true-Sign V

Configuration Reference

V4.1 - 12/2023

Swiss IT Security AG
Etzelmatt 3
5430 Wettingen
Schweiz

+41 848 088 088
www.sits.ch
info@sits.ch

Copyright © 2023 by Swiss IT Security AG

Trademark Notice

Contents

1 Overview ... 9
1.1 true-Sign V registry configuration layout ... 9
2 true-Sign V Application ... 10
2.1 Configuration locations ... 10
2.2 Application Settings ... 10
2.3 Application Theme Colors ... 20
3 Service Provider configuration ... 24
3.1 Configuration locations ... 24
3.2 Service Provider settings ... 25
3.2.1 Disable configuration entry override ... 30
3.2.2 Card reader images ... 31
3.3 Provider GUI texts ... 32
3.4 Authentication specific configuration ... 34
3.4.1 OTP ... 34
3.4.1.1 OTP GUI Texts ... 35
3.4.2 Kerberos ... 37
3.4.3 Certificate ... 38
3.4.4 Browser ... 40
3.4.4.1 Browser GUI Texts ... 41
3.4.4.2 Browser Features ... 41
3.4.4.3 Browser GUI ... 43
3.4.4.4 Windows domain specific User Agents ... 46
3.4.4.5 Allowed navigation domains ... 47
3.4.4.6 Proxy settings ... 48
3.4.5 OAuth2 ... 49
3.4.5.1 Using the internal browser for authentication ... 53
3.4.5.2 Using an external browser for authentication ... 56
3.4.5.3 OAuth2 GUI Texts ... 58
3.5 API specific configuration ... 60
3.5.1 CSC_1_0_4 (Cloud Signature Consortium API V1.0.4.0) ... 60
3.5.1.1 OTP dialog (offline OTP) ... 63
3.5.1.2 OTP dialog (online OTP) ... 64
3.5.1.3 PIN dialog ... 65

3.5.1.4 Implicit authorization dialog ... 66
3.5.1.5 Certificate type configuration ... 66

4 true-Sign V Cert Store Provider configuration ... 70

4.1 Configuration locations ... 70
4.2 Default Application Profile ... 70
4.3 Application Profiles ... 71
4.3.1 Application version specific settings ... 72
4.3.1.1 Version matching ... 73
4.3.1.2 Order of checks when multiple version specific configurations are available ... 74
4.3.1.3 Certificate specific provider configurations ... 74
4.4 Certificate Filters ... 75
4.5 Crypto Providers ... 76
4.6 Custom machine store support ... 76

Appendix A: Service Configuration ... 77

A.1 Service configuration entries ... 77
A.2 Security Flags ... 79

Appendix B: Auto Provision / Refresh Configuration ... 80

B.1 Auto provision / refresh configuration entries ... 80

Appendix C: Certificate Filter Definition ... 81

C.1 LDAP Style Filter Definition ... 82
C.2 Legacy Filter Definition ... 94

Appendix D: true-Sign V Providers ... 97

Appendix E: true-Sign V Protocol Handler ... 103

Tables

Table 1: true-Sign V application configuration locations 10

Table 2: true-Sign V application configuration 19

Table 3: true-Sign V theme configuration location 22

Table 4: true-Sign V theme configuration 23

Table 5: Provider configuration locations 24

Table 6: Provider configuration 30

Table 7: Provider disable override location 30

Table 8: Installed card reader images 31

Table 9: Provider GUI texts 33

Table 10: OTP authentication settings 35

Table 11: OTP GUI texts 36

Table 12: Kerberos authentication settings 37

Table 13: Certificate authentication settings 38

Table 14: Browser authentication settings 41

Table 15: Browser GUI texts 41

Table 16: Browser features 43

Table 17: Browser GUI 45

Table 18: Browser user agents 46

Table 19: Allowed navigation domains 47

Table 20: Proxy settings for the WebView2 browser 48

Table 21: OAuth2 authentication settings 53

Table 22: OAuth2 GUI texts 58

Table 23: CSC settings 62

Table 24: Implicit authorization text 63

Table 25: Certificate type settings 68

Table 26: Certificate type icons 68

Table 27: Certificate type GUI texts 69

Table 28: true-Sign V Cert Store Provider configuration locations 70

Table 29: Default application profile configuration settings 70

Table 30: Application profile generic configuration settings 72

Table 31: Version specific application profile configuration settings 73

Table 32: Version comparison operators 74

Table 33: Certificate specific provider configuration settings 75
Table 34: Certificate Filter configuration elements 75
Table 35: Crypto Provider Aliases 76
Table 36: Service configuration entries 78
Table 37: Service security flags 79
Table 38: Auto provisioning and refresh configuration 80
Table 39: Quoted characters for certificate filters 82
Table 40: Operators for certificate filters 84
Table 41: Session variables for use in certificate filters 85
Table 42: Available attributes and allowed operators for certificate filters 86
Table 43: Legacy certificate filter types 94
Table 44: Legacy certificate filter operator instructions 95
Table 45: Valid legacy certificate filter values 96
Table 46: Cryptographic provider names and types 97
Table 47: Cryptographic providers supported by OS 97
Table 48: Cryptographic provider supported by applications 98
Table 49: Cryptographic provider supported by runtime environments 99
Table 50: Cryptographic provider supported by development tools 100
Table 51: Signature algorithm support 101
Table 52: Encryption algorithm support 102
Table 53: URI parameters for OAuth2 authentication 104

Figures

Figure 1: true-Sign V registry configuration layout 9
Figure 2: Account list dialog width configurations 20
Figure 3: Provider and certificate tiles in dialogs 20
Figure 4: Active certificate tile 21
Figure 5: Inactive certificate tile and active provider tile 21
Figure 6: Disabled certificate and provider tiles 22
Figure 7: Provider configuration keys 24
Figure 8: Provider configuration sub-keys 25
Figure 9: Provider language sub keys 32
Figure 10: Provider name and description 33
Figure 11: Password tool tip 33
Figure 12: Password tool tip not set 33
Figure 13: OTP sub-key 34
Figure 14: Provider language sub keys 35
Figure 15: OTP hints and tool tips 36
Figure 16: Multiple authentication certificates available 39
Figure 17: No suitable authentication certificate available 39
Figure 18: Browser configuration sub keys 40
Figure 19: Proxy authentication in browser 49
Figure 20: OAuth2 configuration sub keys 49
Figure 21: Missing Microsoft Edge WebView2 runtime error 53
Figure 22: Download link for Microsoft Edge WebView2 runtime 54
Figure 23: Task Manager processes with internal browser 54
Figure 24: Task Manager WebView2 process details 55
Figure 25: External authentication notification 56
Figure 26: External browser MIME handler confirmation 56
Figure 27: MIME handler authentication error 57
Figure 28: Provider language sub keys 58
Figure 29: OAuth2 hints and tool tips 59
Figure 30: CSC sub-key 60
Figure 31: CSC offline OTP dialog elements 63
Figure 32: CSC online OTP dialog elements 64

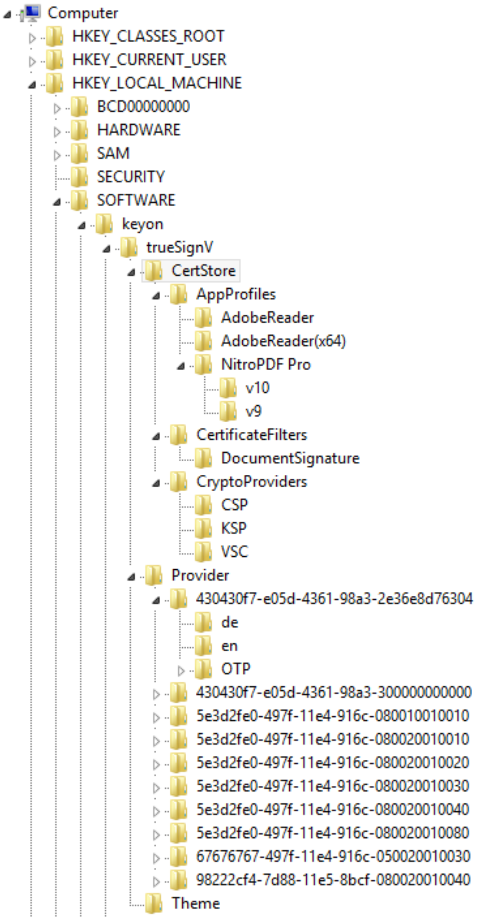
Figure 33: CSC PIN dialog elements 65
Figure 34: CSC implicit authorization dialog 66
Figure 35: CSC cert tile elements 66
Figure 36: Application Profile Keys 71
Figure 37: Version specific application profile keys 72
Figure 38: Executable file version properties 73
Figure 39: Certificate specific provider configurations 75
Figure 40: Certificate Filter Keys 75
Figure 41: Crypto Provider Keys 76
Figure 42: Add Account dialog triggered by protocol handler error 103
Figure 43: Protocol handler error when provider is unknown 104

1 Overview

true-Sign V is configured using registry setting. The settings cover three main areas:

- **true-Sign V Application**
  - Application settings
  - Theme colors
- **Service Provider**
  - Description and GUI texts
  - Service connection information
  - Authentication settings
- **true-Sign V Certificate Store Provider**
  - Application profiles
  - Certificate filters

1.1 true-Sign V registry configuration layout



```
Computer
  HKEY_CLASSES_ROOT
  HKEY_CURRENT_USER
  HKEY_LOCAL_MACHINE
    BCD00000000
    HARDWARE
    SAM
    SECURITY
    SOFTWARE
      keyon
        trueSignV
          CertStore
            AppProfiles
              AdobeReader
              AdobeReader(x64)
              NitroPDF Pro
                v10
                v9
            CertificateFilters
              DocumentSignature
            CryptoProviders
              CSP
              KSP
              VSC
          Provider
            430430f7-e05d-4361-98a3-2e36e8d76304
              de
              en
              OTP
            430430f7-e05d-4361-98a3-300000000000
            5e3d2fe0-497f-11e4-916c-080010010010
            5e3d2fe0-497f-11e4-916c-080020010010
            5e3d2fe0-497f-11e4-916c-080020010020
            5e3d2fe0-497f-11e4-916c-080020010030
            5e3d2fe0-497f-11e4-916c-080020010040
            5e3d2fe0-497f-11e4-916c-080020010080
            67676767-497f-11e4-916c-050020010030
            98222cf4-7d88-11e5-8bcf-080020010040
          Theme
```

Figure 1: true-Sign V registry configuration layout

2 true-Sign V Application

2.1 Configuration locations

The following locations are searched for the true-Sign V application configuration:

Order	Root Key	true-Sign V Configuration Locations
1	`HKEY_LOCAL_MACHINE`	`SOFTWARE\keyon\trueSignV`
2	`HKEY_LOCAL_MACHINE`	`SOFTWARE\Policies\keyon\trueSignV`
3	`HKEY_CURRENT_USER`	`SOFTWARE\keyon\trueSignV`
4	`HKEY_CURRENT_USER`	`SOFTWARE\Policies\keyon\trueSignV`

Table 1: true-Sign V application configuration locations

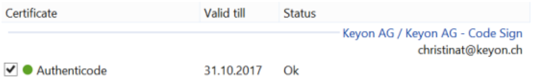
ⓘ Entries in a higher order location will overwrite entries in lower order locations if they share the same name.

2.2 Application Settings

The following table shows the configuration entries for the true-Sign V Application:

Name	Type	Description
`AccountDialogWidth`	`DWORD`	The width of the account list dialog in pixels for a 96 DPI display. See Figure 2. If `0`, a default width is used. If non-zero, the actual width in pixels is calculated for the DPI settings of the display where the account list dialog is shown and both minimum and maximum width restrictions are applied to ensure that the dialog does not exceed the width of the display and the list and buttons are always shown. Default: `0x00000000`
`AccountListCertColWidth`	`DWORD`	The width of the status column in the account list dialog in pixels for a 96 DPI display. See Figure 2. If `0`, a default width is used. If non-zero, the actual width in pixels is calculated for the DPI settings of the display where the account list dialog is shown. Default: `0x000000C8`

Name	Type	Description
`AccountListStatus ColWidth`	`DWORD`	The width of the status column in the account list dialog in pixels for a 96 DPI display. See Figure 2. If `0`, the width is calculated automatically to use the available space after applying certificate and validity column width. If non-zero, the actual width in pixels is calculated for the DPI settings of the display where the account list dialog is shown. Default: `0x00000000`
`AccountListValidity ColWidth`	`DWORD`	The width of the status column in the account list dialog in pixels for a 96 DPI display. See Figure 2. If `0`, a default width is used. If non-zero, the actual width in pixels is calculated for the DPI settings of the display where the account list dialog is shown. Default: `0x00000050`
`AllowCryptUIOnline Lookups`	`DWORD`	Enables online revocation checks for certificates in the certificate's details dialog. Note that this may introduce delays when opening the certificate details dialog. Default: `0x00000000` (false)

`AllowDisabling Certificates`	`DWORD`	When set to a non-zero value, certificate entries have a checkbox on the left allowing a user to disable the certificate:
		
		Certificate Valid till Status
		Keyon AG / Keyon AG - Code Sign christinat@keyon.ch
		☑ ● Authenticode 31.10.2017 Ok
		Disabled certificate are not available for use and are not propagated to the user's certificate store.
		The state settings are saved in the registry under
		`HCKU\Software\keyon\trueSignV\`
		`DisabledCertificates`
		Default: `0x00000000` (false)

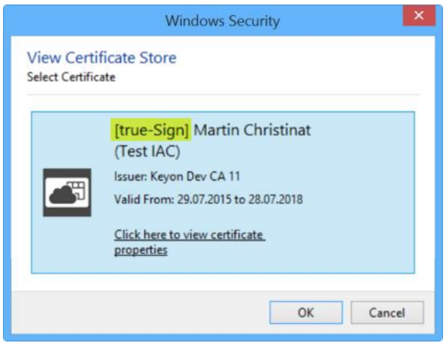
`ApplicationTitle`	`REG_SZ`	The application title to use for application windows and dialogs. Default: `true-Sign V`
`EnableEdgeLegacy Support`	`DWORD`	Enables support for legacy Edge browser. Default: `0x00000000` (false)

Name	Type	Description
`EnableLogSettingsInAbout`	`DWORD`	Enables log settings context menu in the about dialog. Default: `0x00000001` (true)
`EnableMachineStoreSupport`	`DWORD`	Enables support for use of the certificates in the machine context. Note that this feature requires a special registration of the CSP and KSP and is only supported when a single true-Sign V instance is running on the system. The configuration entry `MachineStoreAllowedAccounts` specifies which additional local or domain accounts or groups can use certificates in the machine context. Please note that this is a feature that is likely only used by software developers in certain scenarios and not by ordinary business users. Default: `0x00000000` (false) Note that the KSP, CSP and CertStore provider will read this setting only from HKLM`\SOFTWARE\keyon\trueSignV` and not the other configuration locations.

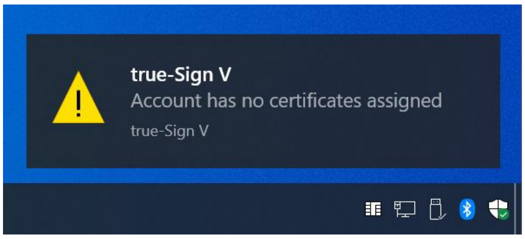
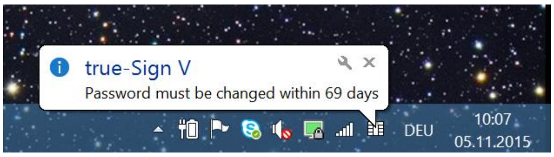
`ExcludedProcesses`	`REG_SZ`
	or
	`MULTI_SZ`
		List of full process paths of local processes that cannot use keys and in case of the Virtual Smart Card provider will see only an empty Smart Card (comma separated if type is `REG_SZ`).
		The process path can contain environment variables enclosed in `%` characters such as `%SystemRoot%`.
		Sample:
		`%SystemRoot%\system32\svchost.exe`
		Default: *not set*

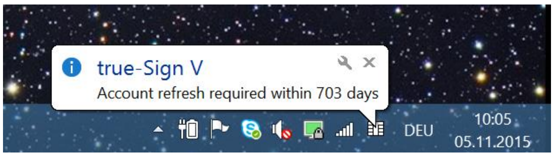
Name	Type	Description

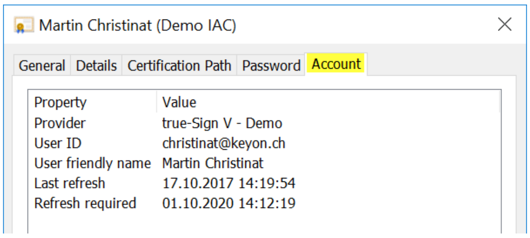
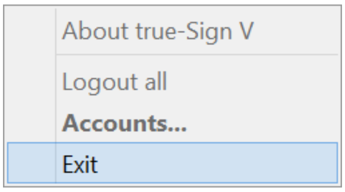
`ExcludedRemoteProcesses`	`REG_SZ`
	or
	`MULTI_SZ`
		List of full process paths of remote processes that will see only an empty Smart Card (comma separated if type is `REG_SZ`).
		The process path can contain environment variables enclosed in `%` characters such as `%SystemRoot%`. Note however that the remote system may use different environment variables.
		Sample:
		`%SystemRoot%\system32\svchost.exe`
		Default: *not set*

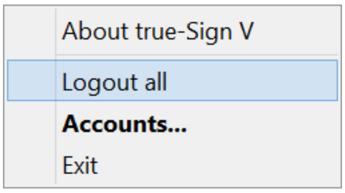
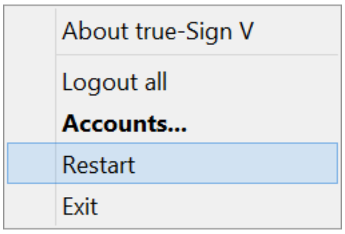
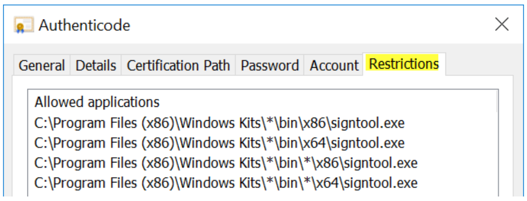
`FriendlyNamePrefix`	`REG_SZ`	Defines the friendly name prefix set in the certificate store for the certificate. Note that the provider configuration may define its own friendly name prefix, which will overwrite this value.  View Certificate Store Select Certificate [true-Sign] Martin Christinat (Test IAC) Issuer: Keyon Dev CA 11 Valid From: 29.07.2015 to 28.07.2018 Click here to view certificate properties The friendly name is used by Windows in certain certificate selection dialogs, but it is up to the application to make use of the friendly name or not. Default: *not set*
`HidePasswordChangeButtonUnlessApplicable`	`DWORD`	When set to a non-zero value, hides the *Change Password* button in the account dialog, unless the selected certificate supports changing the password. Default: `0x00000000` (false)

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>LogLevel</td><td>DWORD</td><td>Defines the log level for the true-Sign V application log (see LogFile). The following log levels are available:
<table>
<tr><td>0</td><td>None</td></tr>
<tr><td>1</td><td>Error</td></tr>
<tr><td>2</td><td>Warning</td></tr>
<tr><td>3</td><td>Info</td></tr>
<tr><td>4</td><td>Debug</td></tr>
<tr><td>5</td><td>Trace</td></tr>
</table>
Any events with level an equal or lower are logged.
Default: 0x00000000 (none)</td></tr>
<tr><td>LogFile</td><td>REG_SZ</td><td>The full log file name. The log file path may contain environment variables enclosed in % characters such as %AppData%.
Default: <i>not set</i></td></tr>
<tr><td>LogMaxGenerations</td><td>DWORD</td><td>The number of rotated log files to keep if logs are rotated by size or age. If set to a value > 0, the oldest rotated logs are deleted to ensure that only the specified number of rotated log files are kept.
If not set or 0, rotated logs will not be deleted.
Default: <i>not set</i></td></tr>
<tr><td>LogRotateAge</td><td>DWORD</td><td>Rotate the log file at startup if the current log is older than (i.e. was created before) the number of days specified.
If not set or 0, no rotation based on age will occur.
If the log is rotated, a timestamp of the form _YYYYMMDDHHmm is added to the filename.
Default: <i>not set</i></td></tr>
<tr><td>LogRotateSize</td><td>DWORD</td><td>Rotate the log file at startup if the current log file size exceeds the number of bytes specified.
If not set or 0, no rotation based on size will occur.
If the log is rotated, a timestamp of the form _YYYYMMDDHHmm is added to the filename.
Default: <i>not set</i></td></tr>
</table>

Name	Type	Description
`LogoutOnLock`	`DWORD`	When set to a non-zero value, a workstation lock will discard any cached authentication information and certificate passwords. The user will need to authenticate himself (e.g. using OTP) and provide the certificate password for the next crypto operation. Default: `0x00000000` (false)
`MachineStoreAllowedAccounts`	`REG_SZ` or `MULTI_SZ`	List of local and domain accounts or groups that can use keys in the machine context in addition to the user under which true-Sign V is running (comma separated if type is `REG_SZ`). Note that `EnableMachineStoreSupport` must be enabled and the CSP and KSP must be registered for machine store support. Default: *not set*
`NotifyEmptyAccount`	`DWORD`	Show a notification with a tray area balloon if an account has no certificates assigned.  true-Sign V Account has no certificates assigned true-Sign V If set to `0`, no notification will be shown to inform the user of an empty account. Default: `0x00000000` (no notification)
`NotifyPINChangeDaysBefore`	`DWORD`	The time in days during which to notify the user with a tray area balloon at startup, workstation unlock and once every hour before a PIN change is required.  true-Sign V Password must be changed within 69 days If set to `0`, no notification will be shown to inform the user of the due password change. Default: `0x00000000` (no advanced notification)

Name	Type	Description
`NotifyRefreshDaysBefore`	`DWORD`	The time in days during which to notify the user with a tray area balloon at startup, workstation unlock and once every hour before an account refresh is required.  true-Sign V Account refresh required within 703 days If set to `0`, no notification will be shown to inform the user of the due account refresh. Default: `0x00000000` (no advanced notification)
`PositionRelativeToTray Area`	`DWORD`	When set to a non-zero value, the account dialog is positioned next to the tray area instead of shown centered on the screen. The relative positioning works as expected even if the taskbar is docked on top or on the side of the screen. Default: `0x00000000` (false)
`ProviderClientTimeout`	`DWORD`	The timeout in milliseconds which a provider (CertStore, CSP or KSP) waits for the true-Sign V application to respond to a request. Note that this timeout is not applied when true-Sign V is waiting for user input e.g. for user authentication or entering the certificate password. Default KSP and CSP: `0x00001388` (5000) Default CertStore: `0x000007D0` (2000) Note that this setting is only supported under `HKLM\SOFTWARE\keyon\trueSignV` and not the other configuration locations.

