



BUSINESS GUIDE



---

## VeriSign® Microsoft® Office/Visual Basic for Applications (VBA) Code Signing Digital Certificates

Realizing the Possibilities of Internet Software  
Distribution



Where it all comes together.™



**CONTENTS**

- + What Is Developer Code Signing? 3
- + Who Needs a Code Signing Digital Certificate? 3
- + What Does Office/VBA Macro Signing Look Like to Consumers? 4
- + Technical Overview: (Optional Reading) 4
  - What Is a Digital Certificate? 4
  - CAs 5
  - How Does Office/VBA Work with VeriSign Digital Certificates? 5
  - Timestamping 6
  - The Six Steps to Signing Code 6
  - How Do You Require a Password before the Application Reuses Your Private Key? 8
- + Conclusion 8



## What Is Developer Code Signing?

---

When customers buy software in a store, the source of that software is obvious. Customers can tell who published the software, and they can see whether the package has been opened. These factors enable customers to make decisions about which software to purchase and how much to trust that software. When customers download software from the Internet, the most they see is a message warning them about the dangers of using the software. The Internet lacks the subtle information provided by packaging, shelf space, shrink wrap, and the like. Without an assurance of the software's integrity, and without knowing who published the software, it is difficult for customers to know how much to trust software. It is difficult to choose whether to download the software from the Internet.

The solution to these issues is a VeriSign® Digital Certificate. VeriSign is Microsoft's preferred provider of digital certificate services. Through the use of digital signatures, software developers can include information about themselves and their code with their programs.

When users download software code signed with a VeriSign Digital Certificate, they can be assured of

- **Content Source**—The software really comes from the publisher who signed it.
- **Content Integrity**—The software has not been altered or corrupted since it was signed.

Users benefit from this software accountability because they know who published the software and that the code has not been tampered with. In the extreme case, where software performs unacceptable or malicious activity on their computers, users can also pursue recourse against the publisher. This accountability and potential recourse serve as a strong deterrent to the distribution of harmful code.

Developers benefit from Microsoft® Office/VBA Macro Signing because it puts trust in their name and makes their products harder to falsify. By signing code, developers build a trusted relationship with users, who then learn to confidently download signed software from that publisher. With Office/VBA Macro Signing, developers can create signed macros, and users can make educated decisions about which software to download, knowing who published the software and that it has not been tampered with.

## Who Needs a Code Signing Digital Certificate?

---

Any publisher who plans to distribute code or content over the Internet or over corporate extranets risks impersonation and tampering. The VeriSign® Software Publisher Digital Certificate for Office/VBA protects against these hazards. In particular, if you are distributing macros, you will want to sign them using a VeriSign® Microsoft® Office/VBA Code Signing Digital Certificate.

VeriSign offers a Class 3 digital certificate designed for commercial software publishers. This class of digital certificate provides greater assurance about the identity of a publishing organization and is designed to represent the level of assurance provided today by retail channels for software.

## What Does Microsoft Office/VBA Macro Signing Look Like to Consumers?

Microsoft® Word, Excel, PowerPoint®, and Outlook® 2000 and later applications support signing and verifying digital signatures on VBA code. Other third-party applications with VBA 70 may also support digital signatures in VBA code (check with your application). If an end user of one of these applications encounters an unsigned VBA macro, the following will occur:

- If the application's security settings are set on "High," the client application will not permit the unsigned code to run.
- If the application's security settings are set on "Medium," the client application will display a warning, which asks the user if he or she wants to enable or disable this unsigned code.

By contrast, if a user encounters signed VBA code in a file, the user is informed of the following:

- The true identity of the publisher
- That there is no problem with the certificate (the lack of additional warnings)
- That the authenticity of the above information is provided by VeriSign (by clicking the "Details" button)

Users can choose to trust all subsequent VBA code from the same publisher source.

Simply by clicking the "More Info" button, users can inspect the certificate and verify its validity.

## Technical Overview: (Optional Reading)

### What Is a Digital Certificate?

A digital certificate is a form of electronic credential for the Internet. Similar to a driver's license, employee ID card, or business license, a digital certificate is issued by a trusted third party to establish the identity of the certificate holder. The third party who issues certificates is known as a certification authority (CA).

