

# Usage

## How do I prevent loading multiple license files by a single instance of a license server?

To prevent the license server from loading multiple [license files](#), you should embed an ID string into the COMMENT field against each feature in a single license file. This way, all features in one file will then have the same ID, but each license file will have a different id; therefore, if a license file is remade, it will be assigned a new ID.

With LmxServerStartup override function, we will validate all of the license features of the server to make sure they all have the same ID which will prevent a user from using multiple license file with a single instance of the server and also from copying feature lines from one license file to another.

## How are files protected by client store are stored?

Client store is one of the methods used by LM-X to protect your application. Client store uses [secure store](#) by supplying a means of storing user content more securely. The default locations where the files are stored differ depending on an operating system. There is no way of removing all files, but you can delete individual pieces of content stored using LM-X API.

Please note that determining whether a file is corrupted is impossible.

See [Client store](#) for more information.

## LM-X UI and demo do not work with Chrome version 42 and later

As of Chrome version 42, NPAPI is disabled by default. If LM-X online demo or UI do not work with your Chrome version 42 and later browser, you probably need to enable the NPAPI plug-in, as recommended by [Java online help resources](#).

To enable NPAPI:

1. Type "[chrome://flags/#enable-npapi](#)" into your Chrome browser bar.
2. Click the **Enable** link for the **Enable NPAPI** configuration option.
3. Click the **Relaunch** button at the bottom of your configuration page.

## Is it possible to simultaneously run several license servers from different vendors on the same host?

LM-X License Manager is designed to let you simultaneously run multiple instances of different license servers from different vendors on the same host. (Both license servers will share UDP port 6200. )

You should also ensure that each server runs on a unique TCP port.

See [License server issues](#) for more information.

## Where can I download MinGW installers?

You can [compile the LM-X SDK](#) on a Windows machine using MinGW, a minimalist development environment for Microsoft Windows applications.

Click [here](#) to download MinGW for 32 and 64-bit Windows.

## Does LM-X have to be implemented as part of an executable file?

LM-X License Manager is not an executable wrapper, and therefore has to be included into the source code prior to compilation.

## Does LM-X offer protection against reverse engineering and tampering with executable content?

LM-X License Manager does not provide executable code protection against reverse engineering and accidental or deliberate tampering with the executable content. If you want to use software that provides protection from reverse engineering, you will need to use a third party solution.

Please note that LM-X License Manager allows you to perform a clock check on the client side to ensure users have not backdated or tampered with their system clock.

See [System clock check](#) for more information about protection against time tampering.

## How to use LmxLicenseProvider as an implementation of the abstract base class provided by .NET LicenseProvider?

The default implementations of .NET LicenseProvider for your licensing model offer only basic features that are unlikely to satisfy today's needs for security and flexibility. This is where LmxLicenseProvider can help you to make your licensing model more robust.

For more information about how LmxLicenseProvider can boost LM-X License Manager's capabilities in your .NET licensing model, see [Using LmxLicenseProvider as an implementation of the abstract base class provided by .NET LicenseProvider](#).

## Does LM-X License Manager support vendor defined strings or custom data fields?

LM-X supports vendor-defined strings and custom data fields, and makes it possible for you to extend LM-X with additional information in your licenses.

You can set the DATA field in the license to specify a comment, such as additional licensing options or restrictions, as described in [FEATURE settings](#).

## Do I need multiple connections to the license server (LMX\_HANDLE)?

You need only one connection to the license server (LMX\_HANDLE), which is done when you call LMX\_Init. You should call LMX\_Init only once, and continue to use the single handle throughout the lifetime of your application.

See [LMX\\_Init](#) for more information.

## Why does LM-X report first-chance exception in Visual Studio?

See [Installation issues](#) for more information about this problem and steps to resolve it.

## What Linux compilers does LM-X License Manager support?

For Linux (x86 and x64), LM-X supports the use of GCC (GNU Compiler Collection). Other compilers that use output compatible with GCC, including custom compilers, as well as Clang, will also work but are not officially supported.

## Do you support internet activation in LM-X License Manager?

By using our [License Distribution Service](#) you can enable activations directly from within your application. This means you can allow purchasing users to process their orders and let evaluating users obtain an evaluation license with no additional work on your part.

## Should I lock my license to only one hostid or use multiple hostids?

Locking your licensed application to an Ethernet card, BIOS or harddisk HostID provides enough security to meet most needs. If you require more security, you may lock to multiple HostIDs using one of methods described in [Deciding whether to lock a license to one or multiple HostIDs](#).

## Why do I get the error "Unable to use license server" when I run a .NET application on a 64-bit machine?

When you have a 32-bit .NET application protected with a 32-bit LM-X library (lmxnet.dll), the application will run successfully on a 32-bit machine, but if the application is run on a 64-bit machine the following error might occur:

```
"Unable to use license server."
```

The problem is that .NET attempts to run the 64-bit version of the .NET framework when the application is attempted to be used on a 64-bit machine.

