A-Sign TSE API Developer Manual (asignTSE.EXE Edition)

Date	Revision	Author	Changes
23/06/2020	1.0.0.0	Thomas Schützenhöfer	EXE description
25/03/2021	1.1.1.0	Thomas Schützenhöfer	update logging, qrcode, add functions
30/03/2021	1.1.1.2	Thomas Schützenhöfer	add description for additional functions
17/08/2021	1.1.2.2	Thomas Schützenhöfer	add genqrcodecustomsize function

Introduction

A-Trust provides a shared library for using **a.sign TSE Online**, which provides the neccessary functionality required for German Kassensicherheitsverordnung. The **A-Trust Middleware asignTSE.EXE** provides an easy to use On- and Offline solution.

Installation

The library must be placed in the same directory as the executable **(asignTSE.EXE)**. The provided configuration file must be placed in the working directory of the executable.

The data returned will temporarily be written to asigntseEXE_out.txt. An example how parse and use the data in asigntseEXE_out.txt is provided in asigntseBAT.bat.

Logging

- asigntseEXE build in logging:
  - The asigntseEXE.exe will log the asigntseEXE_date.txt.
- The configuration for the login will be added in a later update.
- asigntseBAT:
  - The asigntseBAT.bat write its own logfile to asigntseEXE_log.txt

    (Example: Provided in asigntseBAT.bat)

WithTse function family

Most functions have a WithTse version with additional parameters `tseId`. This parameter specifies the config entries, the library should use for the call. Functions without the `tseId` parameter use the `[default]` section. To see all function variants, have a look at Appendix A.

Preparation

After installing the setup, copy the following files to the preferred folder.

1. **AsigntseEXE.zip:**
    - Required: asigntseEXE.exe
    - Optional: asigntseBAT.bat, errortext.txt
2. **Setup directory: (specified in the setup) - copy all files**
    - .\A-Trust GmbH\a.sign SMAERS\bin
    - .\A-Trust GmbH\a.sign SMAERS\asigntseonline.conf

To test the asigntseEXE you can open a cmd.exe and navigate to the folder of the asigntseEXE. Then run **asigntseEXE.exe help** to display all possible functions. The next step is to initialize the dll.

Initialization

Lifecycle State of the TSE:

0 = Unknown

The initial state of the `TSE`.

1 = NotInitialized

After installing the state is set to the `NotInitialized`.

2 = Active

If initializeDescriptionSet or initializeDescriptionNotSet is called by the user in the administrator role, the `TSE` enters the `Active` state.

3 = Suspended

If at_suspendSecureElement/at_unsuspendSecureElement is called by the user in the administrator role, the `TSE` enters/exits the `Suspended` state.

4 = Disabled

If disableSecureElement is called by the user in the administrator role, the `TSE` exits the `Active` state and enters `Disabled` state.

The following example shows how to initialize the `TSE`.

Example

```
asigntseEXE.exe at_getLifecycleState --cfgsetloggingenabled true
RETURN : 1

asigntseEXE.exe initializeDescriptionSet --authenticateUser Admin AdminPW --
cfgsetloggingenabled true

asigntseEXE.exe at_getLifecycleState --cfgsetloggingenabled true
RETURN : 2

... Initialization completed

asigntseEXE.exe at_registerclientid 1
asigntseEXE.exe at_registerclientid 2

asigntseEXE.exe starttransaction 1
...
...
```

NOTE: AdminPW is defined during the Setup.

NOTE: --cfgsetloggingenabled true is optional, but can be useful for troubleshooting.

Function overview

Transactions:

- startTransaction
- startTransactionWithTse
- finishTransaction
- finishTransactionWithTse

Data export Calls:

- exportData
- exportDataWithTse
- exportDataFilteredByTransactionNumberAndClientId
- exportDataFilteredByTransactionNumberAndClientIdWithTse
- exportDataFilteredByTransactionNumber
- exportDataFilteredByTransactionNumberWithTse
- exportDataFilteredByTransactionNumberInterval
- exportDataFilteredByTransactionNumberIntervalWithTse
- exportDataFilteredByTransactionNumberIntervalAndClientId
- exportDataFilteredByTransactionNumberIntervalAndClientIdWithTse
- exportDataFilteredByPeriodOfTime
- exportDataFilteredByPeriodOfTimeWithTse
- exportDataFilteredByPeriodOfTimeAndClientId
- exportDataFilteredByPeriodOfTimeAndClientIdWithTse
- exportCertificates
- exportCertificatesWithTse

- exportSerialNumbers
- exportAllAndDeleteStoredData
- exportSerialNumbersWithTse
- readLogMessage
- readLogMessageWithTse

Administrative Calls:

- Lifecycle
  - disableSecureElement
  - disableSecureElementWithTse
- UserManagement
  - initializeDescriptionSet
  - deleteStoredData
  - deleteStoredDataWithTse
- Statistics
  - getMaxNumberOfClients
  - getMaxNumberOfClientsWithTse
  - getCurrentNumberOfClients
  - getCurrentNumberOfClientsWithTse
  - getMaxNumberOfTransactions
  - getMaxNumberOfTransactionsWithTse
  - getCurrentNumberOfTransactions
  - getCurrentNumberOfTransactionsWithTse
  - getSupportedTransactionUpdateVariants
  - getSupportedTransactionUpdateVariantsWithTse

Additional Functions:

- genqrcode
- genqrcodecustomsize
- genqrcodefromstring
- genqrcodefromstringcustomsize
- convertStringToBase64
- convertUnixTimeToUTC

A-Trust API Functions

The following functions are not defined in the SE-API specification. They are additional functions specific to the A-Trust implementation.

- at_getVersion
- at_getSignatureAlgorithm
- at_getSignatureAlgorithmWithTse
- at_getPublicKey
- at_getPublicKeyWithTse

- at_getOpenTransactions
- at_getOpenTransactionsWithTse
- at_getSignatureCounter
- at_getSignatureCounterWithTse
- at_getTransactionCounter
- at_getTransactionCounterWithTse
- at_getLifecycleState
- at_getLifecycleStateWithTse
- at_getSerialNumber
- at_getSerialNumberWithTse
- at_suspendSecureElement
- at_suspendSecureElementWithTse
- at_unsuspendSecureElement
- at_unsuspendSecureElementWithTse

Config API Functions

The following functions are also specific to the A-Trust implementation and are used to set the configuration of the library. The config functions are used as paramter.

