

Fujitsu mPollux DigiSign Client Technical References

This reference document contains technical information necessary for system administrators, who are installing Fujitsu mPollux DigiSign Client in their IT system.



Contents

1	DigiSign Client smart card reader software	3
1.1	Supported operating systems and standards	3
1.2	References	3
2	Installation	4
2.1	Installation in Windows operating systems	4
2.2	Smart Card Plug and Play service	4
2.3	Windows Smart Card Logon with ECC certificates	5
2.4	“PIN2” behaviour	6
2.5	Notes for Cryptoki application users	6
2.6	Notes for Cryptoki developers	6
2.7	Notes for users who personalize cards with Cryptoki	6
2.7.1	How to generate key that is protected with signature PIN (PIN2)	7
2.8	Notes for Citrix users	7
2.8.1	Chromium based Edge on Windows Server 2016	7
2.9	Notes for MacOS users	8
2.10	Notes for Linux users	8
3	DigiSign Client settings	8
3.1	Optional settings	13
3.2	Web signer technical notes	14
3.2.1	SCS signer additions	14
3.2.2	Running SCS on a server with multiple desktops	14
3.3	Technical note about “dropAuthentication”	15
4	DigiSign Toolkit	16
5	Windows specific notifications	17
5.1	Smart Card Minidriver	17
5.1.1	Certificate Propagation Service	17
6	Options-dialog	18
6.1.1	Smart Card Plug-and-Play Service	20
7	Web browser SSL/TLS Client Authentication challenges	21

1 DigiSign Client smart card reader software

Fujitsu mPollux DigiSign Client software can be used with a smart card for secure access to electronic services or organization networks or for signing documents or email messages electronically.

1.1 Supported operating systems and standards

DigiSign Client supports the following operating systems and standards.

Supported operating systems and standards	
Computing operating systems	Microsoft supported Windows versions for desktop and server use Linux SUSE Enterprise Desktop Red Hat Enterprise Linux Linux Ubuntu CentOS Linux MacOS, three latest operating system versions
Reader driver interfaces	PC/SC
Smart card operating systems	Aventra MyEID applet for JCOP Atos CardOS 5.3 and 5.3DI Gemalto EID2048 applet Gemalto EID classic with PACE/SM Gemalto IP10 Gemalto IDPrime 3940 Thales MAV5.0 IDEMIA ID.me MIOCOS v1.1 or newer (Atmel) MIOCOS v2.3 (Fujitsu FRAM) Oberthur IAS-ECC v1.0.1 Oberthur FINEID applet SetCOS Java EID applet SetCOS 4.3.1, 4.3.2 and 4.4.1
Standard Cryptographic interfaces	Cryptography API: Next Generation (CNG) CryptoAPI v2.0 PKCS#11 v2.01 CryptoTokenKit for MacOS
Other interfaces	DigiSign Toolkit (DLL) WebSigner and WebToolkit
Cryptographic algorithms	MD4, MD5, SHA (different variants) RC-2, RC-4, DES, 3-DES, AES, RSA, RSA-PSS, RSA-OAEP, ECDSA, ECDH
SSL/TLS protocols	SSL 3.0 is no longer supported TLS 1.0, 1.1, 1.2 and 1.3 are supported

1.2 References

The following documentation is provided with the software:

- *Fujitsu mPollux DigiSign Client Technical References* (this guide)
- *Fujitsu mPollux DigiSign Client Installation and User Guide – Windows*
- *Fujitsu mPollux DigiSign Client Installation and User Guide – Linux*
- *Fujitsu mPollux DigiSign Client Installation and User Guide – Mac OS*

2 Installation

2.1 Installation in Windows operating systems

In the Windows operating system, you can install the DigiSign Client by using the installation wizard (see *Fujitsu mPollux DigiSign Client Installation and User Guide – Windows*) or silently.

When Windows .EXE installer is used, the following command installs DigiSign Client silently without the wizard or the background image. Only installation progress is shown.

```
# <DigiSign installation package> /SILENT
```

The following command installs DigiSign Client silently without even the installation progress.

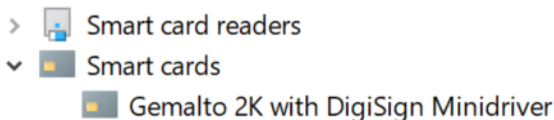
```
# <DigiSign installation package> /VERYSILENT
```

2.2 Smart Card Plug and Play service

When inserting a smart card into a smart card reader after installing DigiSign Client installation

- 1) the SCPnP service detects a card insertion event,
- 2) finds corresponding "minidriver" from the driver cache and
- 3) installs minidriver automatically.

