

Published: November 2007  
For the latest information, please  
see <http://www.microsoft.com/slps>

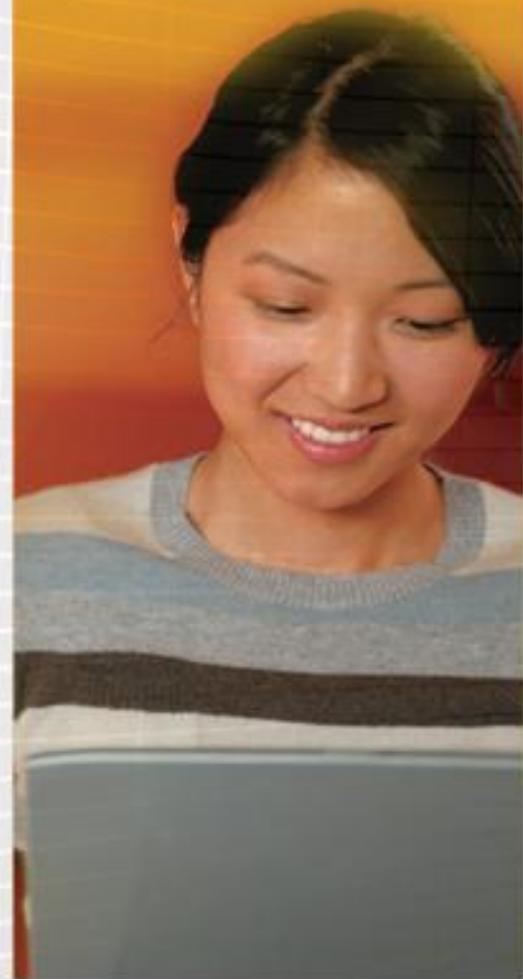
**Microsoft®**



**Microsoft® Software Licensing and Protection (SLP) Services**

# **Technology Overview**

White Paper



*The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.*

*This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.*

*Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.*

*Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

*© 2007 Microsoft Corporation. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.*

*The names of actual companies and products mentioned herein may be the trademarks of their respective owners. Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA*

## Contents

A Comprehensive Solution.....	1
Protect Intellectual Property and Revenues.....	1
Innovative Packaging and Licensing.....	2
Manage Customers and Assets.....	2
Code Protection .....	3
.NET Common Language Infrastructure.....	3
Traditional Approaches to Code Protection .....	4
Obfuscation.....	5
Encryption .....	5
Code Splitting.....	5
SLP Services Approach to Code Protection.....	6
Transformations and Unique Permutations .....	6
Innovative Software Packaging and Licensing .....	9
Feature-level Control .....	10
Flexible Licensing Options.....	10
Dates .....	11
Usage.....	11
Activations.....	12
Machine .....	12
Licensing Monitoring.....	12
Licensing Lifecycle.....	12
Define your Products and Features .....	13
Protect your Code and Bind to Features.....	13
Create and Issue Licenses .....	15
Product Activation .....	16

Post-activation ..... 17

Ease Your Burden ..... 17

For More Information ..... 18

## A Comprehensive Solution

Independent software vendors (ISVs) and corporate development (CorpDev) teams share similar challenges: Both want to control who is using their software and how it is being used, while at the same time reducing the cost of doing business.

Each organization also has their own unique problems:

- ISVs are faced with the need to fight piracy, protect against reverse engineering, enforce software licensing, and reduce time-to-market for new SKUs.
- CorpDev teams need to protect internal applications that contain trade secrets or information about corporate infrastructure, and monitor usage of applications within the enterprise.

Microsoft® Software Licensing and Protection (SLP) Services offers a comprehensive solution for any organization interested in protecting their intellectual property (IP) and software revenues, innovating in the area of packaging and licensing, and managing customers and assets.

## Protect Intellectual Property and Revenues

Software applications often contain sensitive information—trade secrets, algorithms, passwords, or other intellectual property you want to protect and control access to. Applications built on the Microsoft® .NET framework derive a lot of power from the common infrastructure that underlies that technology, but that commonality makes them particularly susceptible to reverse engineering, making it more important than ever to find ways to protect that information.