Digital-certification technology is based on the theory of public key cryptography. In public key cryptography systems, every entity has two complementary keys—a public key and private key—which function only when they are held together. Public keys are widely distributed to users, while private keys are kept safe and only used by their owner. Any code digitally signed with the publisher's private key, can only be successfully verified using the complementary public key. Another way to look at this is that code successfully verified

using the publisher's public key (which is sent along with the digital signature), can only have been digitally signed using the publisher's private key (thus authenticating the source of the code), and has not been tampered with. For more information on public keys and private keys, please see the VeriSign White Paper, "Introduction to Public Key Cryptography."

The purpose of a digital certificate is to reliably link a public/private key pair with its owner. When a CA, such as VeriSign, issues a certificate, it verifies that the owner is not claiming a false identity. Just as when a government issues you a passport, it is officially vouching for the fact that you are who you say you are, when a CA issues you a digital certificate, it is putting its name behind the statement that you are the rightful owner of your public/private key pair.

#### + CAs

CAs, such as VeriSign, are organizations that issue digital certificates to applicants whose identity they are willing to vouch for. Each certificate is linked to the certificate of the CA that signed it.

As the Internet's leading CA, VeriSign has the following responsibilities:

- Publishing the criteria for granting, revoking, and managing certificates
- Granting certificates to applicants who meet the published criteria
- Managing certificates (for example, enrolling, renewing, and revoking them)
- Storing VeriSign root keys in an exceptionally secure manner
- Verifying evidence submitted by applicants
- Providing tools for enrollment
- Accepting the liability associated with these responsibilities
- Time stamping digital signatures

#### + How Does Microsoft Office/VBA Work with VeriSign Digital Certificates?

Microsoft Office/VBA relies on industry standard cryptography techniques such as X.509v3 certificates and Public Key Cryptography Standard (PKCS) #7 and #10. These are well-proven cryptography protocols, which ensure a robust implementation of code signing technology.

Microsoft Office/VBA uses digital signature technology to assure users of the origin and integrity of software. In digital signatures, the private key generates the signature, and the corresponding public key validates it. To save time, the Microsoft Office/VBA protocols use a cryptographic digest, which is a one-way hash of the document.

The process is outlined below:

1. The publisher obtains a VeriSign Code Signing Digital Certificate.
2. The publisher creates code.
3. Using the Microsoft Office utility, the publisher:
  - Creates a hash of the code, using an algorithm such as message digest 5 (MD5) or secure hash algorithm (SHA),
  - Encrypts the hash using his or her private key,
  - Creates a package containing the code, the encrypted hash, and the publisher's certificate.

4. The end user encounters the package.
5. The end user's Microsoft Office utility examines the publisher's certificate. Using the VeriSign root Public Key, which is already embedded in Microsoft Office/VBA-enabled applications, the end user's Microsoft Office utility verifies the authenticity of the Code Signing Digital Certificate (which is itself signed by the VeriSign root Private Key).
6. Using the publisher's public key contained within the publisher's certificate, the end user Microsoft Office or VBA application decrypts the signed hash.
7. The end user's Microsoft Office or VBA application runs the code through the same hashing algorithm as the publisher, creating a new hash.
8. The end user's Microsoft Office or VBA application compares the two hashes. If they are identical, the browser delivers the message that VeriSign has verified the content. The end user then has confidence that the code was signed by the publisher identified in the digital certificate, and that the code has not been altered since it was signed.

The entire process is seamless and transparent to end users, who see only a message that the content was signed by its publisher and verified by VeriSign.

#### **+ Timestamping**

Because key pairs are based on mathematical relationships, which can theoretically be "cracked" with a great deal of time and effort, it is a well-established security principle that digital certificates should expire. Your VeriSign Digital Certificate will expire one year after it is issued. However, most software is intended to have a lifetime of longer than one year. To avoid having to resign software every time your certificate expires, VeriSign and Microsoft introduced a timestamping service. Now, when you sign code, a hash of your code will be sent to VeriSign to be timestamped. As a result, when your code is downloaded, clients will be able to distinguish between:

- Code signed with an expired certificate, which should NOT be trusted
- Code signed with a certificate that was valid at the time the code was signed, but has subsequently expired, which SHOULD be trusted

This means that you will not need to worry about resigning code when your VeriSign Digital Certificate expires. VeriSign is the only certification authority offering the time stamping service. This service is free to all VeriSign Commercial and Individual Code Signing Certificate customers.