Driver can be seen in the "Device Drivers" view as follows:



If not, the cause may be one of the following:

- 1) Minidriver's code-sign certificate and/or time stamping certificate is not trusted
 - a. By default, both certificates are trusted on the default installation of Windows
 - b. In some cases, organizations delete certificates that are not needed that might accidentally break certificate trust chain
- 2) SmartCard PnP service is not running
 - a. Confirm that SCPnP is not turned off:
"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\ScPnP\EnableScPnP"
 - b. SCPnP server is not used when running Window Server. Therefore DS-Client installer installs minidriver automatically during installation procedure

Minidriver installation can be launched manually by running dpinst64.exe (or dpinst.exe on 32-bit operating systems) under DS-Client installation folder (default location: C:\Program Files\Fujitsu\mPollux DigiSign Client).

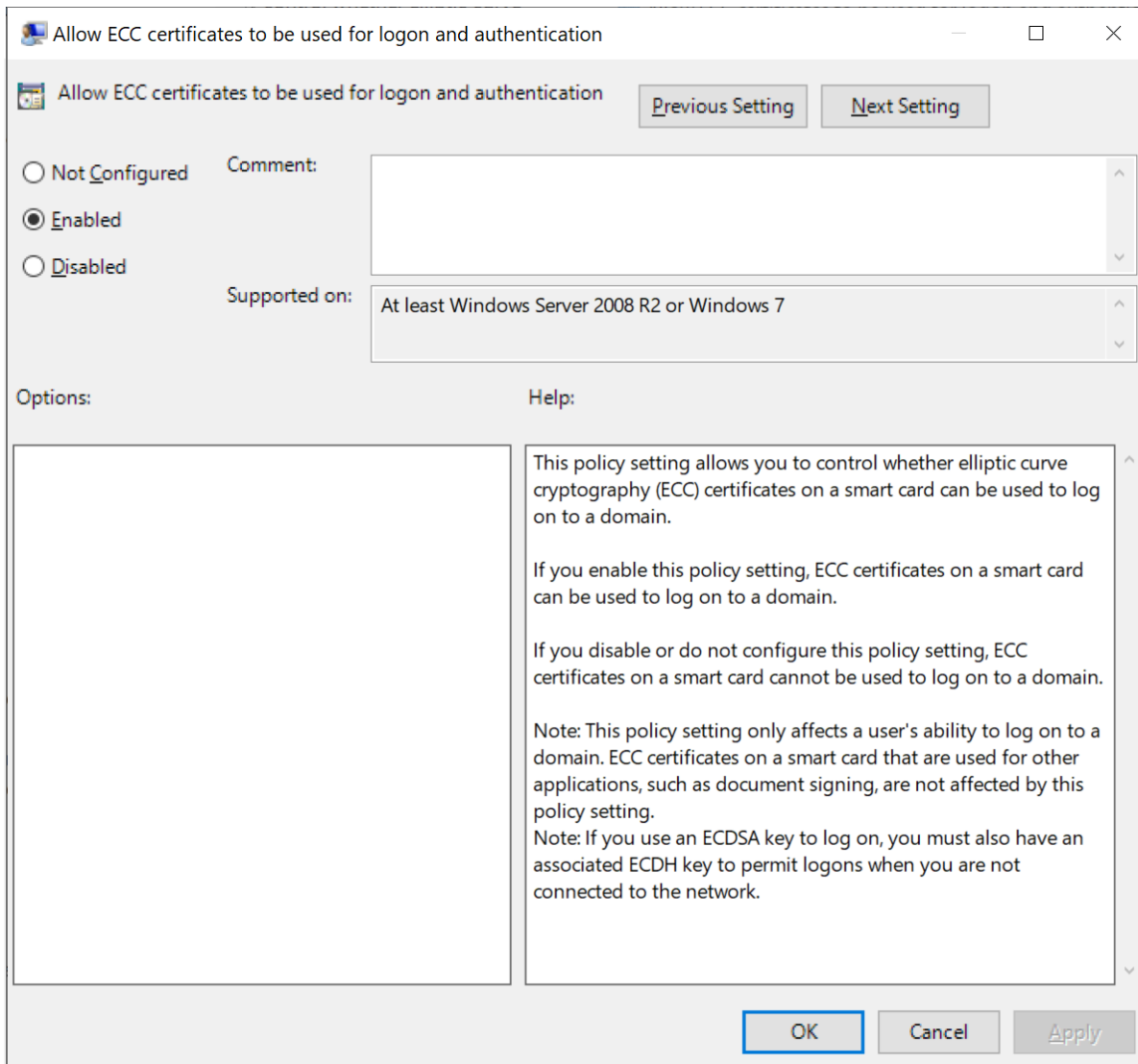
It is possible to use following registry keys to validate minidriver installation:

- 1) Supported smart cards can be found from here:
 - a. Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Calais\SmartCards
 - b. Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Cryptography\Calais\SmartCards
- 2) Installed card readers can be found from here:
 - a. Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Calais\Readers
 - b. Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Cryptography\Calais\Readers
- 3) After installation "dsminidrv.dll" should exist in C:\Windows\System32 folder

2.3 Windows Smart Card Logon with ECC certificates

Windows supports smart card logon to domain workstation. Configuration is described in multiple places and therefore it is not described in this document.