Through the use of the SLP Code Protector application and a SLP Permutation, compiled .NET code—which is easy to reverse engineer—is transformed into Secure Virtual Machine Language (SVML)—which is not. Each Permutation includes a unique transformation into a unique version of the SVML. Also included in the Permutation is the redistributable Secure Virtual Machine, which executes that unique version of SVML. Together, they provide superior code protection for applications built on the .NET Framework.

However, simply protecting intellectual property is only part of the problem. The *Fourth Annual BSA and IDC Global Software Piracy Report* estimates that \$40 billion is lost each year to piracy and license noncompliance. ISVs can lose up to half their realized revenue to piracy. CorpDev teams face similar problems, restricting access to sensitive applications and tracking usage statistics. The need for software

protection has never been greater and—either as standalone protection, or as part of a holistic licensing, packaging, and activation strategy—Microsoft SLP Services has you covered.

## Innovative Packaging and Licensing

More and more, software consumers are demanding new ways of buying software. They want to evaluate software before purchasing it, they want the convenience of online purchase, and the success of software as a service (SaaS) and pay-per-use models demonstrates that software customers want to pay for the specific functionality they use. The challenge of responding to these requests can prove difficult even for large ISVs. These issues are made even more difficult by the challenge of balancing this new flexibility with the increased need to secure software against piracy and license noncompliance.

Microsoft SLP Services is changing the rules, making licensing part of the business logic that enforces how a software package can be used. Using the SLP Licensing Control Portal (through either the SLP Server 2008 or hosted SLP Online Service), software publishers can create “SKUs” with specific feature sets and time- and usage-based access rules by defining different licenses. Thus, demo versions of software can be created, “basic” and “full” versions of software defined, and—with the addition of the ability to collect information about how your application is used—pay-for-use business models enabled.

All of this is accomplished without code changes or any involvement by the development team. We call this *SKU Agility*.

## Manage Customers and Assets

Distributing software is complex enough without the added complications of license management. In today’s software marketplace, it is licenses of software that are sold, not boxes of software. However, until now, there has not been an integrated system to help manage licensing. The process of managing, upgrading, and renewing licenses often involves several distinct systems that are not well integrated, and customers typically do not have access to information about their licenses. This adds up to increased cost of business and decreased user satisfaction.

**Whether you work for an ISV or a corporate development team, SLP Services can help you deliver what your users want, when they want it, and how they want it—while reducing costs and protecting your investment.**

SLP Services separates licensing from the application and provides infrastructure for full lifecycle license management, and then goes further by providing a robust infrastructure for managing application activation. SLP Activations can be issued through the SLP Server 2008 or through the hosted SLP Online Service. License enforcement is managed by the Secure Virtual Machine (SVM)—part of the SLP Permutation and embedded in your application. The SVM controls access to specified functionality according to rules within the license.

With the tools to create new SKUs and experiment with new business models in the hands of the people who interact directly with your customers, SLP Services can help you reduce time-to-market, increase customer satisfaction, and maximize revenues.

Going forward, as part of Microsoft's SLP Services roadmap, future versions of the an application programmer interface (API) will provide rich services to enable integration with Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), and billing systems—helping to enable tight integration of license and usage information into the full enterprise.

## Code Protection

The SLP Services code protection technology is a powerful tool to help fight against piracy, intellectual property theft, and license noncompliance.