Name	Type	Description
`SaveAccountDialogPos`	`DWORD`	When set to a non-zero value, the width of the account list dialog and the widths of the list control columns are saved in the registry when the dialog is closed and restored when the dialog is opened the next time. The settings are saved individually for the display resolution and DPI settings of the display where the dialog is shown under `HCKU\Software\keyon\trueSignV\AccountDialog` Note that the height of the dialog is not saved and always calculated dynamically. Default: `0x00000000` (false)
`ShowAccountTab`	`DWORD`	Shows the account tab in the certificate details dialog:  Martin Christinat (Demo IAC) General Details Certification Path Password Account Property: Value Provider: true-Sign V - Demo User ID: christinat@keyon.ch User friendly name: Martin Christinat Last refresh: 17.10.2017 14:19:54 Refresh required: 01.10.2020 14:12:19 Default: `0x00000001` (true)
`ShowExit`	`DWORD`	Shows an *Exit* menu entry in the tray icon menu when set to a non-zero value:  If the *Exit* menu entry is not shown, the user cannot exit true-Sign V unless he uses the task manager to kill the process. Default: `0x00000001` (true)

Name	Type	Description
`ShowLogoutAll`	`DWORD`	Shows a *Logout all* menu entry in the tray icon menu when set to a non-zero value:  Clicking the Logout all menu item will discard any cached authentication information and certificate passwords. The user will need to authenticate himself (e.g. using OTP) and provide the certificate password for the next crypto operation. Default: `0x00000001` (true)
`ShowRestart`	`DWORD`	Shows a *Restart* menu entry in the tray icon menu when set to a non-zero value:  About true-Sign V Logout all Accounts... Restart Exit Clicking the *Restart* menu item will exit and restart true-Sign V. This option is only useful when engineering new true-Sign V configurations to quickly see the effects of a configuration change. Default: `0x00000000` (false)
`ShowRestrictionsTab`	`DWORD`	Shows the restrictions tab in the certificate details dialog if the certificate use is restricted by policy:  Authenticode General Details Certification Path Password Account Restrictions Allowed applications C:\Program Files (x86)\Windows Kits*\bin\x86\signtool.exe C:\Program Files (x86)\Windows Kits*\bin\x64\signtool.exe C:\Program Files (x86)\Windows Kits*\bin*\x86\signtool.exe C:\Program Files (x86)\Windows Kits*\bin*\x64\signtool.exe Default: `0x00000001` (true)

Name	Type	Description

`SmartCardCommunicationGUID`	`REG_SZ`	Force the use of a specific transmission protocol for the Virtual Smart Card if installed. By default, true-Sign V uses `T=1` as it is more efficient than `T=0`. Certain usage scenarios may require `T=0` if a component does not support `T=1`. (Forwarding the Virtual Smart Card into a guest VM with VMWare Workstation is such an example.)
		To force the use of a specific transmission, set the configuration entry to one of the following GUID strings:
		T=1 `{FA8BC8BE-85D3-47c8-A5F7-4B8DE2C57C28}`
		T=0 `{C214F308-AFCF-49CC-9D3F-5B966BEF8F97}`
		Default:
		`{FA8BC8BE-85D3-47c8-A5F7-4B8DE2C57C28}`

`UILanguage`	`REG_SZ`	Force a specific user interface language to be used. By default, the primary user interface language of the logged in user is used by true-Sign V. By setting this entry, a specific language can be enforced. The following languages are currently supported: en English de German Default: *Primary user interface language of logged in user*

Table 2: true-Sign V application configuration

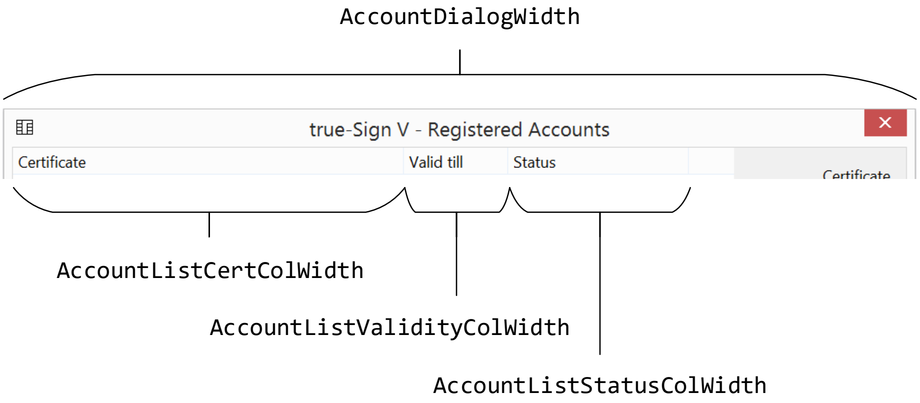

AccountDialogWidth
true-Sign V - Registered Accounts
Certificate
Valid till
Status
Certificate
AccountListCertColWidth
AccountListValidityColWidth
AccountListStatusColWidth

Figure 2: Account list dialog width configurations

2.3 Application Theme Colors

The theme configuration allows setting the colors of the certificate and provider tiles shown in the true-Sign V dialogs. The certificate tile (`CertTile`) contains information about the certificate to which the operation applies, and the provider tile (`ProviderTile`) contains information about the service provider:

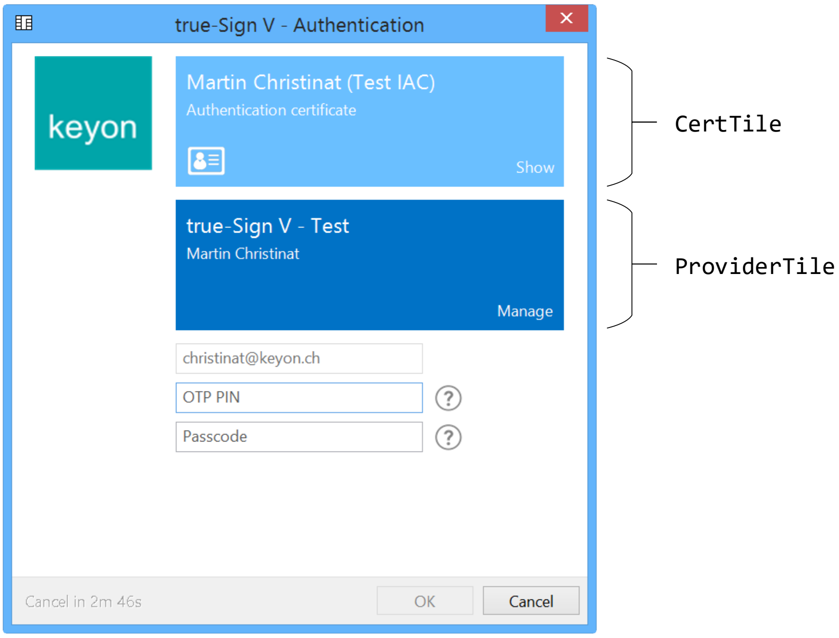
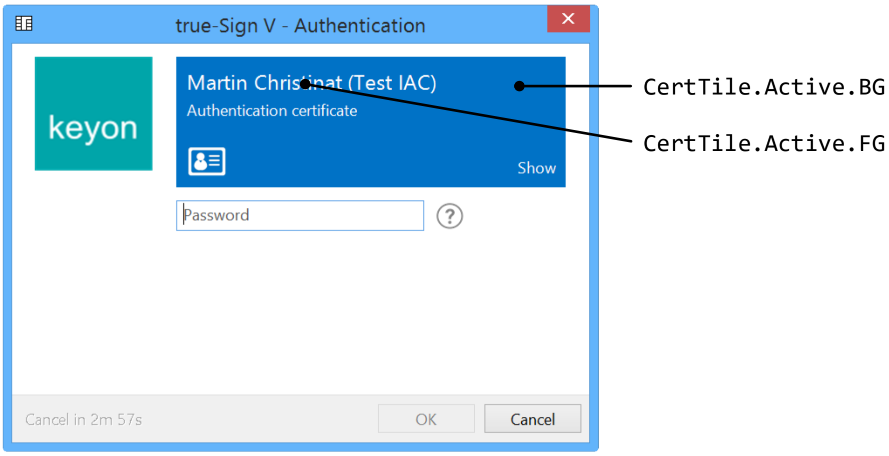


Figure 3: Provider and certificate tiles in dialogs

The foreground (text) and background colors can be set for the three possible states *active*, *inactive*, and *disabled*.

If only one tile is shown in a dialog, it will be considered *active*:

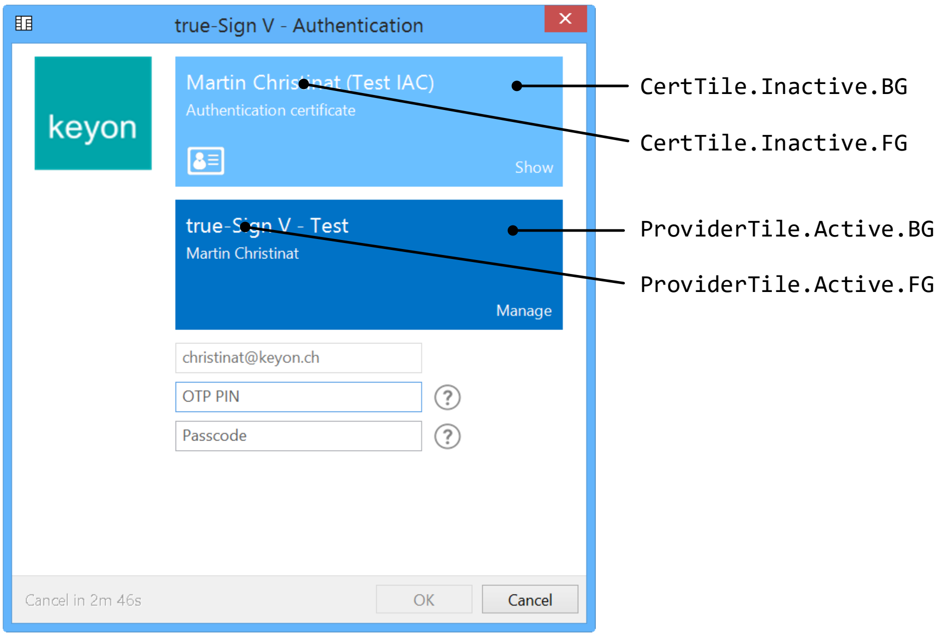

true-Sign V - Authentication
keyon
Martin Christinat (Test IAC)
Authentication certificate
Show
Password
Cancel in 2m 57s
OK
Cancel

CertTile.Active.BG

CertTile.Active.FG

Figure 4: Active certificate tile

If two tiles are shown in a dialog, one will be considered *active* and the other *inactive*. The *active* tile highlights the target of the operation, e.g. the service provider if the user must authenticate himself to the service. The inactive tile is provided for reference purposes, e.g. to show which certificate was requested by the operation triggering the dialog:



CertTile.Inactive.BG

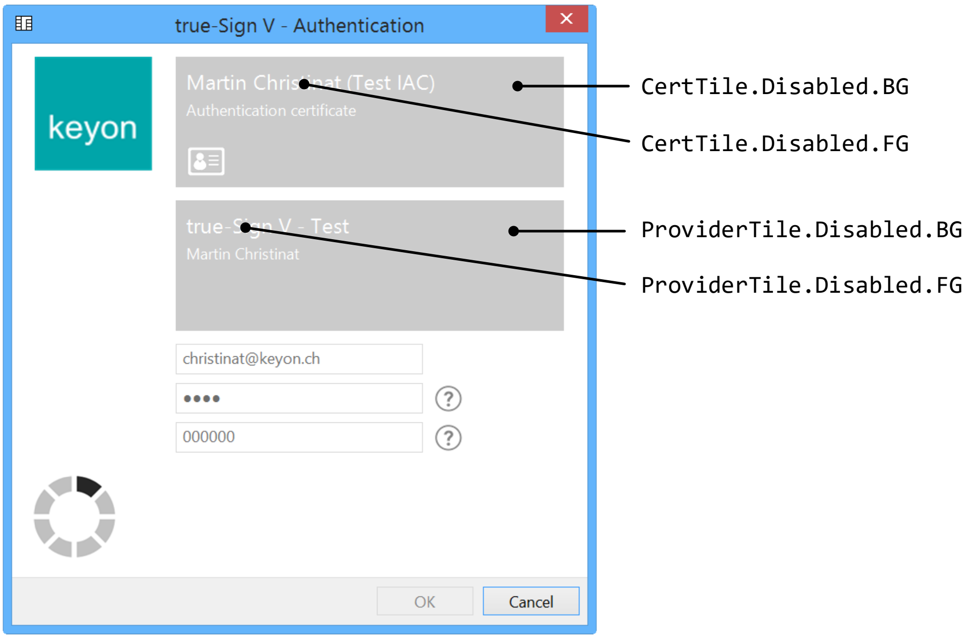
CertTile.Inactive.FG

ProviderTile.Active.BG

ProviderTile.Active.FG

Figure 5: Inactive certificate tile and active provider tile

The *disabled* state is only used during background operations to show to the user that an operation is in progress. If not defined, a standard grey color will be used (recommended):



true-Sign V - Authentication

Martin Christinat (Test IAC)
Authentication certificate

true-Sign V - Test
Martin Christinat

christinat@keyon.ch
••••
000000

Figure 6: Disabled certificate and provider tiles

Unlike other configuration entries, the theme configuration is only read from the following registry key:

Order	Root Key	true-Sign V Theme Configuration Key
1	`HKEY_LOCAL_MACHINE`	`SOFTWARE\keyon\trueSignV\Theme`

Table 3: true-Sign V theme configuration location

Theme colors are encoded in a `DWORD` value as follows:

`0xaarrggbb`

Where

aa	is the opacity (alpha channel). Must be set to `FF`.
rr	is the intensity for red
gg	is the intensity for green
bb	is the intensity for blue

The following table shows the configuration entries affecting the theme colors:

Name	Type	Description
`CertTile.Active.FG`	`DWORD`	Foreground (text) color of an active cert tile. Default: `0xFFFFFFFF`
`CertTile.Active.BG`	`DWORD`	Background color of an active cert tile. Default: `0xFF0072C6`

Name	Type	Description
`CertTile.Inactive.FG`	DWORD	Foreground (text) color of an inactive cert tile. Default: `0xFFFFFFFF`
`CertTile.Inactive.BG`	DWORD	Background color of an inactive cert tile. Default: `0xFF6ABFFF`
`CertTile.Disabled.FG`	DWORD	Foreground (text) color of a disabled cert tile. Default: `0xFFFFFFFF`
`CertTile.Disabled.BG`	DWORD	Background color of a disabled cert tile. Default: `0xCBCBCB`
`ProviderTile.Active.FG`	DWORD	Foreground (text) color of an active provider tile. Default: `0xFFFFFFFF`
`ProviderTile.Active.BG`	DWORD	Background color of an active provider tile. Default: `0xFF0072C6`
`ProviderTile.Inactive.FG`	DWORD	Foreground (text) color of a disabled provider tile. Default: `0xFFFFFFFF`
`ProviderTile.Inactive.BG`	DWORD	Background color of a disabled provider tile. Default: `0xFF6ABFFF`
`ProviderTile.Inactive.BG`	DWORD	Background color of an inactive provider tile. Default: `0xFF6ABFFF`
`ProviderTile.Disabled.FG`	DWORD	Foreground (text) color of a disabled provider tile. Default: `0xFFFFFFFF`
`ProviderTile.Disabled.BG`	DWORD	Background color of a disabled provider tile. Default: `0xCBCBCB`

Table 4: true-Sign V theme configuration

3 Service Provider configuration

3.1 Configuration locations

The following locations are searched for service provider configurations:

Order	Root Key	true-Sign V Provider Configuration Locations
1	`HKEY_LOCAL_MACHINE`	`SOFTWARE\keyon\trueSignV\Provider`
2	`HKEY_LOCAL_MACHINE`	`SOFTWARE\Policies\keyon\trueSignV\Provider`
3	`HKEY_CURRENT_USER`	`SOFTWARE\Policies\keyon\trueSignV\Provider`

Table 5: Provider configuration locations

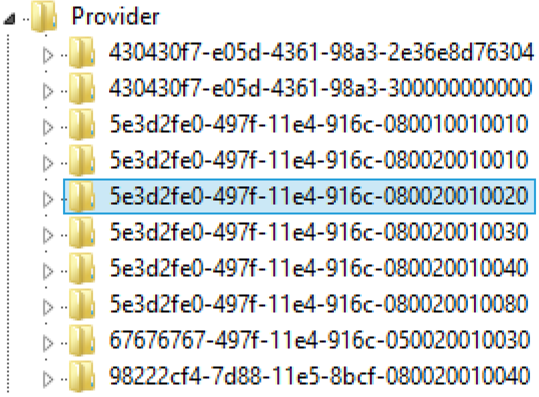
Configurations in a higher order location will overwrite configurations in a lower order location if they share the same provider GUID. Note that only complete configurations will be considered, i.e. it is not possible to overwrite only specific entries in a higher order location.

Each provider has its own configuration key sub-key under

```
…\trueSignV\Provider\
```

with a name in the form of a GUID string.