However, ECC certificates requires additional definition to get them working. Following group policy change must be done to get them accepted (Default Domain Policy -> Computer Configuration -> Administrative Templates -> Windows Components -> Smart Card);



2.4 "PIN2" behaviour

Typically, PIN2 protects nonrepudiation signature key. Even though DigiSign application has PIN1 cache to make user experience likeable, PIN2 is **never stored into DigiSign cache**. In other words, DigiSign pops-up PIN2 dialog only then card needs it to be able to execute desired cryptographic operation.

Please be aware that there are cases when card remembers the PIN2 code and allows user to execute nonrepudiation key operation without asking PIN2 code:

- Some smart cards enables PIN2 code when PIN2 code is unlocked or changed
- PIN2 is introduced by using
 - DigiSign Application GUI
 - Cryptoki, minidriver or toolkit interface
 - Some other application or interface

2.5 Notes for Cryptoki application users

You can find Cryptoki, the PKCS#11 module named `cryptoki.dll`, in the installation directory. Installation directory location depends on used platform, according table below.

Cryptoki location in different platforms	
Platform	Path
Windows	C:\Program Files\Fujitsu\mPollux DigiSign Client\cryptoki.dll
32bit application path in 64bit Windows operating system	C:\Program Files (x86)\Fujitsu\mPollux DigiSign Client\cryptoki.dll
MacOS	/Library/mPolluxDigiSign/libcryptoki.dylib
Linux	/usr/lib/libcryptoki.so

2.6 Notes for Cryptoki developers

By default, the secondary authentication mechanism (auto-login) is enabled. If auto-login causes logical application errors, you can disable it by setting the value of the `disableCryptokiAutoLogin` registry key or environment variable to 1.

2.7 Notes for users who personalize cards with Cryptoki

By default, Cryptoki defines two PIN codes; user PIN and security officer's PIN.

In many cases there is a need to personalize more enhanced profiles than Cryptoki-interface allows. Therefore it is good idea to take a look of "DigiSign_Toolkit", because it offers better tools for card data content manipulation.

However, if Cryptoki is mandatory requirement, there are following alternatives to add more expression power to initialization procedure:

- Used initialization script can be introduced via registry setting (`cryptokiScript`) or
- passing structured tokenName string when calling `C_InitToken()`
 - String may contain following components that are separated from each other with pipe '|' sign (ASCII 124)

- Example string; "tokenName|script|PUK1|PIN2|PUK2|SOPUK|cardSN"
- Initialization script definition is outside of this scope. Please ask more details from your software vendor.

2.7.1 How to generate key that is protected with signature PIN (PIN2)

DigiSign Client's Cryptoki module creates own slot for PIN2 protected objects, like nonrepudiation key. Please look "disableCryptokiPIN2Slot" registry setting for more details.

- Depending defined access conditions:
 - Open RO or RW session to PIN1 protected slot
 - Open RW session to PIN2 protected slot
 - Open RW-SO session to PIN2 protected slot
- Generate or import key into PIN2 protected slot

2.8 Notes for Citrix users

The following setting should be verified:

- Define the path to the smart card cached and set full rights to all users (`SmartCardCachePath=<set path>`)
- When publishing applications, "DigiSign Application" should be published to the same desktop. Accessing card gets faster and certificate loading to certificate storage gets more precise (see chapter "Certificate Propagation Service").
- DigiSign Application should be closed when published application is closed.

For more details about settings, see Chapter 3, "DigiSign Client settings".

2.8.1 Chromium based Edge on Windows Server 2016

By default smart card access doesn't work properly. To enable smart card access, following registry key needs to be added;

HKEY_LOCAL_MACHINE\SOFTWARE\citrix\CtxHook\AppInit_DLLs\SfrHook\msedge.exe

Please see; <https://support.citrix.com/article/CTX296999>

2.9 Notes for MacOS users

MacOS M1 architecture support has been added to the version 4.1.4e and DS Client works seamlessly with older and newer architecture computers. Older DS Client versions works with M1 computers only in "intel" mode, and doesn't co-operate with for example browsers that are running in "apple" mode.

Starting from version 4.1.0, deprecated "tokend"-interface has been replaced with Apple's currently supported smart card interface, "CryptoTokenKit".

From the user's point of view, the biggest change is that smart card certificates cannot be seen in the keychain's graphical user interface. However, if there is a need to see if CryptoTokenKit is properly installed, following commands may be used:

1. Open terminal
2. Write "pluginkit -vv -m -p com.apple.ctk-tokens"
 - a. Installed plugins are listed
3. List smart cards and certificates
 - a. "system_profiler SPSmartCardsDataType"
4. User pairing to the inserted smart card
 - a. Query status: "sc_auth pairing_ui -s status"
 - b. Enable pairing: "sc_auth pairing_ui -s enable"
 - c. Unpair inserted card: "sc_auth unpair"
 - d. List paired cards: "sc_auth list"

2.10 Notes for Linux users

To enable SSL/TLS connection to SCS signing and diagnostic services, the DigiSign Application must create a local trusted certificate chain. If the certificates are not trusted, web browsers will not allow connections to the local web server. Therefore, it is necessary to add a local root certificate named "Fujitsu DigiSign Local Root-CA certificate" as a trusted root certificate.

