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

Date	Revision	Author	Changes
20/07/2020	1.0.0.0	Thomas Schützenhöfer	asigntse.exe premium description
25/03/2021	1.1.1.0	Thomas Schützenhöfer	asigntse.exe logging, exportdata changes
30/03/2021	1.1.1.2	Thomas Schützenhöfer	asigntse.exe rename premium to einheit

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.

This document is a supplement to the Developer manual and describes the additional features available in the ASign Einheit solution. This solution provides a wrapper around the SE-API implementations of other manufacturers to the ASignOnline API. This allows developers who need to use both online and chip-based solutions to develop using a common interface.

Note that there are some features relevant to a chip-based solution that are not relevant to the online solution and vice versa. We will endeavour to provide the most important functions from the chip implemenations, though some non-standard functions may only be accessible through the provider's library.

Installation

The A-Trust Einheit library file and the Crytovision library file must be placed in the same directory as the executeable **(asignTSE.EXE)**. The provided configuration file must be placed in the working directory of the executable.

The configuration file `asigntseonline.conf` will need to contain entries for `tss_type`, `conn_param` and optionally for `time_admin_id` and `time_admin_pwd` (see Automatic Time Management). For Cryptovision, `tss_type` must have the value 2. The entry `conn_param` specifies the device path where the Cryptovision chip is mounted.

Example configuration for a Cryptovision card

```
[default]
tss_type=2
conn_param=H:
time_admin_id=TimeAdmin
time_admin_pwd=22222222
```

Automatic Connection Management

The A-Sign TSE API manages the connection to the TSE chip automatically using the connection parameter value provided in the configuration file.

Automatic Time Management

If your configuration file contains the entries `time_admin_id` and `time_admin_pwd`, then the A-Trust Wrapper will use these credentials to log in as the Time Administrator automatically and update the time from the local time source whenever it is required.

Note that If you do not supply these details with your configuration then you can still perform this process manually. In this case you would need to explicitly call `authenticateUser` and authenticate in the time administrator role, set the time using the function `updateTime` and then call `logout` to logout of the time administrator role. This sequence of commands would need to be performed on each startup, at regular intervals as recommended by the chip manufacturer (for Cryptovision see `cv_getTimeSyncInterval`), and whenever time errors occur (see updateTime).

Preparation

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

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

      NOTE: The EXE runs on both 32 and 64 bit. Always use the 32bit Version of the asigntse.dll and se-api-ex-c.dll.

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

The following example shows how to provision a Cryptovision token that is in the factory state.

To do this we first call the function `cv_initializePinValues` (which is a wraps the Cryptovision function `se_initializePinValues`) to create the initial PIN and PUK values. The function `cv_getPinStatus` can be used to check the TSE Pin status. (0 = already initialized).

The function `cv_initializePinValues` has 4 Parameter.

- AdminPin - Requires 8 digits
- AdminPuk - Requires 10 digits
- TimeAdminPin - Requires 8 digits - must match the entry in the config file

- TimeAdminPuk - Requires 10 digits

Next we need to execute the SE API function `initializeDescriptionSet` to put the TSE into the initialized state. The function `initializeDescriptionSet` requires a login. `InitializeDescriptionSet` will execute an authenticate with `--authenticateuser username password` parameter.

The function `initializeDescriptionSet` has 3 Parameter.

- --authenticateuser - indicate an authentication
- Username - indicate the user used for login - Use : Admin
- Password - indicate the password set for the Username during `cv_initializePinValues`

We can call the function `at_getLifecycleState` to check if the TSE Status is initialized. The lifecycle state: 0=unknown, 1=not initialized, 2=active, 3=deactivated, 4=disabled

We then call the function `cv_registerClientId --authenticateuser username password` . The asigntseEXE.exe, will log out of the administrator role using the SE API function `logOut` automatically.

Example: Initialization

```
asigntseEXE.exe cv_initializepinvalues 12345678 1111111111 22222222 something!

asigntseEXE.exe initializedescriptionset --authenticateuser Admin 12345678

asigntseExe.exe cv_registerClientId clientId_1 --authenticateuser Admin 12345678
asigntseExe.exe cv_registerClientId clientId_2 --authenticateuser Admin 12345678
asigntseExe.exe cv_registerClientId clientId_3 --authenticateuser Admin 12345678
...
```

Troubleshooting:

To display the log messages from the asigntse.dll use the additional parameter `--cfgsetloggingenabled true`. This will output the logging to the console. In addition more `--cfg...` functions can be specified to configure the logging.

Additional Functions

The following additional functions are provided.

