

Entrust Certificate Services

# Authenticode Signing

## User Guide

Document issue: 2.0

Date of Issue: October 2009



# Revision information

Table 1: Revisions in this document

Document issue and date	Section	Description
Document issue 2 Oct. 26, 2009	Opening paragraph "Signing Microsoft Authenticode" on page 8	Notes specifying that Entrust code signing certificates for Microsoft® Authenticode cannot be used to sign kernel mode software have been added to the guide.

**Copyright © 2009 Entrust. All rights reserved.**  
Entrust is a trademark or a registered trademark of Entrust, Inc. in certain countries. All Entrust product names and logos are trademarks or registered trademarks of Entrust, Inc. in certain countries. All other company and product names and logos are trademarks or registered trademarks of their respective owners in certain countries.

This information is subject to change as Entrust reserves the right to, without notice, make changes to its products as progress in engineering or manufacturing methods or circumstances may warrant.

Export and/or import of cryptographic products may be restricted by various regulations in various countries. Export and/or import permits may be required.

## Obtaining technical support

For support assistance by telephone call one of the numbers below:

- 1 (866) 267-9297 in North America
- 1 (613) 270-2680 outside North America

You can also email Customer Support at:

- [SSL@entrust.com](mailto:SSL@entrust.com)

# Code signing

This guide contains information about signing Microsoft® Authenticode files. Sections in this guide include:

- “The code signing process for Microsoft Authenticode”
- “Verifying the authenticity of the software”
- “Obtaining and using an Entrust Microsoft Authenticode signing certificate”

Use Entrust certificates for Microsoft® Authenticode to sign CAB, CAT, CTL, DLL, EXE, and OCX files. Browsers use the signature and its accompanying information to provide some confidence to the end user that the code is from a legitimate source and is free of tampering. Entrust offers PKCS#7 (Public Key Cryptography Standard # 7) certificates for use with Authenticode.

---

**Note:** You cannot use Entrust code signing certificates for Microsoft Authenticode to sign kernel mode software.

---

- For information about using an Entrust certificate with Microsoft Office files see the *Entrust Certificate Services Microsoft Office and VBA Code Signing Guide* available from [www.entrust.net](http://www.entrust.net).
- For information about using an Entrust certificate with Java code see the *Entrust Certificate Services Java Code Signing Guide* available from [www.entrust.net](http://www.entrust.net).

# The code signing process for Microsoft Authenticode

When the code is signed, several pieces of information are added to the file. This information is used when the code is downloaded through your browser to authenticate the author of the code and to check for tampering.

The bundle that is used to verify the authenticity of the code is created during two sequences of events.

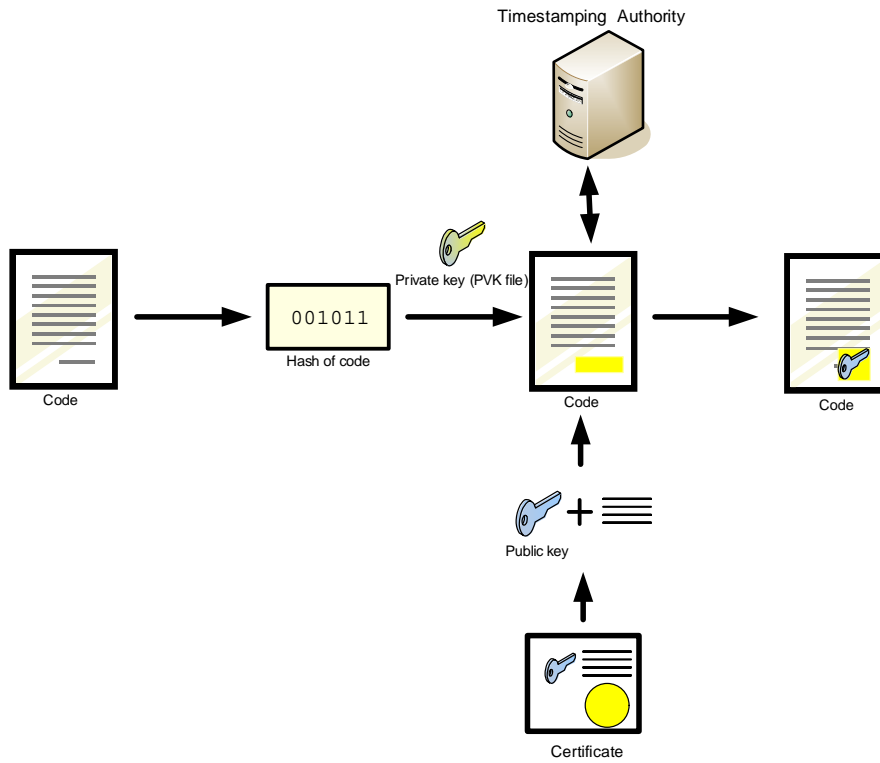
- A mathematical representation of the code, called a hash, is created and signed. The hash and signature are timestamped, hashed (with the timestamp) and signed again.
- The timestamp and second signature are applied by a timestamping authority (TSA). Timestamping Authorities are usually maintained by a third party (such as Entrust) that can insure the validity of the timestamp.