Different web browsers use different root certificate stores. Some browsers, such as Chrome, use trusted user certificates from the Network Security Services (NSS) user store located in the ~/.pki folder. However, latest Firefox versions use certificate storage under ~/.mozilla folder.

DigiSign Client installer updates both storages during installation and asks user password if repositories are password protected.

3 DigiSign Client settings

This section describes the settings that can be used to change the behavior of DigiSign Client. The settings can be found in the following locations:

- Windows registry settings:
 - Registry keys in Windows 32-bit operating systems:
 - HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\DigiSign Client
 - Registry keys in Windows 64-bit operating systems:
 - 64-bit applications:
 - HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\DigiSign Client
 - 32-bit applications:
 - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Fujitsu\DigiSign Client
- File in Linux: /etc/xdg/Fujitsu/CCryptoLib.conf
- File in Mac OS: /Library/Preferences/com.fujitsu.CCryptoLib.conf

DigiSign Client and mPollux Service Manager settings

Key	Default value	Description
aboutDialogBitmap	Empty	Default path to the image in the About dialog and tab.
allowUnknownCards	1	If set to 0, parse only known cards
buildReaderListTimer	5	The interval in seconds in which DigiSign polls the smart card. If you are using DigiSign through a slow remote connection or a thin client, increase the value to reduce the polling frequency.
certExpirationWarningDays	30	Warn about certificates that will expire within defined count of days
changePinCacheCount	5	How many old PIN codes are stored (if PIN expiration feature is turned on)
changePinInitialPeriod	0	If defined and PIN expiration object is defined at the first time, PIN should be changed according this definition. Unit is days.
changePinPeriod	0	How often PIN should be changed (days). If set to '0', PIN expiration information is not stored into card
closeBrowsers	0/1	Depending version; Tries to close web browser when card is removed from the reader
closeBrowsersConfirm	0	If closeBrowsers is set, ask confirmation form the user before closing the browsers
closeBrowsersExcludeReader	0	Do not close browsers if received event comes from this reader. Same wildcards are allowed than in 'excludeReader'
closeGUIServerWhenLock	1	If defined, GUI server is closed when screen is locked
cryptokiPinCache	1	If defined, PIN cache is used from cryptoki- interface. This setting has no effect if 'disableCryptokiAutoLogin' is set
cryptokiScript	Empty	Overwrite card initialization script.
cryptokiSoftwareKeyGen	0	If defined, key generation is performed by software - not by card
DataPath	Depends on the operating system	Linux and Mac only: The pipe path.
disableAuthTab	0	If set to 1, the Authentication tab in DigiSign Manager is hidden.
disableCryptokiAutoLogin	0	If set to 1, the DigiSign PIN dialog is disabled. The user is shown the program's own dialog with the program's own functionality.
disableCryptokiBufferCheck	0	If set to 1, the set buffer size for cryptoki may be exceeded.
disableCryptokiPIN2Slot	0	By default, cryptoki creates two slots for PIN codes, one for Pin 1 and another for PIN2. If set to 1, PIN 2 is disabled.
disablePinUnblocking	0	If set to 1, PIN unblocking is disabled from DigiSign Application's GUI.
dropAuthentication	0	If set to 1, the user must re-authenticate after each asymmetric key operation. Please note that all cards doesn't support this feature.
dropAuthenticationTimeout	30	If set, defines timeout in minutes when unused PIN cache is cleared. Default value is 30 minutes.
excludeReader	Empty	A list of those PC/SC (Personal Computer/Smart Card) readers that