To resolve this problem, do one of the following:

- If it is important to provide 64-bit support, build your application with a 64-bit version of the lmxnet.dll.
- If providing 64-bit support is not necessary, force your application to be 32-bit by using the switch `/platform:x86` when compiling your code. This solution requires virtually no work and will solve the problem.

## Can end users administer and monitor their licenses?

With LM-X License Manager, software vendors can provide end users with a license file they can use as a standalone license or with a license server. Our license monitoring software, [License Statistics](#), lets endusers monitor the use of floating or standalone licenses.

## We use custom hardware and can access the serial number information from this equipment. Is there a way to lock LM-X licenses to this hardware device?

You can lock LM-X to your hardware device by specifying a custom hostid and using the LMX\_HOSTID\_CUSTOM hostid type together with LMX\_Hostid or LMX\_HostidSimple to retrieve the LM-X hostid. For this to work, you must write a custom hostid callback function as shown in the hostid example that comes with LM-X, and then register this function using:

```
LMX_SetOption(LmxHandle, LMX_OPT_CUSTOM_HOSTID_FUNCTION, (LMX_OPTION) MyHostid);
```

After registering the custom callback function, LMX\_Hostid and LMX\_HostidSimple will call the function and retrieve the LM-X hostid. Note that the LM-X Config Tool and LM-X End-user Utility will not print out custom hostids. To let users send back custom hostids, you must make your own hostid utility and send it to users.

## How do I renew my software?

There is no need to contact us regarding your software renewal. When your X-Formation software product is nearing expiration, we'll send you an automatically generated email with a Proforma invoice for license renewal.

- For 60-day payment terms, the renewal invoice will be sent to you 90 days prior to the software expiration date.
- For 30-day payment terms, the renewal invoice will be sent to you 60 days prior to the software expiration date.

If you require your Purchase Order number to be included in the invoice, you can add it using the form at <http://www.x-formation.com/sa/set-po-number.html>. You will need your activation key and quote/Proforma number to complete this form. For your convenience, your emailed quote includes a personalized link that automatically fills in your activation key and quote/Proforma number on this form.

Please note that X-Formation's invoicing system is completely automated; therefore, if you require a different method of invoicing other than the automatic Proforma invoice or require a printed invoice, a fee of 50 EUR will be charged for administrating the invoice.

## Why should I renew my software maintenance?

There are many important benefits to keeping your X-Formation software maintenance up to date.

All perpetual licenses include 1 year of updates and support (any licenses purchased after the original purchase are added to your current maintenance plan). After the first year, maintenance is available for just 25% of the current, regular (non-discounted) price. This is a significant savings over allowing maintenance to lapse and then repurchasing the software at a later date.

Renewing your software maintenance gives you continued access to the best product support in the industry, which includes:

- Prompt responses to your technical problems by phone or email.
- Participation in [Customer Driven Development](#).
- Access to your existing license files using [License Activation Center](#).

Most importantly, you'll continue to have uninterrupted access to new versions of the software as they're released. We're constantly improving our products, based directly on feedback from customers like you, so you won't want to miss out on our upcoming releases. (Note that your maintenance must be up to date to purchase any add-on software product.)

## LM-X UI and demo do not work with Chrome version 42 and later

As of Chrome version 42, NPAPI is disabled by default. If LM-X online demo or UI do not work with your Chrome version 42 and later browser, you probably need to enable the NPAPI plug-in, as recommended by [Java online help resources](#).

To enable NPAPI:

1. Type "[chrome://flags/#enable-npapi](#)" into your Chrome browser bar.
2. Click the **Enable** link for the **Enable NPAPI** configuration option.
3. Click the **Relaunch** button at the bottom of your configuration page.

## We use custom hardware and can access the serial number information from this equipment. Is there a way to lock LM-X licenses to this hardware device?

You can lock LM-X to your hardware device by specifying a custom hostid and using the LMX\_HOSTID\_CUSTOM hostid type together with LMX\_Hostid or LMX\_HostidSimple to retrieve the LM-X hostid. For this to work, you must write a custom hostid callback function as shown in the hostid example that comes with LM-X, and then register this function using:

```
LMX_SetOption(LmxHandle, LMX_OPT_CUSTOM_HOSTID_FUNCTION, (LMX_OPTION) MyHostid);
```

After registering the custom callback function, LMX\_Hostid and LMX\_HostidSimple will call the function and retrieve the LM-X hostid. Note that the LM-X Config Tool and LM-X End-user Utility will not print out custom hostids. To let users send back custom hostids, you must make your own hostid utility and send it to users.