

Digital Signatures with Transceiver

Signatures and certificates

Transceiver offers a solution for creating digital signatures on any file, not limited to Word, Excel or PDF. The digital signatures are not stored in the file itself, but in a separate file. E.g.: if you have a file “Contract.pdf”, the signatures are stored in “Contract.pdf.t-sig”.

This T-Sig file is based on the XAdES 1.4.1¹ standard. To create a signature, you need an X.509 v3 certificate that supports non-repudiation (like the Belgian eID card²). Optionally, the signatures can also be timestamped by a trusted timestamping authority following the RFC 3161³ standard.

Transceiver products that support digital signatures

- **Transceiver Communicator**
An interactive Windows application to securely transmit and receive files.
- **Transceiver Automator**
A workflow-driven Windows service to automate the handling of incoming and outgoing files.
- **Transceiver Q-Sign**
An interactive Windows application to create and verify digital signatures.
(not publically available yet)

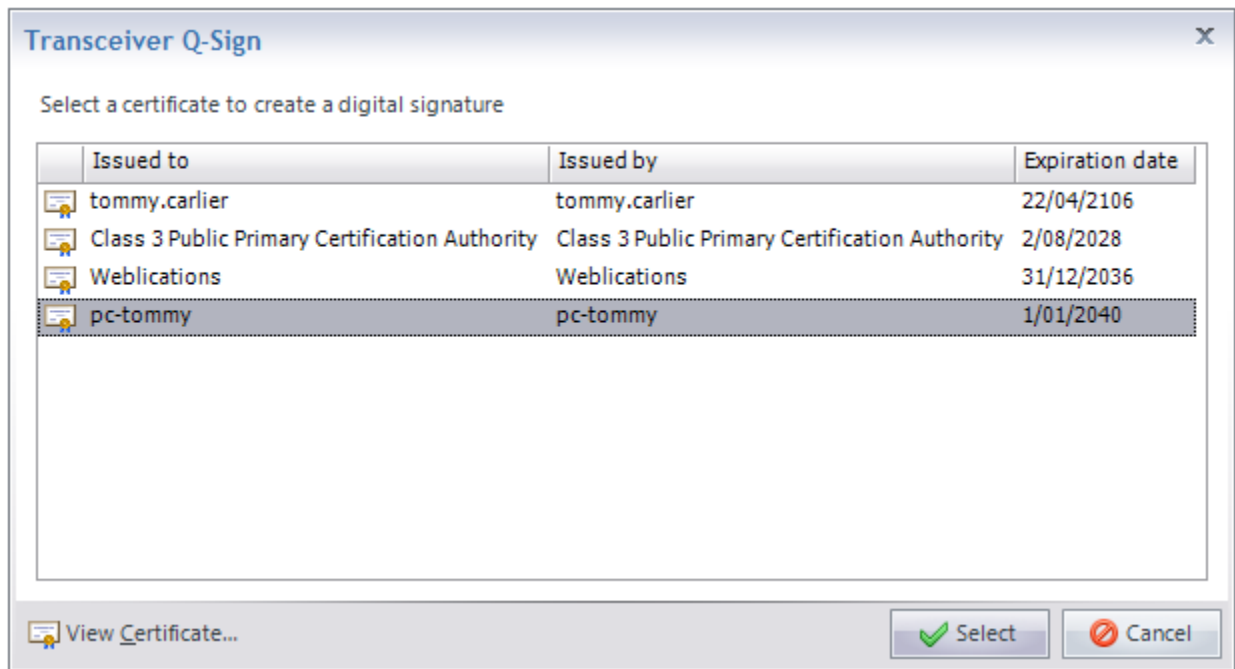
¹ XAdES 1.4.1: http://uri.etsi.org/01903/v1.4.1/ts_101903v010401p.pdf

² Belgian eID: <http://eid.belgium.be/>

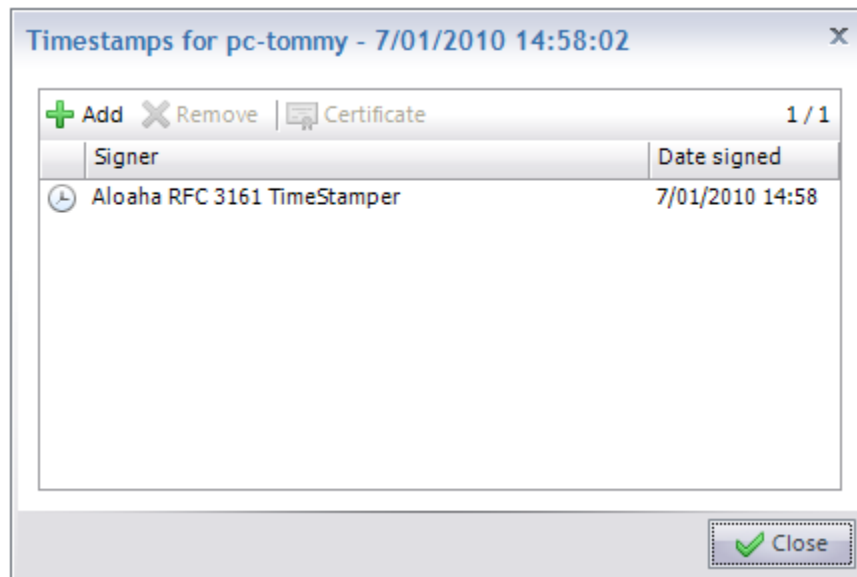
³ RFC 3161: <http://tools.ietf.org/html/rfc3161>