		DigiSign should ignore. This list enables DigiSign to be used in computers that have several card readers. In the list, wild cards are allowed; for example, *O2* *CCID* ignores all readers that contain the substring O2 or CCID.
forceSelectApplet	0	If set to 1, DigiSign always selects the EID applet at startup.
HTTPSignerEnabled	0	Older web signer (HTTP-Signer) interface is disabled by default (starting from version 4.2.6). Look scsSignerEnabled to enable/disable newer SCS web signer interface.
HTTPToolkitEnabled	0	If set to 1, enables the httpToolkit interface that can be used for creating spare cards on work stations.
Language	Empty	Sets the language for the user interface. The possible values are fi (Finnish), en (English), and sv (Swedish).
managerBannerBitmap	Empty	The path to the default banner image in mPollux Service Manager.
minidriverAuthMode	1	If set to 1, minidriver uses DigiSign dialog. Otherwise Window's own dialog is presented.
minidriverAutoAuthExclude	0	Applications matching this expression are not allowed to use DigiSign's PIN dialog resulting that Windows' own PIN dialog is used. Expression format is the same than with 'exclude reader'
minidriverPinCache	1	If set to 1, PIN1 value is stored to cache to avoid unnecessary PIN requests. PIN cache is cleared when card is removed from the reader.
minidriverReadOnly	0	If set to 1, minidriver acts as 'RO minidriver'
logonDomain	Empty	Windows minidriver setting; View only certificates that contains defined domain in UPN. Domain name must be entered with '@', like for example "@domain.com"
pinComplexityCheck	0	Bit mask that defines what kind of PINs are accepted; bit 0 = no same numbers, 1 = no followed numbers, 2 = avg diff must be <=4, 3 = only numbers, 4 = PIN must contain lowerCase, upperCase and digit
pinDialogBitmap	Empty	The path to the default image in the PIN dialog.
pinIgnorePinTypeCheck	0	If defined, pin type restrictions are ignored (including pinNumericValuesOnly)
pinMinLength	0	If defined, this setting overrides definition read from smart card
pinNumericValuesOnly	0	If defined, this setting overrides definition read from smart card
removeAllCertificatesOnEmptyEvent	1	If set to 1, certificates are removed from certificate storage when last smart card is removed from the card reader(s).
scsSignerEnabled	1	If set to 1, web signer interface is enabled. Please see more details how web signer interface works from https://dvv.fi/fineid-maaritykset
showAsn1CertInWindows	0	If the operating system supports showing the contents of the certificate, the contents are shown in the format set by the operating system. If this value is set to 1, the certificates are always shown in ASN.1 format, regardless of the operating system capabilities.
SmartCardCachePath	Depends on the operating system	The path to the temporary directory for the smart card cache files.
SmartCardSNCache	3	Smart card caching enabling flag:

		<p>SmartCardSNCache = 0 - - Cache is disabled. It is not recommended to use this setting because it decreases performance.</p> <p>SmartCardSNCache = 1 Smart card data content (cache file) is stored in plain text format.</p> <p>SmartCardSNCache = 2 Cache file is stored in encrypted format with fixed key. Use this setting when there is a need to share same cache between different computers.</p> <p>SmartCardSNCache = 3. Recommended value(set during imnstallation); Cache file is stored in encrypted format with computer specific key. Cache file can't be used with another computer.</p>
SplashBitmap	0	Path to bitmap that is shown during DS application startup
strictSCSkeyPolicy	0	SCS web signer option; If "keyalgorithms" is defined with one key type and card doesn't contain such keys for requested operation: 0 => allow signature operations with other key types 1 => signature requesting operation is cancelled If 'strictKeyPolicy' is set in SCS request, it overwrites this selection.
toolkitPinCache	1	If set, PIN cache is allowed to DS Toolkit
translateCardLabels	1	If set to 0, display card labels as they are written to the smart card. If set to 1, display translated labels based on selected language settings.
userLevel	0	If set to 0, all advanced features are disabled. Possible values are 0, 1, 2 and 3. userLevel > 0: Remove object + Compute PIN/PUK challenge userLevel > 1: PKCS#12 import userLevel > 2: Card initialization features
validTokenInfoLabel	0	If set, accept only those cards that label's meets this expression. Expression format is the same than with 'exclude reader'
loadCertificates	1	If set, DigiSign application will load and clean card certificates when card is inserted. Otherwise, minidriver and Certification Propagation service is responsible for loading certificate.
pdfOnlyDS	0	If set, only certificates with digital-signature key-usage can be selected.
pdfOnlyNonRep	1	If set, only certificates with non-repudiation key-usage can be selected.
pdfAddRevocationInfo	1	If set, revocation info is added into signature (OCSP-response or CRL)
pdfRevocationList	1	If set, CRL is added into revocation info if OCSP-response can't be requested
pdfRevocationListMaxSize	(not set)	CRL maximum size in kilobytes.
pdfSignatureSubFilter	1	0 = adbe.pkcs7.detached 1 = ETSI.CAdES.detached
pdfTimeStampServer	(not set)	URL of RFC-3161 compliant time-stamp server. If set, timestamp of signing time is requested into document to support long-term pdf signatures.
pdfLockDocument	0	If set, add certification instead of signature and mark document locked. In other words, appending new content is forbidden, including additional signatures.
versionValidation	1	If set, DigiSign application will check if the newest version is available on the download site. Version validation depends from the user preferences from version validation dialog.

pdfAllowMultiselect	0	If set, multiple pdf documents selection for signature is available in Windows explorer.
cryptokiDisabled	0	Disable using cryptoki token library in the system.
scsPinPolicy	0	If set to "1" DigiSign doesn't take pin from cache.
enableHttpProxy	0	Enable http proxy for authentication

3.1 Optional settings

These settings are not set by application installer but may be used in certain use-case scenarios:

Key	Default	Description
gpMasterKey	OS specific default key	Global platform master key. Used while initializing GP SCP03 compatible cards.
gpCardManager	OS specific default AID	Global platform card manager AID (ascii-hex encoded string)
scCVCert	Default	Base64 encoded CVC certificate to be stored into card during initialization.
scCVCSignerKey	Default	CVC issuer's public key to be stored into card during initialization. Base64 encoded PKCS#1 or PKCS#8 formatted public key.
scTerminalKey	Default	Terminal side private key to be used during external authenticate. Stored as ASCII-HEX encoded string.
scMaxRsaObjectSize	OS specific maximum size	Depending on card operating system. Usually 2048 or 4096.
activationDialogWallpaper	Default	Activation dialog's background image.
activationDialogTextColor	white	Activation dialog's text color. Suggested values 'black' and 'white'
translateCardLabels	"1"	If set to "1", card labels like PIN, key and certificate labels are translated to the selected GUI language. If set to "0", labels are displayed as they appear in the card application.
executeInsert	Command line to execute, for example: <pre>echo INSERT %time% "\$readername\$" >> test.txt</pre>	Execute user defined command line operation when the card has been inserted to the reader. Available pre-defined macro is; <ul style="list-style-type: none"> readername = used smart card reader name
executeReady	Example: <pre>echo READY %time% "\$readername\$" "\$cardtype\$" "\$serialnumber\$" >> test.txt</pre>	Execute user defined command line operation when the card content reading is done. Available pre-defined macros are; <ul style="list-style-type: none"> readername = used smart card reader name serialnumber = smart card SN cardtype = smart card type
executeRemove	Example: <pre>echo REMOVE %time% "\$readername\$" >> test.txt</pre>	Execute user defined command line operation when the card has been removed from the card reader. Available pre-defined macro is; <ul style="list-style-type: none"> readername = used smart card reader name
pdfHideFromMenu	"0"	If set to "1", .pdf document signing and validation menu entries are removed from the context menu and explorer plugin.

3.2 Web signer technical notes

DigiSign Client implements two different interfaces for web signing:

- Older, HTML-based WebSigner
- Newer, HTML5-based SCS-signer

Both interfaces follows specifications that are publicly available from <https://dvv.fi/en/fineid-specifications>

3.2.1 SCS signer additions

By default, SCS V1.1 defines optional "signatureType" that specifies two signature types

- "signature" – that produces "raw" signature of given data or digest
- "cms" – resulting attached or detached cms signature

In addition to default signature types, DigiSign Client's SCS interface supports following signature types:

- "pkcs7" – compliant with "cms" definition
- "xml" – computes xmldsig-signature and returns xml-document that includes given data inside signature-tag
- "cms-pades" – When contentType is "digest" and signatureType is "cms-pades", DigiSign Client generates pades-compliant signature without timeStamp.

3.2.2 Running SCS on a server with multiple desktops

Please be aware that SCS is designed for single desktop systems. Running SCS on a server that offers multiple desktops doesn't work for one simple reason: SCS interface is available from two fixed TCP/IP ports, 53951 for http and 53952 for https. Ports are opened from desktop application (DSApp) that offers graphical user interface that is linked to web signing interface.

In other words, due technical limitations of TCP/IP port, it can be opened for listening from only one desktop at time, when multi-desktop system requires multiple simultaneous connections.

One solution is Fujitsu Finland's "SCS Broker" that forwards signature requests to correct desktop. Please ask from your dealer for more details about computing web based signatures with multi-desktop environments.

If signing services are not needed, it is suggested to turn them off by setting "HTTPSignerDisabled" from "0" to "1".

3.3 Technical note about "dropAuthentication"

DigiSign settings describes a feature called "drop authentication". It means that smart card remembers some pin codes, typically PIN1 (basic PIN) as far as card has been reset.

In some critical use cases, there is a need to reset PIN code after each key operation, like signature or decrypt. From technical point of view, drop authentication is implemented by re-selecting card EID-application, that typically reset the EID-application. Alternative implementation would be to reset smart card via smart card driver. Unfortunately this method doesn't work "directly". In other words, even if application tells smart card driver to reset the card, it doesn't necessary act as requested. Therefore re-selecting EID application works better, as far as card itself resets the state of EID application.

Unfortunately all cards doesn't support this feature. Based on practical tests, following cards supports this feature;

- FinEID (2016,2022,2023)
- Aventura MyEID cards
- Gemalto/Thales IDPrime 940

Following cards doesn't support this feature;

- IDEMIA ID.ME (issued 29.8.2019)
- IDEMIA ID.ME (temporary card)

4 DigiSign Toolkit

The DigiSign Toolkit is a C interface that is included in the DigiSign Client Windows installation package. The Toolkit allows you to program your own system to use DigiSign Client. The Toolkit provides functions such as the following:

- Searching for user certificates
- Computing and verifying digital signatures
- Authenticating against the mPollux Server
- Transmitting Certificate Management Protocol (CMP) messages to different Certificate Authority (CA) systems

After default installation in Windows environment, you can find the Toolkit in the following directory:

```
C:\Program Files\Fujitsu\mPollux DigiSign Client\Toolkit
```

For more information on the Toolkit, see the `DigiSign_Toolkit.h` file in that directory.