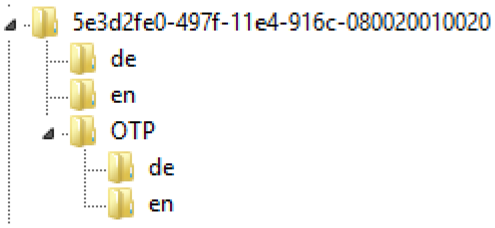
```
5e3d2fe0-497f-11e4-916c-080020010020
```



```
Provider
    430430f7-e05d-4361-98a3-2e36e8d76304
    430430f7-e05d-4361-98a3-300000000000
    5e3d2fe0-497f-11e4-916c-080010010010
    5e3d2fe0-497f-11e4-916c-080020010010
    5e3d2fe0-497f-11e4-916c-080020010020
    5e3d2fe0-497f-11e4-916c-080020010030
    5e3d2fe0-497f-11e4-916c-080020010040
    5e3d2fe0-497f-11e4-916c-080020010080
    67676767-497f-11e4-916c-050020010030
    98222cf4-7d88-11e5-8bcf-080020010040
```

Figure 7: Provider configuration keys

Under each provider key, there are-sub keys for the language dependent GUI configuration using two-digit language codes according to ISO 639-1 and specific settings for the authentication method used by the provider:

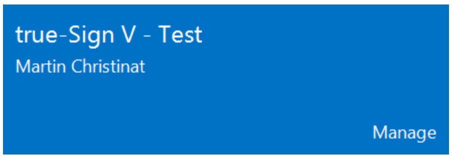


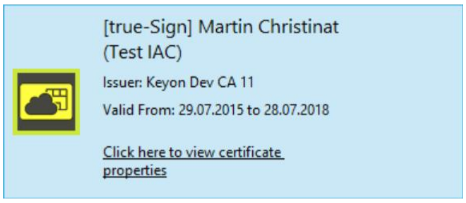
```
5e3d2fe0-497f-11e4-916c-080020010020
    de
    en
    OTP
        de
        en
```

Figure 8: Provider configuration sub-keys

3.2 Service Provider settings

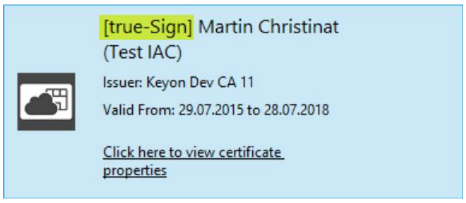
The following table shows the configuration entries for the service providers:

Name	Type	Description
`AccountManagementURL`	`REG_SZ`	The URL to open when the user clicks on the *Manage* link in the provider tile  true-Sign V - Test Martin Christinat Manage or the *Manage* button in the account list dialog. If not defined, the provider tile will not have a *Manage* link and the *Manage* button in the account list dialog will be disabled when a certificate managed by this provider is selected. Default: *not set*
`APIType`	`REG_SZ`	The backend API the provider uses. Can be one of: `SCS_SOAP` Standard true-Sign V backend `CSC_1_0_4` Cloud Signature Consortium API Default: `SCS_SOAP`
`APIVersion`	`DWORD`	The API version the backend supports. Default: `0x00000001`

Name	Type	Description
`AuthenticationType`	`REG_SZ`	The user authentication type for this provider. The type can be one of the following: `OTP` One-Time password-based authentication `Certificate` Client certificate-based authentication `Kerberos` Kerberos based authentication `Browser` Browser based authentication `OAUTH2` Browser based OAuth2 authentication The specific configuration for the authentication type is defined in a sub-key with the name of the authentication type.
`CardReaderImage`	`REG_SZ`	The card reader image to set for the certificate in the certificate store. The card reader image is show in Windows 8 or higher when the standard certificate selection dialog of Windows is used:  [true-Sign] Martin Christinat (Test IAC) Issuer: Keyon Dev CA 11 Valid From: 29.07.2015 to 28.07.2018 Click here to view certificate properties The path to the image must be relative to `%ProrgramData%` and the image must be a BMP image of dimensions `200` x `200` with 24-bit color depth plus alpha channel (i.e. 32-bit). Note that the image is not used for dialogs that read the certificate directly from the Virtual Smart Card, i.e. logon dialogs used e.g. for *Run as* or RDP logins to remote systems. See section 3.2.2 for images installed with true-Sign V. Default: *not set*
`Disabled`	`DWORD`	Disable this provider if set to a non-zero value. This option can be useful, e.g. to disable a provider for certain users using the Active Directory Group Policy. Default: `0x00000000` (false)

Name	Type	Description

`EnableOnlyForMembersOf`	`REG_SZ`
	or
	`MULTI_SZ`
		List of Active Directory group SAM account names (comma separated if type is `REG_SZ`). If present and not empty, the current user must be a member in one of the groups for the provider to be enabled.
		Sample:
		`TRUESIGN\G-L-TSV-CodeSigner`
		Default: *not set*

`FriendlyNamePrefix`	`REG_SZ`	Defines the friendly name prefix set in the certificate store for a certificate managed by this provider.  [true-Sign] Martin Christinat (Test IAC) Issuer: Keyon Dev CA 11 Valid From: 29.07.2015 to 28.07.2018 Click here to view certificate properties The friendly name may be used in certain certificate selection dialogs, but it is up to the application to make use of the friendly name or not. Default: *not set*
`OrderPreference`	`DWORD`	The order to use when sorting the providers in the add provider dialog drop down selection. Providers are sorted based on ascending `OrderPreference` and, if they use the same `OrderPreference`, based on their name. Default: `0x00010000`

Name	Type	Description
`PolicyIssuerCertificateFilter`	`REG_SZ`	A certificate filter specification for validating an issuer of the policy signer certificate. See *Appendix C: Certificate Filter Definition* for details on the filter definition language. This is not used for CSC API type. Note that if a trust provider DLL matching the GUID of the provider is present, the CA certificates present in the trust provider DLL will be used to validate the policy signer certificate and the contents of `PolicyIssuerCertificateFilter` are ignored. Only one of the issuers must match the filter. The issuer chain is built using the certificate store. User or machine store can be selected using `PolicySignerValidationPolicy`. Default: *not set*
`PolicySignerCertificateFilter`	`REG_SZ`	A certificate filter specification for validating the policy signer certificate. See *Appendix C: Certificate Filter Definition* for details on the filter definition language. This is not used for CSC API type. Note that if a trust provider DLL matching the GUID of the provider is present, the CA certificates present in the trust provider DLL will be used to validate the policy signer certificate and the contents of `PolicySignerCertificateFilter` are ignored. Default: *not set*

Name	Type	Description
`PolicySignerValidationPolicy`	`DWORD`	Defines additional validation requirements of the policy signer certificate in addition to match the `PolicyIssuerCertificateFilter` and the `PolicySignerCertificateFilter`. This is not used for CSC API type. The requirements are a bit mask of the following values: `0x80000000` Build and validate the chain using CryptoAPI. This is a base requirement for all other checks. `0x00000001` Use only the machine store for chain building and not the user store. `0x00000010` Check if all certificates in the chain are valid at the time of validation. `0x00000100` Check all certificates except the root certificate for revocation. Only if the revocation status is available and none of the certificates is revoked, the policy will be considered valid. Note that this may cause delays at startup and when adding providers if the revocation information is not available. Default: *not set*
`PolicyStore`	`REG_SZ`	The directory to use for storing downloaded policies. This directory must be located under a user dependent location such as the user profile. The directory is created by true-Sign V if it does not exist. The `PolicyStore` path may contain environment variables enclosed in `%` characters. Sample: `%APPDATA%\keyon\trueSignV` Default: *not set*
`Service*`		The service configuration for the crypto service. See *Appendix A: Service Configuration* for a detailed description of the service configuration options.

Name	Type	Description
TargetCertStoreName	REG_SZ	The name of the certificate store in which to store the user's certificates. Note that the user under which true-Sign V runs must have write access to this store. Default: MY if the true-Sign V Cert Store provider is not installed, trueSignV if it is installed.
TargetCertStoreType	DWORD	The type of the certificate store specified in TargetCertStoreName. Default: 0x00010000 (CURRENT_USER)
TargetProviderName	REG_SZ	See *Appendix D: true-Sign V Providers* for possible provider names. Note that the provider must be installed in order to be used. Default: Microsoft Base Smart Card Crypto Provider
TargetProviderType	REG_SZ	Default: 0x00000001 (PROV_RSA_FULL)

Table 6: Provider configuration

3.2.1 Disable configuration entry override

The following location is searched for a Disabled entry that will override the entry of the service provider configuration used if present:

Root Key	true-Sign V Provider disable override configuration Location
HKEY_CURRENT_USER	SOFTWARE\keyon\trueSignV\Provider\*{Provider GUID}*\

Table 7: Provider disable override location

This override mechanism allows a user to enable or disable configurations by setting a registry entry for which no administrative permissions are required.

3.2.2 Card reader images

Card reader images are only used in certain system certificate selection dialogs in Windows 8 or higher.

The following card reader images are installed by true-Sign V:

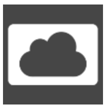
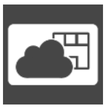
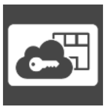
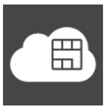
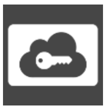
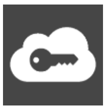
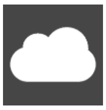
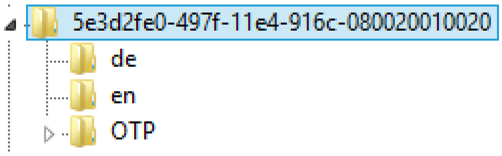
Image	Path
	`keyon\SmartCard\Reader\Images\trueSignVCloud.bmp`
	`keyon\SmartCard\Reader\Images\trueSignVCloudCard.bmp`
	`keyon\SmartCard\Reader\Images\trueSignVCloudCardKey.bmp`
	`keyon\SmartCard\Reader\Images\trueSignVCloudCardWhite.bmp`
	`keyon\SmartCard\Reader\Images\trueSignVCloudKey.bmp`
	`keyon\SmartCard\Reader\Images\trueSignVCloudKeyWhite.bmp`
	`keyon\SmartCard\Reader\Images\trueSignVCloudWhite.bmp`

Table 8: Installed card reader images

3.3 Provider GUI texts

Provider GUI texts are configured in sub keys under the provider configuration with the ISO 639-1 two-character primary language code, e.g. `en` for English or `de` for German as the sub key name.

```
…\trueSignV\Provider\5e3d2fe0-497f-11e4-916c-080020010020\en
```



5e3d2fe0-497f-11e4-916c-080020010020
de
en
OTP

Figure 9: Provider language sub keys

The GUI texts for `en` are always required since `en` is used as a fallback if no specific configuration for the user's primary display language is present.

Under each language specific sub-key, the following configuration elements are available:

Name	Type	Description
`Name`	`REG_SZ`	The name to show in the provider tile when adding a new account. See figure below.
`Description`	`REG_SZ`	The description to show in the provider tile when adding a new account. See figure below.
`DialogTitlePassword`	`REG_SZ`	A custom text to append to the application name in the password entry dialog instead of *Authentication*. Note that this text is not used when adding or refreshing an account. It is only used when an authentication dialog is shown as part of a signature or decryption operation.
`ToolTipPassword`	`REG_SZ`	The tool tip text to use for the password entry field. If `ToolTipPassword` is missing or empty, no tool tip icon will be shown. See figure below. You can use `\n` to insert line breaks.

Name	Type	Description
`ToolTipTransportPassword`	`REG_SZ`	The tool tip text to use in the change password dialog for the password entry field if the password was never changed after the initial generation by the service provider. If `ToolTipTransportPassword` is missing or empty, no tool tip icon will be shown. You can use `\n` to insert line breaks.

Table 9: Provider GUI texts

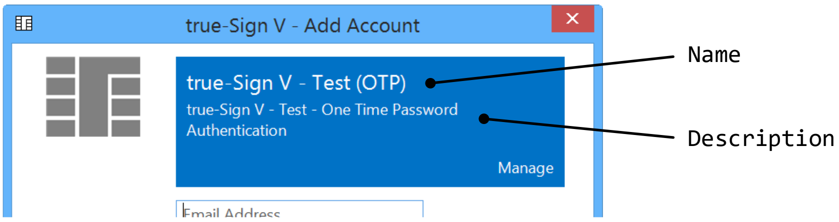
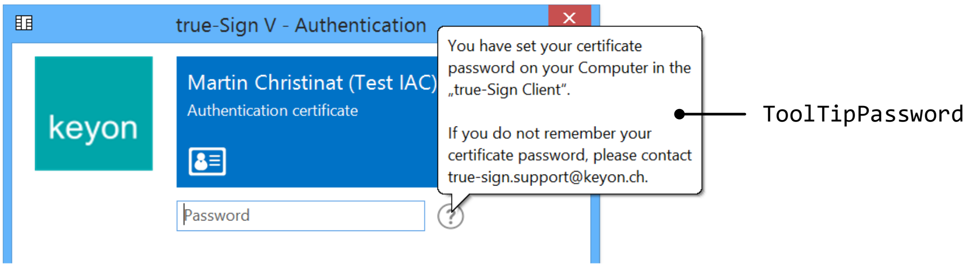

true-Sign V - Add Account
true-Sign V - Test (OTP)
true-Sign V - Test - One Time Password Authentication
Manage
Email Address
Name
Description

Figure 10: Provider name and description


true-Sign V - Authentication
keyon
Martin Christinat (Test IAC)
Authentication certificate
Password
ToolTipPassword

You have set your certificate password on your Computer in the „true-Sign Client“.

If you do not remember your certificate password, please contact true-sign.support@keyon.ch.

Figure 11: Password tool tip

If `ToolTipPassword` is not set or empty, the tool tip icon will not be shown:

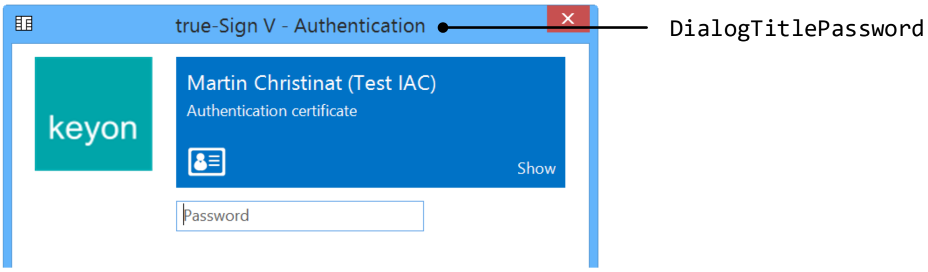

true-Sign V - Authentication
keyon
Martin Christinat (Test IAC)
Authentication certificate
Show
Password
DialogTitlePassword

Figure 12: Password tool tip not set

3.4 Authentication specific configuration

Depending on the type of user authentication the provider specifies in `AuthenticationType`, a distinct sub key is required:

`OTP`	One-Time password-based authentication
`Certificate`	Client certificate-based authentication
`Kerberos`	Kerberos based authentication
`Browser`	Browser based authentication
`OAuth2`	Browser based authentication with OAuth2

The authentication sub-key may have additional sub-keys, e.g. for language dependent GUI texts:

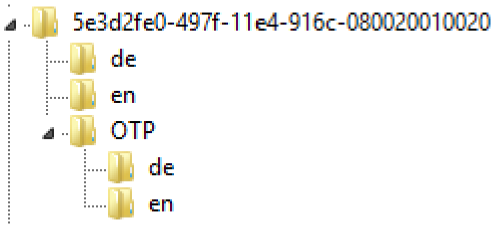

```
5e3d2fe0-497f-11e4-916c-080020010020
    de
    en
    OTP
        de
        en
```

Figure 13: OTP sub-key

3.4.1 OTP

If the provider specifies `OTP` as the `AuthenticationType`, the following settings are available under the `OTP` sub-key:

Name	Type	Description
`Service*`		The service configuration for the authentication service. See *Appendix A: Service Configuration* for a detailed description of the service configuration options.

Name	Type	Description

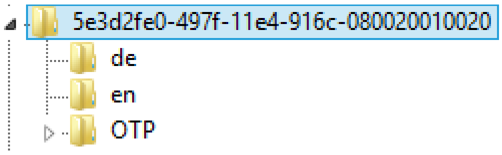
`ServiceParameters`	`MULTI_SZ`	The HTTP POST parameters used when authenticating the user. The parameters depend on the authentication service and are configured as key value pairs in the form key=value. The value part may contain one of the following variables within `{}` brackets:
		`user` The user id entered for authentication
		`otp` The OTP code entered for authentication
		`password` The password entered for authentication
		Sample:
		`VRAuthUsername={user}`
		`VRAuthOTP={otp}`
		`VRAuthPassword={password}`
		`VRAuthOrg=`

Table 10: OTP authentication settings

3.4.1.1 OTP GUI Texts

OTP GUI texts are configured in sub keys under the OTP configuration with the ISO 639-1 two-character primary language code, e.g. `en` for English or `de` for German as the sub key name.

`…\OTP\en`



5e3d2fe0-497f-11e4-916c-080020010020
de
en
OTP

Figure 14: Provider language sub keys

The GUI texts for `en` are always required since `en` is used as a fallback if no specific configuration for the user's primary display language is present.

Under each language specific sub-key, the following configuration elements are available:

Name	Type	Description
`DialogTitleAuthentication`	`REG_SZ`	A custom text to append to the application name in the OTP authentication dialog instead of *Authentication*.
`HintUserID`	`REG_SZ`	The hint to show as gray text in the empty user entry field.
`HintOTP`	`REG_SZ`	The hint to show as gray text in the empty OTP entry field.
`HintPassword`	`REG_SZ`	The hint to show as gray text in the empty password entry field.
`ToolTipUserID`	`REG_SZ`	The tool tip text to use for the user entry field. If `ToolTipUserID` is missing or empty, no tool tip icon will be shown next to the entry field. You can use `\n` to insert line breaks.
`ToolTipOTP`	`REG_SZ`	The tool tip text to use for the OTP entry field. If `ToolTipOTP` is missing or empty, no tool tip icon will be shown next to the entry field. You can use `\n` to insert line breaks.
`ToolTipPassword`	`REG_SZ`	The tool tip text to use for the password entry field. If `ToolTipPassword` is missing or empty, no tool tip icon will be shown next to the entry field. You can use `\n` to insert line breaks.

Table 11: OTP GUI texts

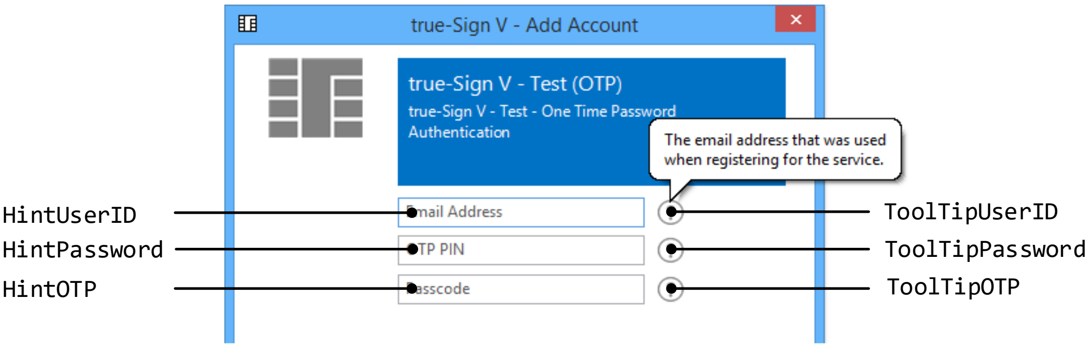

true-Sign V - Add Account
true-Sign V - Test (OTP)
true-Sign V - Test - One Time Password Authentication
The email address that was used when registering for the service.
Email Address
OTP PIN
Passcode
HintUserID
HintPassword
HintOTP
ToolTipUserID
ToolTipPassword
ToolTipOTP

Figure 15: OTP hints and tool tips

3.4.2 Kerberos

If the provider specifies `Kerberos` as the `AuthenticationType`, the following settings are available under the `Kerberos` sub-key:

Name	Type	Description
`Auto*`		The auto provision and refresh configuration for the authentication service. See Appendix B: Auto Provision / Refresh Configuration for a detailed description of the service configuration options.

Table 12: Kerberos authentication settings

ⓘ There are no additional GUI texts used with Kerberos authentication.

