

# Signing your MSI with a digital signature

Once you have created an MSI package, you may want to sign it using a digital signature. A digital signature will enable the recipient of the MSI package able to verify the origin and ensure that nobody modified the content of the package. It is highly recommended that you sign your MSI files if you plan to distribute them to other users outside your organization.

There are many ways to sign your package. In this example, we will look at signtool.exe, which is a Microsoft program. It is distributed with the Windows Platform SDK, so if you are a software developer, you may already have it installed. Otherwise, you can download the signtool.exe and the SDK from Microsoft's web site.

You can find a lot of general documentation about using signtool on the Internet. Therefore, we will focus on a simple example that shows you how to use it with an MSI package.

Before you can start signing your packages, you must get yourself a code signing certificate. Most leading certificate providers have this type of certificate for sale. A certificate usually expires after 1 to 5

years but that does not mean that the packages you have signed will stop working.

If you get your certificate in a file, it will typically have a PFX or P12 extension. The command line for signtool needs the thumbprint property of your certificate. You can get the certificate thumbprint using the following PowerShell command:

```
Get-PfxCertificate -FilePath .\certificate.pfx
```

Once you have the thumbprint, you can use the following command line for signtool to sign your MSI:

```
signtool.exe sign /f certificate.pfx /d "My Description" /p PW1234 /v /sha1 1475F2E273906580AF578416E6ACE8C91AE3E62D /t  
"http://timestamp.comodoca.com/authenticode" setup.msi
```

In PowerShell you may need the `--%` to parse the parameters correctly:

```
.\signtool.exe --% sign /f certificate.pfx /d "My Description" /p PW1234 /v /sha1 1475F2E273906580AF578416E6ACE8C91AE3E62D  
/t "http://timestamp.comodoca.com/authenticode" setup.msi
```

It is important that you include the /d for the description. Otherwise, Windows will show the name of your package as a random number. The random name can confuse end users when they see the UAC prompt for access to make changes to the computer. As mentioned, this will go away with the /d parameter.

**Contact**

**Download**

**Maintenance**

**About**

**Order PRO**

**Reseller Information**

**Support**

**Getting started**

**Faq**

**Documentation**