5 Windows specific notifications

This chapter includes windows specific notifications that can be ignored while using other platforms.

5.1 Smart Card Minidriver

Windows security interfaces (CryptoAPI and Microsoft Cryptography API: Next Generation (CNG)) needs *provider* to connect smart card to existing security infrastructure. Older DigiSign Client versions (from 1.0 – 3.2.X) included so called “*CSP module*” to access keys and certificates from smart card.

Because of the need of new cryptographic features, “*Windows Smart Card Minidriver*” was introduced first time in DigiSign Client version 3.5.0. Starting from version 4.0.0, CSP is no longer included to installation package. Current version follows “Windows Smart Card Minidriver Specification” with level 6.

There are two remarkable differences between 3.X and 4.X versions:

- 1) How certificates are moved from card to certificate storage and
- 2) How correct interface is introduced to the operating system when card is inserted.

5.1.1 Certificate Propagation Service

Certificate Propagation Service is native service of Microsoft Windows operating system. It monitors smart cards when new card is found, it loads certificates to certificate repository.

However, there are two major difficulties with CertPropService:

- 1) It doesn't introduce “Friendly Names” that makes it difficult to user to figure out what certificate is requested
- 2) It doesn't remove certificates from certificate storage, that causes practical difficulties when card is not present.

DigiSign Application tries to fix situation as follows:

Versions 4.0.0 ... 4.0.14;

- Relies on CertPropService. If service is turned off, certificates are not loaded
- When all cards are removed from the readers, DSApp removes certificates from certificate storage

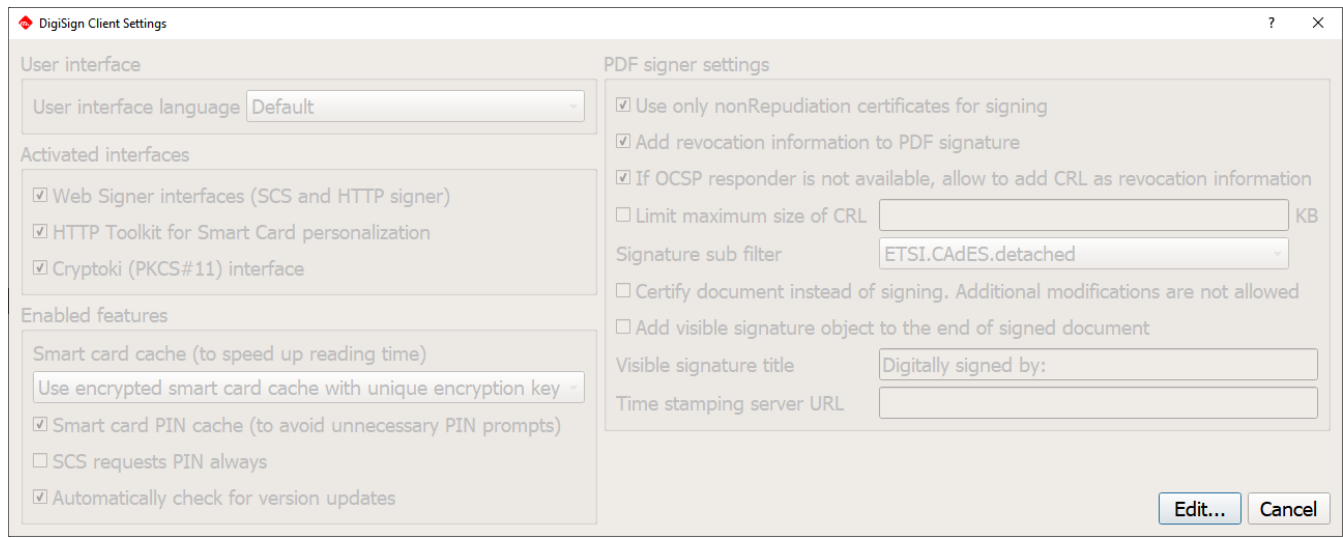
Versions 4.0.16 ...

- CertPropService is not used; DigiSign Application introduces certificates with friendly name
- When card is removed, that card's certificates are removed from certificate storage
- When application is closed, all card associated certificates are removed from certificate storage

6 Options-dialog

Windows and MacOS operating system offers the possibility to change some settings with the help of "Options" dialog. Dialog can be opened from DigiSign Application's taskbar menu and it offers possibility to change certain settings. Changing settings is enabled after entering "elevated" user credentials when clicking "Edit..." button.

Windows ".exe" installer displays the same dialog and allows you to modify settings during the installation of the application.