3.4.3 Certificate

If the provider specifies `Certificate` as the `AuthenticationType`, the following settings are available under the `Certificate` sub-key:

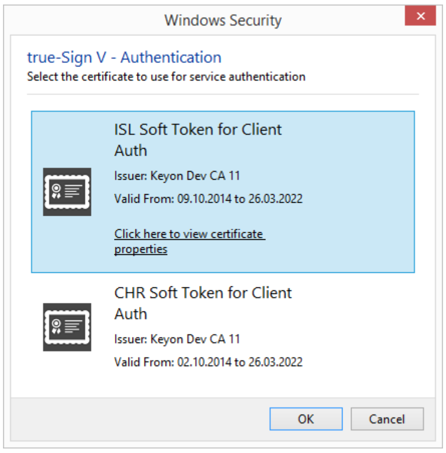
Name	Type	Description
`AuthenticationCertificateFilter`	`REG_SZ`	The certificate filter definition to use for selecting the authentication certificate. Only certificates matching this filter will be used for authentication. See *Appendix C: Certificate Filter Definition* for details on the filter definition language. Note that certificates provided by true-Sign V cannot be used for authentication as this could lead to an authentication loop. Default: *not set*
`Auto*`		The auto provision and refresh configuration for the authentication service. See Appendix B: Auto Provision / Refresh Configuration for a detailed description of the service configuration options.
`CertStoreName`	`REG_SZ`	The name of the certificate store in which to search the user's authentication certificate Default: `MY`
`CertStoreType`	`DWORD`	The type of the certificate store specified in `CertStoreName`. Default: `0x00010000` (CURRENT_USER)

Table 13: Certificate authentication settings

There are no additional GUI texts used with client certificate authentication.

You cannot use certificates provided by true-Sign V for service provider authentication as this could lead to an authentication loop.

If more than one certificate is available matching the certificate filter, a selection dialog is shown when you add the account or the first time after true-Sign V startup when the authentication for a provisioned account is required due to a crypto operation:



true-Sign V - Authentication

Select the certificate to use for service authentication

ISL Soft Token for Client Auth

Issuer: Keyon Dev CA 11

Valid From: 09.10.2014 to 26.03.2022

Click here to view certificate properties

CHR Soft Token for Client Auth

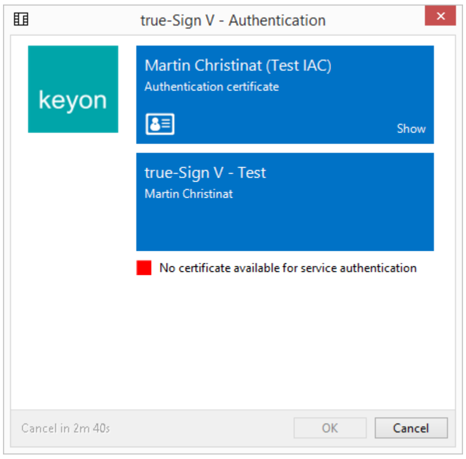
Issuer: Keyon Dev CA 11

Valid From: 02.10.2014 to 26.03.2022

Figure 16: Multiple authentication certificates available

If the authentication is successful, the selected certificate is remembered and used for subsequent authentications until true-Sign V exits.

If no certificate is available that matches the configured filter, an error will be shown:



Martin Christinat (Test IAC)

Authentication certificate

true-Sign V - Test

Martin Christinat

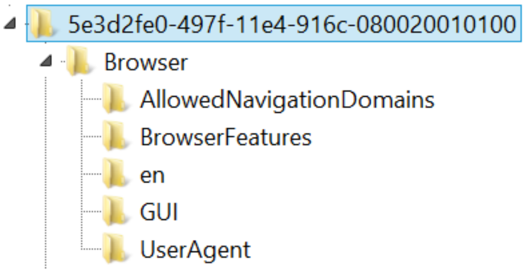
🟥 No certificate available for service authentication

Cancel in 2m 40s

Figure 17: No suitable authentication certificate available

3.4.4 Browser

If the provider specifies `Browser` as the `AuthenticationType`, the following settings and sub keys are available under the `Browser` sub-key:



- 5e3d2fe0-497f-11e4-916c-080020010100
  - Browser
    - AllowedNavigationDomains
    - BrowserFeatures
    - en
    - GUI
    - UserAgent

Figure 18: Browser configuration sub keys

The following settings are available directly under the `Browser` key:

Name	Type	Description
`AuthCookieName`	`REG_SZ`	The name of the authentication cookie that the authentication service sets after a successful authentication. Default: `AL_SESS-S`
`RegistrationURL`	`REG_SZ`	The URL to use when adding a new account using this provider. This URL can be different than the `ServiceURL` and may contain the following variables that are replaced before opening the URL in the embedded browser: `{upn}` The UPN of the logged in user `{upnName}` The user part of the UPN of the logged in user `{upnDomain}` The domain part of the UPN of the logged in user `{samName}` The SAM account name of the logged in user `{samDomain}` The SAM domain of the logged in user
`Service*`		The service configuration for the authentication service. See *Appendix A: Service Configuration* for a detailed description of the service configuration options. The `ServiceURL` may contain the variables described for the `RegistrationURL` and can contain the following additional variable: `{userid}` The user id from the account policy

Name	Type	Description
`TrySilentFirst`	`DWORD`	Try the authentication using the URL without showing a browser window in case an integrated authentication mechanism can be used and user interaction is only required as an exception. Default: `0x00000000` (false)

Table 14: Browser authentication settings

3.4.4.1 Browser GUI Texts

Browser GUI texts are configured in sub keys under the Browser configuration with the ISO 639-1 two-character primary language code, e.g. `en` for English or `de` for German as the sub key name.

```
…\Browser\en
```

! The GUI texts for `en` are always required since `en` is used as a fallback if no specific configuration for the user's primary display language is present.

Under each language specific sub-key, the following configuration elements are available:

Name	Type	Description
`DialogTitleAuthentication`	`REG_SZ`	A custom text to append to the application name in the Browser authentication dialog instead of *Authentication*.

Table 15: Browser GUI texts

3.4.4.2 Browser Features

Browser features are configured under the `BrowserFeatures` sub key:

```
…\Browser\BrowserFeatures
```

Name	Type	Description
`AllowFormAutofill`	`DWORD`	Allow automatic form filling of information such as passwords. For security reasons it is not recommended to enable this setting. Default: `0x00000000` (no)
`AllowNewWindows`	`DWORD`	Allow a page or link to open a new browser window. If disabled, the requested content is opened in the existing WebView2 window. Default: `0x00000000` (no)

Name	Type	Description
`AllowPasswordAutosave`	`DWORD`	Allow saving and automatic fill in of usernames and passwords, including proxy credentials. For security reasons it is not recommended to enable this setting. Default: `0x00000000` (no)
`AllowScripts`	`DWORD`	Allow Javascript on a page. Note that almost all modern web sites require Javascript support. Default: `0x00000001` (yes)
`AllowScriptDialogs`	`DWORD`	Allow Javascript to show notification dialogs. Default: `0x00000001` (yes)
`AllowSingleSignOnUsingOSPrimaryAccount`	`DWORD`	Allow to use the primary OS account for login. This includes Kerberos and Azure AD credentials and allows to implement SSO. Default: `0x00000000` (no)
`AutoCloseOnError`	`DWORD`	Automatically close the WebView2 window in case of an error response (HTTP status code >= 400) after the given time in milliseconds. If set to 0, the browser window must be closed by the user. Default: `0x00001388` (yes, after 5 seconds)
`BrowserArguments`	`REG_SZ`	Command line arguments to pass to the Microsoft Edge WebView2 instance. Note that invalid arguments may cause incorrect behavior. Default: *not set*
`EnableScrollbars`	`DWORD`	Enable scrollbars in the WebView2 window. Default: `0x00000000` (no)
`EnableStatusBar`	`DWORD`	Enable the Chromium loading status overlay in the lower left corner of the WebView2 window. Default: `0x00000001` (yes)
`ShowDevTools`	`DWORD`	Open the WebView2 development tools window whenever a WebView2 window is opened. This allow to debug login page related issues. Default: `0x00000000` (no)

Name	Type	Description
`ShowWaitPage`	`DWORD`	Show the built in wait page before navigating to the authorize URI. This prevent showing a blank page when the authorize web server is not responsive. Default: `0x00000001` (no)
`OpenUnacceptedDomainsExternal`	`DWORD`	Open links that are not in the list of Allowed navigation domains (0) externally using the system default browser is enabled (1). If disabled (0), opening links that point to a site not in the list of external domains will be denied and an error page is shown instead. Default: `0x00000000` (no)
`PersistCookies`	`DWORD`	Persist cookies set in WebView2. If cookies are not persisted, any settings made during the authentication process will be discarded when the WebView2 window is closed. Default: `0x00000001` (yes)
`Zoom`	`DWORD`	The zoom factor for the WebView2 content in percent. Default: `0x00000064` (100%)

Table 16: Browser features

3.4.4.3 Browser GUI

The browser GUI is configured under the `GUI` sub key.

```
…\Browser\GUI
```

Name	Type	Description
`CX`	`DWORD`	The width of the WebView2 window in dialog base units. Dialog base units are translated to actual pixels using the current DPI settings. Default: `0x00000046` (70)
`CY`	`DWORD`	The height of the WebView2 window in dialog base units. Dialog base units are translated to actual pixels using the current DPI settings. Default: `0x00000032` (50)

Name	Type	Description
`OffsetX`	`DWORD`	The horizontal offset of the WebView2 window in dialog base units relative to its authentication parent window. If set to `0`, the WebView2 window is centered over the parent authentication window. Default: `0x00000000` (0)
`Resizable`	`DWORD`	Allow to resize the WebView2 window (`1`) or not (`0`). Default: `0x00000001` (true)
`TitleDecorationAdd`	`REG_SZ`	The optional title decoration to add to the browser window when a new account is added. The following strings are replaced when creating the decoration: Subkey `{provider_name}` The name of the provider Default: *not set*
`TitleDecorationRefresh`	`REG_SZ`	The optional title decoration to add to the browser window when an account is refreshed. The following strings are replaced when creating the decoration: `{provider_name}` The name of the provider `{user_id}` The id of the user `{friendly_name}` The friendly name of the user Default: *not set*

Name	Type	Description
`TitleDecorationUse`	`REG_SZ`	The optional title decoration to add to the browser window when a crypto operation is executed. The following strings are replaced when creating the decoration: `{provider_name}` The name of the provider `{user_id}` The id of the user `{friendly_name}` The friendly name of the user `{cert_friendly_name}` The friendly name of the certificate Default: *not set*

Table 17: Browser GUI

3.4.4.4 Windows domain specific User Agents

Configure the user agent to set depending on the current DNS domain of the computer running true-Sign V. Setting a specific user agent allows to prevent automatic login if true-Sign V is run on a computer that is not a member of a specific domain.

Name	Type	Description
`(default)`	`REG_SZ`	The user agent to set when no DNS domain is configured for the computer running true-Sign V. Default: *not set*
`<DNS Domain>`	`REG_SZ`	The user agent to set when the computer running true-Sign V is a member of `<DNS Domain>`.

Table 18: Browser user agents

! Note that the AJAX.NET or other JavaScript frameworks may fail if the user agent is not set correctly.

Samples

- User agent string that will force form-based authentication in ADFS 2012R2 but still support AzureAD password change (note that this will not work for `__doPostBack` pages for work account password recovery):

  ```
  Mozilla/4.0 (compatible; trueSignV4) like Gecko
  ```

- Standard user agent that will allow Windows Integrated authentication when ADFS is used:

  ```
  Mozilla/4.0 (compatible; MSIE 11.0; Windows NT 6.1; WOW64) like Gecko
  ```

3.4.4.5 Allowed navigation domains

Configure the allowed domains that may be visited during the authentication process. Configuring the allowed domains prevents users from breaking out of the authentication process.

```
…\Browser\AllowedNavigationDomains
```

Name	Type	Description
<Name>	REG_SZ	URL without path. Sample: https://login.microsoftonline.com

Table 19: Allowed navigation domains

ⓘ If no allowed navigation domains are defined, no restrictions are applied, and all links clicked that are not opening a new browser window are opened in internal browser.

! If a web page contains a link, which opens a new browser window, it may not fall und the restriction.

3.4.4.6 Proxy settings

Configure the proxy settings for the WebView2 browser. The settings are translated to command line options as described in[1].

```
…\Browser\Proxy
```

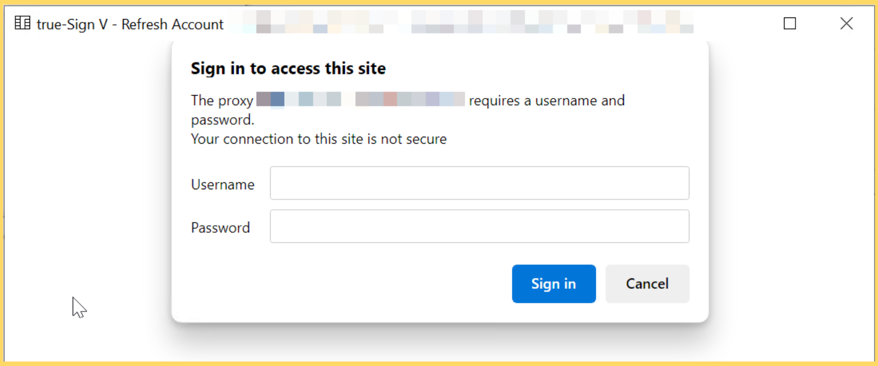
Name	Type	Description
`Type`	`REG_SZ`	The type of proxy configuration to use: `System` – Use the system configured proxy `None` – Do not use a proxy `AutoDetect` – Automatically detect the proxy configuration `PAC` – Download and use a Proxy Automatic Configuration (PAC) file from the URL specified in the `Config` value. `Server` – A specific proxy server to use. The server is specified in the `Config` value. Default: `System`
`Config`	`REG_SZ`	Configuration depending on the `Type` configured. Check the Edge proxy description[1] for details.
`BypassList`	`REG_SZ`	; separated list of domains and IP addresses[1] for which to bypass the proxy.
`User`	`REG_SZ`	If `Type` is `Server`, the username to use for authentication. Ignored if `Password` is not set. Default: *not set*
`Password`	`REG_SZ`	If `Type` is `Server`, the password to use for authentication. Ignored if `User` is not set. Default: *not set*

Table 20: Proxy settings for the WebView2 browser

ⓘ These proxy settings only apply to the WebView2 browser. API calls for the signature service require their own configuration. See Appendix A:

[1] https://learn.microsoft.com/en-us/deployedge/edge-learnmore-cmdline-options-proxy-settings

The WebView2 component will ask for proxy credentials when no user and password is set or otherwise required.



true-Sign V - Refresh Account [redacted]

Sign in to access this site

The proxy [redacted] requires a username and password.
Your connection to this site is not secure

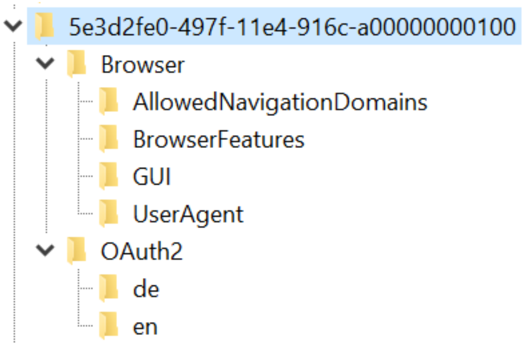
Figure 19: Proxy authentication in browser

To allow the user to save the proxy credentials, `AllowPasswordAutosave` must be enabled.

Note that adding an account and authenticating an existing account use two different WebView2 profiles. Saving the proxy credentials in one profile will not allow their use in the other profile, the user will have to enter and save the credentials in the account profile as well.

3.4.5 OAuth2

If the provider specifies `OAuth2` as the `AuthenticationType`, the following settings and sub keys are available under the `Browser` and the `OAuth2` sub-keys:



- 5e3d2fe0-497f-11e4-916c-a00000000100
  - Browser
    - AllowedNavigationDomains
    - BrowserFeatures
    - GUI
    - UserAgent
  - OAuth2
    - de
    - en

Figure 20: OAuth2 configuration sub keys

Since the `OAuth2` authentication is also browser based, it extends the `Browser` authentication and all settings available for the `Browser` authentication type are thus also available for the `OAuth2` authentication type.

The following settings are available directly under the `OAuth2` key:

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>AuthorizeURI</td><td>REG_SZ</td><td>The URI of the OAuth2 authorization endpoint (/authorize).
The following placeholders will be replaced in the AuthorizeURI:
<table>
<tr><td>{client_id}</td><td>The ClientID.</td></tr>
<tr><td>{code_challenge}</td><td>In case of PKCE, the PKCE challenge.</td></tr>
<tr><td>{code_challenge_method}</td><td>In case of PKCE, the challenge method used for creating the challenge (e.g., S256).</td></tr>
<tr><td>{lang}</td><td>The language code (e.g., en) the true-Sign V client uses.</td></tr>
<tr><td>{login_hint}</td><td>The user id if known. When adding an account, the user is empty.</td></tr>
<tr><td>{redirect_uri}</td><td>The RedirectURI</td></tr>
<tr><td>{response_type}</td><td>The ResponseType</td></tr>
<tr><td>{scope}</td><td>The Scope</td></tr>
<tr><td>{state}</td><td>A random GUID generated for each authorization request.</td></tr>
</table></td></tr>
<tr><td>Browser</td><td>REG_SZ</td><td>Use the WebView2 browser (internal) or the system default browser (external).
Note that if the system browser (external) is used, true-Sign V has no control over cookies and other security measures. It is also not possible for true-Sign V to logout a user.
Default: internal</td></tr>
</table>

Name	Type	Description
`ClientAuthentication`	`REG_SZ`	The type of client authentication to use for the authorization code flow. The following mechanisms are supported: `None` Do not use any information for the client authentication. `IdOnly` Use only `client_id` for the client authentication. If the `ClientID` is not configured, the user will be presented to enter the missing information. `IdAndSecret` Use `client_id` and `client_secret` for the client authentication. If `ClientID` and/or `ClientSecret` is not configured, the user will be presented to enter the missing information. `PKCE` Use PKCE (RFC 7636: Proof Key for Code Exchange) for the client authentication. If the `ClientID` is not configured, the user will be presented to enter the missing information. Default: *not set*
`ClientID`	`REG_SZ`	The `client_id` to use. See `ClientAuthentication`. Default: *not set*
`ClientSecret`	`REG_SZ`	The `client_secret` to use. See `ClientAuthentication`. Default: *not set*

Name	Type	Description
CookieDeletionPolicy	REG_SZ	The cookie deletion policy for OAuth2 authentications when the internal WebView2 based browser is used. This allows to clear cookies that remember a logged in user which can cause confusion when multiple accounts are configured for the same provider. The following policies are supported: Never: Do not delete any cookies. AuthorizeDomainOnly: Only delete cookies for the domain specified in AuthorizeURI. SpecifiedDomains: Clear cookies for the domains listed in CookieDomains. AllDomains: Delete all cookies. Note that this will log out other sessions as well. Use with caution. Default: Never
CookieDomains	REG_SZ	A comma separated list of domain names for which cookies shall be deleted before OAuth2 authentication. Only available when the internal WebView2 based browser is used This list is only used if CookieDeletionPolicy is set to SpecifiedDomains and may include .domain and , domain/path entries.
Scope	REG_SZ	The authorization scope to use. Default: service
OAuth2EndpointURI	REG_SZ	The OAuth2 base URI, if it differs from the CSC base URI. This URI if specified, will be used as the base URI for the /oauth2/token endpoint. Default: *not set*
RedirectURI	REG_SZ	The redirect URI associated with the client. Upon redirection to the URI, the authorization code flow is assumed to be completed. Default: *not set*
ResponseType	REG_SZ	The authorization response_type (grant) to use. Default: code

Name	Type	Description
`UsernameFormField`	`REG_SZ`	If present, any text input form field with the given ID will be set to the user id. Default: *not set*

Table 21: OAuth2 authentication settings

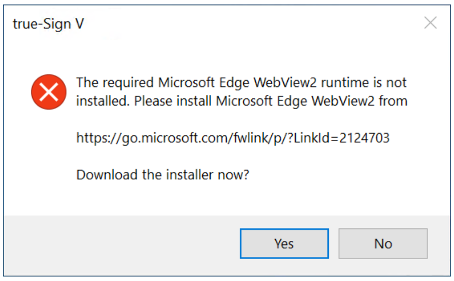
> ! Note that the `BrowserFeatures` / `PersistCookies` setting defines if persistent cookies are allowed at all. If persistent cookies are allowed, the `CookieDeletionPolicy` and `CookieDomains` settings can be used to delete cookies that retain account specific account information and may cause problems if multiple different user accounts are configured for a given provider.