The entire sequence takes place as follows:

- The code is passed through a hashing algorithm creating a hash of the file. The hash is an exact numerical representation of the file. The hash is only reproducible using the unaltered file and the hashing algorithm that was used to create the hash. The hash is bundled with the file.
- The hash is signed using the signer's private key.
  - Information identifying the creator of the signature is drawn from the signer's certificate and incorporated into the signature.
  - Information about the CA or CAs that signed the signer's certificate is drawn from the signer's certificate and incorporated into the signature.
- The signer's public key is added to the bundle as it is required to authenticate the code when it is verified.
- The signature is sent to the timestamping authority (TSA).
  - The TSA adds a timestamp to the to the bundled information and computes a new hash.
  - The TSA signs the new hash with its private key creating a new bundle of information.
  - The timestamped bundle, original bundle that was sent to the TSA and the time stamp are re-bundled with the original code.

**Figure 1:** The code-signing process for Microsoft Authenticode

Authenticode signing process



# Verifying the authenticity of the software

When the the end user's browser loads the code, it checks the authenticity of the software using the signer's public key, signature and the hash of the file. The timestamp is checked using a similar process.

If both the timestamp and the signature are verified successfully, the browser accepts the code as valid. If either the timestamp or signature are not successfully verified, the browser will react by warning the user or rejecting the code, according to the level of security being used.

## Verifying the timestamp

The following sequence of events is used to verify the timestamp.

