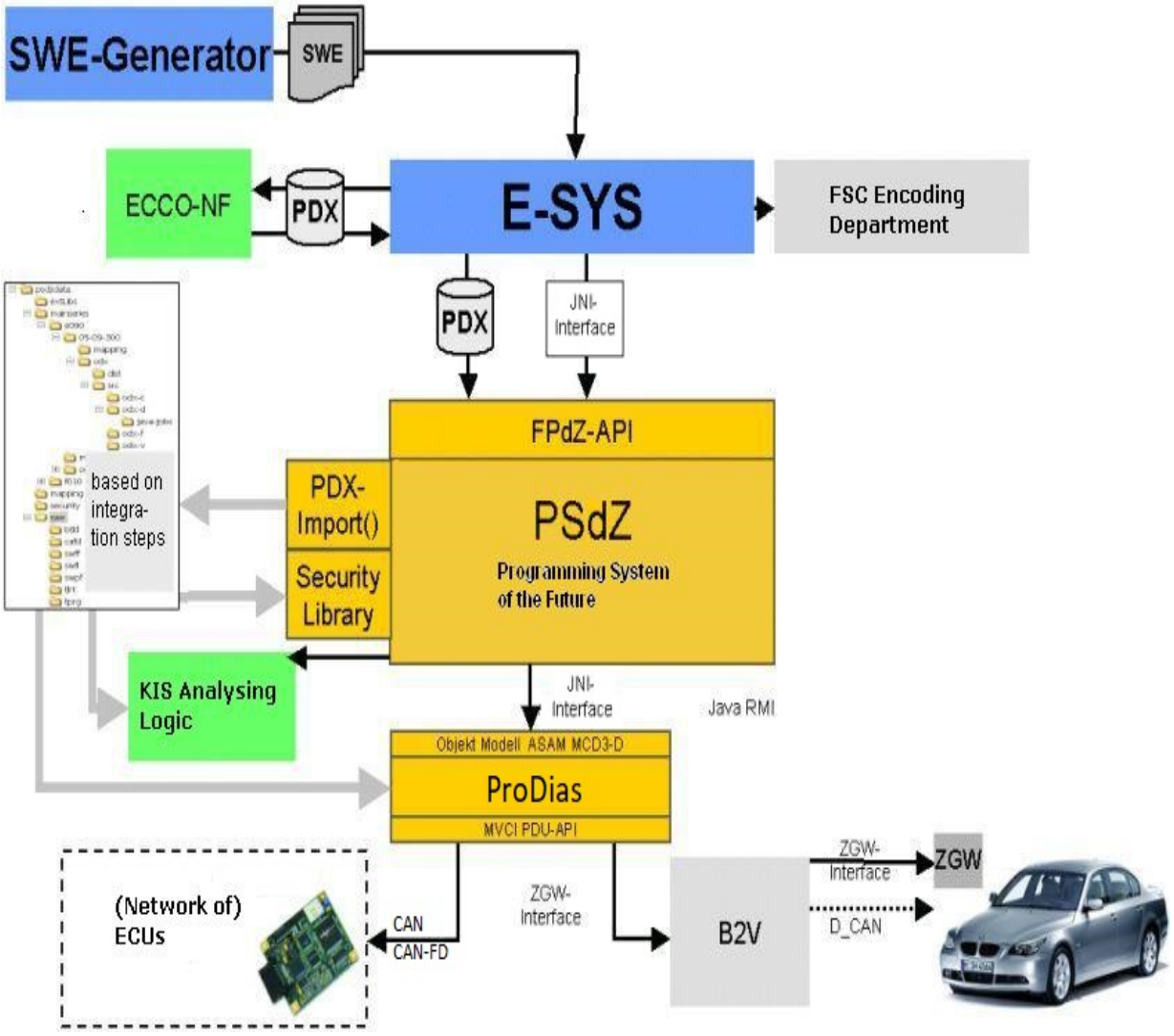
Introduction

Welcome to E-Sys Help System! With E-Sys you can program / flash Emus and encode them.
In addition to that E-Sys offers a bunch of editors and viewers for editing the populating with data necessary for that.

Subsequent we display the architecture of the systems involved in the programming:



E-Sys needs a correct populating with data for programming and encoding. Also necessary is a core program system that provides the services for communication and logistic. For the populating with data a

PDA teplate is unnecessary with populating of communication (diagnosis, protocol, connection information) for the board net system. The software logistics provides this template.

In this template the developer inserts his software to be transferred in E-Sys into PDX-Charger in the form of SWEs. The SWEs are generated from SWE-Generator.
The PSdZ ("Programmiersystem der Zukunft" programming system of the future) provides the services necessary for programming and encoding. For the implementation of the communication services PSdZ uses the D-Server; for the implementation of the logistic services KIS(compatibility and information system [still not completed]) is used. The implementation on the CanCard interface happens directly by the D-Server, the implementation of the communication over Ethernet/ZGW happens by the B2V-Server (BusinessToVehicle).
To realize programming and/or encoding the developer needs a TAL (transaction list) which contains at least one SWDeploy transaction (for programming SWEs) or one CDDeploy transaction (for encoding). Furthermore a FA ("Fahrzeugauftrag" vehicle order) is necessary.
E-Sys provides the TAL-Editor and FA/FP-Editor for creation and editing of TALs and FAs, besides the module TAL-Calculating for calculating a TAL from a pair of actual system installation table ("SVT-Actual") and reference system installation table ("SVT-Target"); SVT abbreviates "System-Verbau-Tabelle". Also in the module VCM all elements (TAL, FA, SVT-Actual and SVT-Target) of the VCM (Vehicle Configuration Management) can be read.
E-Sys provides an own module Coding for encoding. Here the codable states are detected and executed in the module TAL- Processing. This coding data can be inspected in E-Sys in CAF-Viewer (CAF=CodingApplicationFile). The coding data read out from the ECU can be inspected and changed in the format of a FDL (Function Data List) in FDL-Editor.
E-Sys is delivered (Stand 1.0.0) with a populating of data for programming and encoding of the evalboard (diagnosis address 7E) by CANCardX. For programming and encoding by ZGW an adequate populating of data must be obtained from the software logistic and imported in PDA-Charger.

Training of vehicle programming and E-Sys handling can be found at the BMW-Intranet:

Basic training vehicle-programming (focus for new employees who need to know the basic of the vehicle programming at BMW)
Advanced training of the E-Sys handling (focus for employees who need a more detailed handling

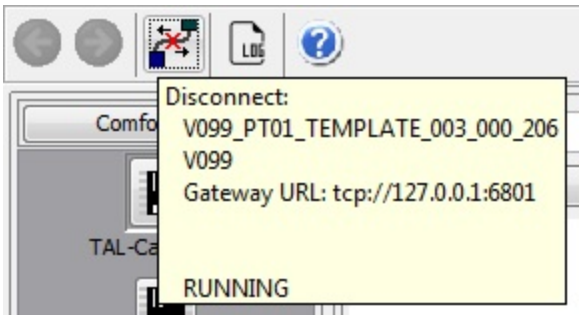
GUI

The user interface of E-Sys provides the standard features of a Windows interface so you can be familiar soon with the handling of E-Sys.
The GUI is adapted to the look and feel of Microsoft Outlook. The base elements you use the most time are set forms, dialogs, symbol bars, menus and the buttons for switching between the set forms.

The E-Sys user interface consists of four interaction elements:

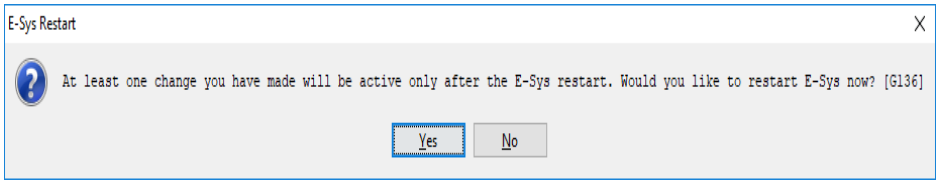
- **menu bar:** The discrete functions of the program can be called by menu items. The structure of the menu is divided into function groups (upper menu bar) and sub-functions called by sub menus.
- **symbol bar:**
  Important functions can be called by special buttons in the symbol bar by a mouse click.
- **navigation bar:**
  The E-Sys user interface uses some views for data input. These are called by special buttons in the left navigation bar.
- **status bar:**
  Global data of the application are displayed in the status bar.

In addition, the tooltip displays further information. For example, in the following screen, the tooltip for Connection establishment disconnection is displayed.
Here you can find the current connection that was previously displayed in the status bar:



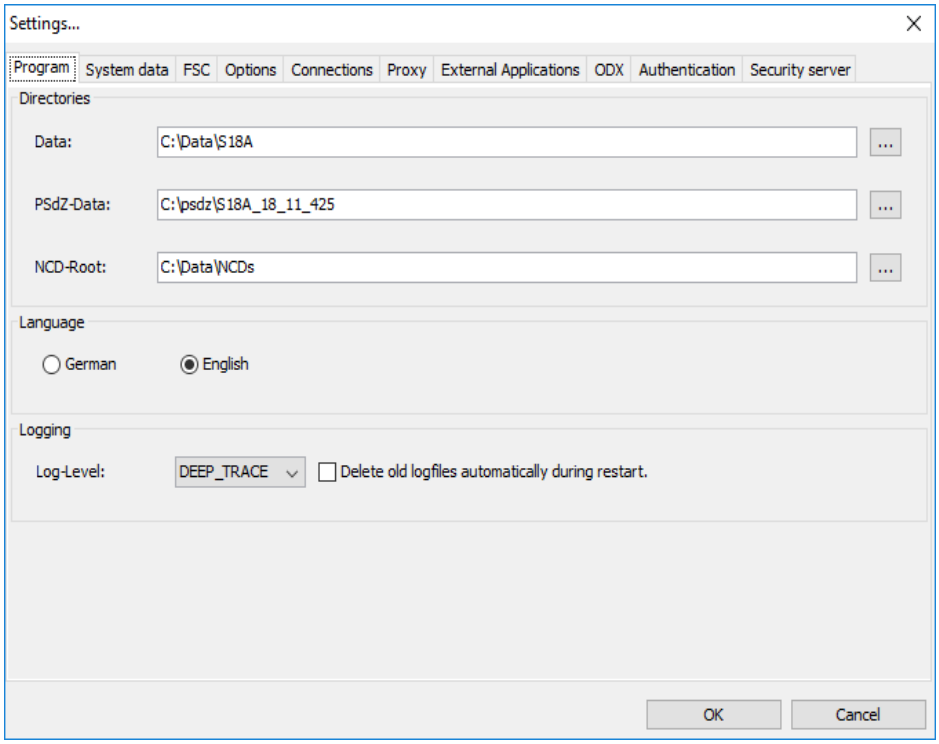
Options

In the dialog "Options" the global settings of E-Sys are specified. Some options require a new start of E-Sys to take effect. Then the following dialog appears:



After pressing the "Yes" button E-Sys will be closed and started again with the new settings.

Program



Directories	
Data	Data directory (Default: C:\Data)
PSdZ-Data	PSdZ-Data directory (Default: C:\Data).
NCD-Root	NCD-Root directory (Default: C:\Data\CAF\NCD).
Language	
German	Set to German language
English	Set to English language
Logging	
Log-Level	Setting of log levels
Delete old log files automatically during restart	During start all old log files are deleted

Directories:

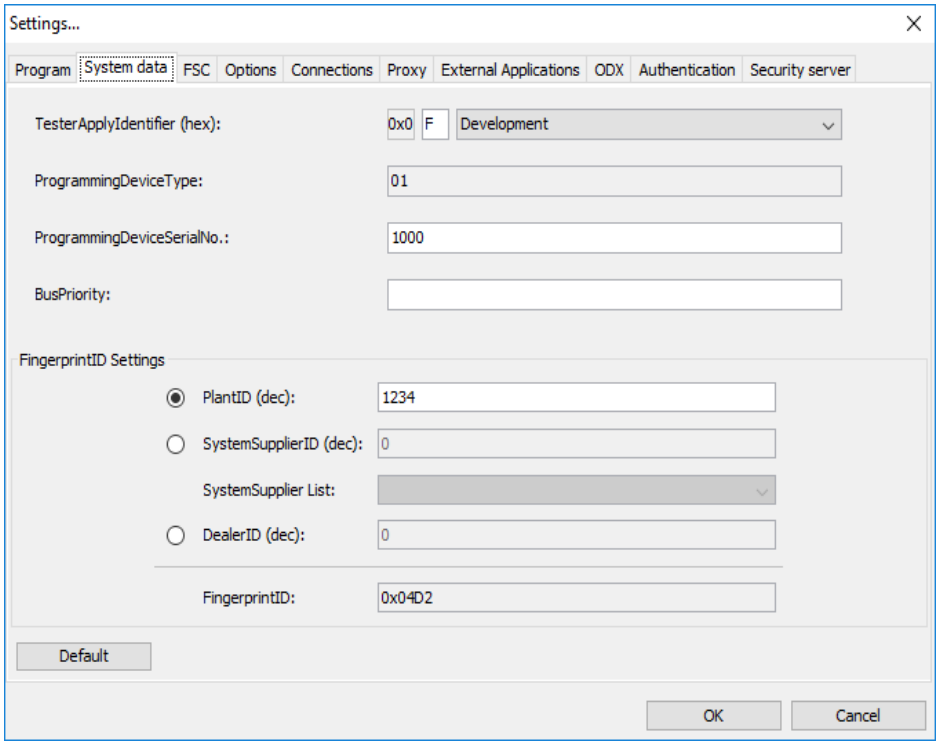
- In the D**ata** directory, E-Sys stores all relevant data (Log, TAL, executedTAL, SVT, FA, etc.).
- For **PSdZ-Data**, a directory must be entered in which the psdzdata directory is to be created. This is where the contents of the imported containers are stored.
- For **NCD-Root**, a directory must be entered in which the subdirectories "unsigned", "signed" and "default" (delivery state) should be created.
  This directory stores the net encoding data used for encoding.

Language:
Select the language used to represent E-Sys.

Logging:
Here you can set the information with which the E-Sys-Log is enriched.
The default setting is "DEEP_TRACE".
For analysis of errors (programming, coding or e-sys errors), the log level "DEEP_TRACE" is mandatory.

OFF:
The log does not include any additional log output.
WARN:
The log also includes WARN editions
INFO:
The log also includes INFO and WARN outputs
FINE:
The log also includes FINE, INFO and WARN outputs
DEBUG:
The log also includes DEBUG, FINE, INFO and WARN outputs
TRACE:
The log also includes TRACE, DEBUG, FINE, INFO and WARN outputs
DEEP_TRACE:
The log also includes DEEP_TRACE, TRACE, DEBUG, FINE, INFO and WARN outputs

System data

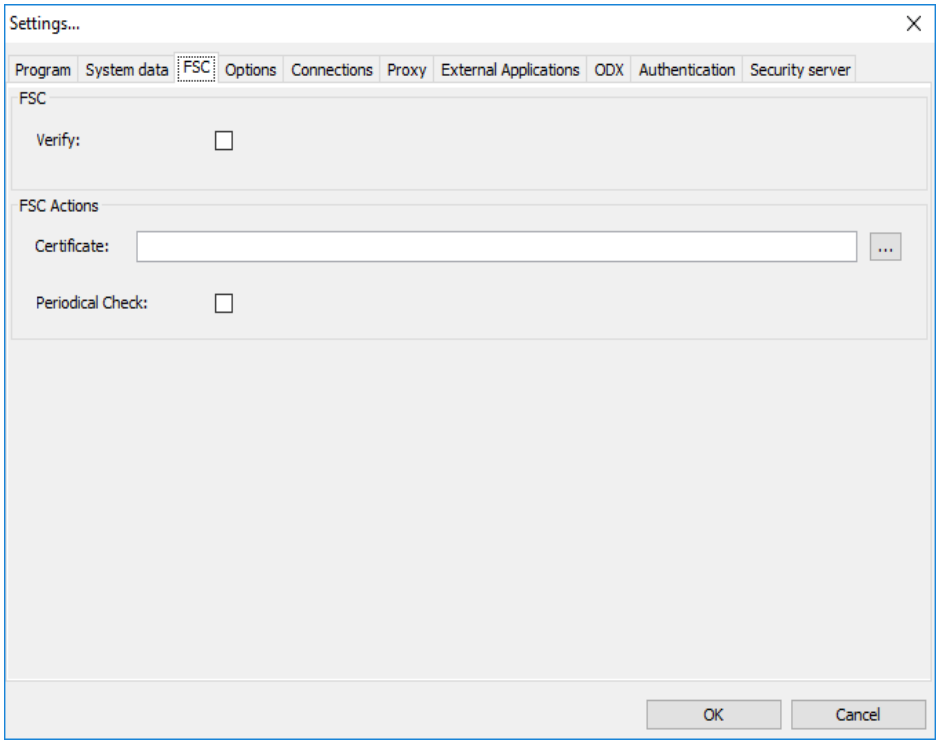


TesterApplyIdentifier (hex)	Specifies where the ECU was programmed (Systemsupplier, Development, RSU, ...)
ProgrammingDeviceType	Type of tester device.
ProgrammingDeviceSerialNo.	Unique identifier of programming device.
BusPriority	Specifies bus priority.
FingerprintID Settings	
PlantID (dec)	Specifies the identifier of the plant. (SystemSupplier, vehicle plant, HO, ...). The Plant ID (in which plant I am) must be known by the user himself.
SystemsupplierID (dec)	Specifies the Supplier-ID with following meaning: 0...99 (BCD coded e.g. 57 -> Fingerprint 0x0054) 100...153 (not allowed) 154...65535 (Decimal e.g. 154 -> Fingerprint 0x009A) Automatically set when selected via the "Systemsupplier List".
Systemsupplier List	Drop-down-List Suppliers Automatically set when selected via the "SystemsupplierID" (if the supplier is known).
DealerID (dec)	Specifies the number of the dealer.
FingerprintID	Fingerprint ID is calculated according to the selected fingerprint ID.
Default-Button	When the default button is set, the default values for the "System Data" tab are restored.

TesterApplyIdentifier:
The tester insert ID as well as the FingerprintID are elementary components of the fingerprint, which is written to the ECU before each programming.
Thus, it is understandable under which UseCase the control unit was last treated.

FingerprintID Settings:
Using the radio buttons, the user selects whether a plant ID (PlantID), a supplier ID (SystemsupplierID) or a dealer ID (DealerID) should be transferred to the fingerprint.

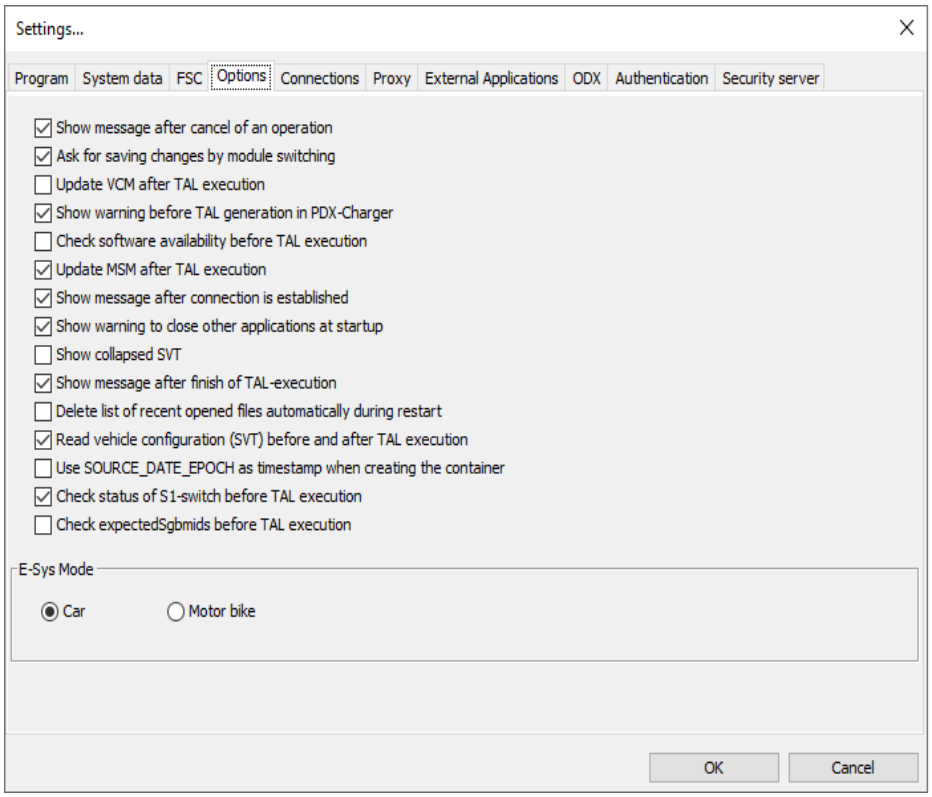
FSC



FSC	
Verify	Verify FSC; can be switched off for performance reason. Indicates whether the unlock codes should be validated regularly.
FSC Actions	
Certificate	Specification of a Certificate. In several functions this Certificate is compared with the Certificate in the ECU.
Periodical Check	Before every action the actual state is checked (still not implemented).