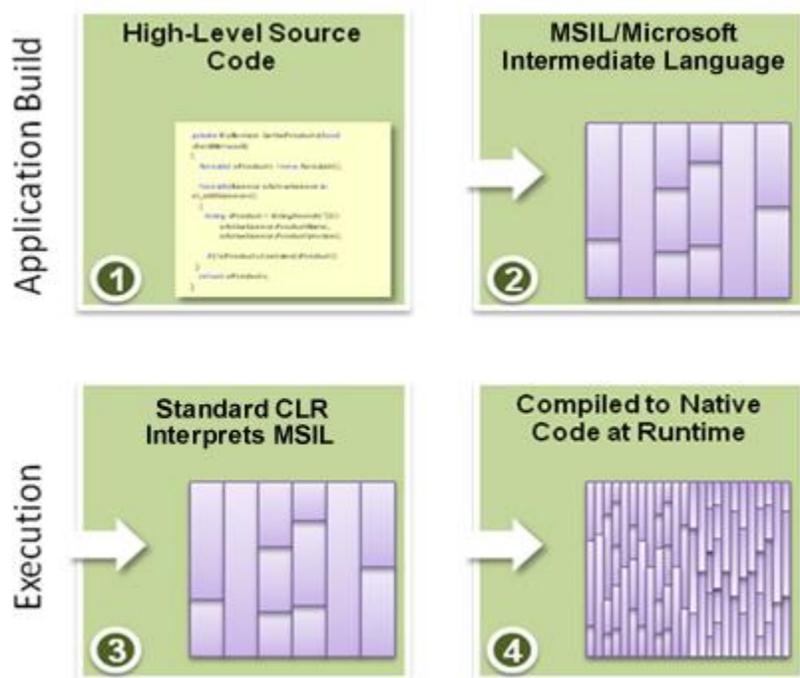
## .NET Common Language Infrastructure

Microsoft designed the .NET Framework with many goals in mind. One of these was to provide platform independence for Microsoft® Windows® applications, and to provide a common infrastructure against which multiple languages could easily interact. Key to that common infrastructure is the Common Language Runtime, which is platform independent and runs above the computer's processor. Acting as a "virtual processor," it executes a platform-independent instruction set known as the Microsoft Intermediate Language (MSIL). See Figure 1.

All .NET code is compiled down into MSIL, and it the MSIL which runs against the CLR. The CLR, in turn, converts the instructions into platform-specific native code on demand, a technique referred to as Just-In-Time (JIT) compilation.

Although this technique has distinct advantages, the nature of the MSIL is such that it carries a lot of the structure and information from the original high-level language. The end result is that the Microsoft Intermediate Language is easily decompiled or reverse engineered—converted back into a high-level language—using easily-available “shareware” tools.

**Figure 1 Traditional process of compiling and running .NET application code**



Once decompiled, any trade secrets within the code are easily readable. Also exposed is information about your company’s infrastructure, like database connection strings and passwords. This weakness has prevented some organizations from making the switch to .NET as a development platform.

SLP Services has a solution to this problem, enabling ISVs and CorpDev teams to help protect sections of their .NET code that contain sensitive information.

### Traditional Approaches to Code Protection

Before looking at SLP Services’ innovative new approach to code protection, we will take a look at conventional techniques for code protection: *obfuscation*, *encryption*, and *code splitting*. These methods each have their strengths, but also have weaknesses which can render them ineffective or undesirable.

## Obfuscation

Recognizing that compiled .NET code can be easily rendered back into a high-level language, software was developed to alter the MSIL so that, upon reverse engineering, the resulting high-level code would be *obfuscated*—simply made too difficult to read for most people.

Most obfuscation programs perform two tasks:

- Change the meaningful names of classes, methods, parameters, and variables into meaningless text.
- Change the flow of the code so that it has the same end result, but is much more difficult to read.

Obfuscation, though effective at slowing down hackers, possesses inherent weaknesses. First, obfuscation does not prevent decompiling and reverse engineering the code—it does not change the structure or the logic of the code, it only makes the code more difficult to understand. Second, defects and other hard-to-track problems can be introduced along with the renaming and flow control changes.

Obfuscation is a good start, but it does not go far enough as a standalone solution to the problem of reverse engineering.

## Encryption

Some attempts at protecting code involve taking the code and encrypting it. There are several tools available to encrypt Microsoft .NET assemblies, each one starts by using an encryption algorithm to encode the .NET modules, which are then decrypted at runtime. The problem here is that the key is delivered along with the lockbox. In order to execute the encrypted MSIL, it must be decrypted before the Common Language Runtime (CLR) can act on it. This leaves an opening for a hacker, who can use automated tools to recover the key and then use the decryption engine to decrypt the code.

## Code Splitting

Code splitting is another approach to help protect application code from reverse engineering and typically works by breaking the application code into two pieces. The less-sensitive portion of the code is

**Obfuscation is a good start, but it does not prevent reverse engineering, just makes it more difficult.**

**Encryption leaves an opening for hackers.**

**Code splitting is stronger still, but it can be costly and inefficient.**

delivered in its usual form; the more sensitive piece of code is delivered on special external hardware, such as a smart card or a hardware dongle (that is, a security key).