Creating a digital signature interactively

Select a certificate to sign with:

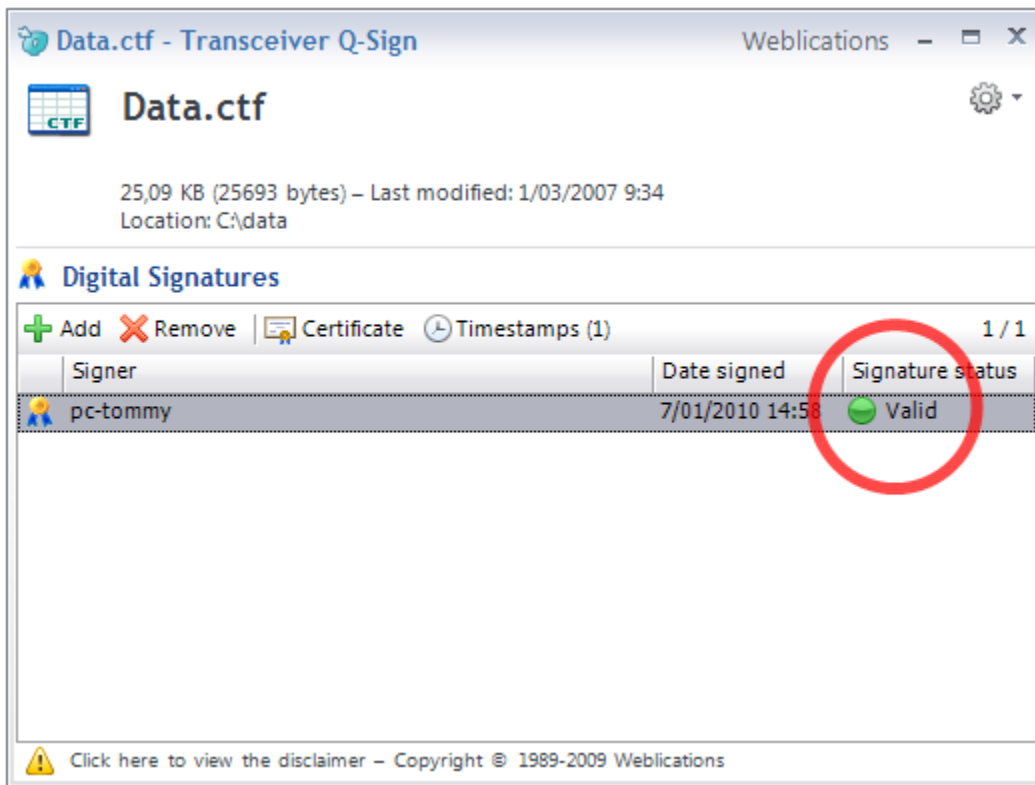


Optionally, timestamp the signature:

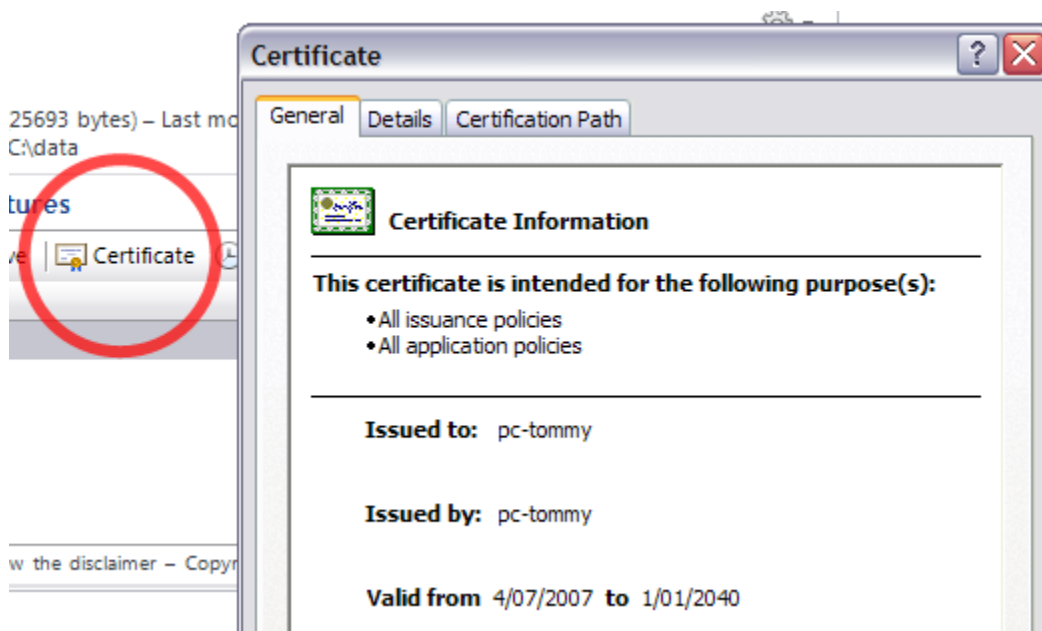


Verifying a signature

Check the signature status:

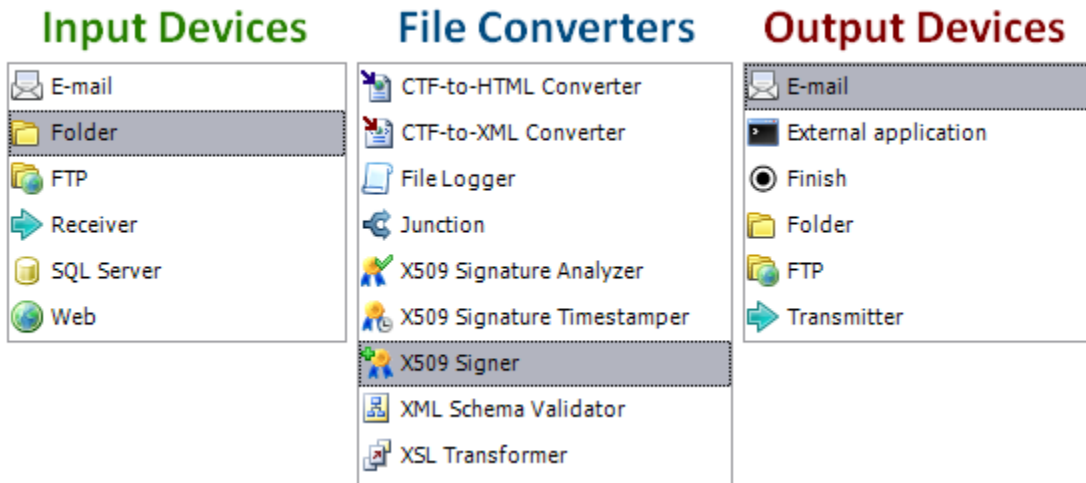


Optionally, check the certificate:



Creating a digital signature automatically using Transceiver Automator

With Transceiver Automator you can create workflows to automatically create digital signatures and trusted timestamps for files that are received from e-mail, FTP, Transceiver, the Web or your local file system. After processing the file, it can be stored locally or on an FTP-server, transmitted via e-mail or Transceiver, or processed by an external application.



Here's an example of a workflow that gets files from a folder, adds a digital signature, adds a timestamp and stores the file (with signature) in a different folder:

Source	Destination
Input Folder Folder c:\data\test\in	~Sign X509 Signer
~Sign X509 Signer	~Timestamp X509 Signature Timestamper http://tsa.aloaha.com/(signatures:Last)
~Timestamp X509 Signature Timestamper http://tsa.aloaha.com/(signatures:Last)	Output Folder Folder c:\data\test\out\{SubFolder}

After dropping the file Data.ctf in the input folder, the file and signatures appear in the output folder:

CTF Data (Compact Table Format)



Transceiver Signatures



Transceiver Automator provides an API (in .NET 2.0) that enables you to write your own *input devices* (plug-ins that receive files), *output devices* (plug-ins that transmit files) or *file converters* (plug-ins that convert files).

The image shows a screenshot of Visual Studio. On the left, the Solution Explorer displays the namespace `Weblications.Transceiver.Automator.Extensibility` with a tree view of its members. The `FileConverter` class is highlighted. On the right, the Disassembler window shows the disassembled code for the `FileConverter` class, which is an abstract class inheriting from `Plugin`.

```

public abstract class FileConverter : Plugin
{
    // Fields
    private int fFileConverterPK;
    private InputDevice fInputDevice;
    private readonly object fLockUpdates;
    private string fName;
    private OutputDevice fOutputDevice;

    // Methods
    protected FileConverter();
    internal override string GetNameToLogStar
    internal void Initialize(InputDevice inputDevice)
    internal void Process(IAutomatorFileOutput outputDevice)
    protected abstract void ProcessCore(IAutomatorFileOutput outputDevice)
    internal void Update(string name, ParameterCollection parameters)

    // Properties
    internal int FileConverterPK { get; set; }
    internal InputDevice InputDevice { get; }
    public string Name { get; internal set; }
    internal OutputDevice OutputDevice { get; }
}

```