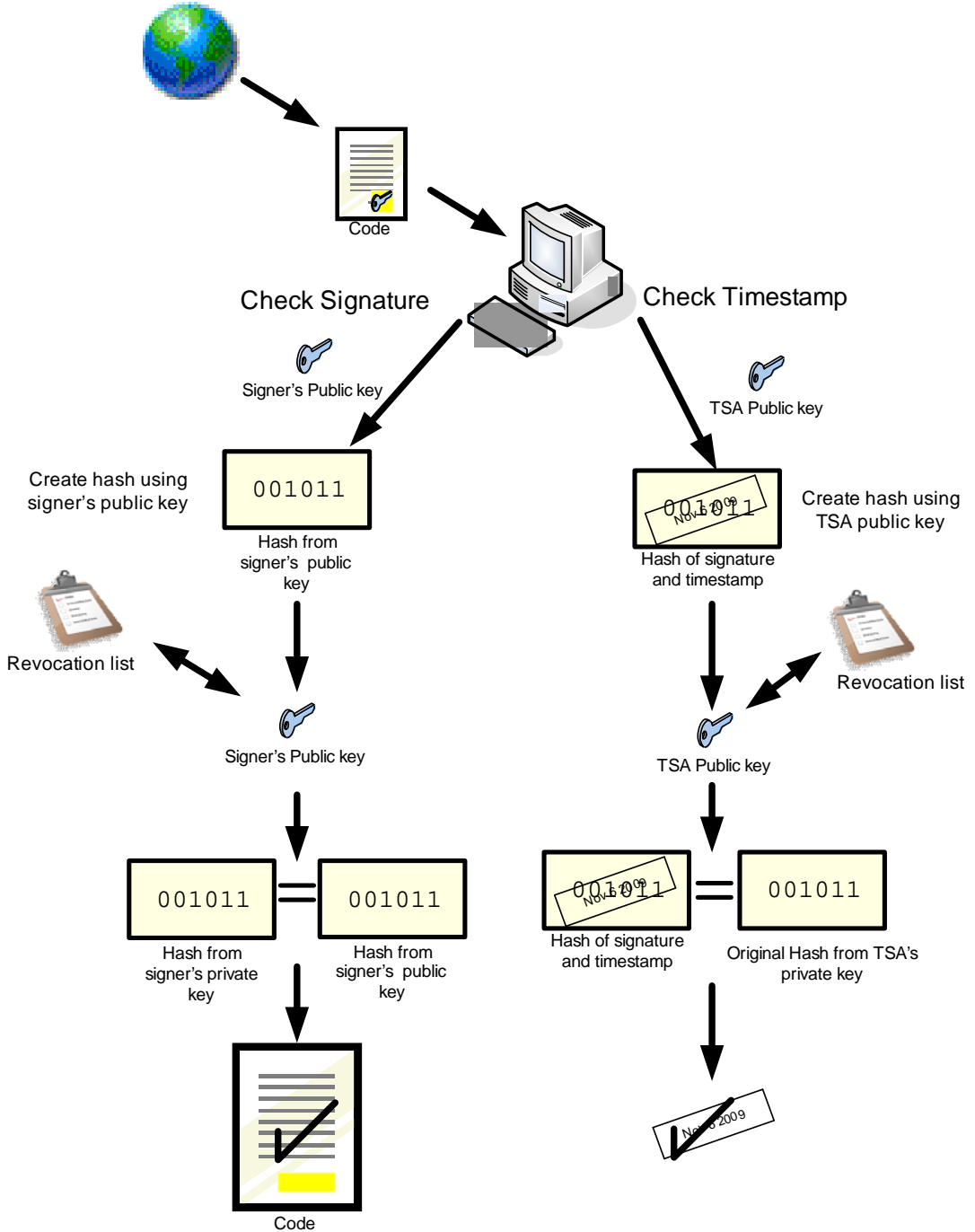
- The timestamp is added to the bundled signature information and the combined signature and timestamp are hashed.
- The Timestamping Authority's public key is applied to the timestamped signature block revealing the hash calculated by the TSA.
- The validity of the TSA's public key is verified by checking its expiry date and consulting the revocation lists to be sure that it has not been revoked.
- The two hashes are compared. If the hashes are equal, the timestamp is considered to be valid.

## Verifying the signature

The signature is verified as follows:

- The original code is passed through a hashing algorithm creating a hash.
- The public key of the designer or publisher is extracted from the bundle and applied to the signature information. Applying the public key reveals the hash that was calculated when the file was signed.
- The expiry date of the public key is checked.
- The public key is checked against the revocation lists to be sure that it is valid.
- The two hashes are compared. If equal, the signature is considered to be valid.
- If the file is considered to be valid it is accepted by the browser. If the file is not considered to be valid the browser takes the security measure appropriate to its current level of security.

**Figure 2:** Verifying the authenticity of the code



# Obtaining and using an Entrust Microsoft Authenticode signing certificate

When you install the certificate, a private key (PVK) is created on your machine. This process provides added security as the private key does not exist until it is created on the signer's computer. Microsoft Authenticode is signed using Signtool—an application that is included when you download and install Microsoft .NET.

## Obtaining a certificate from Entrust

To obtain a code signing certificate from the Entrust, log into the Entrust Web site URL <https://buy.entrust.net/buy>. Code signing certificates are available to users who have registered for the Entrust Certificate Management System (CMS). For information about enrolling in the CMS see the *Entrust Certificate Management System Enrollment Guide*. For information about buying and managing code signing certificates see the *Entrust Certificate Management System User Guide*.

