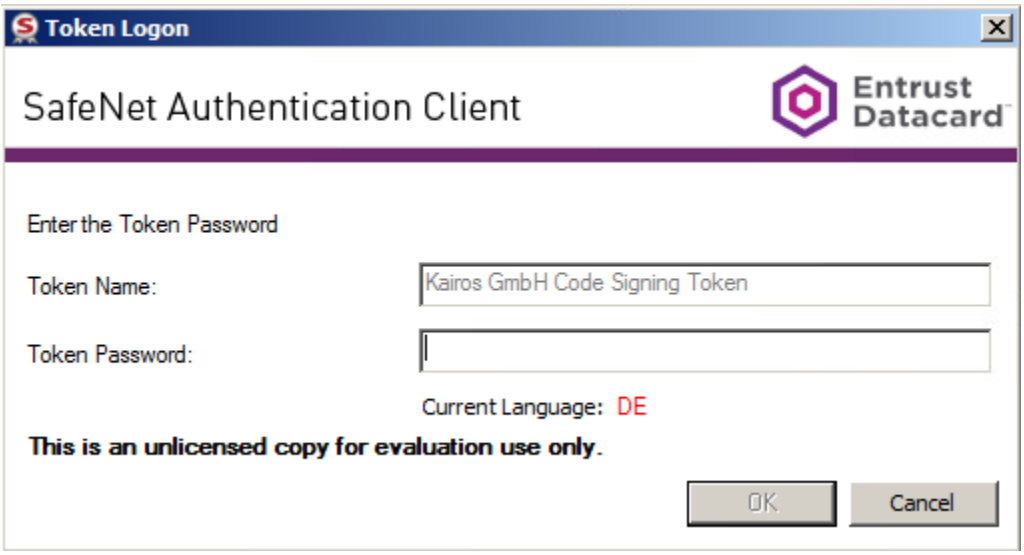
How-To MSI-Installer: EV certificate code signing with build server

In order to sign an MSI package to pass Windows Smart Screen, we need an EV certificate. The certificate with the public and private keys is shipped on a smart card USB eToken by the certificate vendor and cannot be copied to a hard disk. So the USB eToken has to be available on the build servr. Additionally the build agent needs the smart card software SafeNet Authentication Client to pass the certificate to the sigining tool.

> The SafeNet Authentication Client does not find the USB eToken if you connect via RDP. You need a console, VNC or Team Viewer session if your build agent is virtualized.
>
> The VNC password for msi.kaiorsbochum.de kan be found in the KairoIT.kdbx file.

Passing the eToken Password

Unfortunately the SafeNet Authentication Client Tools request the smart card token password at least once after every server reboot in a user session.



Advanced Installer by default uses the signtool.exe from the Microsoft SDK that is shipped with Advanced Installer. It is called using the following parameters depending on the build stage:

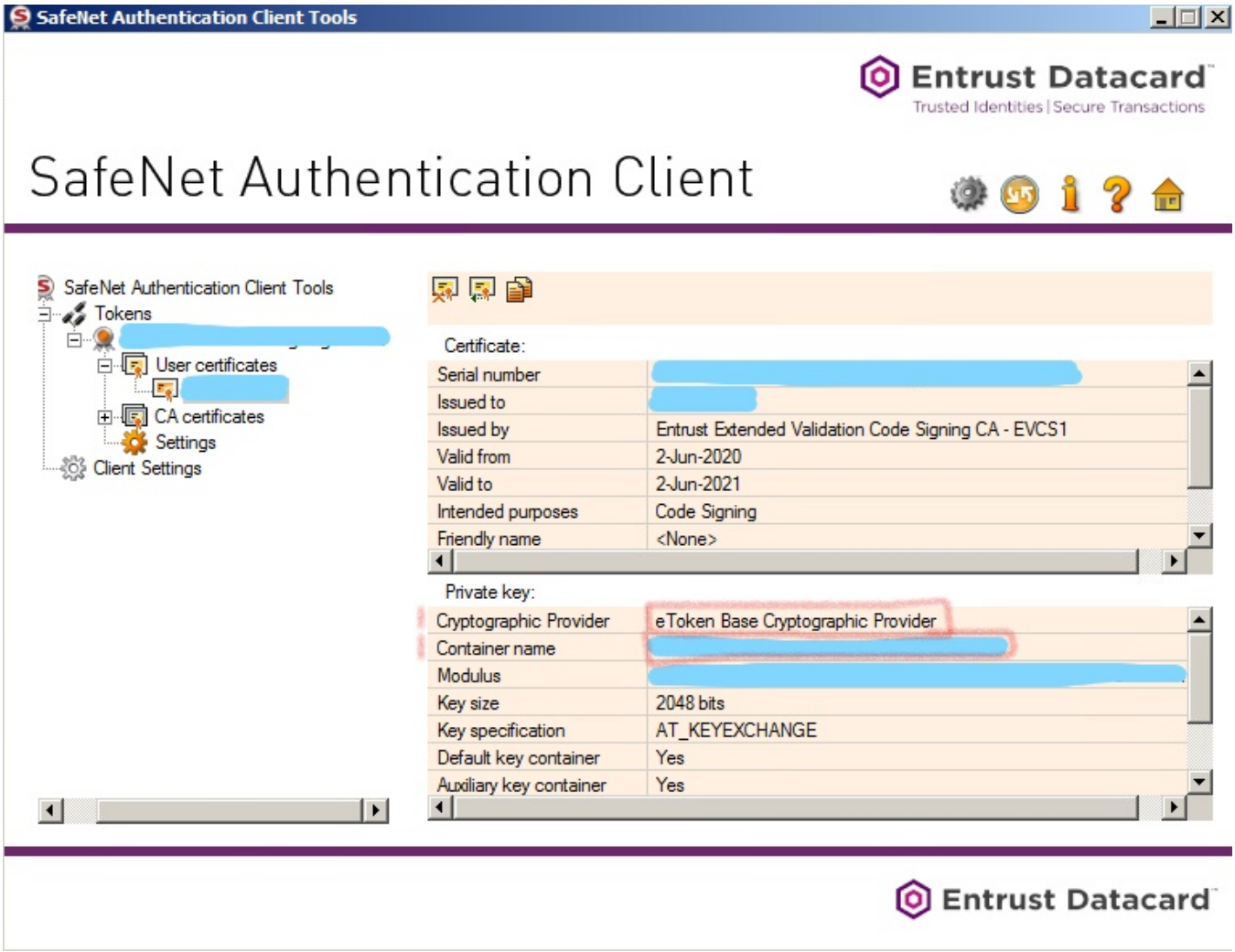
```
signtool.exe /a /d <product name> /t http://some.timestamp.server <file_to_sign>
signtool.exe /a /d <product name> /tr http://some.timestamp.server /td sha256 <file_to_sign>
```