3.4.5.1 Using the internal browser for authentication

Since version 4.1, true-Sign V uses the Microsoft Edge WebView2[2] runtime for the embedded browser. This runtime is automatically updated by Windows Update and provides the latest JavaScript and web browser features like the Microsoft Edge browser does unlike the previously used Trident engine which was based in Internet Explorer 11 and is no longer maintained by Microsoft.

> ℹ The EXE installers will automatically install the Microsoft Edge WebView2 evergreen runtime if WebView2 is not installed on the system.
>
> The MSI installers will not automatically install a missing Microsoft Edge WebView2 runtime as they are intended for enterprise deployments. Future Windows 10 and Windows 11 versions are expected to contain the Microsoft Edge WebView2 runtime by default.

If the Microsoft Edge WebView2 runtime is not present when a browser-based authentication is required by true-Sign V, the error show will directly allow to download the missing runtime installer using the system browser:



true-Sign V

The required Microsoft Edge WebView2 runtime is not installed. Please install Microsoft Edge WebView2 from

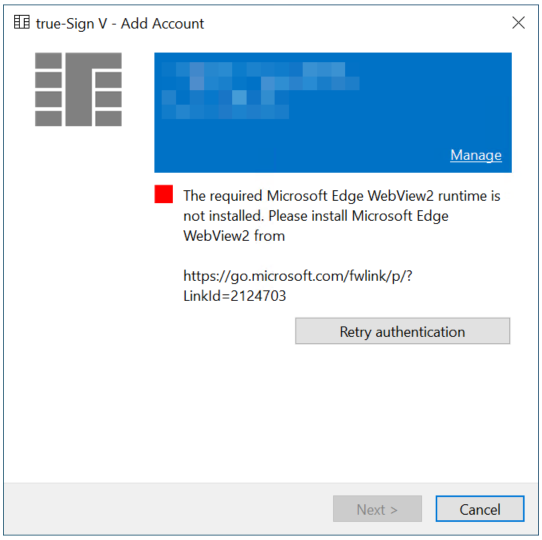
https://go.microsoft.com/fwlink/p/?LinkId=2124703

Download the installer now?

Figure 21: Missing Microsoft Edge WebView2 runtime error

[2] https://developer.microsoft.com/en-us/microsoft-edge/webview2/

The download link can also be copied from the authentication dialog after closing the error popup:



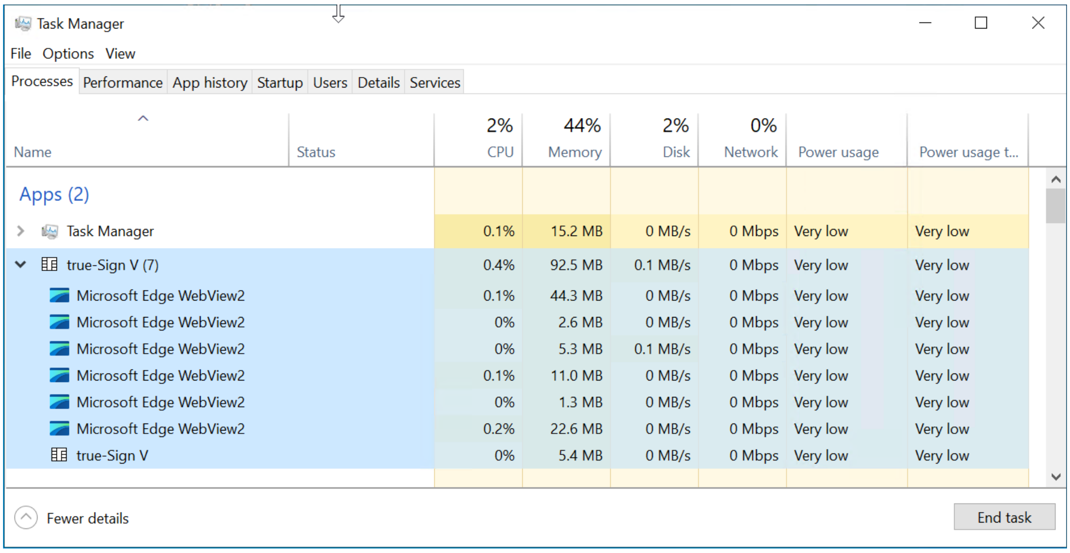
[redacted]

The required Microsoft Edge WebView2 runtime is not installed. Please install Microsoft Edge WebView2 from

https://go.microsoft.com/fwlink/p/?LinkId=2124703

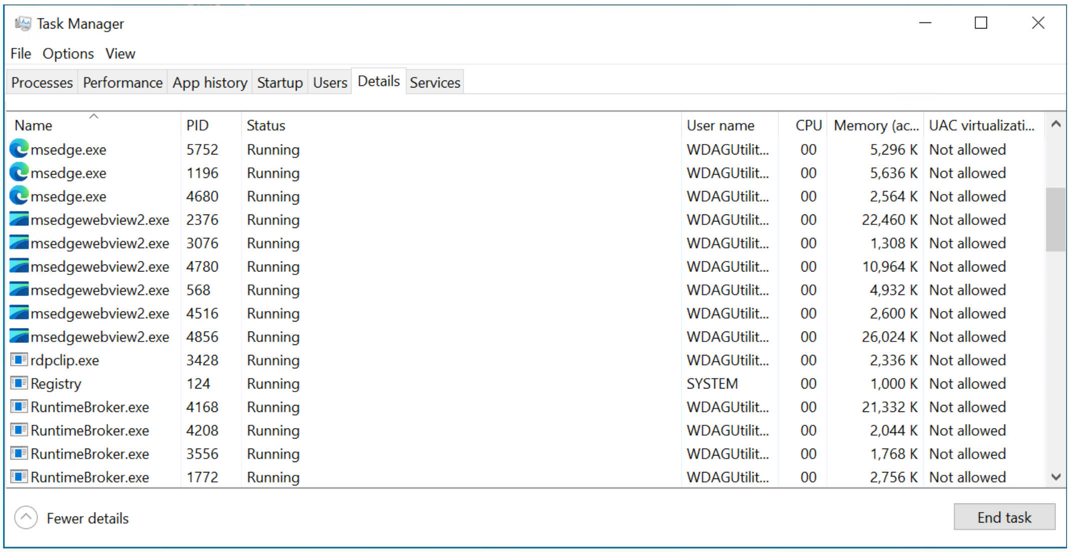
Figure 22: Download link for Microsoft Edge WebView2 runtime

The Microsoft Edge WebView2 runtime runs the browser in external processes named `msedgewebview2.exe`:


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
Fewer details
End task

Name	Status	2% CPU	44% Memory	2% Disk	0% Network	Power usage	Power usage t...
Apps (2)							
Task Manager		0.1%	15.2 MB	0 MB/s	0 Mbps	Very low	Very low
true-Sign V (7)		0.4%	92.5 MB	0.1 MB/s	0 Mbps	Very low	Very low
Microsoft Edge WebView2		0.1%	44.3 MB	0 MB/s	0 Mbps	Very low	Very low
Microsoft Edge WebView2		0%	2.6 MB	0 MB/s	0 Mbps	Very low	Very low
Microsoft Edge WebView2		0%	5.3 MB	0.1 MB/s	0 Mbps	Very low	Very low
Microsoft Edge WebView2		0.1%	11.0 MB	0 MB/s	0 Mbps	Very low	Very low
Microsoft Edge WebView2		0%	1.3 MB	0 MB/s	0 Mbps	Very low	Very low
Microsoft Edge WebView2		0.2%	22.6 MB	0 MB/s	0 Mbps	Very low	Very low
true-Sign V		0%	5.4 MB	0 MB/s	0 Mbps	Very low	Very low

Figure 23: Task Manager processes with internal browser


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
Fewer details
End task

Name	PID	Status	User name	CPU	Memory (ac...	UAC virtualizati...
msedge.exe	5752	Running	WDAGUtilit...	00	5,296 K	Not allowed
msedge.exe	1196	Running	WDAGUtilit...	00	5,636 K	Not allowed
msedge.exe	4680	Running	WDAGUtilit...	00	2,564 K	Not allowed
msedgewebview2.exe	2376	Running	WDAGUtilit...	00	22,460 K	Not allowed
msedgewebview2.exe	3076	Running	WDAGUtilit...	00	1,308 K	Not allowed
msedgewebview2.exe	4780	Running	WDAGUtilit...	00	10,964 K	Not allowed
msedgewebview2.exe	568	Running	WDAGUtilit...	00	4,932 K	Not allowed
msedgewebview2.exe	4516	Running	WDAGUtilit...	00	2,600 K	Not allowed
msedgewebview2.exe	4856	Running	WDAGUtilit...	00	26,024 K	Not allowed
rdpclip.exe	3428	Running	WDAGUtilit...	00	2,336 K	Not allowed
Registry	124	Running	SYSTEM	00	1,000 K	Not allowed
RuntimeBroker.exe	4168	Running	WDAGUtilit...	00	21,332 K	Not allowed
RuntimeBroker.exe	4208	Running	WDAGUtilit...	00	2,044 K	Not allowed
RuntimeBroker.exe	3556	Running	WDAGUtilit...	00	1,768 K	Not allowed
RuntimeBroker.exe	1772	Running	WDAGUtilit...	00	2,756 K	Not allowed

Figure 24: Task Manager WebView2 process details

The Microsoft Edge WebView2 runtime stores browser data used with true-Sign V authentication operations, including cookies, saved passwords if enabled etc. in the user’s local application data directory under:

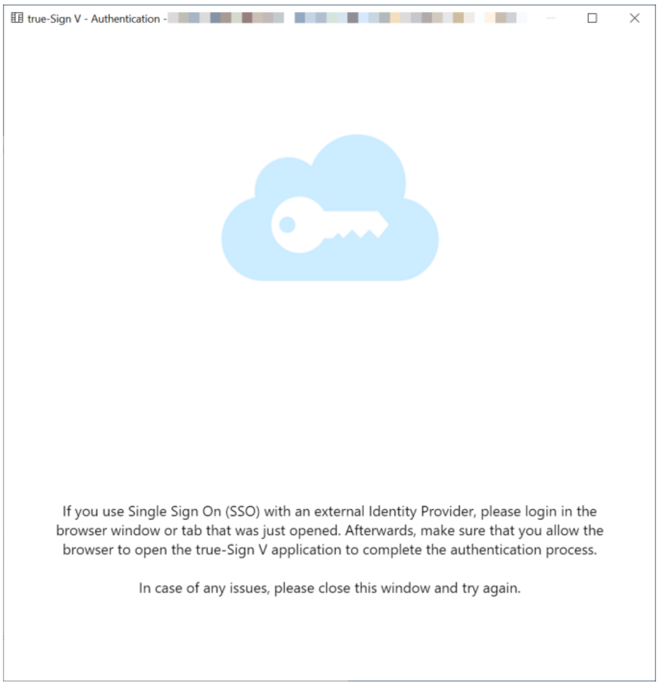
```
%LOCALAPPDATA%\keyon\trueSignV\EBWebView
```

In case of problems with the browser based authentication, one can safely delete the folder and try again.

3.4.5.2 Using an external browser for authentication

Using an external browser has a worse user experience and less intuitive flow than using the internal WebView2 browser. You should only use an external browser if special functionality is needed for the login which is not provided by the internal browser.

When the authentication should be handled by the system default browser (`Browser` set to `external`), true-Sign V will show a window with a help text and then open the system browser with the authentication URL:



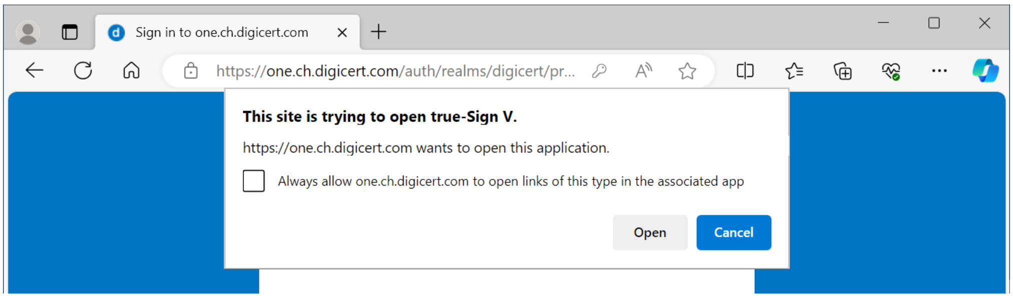
true-Sign V - Authentication - [redacted]

If you use Single Sign On (SSO) with an external Identity Provider, please login in the browser window or tab that was just opened. Afterwards, make sure that you allow the browser to open the true-Sign V application to complete the authentication process.

In case of any issues, please close this window and try again.

Figure 25: External authentication notification

When an external browser is used for the OAuth2 process, the redirect at the end of the process will trigger the true-Sign V MIME handler. For security reasons the user will have to allow the browser to execute the external MIME handler at the first use and in subsequent uses, if the user does not opt to automatically allow the use in future sessions:



Sign in to one.ch.digicert.com

https://one.ch.digicert.com/auth/realms/digicert/pr...

This site is trying to open true-Sign V.

https://one.ch.digicert.com wants to open this application.

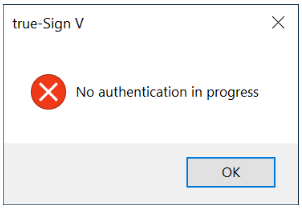
□ Always allow one.ch.digicert.com to open links of this type in the associated app

Figure 26: External browser MIME handler confirmation

Note that the browser window or tab is not closed after external MIME handler is executed. The user should close the tab.

If the tab is kept open, refreshing the tab will cause the true-Sign V MIME handler to be called again. This can also occur if the bowser is configured to re-open previous sessions at startup.

If the MIME handler is called again with the result for a previously completed authentication operation, the following error will be shown:



true-Sign V

No authentication in progress

Figure 27: MIME handler authentication error

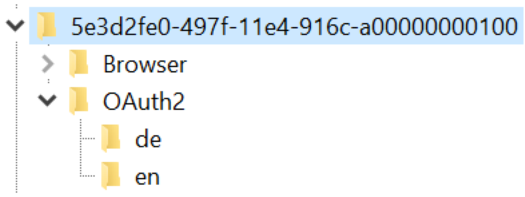
Note that authentication information like passwords and cookies may be cached by the external browser. true-Sign V has no control about the handling of such information when an external browser is used. Configurations options like allowed navigation domains and cookie policies are not available when an external browser is used for authentication.

3.4.5.3 OAuth2 GUI Texts

`OAuth2` GUI texts are configured in sub keys under the `OAuth2` configuration with the ISO 639-1 two-character primary language code, e.g. `en` for English or `de` for German as the sub key name.

The texts are only used if the user needs to provide the `ClientID` and/or `ClientSecret`, depending on the client authentication method configured.

`…\OAuth2\en`



```
5e3d2fe0-497f-11e4-916c-a00000000100
  Browser
  OAuth2
    de
    en
```

Figure 28: Provider language sub keys

> The GUI texts for `en` are always required since `en` is used as a fallback if no specific configuration for the user's primary display language is present.

Under each language specific sub-key, the following configuration elements are available:

Name	Type	Description
`HintUserID`	`REG_SZ`	The hint to show as gray text in the empty user entry field.
`HintClientID`	`REG_SZ`	The hint to show as gray text in the empty ClientID entry field.
`HintClientSecret`	`REG_SZ`	The hint to show as gray text in the empty ClientSecret entry field.
`ToolTipUserID`	`REG_SZ`	The tool tip text to use for the user entry field. If `ToolTipUserID` is missing or empty, no tool tip icon will be shown next to the entry field. You can use `\n` to insert line breaks.
`ToolTipClientID`	`REG_SZ`	The tool tip text to use for the ClientID entry field. If `ToolTipClientID` is missing or empty, no tool tip icon will be shown next to the entry field. You can use `\n` to insert line breaks.
`ToolTipClientSecret`	`REG_SZ`	The tool tip text to use for the ClientSecret entry field. If `ToolTipClientSecret` is missing or empty, no tool tip icon will be shown next to the entry field. You can use `\n` to insert line breaks.

Table 22: OAuth2 GUI texts

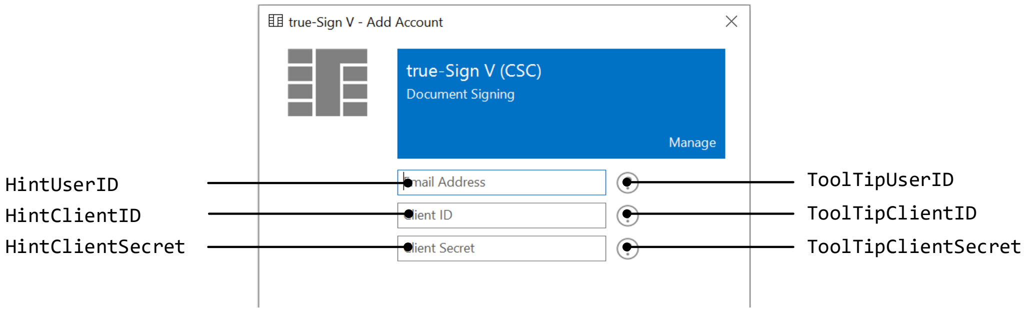


Figure 29: OAuth2 hints and tool tips

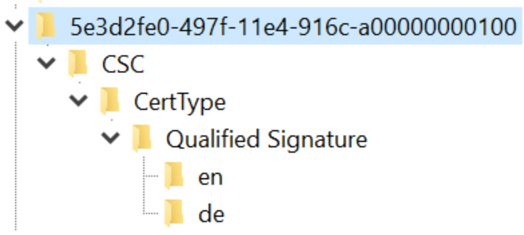
The client secret field is only shown if the configured client authentication method requires a client secret.

3.5 API specific configuration

Depending on the `APIType` specified for a provider, additional configuration entries are required.