- --cfgSetConfigFile
- --cfgTseAdd
- --cfgTseRemove
- --cfgSetLoggingEnabled
- --cfgSetLoggingStderr
- --cfgSetLoggingFile
- --cfgSetLogDir
- --cfgSetLogLevel
- --cfgSetLogAppend
- --cfgSetLogColors
- --cfgSetLogDetails
- --cfgSetLogStderrColors
- --cfgSetHttpProxy

Transactions

StartTransaction:

The function startTransaction must be called at the beginning of each transaction and finishTransaction is called when the transaction is finished.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
clientId	in	REQUIRED	Represents the ID of the application that has invoked the function.
transactionNumber	out	-	Represents a transaction number that has been assigned by the Secure Element to the process.
logTime	out	-	Represents the point in time of the Secure Element when the log message was created. Time is given in UTC.
signatureValue	out	-	Represents the signature value. (hex)
signatureValueB64	out	-	Represents the signature value. (base64)
serialNumber	out	-	Represents hash value over the public key of the key pair that is used for the creation of signature values in transaction log messages.
signatureCounter	out	-	Represents the current value of the signature counter.

Example

```
asigntseEXE.exe starttransaction "clientId"

asigntseEXE.exe starttransaction Kasse_1
```

FinishTransaction:

- call finishTransaction and provide:
  - `clientID`
  - `transactionNumber` received from startTransaction
  - `processType` = depends on the type of transaction eg. Beleg -> processType = `Kassenbeleg-V1`
  - `processData` = depends on the type of transaction see Beleg or Bestellung or SonstigerVorgang
- finishTransaction returns:
  - `transactionNumber` accociated startTransaction
  - `logTime` (eg.: 2019-07-20T09:11:24:000Z) - all dateTime values UTC
  - `signatureValue` (hex)
  - `signatureValue` (b64)
  - `signatureCounter` (eg. 4711)
- This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
clientId	in	REQUIRED	Represents the ID of the application that has invoked the function.
starttransactionNumber	in	REQUIRED	Parameter is used to unambiguously identify the current transaction.all the necessary information regarding the initial state of the process.
processData	in	REQUIRED	Represents all the information regarding the final state of the process.
processType	in	OPTIONAL	Identifies the type of the transaction as defined by the application. This string MUST NOT contain more than 100 characters.
transactionNumber	out	-	transactionnumber of the associated starttransaction
logTime	out	-	Represents the point in time of the Secure Element when the log message was created.
signatureValue	out	-	Represents the signature value. (hex)
signatureValueB64	out	-	Represents the signature value. (b64)
signatureCounter	out	-	Represents the current value of the signature counter.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_START_TRANSACTION_FAILED	The execution of the Secure Element functionality to start a transaction failed.
ERROR_RETRIEVE_LOG_MESSAGE_FAILED	The execution of the Secure Element functionality to retrieve log message parts has failed.
ERROR_STORAGE_FAILURE	Storing of the log message failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_TIME_NOT_SET	The managed data/time in the Secure Element has not been updated after the initialization of the SE API or a period of absence of current for the Secure Element.
ERROR_CERTIFICATE_EXPIRED	The certificate with the public key for the verification of the appropriate type of log messages is expired. Even if a certificate expired, the log message parts are created by the Secure Element and stored by the SE API.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

Example

```
asigntseEXE.exe finishtransaction "clientID" "startTransactionNumber"
"processType" "processData"

asigntseEXE.exe starttransaction Kasse_1 21 Kassenbeleg-V1
Beleg^75.33_7.99_0.00_0.00_0.00^10.00:Bar_5.00:Bar:CHF_5.00:Bar:USD_64.30:Unbr
```

Data export Calls:

exportDataFilteredByTransactionNumberAndClientId:

Exports the transaction log messages, containing the process and protocol data, that are relevant for a certain interval of transactions. The transaction log messages in this interval correspond to the passed clientId.

Additionally, the function exports all system log messages and audit log messages whose signature counters are contained in the interval.

Furthermore, additional files are exported that are needed to verify the signatures in the log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
transactionNumber	in	REQUIRED	Defines the transaction number (inclusive) regarding the start of the interval of relevant log messages.
clientId	in	REQUIRED	ID of a client application that has used the API to log transactions. Only transaction log messages that corresponds to the clientId are relevant for the export.
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only returns the log messages if the number of relevant records is less or equal to the number of maximum records. If the value of the parameter is 0, the function returns all selected log messages.
exportedData	out	-	Selected log messages and additional files needed to verify the signatures included in the log messages.
tseId	in	(REQUIRED)	ID of the TSE to use in a multi-TSE environment.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_PARAMETER_MISMATCH	Mismatch in parameters of the function.
ERROR_TRANSACTION_NUMBER_NOT_FOUND	No data has been found for the provided transaction numbers.
ERROR_ID_NOT_FOUND	No data has been found for the provided clientId.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportDataFilteredByTransactionNumberAndClientId
"transactionnumber" "clientId" "maximumNumberRecords"
asigntseEXE.exe exportDataFilteredByTransactionNumberAndClientId 123 Kasse_1 0