#### **+ The Six Steps to Signing Code**

Signing Code is an easy six-step process. By following the instructions below, you will be signing code in no time.

Step 1: Make Sure You Are Running the Correct Versions of All Tools

The tools include:

- Internet Explorer 4.0 or later
- Windows® 98 or 2000 and later

Step 2: Apply for a VeriSign Microsoft Office/VBA Code Signing

### Digital Certificate

For instructions on obtaining a Code Signing Digital Certificate, go to VeriSign enrollment for instructions on obtaining a Code Signing Digital Certificate.

In the process of applying for a VeriSign Code Signing Digital Certificate, your browser will generate a private key. You should store this private key (called MyPrivateKey.pvk) on a floppy disk that you store in a safety deposit box or other secure location. Please make a backup copy of this private key, as you will need this key to sign code. This key is never sent to VeriSign, so if you lose this private key, you will be unable to sign code. If this key is lost or stolen, please contact VeriSign immediately.

### Step 3: Pick up Your Digital Certificate

Once you have completed the application process, VeriSign will take a number of steps to verify your identity. For commercial publishers, VeriSign does a considerable amount of background checking. As a result, it will take approximately three to five days to verify your information and issue a certificate.

At the end of this process, VeriSign will send you an email containing a personal identification number (PIN). Follow the instructions in this email to pick up your certificate, and save it as a file (e.g. MyCredentials.spc).

Please note that you must use the same machine to apply for and obtain your certificate. You can then use the private key and certificate to sign files on a different machine.

### Step 4: Prepare for Timestamping

Set the registry key, HKEY\_Current\_User\Software\Microsoft\VBA\Security\TimeStampURL, to: [timestamp.verisign.com/scripts/timestamp.dll](http://timestamp.verisign.com/scripts/timestamp.dll)

That is the uniform resource locator (URL) for the VeriSign timestamping service. Please note that “timestamp.dll” does not contain the letter “e”.

### Step 5. Sign Your Files

You can now sign your .doc, .dot, .xls, .xlt, .xla, .ppt, .pps, and .ppa files. To sign, load the file in the appropriate Office application, and use the Digital Signature command on the Tools menu in the Visual Basic Editor.

### Step 6: Test Your Signature

Close and reopen your file with the appropriate Office application. If your signing process was successful, this will bring up a security warning with the digital signature information. Congratulations, you have just digitally signed your file. If the file is tampered with in any way after it has been signed, Office will attempt to re-sign. If the current user does not have the certificate, the user will be notified. This makes it easy for you to edit your macros and keep them signed. To add more security for your private key, we recommend you password-protect your private key.

### + How Do You Require a Password before the Application Reuses

### Your Private Key?

With Internet Explorer 5 and above, you can make it so that a user has to type a password before Office uses the private key of any of your personal certificates. This should prevent the unauthorized use of your certificate if you leave your machine unsecured. The password prompt helps to notify you when your project has changed and Office is attempting to automatically re-sign your code. If you did not intentionally change your code, forms, or add or remove ActiveX controls, then you should suspect a virus entered your VBA code.

To set a password for your private key, use the following steps:

1. When in the Certificate Manager Import Wizard, choose the “Enable strong private key protection” checkbox (you will see this checkbox in the same dialog where you enter the password for your exported .pfx file).
2. When you choose to “Finish” the Wizard, you will see a Private Key Container dialog; choose the “Set Security Level...” button.
3. Choose the “High” option to specify a password.
4. Choose “Next.”
5. Write your name in the “Password for” textbox.
6. Type your new Password for that name and confirm it.
7. Choose “Finish.”
8. The Certificate Store will ask you for the password again. This is the dialog that you will see whenever your private key is used. It specifies what the private key is being used for and asks for your password to authorize it. Type your password.
9. Choose “OK” to finish.

## Conclusion

Microsoft and VeriSign are committed to making the Internet a secure and viable platform for commerce and the distribution of content. With Code Signing for Microsoft Office/VBA and VeriSign Code Signing Digital Certificates, your code will be as safe and trustworthy to your customers as it would be if you shrink-wrapped it and sold it off a store shelf.

**Visit us at [www.VeriSign.com](http://www.VeriSign.com) for more information.**