This method has proven effective and offers a higher level of protection relative to other solutions on the market, but it also has several disadvantages. An external device carries additional costs and can be cumbersome for users. In addition, the software publisher must manually convert code to the format of the specific device. Pushing out software patches and updates becomes complicated, and online distribution might not be possible. Finally, the code is physically deployed to the user, and though it would require greater skill, it could still be reverse engineered.

Although code splitting is a highly reliable solution, these limitations can make it costly and inefficient.

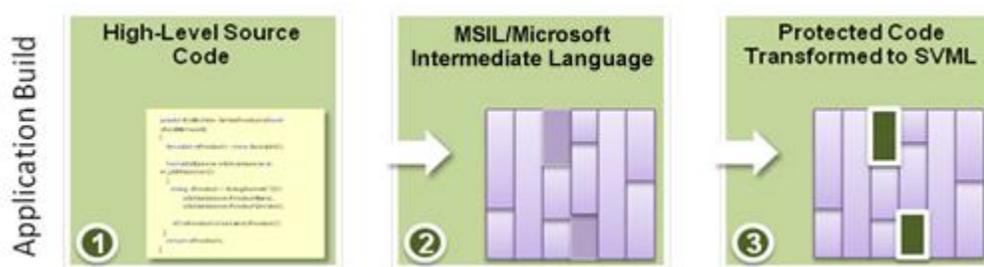
### **SLP Services Approach to Code Protection**

Each of the three traditional methods for protecting code has strengths and weaknesses, and the technique used by SLP Services takes the best from each method to provide a more effective, convenient software-only solution.

The fundamental strategy used by SLP Services is to take sensitive code and pass it through a one-way transformation that leaves the original logic effectively encrypted *and* obfuscated, then split it away from direct execution on the CLR.

### **Transformations and Unique Permutations**

Key to the SLP Services code protection strategy is code transformation. Using the SLP Code Protector application, compiled MSIL is transformed into Secure Virtual Machine Language (SVML) which can no longer be run directly by the CLR or reverse engineered by decompilation tools, and then encrypted so that the resulting SVML “byte code” cannot be directly inspected. See Figure 2.

**Figure 2 SLP Services code protection**

Decompiling of normal Microsoft .NET code is possible because MSIL is a fixed and widely known specification. The SVML command set, on the other hand, changes with each unique SLP Permutation. Because each version of SVML is different, each also requires a unique Secure Virtual Machine (SVM) to execute the SVML. This combination of unique Secure Virtual Machine Language and a unique Secure Virtual Machine to execute it on comprises a SLP Permutation.

If SVML were simply another “byte code” made up of instructions to be executed, a very skilled hacker could, theoretically, spend the time to reverse engineer the entire command set and use this knowledge to build a decompiler; however, because the SVML is actually encrypted, there is a huge barrier to even that already complex task. Further, because each SVM is different, any decompiler would only be effective against a specific SVM associated with that specific permutation. Different applications from the same vendor could use different permutations to ensure that any compromise of security is limited to a single application. This layered approach makes SLP Services a premier code protection technology.

This ability to create an encrypted, unique, and transformed version of your code is a powerful protection against reverse engineering, and simple to achieve. After building your .NET application, use SLP Code Protector to identify the methods you want to protect. Because there is a performance cost to decrypting and executing protected code, it is best to transform only those methods that contain sensitive intellectual property, information about your enterprise infrastructure, passwords, etc. After protection is complete, the protected code—when viewed in a .NET decompiler—shows only a call into the SVM with a string of random characters.

When your application is executed, the CLR handles the original unprotected MSIL, but the SVM executes the protected SVML. The SVM links directly into your .NET application and works as an integral part of your application logic, thus making it more difficult to bypass licensing routines.