asigntseEXE.exe exportDataFilteredByTransactionNumberAndClientIdWithTse
"transactionnumber" "clientId" "maximumNumberRecords" "tseId"
asigntseEXE.exe exportDataFilteredByTransactionNumberAndClientId 123 Kasse_1 0
TSE_1
```

exportDataFilteredByTransactionNumber

Exports the transaction log messages, containing the process and protocol data, that correspond to a certain transaction.

Additionally, the function exports all system log messages and audit log messages whose signature counters are contained in the following interval:
Signature counter of the transaction log message for the start of the transaction and the signature counter of the transaction log message for the end of the transaction (inclusive)

Furthermore, additional files are exported that are needed to verify the signatures in the log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
transactionNumber	in	REQUIRED	Indicates the transaction whose corresponding log messages are relevant for the export.
exportedData	out	-	Selected log messages and additional files needed to verify the signatures included in the log messages.
exportedDataLength	out	-	Length of the array that represents the exportedData.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_TRANSACTION_NUMBER_NOT_FOUND	No data has been found for the provided transactionNumber.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportDataFilteredByTransactionNumber "transactionnumber"
"maximumNumberRecords"
asigntseEXE.exe exportDataFilteredByTransactionNumber 123 0

asigntseEXE.exe exportDataFilteredByTransactionNumberAndClientIdWithTse
"transactionnumber" maximumNumberRecords" "tseId"
asigntseEXE.exe exportDataFilteredByTransactionNumber 123 0 TSE_1
```

exportDataFilteredByTransactionNumberInterval

Exports the transaction log messages, containing the process and protocol data, that are relevant for a certain interval of transactions.

Additionally, the function exports all system log messages and audit log messages whose signature counters are contained in this interval.

Furthermore, additional files are exported that are needed to verify the signatures in the log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
startTransactionNumber	in	REQUIRED	Defines the transaction number (inclusive) regarding the start of the interval of relevant log messages.
endTransactionNumber	in	REQUIRED	Defines the transaction number (inclusive) regarding the end of the interval of relevant log messages.
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only returns the log messages if the number of relevant records is less or equal to the number of maximum records. If the value of the parameter is 0, the function returns all selected log messages.
exportedData	out	REQUIRED	Selected log messages and additional files needed to verify the signatures included in the log messages.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_PARAMETER_MISMATCH	Mismatch in parameters of the function.
ERROR_TRANSACTION_NUMBER_NOT_FOUND	No data has been found for the provided transaction numbers.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportDataFilteredByTransactionNumberInterval
"startTransactionNumber endTransactionNumber "maximumNumberRecords" "filename"
asigntseEXE.exe exportDataFilteredByTransactionNumberInterval 0 123 0
exportDataTransNrInterval.tar

asigntseEXE.exe exportDataFilteredByTransactionNumberIntervalWithTse
"startTransactionNumber" "endTransactionNumber" "maximumNumberRecords" "filename"
"tseId"
asigntseEXE.exe exportDataFilteredByTransactionNumberIntervalWithTse 0 123 0
exportDataTransNrInterval.tar TSE_1
```

exportDataFilteredByTransactionNumberIntervalAndClientId

Exports the transaction log messages, containing the process and protocol data, that are relevant for a certain interval of transactions. The transaction log messages in this interval corresponds to the passed clientId.

Additionally, the function SHALL export all system log messages and audit log messages whose signature counters are contained in the interval.

Furthermore, additional files are exported that are needed to verify the signatures in the log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
startTransactionNumber	in	REQUIRED	Defines the transaction number (inclusive) regarding the start of the interval of relevant log messages.
endTransactionNumber	in	REQUIRED	Defines the transaction number (inclusive) regarding the end of the interval of relevant log messages.
clientId	in	REQUIRED	ID of a client application that has used the API to log transactions. Only transaction log messages that corresponds to the clientId are relevant for the export.
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only returns the log messages if the number of relevant records is less or equal to the number of maximum records. If the value of the parameter is 0, the function SHALL return all selected log messages.
exportedData	out	REQUIRED	Selected log messages and additional files needed to verify the signatures included in the log messages.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_PARAMETER_MISMATCH	Mismatch in parameters of the function.
ERROR_TRANSACTION_NUMBER_NOT_FOUND	No data has been found for the provided transaction numbers.
ERROR_ID_NOT_FOUND	No data has been found for the provided clientId.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportDataFilteredByTransactionNumberIntervalAndClientId
"startTransactionNumber endTransactionNumber "clientId "maximumNumberRecords"
"filename"
asigntseEXE.exe exportDataFilteredByTransactionNumberIntervalAndClientId 0 123
Kasse_1 0 exportDataTransNrIntervalClientId.tar

asigntseEXE.exe exportDataFilteredByTransactionNumberIntervalAndClientIdWithTse
"startTransactionNumber" "endTransactionNumber" "clientId" "maximumNumberRecords"
"filename" "tseId"
asigntseEXE.exe exportDataFilteredByTransactionNumberIntervalAndClientIdWithTse 0
123 Kasse_1 0 exportDataTransNrIntervalClientId.tar TSE_1
```

exportDataFilteredByPeriodOfTime

Exports the transaction log messages, system log messages and audit log messages that have been created in a certain period of time.

Furthermore, additional files are exported that are needed to verify the signatures in the log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required	Description
startDate	in	CONDITIONAL	Defines the starting time (inclusive) for the period in that the relevant log messages have been created. See DateFormat. If a value for the input parameter endDate is passed, startDate is [OPTIONAL]. If no value for the input parameter endDate is passed, startDate is [REQUIRED].
endDate	in	CONDITIONAL	Defines the end time (inclusive) for the period in that relevant log messages have been created. See DateFormat. If a value for the input parameter startDate is passed, endDate is [OPTIONAL]. If no value for the input parameter startDate is passed, endDate is [REQUIRED].
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only returns the log messages if the number of relevant records is less or equal to the number of maximum records. If the value of the parameter is 0, the function returns all selected log messages.
filename	in	REQUIRED	Defines the name of the file in which the data will be written.
exportedData	out	-	Selected log messages and additional files needed to verify the signatures included in the log messages.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_PARAMETER_MISMATCH	Mismatch in parameters of the function.
ERROR_NO_DATA_AVAILABLE	No data has been found for the provided selection.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

Format for startDate and endDate: yyyy.mm.dd_hh:mm:ss or int64

```
asigntseEXE.exe exportDataFilteredByPeriodOfTime "startDate" "endDate"
"maximumNumberRecords" "filename"
asigntseEXE.exe exportDataFilteredByPeriodOfTime 2019.01.15_00:00:00
2019.21.01_00:00:00 0 exportDataPTime.tar

asigntseEXE.exe exportDataFilteredByPeriodOfTimeWithTse "startDate" "endDate"
"maximumNumberRecords" "filename" "tseId"
asigntseEXE.exe exportDataFilteredByPeriodOfTimeWithTse 2019.01.15_00:00:00
2019.21.01_00:00:00 0 exportDataPTime.tar 0 exportDataPTime.tar TSE_1
```

exportDataFilteredByPeriodOfTimeAndClientId

Exports the transaction log messages, system log messages and audit log messages that have been created in a certain period of time.

