

## libusb-win32

libusb-win32 is a port of the USB library [libusb 0.1](#) to the Microsoft Windows operating systems (Windows 2000, Windows XP, Windows Vista and Windows 7; Windows 98 SE and Windows ME for versions up to 0.1.12.2 ). The library allows user space applications to access many USB device on Windows in a generic way without writing any line of kernel driver code.

Vista/7/2008/2008R2 64 bit are supported from version 1.2.0.0 since a Microsoft KMCS accepted digital signature is embedded in the kernel driver libusb0.sys. Moreover, libusb-win32 based device drivers can also be submitted for Microsoft WHQL testing. Several companies have successfully finished the WHQL testing with their libusb-win32 based driver package. This will allow the customers to install the driver without a warning under current 32bit/64bit Windows.

Take note that the latest Windows Logo Kit (WLK) license lists GPL as an "excluded license" so that it is not clear that WHQL submission is possible or not. You may have to check with Microsoft. The project administrators will try to address the issue by releasing the driver with dual license in the future: Modified\_BSD and GPL v3 license.

### Features:

- Can be used as a filter driver for existing, already installed devices. This feature allows libusb-win32 to communicate with many installed USB device. The filter driver may cause problems for certain device, including potential BSODs for certain setup. Therefore it should not be used by inexperienced end users. We recommend that it not to be redistributed to end users. Rather it should be considered more as a development tool for developers or power users of Windows.

Starting with v1.2.2.0, a GUI (filter wizard) is provided and it is only to attach the filter driver to particular USB device of interests. So this should be safer to use than the older behavior which tries to attach the filter to all USB device (acting as class filters to all possible device class). It is still possible to do that by using the console version of install-filter.exe but this is only recommended to be used by developers and power users.

When you use the device filter mode using the Filter Wizard GUI, you may have to repeat the process if you unplug the device and plug it back in.

- Can be used as a normal device driver for devices for which no driver exists (self build/developed USB hardware, etc). You can also replace the existing device driver with libusb-win32 device driver if desired. This is the preferred way to use libusb-win32.

When you use the Inf-Wizard GUI to replace a device with inbox driver (like HID or USB Mass Storage Device), you may need to repeat the process if you plug in a same device but with different serial number. This is because Windows prefers inbox driver (signed) and ranks it higher than the libusb-win32 driver package.

- The two methods described above can be used in parallel. However, the device driver mode is the preferred way to use libusb-win32 followed by using device filter driver mode. The class filter driver is not recommended to be used.
- 100% API and functional compatible with the libusb 0.1 project.
- Supports all USB transfer: Control, Bulk, Interrupt and Isochronous transfers. Take note the libusb 0.1 under other OS (Linux, Mac OS X, BSDs, etc) does not support Isochronous Transfer under other Operating Systems like Linux. libusb-win32 also has its own asynchronous API which is not available to libusb-0.1 under Linux or other OS.
- Supports all Standard Device Requests (control messages) described in chapter 9 of the USB specification.
- Supports vendor specific control messages.

### License

- The library (DLL and import lib, examples, installers) is distributed under the terms of the GNU Lesser General Public License ([LGPL](#)). Take note the installers in the older versions of libusb-win32 (before 1.2.0.0) is distributed under GPL, not LGPL.
- The driver portion (libusb0.sys) is distributed under the terms of the GNU General Public License ([GPL](#)).
- This license combination explicitly allows the use of this library in commercial, non-Open-Source applications. Read the licenses carefully and apply all of their requirements before including this library in a commercial application!
- In the future the project administrators will add a New BSD option to the license to address the WHQL Licensing issue.

### Support

If something isn't working as expected, make sure that you have installed the the latest version of libusb-win32 and the latest service packs for your OS before requesting for any support.

Available Support Options:

- A [mailing list](#) is available for discussions, questions, bug reports, feature request, and other issues. It is the preferred support channel. Please [subscribe](#) to the list first before posting.
- The project page offers different forms which can be filled out to get support, to report bugs, or to request new features. Please describe your problems and your system as precise as possible (OS, service packs, version of libusb-win32, type of device, output of "testlibusb-win.exe", etc.). This will make solving problems a lot easier.
- Debug version of the libusb-win32 are provided from libusb-1.2.3.0 onwards. Together with [Microsoft DebugView](#), detailed debug information can be printed out to facilitate easier debugging process.

You should also read the [libusb-win32 FAQ](#) (Frequently Asked Questions) Wiki Page.

### Download

Source code and binary packages can be downloaded from the [project download site](#). Source code is also available via anonymous [Subversion](#).