Additional information:
For more information about FSC (activation codes) an E-Sys, see comfort- or expert-mode.

Options



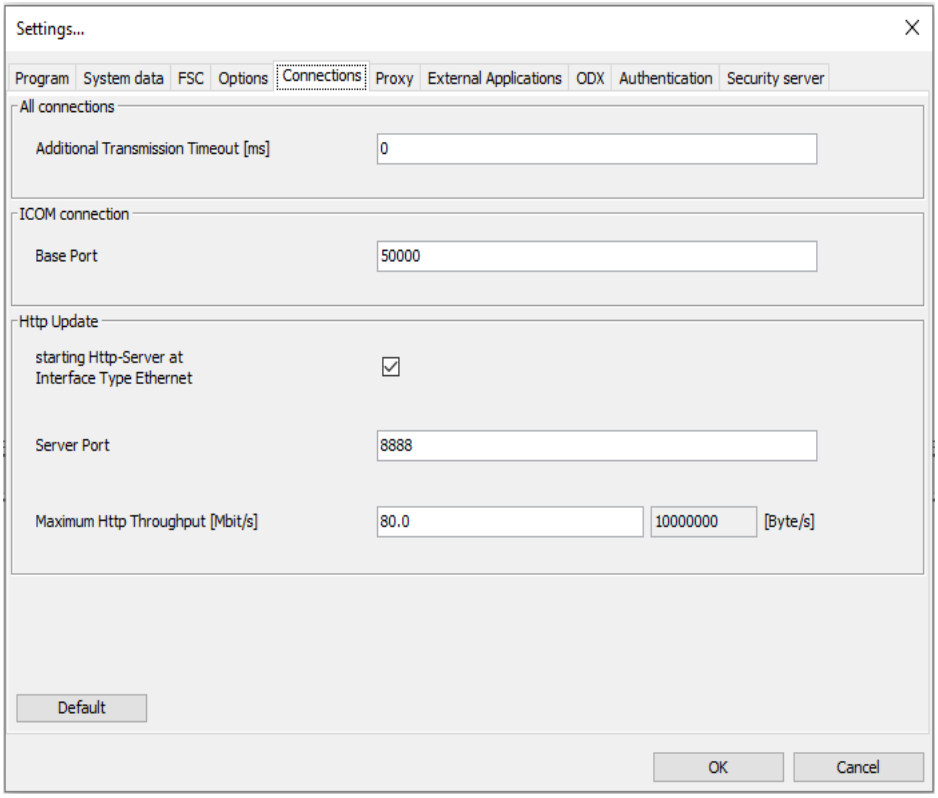
Show message after cancel of an operation	Here you can turn on and off the confirmation that an action has been canceled.
Ask for saving changes by module switching	Here you can turn on and off the query to save.
Update VCM after TAL execution	Select whether to update VCM.
Show warning before TAL generation in PDX-Charger	Here the warning can be switched on and off.
Check software availability before TAL execution	Select whether to check whether the SWEs to be used exist before processing the TAL.
Update MSM after TAL execution	Select whether to update MSM (Master-Securety Module). If so, the transport keys (TSL) are updated.
Show message after connection is established	Here you can switch on and off the confirmation that a connection has been established.
Show warning to close other applications at startup	Here you can switch on and off the Warning shown at startup.
Show collapsed SVT	Here you can specify if the SVT is shown collapsed or extended by default.
Show message after finish of TAL-execution	If enabled, a popup window is displayed after TAL-execution indicating the status.
Delete list of recent opened	Here you can decide if the list of recent opened files will be

files automatically during restart.	deleted after E-Sys restart or not.
Read vehicle configuration (SVT) before and after TAL execution.	If enabled, readSVT will be done before and after the TAL execution and will be logged in the Log File.
Use SOURCE_DATE_EPOCH as timestamp when creating the container	Select whether to use the current system time or the environment variable SOURCE_DATE_EPOCH when creating a PDX container.
Check status of S1-switch before TAL execution	Only relevant if TesterApplyIdentifier=0xF. This check is always carried out for TesterApplyIdentifier != 0xF. In this case, specifies whether the S1 test should be carried out or not. This check verifies that the primary and secondary energy-network are connected.
Check expectedSgbmids before TAL execution	Flag whether the list of expectedSgbmids shall be verified before TAL-execution.
E-Sys Mode	
Car	E-Sys is used for programming/coding car CONTROL units.
Motor bike	E-Sys is used for programming/coding motorcycle control units. (See Motorbike)

SOURCE_DATE_EPOCH:
The environment variable SOURCE_DATE_EPOCH is entered in the META data when the PDX containers are created.
If this value is not set, the current system time is used.
If the value is set, the contents of the environment variable SOURCE_DATE_EPOCH is used.
For more information, see: https://reproducible-builds.org/docs/timestamps/.
The user is responsible for ensuring that the variable contains the correct value.
If the variable is not set or does not contain a valid timestamp, the current system time is used.

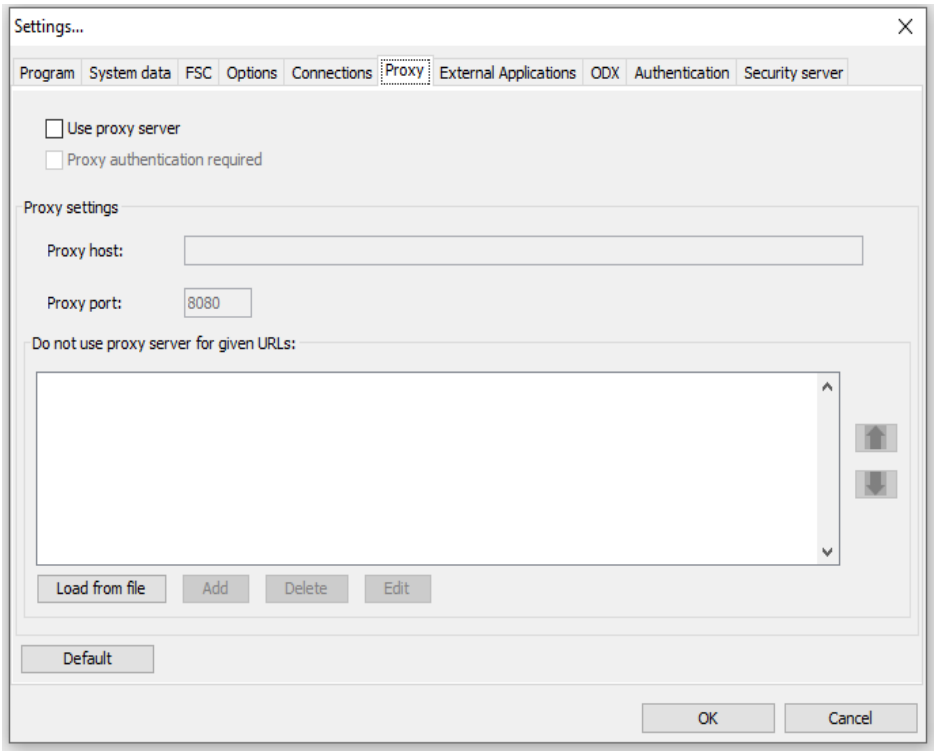
E-Sys Mode:
Selection of control units should be treated with cars or motorcycle routines.
Especially at VCM, different routines are available here.

Connections



All connections	
Additional Transmission Timeout [ms]	This time is independent of the selected interface with which the connection to the control unit / vehicle is established.
ICOM connection	
Base Port	Base Port, will be used to calculate the port of the connection using port mapping.
Http Update	
starting Http-Server at Interface Type Ethernet	Determines whether the PSdZ internal Http server should be used for flashing over Ethernet.
Server Port	Port on which the Http Update Server is started. 8888 is the number for the PSdZ internal server.
Maximum Http Throughput [Mbit/s]	Maximum Http throughput in Mbit per second that is allowed to the Http server. The input field can either be empty or contain a number between 0,008192 (1024 Byte/s) and 2500,0. Floating point numbers are allowed. The right, non-editable field shows the value converted into bytes per second.
Default-Button	When the default button is set, the default values for the "Connections" tab are restored.

Proxy



Use proxy server	Whether to use a proxy for the backend connections. Subsequent settings can only be edited if the check mark is set here.
Proxy authentication required	Switching on and off the using of authentication for proxy server. The prompt for entering the login data for proxy does not appear until a backend interface is called up.
Proxy settings	
Proxy host	URL of the proxy server
Proxy port	Port of the proxy server
Do not use proxy server for given URLs	
Load from file	Here you can load a txt file with different URLs.
Add	Allows you to add individual URLs.
Delete	This can be used to delete a selected URL (only single selection is possible).
Edit	Can be used to edit a selected URL.
Arrow-Buttons	The arrow buttons can be used to move a marked URL down or up in order.
Default-Button	When the default button is set, the default values for the "Proxy" tab are restored. (URLs will not be deleted!)

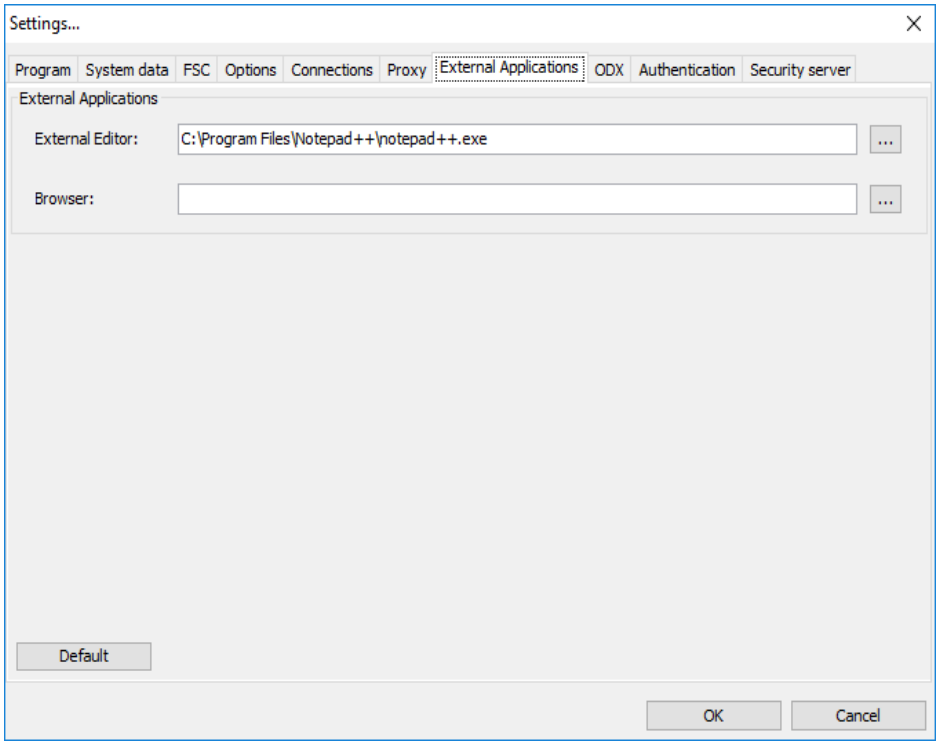
Do not use proxy server for given URLs:
The URLs entered here do not use the proxy.

Load file:
In the txt-file, the URLs may only be entered in one line and only separated with a semicolon ";". E-Sys does not support another formats of the URLs.
Example: https://www.google1.de;https://www.bmwgroup.com;https://www.bing2.com

Add:
Only one URL may be added at a time.
Multiple entries (separated with semicolon ";") are invalid.

Additional information:
Only BASIC is supported as an authentication method for a proxy.

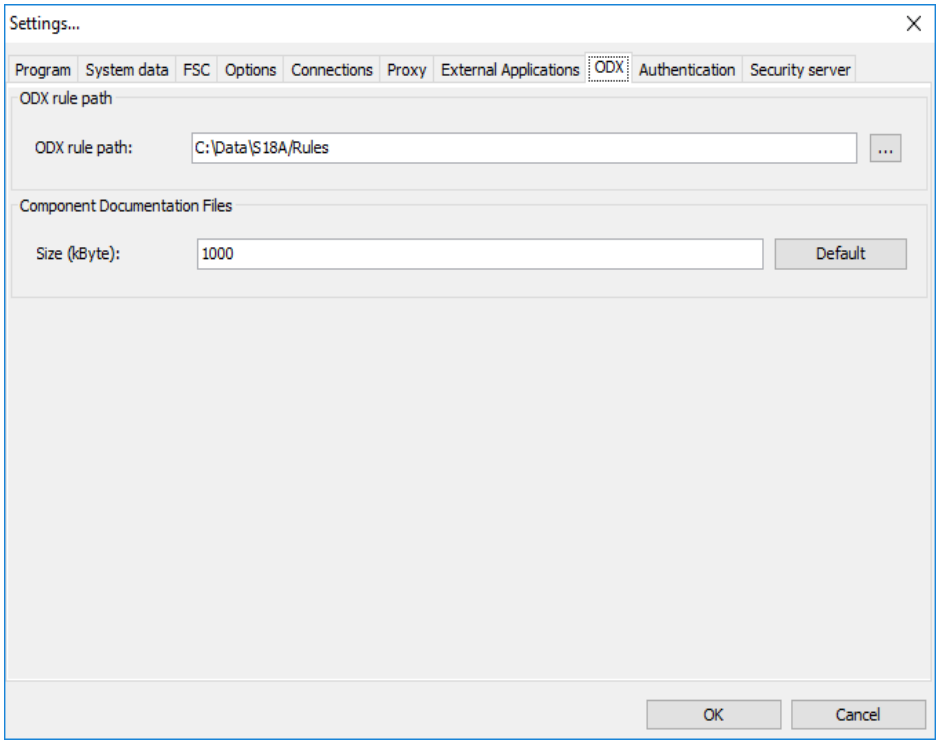
External Applications



External Applications	
External Editor	Path of the external editor.
Browser	Path of the external browser.
Default-Button	When the default button is set, the default values for the "External Applications" tab are restored. (Both fields are emptied)

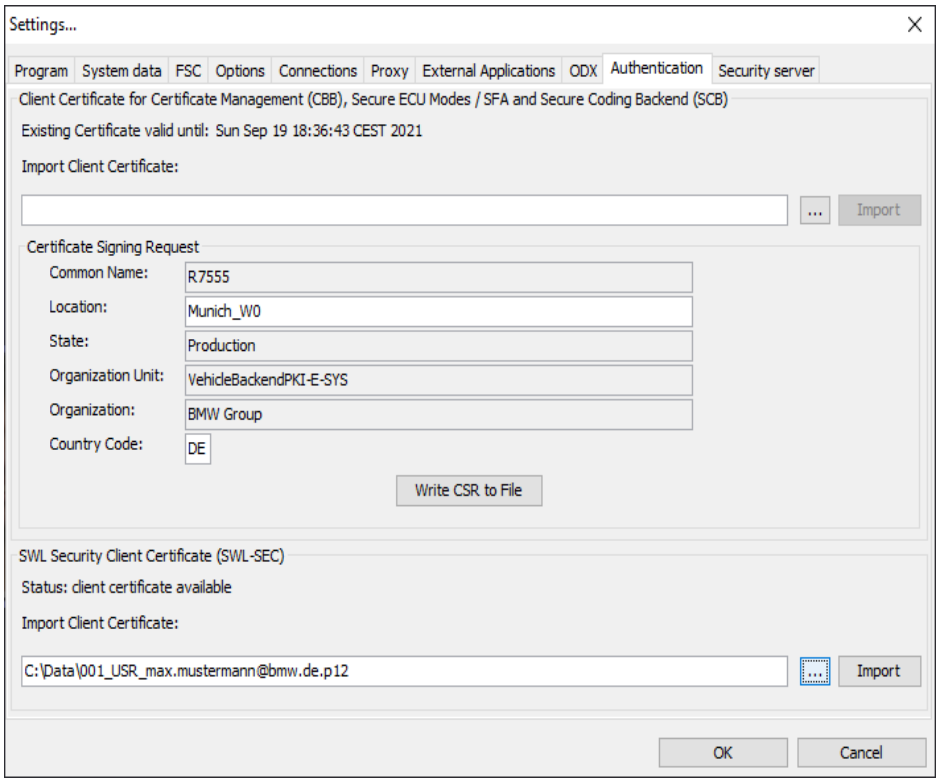
Additional information:
In the "External Applications" tab, the editor and the browser are indicated to display data.

ODX



ODX rule path	
ODX rule path	Path of the ODX rule directory.
Component Documentation Files	
Size (kbyte)	Maximal size of the documentation in a component-container.
Default-Button	Pressing the default button restores the default value for the size of the "Component Documentation".

Authentication



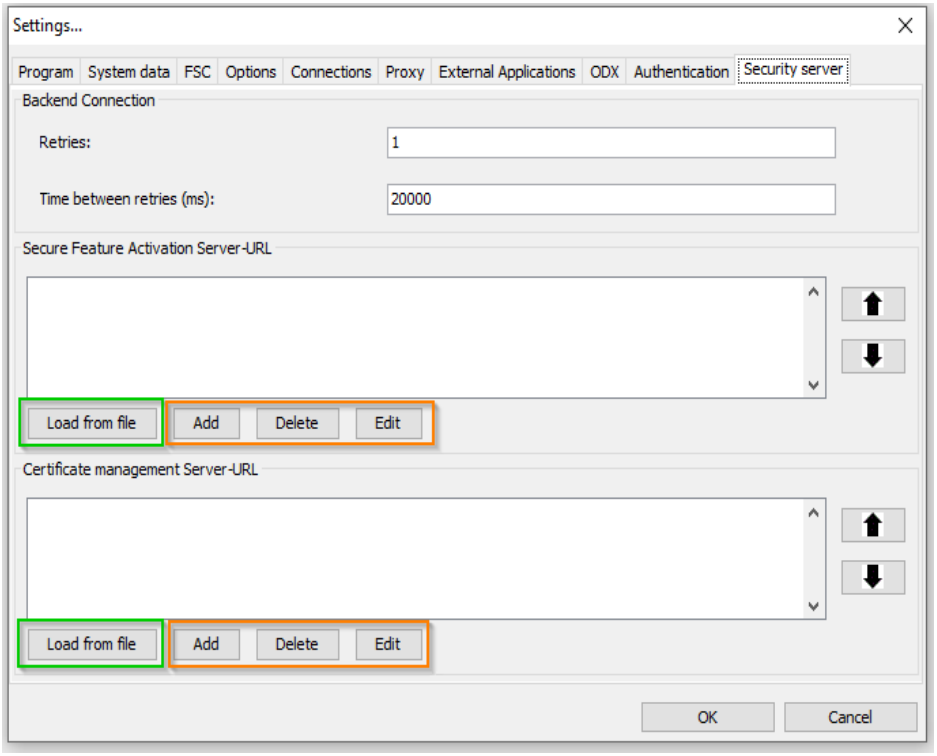
Client Certificate for Certificate Management (CBB), Secure ECU Modes / SFA and Security Coding Backend (SCB)	
Import Client Certificate	Import functionality of a client Certificate for connecting to the CBB. Connection to the CBB is required as part of Certificate management in order to calculate bindings for ECUs.
Import Client Certificate	Import functionality of a client Certificate for connecting to the SWL-SEC. Connection to the SWL-SEC is required as part of secure coding in order to sign net coding data.
Certificate Signing Request	
Common Name	Input field (cannot be changed)
Location	Input field (can be customized e.g. "Munich_W0")
State	Input field (cannot be changed)
Organization Unit	Input field (cannot be changed)
Organization	Input field (cannot be changed)
Country Code	Input field (can be customized e.g. "DE")
Write CSR to File	Create a CSR

Client Certificate for CBB and Secure ECU Modes / SFA:
Here, the client Certificate is imported.
This is required to communicate with the back-end systems for Certificate management (CBB) as well as for SFA (Secure Feature Activation) from the BMW-Intrant.