The transaction log messages in this period of time corresponds to the passed clientId.

Furthermore, additional files are exported that are needed to verify the signatures in the log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
startDate	in	CONDITIONAL	Defines the starting time (inclusive) for the period in that the relevant log messages have been created. See DateFormat. If a value for the input parameter endDate is passed, startDate is [OPTIONAL]. If no value for the input parameter endDate is passed, startDate is [REQUIRED].
endDate	in	CONDITIONAL	Defines the end time (inclusive) for the period in that relevant log messages have been created. See DateFormat. If a value for the input parameter startDate is passed, endDate is [OPTIONAL]. If no value for the input parameter startDate is passed, endDate is [REQUIRED].
clientId	in	REQUIRED	ID of a client application that has used the API to log transactions. Only transaction log messages that corresponds to the clientId are relevant for the export.
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only returns the log messages if the number of relevant records is less or equal to the number of maximum records. If the value of the parameter is 0, the function SHALL return all selected log messages.
filename	in	REQUIRED	Defines the name of the file in which the data will be written.
exportedData	out	-	Selected log messages and additional files needed to verify the signatures included in the log messages.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_PARAMETER_MISMATCH	Mismatch in parameters of the function.
ERROR_NO_DATA_AVAILABLE	No data has been found for the provided selection.
ERROR_ID_NOT_FOUND	No data has been found for the provided clientId.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe ExportDataFilteredByPeriodOfTimeAndClientId "startDate" "endDate"
"clientId" "maximumNumberRecords" "filename"
asigntseEXE.exe ExportDataFilteredByPeriodOfTimeAndClientId 2019.01.15_00:00:00
2019.21.01_00:00:00 Kasse_1 0 exportDataPTimeCId.tar

asigntseEXE.exe exportDataFilteredByPeriodOfTimeWithTse "startDate" "endDate"
"clientId" "maximumNumberRecords" "filename" "tseId"
asigntseEXE.exe exportDataFilteredByPeriodOfTimeWithTse 2019.01.15_00:00:00
2019.21.01_00:00:00 Kasse_1 0 exportDataPTime.tar 0 exportDataPTimeCId.tar TSE_1
```

exportData

Exports all stored transaction log messages, system log message and audit log messages. Furthermore, additional files are exported that are needed to verify the signatures in the log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only returns the log messages if the number of relevant records is less or equal to the number of maximum records. If the value of the parameter is 0, the function returns all stored log messages.
filename	in	REQUIRED	Defines the name of the file in which the data will be written.
exportedData	out	-	All stored log messages and additional files needed to verify the signatures included in the log messages.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportData "maximumNumberRecords" "filename"
asigntseEXE.exe exportData 0 exportData.tar

asigntseEXE.exe exportDataWithTse "maximumNumberRecords" "filename" "tseId"
asigntseEXE.exe exportDataWithTse 0 exportDataPTimeCId.tar TSE_1
```

exportCertificates

Exports the certificates of the certificate chains. These certificates belong to the public keys of the key pairs that are used for the creation of signature values in log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
filename	in	REQUIRED	Defines the name of the file in which the data will be written.
certificates	out	-	The TAR archive that contains all certificates that are necessary for the verification of log messages. The format of the TAR archive and the contained certificates SHALL conform to BSI TR-03151.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_EXPORT_CERT_FAILED	The collection of the certificates for the export failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportCertificates "filename"
asigntseEXE.exe exportCertificates exportCert.tar

asigntseEXE.exe exportCertificatesWithTse "filename" "tseId"
asigntseEXE.exe exportCertificatesWithTse exportCert.tar TSE_1
```

Convert Certificate Format

On Windows 10/Windows Server 2016 you can convert CER to the DER (PEM) certificate file format from the Windows build-in certificate export tool.

Windows:

1. Run the File Explorer, locate and double click your .cer file.
2. In the certificate properties window go to the **Details** tab and click on the “Copy to File” button.
3. Press **Next** on the first step of Certificate Export Wizard.
4. Now you need to select the certificate export format. Select the option “BASE-64 encoded X.509 (.CER)” and click Next.
5. Specify the file name
6. Press the **Finish** button

OpenSSL:

1. Download and install openssl.
2. Open a new cmd.exe and execute openssl with the following commands:
    - path/to/openssl/bin/Openssl.exe x509 -inform DER -outform PEM -in my_certificate.crt -out my_certificate.crt.crt

or

- path/to/openssl/bin/Openssl.exe x509 -inform DER -outform PEM -in my_certificate.crt -out my_certificate.crt.pem

NOTE: You can add OpenSSL to your PATH to execute OpenSSL.exe without the Path

- Hit the Windows button on your keyboard or click it in the task bar, then search for "Environment Variables".
- In the following screen, click "Environment Variables".
- A screen will pop up showing User variables and System variables. In the User variables section, select Path and click Edit.
- Click on Browse.
- Go to where the openssl.exe is, which should be at "This PC > Windows (C:) > Program Files > OpenSSL - Win64 > bin" and select that folder. Click OK.

readLogMessage

Reads a log message that bases on the last log message parts that have been produced and processed by the Secure Element.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
filename	in	REQUIRED	Defines the name of the file in which the data will be written.
logMessage	out	-	Contains the last log message that the Secure Element has produced.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_NO_LOG_MESSAGE	No log message parts are found.
ERROR_READING_LOG_MESSAGE	Error while retrieving log message parts.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

Example

```
asigntseEXE.exe readLogMessage "filename"
asigntseEXE.exe readLogMessage LogMessage.asn1

asigntseEXE.exe readLogMessageWithTse "filename" "tseId"
asigntseEXE.exe readLogMessageWithTse LogMessage.asn1 TSE_1
```

exportSerialNumbers

Exports the serial number(s) of the SE API. A serial number is a hash value of a public key that belongs to a key pair whose private key is used to create signature values of log messages.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
filename	in	REQUIRED	Defines the name of the file in which the data will be written.
serialNumbers	out	-	The serial number(s) of the SE API. The serial number(s) SHALL be encoded in the TLV structure defined in BSI TR-03151.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_EXPORT_SERIAL_NUMBERS_FAILED	The collection of the serial number(s) failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportSerialNumbers "filename"
asigntseEXE.exe exportSerialNumbers exportSerialNumber.tar

asigntseEXE.exe exportSerialNumbersWithTse "filename" "tseId"
asigntseEXE.exe exportSerialNumbersWithTse exportSerialNumber.tar TSE_1
```

exportAllAndDeleteStoredData

This function replaces deletestoreddata.