## Signing Microsoft Authenticode

The procedure in this section assumes:

- that you have purchased and installed an Entrust certificate for signing Authenticode.
- Microsoft .NET version 2.0 or higher has been installed on your machine.

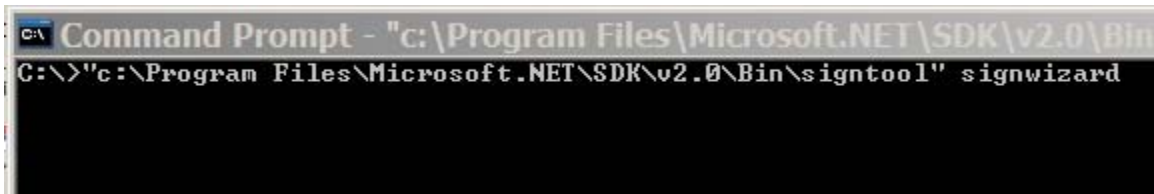
To sign code using signtool with SPC and PVK files

- 1 Start the signtool wizard from the Command prompt. Signtool is included in Microsoft.NET version 2.0 or higher. To start the wizard open a Command prompt and type:

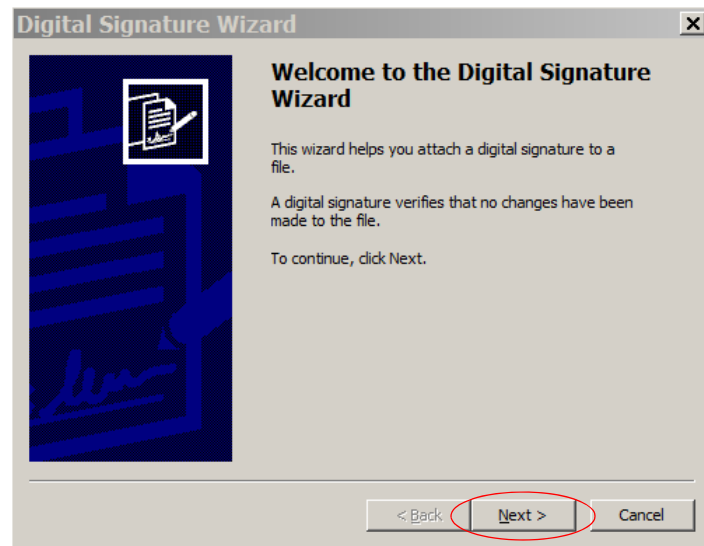
```
"<Path to the signtool executable>\signtool" signwizard
```

For example:

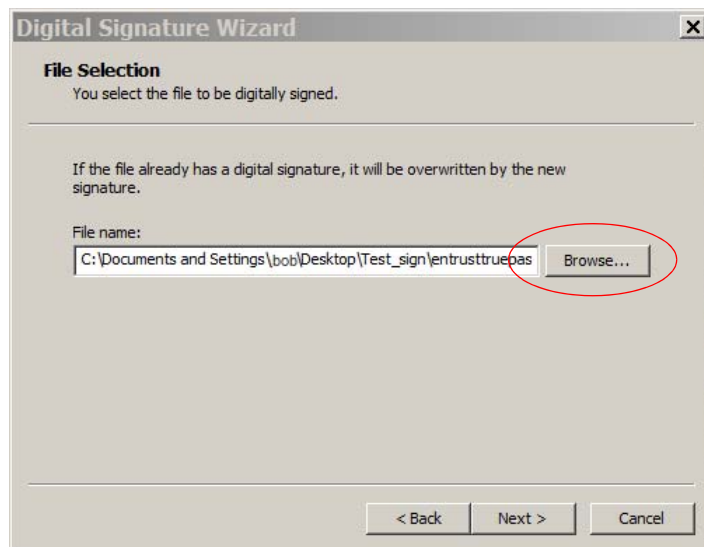
```
"C:\Program Files\Microsoft.NET\SDK\v2.0\Bin\signtool" signwizard
```



- 2 Click **Next** in the **Welcome** page of the wizard.



- 3 On the **File Selection** page, click **Browse** and select the file to sign.



