While you can create a self-signed code-signing (SPC) certificate in one go, I prefer to do the following:

Creating a self-signed Certificate Authority (CA)

makecert -r -pe -n "CN=My CA" -ss CA -sr CurrentUser ^

-a sha256 -cy authority -sky signature -sv MyCA.pvk MyCA.cer

(^ = allow batch command-line to wrap line)

This creates a self-signed (-r) certificate, with an exportable private key (-pe). It's named "My CA", and should be put in the CA store for the current user. We're using the sha256 algorithm. The key is meant for signing (-sky).

The private key should be stored in the MyCA.pvk file, and the certificate in the MyCA.cer file.

Importing the CA Certificate

Because there's no point in having a CA certificate if you don't trust it, you'll need to import it into the Windows certificate store. You *can* use the Certificates MMC snapin, but from the command line:

certutil -user -addstore Root MyCA.cer

Creating a code-signing (SPC) Certificate

makecert -pe -n "CN=My SPC" -a sha256 -cy end ^

-sky signature ^

-ic MyCA.cer -iv MyCA.pvk ^

-sv MySPC.pvk MySPC.cer

Pretty much the same as above, but we're providing an issuer key and certificate (the -ic and -iv switches).

We'll also want to convert the certificate and key into a PFX file:

pvk2pfx -pvk MySPC.pvk -spc MySPC.cer -pfx MySPC.pfx

If you want to protect the PFX file, add the -po switch, otherwise PVK2PFX creates a PFX file with no passphrase.

Using the certificate for signing code

signtool sign /v /f MySPC.pfx MyExecutable.exe

If you import the PFX file into the certificate store (you can use PVKIMPRT or the MMC snapin), you can sign code as follows:

signtool sign /v /n "Me" /s SPC /d http://www.me.me ^

/t http://timestamp.url MyExecutable.exe

Some possible timestamp URLs for signtool /t are:

* http://timestamp.verisign.com/scripts/timstamp.dll
* http://timestamp.globalsign.com/scripts/timstamp.dll
* http://timestamp.comodoca.com/authenticode

Full Microsoft Documentation

* signtool: <http://msdn.microsoft.com/en-us/library/8s9b9yaz.aspx>
* makecert: <http://msdn.microsoft.com/en-us/library/bfsktky3.aspx>
* pvk2pfx : <http://msdn.microsoft.com/en-us/library/windows/hardware/ff550672(v=vs.85).aspx>

Downloads

For those who are not .NET developers, you will need a copy of the Windows SDK and .NET framework. A current link is available here: [SDK & .NET](http://msdn.microsoft.com/en-us/windowsserver/bb980924.aspx) (which installs makecert here: C:\Program Files\Microsoft SDKs\Windows\v7.1). Your mileage may vary.