### Installation

#### Installation

#### Filter Driver Installation

- Please use the latest release version.
- Versions up until 0.1.12.2 have serious bugs related to the filter drivers under Vista and Windows 7 and some XP installations. Please use later versions (1.1.14.0 or newer). For 64bit Windows Vista/7/2008/2008R2, the version should be 1.2.0.0 or later. We always recommend users to use the latest release version available.
- The filters driver is installed by a user friendly GUI installer which makes the install and uninstall process easier and more secure. Starting with 1.2.2.0, a GUI for installing the filter driver (Filter Wizard) is the preferred way to use the filter. It only tries to attach the filter driver to a particular USB device. You can still use the command line install-filter.exe application to install class filter. But it is not recommended.
- Log in as a user with administrator privileges.
- **Download** the latest filter driver installer (libusb-win32-devel-filter-x.x.x.x.zip and then unzip, or libusb-win32-devel-filter-x.x.x.x.exe ).
- Close all applications which use USB devices before installing.
- Run the installer, and follow its instructions. Do not run the installer from an USB storage device, this is especially important for versions prior to 1.2.2.0.
- Run the test program (testlibusb-win.exe) from the system's start menu. This program will verify the correct installation and print the descriptors of the USB devices accessible by the library.
- A reboot may not be necessary but is recommended.

## Device Driver Installation

- Please use the latest release version.
- The device driver is distributed as a separate package which includes everything to use libusb-win32 for single devices as a normal device driver. The installation of the filter driver is not necessary any more!
- Log in as a user with administrator privileges.
- **Download** the latest device driver binary package (libusb-win32-bin-x.x.x.x.zip).
- Extract it to a temporary directory.
- Use the INF-Wizard program to generate the INF file (modify the vendor and product IDs, strings etc). Create different inf-files to install different types of devices (devices with different IDs).
- Unplug the device(s) from the system. This is optional.
- Open the Windows Device Manager and remove all incorrectly installed USB devices (device entries with a yellow exclamation mark). This is optional.
- Reconnect the device(s) to the system.
- When Windows asks for a driver, choose the inf-file(s) created above. Windows will warn that the driver is is not 'digitally signed'. Ignore this message and continue with the installation. Since version 1.2.0.0, a valid digital signature is embedded inside libusb0.sys for AMD/Intel x86\_64 version of Windows so that the users can install the driver as well under 64bit x86\_64 version of Windows Vista/7/2008/2008R2. Please read more about the [Microsoft Kernel Mode Code Signing \(KMCS\)](#) policy for more information.
- Open the Windows Device Manager to verify that the device is installed correctly. Run the test program (testlibusb-win.exe) from the 'bin directory'. It should print out the descriptors of your device(s).
- A reboot isn't necessary.
- Starting from version 1.2.1.0, the inf-wizard.exe has embedded driver binaries and provide an option to install the driver at the end of the process.

## Removing

### Removing the Filter Driver

- To remove device filter driver, run the GUI filter driver wizard to remove it.
- To remove the class filter driver open the Control Panel, open 'Software', choose the 'LibUsb?-Win32-x.x.x.x' entry, and remove it. Take note class filter driver is not recommended to be used.
- A reboot isn't necessary.
- If the above failed, you can manually run "install-filer -u" as admin. After that you can remove the other relevant files. Again, take note class filter driver is not recommended to be used.

### Removing the Device Driver

- The device driver can not be easily removed from the system. You can however try to use [usbdevview](#) to remove the entries. Under Vista and Windows 7, you can use [pnputil](#) to remove the driver package.

## Updating

### Updating the Filter Driver

- Remove the old version first (see above). This is the recommended method even though it is not strictly necessary.
- Install the new version as described above.

### Updating the Device Driver

- **Download** the latest device driver binary package (libusb-win32-device-bin-x.x.x.x.zip or tar.gz).
- Modify the inf-file as described in the Installation section.
- Open the Device Manager and select the device you want to update.
- Choose 'Properties->Driver->Update'. Disable the automatic installation and select the new inf-file manually.
- Since 1.2.1.0, inf-wizard.exe can be used to automatically install/update the device driver.

## Development

### Requirements to build libusb-win32 from source package

- A WinXP or later system.
- The Windows WDK 6001.18002 or later. You can get [WDK from Microsoft](#) . From 1.1.14.0 onwards, WDK will be the official tools to build the kernel driver files since it is the primary tools Windows driver developers use. It will be the official tools to build the release packages as well. MinGW and MinGW-w64 will not be supported as the driver building tool. Even if they can build the driver from the source codes, the resultant libusb0.sys will not be supported.
- **MinGW/MSys** is optional. It is required if you want to build the MinGW GCC import library to work with the libusb0.dll. It can also be used to build the test programs. [MinGW-w64](#) is not officially supported but it can be used to build the import library and test programs for 64bit Windows.

- [Cygwin](#) is not officially supported but there is a port of libusb-win32 for Cygwin which is outdated. Please do not use the driver binary and driver installation tools provide by the Cygwin package which may lead to BSODs or USB lockups.
- [Borland C++ 5.5](#) or above is optional, it is only required to build the import library file for Borland C++. Borland C++ is no longer supported so YMMV with the import library.
- [Inno Setup 5 or above](#) install system is required if you want to build the installer.