Supported file types have the extension CAB, CAT, CTL, DLL, EXE, and OCX.

---

**Note:** You cannot use Entrust code signing certificates for Microsoft Authenticode to sign kernel mode software.

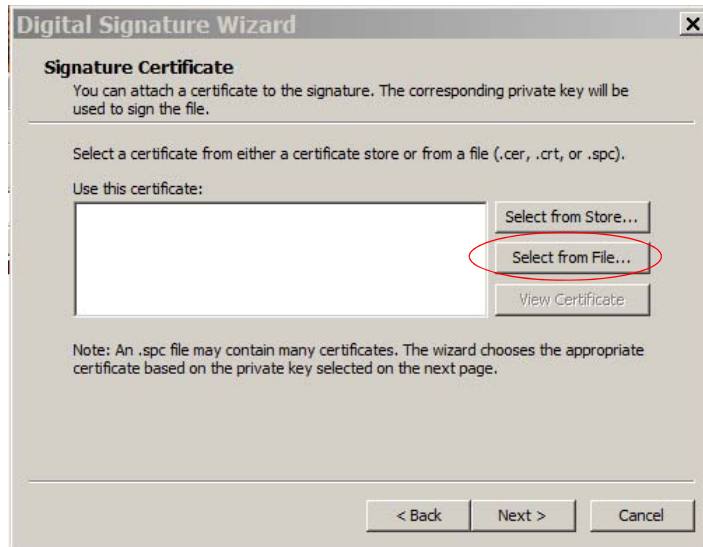
---

Click **Next** when you have finished selecting the file.

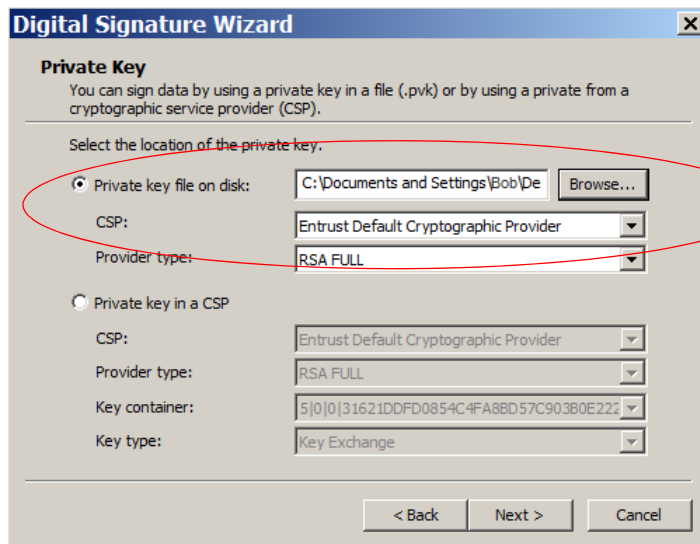
- 4** On the **Signing Options** page, select **Custom** and click **Next**.



- 5 Select the signature certificate to use to sign the code. On the **Signature Certificate** page, click **Select from File**. Browse to the location of your SPC file, select it, and click **Open**. Click **Next**



- 6 Select the private key file to use. On the **Private Key** page, select **Private key file on disk**. Browse to the location of the PVK file, select it and click **Open**. Click **Next**.



- 7 When prompted, enter the password for your private key and click **OK**.



- 8 On the **Hash Algorithm** page, click **Next** to accept the default hash algorithm.



- 9 On the **Additional Certificates** page, click **Next** to accept the default values.

The screenshot shows the 'Additional Certificates' dialog box in the Digital Signature Wizard. The title bar reads 'Digital Signature Wizard'. The main heading is 'Additional Certificates'. Below the heading, a paragraph states: 'You decide which certificates to include in the digital signature: all or selected certificates in the certification path, or additional optional certificates.'

There are two main sections:

- Certificates in the certification path:** This section contains three radio buttons:
  - All certificates in the certification path, except for the root certificate
  - All certificates in the certification path, including the root certificate (circled in red)
  - Only the signature certificate
- Other certificates (optional):** This section contains two radio buttons:
  - Certificates contained in the following PKCS #7 Certificates (.p7b) file:
    - File name: [text box] [Browse...]
  - Certificates in the following certificate store:
    - Store: [text box] [Browse...]
  - No additional certificates (circled in red)

At the bottom of the dialog, there are three buttons: '< Back', 'Next >', and 'Cancel'.