- Cryptovision functions
  - cv_exportData
  - cv_exportMoreData
  - cv_getERSMappings
  - cv_getWearIndicator
  - cv_getApiVersion
  - cv_getApiVersionString

cv_exportData

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 SHALL correspond 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 determined either by transaction number or by date. A combination is not supported.

Note

1. If transactionNumber has been provided, neither startDate nor endDate SHALL be provided.
2. If startTransactionNumber and endTransactionNumber have been provided, neither startDate nor endDate SHALL be provided.
3. If provided, startDate and/or endDate MUST be valid date/time values.
4. If startDate and endDate have been provided, endDate MUST lay after startDate.
5. If a startDate and/or endDate have been provided, transactionNumber MUST NOT be provided.

Parameters

Name	In/Out	Required?	Description
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
transactionNumber	in	REQUIRED	Defines the transaction number of relevant log messages. (null for undefined)
startTransactionNumber	in	REQUIRED	Defines the transaction number (inclusive) regarding the start of the interval of relevant log messages. (0 or null for undefined)
endTransactionNumber	in	REQUIRED	Defines the transaction number (inclusive) regarding the end of the interval of relevant log messages. (0 or null for undefined)
startDate	in	REQUIRED	Defines the starting time (Format: yyyy.mm.dd_hh:mm:ss or int64) for the period in that the relevant log messages have been created. (0 or null for undefined)
endDate	in	REQUIRED	Defines the end time (Format: yyyy.mm.dd_hh:mm:ss or int64) for the period in that relevant log messages have been created. (null for undefined)
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only return 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 return all selected log messages.
filename	in	REQUIRED	for the exported data

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_PARAMETER_MISMATCH	Mismatch in parameters of the function.
ERROR_NO_DATA_AVAILABLE	No data has been found for the provided selection.
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_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.

Example

```
asigntseEXE.exe cv_exportData clientid transactionNumber startTransactionNumber
endTransactionNumber startDate endDate maximumNumberRecords file

asigntseEXE.exe client_1 10 null null null null 0 export_data.tar
```

cv_exportMoreData

Parameters

Name	In/Out	Required?	Description
serialNumberKey	in	REQUIRED	ID of the key used in previous log entry (SHA256 hash value of the public key)
previousSignatureCounter	in	REQUIRED	last seen signature counter
maximumNumberRecords	in	REQUIRED	If the value of this parameter is not 0, the function only return 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 return all selected log messages.
filename	in	REQUIRED	for the exported data

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
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_SE_API_NOT_INITIALIZED	The SE API has not been initialized.
ERROR_TOO_MANY_RECORDS	The amount of requested records exceeds the parameter maximumNumberRecords.

Example

```
asigntseEXE.exe cv_exportMoreData serial_number_key(hex) prev_sig_counter
max_records filename
```

cv_getERSMappings

Read all mappings of ERS and belonging public key serial number.

Parameters

In/Out	Required?	Name	Description
out	-	**mappingData**	Buffer for the returned DER encoded mapping data (sequence of ERS mappings)

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.

Example

```
asigntseEXE.exe cv_getERSMappings
```

cv_getWearIndicator

Get Wear Indicator.

Parameters

In/Out	Required?	Name	Description
out	-	**wearIndicator**	For values below 100, typical data retention is more than 10 years. Bigger values indicate shorter data retention, but at least 1 year.

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.

Example

```
asigntseEXE.exe cv_getWearIndicator
```

cv_getApiVersionString

Get version strings of SE-API-C.

Parameters

In/Out	Required?	Name	Description
out	-	**apiversionstring**	SE-API-C version string (null terminated)

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.

Example

```
asigntseEXE.exe cv_getApiVersionString
```

cv_getAvailableLogMemory

Supplies size of available log memory in bytes.

Parameters

In/Out	Required?	Name	Description
out	-	**sizeOfMemory**	size of available log memory in bytes [REQUIRED]

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_getAvailableLogMemory
```

cv_getCertificateExpirationDate

Supplies the expiration date of a certificate.

Parameters

In/Out	Required?	Name	Description
in	REQUIRED	**serialNumberKey**	ID of the belonging key (SHA256 hash value of the public key).
out	-	**sizeOfMemory**	size of available log memory in bytes [REQUIRED]

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_getCertificateExpirationDate serialNumberKey
```

cv_getCertificationId

Get certification ID.

Parameters

In/Out	Required?	Name	Description
out	-	**certificationId**	Returned identifier string (not null terminated)

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_SE_COMMUNCATION_FAILED	Secure Element communication failed

Example

```
asigntseEXE.exe cv_getCertificationId
```

cv_getFirmwareId

Get firmware identifier of a TSE.