Certificate Signing Request (CSR):
Here, a client CSR file is generated based on the input fields for the current computer and stored locally (see pop-up window).
This client CSR file can be used to generate a client Certificate through the b2b portal (see howtoo at BMW-Intranet).
The client Certificate can be imported into E-Sys via "Import Client Certificate".

Additional information:
A client Certificate is required to communicate with the CBB (backend for Certificate Management functionality) or the SFA backend.
You can only import a client Certificate if it matches the last generated client CSR file.

Security server



Backend Connection	
Retries	Number of attempts to establish a connection with a backend in the event of an error. If the connection to 1.Server URL fails, repeating >0 will proceed to the next server URL in the list.
Time between retries (ms)	Time to wait before attempting to use a new server URL.
Secure Feature Activation	

Server-URL	
Load from file	Here you can load a txt file with different URLs. (green marking)
Add	Allows you to add individual URLs. (orange marking)
Delete	This can be used to delete a selected URL (only single selection is possible). (orange marking)
Edit	Can be used to edit a selected URL. (orange marking)
Certificate management Server-URL	
Load from file	Here you can load a txt file with different URLs. (green marking)
Add	Allows you to add individual URLs. (orange marking)
Delete	This can be used to delete a selected URL (only single selection is possible). (orange marking)
Edit	Can be used to edit a selected URL. (orange marking)
Arrow-Buttons	The arrow buttons can be used to move a marked URL down or up in order.

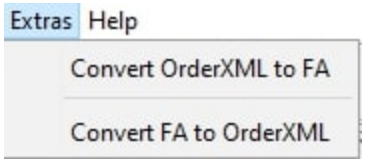
Load file:
In the txt-file, the URLs may only be entered in one line and only separated with a semicolon ";". E-Sys does not support another format of the URLs.
Example:
https://www.google1.de;https://www.bing2.com;https://www.bing3.com;https://www.bing4.com

Add:
Only one URL may be added at a time.
Multiple entries (separated with semicolon ";") are invalid.

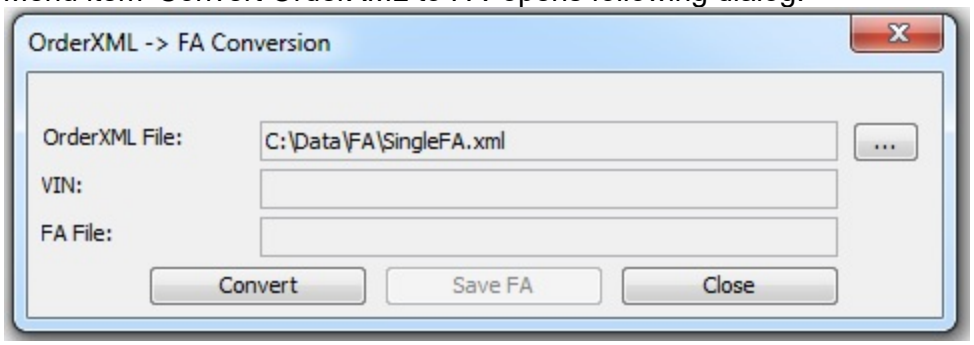
Additional information:
Information on the subject of SFA and SFA backend URL can be found on the BMW intranet here.
Information on Certificate management and CBB URLs can be found on the BMW intranet here.

Extras

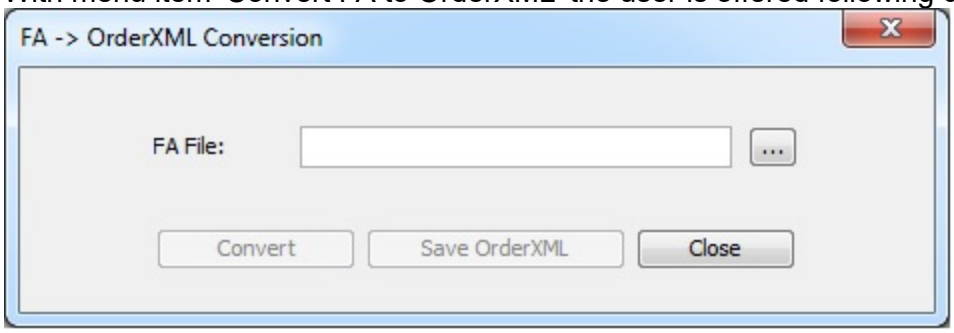
The Extras menu offers functions for converting FA-files:



Menu item 'Convert OrderXML to FA' opens following dialog:



With menu item 'Convert FA to OrderXML' the user is offered following dialog:



Comfort Mode

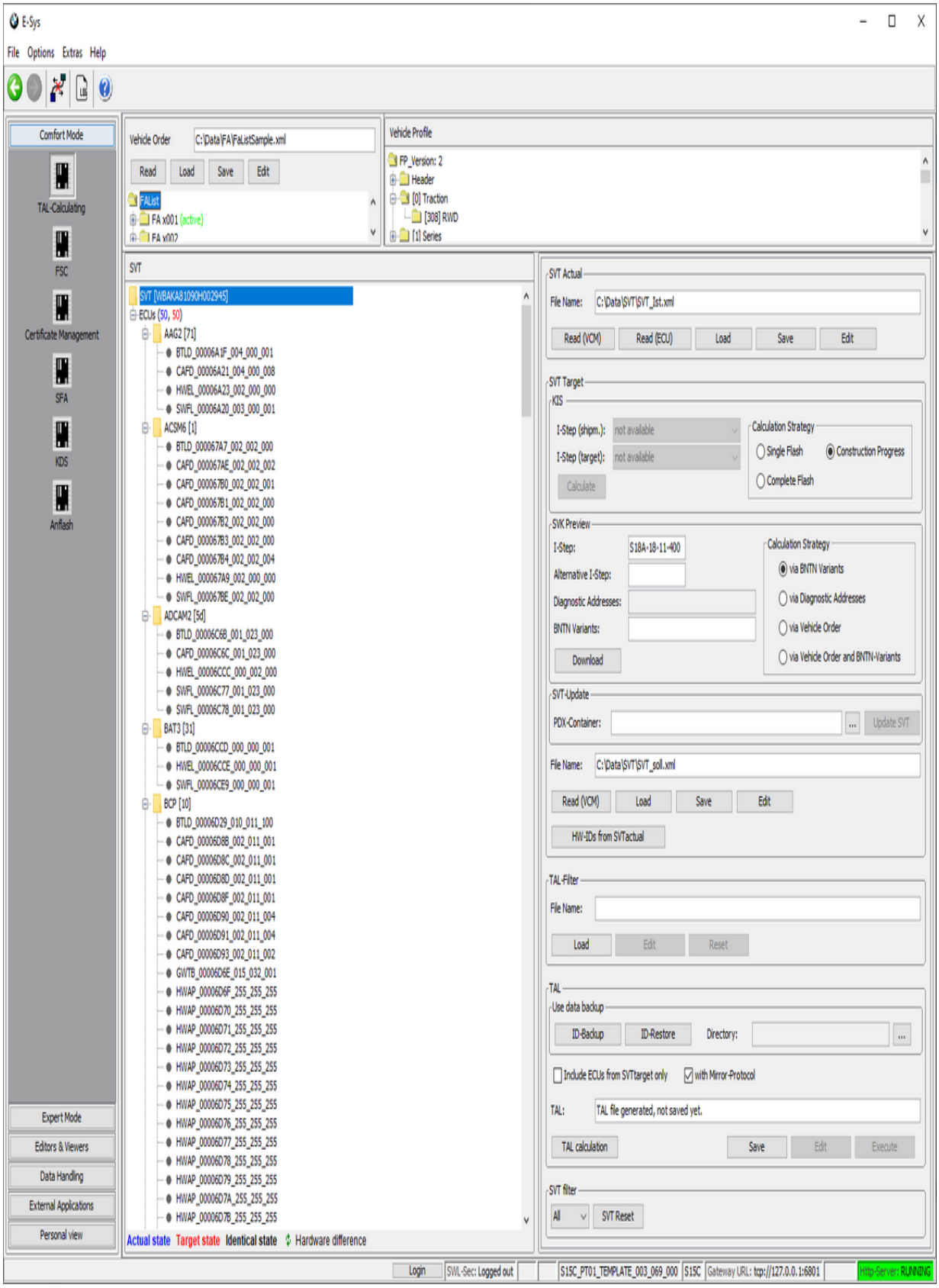
Calculation of TAL
FSC
Certificate-Management
SFA / ECU-Modes
KDS
ANFLASH

Calculation of TAL

The module "Calculation of TAL" calculates a TAL (transaction list) from a pair of actual system installation table ("SVT-Actual") and reference system installation table ("SVT-Target"). This TAL can be executed in the module TAL-Processing. SVT-Actual specifies the actual state or installation of the system i.e. which hardware unities and which software unities reside in the ECU/vehicle.
SVT-Target specifies the state of the ECU/vehicle to be reached after the processing of the TAL.

By clicking the checkbox "Use data backup" and specification of the backup directory the individual information of the former exchanged ECU can be taken-up into the TAL creation process.

The backup data is created in mask TAL-Processing.


E-Sys
File Options Extras Help
Comfort Mode
TAL-Calculating
FSC
Certificate Management
SFA
KDS
Anflash
Expert Mode
Editors & Viewers
Data Handling
External Applications
Personal view
Vehicle Order
C:\Data\FA\FaListSample.xml
Read Load Save Edit
FAList
FA x001 (active)
Vehicle Profile
FP_Version: 2
Header
[0] Traction
[308] RWD
[1] Series
SVT
SVT [WBAKA81090H002945]
ECUs (50, 50)
AAG2 [71]
BTLD_00006A1F_004_000_001
CAFD_00006A21_004_000_008
HWEL_00006A23_002_000_000
SWFL_00006A20_003_000_001
ACSM6 [1]
BTLD_000067A7_002_002_000
CAFD_000067AE_002_002_002
CAFD_000067B0_002_002_001
CAFD_000067B1_002_002_000
CAFD_000067B2_002_002_000
CAFD_000067B3_002_002_000
CAFD_000067B4_002_002_004
HWEL_000067A9_002_000_000
SWFL_000067BE_002_002_000
ADCAM2 [5d]
BTLD_00006C6B_001_023_000
CAFD_00006C6C_001_023_000
HWEL_00006CCC_000_002_000
SWFL_00006C77_001_023_000
SWFL_00006C78_001_023_000
BAT3 [31]
BTLD_00006CCD_000_000_001
HWEL_00006CCE_000_000_001
SWFL_00006CE9_000_000_001
BCP [10]
BTLD_00006D29_010_011_100
CAFD_00006D8B_002_011_001
CAFD_00006D8C_002_011_001
CAFD_00006D8D_002_011_001
CAFD_00006D8F_002_011_001
CAFD_00006D90_002_011_004
CAFD_00006D91_002_011_004
CAFD_00006D93_002_011_002
GWTB_00006D6E_015_032_001
HWAP_00006D6F_255_255_255
HWAP_00006D70_255_255_255
HWAP_00006D71_255_255_255
HWAP_00006D72_255_255_255
HWAP_00006D73_255_255_255
HWAP_00006D74_255_255_255
HWAP_00006D75_255_255_255
HWAP_00006D76_255_255_255
HWAP_00006D77_255_255_255
HWAP_00006D78_255_255_255
HWAP_00006D79_255_255_255
HWAP_00006D7A_255_255_255
HWAP_00006D7B_255_255_255
Actual state Target state Identical state Hardware difference
SVT Actual
File Name: C:\Data\SVT\SVT_Ist.xml
Read (VCM) Read (ECU) Load Save Edit
SVT Target
KIS
I-Step (shipm.): not available
I-Step (target): not available
Calculate
Calculation Strategy
Single Flash
Construction Progress
Complete Flash
SVK Preview
I-Step: S18A-18-11-400
Alternative I-Step:
Diagnostic Addresses:
BNTN Variants:
Download
Calculation Strategy
via BNTN Variants
via Diagnostic Addresses
via Vehicle Order
via Vehicle Order and BNTN-Variants
SVT-Update
PDX-Container:
Update SVT
File Name: C:\Data\SVT\SVT_soll.xml
Read (VCM) Load Save Edit
HW-IDs from SVTactual
TAL-Filter
File Name:
Load Edit Reset
TAL
Use data backup
ID-Backup ID-Restore Directory:
Include ECUs from SVTtarget only
with Mirror-Protocol
TAL: TAL file generated, not saved yet.
TAL calculation Save Edit Execute
SVT filter
All
SVT Reset
Login
SWL-Sec: Logged out
S15C_PT01_TEMPLATE_003_069_000
S15C
Gateway URL: tcp://127.0.0.1:6801
Http-Server: RUNNING

GUI Elements

Vehicle data	
Vehicle Order	File containing a list of vehicle orders. For selecting a certain vehicle order, this item needs to be activated by using the context menu item ‘Activate FA’. If the list of vehicle orders contains only one element, this order will be automatically activated as soon as the file is loaded.
Read	Read current vehicle order from vehicle.
Load	Load a FA from file system.
Save	Save the current FA.
Edit	Load current vehicle order to FA/FP editor.
Vehicle Profile	The Vehicle profile will be automatically generated from an activated Vehicle Order and displayed in a tree structure.
SVT Actual	The actual SVT
Read (VCM)	Read SVT from vehicle.
Read (ECU)	Read SVT from ECU (functionally).
Load	Load a SVT from file system.
Save	Save the SVT-Actual.
Edit	Edit in the corresponding editor.
SVT Target	The reference SVT.
Read (VCM)	Read SVT-Target from VCM
Load, Save	Load/save an SVT-Target from/to file system.
File Name	The name and path of the SVT file.
Edit	Edit the SVT-Target in the corresponding editor.
HW-IDs from SVTactual	HWEL, HWAP und HWFR elements from SVTactual will be copied to the Emus in the target SVT.
SVT Target (KIS calculation)	
I-Step (shipm.)	Available shipment integration steps.
I-Step (target)	Available target integration steps.
Calculation	Possible strategy to calculate the SVT (Single

strategy	Flash | Complete Flash | Construction Progress).
Calculate	Calculates an SVT-Target by integration step and vehicle order (and SVT_Actual if exists).
SVT Target (SWL-Cockpit)	
I-Step	Text field for the I-Step of the SVT-Target (mandatory)
Alternative I-Step	optional field for an alternative I-Step
Diagnostic Addresses	comma separated list of diagnostic addresses (in hex-format)
BNTN Variants	comma separated list of BNTN-Variants
Calculation strategy	Possible strategy to calculate the SVT (via BNTN-Variants | via Diagnostic Addresses | via Vehicle Order | via Vehicle Order and BNTN-Variants).
Download	Sends request to SWL-Cockpit and receives a calculated SVT-Target.
SVT Target (PDA-Update)	
Load PDA-Container	Load PDA-Container for updating SVT
Update SVT	Start SVT-Update process
TAL-Filter	
File Name	Filename of the TAL-Filter
Load	Opens a file dialogue to load an existing TAL-Filter from a file. The settings from the TAL-Filter will then be visible in the SVT-View, the file path is displayed in the text field. An already loaded/opened filter will be discarded.
Edit	Edit in the corresponding editor. This button is only active if there is an active filter.
Reset	Deletes the active filter together with all corresponding symbols in the SVT-View. This button is only activated if there is an active filter.
TAL Individual data recovery	
ID-Backup	Calculate the IDR-Backup TAL using all necessary

	data.
ID-Restore	Calculate the IDR-Restore TAL using all necessary data.
Directory	The backup directory that contains the individual data. When calculating IDR-Restore TAL, the specification is mandatory.
TAL	
Emus from SVTsoll	If checked only Emus from the target SVT will be considered for the TAL generation.
with Mirror-Protocol	If checked the TAL with mirror protocol actions will be calculated.
TAL Calculation	Calculate the normal TAL or TAL with mirror protocol actions using all necessary data.
Save	Save the generated TAL.
Edit	Edit in the corresponding editor.
Execute	Transfer the TAL into module TAL-Processing.
SVT-Filter	
Filter	Filter the SVT tree in according to the selected transaction.
SVT Reset	Reset SVT-Actual/Soll and delete SVT tree.

SVT-Update

Within section 'SVT-Update' it is possible to update all versions of the SGBMIDs included in the Target-SVT.
With button 'Update SVT' E-Sys will start a check for each SGBMID contained in the SVT and search for versions of that SGBMID in the loaded PDA-Container.
If a different version of a SGBMID was found, the version in the SVT will be updated by the one from the PDA-container.

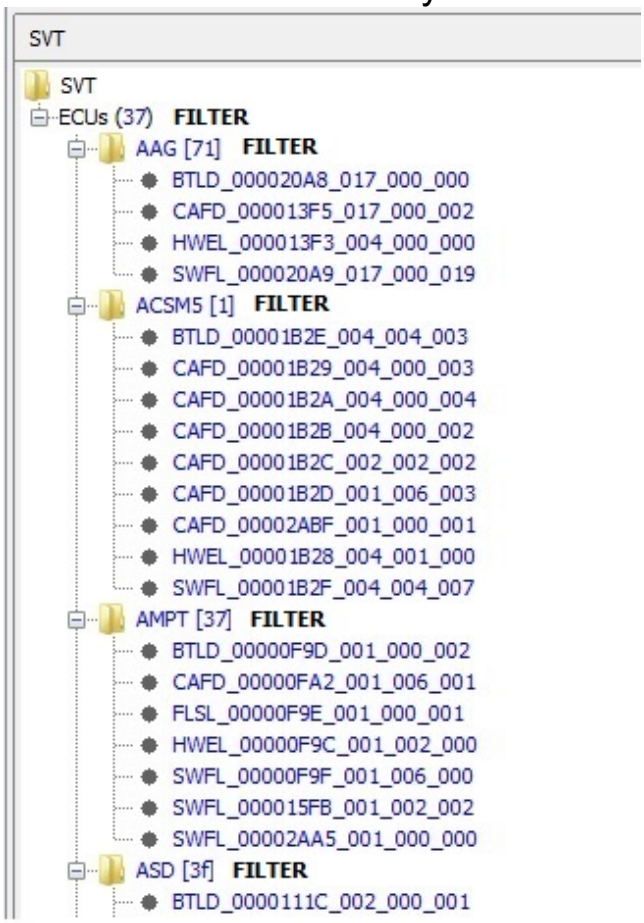
Reading Vehicle Order and VIN

After reading the vehicle order the VIN will always be read too. The VIN will be shown in the element
""FAList/FA/FZAuftrag/Header/Vinlong". The vehicle order and the VIN

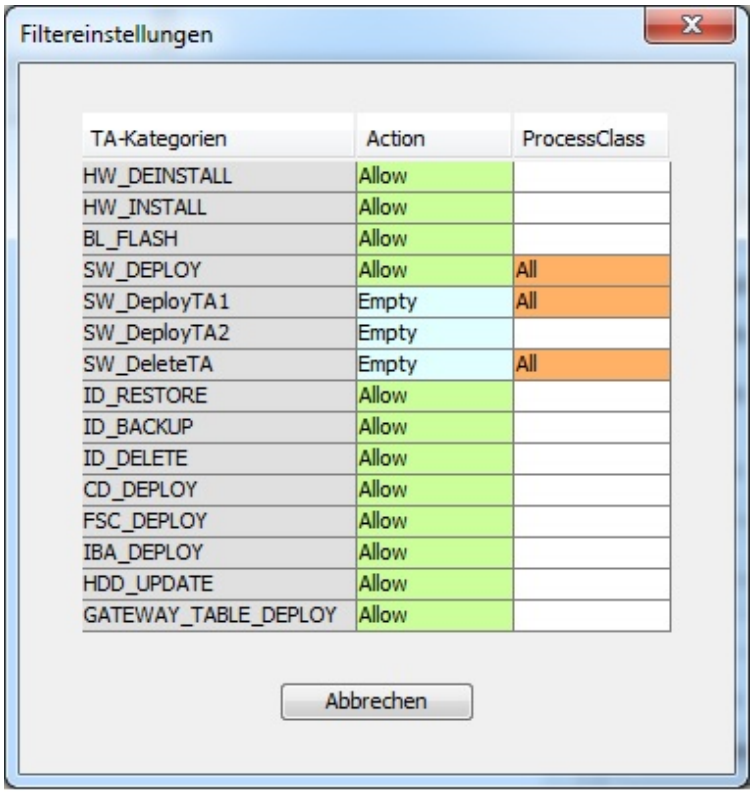
will be read from the master ECU. If an error occurs they will be read from the backup ECU. In the status bar is shown where the vehicle order and the VIN were actually read from.