NOTE: An admin login is required for this function

Parameters

Name	In/Out	Required?	Description
filename	in	REQUIRED	Defines the name of the file in which the data will be written.
authenticateuser	in	-	--authenticate Admin "pw"

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.

Example

```
asigntseEXE.exe exportallanddeletestoreddata data.tar --authenticateuser Admin
12345678
```

User Management

disableSecureElement

The function disableSecureElement disables the Secure Element in a way that none of its functionality can be used anymore.

This function also exists with `withTse`.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_DISABLE_SECURE_ELEMENT_FAILED	The deactivation of the Secure Element failed.
ERROR_TIME_NOT_SET	The managed data/time in the Secure Element has not been updated after the initialization of the SE API or a period of absence of current for the Secure Element.
ERROR_RETRIEVE_LOG_MESSAGE_FAILED	Execution of the Secure Element functionality to retrieve log message parts has failed.
ERROR_STORAGE_FAILURE	Storing of the data of the log message has failed.
ERROR_CERTIFICATE_EXPIRED	The certificate with the public key for the verification of the appropriate type of log messages is expired. Even if a certificate expired, the log message parts are created by the Secure Element and stored by the SE API.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.
ERROR_USER_NOT_AUTHORIZED	The user who has invoked the function disableSecureElement is not authorized to execute this function.
ERROR_USER_NOT_AUTHENTICATED	The user who has invoked the function disableSecureElement has not the status authenticated.

Example

```
asigntseEXE.exe disableSecureElement
```

getMaxNumberOfClients

Supplies the maximal number of clients that can use the functionality to log transactions of the SE API simultaneously.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
maxNumberClients	out	-	Maximum number of clients that can use the functionality to log transactions of the SE API simultaneously.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_GET_MAX_NUMBER_OF_CLIENTS_FAILED	The determination of the maximum number of clients that could use the SE API simultaneously failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

Example

```
asigntseEXE.exe getMaxNumberOfClients
```

getCurrentNumberOfClients

Supplies the number of clients that are currently using the functionality to log transactions of the SE API.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
currentNumberClients	out	-	The number of clients that are currently using the functionality of the SE API.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_GET_CURRENT_NUMBER_OF_CLIENTS_FAILED	The determination of the current number of clients using the SE API failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

Example

```
asigntseEXE.exe getCurrentNumberOfClients
```

getMaxNumberOfTransactions

Supplies the maximal number of simultaneously opened transactions that can be managed by the SE API.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
maxNumberTransactions	out	REQUIRED	Maximum number of simultaneously opened transactions that can be managed by the SE API.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_GET_MAX_NUMBER_TRANSACTIONS_FAILED	The determination of the maximum number of transactions that can be managed simultaneously failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

Example

```
asigntseEXE.exe getMaxNumberOfTransactions
```

getCurrentNumberOfTransactions

Supplies the number of open transactions that are currently managed by the SE API.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
currentNumberTransactions	out	REQUIRED	The number of open transactions that are currently managed by the SE API.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_GET_CURRENT_NUMBER_OF_TRANSACTIONS_FAILED	The determination of the number of open transactions that are currently managed by the SE API failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

Example

```
asigntseEXE.exe getCurrentNumberOfTransactions
```

getSupportedTransactionUpdateVariants

Supplies the supported variants to update transactions.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
supportedUpdateVariants	out	-	The supported variant(s) to update a transaction.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_GET_SUPPORTED_UPDATE_VARIANTS_FAILED	The identification of the supported variant(s) to update transactions failed.
ERROR_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

Example

```
asigntseEXE.exe getSupportedTransactionUpdateVariants
```

authenticateUser

Enables an authorized user or application to authenticate to the SE API for the usage of restricted SE API functions. This function is used as Parameter extension.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
userId	in	REQUIRED	The ID of the user who or application that wants to be authenticated.
pin	in	REQUIRED	The PIN for the authentication.
authenticationResult	out	-	The result of the authentication.
remainingRetries	out	-	The number of remaining retries to enter a PIN.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_SIGNING_SYSTEM_OPERATION_DATA_FAILED	The determination of the log message parts for the system operation data by the Secure Element failed.
ERROR_RETRIEVE_LOG_MESSAGE_FAILED	The execution of the Secure Element functionality to retrieve log message parts has failed.
ERROR_STORAGE_FAILURE	Storing of the data of the log message failed.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled.

unblockUser

Enables the unblocking for the entry of a PIN and the definition of a new PIN for the authentication of authorized users or applications.

This function also exists with `withTse`.

Parameters

Name	In/Out	Required?	Description
userId	in	REQUIRED	The ID of the user who or application that wants to be authenticated.
puk	in	REQUIRED	The PUK of the user/application.
newPin	in	REQURIED	The new PIN for the user/application.
unblockResult	out	-	The result of the unblock procedure.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
UNBLOCK_FAILED	If the execution of attempt to unblock a PIN entry has failed, the return value UNBLOCK_FAILED SHALL be returned.
ERROR_SIGNING_SYSTEM_OPERATION_DATA_FAILED	The determination of the log message parts for the system operation data by the Secure Element failed.
ERROR_RETRIEVE_LOG_MESSAGE_FAILED	The execution of the Secure Element functionality to retrieve log message parts has failed.
ERROR_STORAGE_FAILURE	Storing of the data of the log message failed.
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has already been disabled.

