Supported platforms

Some information on this page refers to LM-X License Manager v5.5 or newer, which added support for 32-bit Linux ARM. If you are using an earlier version, please refer to the documentation for releases prior to v5.5.

The following table lists the platforms currently supported by LM-X License Manager, as well as platform-specific information and limitations.

For Licensed Platform	Download Filename	Requirements/Limitations
FreeBSD (64 bit)	lmx-sdk-*version#*_freebsd_x64.sh	
Linux x64 (64 bit)	lmx-sdk-*version*#_linux_x64.sh	• The x64 build is compiled using glibc v2.28. • For SELinux, an LM-X-enabled application must have textrel_shlib_t to run. The LM-X client tries to add this library during runtime, but if the current user does not have permissions to modify the SELinux configuration, an administrator must issue the following command to add textrel_shlib_t: `semanage fcontext -a -t textrel_shlib_t '/tmp/xf-dll/xf-.*\.tmp'`
Linux ARM (32 bit) armhf, armv7	lmx-sdk-*version#*_linux_arm32.sh	
Linux ARM (64 bit) aarch64, arm64	lmx-sdk-*version*#_linux_arm64.sh	
Mac OS X (Universal) for 10.15+	lmx-sdk-*version#*_darwin_universal.sh	
Windows (32 bit) x86 compiler	lmx-sdk-*version#*_win32_x86.msi	All files have been digitally signed, except: • .NET, Java and dll redistributables • example code These files can be signed with your own digital certificate. The supported Windows versions are 8.1, 10 and 11.
Windows (64 bit) x64 compiler	lmx-sdk-*version#*_win64_x64.msi	

MinGW32 (with GCC 8.1.0)	lmx-sdk-*version*#_mingw32_x86.msi	**MinGW requirements and limitations**
		When using MinGW as a compiler, note that the following are currently *not* supported:
		• HARDDISK, BIOS, and DONGLE_HASPHL HostIDs (see LMX_Hostid)
		• Virtual machine detection (SHARE = VIRTUAL; see FEATURE settings)
		• The flags LMX_LOGICAL_CPU_COUNT and LMX_PHYSICAL_CPU_COUNT, used with LMX_Checkout.
		GCC version 8.1.0 must be used when building LM-X with MinGW. The exemplary output for GCC -v is as follows (using MinGW32 for our example):
		`COLLECT_GCC=gcc`
		`COLLECT_LTO_WRAPPER=C:/mingw32/mingw32/bin/../libexec/gcc/i686-w64-mingw32/8.1.0/lto-wrapper.exe`
		`Target: i686-w64-mingw32`
		`Configured with: ../../../src/gcc-8.1.0/configure --host=i686-w64-mingw32 --build=i686-w64-mingw32 --target=i686-w64-mingw32 --prefix=/mingw32 --with-sysroot=/c/mingw530/i686-530-posix-dwarf-rt_v4-rev0/mingw32 --with-gxx-include-dir=/mingw32/i686-w64-mingw32/include/c++ --enable-shared --enable-static --disable-multilib --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-threads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=release --enable-fully-dynamic-string --enable-version-specific-runtime-libs --disable-sjlj-exceptions --with-dwarf2 --disable-isl-version-check --disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-arch=i686 --with-tune=generic --with-libiconv --with-system-zlib --with-gmp=/c/mingw530/prerequisites/i686-w64-mingw32-static --with-mpfr=/c/mingw530`

MinGW64 (with GCC 8.1.0)	lmx-sdk-*version*#_mingw64_x64.msi	`/prerequisites/i686-w64-mingw32-static --with-mpc=/c/mingw530/prerequisites/i686-w64-mingw32-static --with-isl=/c/mingw530/prerequisites/i686-w64-mingw32-static --with-pkgversion='i686-posix-dwarf-rev0, Built by MinGW-W64 project' --with-bugurl=http://sourceforge.net/projects/mingw-w64 CFLAGS='-O2 -pipe -I/c/mingw530/i686-530-posix-dwarf-rt_v4-rev0/mingw32/opt/include -I/c/mingw530/prerequisites/i686-zlib-static/include -I/c/mingw530/prerequisites/i686-w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -I/c/mingw530/i686-530-posix-dwarf-rt_v4-rev0/mingw32/opt/include -I/c/mingw530/prerequisites/i686-zlib-static/include -I/c/mingw530/prerequisites/i686-w64-mingw32-static/include' CPPFLAGS= LDFLAGS='-pipe -L/c/mingw530/i686-530-posix-dwarf-rt_v4-rev0/mingw32/opt/lib -L/c/mingw530/prerequisites/i686-zlib-static/lib -L/c/mingw530/prerequisites/i686-w64-mingw32-static/lib -Wl,--large-address-aware'` `Thread model: posix` `gcc version 8.1.0 (i686-posix-dwarf-rev0, Built by MinGW-W64 project)`

Note: The platforms supported by LM-X are identified by executables compiled with LM-X rather than by the actual OS. For example, Win32 and Win64 are considered different platforms.

The following table specifies the platforms no longer supported by LM-X License Manager.

Platform	Last Supported Version
CentOS 6	LM-X v5.3.3
CentOS 5	LM-X v4.8.10
Linux (32 bit) ARM	LM-X v4.8.2 (support re-added in v5.5)
Solaris (64 bit) x64	LM-X v4.8.1
Solaris (64 bit) Sparc64	LM-X v4.8.1
AIX PPC64	LM-X v4.7.6
HP-UX (64 bit) IA64	LM-X v4.7.6
Windows 2003	LM-X v4.6.5
Windows XP	LM-X v4.6.5
AIX PPC32	LM-X v4.6
HPUX IA64 ILP32	LM-X v4.6
Solaris (32 bit) x86	LM-X v4.6
Solaris (32 bit) Sparc	LM-X v4.6
Linux IA64	LM-X v4.5.1
FreeBSD (32 bit)	LM-X v4.5.1
Mac OS X (Universal) for 10.6	LM-X v4.5.1
Mac OS X (Universal) for 10.7	LM-X v4.5.1