Displaying TAL-Filter settings

If filter settings have been defined for the root element or any of the ECU elements in the SVT tree view, these elements will be displayed with an additional filter symbol as soon as the TAL-Filter file is loaded.



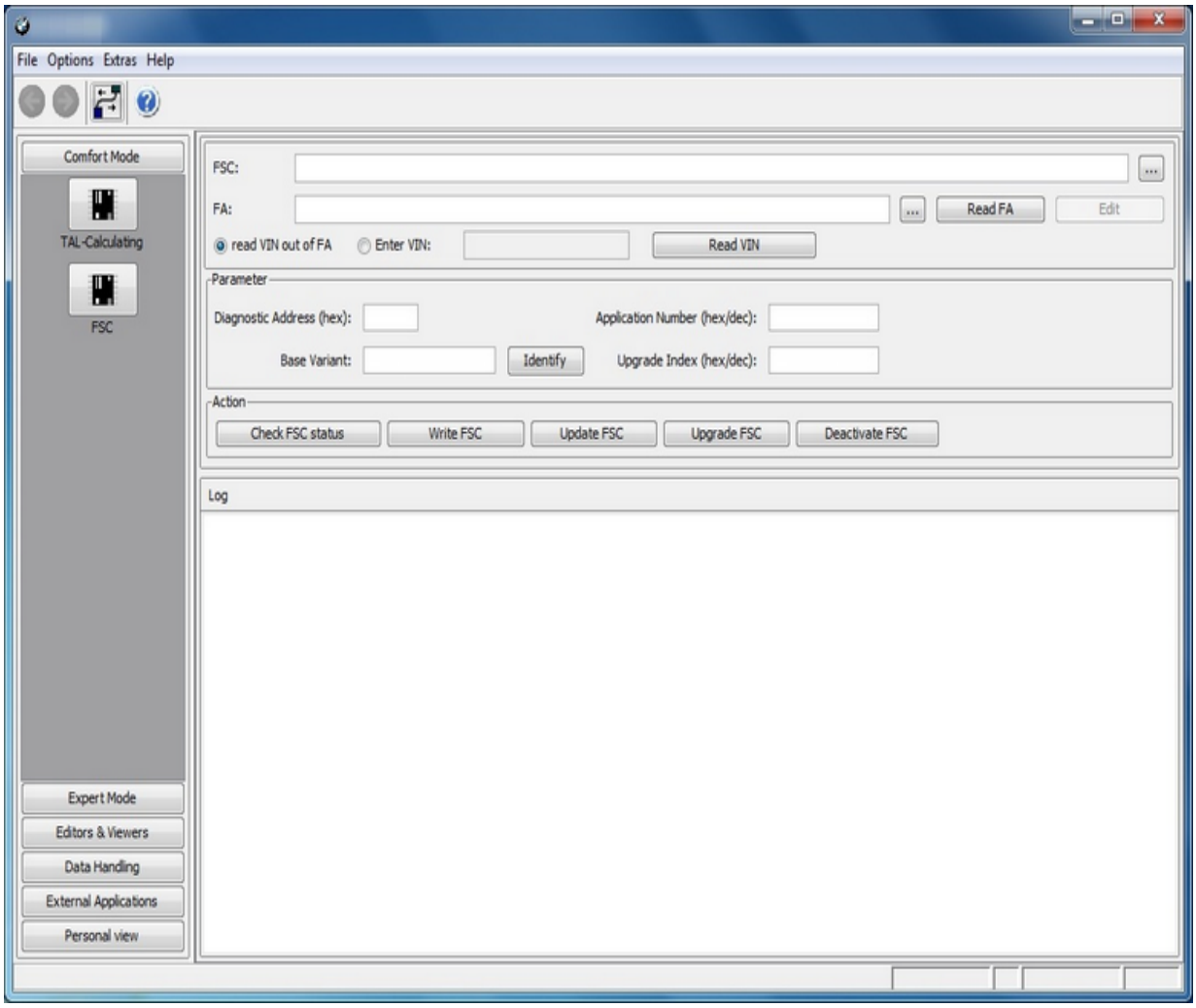
A detailed figure showing all filter settings of an element will be displayed via the context menu entry 'Filter Settings'.


Filtereinstellungen
TA-Kategorien
Action
ProcessClass
HW_DEINSTALL
Allow
HW_INSTALL
Allow
BL_FLASH
Allow
SW_DEPLOY
Allow
All
SW_DeployTA1
Empty
All
SW_DeployTA2
Empty
SW_DeleteTA
Empty
All
ID_RESTORE
Allow
ID_BACKUP
Allow
ID_DELETE
Allow
CD_DEPLOY
Allow
FSC_DEPLOY
Allow
IBA_DEPLOY
Allow
HDD_UPDATE
Allow
GATEWAY_TABLE_DEPLOY
Allow
Abbrechen

FSC

The module FSC serves for inquiring the FSC's state and for writing of FSC into ECU. In addition to that it's possible to (de)activate FSC's already existing in the ECU. Only an activated FSC can really be used.

Corresponding to the topic FSC there are other masks in E-Sys: FSC's can be displayed and edited in **FSC-Editor** easily. In expert mode **FSC-Extended-Module** is available in addition to that.



GUI Elements

FSC File	The FSC to be written.
FA File	the vehicle order corresponding to the ECU.
Diagnosis	of ECU

Address	
Software-ID	Identifier of software
Base Variant	Name of ECU variant
Upgrade Index	of FSC
Poll FSC State	The State of FSC in the ECU is inquired.
Write FSC	The FSC loaded into E-Sys is written to the ECU.
Update FSC	The written FSC (to ECU) is checked and activated.
Upgrade FSC	Combines write FSC and update FSC.
Deactivate FSC	Deactivate FSC in ECU (set to invalid)

ATTENTION

The functionalities provided in this module which include direct server access require to have several preconditions to be fulfilled:

- The system which runs E-Sys must be in a BMW network segment in which the backend systems (CBB) are available.
- There must be a valid E-Sys client Certificate imported into E-Sys.
- At least one URL of a valid backend system (CBB) must be configured in E-Sys.

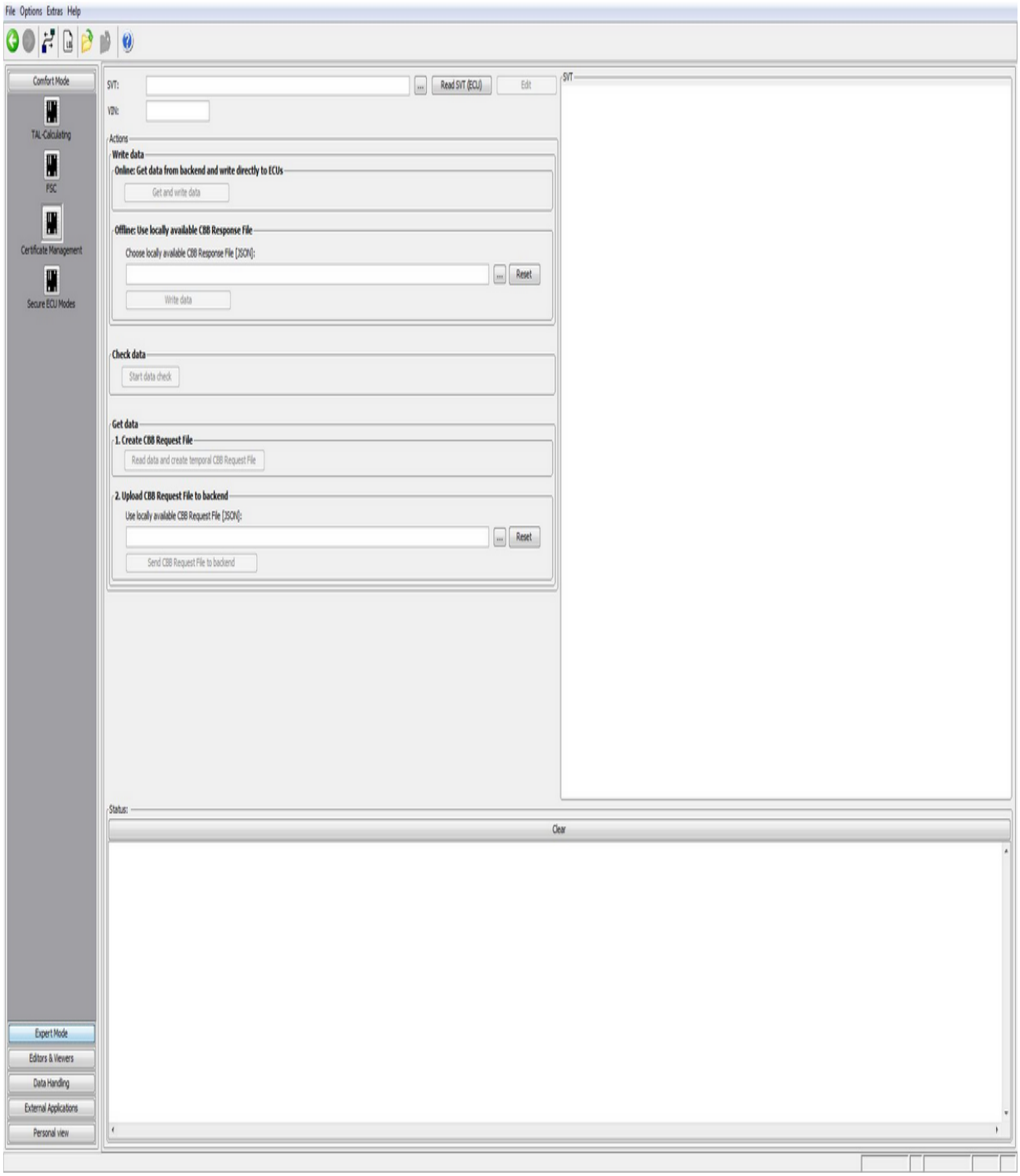
A more detailed set of instructions can be found here: HOWTO (only available within BMW network).
For further questions please contact the support service: support-security-systemfunktionen@bmw.de

Introduction:

As an enabler for new functions and to increase the security of existing functions, individual Certificates are required in the treatment of vehicles. From SP2018, individual Certificates and their binding to the respective vehicle must be installed both in the factory and in the service. In addition, bindings between ECU devices must be distributed. This functionality is provided by this module.

Overview Certificate Management Module at Comfort-Mode:

The comfort Certificate management module now offers easier usability of the standard Certificate management functions.



For this purpose, you always need an SVT, which contains certified Emus. **[Note:** This is recognizable for every ECU on the CertEnabled = "true" attribute in the ecuStatusInfo. The Emus that are CertEnabled are shown in the SVT-View on the right.**]**

It is always treated the entire vehicle. That no filters (blacklist or whitelist) can be passed.

Use Cases:

Get Data:

Step 1: Create CSR Request File

Creates a file in JSON format with the individual Certificate requests (CertificateRequest) that the user either sends with E-Sys to the Backend (only possible it tester is connected to BMW-intranet) or use the B2B Interface.
How to get access to the B2B-Interface is described at the HOWTO as well.

Step 2: Upload Request File To Backend

Uploads the selected request file to the backend system (CBB) and stores the response file locally.

Write Data:

Variant 1: Get Data from Backend and write directly into Vehicle

Includes both steps of use case 'Get Data' but performs the write operation of the containers to the Emus as well.

Variant 2: Write Response File

Writes all containers of the selected response file to the Emus.

Check Data:

Starts the check routine of the Emus and fetches their results after a waiting time which is the maximum of the Emus to ensure all final check results can be fetched. The tree view will be enriched with the

check results.

Secure ECU Modes - Comfort

Introduction:

The module empowers the user on the one hand to switch all Emus, which supports Secure ECU Modes, between the three states (Engineering, Plant, Field).

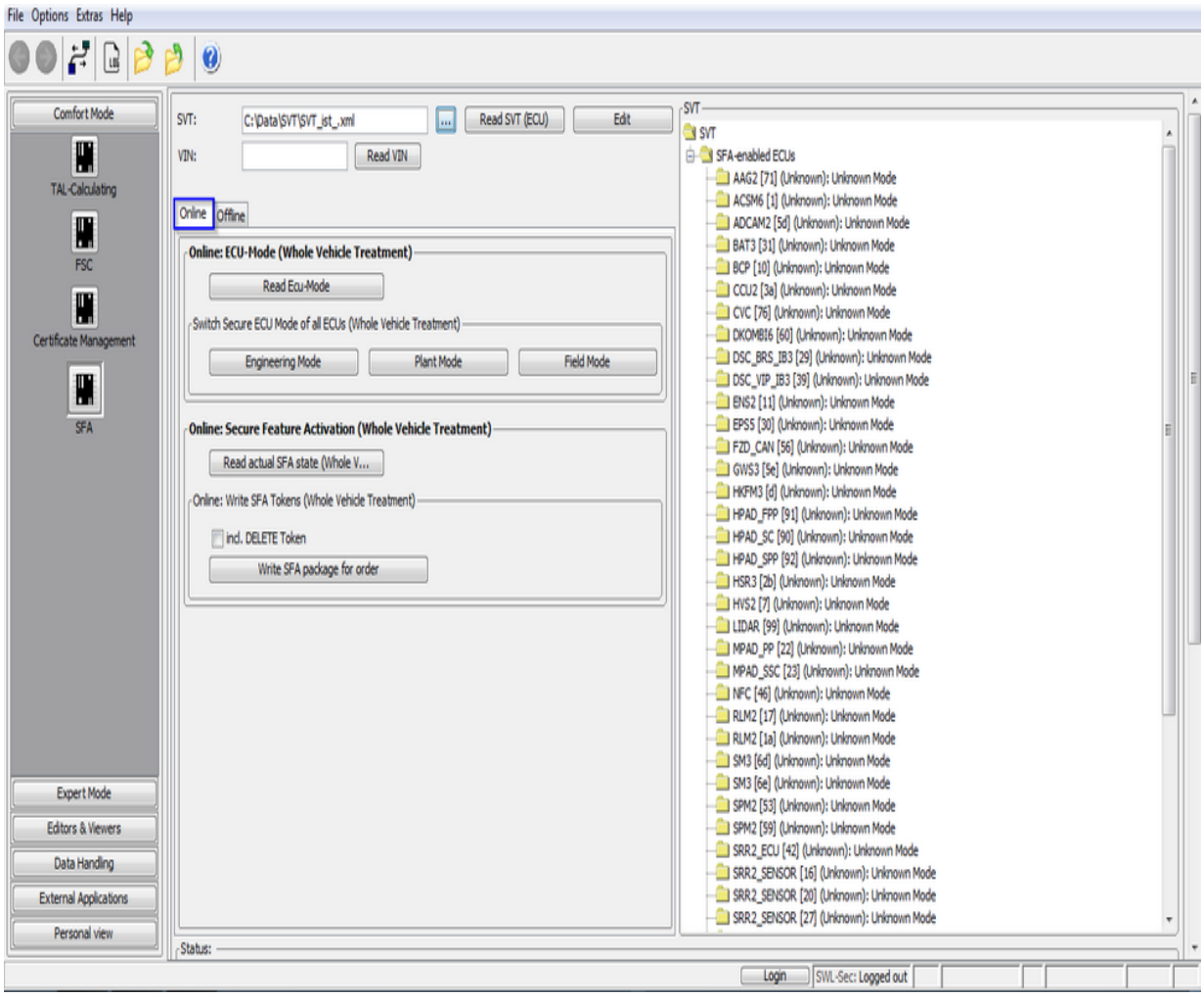
On the other hand the user can activate/deactivate and read the current secure feature status of all Emus, which are supporting these system function.

The Modes Plant and Engineering are only accessible with a valid secure token an the ECU has to be in den Field-Mode.

To switch back into the Field-Mode is no token necessary.

All Emus that are contained in the svt are visualized on the right side in the svt-tree-view.

Online-tap:

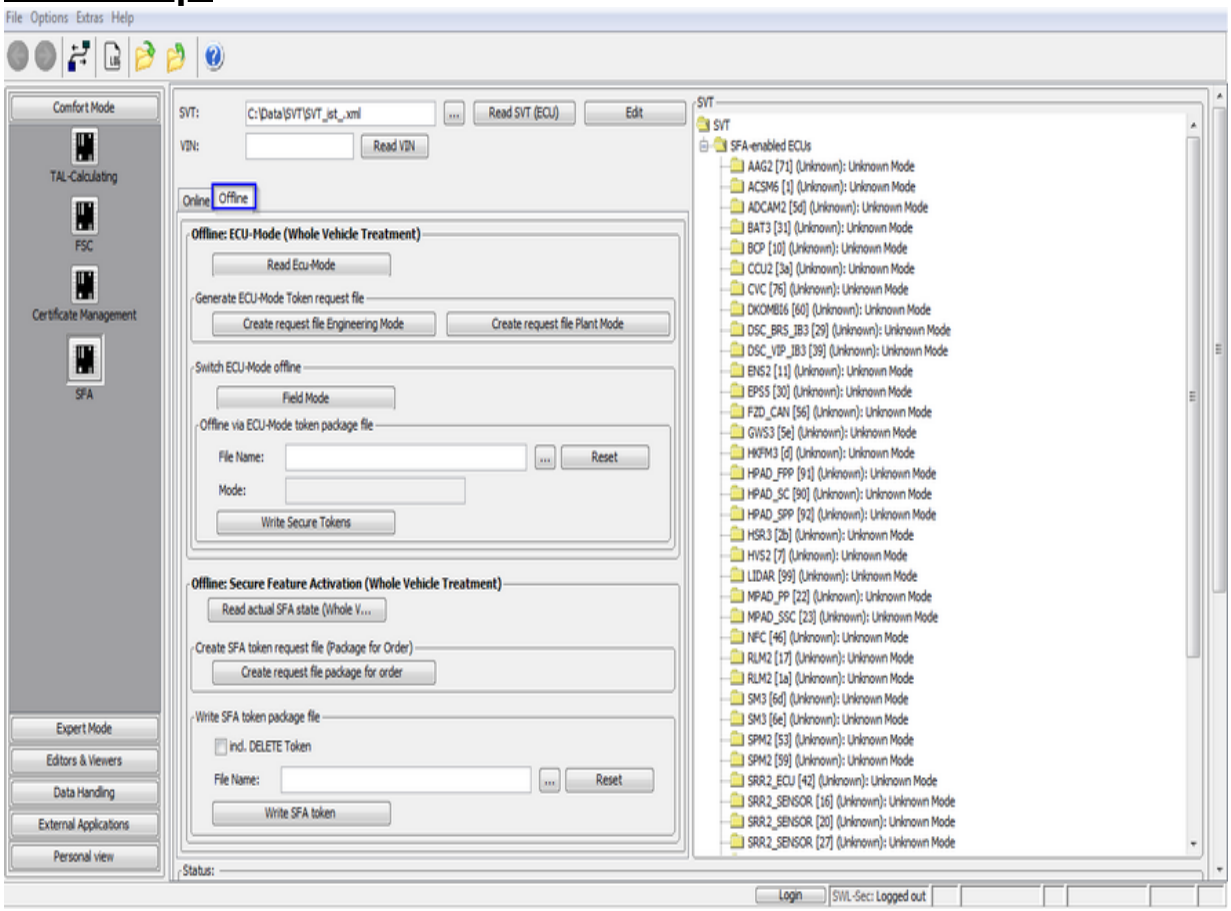


	Description:	Hints:
Read / load SVT	Generates the current SVT of the entire vehicle. In the svt-tree view are only the Emus of the SVT displayed, that support SFA	
Read / Edit VIN	Read the VIN out of the vehicle. Alternative has the user the option to edit the VIN manually.	
Read ECU-Mode	Read the current ECU-Mode of the entire vehicle. The actual ECU-Mode is	

	shown in the svt-view behind each base variant.	
Switch Mode: Plant Mode	Generates a Plant-Mode-Token for all Emus, that provides ECU-Mode, sending the request to the Backend and writing the response at the Emus. The Emus are finally in the Plant-Mode	BMW-intranet connection is necessary
Switch Mode: Engineering Mode	Generates an Engineering-Mode-Token for all Emus, that provides ECU-Mode, sending the request to the Backend and writing the response at the Emus. The Emus are finally in the Engineering-Mode	BMW-intranet connection is necessary
Switch Mode: Field Mode	All Emus are switching into the Field-Mode	
Read SFA status	Read the current SFA-Status of each ECU is shown in the svt-tree view.	
Write SFA package for order token-set	Generates at the backend a new secure token set. In case of new calculation is active the tokens for the already active features, that should be active based on the VIN, are going also to be generated. If the user wants only the features active, that are calculated to be active by the backup-System, "incl. DELETE Token" has be	BMW-intranet connection is necessary

	active. These means that it is possible that features which was active before are now deleted. To reactivate these features a new generated token is required.	