Example

```
asigntseEXE.exe unblockUser user123 1234 1234
```

Additional Functions

genqrcode

Generate and output qrcode to the provided path.

Structure

;;;; ;; ;;;;;

Parameters

Name	In/Out	Required?	Description
ClientId	in	REQUIRED	Represents the ID of the application that has invoked the function.
processType	in	REQUIRED	Identifies the format of the content in processData
processData	in	REQUIRED	Contains the data. (use ^^ instead of ^)
TransactionNumber	in	REQUIRED	Transaction number FinishTransaction.
FinishTransactionSigCounter	in	REQUIRED	Signature counter of the FinishTransaction.
StartTransactionTime	in	REQUIRED	LogTime from the StartTransaction (UnixTime)
FinishTransactionTime	in	REQUIRED	LogTime from the FinishTransaction (UnixTime)
signatureValue	in	REQUIRED	Signature value from the FinishTransaction (b64 encoded)
path	in	REQUIRED	Output path for qrcode. The parameter . indicates the asigntseEXE path.
filename	in	OPTIONAL	no specific filename is provided the default name (qrcode.bmp) is used.

Example

```
asigntseEXE.exe genqrcode 955002-00 Kassenbeleg-V1
Beleg^^0.00_2.55_0.00_0.00_0.00_^^2.55Bar 12 16 1611311064 1611311080
signatureb64 . qrcode.bmp

> The ^ symbol (also called circumflex) is an escape character, therefore 2 must
be used.
```

genqrcodecustomsize

Generate and output qrcode to the provided path.

Parameters

Name	In/Out	Required?	Description
ClientId	in	REQUIRED	Represents the ID of the application that has invoked the function.
processType	in	REQUIRED	Identifies the format of the content in processData
processData	in	REQUIRED	Contains the data. (use ^^ instead of ^)
TransactionNumber	in	REQUIRED	Transaction number FinishTransaction.
FinishTransactionSigCounter	in	REQUIRED	Signature counter of the FinishTransaction.
StartTransactionTime	in	REQUIRED	LogTime from the StartTransaction (UnixTime)
FinishTransactionTime	in	REQUIRED	LogTime from the FinishTransaction (UnixTime)
signatureValue	in	REQUIRED	Signature value from the FinishTransaction (b64 encoded)
path	in	REQUIRED	Output path for qrcode. The parameter . indicates the asigntseEXE path.
filename	in	**REQUIRED**	Defines the qrcode filename.
Scale	in	OPTIONAL	Defines the pixels per module. (Default size is 2)

Example

```
asigntseEXE.exe genqrcodecustomsize 955002-00 Kassenbeleg-V1
Beleg^^0.00_2.55_0.00_0.00_0.00_^^2.55Bar 12 16 1611311064 1611311080
signatureb64 . qrcode.jpg 1

> The ^ symbol (also called circumflex) is an escape character, therefore 2 must
be used.
```

genqrcodefromstring

Generate a qrcode from string

Parameters

Name	In/Out	Required?	Description
InputString	in	REQUIRED	The string to encode.
Path	in	REQUIRED	Defines the path in which the qrcode is saved. (Use . for the working directory)
Filename	in	OPTIONAL	Defines the qrcode filename.

Example

```
asigntseEXE.exe genqrcodefromstring <qr-code-version>;<kassen-seriennummer>;
<processType>;<processData>;<transaktions-nummer>;<signatur-zaehler>; <start-
zeit>;<log-time>;<sig-alg>;<log-time-format>;<signatur>;<public-key> .
qrcode.bmp
```

genqrcodefromstringcustomsize

Generate a qrcode from string

Parameters

Name	In/Out	Required?	Description
InputString	in	REQUIRED	The string to encode.
Path	in	REQUIRED	Defines the path in which the qrcode is saved. (Use . for the working directory)
Filename	in	**REQUIRED**	Defines the qrcode filename.
Scale	in	OPTIONAL	Defines the pixels per module. (Default size is 2)

Example

```
asigntseEXE.exe genqrcodefromstring <qr-code-version>;<kassen-seriennummer>;
<processType>;<processData>;<transaktions-nummer>;<signatur-zaehler>; <start-
zeit>;<log-time>;<sig-alg>;<log-time-format>;<signatur>;<public-key> . qrcode.jpg
1
```

convertStringToBase64

Encode specified data to Base64.

Parameters

Name	In/Out	Required?	Description
input_string	in	REQUIRED	The string to translate to base64.
output_string	out	-	Base64 encoded string.

Example

```
asigntseEXE.exe convertstringtobase64 abcdefghij!"§$%&/(0123456
```

convertUnixTimeToUTC

Converts specified UnixTime to UTC (Universal Time Coordinated).

Parameters

Name	In/Out	Required?	Description
UnixTime	in	REQUIRED	Datetime in UnixTime format.
output_string	out	-	UTC Datetime.

Example

```
asigntseEXE.exe convertUnixTimeToUTC 1617092406

Output: 2021-03-30T08:20:06.000Z
```

A-Trust API Functions

The following functions are not defined in the SE-API specification. They are additional functions specific to our implementation.

at_getVersion

Shows version information of the underlying TSE library.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getVersion
```

at_getSignatureAlgorithm

Supplies the friendly name of the ASN.1 encoded signature algorithm OID encoded into signed data.

Parameters

Name	In/Out	Required?	Description
signatureAlgorithm	out	REQUIRED	Represents the signatureAlgorithm.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getSignatureAlgorithm
```

at_getPublicKey

Export public key of the certificate signing transaction logs.

Parameters

Name	In/Out	Required?	Description
at_getPublicKey hex	out	REQUIRED	Represents the publicKey. (hex)
at_getPublicKey base64	out	REQUIRED	Represents the publicKey. (base64)

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getPublicKey
```

at_getOpenTransactions

Returns a list of transactions which have been started with `startTransaction` but not yet been completed with `finishTransaction`.

Parameters

Name	In/Out	Required?	Description
transactionNumbers	out	REQUIRED	Array that represents the list of open transactions.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getOpenTransactions
```

at_getSignatureCounter

Supplies the current signature counter (last used value) for the certificate signing transaction logs.