**Source Code Before Transformation:**

```
private static OleDbConnection GetDBConnection()
{
    if ( _dbConnection == null )
    {
        string connectionString= "Provider=Microsoft.Jet.OLEDB.4.0;" +
            "DataSource=MedicalImage.mdb;" +
            "Password=medical";

        _dbConnection=newOleDbConnection(connectionString);
        _dbConnection.Open();
    }

    return _dbConnection;
}
```

**After Transformation:**

```
/* Notice that the method signature remains the same, but the
   logic is replaced by a call into the Secure Virtual Machine */
private static OleDbConnection GetDBConnection()
{
    object[] args = new object[0];

    object obj2 = SLMRuntime.ExecMethod( null,
        "DS9FvZGluZyAvV2luQW5zaUVuY29kaW5A0vQmXZ",
        args );

    return (OLEDBConnection) obj2;
}
```

To recap, SLP Services code protection offers many unique and compelling benefits:

- The logic of protected methods is removed and replaced with a call into the SVM, so traditional .NET decompilers cannot reverse engineer protected code.
- The SVM is encrypted to make reverse engineering the new “byte code” even more difficult.
- Because each Permutation is unique, any reverse engineering that was accomplished would be limited to that single Permutation.
- The original code never exists fully in memory.
- Because licensing is based on the process of executing protected functions, the SVM is an integral part of enforcing your licenses.

**Customers are becoming increasingly sophisticated as to how they want to evaluate, purchase, use, and upgrade software packages. Gone are the days of “one size fits all/one price fits all” software sales.**

## Innovative Software Packaging and Licensing

Customers are becoming increasingly sophisticated as to how they want to evaluate, purchase, use, and upgrade software packages. Gone are the days of “one size fits all/one price fits all” software sales. Instead, customers want the ability to try software before they buy it. They want the ability to start small with a “basic” package and step up to the “full” SKU when they need. Business models like software as a service and pay-per-use are showing that customers want to pay for the actual value they are receiving from their software.

Software publishers are feeling the pressure to offer their products to customers in myriad different ways, but the cost and effort to address these requests can be challenging for even large ISVs and corporate development teams.

SLP Services offers a suite of tools to help meet these challenges head-on. Through a combination of feature-level control, flexible licensing options, and license monitoring, SLP Services offers ISVs a rich tool set which allows you to:

- Expand your product offerings easily by enabling specific feature sets for different SKUs without modifying or creating additional code (SKU Agility).
- Personalize features and licensing terms for a single customer.
- Help increase revenue through SaaS, pay-as-you-go, and subscription business models.
- Offer better value to your customers by letting them purchase just the features they need, while maintaining up-sell opportunities and an easy upgrade path.
- Deliver your products electronically, offer on-demand purchasing, and allow customers to share installation media.

Corporate development teams can also realize many benefits. Using SLP Services, they can:

- Provide secure access to features by making them available only to certain computers.
- Help reduce development overhead by enabling single-SKU deployment, enabling specific features for specific users.
- Keep sensitive software in-house by restricting a license to a specific computer.

**SLP Services lets you ship a single version of your application, and let the licenses contain specific rules about how each different installation of the software can be used.**

## Feature-level Control

The license enforcement in SLP Services is carried out as a function of the SVM. On the SLP Licensing Control Portal, features are defined and then mapped against specific methods in the code. When the application is run, before it allows the protected code to run, the SVM first checks to make sure that the associated feature is enabled by the license. Different licenses can enable different sets of features, or specify different rules for how many times or for how long they can be used, all under the control of the SVM.

This flexibility is at the heart of one of the most innovative concepts in SLP Services: the concept of “license as business logic.” As opposed to traditional licensing, SLP Services lets you ship a single version of your application, and let the licenses contain specific rules about how each different installation of the software can be used—and have those rules enforced securely from within the SVM. In addition, if it is requested as part of the license, the SVM will help keep a count of how many times a specific feature has been used.

Though all of this functionality is available with no developer support, developers can use the SLP Services Code Protector SDK for even more flexible and precise licensing control from within the application code.

## Flexible Licensing Options

The SLP Services design allows for a variety of sophisticated licensing options, and they can be combined together as building blocks to deliver the licensing model that works best for your customers. For instance, you could:

- Offer a demo version of your application which works for a set time period.
- Offer a free “basic” version of your software which allows customers to try advanced features a limited number of times, and provides a seamless upgrade path if they purchase a full license.
- Offer a subscription business model, where the application license is renewed quarterly.

The flexibility in licensing options permits real agility in how ISVs can explore new business models and respond to specific customer requests. Because it can all be done without code changes, this can be a nimble process, and, in turn, contributes to an overall sense of increased value on the part of customers.