Offline-tap:



	Description:	Hints:
Read / load SVT	Generates the current SVT of the entire vehicle. In the svt-tree view are only the Emus of the SVT displayed, that support SFA	
Read / Edit VIN	Read the VIN out of the vehicle. Alternative has the user the option to edit the VIN manually.	

Read ECU-Mode	Read the current ECU-Mode of the entire vehicle. The actual ECU-Mode is shown in the svt-view behind each base variant.	
Switch Mode: Field Mode	All Emus are switching into the Field-Mode	
Generates request token set for Engineering Mode	Generates a request token set for the Engineering Mod for all Emus and saved local.	
Generates request token set for Plant Mode	Generates a request token set for the Plant Mod for all Emus and saved local.	
Write Secure Tokens:	Writes the chosen token package file for the ECU mode. The ECU-Mode of the token set is shown.	
Read SFA state	Read the current SFA-Status of each ECU is shown in the svt-tree view.	
Generate "package for Order" request	Generates a request file and save it locally. These file has to be sent manually to the backend where the secure token set ist generated.	
Write SFA token	Write the chosen SFA token set file. During the writing of the token set has the user the option do delete all additional features (feature is not included in the package for order token set).	

Color representation of SFA features in the SVT tree view:

In order to determine the status of all SFA features (activated, deactivated etc.) for each ECU shown in the SVT tree view, the button “Read actual SFA state” must be activated.

This triggers communication with every SFA-capable ECU and its features are read out.

The SVT tree structure is then updated in such a way that each ECU node can be opened up and all available features can be displayed. Each feature is highlighted in color in the SVT tree view.

Color markings mean:

Green - the Feature_ID is **activated**

Orange - the Feature_ID is **deactivated** or **expired**

Red - the Feature_ID is **incorrect**

Black - the Feature_ID is initially **not active**, i.e. no secure token available

Introduction:

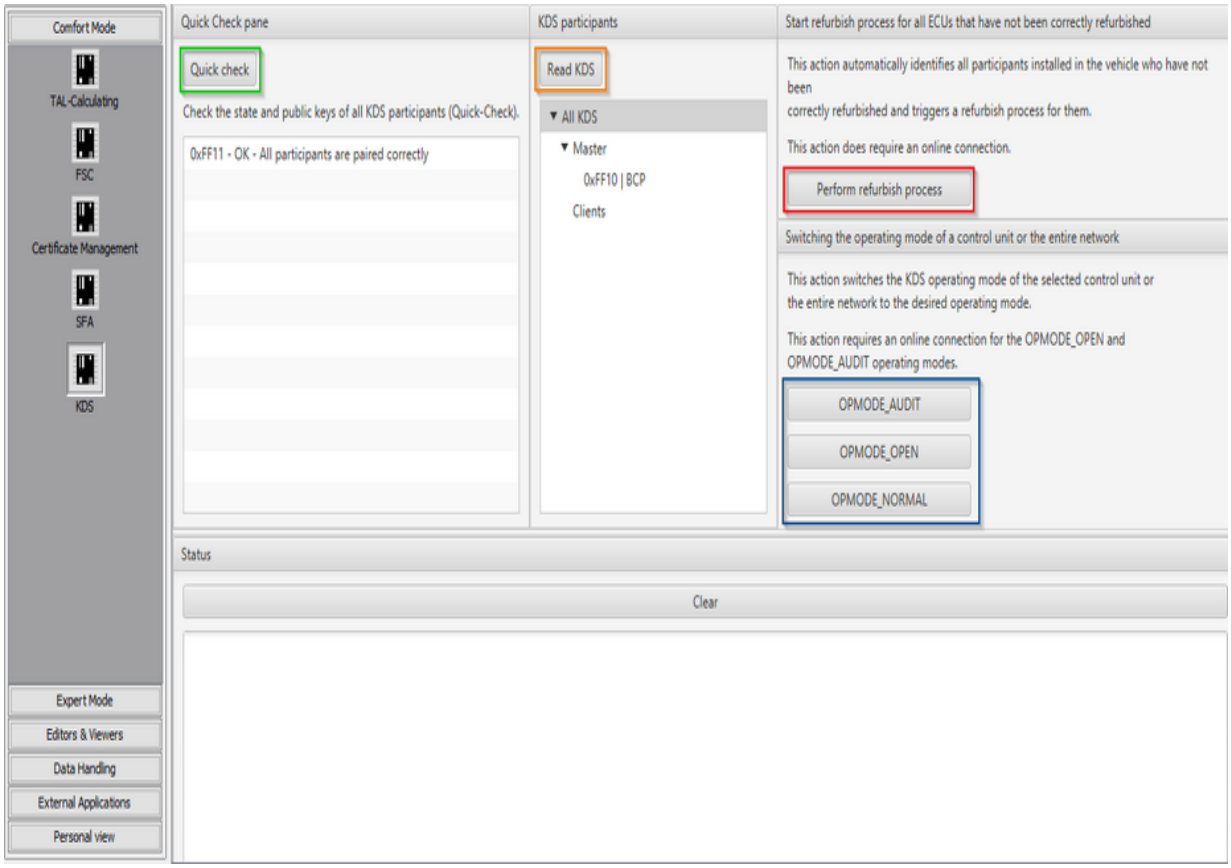
KDS (anti-theft protection 2.0) based on the secure token sets (see SFA / Secure ECU-Mode).
The functionalities provided in this module which include direct server access require to have several preconditions to be fulfilled:

- The system which runs E-Sys must be in a BMW network segment in which the backend systems (SFA-Backend) are available.
- There must be a valid E-Sys client Certificate imported into E-Sys.
- At least one URL of a valid backend system (SFA-Backend) must be configured in E-Sys.

A more detailed set of instructions can be found here: HOWTO (only available within BMW network).

Notifications:

These module supports only the processing of the entire vehicle.
For more KDS functions or the selection of specific Emus please switch to the expert-mode (KDS - Extended)



- **Quick check (marked green):**

Starts the KDS system check. The result is visualized in the GUI and has one of the following values:

KDS quick check results
MASTER_OK_CLIENT_OK
MASTER_OK_CLIENT_INVALID
MASTER_INVALID_CLIENT_OK
MASTER_INVALID_CLIENT_INVALID
ERROR_CLIENT_NOT_PAIRED

- **Read KDS (marked orange):**

Read the entire KDS system and visualize the system. The system is separated into the KDS-master and the KDS-clients tap.

- **Perform refurbish process (marked red): [Only with BMW-Intranet connection possible]**

Identifies all participants in the KDS system, that have not been correctly paired and is pairing the entire KDS system.

- **Switch the operation mode (marked blue):**

Change the operation mode for the chose Emus. If no ECU is selected the entire KDS system is going to switch.
The switch into the modes AUDIT and OPEN requires a BMW-Intranet connection and the valid client-backend-Certificate (E-Sys options / authentication)

ANFLASH

The Anflash module offers the possibility to perform a complete flash cycle in a fully automated way.

The structure of the required configuration files is described in a separate manual. The required master-configuration file can be created or edited with the ANFLASH-Editor.

A prerequisite for the execution of the Anflash process is a valid vehicle connection. In addition, a check is made before the Anflash process to ensure that all the required configuration files exist.
After a successful start, all ECU configs defined in the master config are processed sequentially in the order of their position numbers.
During each execution, the current process step is displayed.

The Anflash process can be aborted during execution via the Stop button.

At the end of the execution, both the overall status and the status of individual ECU configurations are displayed.

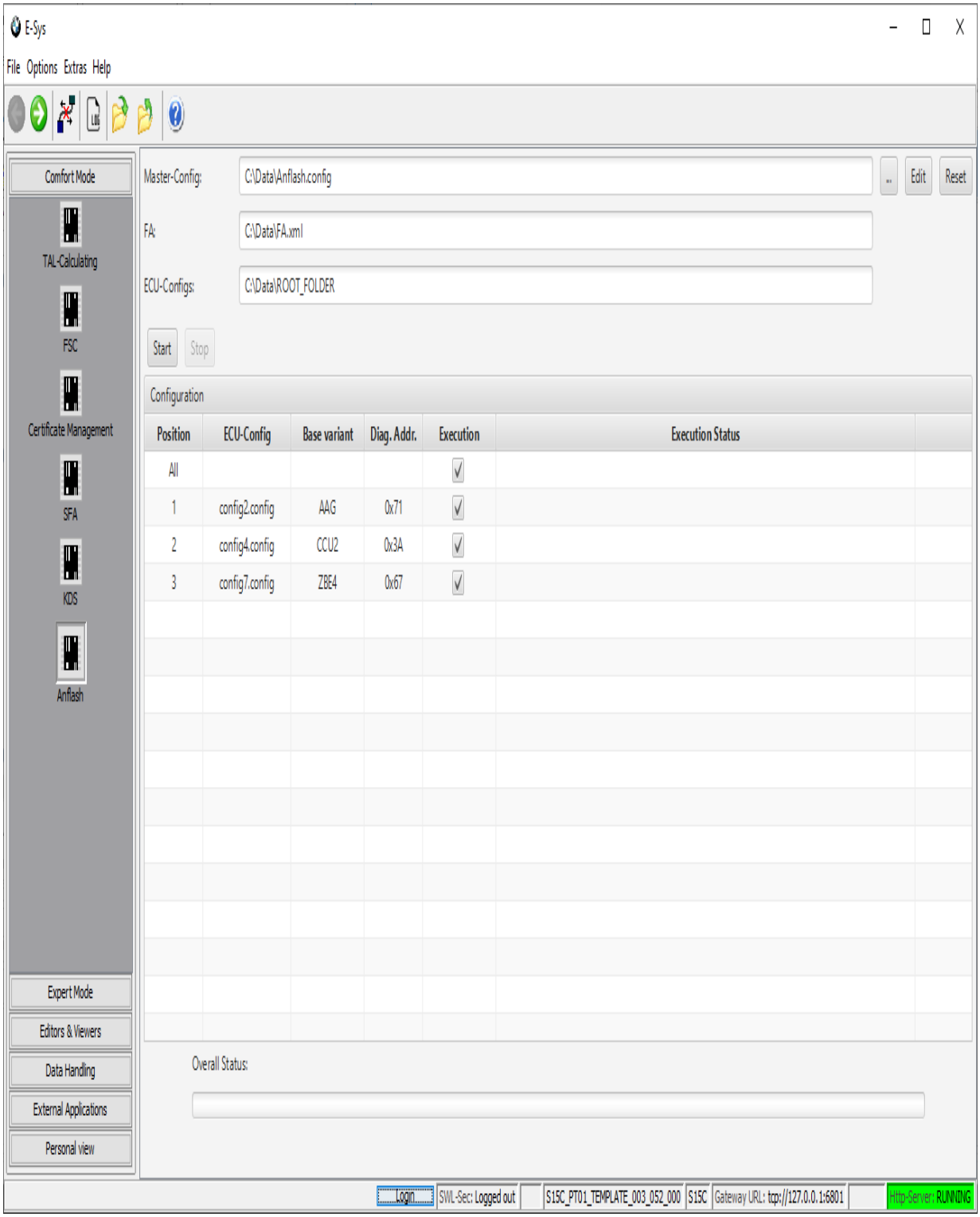
In addition, report files are generated.
This is done on two levels, firstly for each executed ECU-Config and secondly for the entire Anflash process.
The report files are stored in the E-Sys Data directory below the "Anflash-Reports" directory.

Note:

Before starting the Anflash process, a BAT/HAF verification is carried out.

It will be checked whether a BAT/HAF control unit is installed. If this is the case, the status of switch S1 is determined.

When the switch is open, a message appears informing the user about possible risks for the Anflash process.

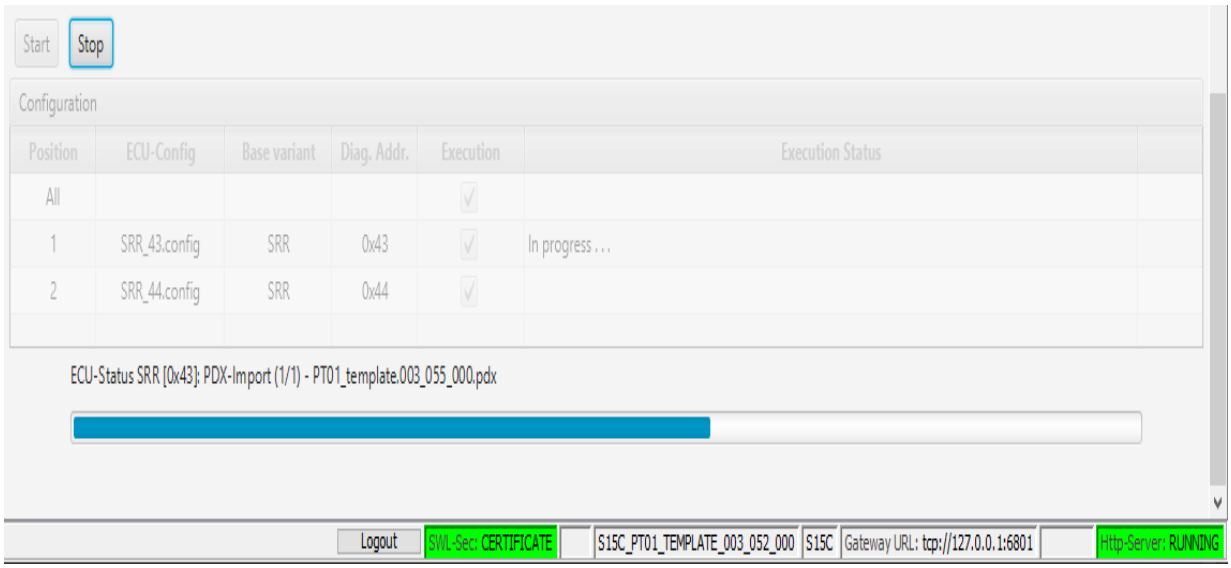


GUI Elements

Master-Config	
Textfield	Absolute path to the Master-Config

Load	Opens a file dialogue to load an existing Master-Config from a file. The ECU-Configs from the Master-Config will then be visible in the Configuration-View, the file path is displayed in the text field. An already loaded/opened file will be discarded.
Edit	Edit in the corresponding editor. This button is only active if there is an file is open and no Anflash process has been started.
Reset	Unload the active filter together with all corresponding symbols in the Table-View. This button is only active if there is an file is open and no Anflash process has been started.
FA	
Textfield	Absolute path to the vehicle order (FA). An absolute path is always displayed, even if it is specified as a relative path in the configuration file.
ECU-Configs	
Textfield	Absolute path to the ROOT directory where all ECU-configuration files are located. An absolute path is always displayed, even if it is specified as a relative path in the configuration file.
Execution	
Start	Start of Anflash process.
Stop	Abort the Anflash process. PDX-Import: The current import process must be completed. TAL-Execution: After the current processed TALline the processing is aborted..
Configuration table	
Columns	List the existing ECU-Configs for the loaded Master-configuration file.

Rows	List the Data of each ECU-Config. The Execution on ECU-Config can be activated or deactivated by checkbox.
Statusanzeige	
Status label	Visualization of the currently step of Anflash process. Example: ECU-STATUS CCU2 [0x3a]: PDX-Import (1/3) - PT01_Container_1.pdx At the end of the execution, the overall status is displayed.
Progress bar	Visualization of progress of an Anflash process. In the case of a TAL Execution, the remaining duration is also displayed.



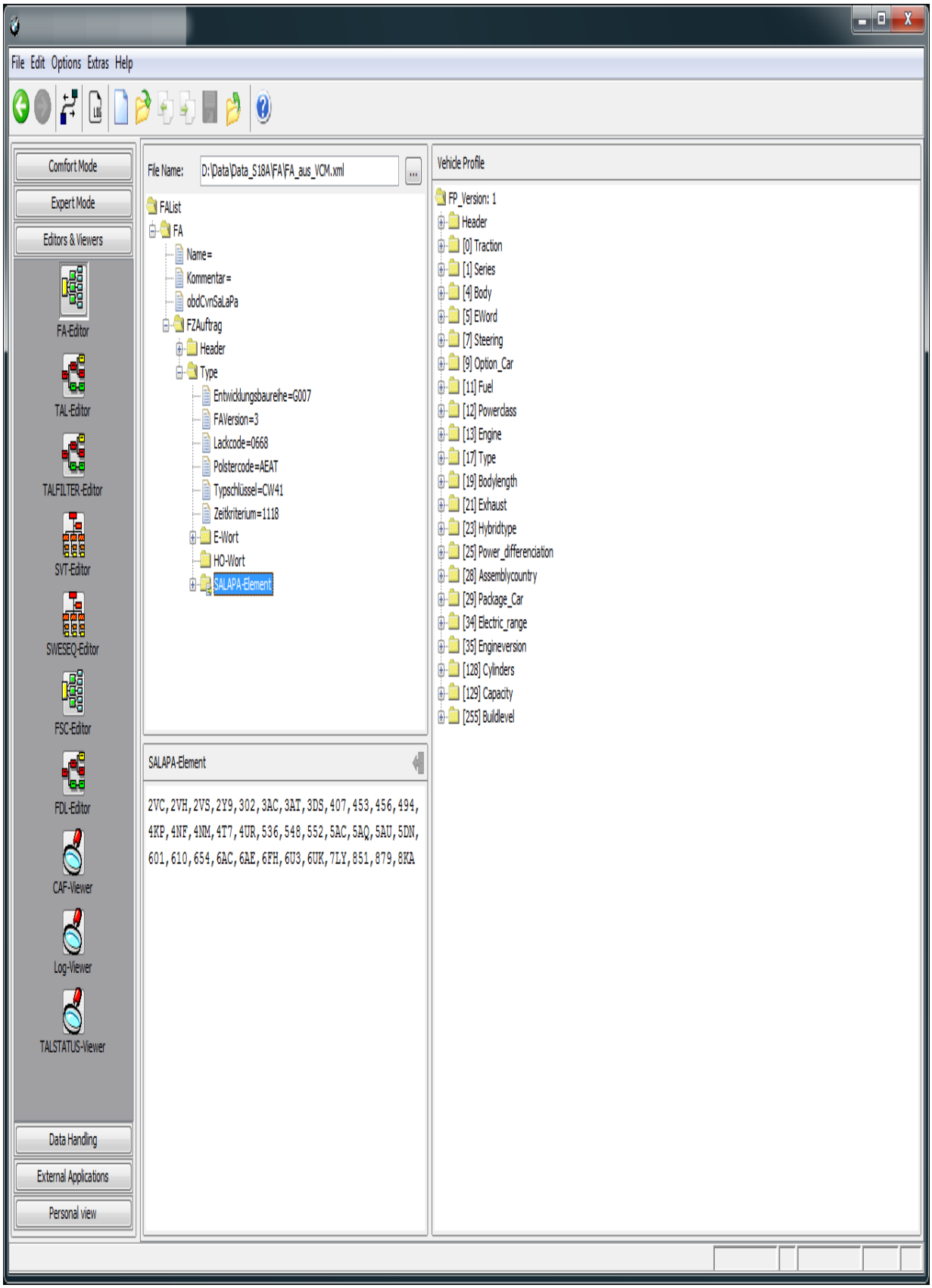
Editors and Viewers

FA/FP-Editor
TAL-Editor
TAL-Filer-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
ANFLASH-Editor
CAF-Viewer
LOG-Viewer
TALSTATUS-Viewer

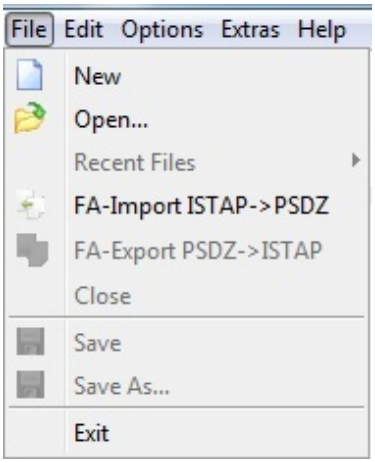
FA/FP-Editor

The module FA/FP-Editor makes for display and edition of FAs (vehicle jobs) and the correspondent FPs (vehicle orders). The FA designates a discrete state of installation of SWEs and HWEs for a vehicle. You can derive the vehicle profiles from the vehicle order.

