

Wiki »

## Building and Installing libffi on Windows

On Windows, the `libffi-6.dll` library needs to be built from source code. An installed and working MinGW (32-bit or 64-bit, depending on Windows OS) is required. After downloading the source code it is required to compile it with `MinGW` for the required target architecture. It is possible to build 64-bit binaries on 32-bit Windows.

Follow these steps to build and install `libffi`:

1. Download the `MSYS (32-bit)` utilities from [https://proj.goldencode.com/downloads/mingw/msys32\\_20111123.zip](https://proj.goldencode.com/downloads/mingw/msys32_20111123.zip) (the original code was found at the [sourceforge msys page](http://www.mingw.org/wiki/MSYS) and details about the utilities can be seen in <http://www.mingw.org/wiki/MSYS>)
2. Unpack MSYS to a desired location. No special installation required. The commands below assume the `MSYS` package is unpacked into `c:\msys`. In case of using another directory correct the `MSYS` usage commands accordingly.
3. The `libffi` source code can be found at <http://sourceware.org/libffi/>. The specific version used by FWD can also be found at [https://proj.goldencode.com/downloads/libffi/libffi\\_3.2.1\\_sources.zip](https://proj.goldencode.com/downloads/libffi/libffi_3.2.1_sources.zip).
4. Unpack the `libffi` sources (e.g. if there is JDK installed `jar xf libffi_3.2.1_sources.zip`).
5. Run MSYS shell - `msys.bat` from `\msys` directory. Inside opened MSYS shell:

- Go to the libffi sources, for example: `cd c:\libffi-3.2.1`
- Prepare target build config:  
32 bit: `sh ./configure`  
or  
64 bit: `sh ./configure --build=x86_64-w64-mingw32 --host=x86_64-w64-mingw32`

The explicit architecture specification is mandatory for 64-bit Windows (and installed 64-bit `MinGW`). The reason is the `MSYS` is a 32-bit package and by default it cannot build a valid configuration for the 64-bit compiler.

- Run `make`

6. After a successful build, these binaries and include files are created:  
32 bit: `libffi-(root)/i686-pc-mingw32/.libs` and `libffi-(root)/i686-pc-mingw32/include`  
64 bit: `libffi-(root)/x86_64-w64-mingw32/.libs` and `libffi-(root)/x86_64-w64-mingw32/include`
7. Copy the headers and libraries to the corresponding MinGW folders. Copy the `.dll` files to the system dll directory (the directory where Windows stores commonly used dll, usually `%windir%\system32\`). The include files and libraries will be used in FWD native code compilation and the dlls are used at runtime to load `P2J.DLL` code:

- Include files  
32-bit: `copy libffi-(root)\i686-pc-mingw32\include\*.h \mingw32\i686-w64-mingw32\include\`  
64-bit: `copy libffi-(root)\x86_64-w64-mingw32\include\*.h \mingw64\x86_64-w64-mingw32\include\`
- Link libraries  
32-bit: `copy libffi-(root)\i686-pc-mingw32\.libs\libffi*.a* \mingw32\i686-w64-mingw32\lib\`

**64-bit:** `copy libffi-(root)\x86_64-w64-mingw32\.libs\libffi*.a* \mingw64\x86_64-w64-mingw32\lib\`

- DLL Runtime library

**32-bit:** `copy libffi-(root)\i686-pc-mingw32\.libs\libffi*.dll %windir%\system32\`

**64-bit:** `copy libffi-(root)\x86_64-w64-mingw32\.libs\libffi*.dll %windir%\system32\`

As a convenience, the **32-bit** and **64-bit** pre-built `libffi6.dll` archives are available. The packages already have the proper `i686-pc-mingw32` and `x86_64-w64-mingw32` directories to use.

© 2004-2017 Golden Code Development Corporation. ALL RIGHTS RESERVED.