Parameters

Name	In/Out	Required?	Description
signatureCounter	out	-	The current signature counter.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getOpenTransactions
```

at_getTransactionCounter

Supplies the current transaction counter (last used value).

Parameter

Name	In/Out	Required?	Description
transactionCounter	out	-	The current transaction counter.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getTransactionCounter
```

at_getLifecycleState

Gets the lifecycle state of the TSE.

Parameters

Name	In/Out	Required?	Description
state	out	-	The lifecycle state: 0=unknown, 1=not initialized, 2=active, 3=deactivated, 4=disabled

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getLifecycleState
```

at_getSerialNumber

Gets the serial number (SHA256 hash of public key) of the transaction log signing key.

Parameters

Name	In/Out	Required?	Description
serial	out	REQUIRED	Buffer for serial number bytes.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_getSerialNumber
```

at_suspendSecureElement

Temporarily suspend the secure element.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_suspendSecureElement
```

at_unsuspendSecureElement

Unsuspend the secure element.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
asigntseEXE.exe at_unsuspendSecureElement
```

Configuration Functions

The configuration functions are used as parameter extenensions.

Example

```
asigntseEXE.exe funnctionname functionparameter --cfgxxxxxx cfgparameter --cfg...

asigntseEXE.exe StartTransaction clientID --cfgsetloggingenabled true --
cfgsetloggingstderr true
```

cfgSetConfigFile

Set path to the config file.

Parameters

Name	In/Out	Required?	Description
path	in	REQUIRED	Path to a asigntseonline.conf

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Usage:

```
--cfgsetconfigfile "PATH/TO/FILE"

--cfgsetconfigfile C:\temp
```

cfgTseAdd

Add Tse config section. The Section will not be written to the config file.

Parameters

Name	In/Out	Required?	Description
tseId	in	REQUIRED	Name of the tse config section.
tseType	in	REQUIRED	Has to be 1 for a.sign TSE or 2 for Cryptovision.
connParam	in	REQUIRED	URL to the asignTSE webserver.
atrustTseID	in	REQUIRED	TSE Identification Number.
timeAdminID	in	(OPTIONAL)	In case of asignTSE do not use timeAdminId
timeAdminPwd	in	(OPTIONAL)	

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgTseAdd tseId tseType connParam (atrustTseID timeAdminID)

--cfgTseAdd TSE_1 1 https://hs-abnahme.a-trust.at/asigntseonline/v1
u0000000000212xx
--cfgTseAdd TSE_1 1 https://hs-abnahme.a-trust.at/asigntseonline/v1
u0000000000212xx abc abc

const char* connParam = "https://hs-abnahme.a-trust.at/asigntseonline/v1";
    const char* username = "u0000000000212xx";
    const char* api_key =
"testapikey45345123foe58392abb2cf3267ebc9cf2abf700a50e1a8bb623e4013844d0";
    cfgTseAdd(
        "default", strlen("default"),
        1,
        connParam, strlen(connParam),
        username, strlen(username),
        api_key, strlen(api_key),
        NULL, 0, NULL, 0
    );
```

cfgTseRemove

Remove Tse config section.

Parameter

Name	In/Out	Required?	Description
tseId	in	REQUIRED	Name of the tse config section.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgTseRemove tseId

--cfgTseRemove TSE_1
```

cfgSetLoggingEnabled

Enable logging.

Parameters

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLoggingEnabled true
```

cfgSetLoggingStderr

Enable logging to stderr.

Parameters

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLoggingStderr true
```

cfgSetLoggingFile

Enable logging to a logfile.

Parameters

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLoggingFile true
```

cfgSetLogDir

Set logfile target directory.

Parameters

Name	In/Out	Required?	Description
path	in	REQUIRED	Target Directory for logfiles.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLogDir C:\PATH\TO\DIR

--cfgSetLogDir C:\temp
```

cfgSetLogLevel

Set verbosity level of the logger.

Parameters

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLogLevel true
```

cfgSetLogAppend

The output file will be created if it does not exist.

If the log append flag is set to true, the logger will append to the output file.
If not set (default), the output file will be truncated to zero length before the logger starts writing to it.

Parameters

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLogAppend true
```

cfgSetLogColors

Enable a colored version of the logline-formatter.

Parameter

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLogColors true
```

cfgSetLogDetails

Enable more detailed log lines.

Parameters

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLogDetails true
```

cfgSetLogStderrColors

Enable a colored version of the logline-formatter.

Parameters

Name	In/Out	Required?	Description
enabled	in	REQUIRED	Set the value of this property to `true` to enable the option or `false` to disable it.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetLogStderrColors true
```

cfgSetHttpProxy

Set the http proxy.

Parameters

Name	In/Out	Required?	Description
proxyUrl	in	REQUIRED	Proxy Url and Port. This overwrites the HTTP_PROXY environment variable.

Return Codes

Code	Description
EXECUTION_OK	If the execution of the function has been successful, the return value EXECUTION_OK will be returned.
ERROR_UNKNOWN	If the execution fails, the return value ERROR_UNKNOWN will be returned.

Example