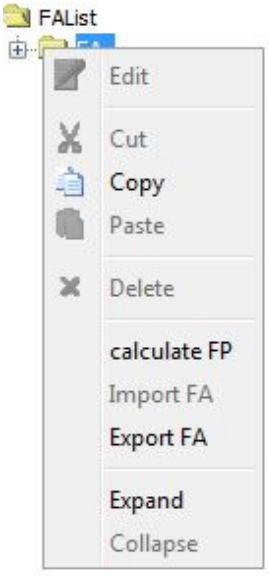
An ECU or a vehicle can contain/save exactly one vehicle order. During the communication with an ECU or with an vehicle over PSdZ always only a single vehicle order is exchanged. But in the local file system a vehicle order always is saved within a list consisting of any number of vehicle jobs marked by an unique ID.


File Edit Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
File Name:
D:\Data\Data_S18A\FA\FA_aus_VCM.xml
FAList
FA
Name=
Kommentar=
obdCvnSaLaPa
FZAuftrag
Header
Type
Entwicklungsbaureihe=G007
FAVersion=3
Lackcode=0668
Polstercode=AEAT
Typschlüssel=CW41
Zeitkriterium=1118
E-Wort
HO-Wort
SALAPA-Element
SALAPA-Element
2VC,2VH,2VS,2Y9,302,3AC,3AT,3DS,407,453,456,494,
4KP,4NF,4NM,4T7,4UR,536,548,552,5AC,5AQ,5AU,5DN,
601,610,654,6AC,6AE,6FH,6U3,6UK,7LY,851,879,8KA
Vehicle Profile
FP_Version: 1
Header
[0] Traction
[1] Series
[4] Body
[5] EWord
[7] Steering
[9] Option_Car
[11] Fuel
[12] Powerclass
[13] Engine
[17] Type
[19] Bodylength
[21] Exhaust
[23] Hybridtype
[25] Power_differenciation
[28] Assemblycountry
[29] Package_Car
[34] Electric_range
[35] Engineversion
[128] Cylinders
[129] Capacity
[255] Buildlevel

File Menu



Context Menu



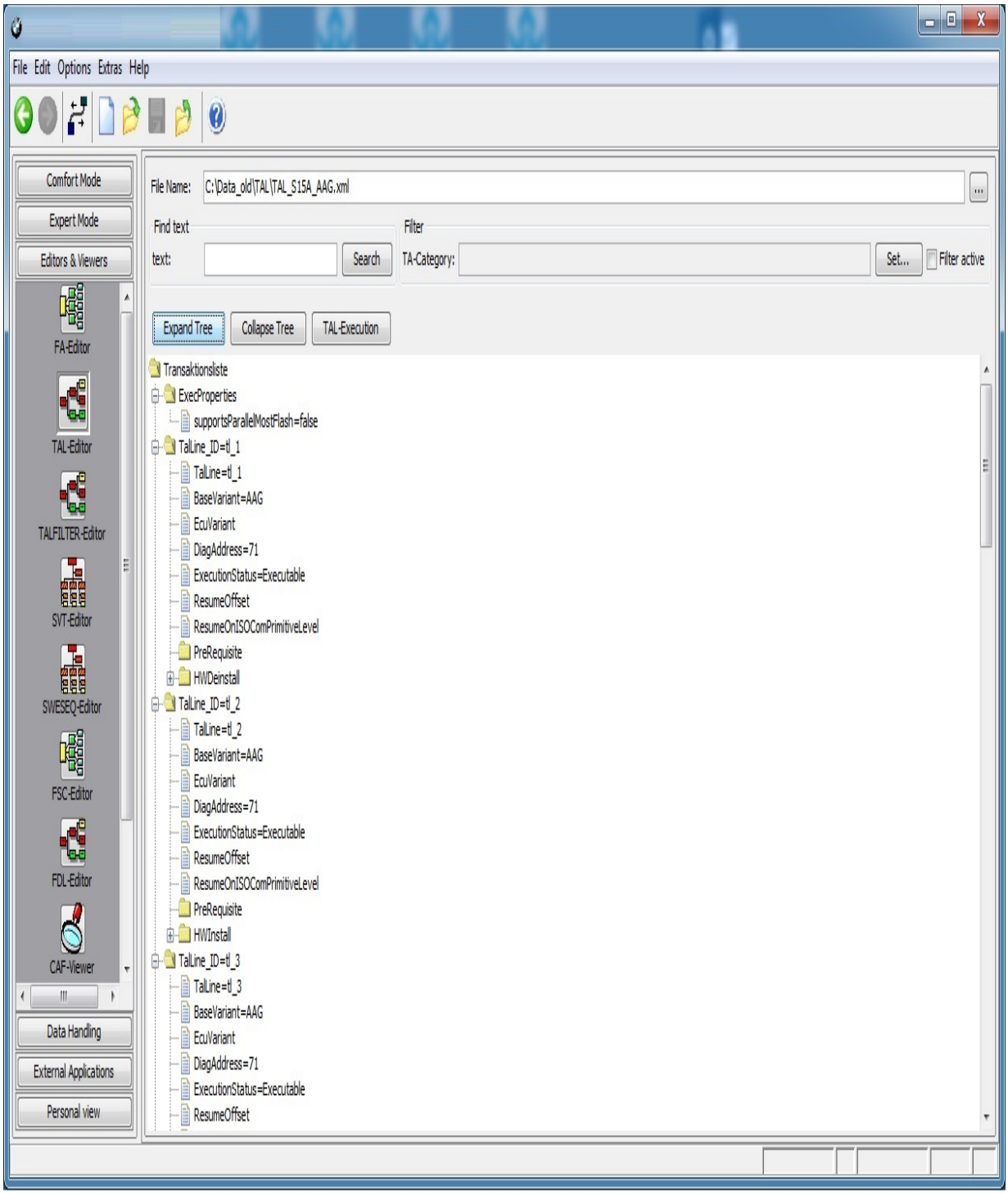
Specifics

- The editors subdivides in two areas. On the left side is the FA list's view. If you select in the left window a FA by the context menu you retrieve the FP. The calculated FP is displayed on the right hand side of the window.
- To display a single FA you have to create an empty FA list and import by context menu the desired FA.
- A single FA can be imported also by context menu.

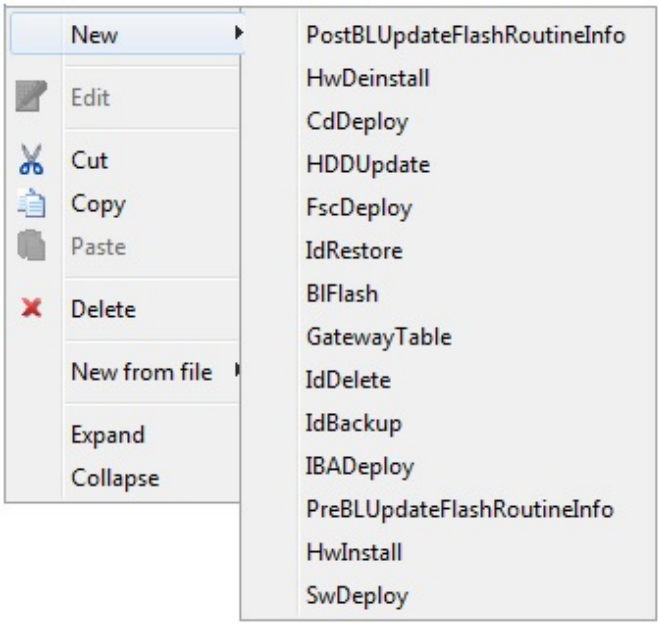
- A single FA can be exported also in ISTAP-Format using the File menu.
- A single ISTAP-Format FA can also be imported using the File menu.

TAL-Editor

With the TAL-Editor you can display and edit transaction lists (TALs). A TAL is a model year plan to transform an ECU, a system of Emus or the entire vehicle from the state corresponding to the actual system installation table to the state corresponding to the reference system installation table. The PSdZ generates the TAL from a pair of SVT-Actual and SVT-Target.
The TAL contains a list of actions sorted by ECU name, e.g. dismount HWE, program SWE, save individual data, etc. After processing this list the vehicle is in the state SVT-Target.



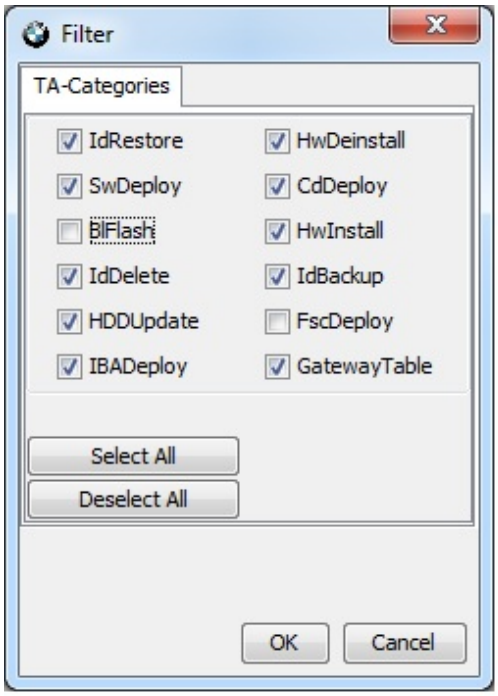
Context-Menu



GUI-Elements

Expand Tree	Open the whole tree up to the first level
Collapse Tree	Close the whole tree including sub trees.
TAL-Execution	Transfer TAL to module TAL-Processing and open it
Find Text	Allows searching for text in the TAL tree view. By pressing the "Search" button, all detected text elements will be marked one after another in the tree view.
Filter	By using checkbox "filter active", you en/disable the filter. By the button "Set..." you switch to a dialog for choosing the particular elements to be affected by the filter.

Filter-Settings


Filter
TA-Categories
IdRestore
HwDeinstall
SwDeploy
CdDeploy
BlFlash
HwInstall
IdDelete
IdBackup
HDDUpdate
FscDeploy
IBADeploy
GatewayTable
Select All
Deselect All
OK
Cancel

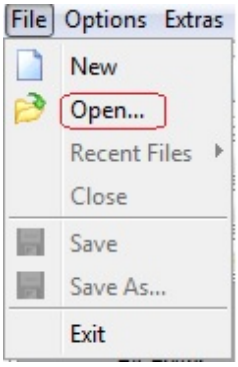
TALFILTER-Editor

Module TALFILTER-Editor offers the possibility to display and modify TAL-Filters.

Creation of new TAL-Filter

There are two ways of creating a new filter file:

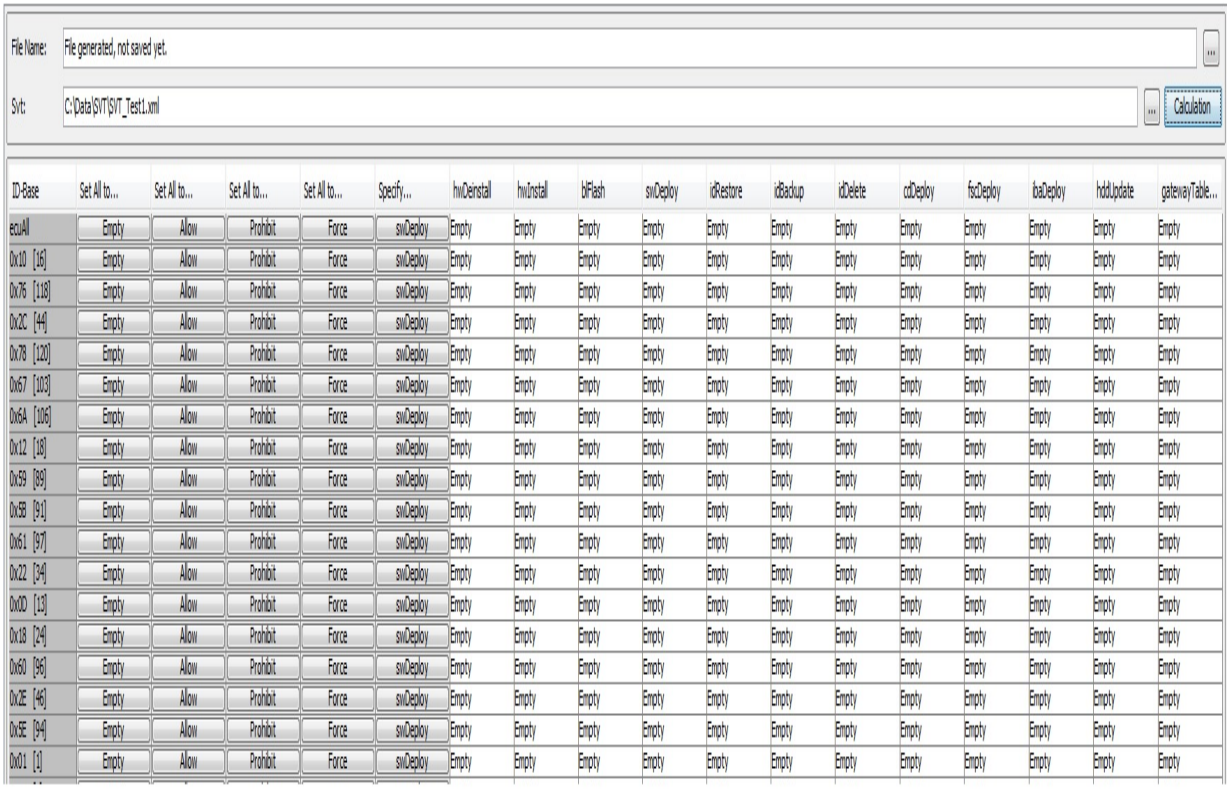
1. with File Menu



This will create an empty TAL-Filter with a content like the one shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<talfilter>
    <allEcu/>
</talfilter>
```

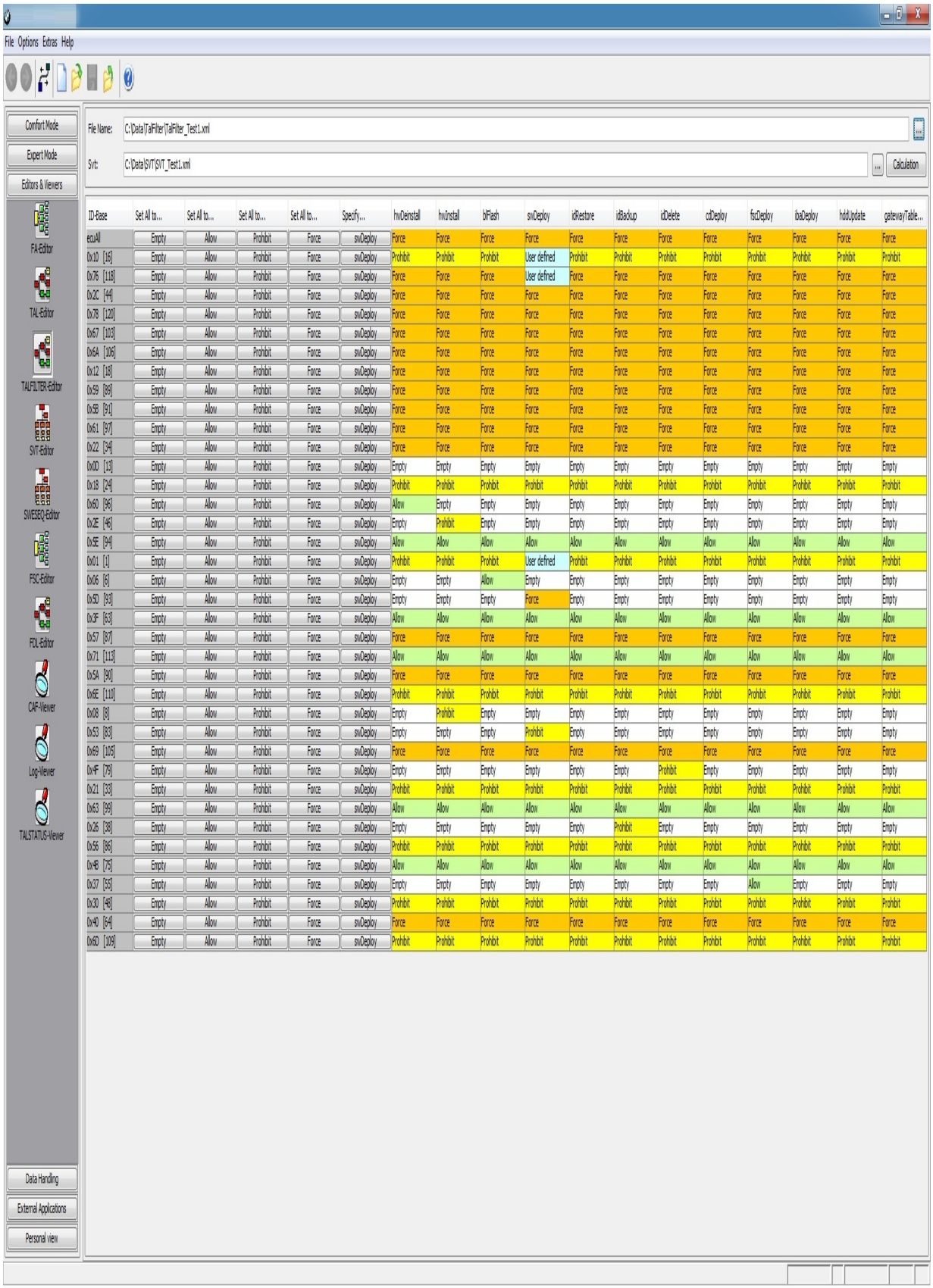
2. with SVT-File



After selecting an existing SVT file, a new TAL-Filter will be created when pressing button 'Calculation'.
This TAL-Filter will contain empty entries for all Emus defined in the SVT file.

Editing and Displaying TAL-Filter settings

As soon as a new TAL-Filter has been created or an existing filter has been loaded, its filter settings will be displayed in a table structure:


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
File Name: C:\Data\TalFilter\TalFilter_Test1.xml
Svt: C:\Data\SVT\SVT_Test1.xml
Calculation
ID-Base | Set All to... | Set All to... | Set All to... | Set All to... | Specify... | hwDeinstall | hwInstall | blFlash | swDeploy | idRestore | idBackup | idDelete | cdDeploy | fscDeploy | ibaDeploy | hddUpdate | gatewayTable...
ecuAll | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x10 [16] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | User defined | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit
0x76 [118] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | User defined | Force | Force | Force | Force | Force | Force | Force | Force
0x2C [44] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x78 [120] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x67 [103] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x6A [106] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x12 [18] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x59 [89] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x5B [91] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x61 [97] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x22 [34] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x0D [13] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty
0x18 [24] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit
0x60 [96] | Empty | Allow | Prohibit | Force | swDeploy | Allow | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty
0x2E [46] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Prohibit | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty
0x5E [94] | Empty | Allow | Prohibit | Force | swDeploy | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow
0x01 [1] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | User defined | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit
0x06 [6] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Empty | Allow | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty
0x5D [93] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Empty | Empty | Force | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty
0x3F [63] | Empty | Allow | Prohibit | Force | swDeploy | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow
0x57 [87] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x71 [113] | Empty | Allow | Prohibit | Force | swDeploy | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow
0x5A [90] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x6E [110] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit
0x08 [8] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Prohibit | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty
0x53 [83] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Empty | Empty | Prohibit | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty
0x69 [105] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x4F [79] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Empty | Empty | Empty | Empty | Empty | Prohibit | Empty | Empty | Empty | Empty | Empty
0x21 [33] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit
0x63 [99] | Empty | Allow | Prohibit | Force | swDeploy | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow
0x26 [38] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Empty | Empty | Empty | Empty | Prohibit | Empty | Empty | Empty | Empty | Empty | Empty
0x56 [86] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit
0x4B [75] | Empty | Allow | Prohibit | Force | swDeploy | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow | Allow
0x37 [55] | Empty | Allow | Prohibit | Force | swDeploy | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Empty | Allow | Empty | Empty | Empty
0x30 [48] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit
0x40 [64] | Empty | Allow | Prohibit | Force | swDeploy | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force | Force
0x6D [109] | Empty | Allow | Prohibit | Force | swDeploy | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit | Prohibit

For modifying the filter settings the user is offered several options:

1. Set All to...

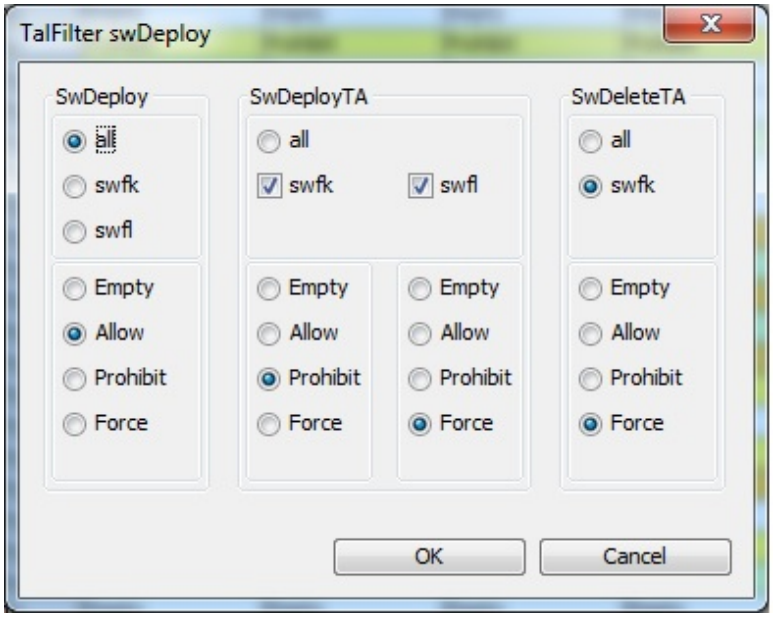
By using one of the buttons 'Empty', 'Allow', 'Prohibit' or 'Force' all Transaction Categories of an ECU can be set to the same value. Additionally these four buttons located in the first row ('ecuAll'), will set all Transaction Categories of all Emus to the specified value.

2. Select Action

Each Transaction category of every ECU can separately be set to one of the following action values:
- Empty
- Allow
- Prohibit
- Force

Note:
For Transaction category swDeploy it is possible to specify user defined settings for swDeployTA and swDeleteTA.
Therefore an additional action value ('user defined') will be available for that category.

3. swDeploy
As mentioned before, it is possible to define additional filter settings for category swDeployTA and swDeleteTA.
This can be done by opening an additional dialog via the swDeploy button:


TalFilter swDeploy
SwDeploy
all
swfk
swfl
Empty
Allow
Prohibit
Force
SwDeployTA
all
swfk
swfl
Empty
Allow
Prohibit
Force
Empty
Allow
Prohibit
Force
SwDeleteTA
all
swfk
Empty
Allow
Prohibit
Force
OK
Cancel

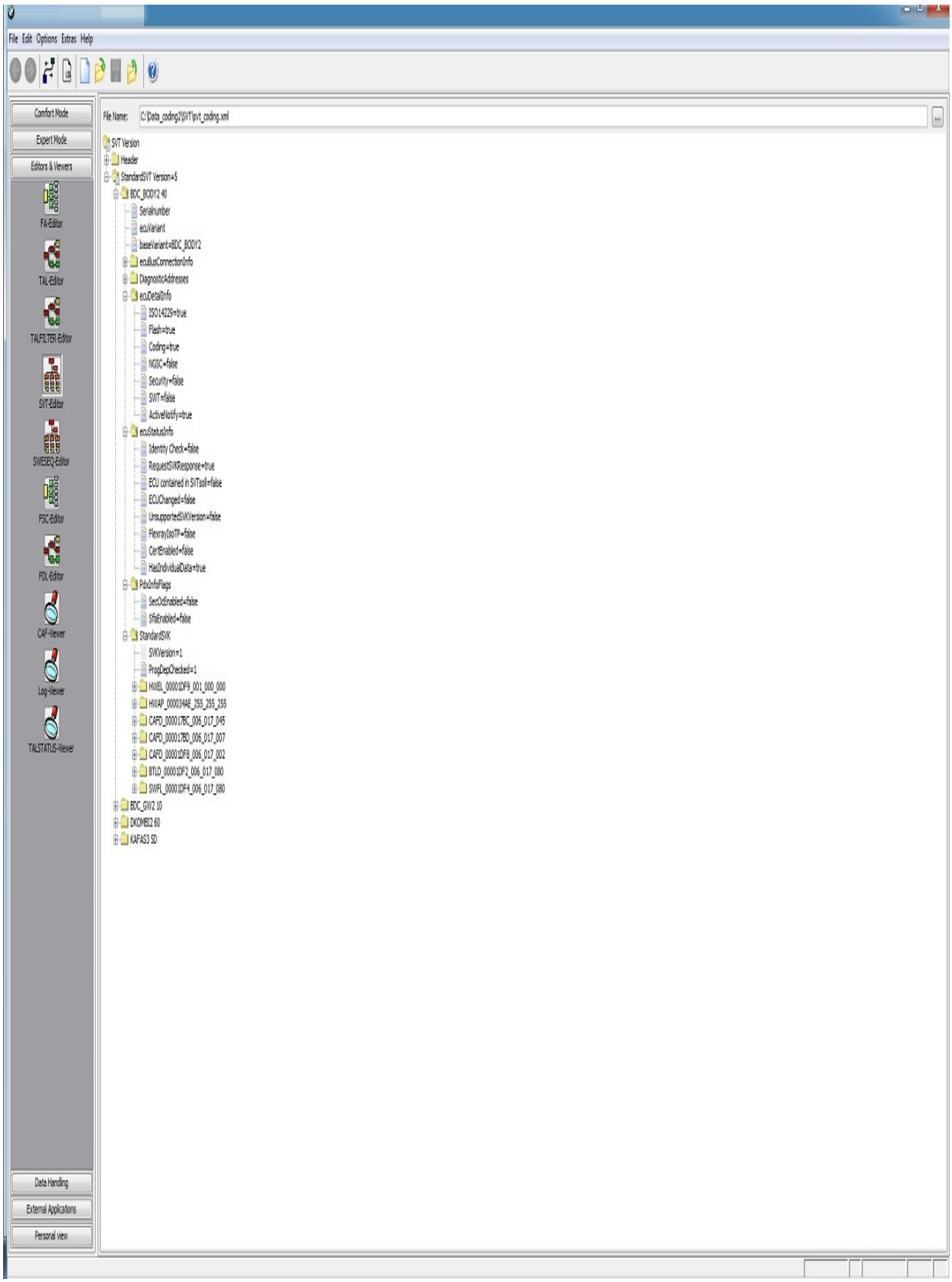
-

SVT-Editor

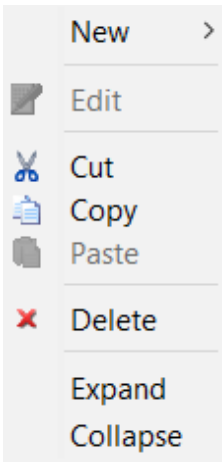
With SVT-Editor you can display and edit SVTs (system installation tables). An SVT reflects a discrete state of a ECU or vehicle.
You have to differentiate between an SVT-Actual which describes the actual state of the vehicle and an SVT-Target which describes the desired state of a vehicle. Trivially the actual state is given and the reference state is created by the user or can be imported from the KIS.

The PSdZ can generate a TAL based on SVT-Actual and SVT-Target. This TAL contains the action list to transform the vehicle from actual to reference state.