3.5.1 CSC_1_0_4 (Cloud Signature Consortium API V1.0.4.0)

If the provider specifies `CSC_1_0_4` as the `APIType`, the following settings and sub keys are available:



```
5e3d2fe0-497f-11e4-916c-a00000000100
  CSC
    CertType
      Qualified Signature
        en
        de
```

Figure 30: CSC sub-key

The following settings are available under the `CSC` sub-key:

Name	Type	Description
`OTPSingleUse`	`DWORD`	Defines if the same OTP value cannot be used for two or more subsequent signatures. Note that it depends on the service if the same OTP during its validity period can be used multiple times or only once. Setting `OTPSingleUse` to `1` (true) will limit the number of signatures to one signature every OTP period of 30s. Default: `0x00000000` (no)
`SupportsExtendTransactionForSCAL1`	`DWORD`	Defines if the provider supports multiple signatures with a single authorization by implementing the use *extendTransaction* API endpoint for credentials with SCAL set to 1 or not defined. The CSC credential info structure defines the maximum number of signatures allowed. Default: `0x00000000` (no)
`SupportsExtendTransactionForSCAL2`	`DWORD`	Defines if the provider supports multiple signatures with a single authorization by implementing the use *extendTransaction* API endpoint for credentials with SCAL set to 2. The CSC credential info structure defines the maximum number of signatures allowed. Default: `0x00000000` (no)

Name	Type	Description
`ProhibitedHashAlgorithms`	`REG_SZ`	Comma separated list of hash algorithm OIDs that are not supported by the remote signature provider and will result in an error when creating the signature. (E.g. use 1.3.14.3.2.26 to prevent the use of SHA-1.) Default: (not set)
`PolicyNotification`	`REG_SZ`	The text shown in the certificate tile under the certificate title text. Please note that this text cannot be localized. The CSC description uses the language of the true-Sign V application if possible. The following placeholders can be used in the string: `{commonName}` Common name element from the certificate `{cscId}` CSC credential ID based on credentials/list → credentialIDs `{cscIdShort}` CSC credential ID up to the first non-alphanumeric character `{cscDescription}` CSC credential description based on credentials/info → description Default: `{cscId}`

Name	Type	Description
`MultisignRestriction`	`DWORD`	Restrict multiple signatures to specific machines (local and remote) and processes if the CSC provider supports the *extendTransaction* API endpoint. 1 Allow multiple signatures only in the same CSP/KSP session. Most restrictive setting that requires the application to use a single CSP/KSP context for all signatures. 2 Allow multiple signatures only on the same machine, process name and process id (PID). Limits the authorization to a single process instance. (Default) 3 Allow multiple signatures only on the same machine and process name. Allows multiple instances of an application started on the same machine. 4 Allow multiple signatures by any process name on the same machine. Allows a multisign authorization to be used by any applications on the same machine. 5 Allow multiple signatures by the same process name on any machine. Allows multiple instances of an application on multiple machines. 6 Allow multiple signatures by any process on any machine. Allows a multisign authorization to be used by any applications on any machine. Default: 2

Table 23: CSC settings

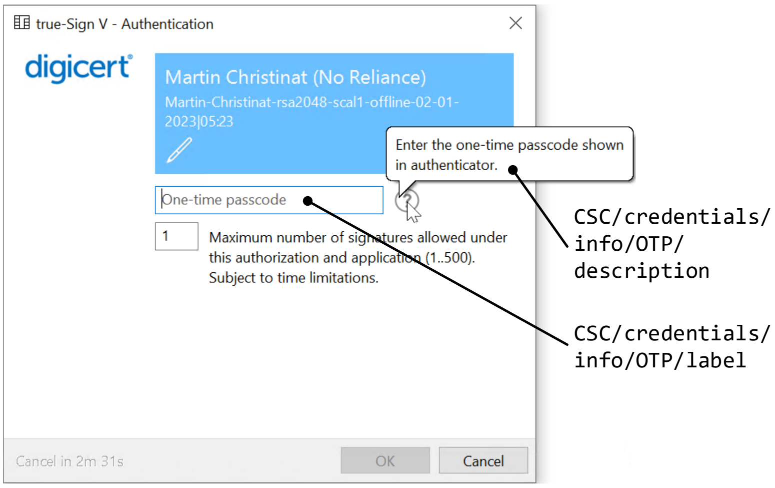
Under each language specific sub-key, the following configuration elements are available:

Name	Type	Description
`ImplicitAuthInfo`	`REG_SZ`	The text to show when the credential authorization method is implicit and the user will need to use a device or app to authorize the signature instead of providing an OTP or PIN. Specify `-` to prevent the dialog to be shown. The following placeholders can be used in the string: `{id}` The session id under which the signature is requested. Default: (not set)

Table 24: Implicit authorization text

3.5.1.1 OTP dialog (offline OTP)

When the CSC remote signing service requires an offline OTP for the signature authorization, the following dialog is displayed when asking for the OTP. The hint and tool tip elements are taken from the CSC credentials/info OTP JSON element, the true-Sign V client sets the language code used for its own GUI when requesting the credential info:



Enter the one-time passcode shown in authenticator.

Maximum number of signatures allowed under this authorization and application (1..500). Subject to time limitations.

Figure 31: CSC offline OTP dialog elements

OTP JSON configuration producing the above dialog:

```
"OTP": {
  "ID": "DigiCert One Document Signing one-time passcode",
  "description": "Enter the one-time passcode shown in authenticator.",
  "format": "N",
  "label": "One-time passcode",
  "presence": "true",
  "provider": "TOTP",
  "type": "offline"
```

```
    }
```

The entry field for the number of signatures is only shown under the following conditions:

a) The credentials/info JSON for the credential contains a `multisign` value > 1

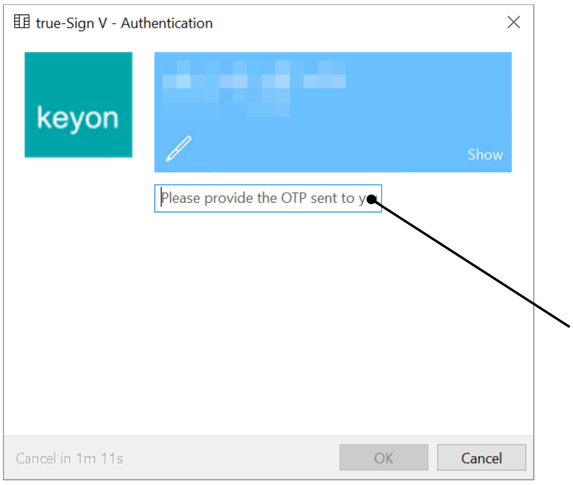
b) The certificate type configuration has `Multisign` either not defined or set to a value > 1

c) The credential has SCAL set to 1 or SCAL is not defined, and the provider configuration has `SupportsExtendTransactionForSCAL1` set to 1 (yes) or the credential has SCAL set to 2 and the provider configuration has `SupportsExtendTransactionForSCAL2` set to 1 (yes)

The prefilled maximum number of signatures is always set to 1 for security reasons, the user must explicitly set a higher value if multiple signatures are intended.

3.5.1.2 OTP dialog (online OTP)

When the CSC remote signing service requires an online OTP for the signature authorization, the following dialog is displayed when asking for the OTP. The hint is taken from the CSC credentials/info OTP JSON element, the true-Sign V client sets the language code used for its own GUI when requesting the credential info:


true-Sign V - Authentication
keyon
Show
Please provide the OTP sent to y
Cancel in 1m 11s
OK
Cancel

`CSC/credentials/info/OTP/label`

Figure 32: CSC online OTP dialog elements

OTP JSON configuration producing the above dialog:

```
"OTP": {
  "presence": true,
  "types": "online",
  "format": "N",
  "label": "Please provide the OTP sent to your number",
  "description": "",
  "ID": "",
  "provider": ""
}
```

If multiple signatures are allowed, a similar additional field is shown as described in 3.5.1.2.

3.5.1.3 PIN dialog

When the CSC remote signing service requires a PIN for the signature authorization, the following dialog is displayed when asking for the PIN. The hint and tool tip elements are taken from the CSC credentials/info PIN JSON element, the true-Sign V client sets the language code used for its own GUI when requesting the credential info:

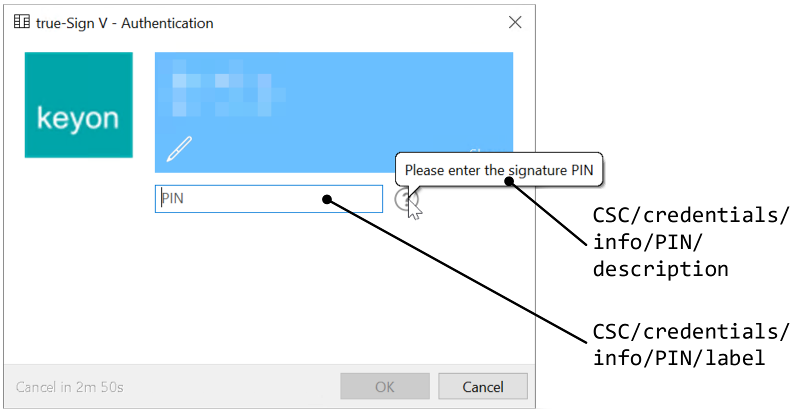


Figure 33: CSC PIN dialog elements

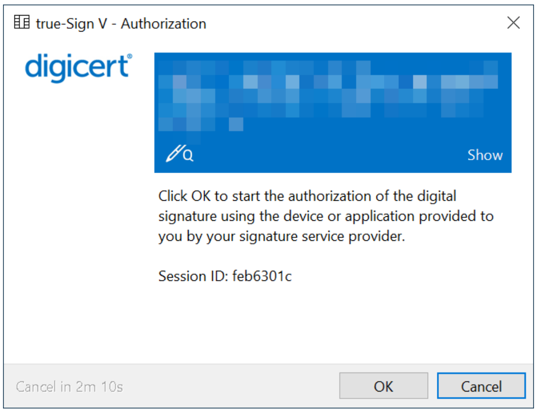
PIN JSON configuration producing the above dialog:

```
"PIN": {
  "description": "Please enter the signature PIN",
  "format": "N",
  "label": "PIN",
  "presence": true
}
```

If multiple signatures are allowed, a similar additional field is shown as described in 3.5.1.2.

3.5.1.4 Implicit authorization dialog

When the CSC remote signing service uses implicit authorization for a credential, the following dialog is displayed to notify the user that he must authorize the signature using the device or application that was provided by the signature service provider:



Click OK to start the authorization of the digital signature using the device or application provided to you by your signature service provider.

Session ID: feb6301c

Figure 34: CSC implicit authorization dialog

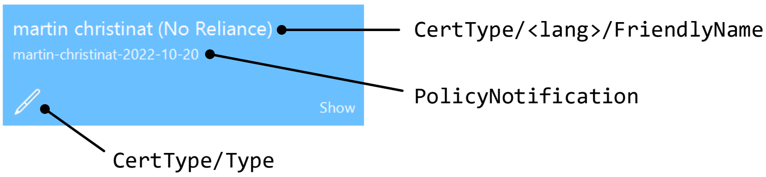
The default text can be changed using the `ImplicitAuthInfo` configuration of the CSC configuration or the certificate type configuration if certificate type specific mechanisms are used. This allows to specify the specific application (e.g. Go>Sign Mobil) to use for authorizing the signature.

This dialog can be suppressed by specifying `-` (minus) as the text in the `ImplicitAuthInfo` configuration.

3.5.1.5 Certificate type configuration

Under the `CSC` key, there is a sub-key `CertType` containing one or multiple sub-keys for each supported certificate type by the provider. Each of these certificate type specific sub keys contains language dependent GUI configurations using two-digit language codes according to ISO 639-1.

The certificate type is used to identify the certificate in the dialogs showing a cert tile. The following elements are part of the cert tile:



martin christinat (No Reliance)
martin-christinat-2022-10-20
Show
CertType/<lang>/FriendlyName
PolicyNotification
CertType/Type

Figure 35: CSC cert tile elements

The Certificate type settings under `CertType/`*`<certificate type alias>`* are as follows:

Name	Type	Description
`Filter`	`REG_SZ`	A certificate filter specification which matches the certificate type. See *Appendix C: Certificate Filter Definition* for details on the filter definition language. Please make sure that a certificate matches only one configured certificate type per provider as otherwise the match is ambiguous and may lead to showing a wrong certificate type information.
`Multisign`	`DWORD`	Limit the number of multiple signatures to the specified value if the CSC provider supports the *extendTransaction* API endpoint. The CSC credential info structure defines the maximum number of signatures allowed. If set to 1, multiple signatures are disabled for this certificate type even if the CSC credential info allows for multiple signatures. Default: Not set, CSC credential info *multisign* is used.
`MultisignRestriction`	`DWORD`	Restrict multiple signatures to specific machines (local and remote) and processes for this certificate type. If present, this setting will override the provider `MultisignRestriction` setting. See `MultisignRestriction` in Table 23: CSC settings for allowed values. Default: Not set, CSC provider setting is used.

Name	Type	Description
Type	DWORD	The certificate type icons to show for this type. The following bitmasks are defined for the available icons: 0x00000001 Qualified signature 0x00000002 Advanced personal signature 0x00000004 Advanced business signature 0x00000008 Simple signature 0x00000010 Authentication 0x00000020 Encryption Multiple icons can be shown if e.g. a certificate allows for signatures and encryption (0x00000022). See table Table 26: Certificate type icons for the icons used. Default: 0x00000008

Table 25: Certificate type settings

Icon	Certificate type	Bitmask
✎Q	Qualified signature	0x00000001
✎+	Advanced personal signature	0x00000002
✎+	Advanced business signature	0x00000004
✎	Simple signature	0x00000008
🪪	Authentication	0x00000010
🔒	Encryption	0x00000020

Table 26: Certificate type icons

Under each language specific sub-key, the following configuration elements are available:

Name	Type	Description
`FriendlyName`	`REG_SZ`	The text to show as the title in the certificate tile identifying the certificate. The following placeholders can be used in the string: `{commonName}` Common name element from the certificate `{cscId}` CSC credential ID based on credentials/list → credentialIDs `{cscIdShort}` CSC credential ID up to the first non-alphanumeric character `{cscDescription}` CSC credential description based on credentials/info → description Default: {commonName}
`ImplicitAuthInfo`	`REG_SZ`	The text to show when the credential authorization method is implicit, and the user will need to use a device or app to authorize the signature instead of providing an OTP or PIN. Specify – (minus) to prevent the dialog from being shown. The certificate type specific configuration will overrule the `ImplicitAuthInfo` configuration set at CSC level. The following placeholders can be used in the string: `{id}` The session id under which the signature is requested. Default: (not set)

Table 27: Certificate type GUI texts

4 true-Sign V Cert Store Provider configuration

The true-Sign V Cert Store Provider can create a distinct view of the certificates for an application using the store. This includes both selecting a subset of certificates available to the application and settings the cryptographic provider to assign with the certificate. It is thus possible to force a specific cryptographic provider, e.g. the CSP `keyon trueSign V Cryptographic Service Provider` if the application cannot use the modern key storage provider.

4.1 Configuration locations

The following locations are searched for true-Sign V Cert Store Provider configurations:

Order	Root Key	true-Sign V Configuration Locations
1	`HKEY_LOCAL_MACHINE`	`SOFTWARE\keyon\trueSignV\CertStore`
2	`HKEY_LOCAL_MACHINE`	`SOFTWARE\Policies\keyon\trueSignV\CertStore`
3	`HKEY_CURRENT_USER`	`SOFTWARE\keyon\trueSignV\CertStore`
4	`HKEY_CURRENT_USER`	`SOFTWARE\Policies\keyon\trueSignV\CertStore`

Table 28: true-Sign V Cert Store Provider configuration locations

Entries in a higher order location will overwrite entries in lower order locations if they share the same name.

4.2 Default Application Profile

A default application profile can be optionally specified under the key

```
…\trueSignV\CertStore\AppProfiles\Default
```

It will be used as a fallback when no application specific profile is defined for the calling process:

Name	Type	Description
`CertFilter`	`REG_SZ`	The certificate filter to apply if not empty. The certificate filter must be specified as described in section Appendix C:. Default: *not set*
`Provider`	`REG_SZ`	The cryptographic service provider to associate with the certificate if not empty. The cryptographic service must exist as specified in section 4.5. Default: *not set, will use the crypto provider defined in the service provider configuration*

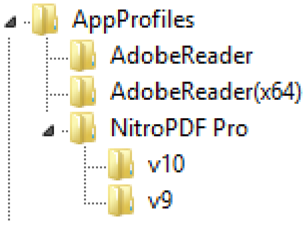
Table 29: Default application profile configuration settings

> You can specify a certificate filter that will never match any certificate (e.g. `authorityKeyIdentifier=00`) to enable certificates only for processes that match a specific application profile.

4.3 Application Profiles

Application profiles are created under the key

```
…\trueSignV\CertStore\AppProfiles
```



- AppProfiles
  - AdobeReader
  - AdobeReader(x64)
  - NitroPDF Pro
    - v10
    - v9

Figure 36: Application Profile Keys

An application profile consists of the configuration identifying the process and defining the default certificate filter and provider and optionally process version specific settings that may override the default certificate filter and provider.

Name	Type	Description
`DefaultCertFilter`	`REG_SZ`	The certificate filter to apply if not empty. The certificate filter must be specified as described in section Appendix C:. Note that a version specific configuration can overwrite this filter. Default: *not set*
`DefaultProvider`	`REG_SZ`	The cryptographic service provider to associate with the certificate if not empty. The cryptographic service must exist as specified in section 4.5. Note that a version specific configuration can overwrite this provider. Default: *not set, will use the crypto provider defined in the service provider configuration*

Name	Type	Description
`ProcessPathSpec`	`REG_SZ`	The process path pattern defining if a given process matches this application profile. The true-Sign V Cert Store Provider will check the path of the process that loaded it against this value to determine if this application profile should be applied. The path may contain environment variables enclosed in `%` and DOS Wildcards `*` and `?`. The check for a match uses the Microsoft `PathMatchSpec` API function. The check is not case-sensitive. Sample: `%ProgramFiles%\Adobe\Reader *\Reader\AcroRd32.exe`

Table 30: Application profile generic configuration settings

4.3.1 Application version specific settings

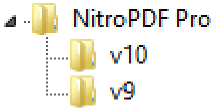
Different versions of an application may support different cryptographic service providers and may even require different certificate filters. Older application versions e.g. may only support CSPs for cryptographic operations and will use any certificate regardless of the intended usage while later versions support KSP and will only show suitable certificates.

The true-Sign V Cert Store Provider allows using distinct provider and certificate filter configurations for different versions of an application even if they are stored in the same location. (If the different versions use distinct installation paths, the path can be used to differentiate between the versions.)

Version specific application profiles are created under the application profile key

```
…\trueSignV\CertStore\AppProfiles\<Application>
```

and may use an arbitrary key name for each version:



NitroPDF Pro
v10
v9

Figure 37: Version specific application profile keys

The version specific application profile key contains the following configuration settings:

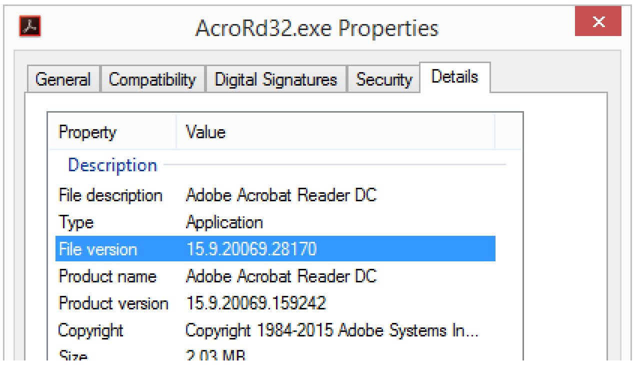
Name	Type	Description
`CertFilter`	`REG_SZ`	The certificate filter to apply if not empty. The certificate filter must be specified as described in section Appendix C:. Note that a version specific configuration can overwrite this filter. Default: *not set*

Name	Type	Description
`Provider`	`REG_SZ`	The cryptographic service provider to associate with the certificate if not empty. The cryptographic service must exist as specified in section 4.5. Note that a version specific configuration can overwrite this provider. Default: *not set, will use the crypto provider defined in the service provider configuration*
`VersionMatch`	`REG_SZ`	The version to match. If the file version of the process matches this version considering the matching instruction, the provider specified with `Provider` and the cert filter specified by `CertFilter` are used. See section 4.3.1.1 for details on the version matching.

Table 31: Version specific application profile configuration settings

4.3.1.1 Version matching

The application version is taken from the file version properties of the process executable loading the certificate store. This version can be shown by clicking properties on the process in the Windows Explorer:



Property	Value
Description	
File description	Adobe Acrobat Reader DC
Type	Application
File version	15.9.20069.28170
Product name	Adobe Acrobat Reader DC
Product version	15.9.20069.159242
Copyright	Copyright 1984-2015 Adobe Systems In...
Size	2.03 MB

Figure 38: Executable file version properties

A file version may contain from one up to four numbers and has the form

```
major[.minor[.revision[.build]]]
```

In order to match versions, the configuration uses version-matching strings of the format

```
[operator]major[.minor[.revision[.build]]]
```

where the operator determines the kind of match as described in the following table:

Operator	Description
<	Matches if the version of the process is lower than the configured version
<=	Matches if the version of the process is lower or equal to the configured version
= or no operator	Matches if the version of the process is equal to the configured version
>=	Matches if the version of the process is higher or equal to the configured version
! or <>	Matches if the version of the process is not equal to the configured version

Table 32: Version comparison operators

Examples: `>=15`

Will match any file version that has major version 15 or higher regardless of their minor, revision or build versions

`<10.2`

will match any file version with major number 10 if minor is less than 2 (i.e. 0 or 1) and any file with major number 9 or less regardless of their minor, revision or build versions

4.3.1.2 Order of checks when multiple version specific configurations are available

When multiple version specific settings are available, the version specific settings are checked in the order from the check with the highest version number to the check specifying the lowest version number. The first matching check will determine the certificate filter and provider to use.

This ordering allows to behave correctly if we have settings for e.g. `>10`, `>9`, `>6` (we only know the past versions) and the process version is `11.2.3`. While all of the checks will match, only the `>10` match makes sense in this case as it is expected to provide a configuration that is suitable unlike e.g. the `>6` configuration.

4.3.1.3 Certificate specific provider configurations

It is possible to configure certificate specific providers under a version specific configuration. Unlike the certificate filter configuration which limits the certificates visible to the application, the certificate specific configuration changes the cryptographic provider for matching certificates.

