Run "RemoveSecondSystem.exe" from the Second System installation package and reboot.

What should I expect to run with Second System enabled?

As of the date of writing, the following applications should run:

- Latest version of Mozilla Firefox
- Latest version of Google Chrome
- Latest version of Steam client
- Latest version of BeamNG.drive
- Current releases of many Electron/CEF applications such as Visual Studio Code, Signal, etc.
- LibreOffice 25.2.5.2 (Windows Vista; already compatible with Windows 7)
- WACUP 1.99
- and many other applications!

Some applications have known issues:

-Windows Installer: there are issues running the service through Second System, which means that it is currently set to bypass Second System. Therefore, it is presently incapable of running modules that are incompatible with Windows 7, specifically the Google Chrome and Microsoft Edge Enterprise MSI installers.

-Mozilla Firefox: web installer is incompatible. Use standard offline installer.

-Mozilla Firefox: Window buttons also do not render properly by default; this can be corrected by selecting the titlebar to appear separately from the tab strip, or downloading and installing the Segoe Fluent Icons font here:

<https://learn.microsoft.com/en-us/windows/apps/design/downloads/#fonts>.

-Google Chrome: web installer and Enterprise MSI package are incompatible. Use Enterprise "Bundle" installer.

-Google Earth Pro: use offline installer on Windows Vista and 7. There may also be some issues with DXGI/D3D10 on Windows Vista.

-Notepad++: disable Second System on Windows 7 due to DirectWrite incompatibility.

What if my application does not run as expected?

If the application already ran on the system before Second System was installed:

Right-click on the application's drop-down menu and select "Disable Second System". The application will launch without Second System's intervention. This setting will be preserved for future runs of the application, which means a single click will be sufficient afterwards.

If the application requests a certain version of Windows:

The default profile in Second System will launch applications with the version information from Windows 10, build 1507. The application drop-down menu will also be populated with profiles targeting other versions of Windows, such as Windows 7, Windows 8.1 and Windows 11. Clicking on each option will also change the default preference for the application.

What if I want to only enable Second System on a per-application basis or use a different set of defaults?

Create the registry value *DefaultProfile* in the key `HKEY_CURRENT_USER\Software\LSC\SecondSystem`, type `REG_SZ` (string). Set the value to the name of your preferred profile; set it to "Passive" if you do not want Second System to be applied to applications by default. This default can still be overridden by the right-click menu as shown above, and applications that were previously run through Second System will have the same settings as before.

If any other type of error appears:

Please contact LSC Support. The following section of the manual may also be of interest.

Second System Technical Information

Second System is built around a "launcher" application which creates processes in a suspended state and then injects Second System's `ntdll.dll` wrapper, named `ntdlx.dll`, into the process. The wrapper then uses the information for the application's specified profile to determine the operating system version to use for identification purposes, as well as to determine which modules are to be substituted.

Second System profile information is stored in the registry key `HKEY_CURRENT_USER\Software\LSC\SecondSystem`. The `SecondSystem` key contains the paths of every application launched through Second System as well as its associated profile.

Profiles themselves are comprised of three keys in the following naming format, where 'X' is the name of the profile:

`X_Config` – containing version information

`X_Modulex64` – containing information regarding 64-bit module substitution

`X_Modulex86` – containing information regarding 32-bit module substitution

Config currently has five supported fields:

BuildNumber: `DWORD`, corresponds to the build number reported by the operating system

MajorVersion: DWORD, corresponds to the major operating system version (6 for Windows Vista, 7 and 8.x; 10 for Windows 10 and 11) reported by the operating system

MinorVersion: DWORD, corresponds to the minor operating system version (1 for Windows 7, 2 for Windows 8.0, 3 for Windows 8.1 and 0 for Windows Vista, 10 and 11) reported by the operating system

CSDVersionInt: DWORD, corresponds to the number of the current service pack. This is not to be confused with the service pack string shown in the “About Windows” box.

PlatformId: DWORD, corresponds to the platform ID (2 for Windows NT, 1 for Windows 9x and 0 for Win32s) reported by the operating system. Windows NT system components will generally fail if the value is not 2.

If there is no entry for a given field in the profile’s Config key, the value exposed by the operating system will be used in its place.

The Modulex64 and Modulex86 keys follow the same format; REG_SZ values where the name is the name of the target module and the data is a string representing the name of the replacement module. Some of the target modules, such as kernel33.dll or kernelbas0.dll, are not in fact, modules, but aliases for the original module, with the objective of preventing circular dependencies on wrapper modules.

The replacement modules can be located in any part of the system PATH, with System32/SysWOW64 taking precedence over other locations, including Second System’s module directories.

The module replacement capability can be used to implement redirection for one of the many virtual API set (api-ms-win-*) DLLs to existing DLLs, or in some cases, introduce modules from other versions of Windows to outright replace system defaults without compromising the system’s integrity*. *Then, you really do have a Second System!*

*ntdll, gdi32 and user32.dll are notable exceptions due to their hooks into kernel mode.