As seen in Figure 3, there are several options in defining your licenses, all of which you can combine.

Figure 3. Setting license options

## Dates

You can give licenses either a “lifetime” of a specific number of days, or explicitly set start and expiration dates. You can also mark any license as renewable; after the license has expired, the application will automatically try to connect to the server and renew its license. To make reactivation more convenient for the customer, you can establish a grace period that allows a license to remain active for a set number of days after its expiration.

## Usage

Licenses can be configured to permit a maximum number of uses. These limits can be set independently for each feature, or can be marked as “unlimited.”

## Activations

You can allow automatic activation over the Internet or manual activation for offline computers. Plus, you can limit the number of times a customer can activate a single license. Customers can also, if you allow it, deactivate licenses on a particular computer and activate the license on a different machine.

## Machine

You are able to control where the software runs by limiting a license to a single computer, and specify if the license can be used on a virtual computer.

## Licensing Monitoring

As part of the licensing enforcement process, you can specify that the licensed application maintain a count of how many times a customer has accessed a protected feature and to transmit this information back to an SLP Server 2008 installation or the SLP Online Service.

This information can be useful in a variety of ways. Most obviously, it provides real-world business intelligence about how your application is being used by your customers, allowing you to determine which features are most- and least-commonly accessed and use that information to direct future development. It also enables the opportunity for new business models based on the actual usage of the product, providing true utility- or consumption-based pricing models.

In a corporate environment, monitoring application usage can be very valuable. In addition to tracking feature use to guide development, you can employ usage statistics to build return on investment metrics and justify investment in additional features. Consequently, building, deploying, and monitoring software in a corporate environment becomes an interactive process with the user. Finally, the ability to control and track feature access can play an important role in enterprise security.

## Licensing Lifecycle

The SLP Services suite of tools is used throughout the software development and business lifecycles to offer superior code protection and flexible product packaging and licensing.

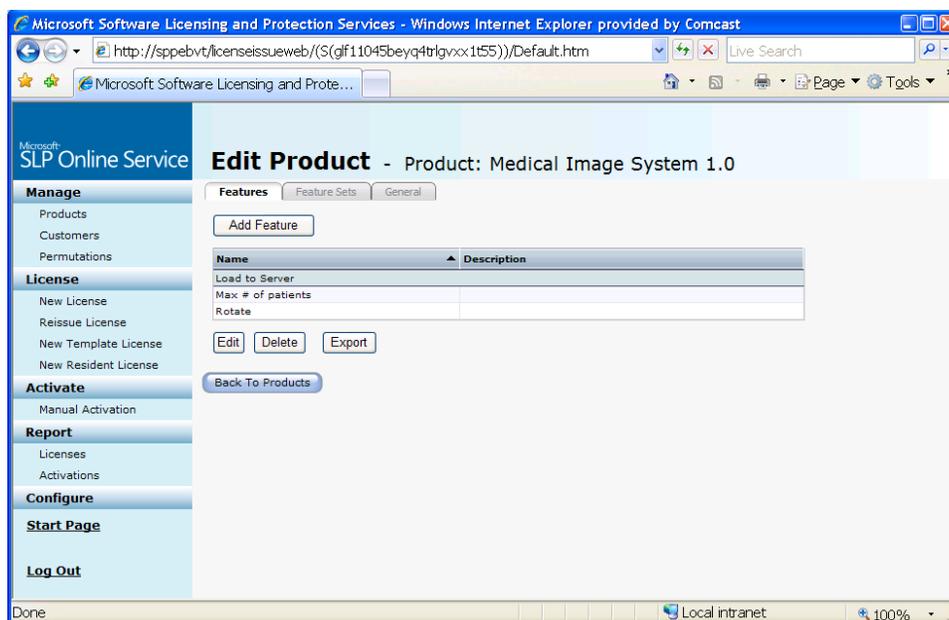
**A licensed application can keep count of how many times a protected feature is used, to facilitate the collection of usage data to guide future development, and enable the development of utility- and consumption-based pricing models.**

## Define your Products and Features

Using the SLP Licensing Control Portal deployed locally using SLP Server 2008 or through the hosted SLP Online Service, it is a simple task to define the products you want to offer and the features within those products. In general, you can define features at the level that you want to control them. This could be a fairly precise definition like “Custom Data Transformation” or a broader set of functionality like “File Management” See Figure 4.

