# Using LmxLicenseProvider as an implementation of the abstract base class provided by .NET LicenseProvider

If you've ever used .NET LicenseProvider for your licensing model, you probably know that the default implementations of this abstract class 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.

LmxLicenseProvider lets you use LM-X License Manager's capabilities in your .NET licensing model, providing solutions for existing or home-made implementations, such as automatic server discovery, license borrowing, redundant server capabilities, and many more LM-X features. In addition, the LM-X .NET API offers extensive functionality that you may find useful for modifying the behavior of LmxLicenseProvider.

The easiest and fastest way to integrate LmxLicenseProvider into your application is to compare the provided sample code, lmxtest_licenseprovider.cs, located under the dotnet examples directory in your LM-X distribution, with a clear example.

One such example, http://msdn.microsoft.com/en-us/library/system.componentmodel.licfilelicenseprovider.aspx,  shows you how to use LicFileLicenseProvider, one of the default implementations of LicenseProvider.

The blue lines in the example below show the changes you would make to LicFileLicenseProvider to integrate LmxLicenseProvider into your application. As the example shows, the basic steps for enabling the licensing schema for your component or control class are:

1. Apply a LicenseProviderAttribute to the class.
2. Add a License field to the class.
3. Validate the license in the constructor.
4. Dispose of the granted license in the finalizer of the class or before it is called.

```
using System;
using XFormation;
using System.ComponentModel;
using System.Windows.Forms;


// Adds the LicenseProviderAttribute to the control.
[LicenseProvider(typeof(LmxLicenseProvider))]


public class MyControl : Control {
  // Creates a new, null license.
  private License license = null;
  private string strFeatureName = "feature";
  private int nMajorVer = 1;
  private int nMinorVer = 0;
  private int nCount = 1;
  public MyControl () {
    // Adds Validate to the control's constructor.
    if (nMajorVer >= 0 && nMinorVer >= 0 && nCount >= 0 && strFeatureName.Length > 0)
      license = LicenseManager.Validate(typeof(MyControl), this);
    else throw new LicenseException(typeof(MyControl), this, "Fields are not properly set.");
    // Insert code to perform other instance creation tasks here.
  }
  protected override void Dispose(bool disposing) {
    if (disposing) {
      if (license != null) {
        license.Dispose();
        license = null;
      }
    }
  }
}
```

As you can see, modifying a LicenseProvider base class to use LmxLicenseProvider requires only a few minor changes and new lines of code, making it quick and easy for you to realize the significant benefits LM-X can bring to your .NET licensing model.