- 10 On the **Data Description** page, optionally enter a description of the data being signed or the URL of a Web page containing the description.

The screenshot shows the 'Data Description' dialog box in the Digital Signature Wizard. The title bar reads 'Digital Signature Wizard'. The main heading is 'Data Description'. Below the heading, a paragraph states: 'You can add a description of the data you are signing or a Web location containing a description.'

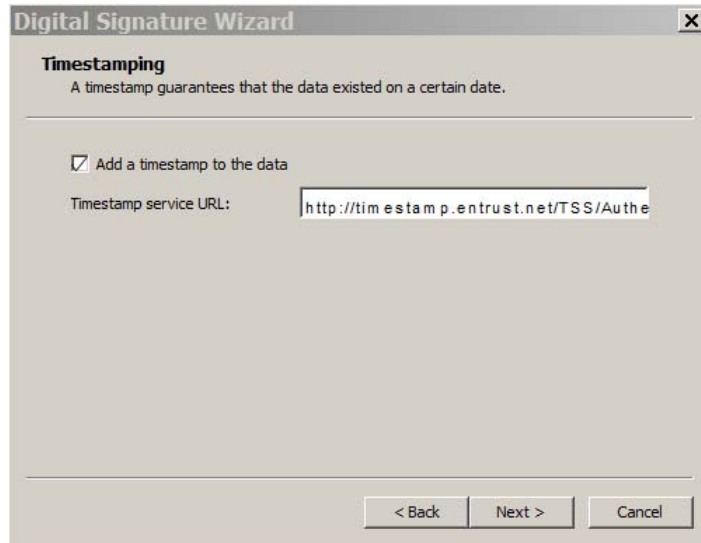
Below this, a paragraph reads: 'Either type a description, or provide a Web address that points to a site that contains a description.'

There are two text input fields:

- Description (optional):** [text box]
- Web location (optional):** [text box]

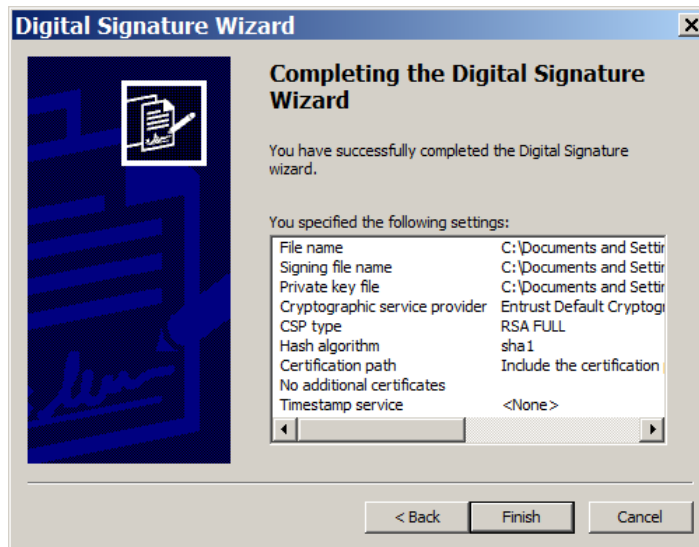
At the bottom of the dialog, there are three buttons: '< Back', 'Next >', and 'Cancel'.

- 11 If you want to timestamp your signature, select the timestamping checkbox and enter the URL of the Entrust timestamping service <http://timestamp.entrust.net/TSS/AuthenticodeTS>. A timestamp is used to indicate that the file has existed, without tampering, since the date of the timestamp.



- 12 On the **Completing the Digital Signature Wizard** page, check the information that you have entered in previous steps. If the information is correct, click **Finish**,

if not use the **Back** button to return to the original page where the incorrect information was entered and correct any errors.



**13** When prompted, enter the password for your private key and click **OK**.