Certificate specific provider configurations created in the sub key `CertSpecificProviders` under the version key:

```
…\trueSignV\CertStore\AppProfiles\<Application>\<version>\CertSpecificPro
viders
```

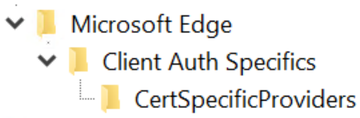


Figure 39: Certificate specific provider configurations

The `CertSpecificProviders` key contains one or more of the following configuration settings:

Name	Type	Description
CertificateFilterName	REG_SZ	The cryptographic service provider to associate with the certificate filter specified as *CertificateFilterName*. The cryptographic service must exist as specified in section 4.5. The certificate filter must be specified as described in section 4.4.

Table 33: Certificate specific provider configuration settings

If the process version matches (use >=0 to match all versions) and the certificate filter matches, the certificate, the certificate specific provider will win over all other provider configurations.

4.4 Certificate Filters

Application profiles reference certificate filters by a short alias (i.e. the sub-key name) under the key

```
…\trueSignV\CertStore\CertificateFilters
```

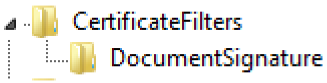


Figure 40: Certificate Filter Keys

Each key can have any number of arbitrarily named values that each specifies a certificate filter. If any of the filters match, the certificate will be included in the store.

Name	Type	Description
FilterName	REG_SZ	The certificate filter definition to use. See *Appendix C: Certificate Filter Definition* for details on the filter definition language.

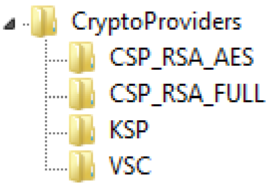
Table 34: Certificate Filter configuration elements

4.5 Crypto Providers

Application profiles reference crypto providers by a short alias under the key

```
…\trueSignV\CertStore\CryptoProviders
```

containing the full provider name and type as values.



CryptoProviders
- CSP_RSA_AES
- CSP_RSA_FULL
- KSP
- VSC

Figure 41: Crypto Provider Keys

Depending on the installed components, the following crypto provider definitions are available and should not be changed:

Alias	Referenced provider
CSP_RSA_AES	`keyon trueSign V RSA and AES Cryptographic Service Provider`
CSP_RSA_FULL	`keyon trueSign V Cryptographic Service Provider`
KSP	`keyon trueSign V Key Storage Provider`
VSC	`Microsoft Base Smart Card Crypto Provider`

Table 35: Crypto Provider Aliases

See *Appendix D: true-Sign V Providers* for more details on the true-Sign V providers.

Do not change the settings for the crypto providers. They are simply present for reference purposes by application profile settings.

4.6 Custom machine store support

If machine store support is enabled for true-Sign V, you must register the true-Sign V Cert Store Provider for the `MY` (or any other machine certificate store such as `WSUS`) as follows using a local administrator account:

Registration for machine MY store:

```
rundll32 trueSignCertStore.dll,RegisterInCustomMachineStore MY
```

Deregistration for machine MY store:

```
rundll32 trueSignCertStore.dll,UnregisterFromCustomMachineStore MY
```

If you need machine store support for 32-Bit applications, use `C:\Windows\SysWOW64\rundll32.exe` when registering or deregistering the store.

Appendix A: Service Configuration

A.1 Service configuration entries

Name	Type	Description
`ServiceURL`	`REG_SZ`	The service URL
`ServiceTimeout`	`DWORD`	The timeout to set for the service call in milliseconds. Default: `0x0000EA60` (60000ms, 1min)
`ServiceSecurityFlags`	`DWORD`	Set flags to weaken the security checks made by the WinINet API. See chapter Security Flags for details on the available flags. Default: `0x00000000` (no flags set)
`ServiceIssuerFingerprints`	`REG_SZ` or `MULTI_SZ`	Hex string of a 20-byte SHA1 hash value. Multiple hex strings are defined by using the REG_MULTI_SZ type. If defined, the SHA1 hash of the CA that issued the server certificate must match one of these values. If not set, no check is performed and the CA certificates present in the Windows certificate store determine the trust. Default: *not set*
`ServiceNegotiateSupport`	`DWORD`	Enable proxy or server authentication using Kerberos or NTLM if set to a non-zero value. If the proxy or server requires authentication, a multi-step negotiation handshake needs to be executed before the actual service call can take place. Set this value to a non-zero value if your environment uses a proxy that requires authentication or if the server behind the `ServiceURL` requires NTLM or Kerberos authentication. Default: `0x00000000` (false)

Name	Type	Description

`ServiceProxyNames`	`REG_SZ`	If given, the default Windows proxy settings are not used but the proxies configured using `ServiceProxyNames` are called when connecting to the `ServiceURL`.
		To explicitly prevent the use of a proxy, specify either `direct` or none as the server's name. If a proxy server is to be used, the string must follow the format:
		`<protocol>=<protocol>://<proxy_name>`
		Sample:
		`https=http://proxy.keyon.local:8080`
		Multiple proxies are specified using a space as the delimiter. Set `ServiceNegotiateSupport` to a non-zero value if the proxy requires any form of authentication (Basic Authentication, NTLM or Kerberos).
		Default: *not set* (use system proxy settings)

`ServiceProxyUser`	`REG_SZ`	Defines the proxy user to set if the proxy specified with `ServiceProxyNames` requires basic authentication. If the proxy uses Kerberos or NTLM authentication, the `ServiceProxyUser` must not be set. Default: *not set*
`ServiceProxyPassword`	`REG_SZ`	Defines the password for the proxy user to set if the proxy specified with `ServiceProxyNames` requires basic authentication. If the proxy uses Kerberos or NTLM authentication, the `ServiceProxyPassword` must not be set. Default: *not set*

Table 36: Service configuration entries

A.2 Security Flags

Service security flags can be set to relax the security checks made by the WinINet API. The security flags are described on the following MSDN page under INTERNET_OPTION_SECURITY_FLAGS:

> https://msdn.microsoft.com/en-us/library/windows/desktop/aa385328%28v=vs.85%29.aspx

The following values can be combined using an or operation to set multiple flags:

Value	Description	Impact
`0x00000080`	Ignores certificate revocation problems. The connection will succeed even if the revocation state of the certificate cannot be determined.	⚠
`0x00000100`	Ignores unknown certificate authority problems. The connection will succeed even if the CA is not present in the Root or CA certificate store. This will allow man-in-the-middle attacks.	⊗
`0x00000200`	Ignores incorrect certificate usage problems. The connection will succeed even if the certificate chain has incorrect key usage or extended key usage restrictions for TLS server authentication.	⊗
`0x00001000`	Ignores non-matching common name or *dNSName* entries in the *subjectAltName* extension if present. The connection will succeed even if the server certificate is not valid for the requested host name. This will allow man-in-the-middle attacks.	⊗
`0x00002000`	Ignores invalid certificate dates such as expired or not yet valid certificates. The connection will succeed even if any of the certificates in the certificate chain is not yet valid or expired.	⊗

Table 37: Service security flags

Security flags are usually only set for testing purposes as they have a severe impact on the connection security. Ignoring certificate revocation problems, however, can be useful in an enterprise environment when clients cannot download the CRL.

Note that when the SOAP backend is used, PINs are encrypted independent of the transport security used for the server. This guarantees that PINs entered in the true-Sign V application can only be decrypted by the crypto service hosted by the service provider.

Appendix B: Auto Provision / Refresh Configuration

B.1 Auto provision / refresh configuration entries

Name	Type	Description
`AutoMaxConsecutiveErrors`	`DWORD`	Defines the maximum number of consecutive errors allowed when trying to auto-provision or auto-refresh the provider. If the maximum number of errors is reached, no further attempts are made until true-Sign V is restarted. Default: `0x00000000` (no limit)
`AutoProvision`	`DWORD`	Enable auto-provisioning for the provider if not `0`. If enabled, true-Sign V the application will try adding an account for the provider at startup if no account information is present. This operation is silent, i.e. there are no GUIs presented to the user in both the success and failure cases. Default: `0x00000000` (false)
`AutoRefresh`	`DWORD`	Enable auto-refresh of the account information (e.g. assigned certificates) if not `0`. If enabled, true-Sign V will check every `AutoRefreshInterval` seconds for an updated account policy. Default: `0x00000000` (false)
`AutoRefreshInterval`	`DWORD`	The auto-refresh interval in seconds if `AutoRefresh` is enabled. Default: `0x00007080` (28800s = 8h)

Table 38: Auto provisioning and refresh configuration

Appendix C: Certificate Filter Definition

true-Sign V currently supports two kinds of filter expressions:

LDAP Style Filter Definition

A string-based description language inspired by LDAP search filters. This filter definition was introduced with version 2.4 of true-Sign V and allows to create complicated constructs with AND, OR and NOT operations.

Example (with line breaks for clarity):

```
(&
   (keyUsage~=digitalSignature)
   (!
       (extendedkeyUsage~=id-kp-clientAuth)
   )
)
```

Legacy Filter Definition

A simple string-based description language developed for the initial version of true-Sign V. This language allows multiple checks, but the result of all checks is combined using an AND operation. It is therefore not possible with this language to create a filter that e.g. matches certificates issued by either CA1 or CA2.

Example:

```
KU+digitalSignature;EKU-id-kp-clientAuth
```

Since the new *LDAP Style Filter Definition* always starts with a bracket in contrast to the *Legacy Filter Definition*, which never starts with a bracket, either format can be used to define a certificate type. If the filter expression starts with (as the first non-whitespace character, the *LDAP Style Filter Definition* is assumed, otherwise the expression is assumed to be a *Legacy Filter Definition.*

ⓘ Subsequent major versions of true-Sign V will only support the *LDAP Style Filter Definition* syntax.

C.1 LDAP Style Filter Definition

The *LDAP Style Filter Definition* allows filtering certificates using a syntax like the LDAP search filter specified in RFC2254[3]. It allows matching certificate attributes using different operators and allows combining the results of multiple filters using logic expressions.

Filter definition in ABNF notation

```
<filter> ::= '(' <filtercomp> ')'
<filtercomp> ::= <and> | <or> | <not> | <item>
<and> ::= '&' <filterlist>
<or> ::= '|' <filterlist>
<not> ::= '!' <filter>
<filterlist> ::= <filter> | <filter> <filterlist>
<item> ::= <simple> | <present> | <extensible>
<simple> ::= <attribute> <operator> <value>
<operator> ::= <equal> | <contains> | <ge> | <le>
<equal> ::= '='
<contains> ::= '~='
<ge> ::= '>='
<le> ::= '<='
<present> ::= <attribute> '=*'
<extensible> ::= <attribute> ':' <parameter> ':' <filtertype> <value>
<colon> ::= ':'
```

Where `<attribute>`, `<parameter>` and `<value>` are strings. The following characters in these strings must be quoted using \xx:

Character	Quoted value
*	\2a
(	\28
)	\29
\	\5c
:	\3a
{	\7b
}	\7d

Table 39: Quoted characters for certificate filters

[3] https://tools.ietf.org/search/rfc2254

Single filter

A single filter performs a match for a specific attribute or property of the certificate:

```
(<attribute>[:<parameter>:]<operator>[<value>])
```

Compound filter

A compound filters perform a logic operation on the result of multiple single or compound filters:

- Logical "AND" operation, compound filter matches only if every $filter_n$ matches:

  (&($filter_1$)($filter_2$)[($filter_3$)[($filter_4$)…]])

  Where ($filter_n$) can be either a single filter or a compound filter.

- Logical "OR" operation, compound filter matches if any $filter_n$ matches:

  (|($filter_1$)($filter_2$)[($filter_3$)[($filter_4$)…]])

  Where ($filter_n$) can be either a single filter or a compound filter.

- Logical "NOT" operation, compound filter matches if $filter_1$ does not match:

  (!($filter_1$))

  Where ($filter_1$) can be either a single filter or a compound filter.

Since the filters within a compound filter can be compound filters as well, nesting filters allows create complicated logic operations:

Sample:

```
(&(keyUsage~=digitalSignature)(!(extendedkeyUsage~=id-kp-clientAuth)))
```

In this case, the Key Usage must have the *digitalSignature* Bit set and the Extended Key Usage must not include the OID for *id-kp-clientAuth* (1.3.6.1.5.5.7.3.2).

Interpreting compound filter strings can be somewhat difficult due to the dense notation:

(&($filter_1$)(!(|($filter_2$)($filter_3$))))

Step 1: Reformat to use several lines and indention:

```
(&
        (filter_1)
        (!
                (|
                        (filter_2)
                        (filter_3)
                )
        )
)
```

Step 2: Convert to textual logical expression:

```
filter₁ AND (NOT (filter₂ OR filter₃))
```

Operators

The operators available in a single filter are

Operator	Description / Value
=	Equal. The filter matches if the attribute or property is equal to the value. For string values, a case dependent match is performed.
=*	Present. The filter matches if the attribute or property is present. The value of the attribute does not matter for this filter, only that the attribute or property is present.
~=	Contains. The filter matches if the attribute or property contains the specified value. For string values, a case independent sub string match is performed.
>=	Greater or equal. The filter matches if the attribute or property is greater or equal than the specified value. Only useful for integer attributes.
<=	Less or equal. The filter matches if the attribute or property is less or equal than the specified value. Only useful for integer attributes.

Table 40: Operators for certificate filters

Session variables

The value part can contain references to session variables that are expanded before the filter is applied. Variables are referenced using the form:

{*variable*}

Variable	Description / Value
`COMPUTER_DNS_DOMAIN`	The DNS domain name of the computer if the computer is joined to a domain. Sample: `keyon.local`
`COMPUTER_DNS_FQDN`	The fully qualified domain name of the computer if the computer is joined to a domain. Sample: `WIN-CHR-01.keyon.local`
`COMPUTER_DNS_HOSTNAME`	The host name of the computer. Sample: `WS-WIN-CHR-01`
`COMPUTER_NETBIOS_HOSTNAME`	The NetBIOS name of the computer. Sample: `WS-WIN-CHR-01`
`USER_LOGIN_DOMAIN`	The NetBIOS domain name of the logged in user. Sample: `KEYON`
`USER_LOGIN_NAME`	The login name (sAMLoginName) of the logged in user. Sample: `christinat`
`USER_UPN`	The user principal name (UPN) of the logged in user. For local users, an artificial UPN of the form <login name>@<local machine name> is used. Sample: `christinat@keyon.ch`
`USER_UPN_DOMAIN`	The domain name part of the UPN of the logged in user. Sample: `keyon.ch`
`USER_UPN_NAME`	The user name part of the UPN of the logged in user. Sample: `christinat`

Table 41: Session variables for use in certificate filters

Sample filter for matching certificates containing the UPN of the logged in user:

```
(subjectAltName:upn~={USER_UPN})
```

Available attributes and allowed operators

Attribute	Param	Operator					Certificate attribute / Check
		=	=*	~=	>=	<=	
authoritykeyidentifier		●	●	●			Authority key identifier in cert
certificatepolicy		●	●	●			Certificate policy in cert
certificatetemplate		●	●	●			Certificate template in cert
extendedkeyusage		●	●	●			Extended key usage in cert
extension	●		●				Extension by OID in cert
fingerprint		●		●			SHA-1 fingerprint of cert
issuerdn		●	●	●			Issuer DN in cert
keylength		●		●	●	●	Length of public key in cert
keytype		●					Type of public key in cert
keyusage		●	●	●			Key usage in cert
lifetime		●		●	●	●	Total lifetime of cert
qcstatements	(●)	●	●	●			Qualified certificate statement in cert
remaininglifetime		●		●	●	●	Remaining lifetime of cert
serialnumber		●		●			Serial number in cert
subjectaltname	●	●	●	●			Subject alt name part of cert
subjectdn		●	●	●			Subject DN in cert

Table 42: Available attributes and allowed operators for certificate filters

Notes

- Attribute names are not case sensitive, e.g. `subjectaltname` and `subjectAltName` refer to the same attribute.
- Some attributes allow or require a parameter to specify which part of an attribute should be matched.
- Only numeric values such as key length or lifetimes are supported for the less than (<=) and greater than (>=) filter types.
- Attributes that are always available such as the fingerprint or the public key type cannot be matched for presence.

If a filter contains an unsupported attribute, the match for this filter will always be false, i.e. a non-match. Please make sure you use only available attributes, e.g. by verifying the filter using the *FilterCertificate* utility provided.

authoritykeyidentifier

Description	This attribute can be used to filter certificates issued by a specific CA if the issued certificate contains an *Authority Key Identifier* extension (2.5.29.35).
Value	A hex string with characters 0..9 and A..F or a..f. The match with the value is performed case insensitive after converting the *Authority Key Identifier* binary data to a hex string. Note that you can specify only part of an *Authority Key Identifier* when using ~= but this may lead to ambiguous matches.
Samples	`(authoritykeyidentifier=3c212c0670069ee827ccb0e0c1875b178eab80f9)`

certificatepolicy

Description	This attribute can be used to filter certificates containing a *Certificate Policies* extension (2.5.29.32) with a specific *Policy Identifier* OID.
Value	Comma separated list of one or more OIDs of *Policy Identifiers*.
Samples	`(certificatepolicy=1.3.6.1.4.1.8024.1.200)`

certificatetemplate

Description	This attribute can be used to filter certificates containing a *Certificate Template Information* extension (1.3.6.1.4.1.311.21.7) with a specific certificate template OID. (Microsoft specific extension used by Active Directory Certificate Services)
Value	The OID of a certificate template.
Samples	`(certificatetemplate=1.3.6.1.4.1.311.21.8.7394417.7803492.8669340.7442408.10992920.126.11126878.8791720)`

extendedkeyusage

Description	This attribute can be used to filter certificates containing an *Extended Key Usage* extension (2.5.29.37) for specific usage OIDs.
Value	A comma separated list of one or more OIDs and/or strings from the following list: ▪ `anyExtendedKeyUsage` ▪ `id-kp-serverAuth` ▪ `id-kp-clientAuth` ▪ `id-kp-codeSigning` ▪ `id-kp-emailProtection` ▪ `id-kp-timeStamping` ▪ `id-kp-OCSPSigning` ▪ `id-ms-kp-sc-logon` ▪ `id-ms-kp-document-signing`

Samples	Certificates that allow only client authentication and no other usages:
	`(extendedkeyusage=id-kp-clientAuth)`
	Certificates that allow client authentication but may allow other usages as well:
	`(extendedkeyusage~=id-kp-clientAuth)`
	Certificates that allow client and server authentication but may allow other usages as well:
	`(extendedkeyusage~=id-kp-clientAuth,id-kp-serverAuth)`
	Certificates that allow a specific usage identified by OID (in the sample the same as id-ms-kp-sc-logon) but may allow other usages as well:
	`(extendedkeyusage~=1.3.6.1.4.1.311.20.2.2)`

extension

Description	This attribute can be used to filter certificates containing an extension with a specific OID.
Parameter	The OID of the extension
Value	None as only =* is allowed for `extension`
Samples	`(extension:1.3.6.1.5.5.7.1.1:=*)`

fingerprint

Description	This attribute can be used to filter a specific certificate based on its SHA-1 fingerprint.
Value	A hex string with characters 0..9 and A..F or a..f. The match with the value is performed case insensitive after calculating the SHA-1 fingerprint of the binary certificate data and converting the fingerprint to a hex string. Note that you can specify only part of a fingerprint when using ~= but this may lead to ambiguous matches.
Samples	`(fingerprint=fbf66d48a9b80ed1df8b0de4ac87d01c2a07c7a3)`

issuerdn

Description	This attribute can be used to filter certificates by contents of the issuer DN.
Value	String. Please note that = matches case sensitive and whitespace after, must be present while ~= matches case insensitive. For exact matches, use e.g. *certutil* to get the correct string to use.