Fortunately the password can be passed to signtool.exe in combination with the certificate container name, the cryptographic provider and the public certificate of your keypair (s. https://stackoverflow.com/questions/17927895/automate-extended-validation-ev-code-signing/47894907#47894907).

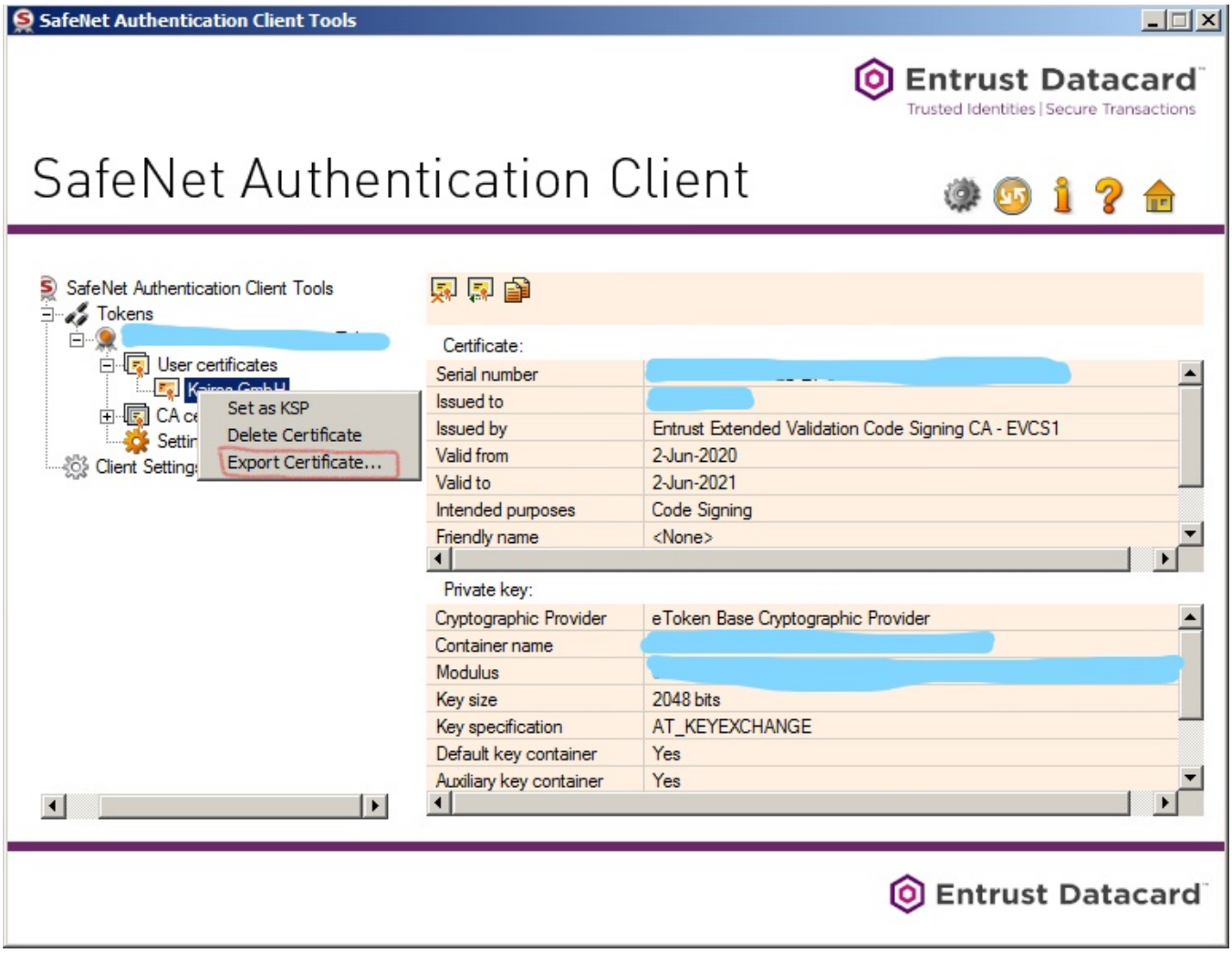
```
signtool.exe sign /f "Kairos_CodeSign.cer" /d "<product name>" /t http://some.timestamp.server /csp
"<cryptographic_provider>" /kc "[{{<token_password>}}]=<certificate_container_name>" "<file_to_sign>"
```