### Build Process

Until 0.1.12.2 (including 0.1.12.2), the package is built with a standard Makefile under MinGW/MSys. Starting from version 1.1.4.0, MinGW/MSys can only be used to build 32bit library and test programs.

- Download the latest source code.
- From a Msys shell run the command "make" to build the library and test programs.

Starting from version 1.1.4.0, batch files in the DDK\_MAKE directory will be the main tools to build the driver, library, and distribution packages.

- Download the latest source code.
- Edit make.cfg according to your particular setup (WDK directory, locations of MinGW, Borland C++, Inno Setup, etc).
- Use the provided batch files to build the 32bit/64bit drivers, library and test programs. Please refer to the output of "make.cmd" for more build options.
- To build the distribution archives and the installer run "make.cmd dist". Please refer to the output of "make.cmd" for more build options.

### For developers who want to use libusb-win32

- To use libusb-win32 in your own programs include the supplied header file usb.h, and link against the import library (libraries for GCC, BCC, and MSVC 32bit/64bit are available).
- To avoid any version conflicts, DO NOT include the DLL libusb0.dll in your application's directory. The DLL is part of the driver and installed automatically to the Windows system directory.
- If you are porting a libusb 0.1.x based program from Unix style systems to Windows, please take note that libusb-win32 differentiates IN Endpoints and OUT Endpoints. So IN Endpoint 1 is 0x81 and OUT Endpoint 1 is 0x01. One more difference is that you need to call `usb_set_configuration()` before `usb_claim_interface()`. This is especially for the filter drive. Take note this is not really necessary under Linux libusb-0.1.
- Please also refer to the following section on documentation and examples.

### Documentation and Examples

This is the detailed [libusb-win32 Documentation](#).

The excellent [libusb-1.0 documentation](#) can also serve as a good reference even though the API is not compatible.

The source codes can also serve as some kind of documentation for the libusb-win32 developers. However, examples are often the best documentations for Open Source Projects like libusb and libusb-win32, especially for the users of the library. Here are some links to [projects and code examples](#) using libusb-0.1 and libusb-win32. Language wrappers (DotNet?, Python, Perl, Ruby, Java, etc) are also listed.

### libusbK: Future of libusb-win32

[libusbK](#) is the next-generation libusb-win32 kernel driver and associated library. The kernel driver (libusbK.sys) is based on KMDF and it will work with either libusb-win32 API or native libusbK API. The library (libusbK.dll) will work with libusb0.sys, libusbK.sys and winusb.sys. It will have dual licenses (Modified BSD and LGPL/GPL with KMDF linking exceptions).

Currently it is under active development. libusb0.sys and libusb-win32 API will still be actively be supported. We plan to enhance libusb-win32 APIs further upon user request and we will even consider bring some of the the enhanced APIs to Linux and Mac OS X.

### Portability

libusb-0.1 is widely supported by operating systems like Linux, FreeBSD, NetBSD, OpenBSD, Darwin/MacOS X and Solaris. libusb-win32 is API compatible with libusb 0.1. So it will greatly help you porting your libusb-0.1 based application to Windows.

FreeBSD 8/9 includes a new FreeBSD-specific reimplementation of the libusb-0.1 API, so your applications will probably work there too. The source code for this library can be found [here](#).

### Mailing list

The libusb-win32-devel mailing list exists for both users of the library and developers interested in contributing to the library itself.

- [Subscribe](#)
- [Archives at SourceForge](#)
- [Archives at GMANE](#)
- [Archives at Nabble](#)

### Links

- [libusb](#): The main libusb project from which libusb-win32 is derived.
- [libusb.org](#): The new libusb website.
- [libusbdotnet](#): .Net and Mono wrapper for libusb, libusb-win32 and more.
- [pyusb](#): Python wrapper for libusb, libusb-win32 and more.
- [usb.org](#): The place to get USB specifications and other general information about USB.
- [www.lvr.com](#): A lot of information about USB hardware and software development.
- [Beyondlogic.org](#): Another site with information about USB hardware and software development.
- [USB Made Simple](#): Introduction to the ABCs of USB.
- [MCCI](#): Sells commercial generic USB drivers for Windows.
- [Jungo](#): Another vendor of generic USB drivers.

- [Thesycon](#): Another vendor of generic USB drivers.
- [Microsoft MSDN](#): You can find Windows specific information about USB Driver development from MSDN.
- [libusb-1.0 Windows Backend](#): Related project -- libusb-1.0 Windows backend
- [libwdi](#): Related project -- Windows Driver Installer library libwdi

© 2011 [Geeknet, Inc.](#) [Terms of Use](#) - [Privacy Policy](#)