Samples	Certificates that contain keyon AG as a sub string (case insensitive):
	`(issuerdn~=keyon AG)`
	Certificates that have an exact issuer DN (case sensitive):
	`(issuerdn=CN=Keyon AG - Certification Authority - 2, O=Keyon AG, C=CH)`

keylength

Description	This attribute can be used to filter certificates based on the length of the public key.
Value	An integer > 0.
Samples	`(keylength>=2048)`

keytype

Description	This attribute can be used to filter certificates based on the type of public key.
Value	The OID of a public key algorithm or one of the following strings: ▪ `RSA` ▪ `ECC` ▪ `DSA`
Samples	`(keytype=ECC)`

keyusage

Description	This attribute can be used to filter certificates containing a *Key Usage* extension (2.5.29.15) for specific key usages.
Value	A comma separated list of one or more strings from the following list: ▪ `digitalSignature` ▪ `nonRepudiation` ▪ `keyEncipherment` ▪ `dataEncipherment` ▪ `keyAgreement` ▪ `keyCertSign` ▪ `cRLSign` ▪ `encipherOnly` ▪ `decipherOnly`

Samples	Certificates that allow only non-repudiation and no other usages:
	`(keyusage=nonRepudiation)`
	Certificates that allow digital signature but may allow other usages as well:
	`(keyusage~=digitalSignature)`
	Certificates that allow client and server authentication but may allow other usages as well:
	`(keyusage~=digitalSignature,nonRepudiation)`

lifetime

Description	This attribute can be used to filter certificates based on their lifetime in days.
Value	An integer > 0.
Samples	`(lifetime>=730)`

qcstatements

Description	This attribute can be used to filter certificates containing a *Qualified Certificate Statements* extension (1.3.6.1.5.5.7.1.3) for an element.
Parameter	Optional: An OID specifying a QcStatement. The QcStatement is expected to have a sequence of strings or object identifiers which are checked for the specified match.
Value	A String.

Samples	Certificates that contain a *Qualified Certificate Statements* extension with a statement indicating that it is a European Qualified Certificate (ETSI TS 101 862, `id-etsi-qcs-QcCompliance`):
	`(qcstatements~=0.4.0.1862.1.1)`
	Certificates that contain a *Qualified Certificate Statements* extension with a statement indicating that the country under which the certificate was issued is `CH` (ETSI EN 319 412-5, `id-etsi-qcs-QcCClegislation`):
	`(qcstatements:0.4.0.1862.1.7:=CH)`

remaininglifetime

Description	This attribute can be used to filter certificates based on their remaining lifetime in days based on the current date and time.
Value	An integer >= 0.

Samples	Certificates that are still valid but expire in the next 30 days:
	`(&(remaininglifetime<=30)(remaininglifetime>=1))`
	Certificates that allow only client authentication and no other usages:
	`(remaininglifetime<=0)`

serialnumber

Description	This attribute can be used to filter certificates based on their serial number.
Value	A hex string with characters 0..9 and A..F or a..f. The match with the value is performed case insensitive after converting the serial number integer to a hex string. Note that you can specify only part of a serial number when using ~= but this may lead to ambiguous matches. Serial numbers are only guaranteed to be unique for a single CA. Please make sure that when serialnumber is used for filtering, another filter for issuerdn or authoritykeyidentifier is also applied (&).
Samples	`(serialnumber=125307a3000000000312)`

subjectaltname

Description	This attribute can be used to filter certificates containing a *Subject Alternative Name* extension (2.5.29.17) for an element.
Parameter	A comma separated list of one or more strings from the following list: ▪ `UPN` ▪ `RFC822` ▪ `EMAIL` (alias for RFC822) ▪ `MAIL` (alias for RFC822) ▪ `DNS` (alias for DNSNAME) ▪ `DNSNAME`
Value	A String.

Samples	Certificates that contain a *Subject Alternative Name* extension with an email address present:
	`(subjectaltname:rfc822:=*)`
	Certificates that contain a *Subject Alternative Name* extension with an email address that contains `@keyon.ch` as a substring:
	`(subjectaltname:rfc822:~=@keyon.ch)`
	Certificates that contain a *Subject Alternative Name* extension with an *otherName* entry for a user principal name (UPN) that contains `@keyon.ch` as a substring:
	`(subjectaltname:upn:~=@keyon.ch)`

subjectdn

Description	This attribute can be used to filter certificates by contents of the subject DN.
Value	A String. Please note that `=` matches case sensitive and whitespace after commas must be present while `~=` matches case insensitive. For exact matches, use e.g. *certutil* to get the correct string to use.
Samples	`(subjectdn~=keyon AG)`

LDAP Style Filter Definition Test Application

The `FilterCertificate` utility can be used to test *LDAP Style Filter Definitions* against certificates read from a file or from a certificate store:

```
FilterCertificate [-?|--help] [-v|--verbose] [-x|--explain]
                  [-o|--show-matches-only] [-m|--machine]
                  [-s|--store <name>] [-f|--cert-file <file>]
                  "<filter expression>"

Certificate filter test utility v1.1

Options:

  -?|--help               Show this help text. Add -v to show filter
                          expression syntax and available attributes
                          and variables.
  -v|--verbose            Enable verbose output.
  -x|--explain            Explain the match result by showing the result of
                          each sub filter as well as the compound filters.
  -o|--show-matches-only  Show only certificates matching the filter.
  -m|--machine            Use the machine store and not the user store if no
                          certificate file is provided.
  -s|--store <name>       The certificate store to use if no certificate
                          file is provided. Defaults to MY if not set.
  -f|--cert-file <file>   The certificate file to use. If not specified the
                          cert store will be used.

  "<filter expression>"   The quoted filter expression to apply.
```

Example

Show only certificates suitable for use with e-mail security and e-mail domain keyon.ch and explain the match result:

```
> FilterCertificate.exe -x -o "(&(subjectaltname:rfc822:~=@keyon.ch)
(extendedkeyusage~=id-kp-emailProtection))"

E=christinat@keyon.ch, CN=Martin Christinat, O=Keyon AG, L=Jona, S=SG,
C=CH / 698FA0491D013BF15D9710E4C0AE8278A9714D8D / CN=QuoVadis Swiss
Advanced CA G2, O=QuoVadis Trustlink Switzerland Ltd., C=CH

    Explanation for match result:

        AND
          [ subjectaltname{RFC822} CONTAINS @keyon.ch ] : true
          [ extendedkeyusage CONTAINS id-kp-emailProtection ] : true
        : true
```

C.2 Legacy Filter Definition

The Legacy Filter Definition is deprecated and will not be supported in future major releases. Please us the *LDAP Style Filter Definition* for certificate filter configurations.

The legacy certificate filter definition consists of a list of semicolon (;) separated single filter instructions:

```
<Filter><Operator><Value>[;<Filter><Operator><Value>[;…]]
```

Each single filter instruction can check one aspect of the certificate, e.g. if the key usage has the *digitalSignature* bit set. The single filter instruction contains an operator determining how the check is actually executed and the value to check against.

Examples: `KU+digitalSignature;EKU-id-kp-clientAuth`

This filter definition requires the Key Usage extension to have the *digitalSignature* bit set and the Extended Key Usage extension must not contain the OID for *id-kp-clientAuth* (1.3.6.1.5.5.7.3.2).

```
AKI=3c212c0670069ee827ccb0e0c1875b178eab80f9;KU+digitalSignature;
EKU+id-kp-clientAuth
```

This filter definition requires that the CA with the *subjectKeyIdentifier* `3c212c0670069ee827ccb0e0c1875b178eab80f9` has issued the certificate, the Key Usage extension has the *digitalSignature* bit set and the Extended Key Usage extension contains the OID for *id-kp-clientAuth* (1.3.6.1.5.5.7.3.2).

Filter types

The following filter types are available:

Filter	Checked Attribute	Description
AKI	Authority Key Identifier Extension	Filters based on the Authority Key Identifier Extension. (Identifies the issuing CA)
IDN	Issuer DN	Filters based on the contents of the Issuer DN
SDN	Subject DN	Filters based on the contents of the Subject DN
UPN	UPN in Subject Alt Name Extension	Filters based on the contents of the User Principal Name in the Subject Alt Name Extension
KU	Key Usage Extension	Filters based on the Key Usage Extension
EKU	Extended Key Usage Extension	Filters based on the Extended Key Usage Extension
CLT	Certificate Lifetime	Filters based on the total cert lifetime in days
RLT	Remaining Lifetime	Filters based on the remaining cert lifetime in days
LEN	Key Length	Filters based on the length of the public key in bits

Table 43: Legacy certificate filter types

Operator instructions

The following operator instructions are available:

Operator	Description
+	Checked attribute must contain the value, e.g. Key Usage must have set the *digitalSignature* bit. In case of numerical values (e.g. certificate lifetime), the attribute must be larger or equal (>=).
-	Checked attribute must not contain the value, e.g. Key Usage must not have the *keyEncipherment* bit set. In case of numerical values (e.g. certificate lifetime), the attribute must be less (<).
=	Checked attribute must contain the identical value, e.g. Authority Key Identifier must be the given value. In case of numerical values (e.g. certificate lifetime), the attribute must be identical (=).
!	Checked attribute must not contain the identical value, e.g. Key Usage can be anything expect only the *nonRepudiation* bit set. In case of numerical values (e.g. certificate lifetime) the attribute must be different (`!=`).

Table 44: Legacy certificate filter operator instructions

Certificate filter values

Depending on the filter type, the following values are valid:

Filter	Valid arguments
`AKI`	A hex string with an even number of characters `0..9` and `A..F`
`IDN`	A complete DN or a sub string. Please note that the string format is the one used by Windows. You can use `certutil` to output issuer name strings in the expected format. Note that the string may not contain the semicolon (;) character.
`SDN`	A complete DN or a sub string. Please note that the string format is the one used by Windows. You can use `certutil` to output subject name strings in the expected format. Note that the string may not contain the semicolon (;) character.
`UPN`	A complete UPN or a sub string

`KU`	One of the following names specified in RFC5280:
	`digitalSignature`
	`nonRepudiation`
	`contentCommitment` (same as `nonRepudiation`)
	`keyEncipherment`
	`dataEncipherment`
	`keyAgreement`
	`keyCertSign`
	`cRLSign`
	`encipherOnly`
	`decipherOnly`

Filter	Valid arguments

`EKU`	A valid OID (e.g. `1.3.6.1.5.5.7.3.2`) or one of the following alias names for well-known OIDs:
	`anyExtendedKeyUsage`
	`id-kp-serverAuth`
	`id-kp-clientAuth`
	`id-kp-codeSigning`
	`id-kp-emailProtection`
	`id-kp-timeStamping`
	`id-kp-OCSPSigning`
	`id-ms-kp-sc-logon`
	`id-ms-kp-document-signing`

`CLT`	A positive integer
`RLT`	A positive integer
`LEN`	A positive integer

Table 45: Valid legacy certificate filter values

Appendix D: true-Sign V Providers

Name	Type	Kind
`keyon trueSign V Cryptographic Service Provider`	1	CSP
`keyon trueSign V RSA and AES Cryptographic Service Provider`	24	CSP
`Microsoft Base Smart Card Crypto Provider`[4]	1	CSP
`keyon trueSign V Key Storage Provider`	0	KSP

Table 46: Cryptographic provider names and types

OS	CSP	KSP	Minidriver (Virtual Smart Card)
Workstation OS (x86 / x64)			
Windows 10	Yes	Yes	Yes
Windows 11	Yes	Yes	Yes
Server OS (x64)			
Windows 2012R2	Yes	Yes	Yes
Windows 2016	Yes	Yes	Yes
Windows 2019	Yes	Yes	Yes
Windows 2022	Yes	Yes	Yes

Table 47: Cryptographic providers supported by OS

[4] Used when the Virtual Smart Card is used. This provider will load the trueSign V minidriver for handling the cryptographic operations.

Application	CSP	KSP	Minidriver (Virtual Smart Card)
Windows Environment			
Desktop (Classic Windows API based)	Yes	Yes	Yes
Universal Windows Platform (UWP, “Store Apps”)	Yes, when enabled. Please contact Swiss IT Security if required.	Yes, when enabled. Please contact Swiss IT Security if required.	Yes
Browsers			
Edge Legacy	Yes, when enabled	Yes, when enabled	Yes
Microsoft Edge	Yes	Yes	Yes
Chrome	Yes	Yes	Yes
Firefox v74 or lower	Yes, by using the trueSignP11 PKCS#11 module	Yes, by using the trueSignP11 PKCS#11 module	Yes, by using the trueSignP11 PKCS#11 module
Firefox v75 and higher	Yes	Yes	Yes
Office Applications running in Desktop environment			
Microsoft Office	Yes, limited to SHA-1 for signatures	Yes	Yes
Adobe Acrobat	Yes, limited to SHA-1 for signatures	Yes	Yes
Mail Clients running in Desktop environment			
Outlook	Yes, limited to SHA-1 for signatures	Yes	Yes
Thunderbird	Yes, by using the trueSignP11 PKCS#11 module	Yes, by using the trueSignP11 PKCS#11 module	Yes, by using the trueSignP11 PKCS#11 module

Table 48: Cryptographic provider supported by applications

The Edge Legacy browser and UWP applications run in a highly restricted sandboxed environment that prevents the required communication of the CSP and KSP providers with the true-Sign V application using named pipes. An alternate communication mechanism using UDP can be enabled using `EnableEdgeLegacySupport` to enable Edge Legacy to use certificates provided by true-Sign V.

Application	CSP	KSP	Minidriver (Virtual Smart Card)
Generic .NET Framework based Applications running in Desktop environment			
.Net 2.	Yes	No	Yes
.Net 3.5	Yes	Yes, basic support was added though not integrated with e.g. X509certificate2	Yes
.Net 4 up to 4.5.x	Yes	Yes, basic support was added though not integrated with e.g. X509certificate2	Yes
.Net 4.6 and higher	Yes	Yes	Yes
.Net 5	Yes	Yes	Yes
.NET Core + Platform Extensions 2.0 or higher	Yes	Yes	Yes
.NET Core 3.1 and higher	Yes	Yes	Yes
Generic Java based Applications			
Java 6 up to Java 12 (MSCAPI security provider)	Yes	No	Yes
Java 13 and higher (MSCAPI security provider)	Yes	Yes	Yes
Java 6 and higher (PKCS11 security provider)	Yes, by using the trueSignP11 PKCS#11 module	Yes, by using the trueSignP11 PKCS#11 module	Yes, by using the trueSignP11 PKCS#11 module

Table 49: Cryptographic provider supported by runtime environments

You can use a Cert Store Configuration for applications based on restricted runtime environments to selectively force the use of the CSP and use the KSP for everything else.

While all true-Sign V CSPs support the SHA2 hash algorithms, some applications always use the provider type `RSA_PROV_FULL` with SHA-1 only.

Application	CSP	KSP	Minidriver (Virtual Smart Card)
Visual Studio IDE and tools			
VS 2012 IDE or higher	Yes, limited to SHA-1 for signatures	No. The IDE requires a CSP to show the certificate in the selection dialogs even if the SignTool used during the build can use the KSP	Yes
SignTool	Yes, limited to SHA-1 for signatures	Yes	Yes
Team Foundation Server	Yes, limited to SHA-1 for signatures	Yes	Yes
Java tools			
jarsigner with MSCAPI provider, Java 12 or lower	Yes	No	Yes
jarsigner with MSCAPI provider, Java 13 or higher	Yes	Yes	Yes
jarsigner with PKCS#11 provider	Yes	Yes	Yes

Table 50: Cryptographic provider supported by development tools

Signature algorithm support by provider and API version. Please note that CSC supports API V2:

Algorithm	CSP	KSP		Minidriver		PKCS#11	
		API V1	API V2	API V1	API V2	API V1	API V2
RSA PKCS#1 MD5	●	●	●	●	●	●	●
RSA PKCS#1 SHA-1	●	●	●	●	●	●	●
RSA PKCS#1 SHA-256	●	●	●	●	●	●	●
RSA PKCS#1 SHA-384	●	●	●	●	●	●	●
RSA PKCS#1 SHA-512	●	●	●	●	●	●	●
RSA PKCS#1 SSL3	●	●	●	●	●	●	●
RSASSA-PSS SHA-1		●	●	●	●		
RSASSA-PSS SHA-256			●		●		
RSASSA-PSS SHA-384			●		●		
RSASSA-PSS SHA-512			●		●		
ECDSA P256			●		●		
ECDSA P384			●		●		
ECDSA P512			●		●		

Table 51: Signature algorithm support

Actual signature algorithm support depends on the backend and the certificate type.

Encryption algorithm support by provider and API version. Please note that CSC does not support decryption:

Algorithm	CSP	KSP		Minidriver		PKCS#11	
		API V1	API V2	API V1	API V2	API V1	API V2
RSA PKCS#1	●	●	●	●	●	●	●
RSA OAEP SHA-1	●	●	●	●	●		
RSA OAEP SHA-256		●	●	●	●		
RSA OAEP SHA-384		●	●	●	●		
RSA OAEP SHA-512		●	●	●	●		

Table 52: Encryption algorithm support

Actual encryption algorithm support depends on the backend and the certificate type. The CSC API does not support encryption.

Appendix E: true-Sign V Protocol Handler

true-Sign V registers a protocol handler that can be used to trigger adding an account for a specific provider or to complete an OAuth2 authentication in an external browser.

This can be used e.g. in instruction e-mails or web pages to help users adding their personal account to true-Sign V.

The format of the true-Sign V URI for adding an account is

```
trueSignV://add/<provider GUID>[?param1=x[&param2=y…]]
```

Sample

```
trueSignV://add/5e3d2fe0-497f-11e4-916c-080020010100
```

This URI can be embedded using a hyperlink in a web page

```
<a href="trueSignV://add/5e3d2fe0-497f-11e4-916c-080020010020">
   Start trueSign V enrollment
</a>
```

Browsers or e-mail applications will usually show a warning when a non-standard protocol URI is clicked.

This will start true-Sign V if not already running and open the *Add Account* dialog for the referenced provider:

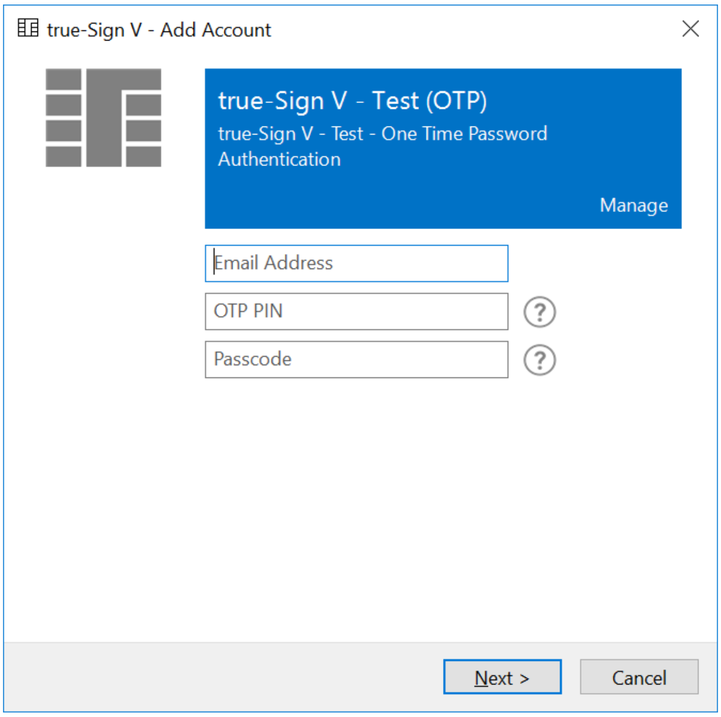
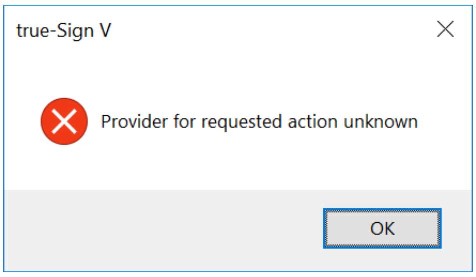

true-Sign V - Add Account
true-Sign V - Test (OTP)
true-Sign V - Test - One Time Password Authentication
Manage
Email Address
OTP PIN
Passcode
Next >
Cancel

Figure 42: Add Account dialog triggered by protocol handler error

If the provider with the given GUID is not configured, an error message is shown instead of the *Add Account* dialog:



true-Sign V

Provider for requested action unknown

Figure 43: Protocol handler error when provider is unknown

Optional parameters supported for `OAuth2` authentication, e.g. if the provider configuration does not contain the `ClientId` and/or `ClientSecret`:

Parameter	Description
`user_name`	The user id
`client_id`	The client id
`client_secret`	The client secret

Table 53: URI parameters for OAuth2 authentication