These features can be grouped together into feature sets, which form the basis of the “virtual SKUs” and will be licensed. You could, for instance, define a “demo” feature set that includes all of the features of your product except saving files; a pair of “basic” and “full” feature sets that contain a portion and all of the features respectively or “home” and “enterprise” feature sets, where the enterprise feature set includes features for syncing with an external data source.

**Figure 4 Edit Product window**



## Protect your Code and Bind to Features

Earlier we looked at using the SLP Code Protector application to transform specific portions of .NET code into Secure Virtual Machine Language. As part of licensing, we will use the same basic process, with two additional steps.

In addition to selecting the methods in the code to protect:

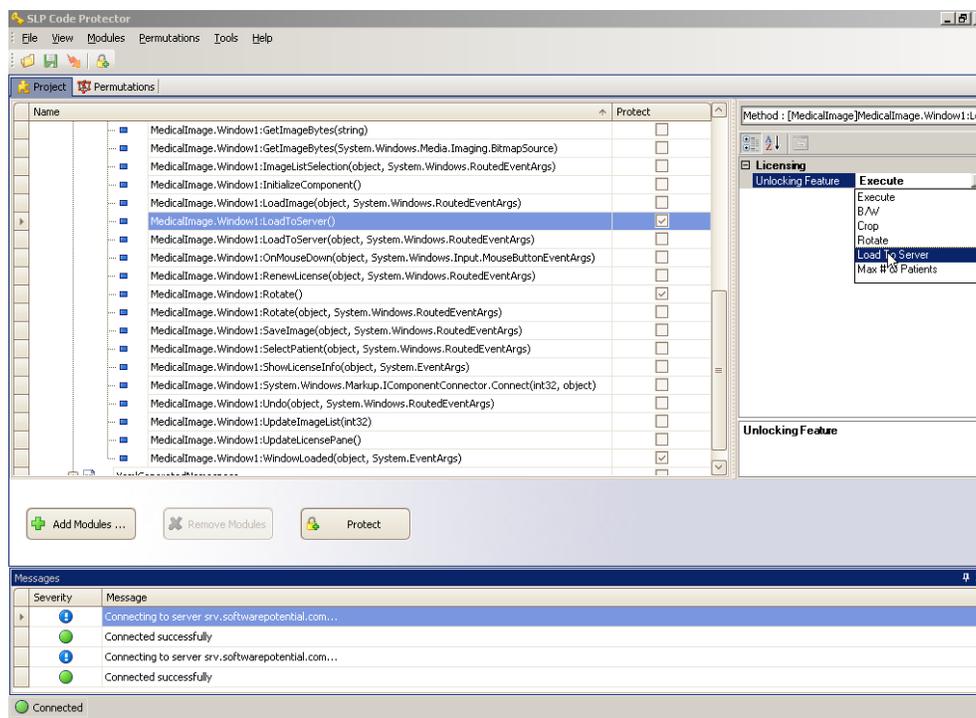
1. Sync the Code Protector application with the SLP Licensing Control Portal so that it knows about your products and features you defined in the previous step.
2. Bind sections of code to the features.

The binding of feature to code is central to the SLP Services licensing enforcement technique. The SVM is responsible for not only executing the protected code, but for first checking that the license-defined “business logic” permits the protected code to be run in the first place.

If the license indicates that a specific feature can be executed, then the SVM will do so; if the license does not permit feature execution, the related piece of protected code will not be run. This is why we sometimes speak of protecting “staging functions.” These are functions that must execute in order for specific pieces of functionality to work.

**The binding of feature to code is central to the SLP Services licensing enforcement. The SVM will check that the license permits a feature to be used before executing the specific piece of protected code associated with that feature.**

**Figure 5 Binding a feature to code**



For instance, in the case of a .NET WinForms application, the process of showing a dialog box involves executing the `InitializeComponent()` method on the associated class. If that method fails to run,

the dialog box will not launch. It is a “staging function” for that functionality, and makes an excellent candidate for protection if you wanted to control access to that feature through the license (Figure 5).