Parameters

In/Out	Required?	Name	Description
out	-	**certificationId**	TSE firmware version string (null terminated) or NULL if TSE connection not open.

Example

```
asigntseEXE.exe cv_getFirmwareId
```

cv_getImplementationVersionString

Get version strings of SE-API-C implementation.

Parameters

In/Out	Required?	Name	Description
out	-	**ImplementationVersion**	SE-API-C implementation version string (null terminated)

Example

```
asigntseEXE.exe cv_getImplementationVersionString
```

cv_getPinStatus

Get TSE PIN object status.

Parameters

In/Out	Required?	Name	Description
out	-	**pinState**	"PIN in transport state" per Admin PIN, Admin PUK, TimeAdmin PIN, TimeAdmin PUK.

Output

Represents the PIN states of the TSE.

The values defined for each PIN are bit-masks.
They may be combined by the "bit-wise or" operator (|).

Enumerator	
stateInitialized	ALL PINs have been initialized
adminPinTransportState	the Admin PIN is in transport state
adminPukTransportState	the Admin PUK is in transport state
timeAdminPinTransportState	the TimeAdmin PIN is in transport state
timeAdminPukTransportState	the TimeAdmin PUK is in transport state

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_SE_COMMUNCATION_FAILED	Secure Element communication failed

Example

```
asigntseEXE.exe cv_getPinStatus
```

cv_getTimeSyncInterval

Supplies the proposed update interval for the CSP time base (number of seconds)

Parameters

In/Out	Required?	Name	Description
out	REQUIRED	**timeSyncInterval**	Sync interval in seconds

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_getTimeSyncInterval
```

cv_getTimeSyncVariant

Supplies flags for the supported time formats of the TSE.

Parameters

In/Out	Required?	Name	Description
out	REQUIRED	**supportedSyncVariant**	Buffer for the supported time sync format

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_getTimeSyncInterval
```

cv_getTotalLogMemory

Supplies size of log memory in bytes.

Parameters

In/Out	Required?	Name	Description
out	REQUIRED	**sizeOfMemory**	size of log memory in bytes

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_getTotalLogMemory
```

cv_initializePinValues

Initialized TSE pin objects.

Note

The length of either PIN must be exactly 8 bytes!

The length of either PUK must be exactly 10 bytes!

Parameters

In/Out	Required?	Name	Description
in	REQUIRED	**adminPin**	Admin PIN value or NULL (PIN not touched). Length = 8
in	REQUIRED	**adminPuk**	Admin PUK value or NULL (PUK not touched). Length = 10
in	REQUIRED	**timeAdminPin**	TimeAdmin PIN value or NULL (PIN not touched). Length = 8
in	REQUIRED	**timeAdminPuk**	TimeAdmin PUK value or NULL (PUK not touched). Length = 10

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_initializepinvalues 12345678 1111111111 22222222 something!
```

cv_mapERStoKey

Map the serial number of an ERS to a signature key.

This assigns an existing private key to a client (cash register).
Function is also used to delete such a mapping.

Parameters

In/Out	Required?	Name	Description
in	REQUIRED	**clientId**	serialnumber of the ERS
in	REQUIRED	**serialNumberKey**	ID of the key to be mapped (SHA256 hash value of the public key); when `NULL` the client-mapping will be removed

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled
ERROR_SIGNING_SYSTEM_OPERATION_DATA_FAILED	Determination of the log message parts for the system operation data by the Secure Element failed
ERROR_RETRIEVE_LOG_MESSAGE_FAILED	Storing of the data for the description of the SE API failed
ERROR_STORAGE_FAILURE	Execution of the Secure Element functionality to retrieve log message parts has failed
ERROR_CERTIFICATE_EXPIRED	Storing of the log message has failed
ERROR_USER_NOT_AUTHORIZED	The user who has invoked the function is not authorized to execute this function
ERROR_USER_NOT_AUTHENTICATED	The user who has invoked the function has not the status authenticated

Example

```
asigntseEXE.exe cv_mapERStoKey clientId_! serialNumberKey
```

cv_registerClientId

Register specified clientId.

Parameters

In/Out	Required?	Name	Description
in	REQUIRED	**clientId**	
in	REQUIRED	**AdminUser**	
in	REQUIRED	**AdminPassword**	

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_registerClientId --authenticateuser username password
```

cv_getUniqueId

Get an identifier guaranteed to be unambiguous for every TSE.

Return Codes

Code	Description
EXECUTION_OK	Execution of the function has been successful.
ERROR_API_NOT_INITIALIZED	The API has not been initialized
ERROR_SECURE_ELEMENT_DISABLED	The Secure Element has been disabled

Example

```
asigntseEXE.exe cv_getUniqueId
```

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.