Options dialog features

Web Signer interfaces	When this feature is turned on, DigiSign application allows web browsers to access local web server to request signatures. Local webs server supports web based signature requests based on Web Signing- and SCS V1.2 interface. Please see https://dvv.fi/en/fineid-specifications
HTTP Toolkit	HTTP Toolkit enables possibility to personalize cards over http interface. There is no need to turn on this interface unless card personalization services are used.
Cryptoki interface	Cryptoki (PKCS#11) is a platform independent interface for cryptographic tokens or devices. The reason why this interface exist is that even Windows and MacOS operating systems contains own cryptographic framework, there are applications that doesn't support them. For example, Firefox browser is good example of such application. Please see more details about Cryptoki and Firefox from User's Guide.
Smart card cache	Reading card content is quite slow procedure and it might take up to 30 seconds. To speed up card reading, card data content can be stored to cache file. There are number of different options how to handle cache files. Please see chapter "DigiSign Client settings" for more details.
Smart card PIN cache	Typically, smart card remembers PIN1 code until card is reset. However, depending on operating system version, card might be reset "too often" causing annoying PIN prompting. To make application more user friendlier, it is possible to use PIN1 cache that introduces PIN code to card on demand.

	Please be aware that quite often user friendliness is opposite of information security. Therefore, to avoid possibility for misuse, card should be removed from the reader every time when it is not used.
SCS requests PIN always	To add information security, it is possible to define SCS interface to request PIN code even in the cases when card remembers the code. If this feature is turned on, PIN is asked every time even if smart card PIN case is turned on.
PDF Signer settings	Please find the details from chapter 3, DigiSign Client Settings

6.1.1 Smart Card Plug-and-Play Service

With older DigiSign Client versions (1.0. ... 3.5.X), CSP module was introduced by installation package and there was no need to do anything else to get correct driver loaded.

Starting from version 4.0.0, minidriver is installed as plug-and-play driver and operating system should load the correct driver when card is inserted. In other words, SCPnP stands for "Smart Card Plug-and-play" service and it is used to install correct minidriver when smart card is inserted to smart card reader.

However, when using Windows Server versions, ScPNP is not supported by operating system. Therefore, when installing versions 4.0.0 ... 4.0.14:

- Minidriver must be installed manually
 - Manual installation can be done in two ways;
 - Double-clicking .inf file that can be found from installation folder or
 - By adding "Legacy Hardware" under "Device Manager"

Versions 4.0.16. ...

- Installer notices if used platform is Microsoft Server
 - Minidriver is installed automatically

7 Web browser SSL/TLS Client Authentication challenges

Quite often SSL/TLS client authentication is associated to be similar method than other web authentication method, like username/password, digest or Windows-SSO authentication. Therefore, it is easy to make false expectation that web application's session is the only storage where user credentials are stored.

Setting up a secure socket session between a client and a server is the most time-consuming step in secure socket handshaking protocol and may take several hundreds of milliseconds to complete. On the other hand, a modern web browser opens multiple simultaneous secure socket sessions against the server to allow the page content to be displayed to the user as quickly as possible. Therefore, renegotiating the shared secret for each http request would make the browsing experience extremely slow.

To fix this problem, SSL/TLS offers mechanism to store the result of slowly negotiated "master secret" and when client opens new session it can just refer previously negotiated session ID with the result of much faster key agreement procedure.

Talking about SSL/TLS level client authentication, handshaking procedure is almost the same compared to the first time handshaking procedure. Only difference is that during handshaking, server requests the client to proof its identity. In DigiSign Client case, proving is done by performing cryptographic operation inside the smart card. If server successfully validates client's identity, "master secret" is shared and server associates authenticated user's certificate into newly created SSL/TLS session.

Here comes the most important part: Authenticated session is stored in the same way like any other SSL/TLS session and when the web browser requests new resource from server, it just refers to SSL/TLS session ID instead of running cryptographic operation inside smart card for every SSL/TLS request.

Web application can request the status of secure socket session, among authenticated user's certificate. Therefore, it is quite possible that session status is stored also inside the web application's session. In other words, authenticated user information may be stored in two different places: inside the secure socket session **AND** the web application session.

Therefore, web application developer should understand following session handling related details, before spending a lot of time while trying to figure out why new session remembers previous session credentials:

- Browser remembers old SSL/TLS sessions, most likely as far as it is running
 - Browser most likely does not forget secure socket sessions when
 - smart card is removed or changed in the reader,
 - browser has crashed and restarted or
 - sometimes even in the case when browser closed on purpose and restarted by the user.
- Server does not clear SSL/TLS sessions automatically when user logs out.
 - SSL/TLS session cache needs to be cleared by web application explicitly.
 - If SSL/TLS session cache is not cleared, a new session may end up to the situation where authentication is not required and a new user gets logged to the system with previous user credentials.

Contact

FUJITSU FINLAND OY
Address: PL 100, 00012 FUJITSU
Phone: +358 29 302 302
Website: www.fujitsu.com/finland

© Copyright 2015 Fujitsu, the Fujitsu logo are trademarks or registered trademarks of Fujitsu Limited in Japan and other countries. Other company, product and service names may be trademarks or registered trademarks of their respective owners. Technical data subject to modification and delivery subject to availability. Any liability that the data and illustrations are complete, actual or correct is excluded. Designations may be trademarks and/or copyrights of the respective manufacturer, the use of which by third parties for their own purposes may infringe the rights of such owner.