> Choosing the best suiting certificate via paramter /a does not work in this case.

You can get the cryptographic provider and the container name from the private key section of your eToken.



The public certificate can be exported from your eToken using SafeNet Authentication Client.

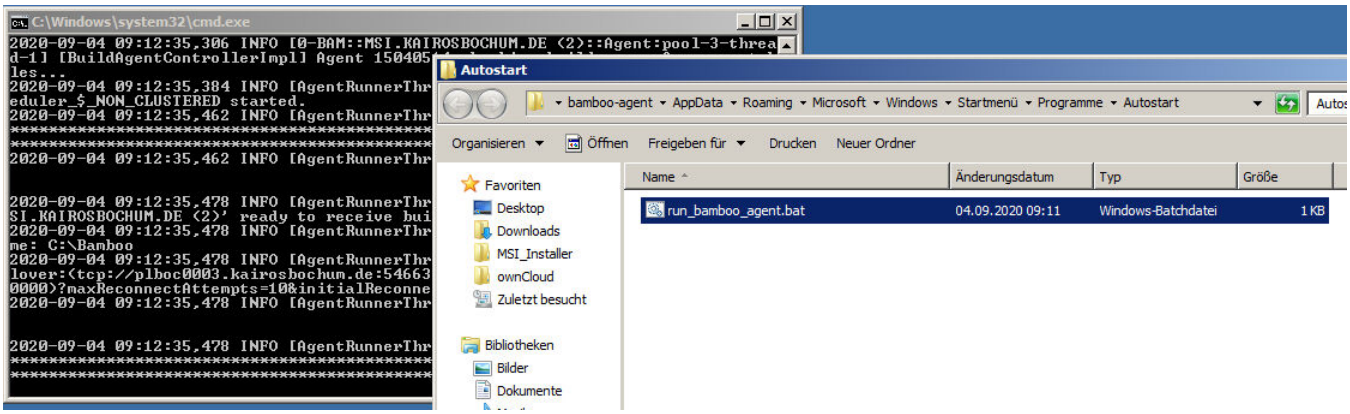


Running the build agent in user session

Our build agent must not run as service on the isolated session 0 but in a user session. A Bambo build agent can be started with the following command line. So we just build a Batch file and copy it to the Windows Autostart folder.

run_bamboo_agent.bat

```
java -Xms256m -Xmx512m -Djava.library.path="C:/Bamboo/lib" -classpath "C:/Bamboo/lib/wrapper.jar;C:/Bamboo/lib
/bamboo-agent-bootstrap.jar" -Dbamboo.home="c:\Bamboo" -Dbamboo.agent.ignoreServerCertName=false -jar C:/Bamboo
/lib/bamboo-agent-bootstrap.jar http://<my_bamboo_server>/agentServer/
```



The agent service user has now to be logged on automatically after a server restart. This can be configured in the registry:

winlogon.reg

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]
"AutoAdminLogon"="1"
"DefaultUserName"="bamboo-agent"
"DefaultPassword"="your_secret"
"DefaultDomainName"="domain.de"
```

> **A security hole big enough to push Jupiter through**
>
> Now that we have a user with Administrator rights always logged on to the build agent, this is not the best solution from a security perspective.

Configuring Advanced Installer Project

At the end we have to tell Advanced Installer to use a signtool.exe with a custom command line instead of the built in tool. The path to the signtool.exe is relative to the .aip file.