```
--cfgSetHttpProxy proxyurl
```

Numeric Error values

Constant	value
ERROR_MISSING_PARAMETER	-3000;
ERROR_FUNCTION_NOT_SUPPORTED	-3001
ERROR_IO	-3002
ERROR_TSE_TIMEOUT	-3003
ERROR_ALLOCATION_FAILED	-3004;
ERROR_CONFIG_FILE_NOT_FOUND	-3005;
ERROR_SE_COMMUNICATION_FAILED	-3006;
ERROR_TSE_COMMAND_DATA_INVALID	-3007
ERROR_TSE_RESPONSE_DATA_INVALID	-3008
ERROR_ERS_ALREADY_MAPPED	-3009
ERROR_NO_ERS	-3010
ERROR_TSE_UNKNOWN_ERROR	-3011
ERROR_STREAM_WRITE	-3012
ERROR_BUFFER_TOO_SMALL	-3013
ERROR_NO_SUCH_KEY	-3014
ERROR_NO_KEY	-3015
ERROR_SE_API_DEACTIVATED	-3016
ERROR_SE_API_NOT_DEACTIVATED	-3017
ERROR_UNKNOWN	-3100
ERROR_AUTHENTICATION_FAILED	-4000
ERROR_UNBLOCK_FAILED	-4001
ERROR_RETRIEVE_LOG_MESSAGE_FAILED	-5001
ERROR_STORAGE_FAILURE	-5002
ERROR_UPDATE_TIME_FAILED	-5003
ERROR_PARAMETER_MISMATCH	-5004
ERROR_ID_NOT_FOUND	-5005
ERROR_TRANSACTION_NUMBER_NOT_FOUND	-5006
ERROR_NO_DATA_AVAILABLE	-5007
ERROR_TOO_MANY_RECORDS	-5008
ERROR_START_TRANSACTION_FAILED	-5009

Constant	value
ERROR_UPDATE_TRANSACTION_FAILED	-5010
ERROR_FINISH_TRANSACTION_FAILED	-5011
ERROR_RESTORE_FAILED	-5012
ERROR_STORING_INIT_DATA_FAILED	-5013
ERROR_EXPORT_CERT_FAILED	-5014
ERROR_NO_LOG_MESSAGE	-5015
ERROR_READING_LOG_MESSAGE	-5016
ERROR_NO_TRANSACTION	-5017
ERROR_SE_API_NOT_INITIALIZED	-5018
ERROR_TIME_NOT_SET	-5019
ERROR_CERTIFICATE_EXPIRED	-5020
ERROR_SECURE_ELEMENT_DISABLED	-5021
ERROR_USER_NOT_AUTHORIZED	-5022
ERROR_USER_NOT_AUTHENTICATED	-5023
ERROR_DESCRIPTION_NOT_SET_BY_MANUFACTURER	-5024
ERROR_DESCRIPTION_SET_BY_MANUFACTURER	-5025
ERROR_EXPORT_SERIAL_NUMBERS_FAILED	-5026
ERROR_GET_MAX_NUMBER_OF_CLIENTS_FAILED	-5027
ERROR_GET_CURRENT_NUMBER_OF_CLIENTS_FAILED	-5028
ERROR_GET_MAX_NUMBER_TRANSACTIONS_FAILED	-5029
ERROR_GET_CURRENT_NUMBER_OF_TRANSACTIONS_FAILED	-5030
ERROR_GET_SUPPORTED_UPDATE_VARIANTS_FAILED	-5031
ERROR_DELETE_STORED_DATA_FAILED	-5032
ERROR_UNEXPORTED_STORED_DATA	-5033
ERROR_SIGNING_SYSTEM_OPERATION_DATA_FAILED	-5034
ERROR_USER_ID_NOT_MANAGED	-5035
ERROR_USER_ID_NOT_AUTHENTICATED	-5036
ERROR_DISABLE_SECURE_ELEMENT_FAILED	-5037
ERROR_CONFIG_VALUE_NOT_FOUND	-5038
ERROR_INVALID_CONFIG	-5039

Constant	value
ERROR_SUSPEND_SECURE_ELEMENT_FAILED	-5040
ERROR_UNSUSPEND_SECURE_ELEMENT_FAILED	-5041
ERROR_GET_OPEN_TRANSACTIONS_FAILED	-5042
ERROR_GET_LIFECYCLE_STATE_FAILED	-5043
ERROR_GET_TRANSACTION_COUNTER_FAILED	-5044
ERROR_GET_SIGNATURE_ALGORITHM_FAILED	-5045
ERROR_GET_SIGNATURE_COUNTER_FAILED	-5045
ERROR_GET_TOTAL_LOG_MEMORY	-5046
ERROR_GET_LOG_TIME_FORMAT	-5047
ERROR_EXPORT_PUBLIC_KEY_FAILED	-5048
ERROR_EXPORT_CERTIFICATE_FAILED	-5049
ERROR_UNSUPPORTED_PREMIUM_FEATURE	-6000
EXECUTION_OK	0

Configuration File

Configuring Technical Secure Systems

Every config file must have a `[default]` section. If a SEAPI function without the `WithTse` postfix gets called, the `[default]` tse is used. Here tss and tse are used synonymously.

Constant	value
tss_type	Reserved for future use.
conn_param	SMAERS Server Url.
atrust_api_key	TSS api key.
atrust_vtss_id	TSS id.

Example

```
[default]
tss_type=1
conn_param=https://hs-abnahme.a-trust.at/asigntseonline/v1
atrust_api_key=880f02caddcd6bad2102aa9962d0fa7f1c42c6618635f8fd07cbfabb131f0b14
atrust_vtss_id=u000000000010123

[tse_1]
tss_type=1
conn_param=https://hs-abnahme.a-trust.at/asigntseonline/v1
```

```
atrust_api_key=880f02caddcd6bad2102aa9962d0fa7f1c42c6618635f8fd07cbfabb131f0b14
atrust_vtss_id=u000000000010123

[tse_2]
tss_type=1
conn_param=https://hs-abnahme.a-trust.at/asigntseonline/v1
atrust_api_key=5c0e7104194bea87f203ecd39f6f7c87a49403384d65f82612c95cf263720345
atrust_vtss_id=u000000000010124
```

General Configuration

In order to set general configuration parameters, there is an optional `[config]` section. The following table shows configuration options and switches.

Option	Type	Description
logging_enabled	Boolean	Enable logging.
logging_stderr	Boolean	Enable logging to stderr.
logging_file	Boolean	Enable logging to a logfile.
log_dir	Path	Set logfile directory.
log_details	Boolean	Enable more detailed log lines.
log_level	error | warn | info | debug | trace	Set verbosity level of the logger.
log_append	Boolean	true: append to the output file, if it exists; false: the output file will be truncated to zero length before the logger starts writing to it.
log_colors	Boolean	Enable a colored version of the logline-formatter.
log_stderr_colors	Boolean	Enable a colored version of the logline-formatter.
http_proxy	URL	Set the http proxy.