## Create and Issue Licenses

The SLP Licensing Control Portal has a powerful set of tools for defining and issuing licenses with a broad set of options. Starting by selecting a specific feature set, and if the license is “commercial” or “non-commercial,” the user can then set a number of options giving precise control over how the features can be used, if statistics about their usage is reported, if there are limits on usage or how the license can be renewed (Figure 6).

**Figure 6 Creating a New License**

Once the license has been defined, individual licenses and activation keys can be generated. The activation key will be sent to the user when the product is purchased or when a new license for an existing installation is requested.

Building new licenses is quick and simple, requiring no effort on the part of the development team. This enables sales and marketing to work directly with the customer to deliver targeted solutions.

## Product Activation

When the software is distributed, it will include the activation key prepared in the previous step. Upon installation, the user will activate the product by entering the activation key into a license activation dialog box, such as the dialog box in Figure 7.

The look and feel of the license activation dialog box can be customized by the software publisher and can include additional features, such as manual license installation or requesting a license online.

**Figure 7 License activation**



The user inputs their activation key. The application connects to the SLP Licensing Control Portal to verify the activation, and the associated license is digitally signed and sent back. If the computer does not have Internet access, the user has the ability to manually install a license file. The license is then bound with the application to act as a decryption key for protected code, and to be used as business rules to control how the application can be used.

## Post-activation

After a user activates the license, you have full control over that license. Using the SLP Licensing Control Portal, you have the ability to turn features on or off, or extend the license. This gives you the opportunity to extend your customer engagement and respond directly to customer requests.

Software activation through SLP Services provides a simple, secure, and positive experience for the user—and provides total visibility and control for you.

## Ease Your Burden

All software organizations face challenges protecting their IP from piracy or corporate espionage, and the ability to easily reverse engineer .NET applications is a big risk for ISVs and corporate development teams alike. Furthermore, those in the business of selling software require a way to help enforce license compliance and protect revenue. Traditional code protection technologies are unreliable, carry a heavy cost, or limit how you can sell and deploy your software.

At the same time, customers are demanding new ways of evaluating, purchasing, and using software. You need a way to respond to customer needs with agile, capable licensing models, and the ability to explore new business models. Traditional license management strategies put a heavy burden on the development team by making licensing a product feature.

**Table 1 Software Licensing and Protection Services Product Family**

Family	Product	Functionality
Code Protection	Code Protector	Transforms .NET MSIL into a protected, encrypted SVMIL and binds code to product features.
Code Protection and Licensing	Permutation	Provides a transformation of MSIL into a unique Secure Virtual Machine Language. Includes a redistributable Secure Virtual Machine to be embedded into protected applications. Allows a digital signature for secure licensing.
Packaging and Licensing	SLP Service 2008 (deployed locally) SLP Online Service (hosted)	Enables product and feature definition, SKU creation, license issuance, and activation response.
Licensing	Activation Packs	Packs of licenses (commercial or non-commercial) to be issued to users to activate the software.

Microsoft Software Licensing and Protection Services eases these burdens by providing easy-to-implement code protection, and by putting licensing control in the hands of the people who need it most—the people who interact with your customers. See Table 1 for details about the functionality Microsoft offers within the Software Licensing and Protection Services product family.

SLP Services helps provide:

- Positive control over your company’s intellectual property.
- The ability to respond to your customer needs with dynamic licensing.
- Increased efficiency by streamlining business processes.

By managing the entire software business lifecycle, SLP Services can help you more efficiently protect, package, license, and control the use of your software.

Whether you work for an ISV or a corporate development team, SLP Services can help you deliver what your customers and users want, when they want it, and how they want it—while reducing costs and protecting your investment.

**Whether you work for an ISV or a corporate development team, SLP Services can help you deliver what your users want, when they want it, and how they want it—while reducing costs and protecting your investment.**

## For More Information

Please refer to the following resources for more information on the topics covered in this document and for related topics:

Microsoft Software Licensing and Protection Services Web site: [www.microsoft.com/slps](http://www.microsoft.com/slps).

For specific questions, contact us at [slpsinfo@microsoft.com](mailto:slpsinfo@microsoft.com).