Hint: Whereas an SVT contains the state for a vehicle in the SVK (system installation identifier) only the state of a single ECU is recorded. Therefore an SVT consists at least of one SVK. Since an SVK can be displayed as an SVT with a single SVK no separate SVK-Editor exists.

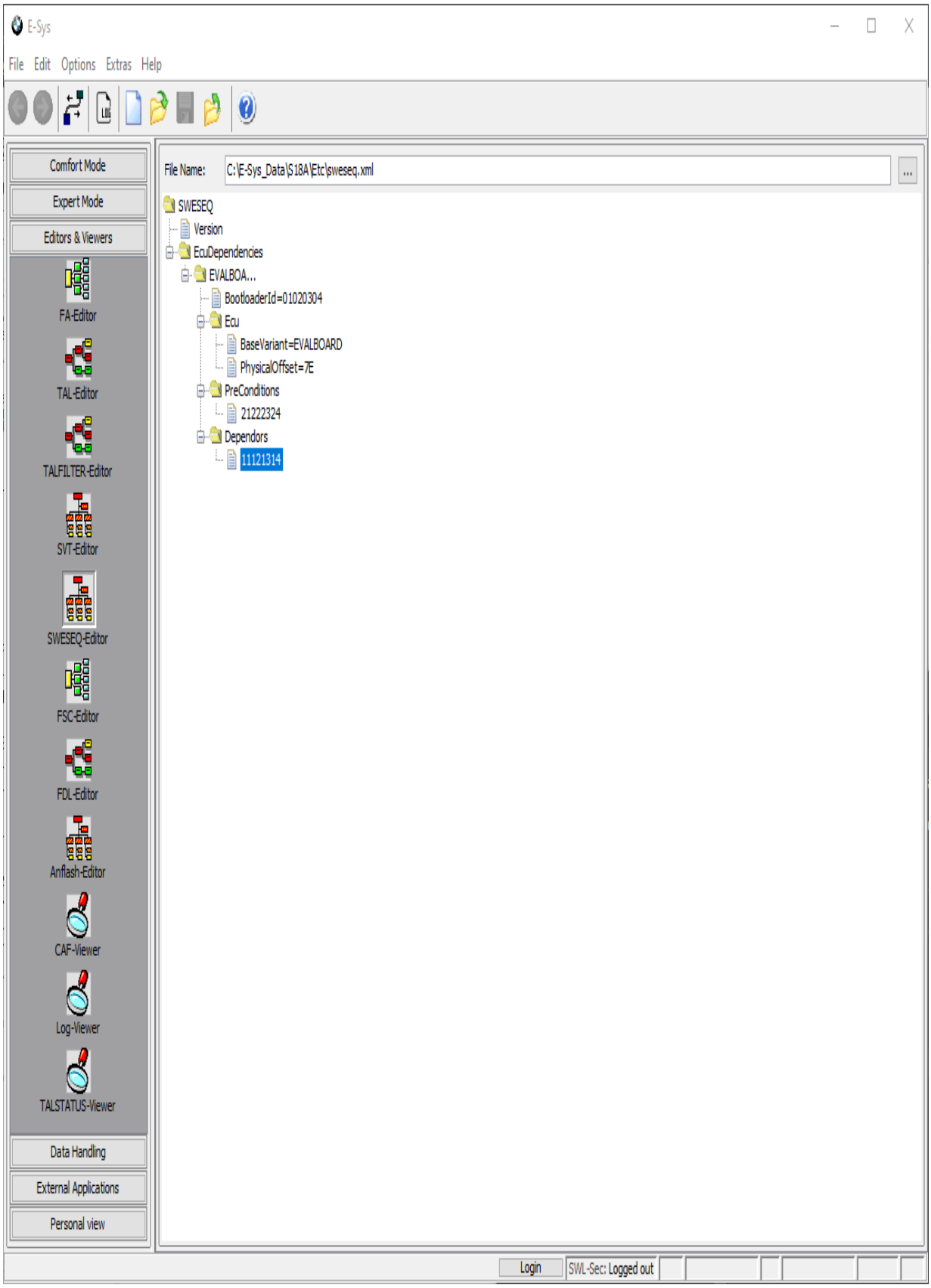


Context-Menu

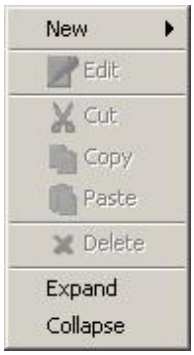

New
Edit
Cut
Copy
Paste
Delete
Expand
Collapse

SWESEQ-Editor

Module SWESEQ Editor offers displaying and modification of a SWE Sequence.


E-Sys
File Edit Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
Anflash-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
File Name:
C:\E-Sys_Data\S18A\Etc\sweseq.xml
SWESEQ
Version
EcuDependencies
EVALBOA...
BootloaderId=01020304
Ecu
BaseVariant=EVALBOARD
PhysicalOffset=7E
PreConditions
21222324
Dependors
11121314
Login
SWL-Sec: Logged out

Context Menu



The menu options offers insertion of a new element, editing, cut, copy, paste, deletion, expanding or collapsing of an element within the SWE sequence.

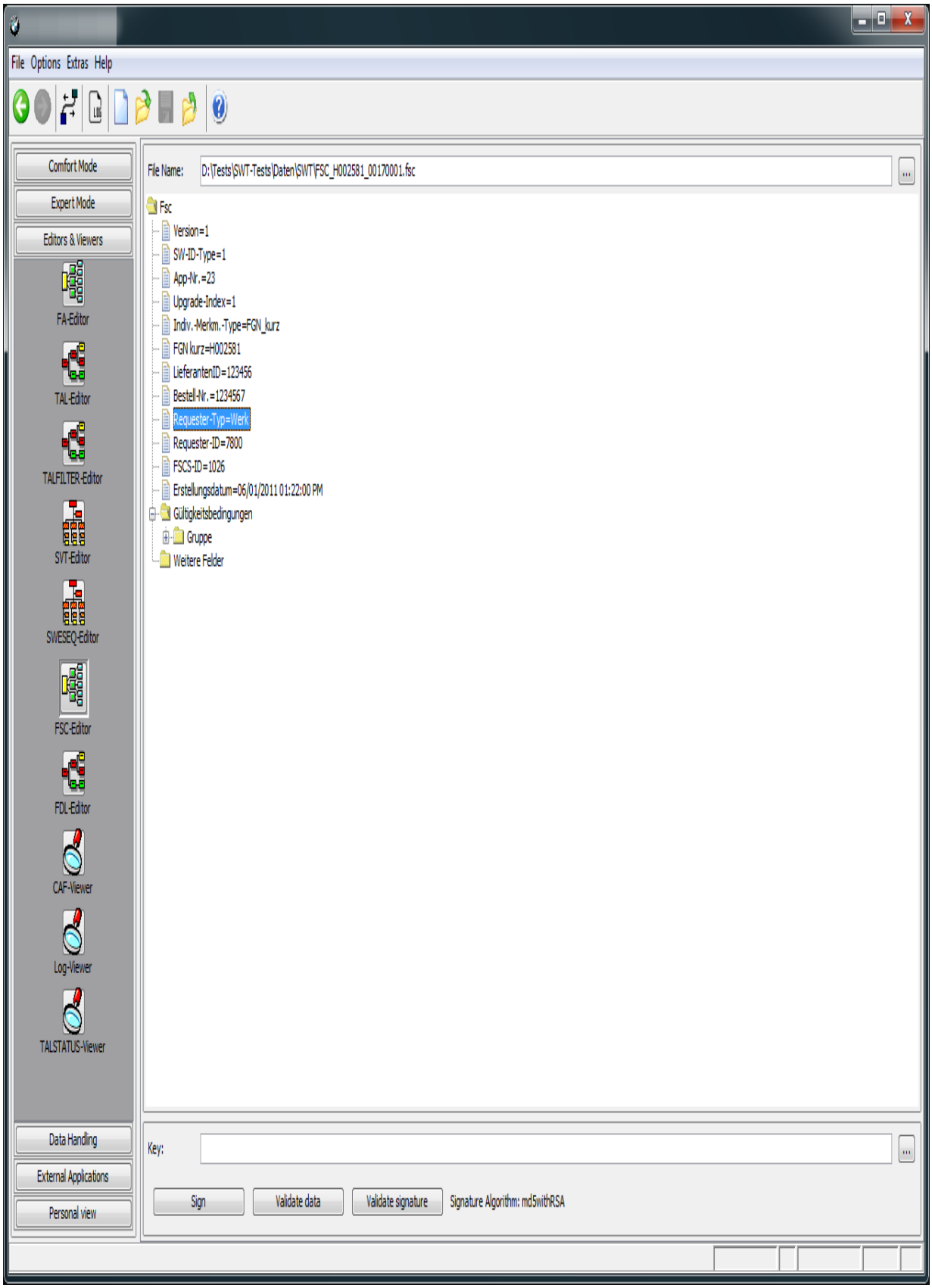
GUI Elements

File Name	Currently opened file.
"..."	Button for opening a SWESEQ file.

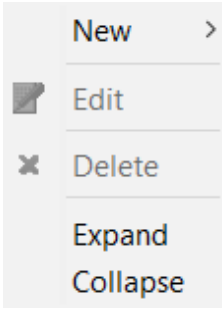
FSC-Editor

Module FSC Editor serves for display and modification of FSC's.

Corresponding to the topic FSC there are other masks in E-Sys: Module **Freischaltcode** (FSC) serves for inquiring the FSC State, for writing and (de-)activating of FSC. In expert mode the **FSC-Extended-Module** is available in addition to that.


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
File Name:
D:\Tests\SWT-Tests\Daten\SWT\FSC_H002581_00170001.fsc
Fsc
Version=1
SW-ID-Type=1
App-Nr.=23
Upgrade-Index=1
Indiv.-Merkm.-Type=FGN_kurz
FGN kurz=H002581
LieferantenID=123456
Bestell-Nr.=1234567
Requester-Typ=Werk
Requester-ID=7800
FSCS-ID=1026
Erstellungsdatum=06/01/2011 01:22:00 PM
Gültigkeitsbedingungen
Gruppe
Weitere Felder
Key:
Sign
Validate data
Validate signature
Signature Algorithm: md5withRSA

Context Menu



The menu options serves for selection of a line to be edited.
Or for insert of a new line or deletion of a line within the FSC.

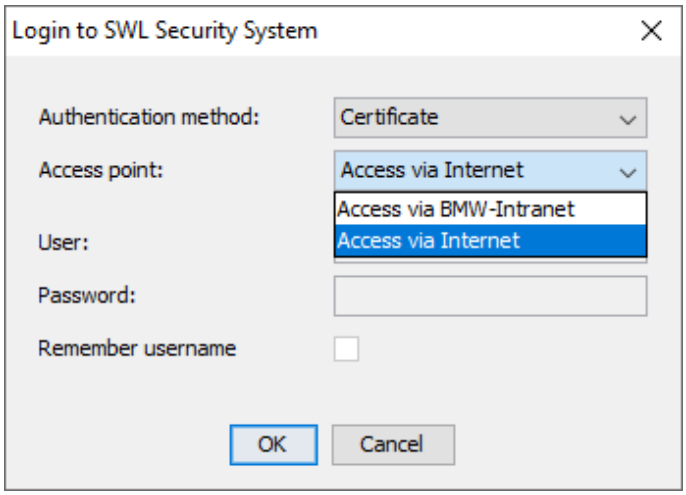
GUI Elements

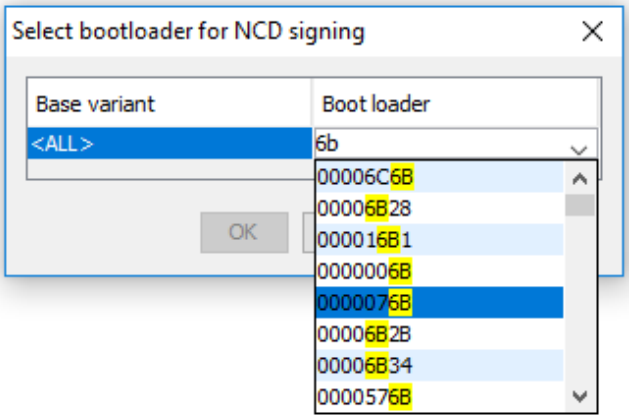
FSC-File	Selection of the FSC to be displayed and (eventually) modified.
"..."	Button for specifying the path to the FSC file.

FDL-Editor

The FDL-Editor displays the netto-coding-file (NCD) in a visual format the FDLs (function data lists).
The NCDs can be edited within these module.

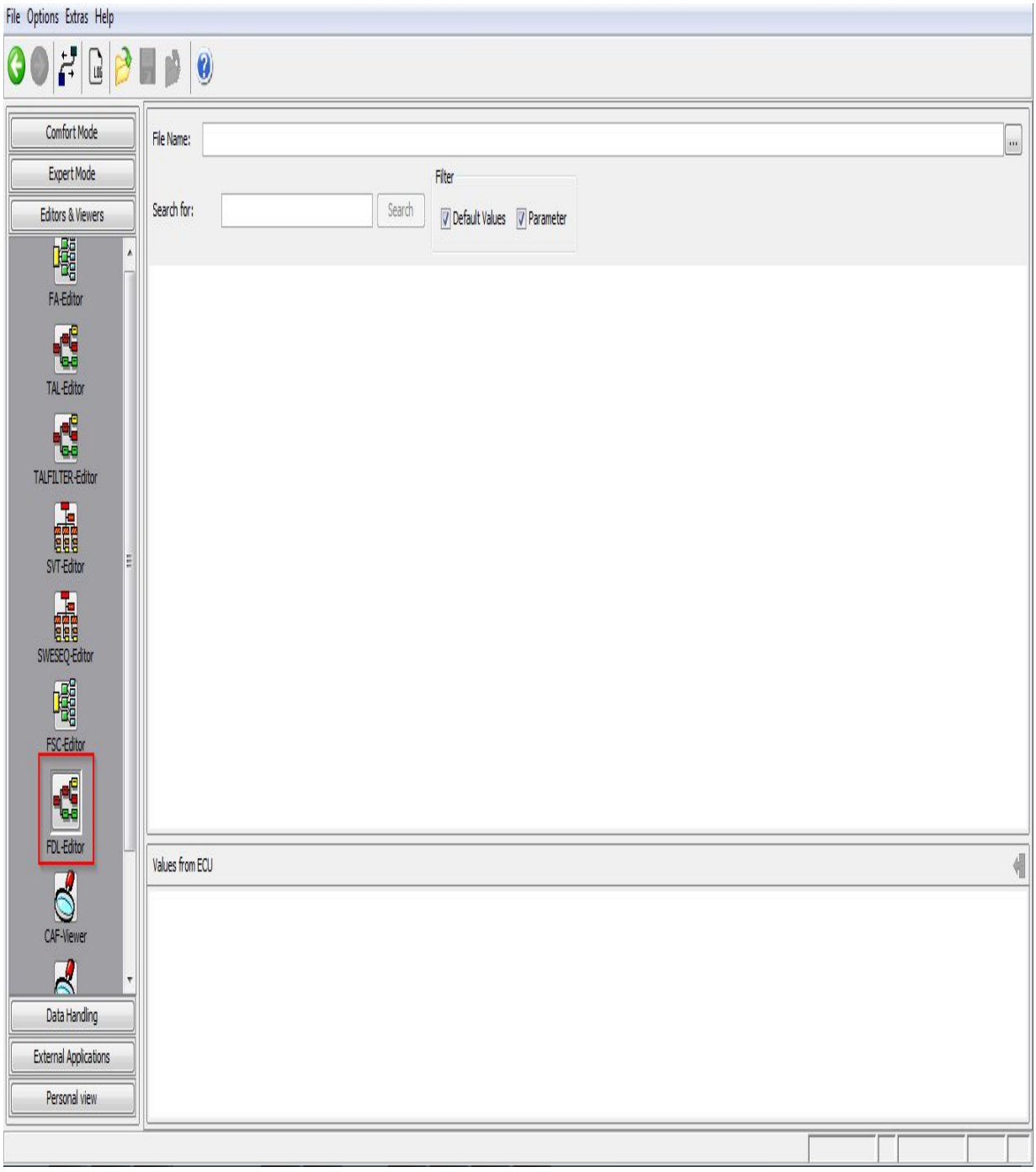
Before the file is saved, it is automatically signed. Here can be required the access data for the SWL-Security Backend (see secure coding) and the BTLD number.





Signing of NCDs for Coding 2 or Coding 3 is only possible if the code verification stamp (CPS) in the NCD file is valid.
A function data list is in principle a CAF (Coding Application File) which is aggregated with the data residing actually in the ECU.
A CAF exists prior to ECU coding as data record. But a FDL is generated by readout of data from a ECU and the input of the result in

the corresponding CAF.

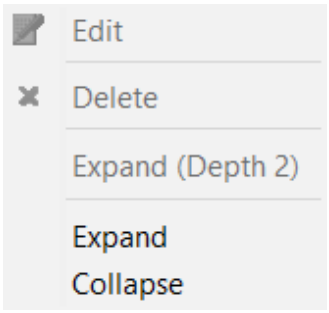


File-Menu

- Save as FWL...

The coding data is stored as a readable function value list (FWL).

Context-Menu



GUI-Elements

1. Search Function Name	searches the specified text in the whole tree. If there is a match the tree is expanded to the place of finding. You can continue the search with button F3.
Filter	filters corresponding sub trees

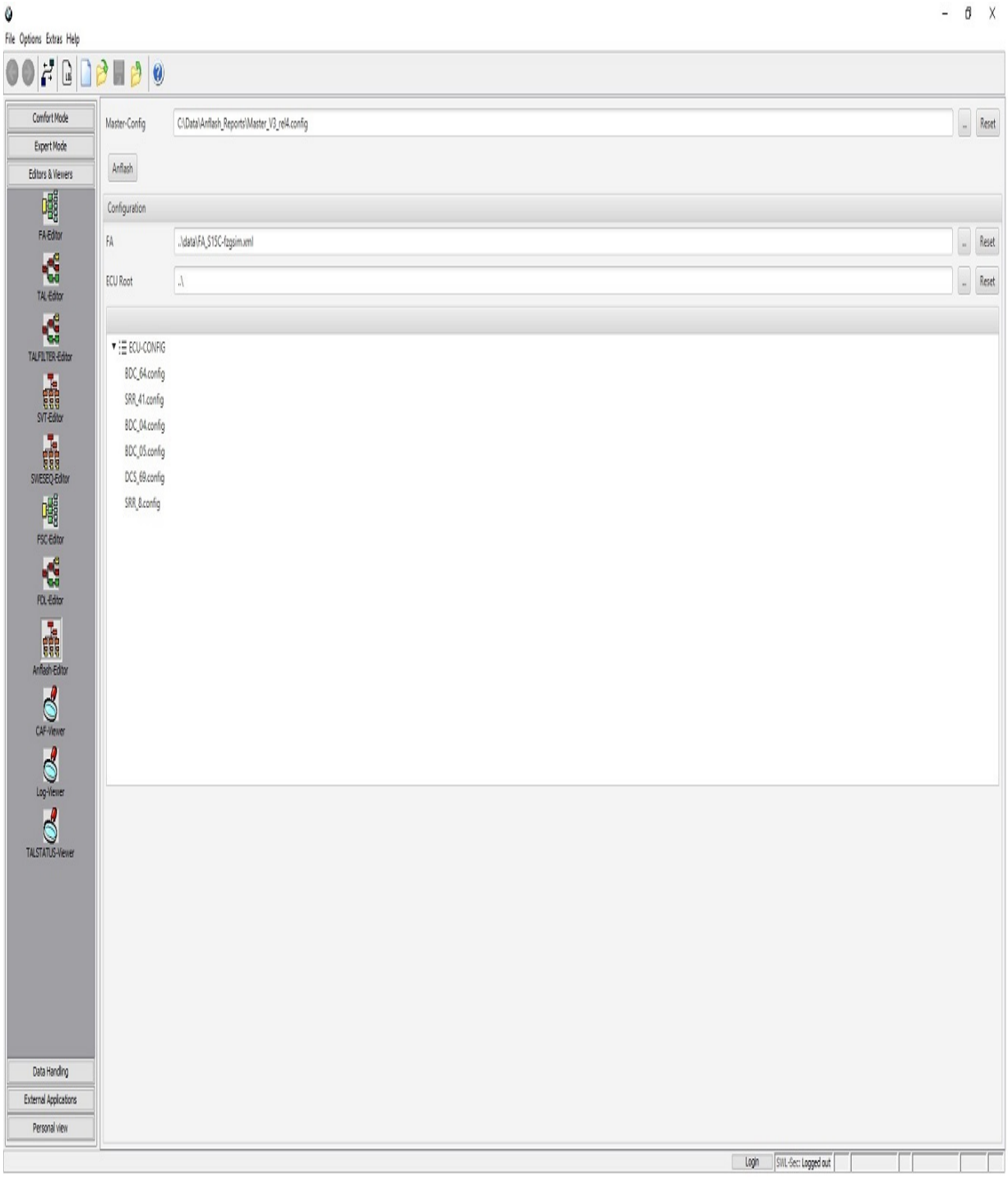
ANFLASH-Editor

Module Anflash-Editor offers the possibility to display and modify Master configuration files.

The structure of the required configuration files is described in chapter 2.1 of the operation manual.

In the operation manual, the usage of the Anflash-Editor will be explained in detail.

By pressing the 'Anflash' button, the currently loaded master-configuration file can be opened and executed in the ANFLASH module.



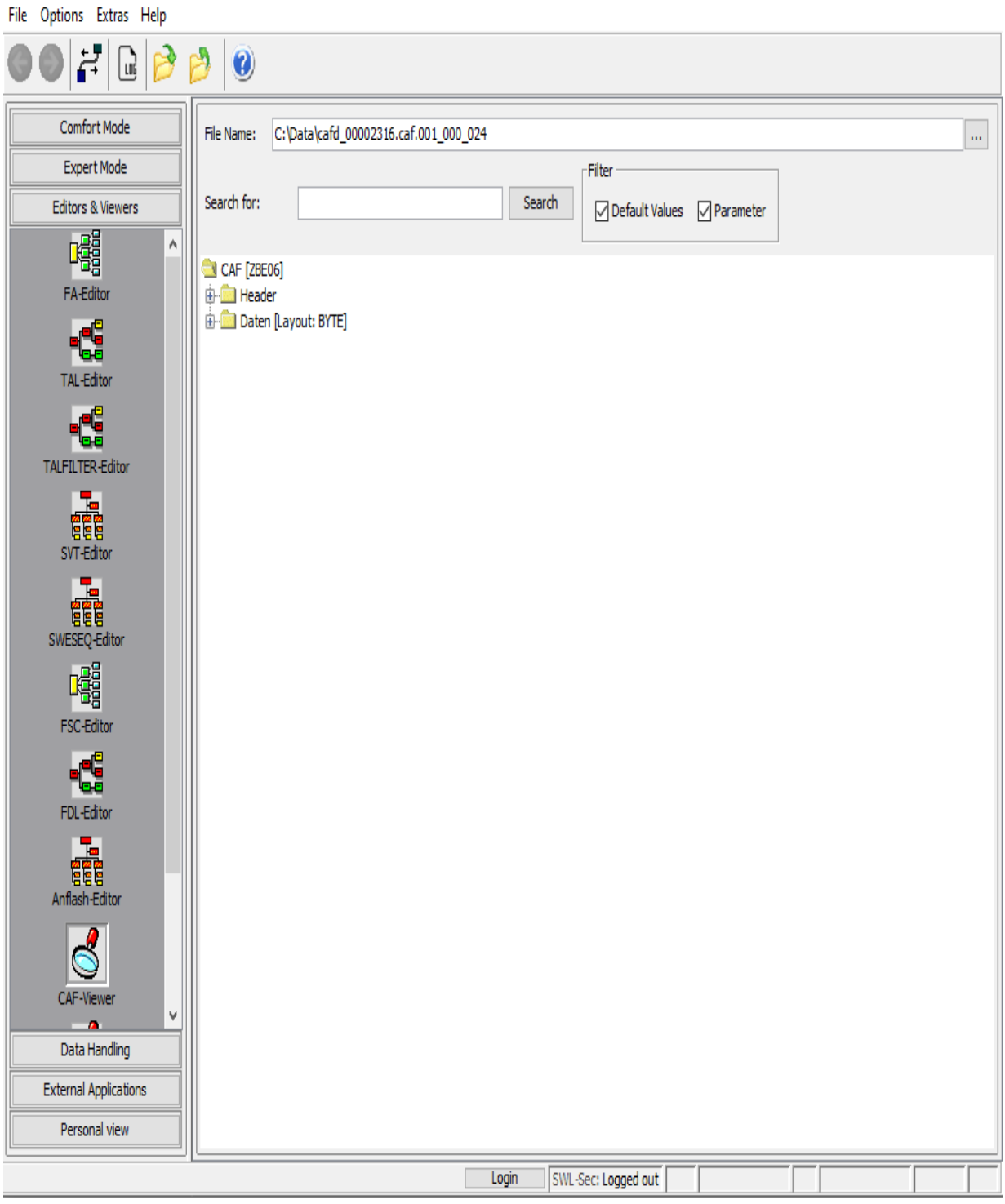
GUI Elements

Master-Config	
Textfield	Absolute path to the Master-configuration file.
Load	Opens a file dialogue to load an existing Master-

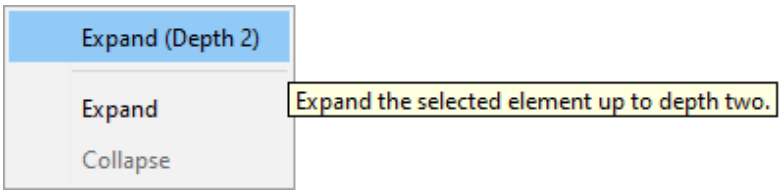
	Config from a file.
Reset	Closes the active file together with all corresponding elements in the Editor-View. This button is only active if a file has been opened before.
Anflash	Opens the currently loaded master-configuration file in module ANFLASH.
FA	
Textfield	Absolute or relative path to the vehicle order (FA). Relative paths can be entered manually.
Load	Opens a file dialogue to load a vehicle order file.
Reset	Resets the content of the text field back to the value, that is defined in the master-configuration file.
ECU Root	
Textfield	Absolute or relative path to the ROOT directory where all ECU-configuration files are located. A relative path entry can be entered manually.
Load	Opens a file dialogue for selection a folder.
Reset	Resets the content of the text field back to the EcuRoot entry, that is defined in the master-configuration file.
ECU-CONFIG	
Tree View	List the names of the ECU-Configs as defined in the Master-configuration file.

CAF-Viewer

The CAF-Viewer views CAFs (Coding Application Files).
A CAF file is a special SWE (software unity) that contains the coding data of a single ECU. All coding data of a single ECU can be distributed on multiple CAFs. As well the entire application software of a ECU can be distributed on multiple SWEs.
The coding data is divided in discrete coding data groups within the CAF file. Every coding data group contains at least one function which has at least one parameter.



Context Menu

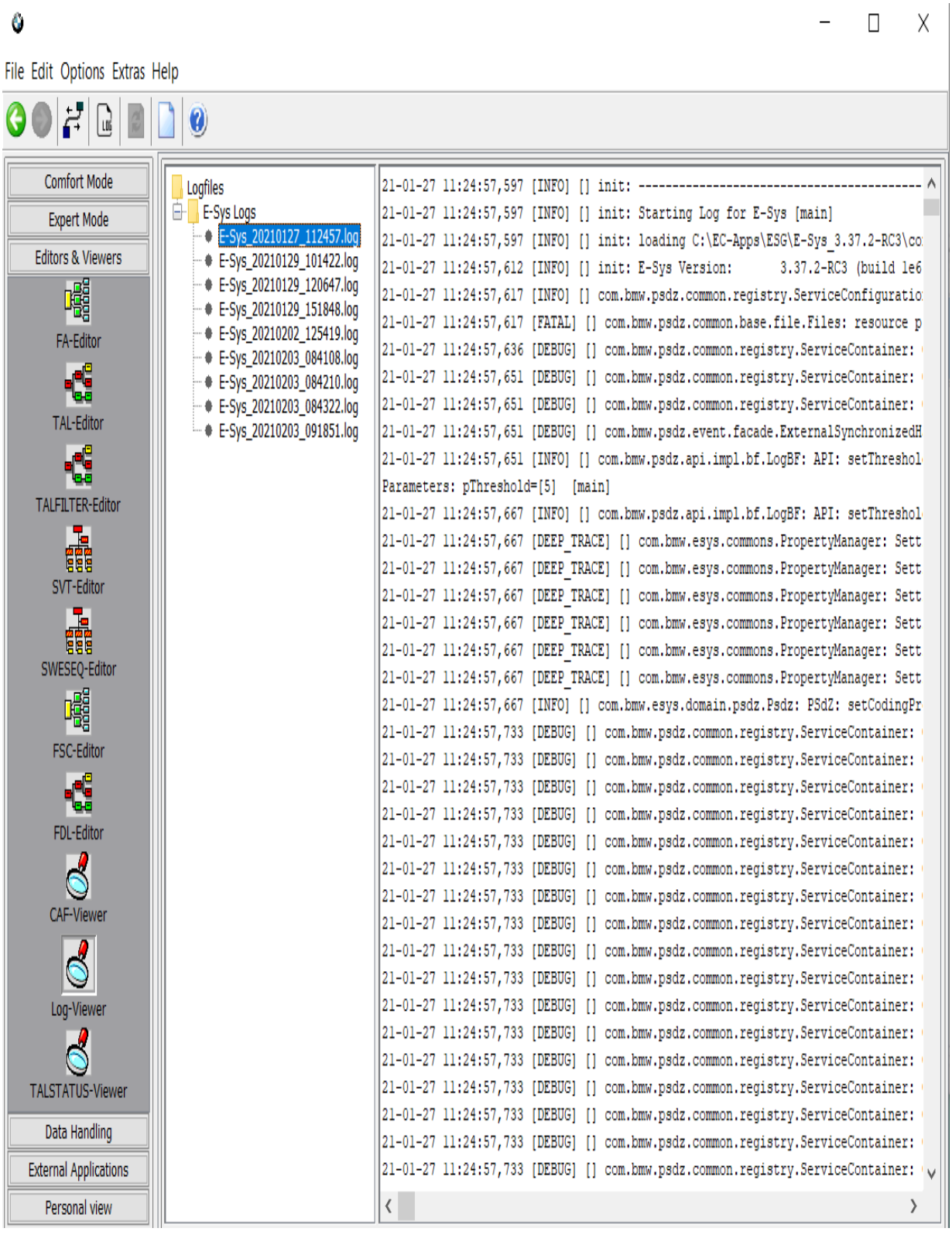


GUI Elements

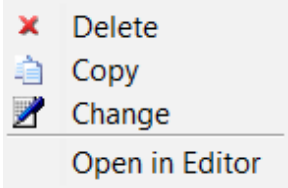
Search Function Name	searches the specified text in the whole tree. If there is a match the tree is expanded to the place of finding. You can continue the search with button F3.
filter	filters corresponding sub trees

LOG-Viewer

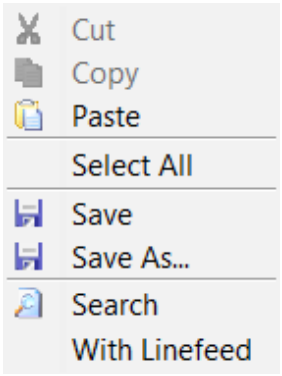
The log file viewer makes for view log files and protocols as well as manage them.



Context-Menu (left side)



Context-Menu (right side)



Specifics

- You have to define a view before viewing log files. Do this on the root on the left side by the context menu. Specify name, path and extension on this occasion.
- The extension can be specified in the form of a regular expression. E.g.: *.*, *.log
- In the view (right window) you have discrete possibilities of editing.

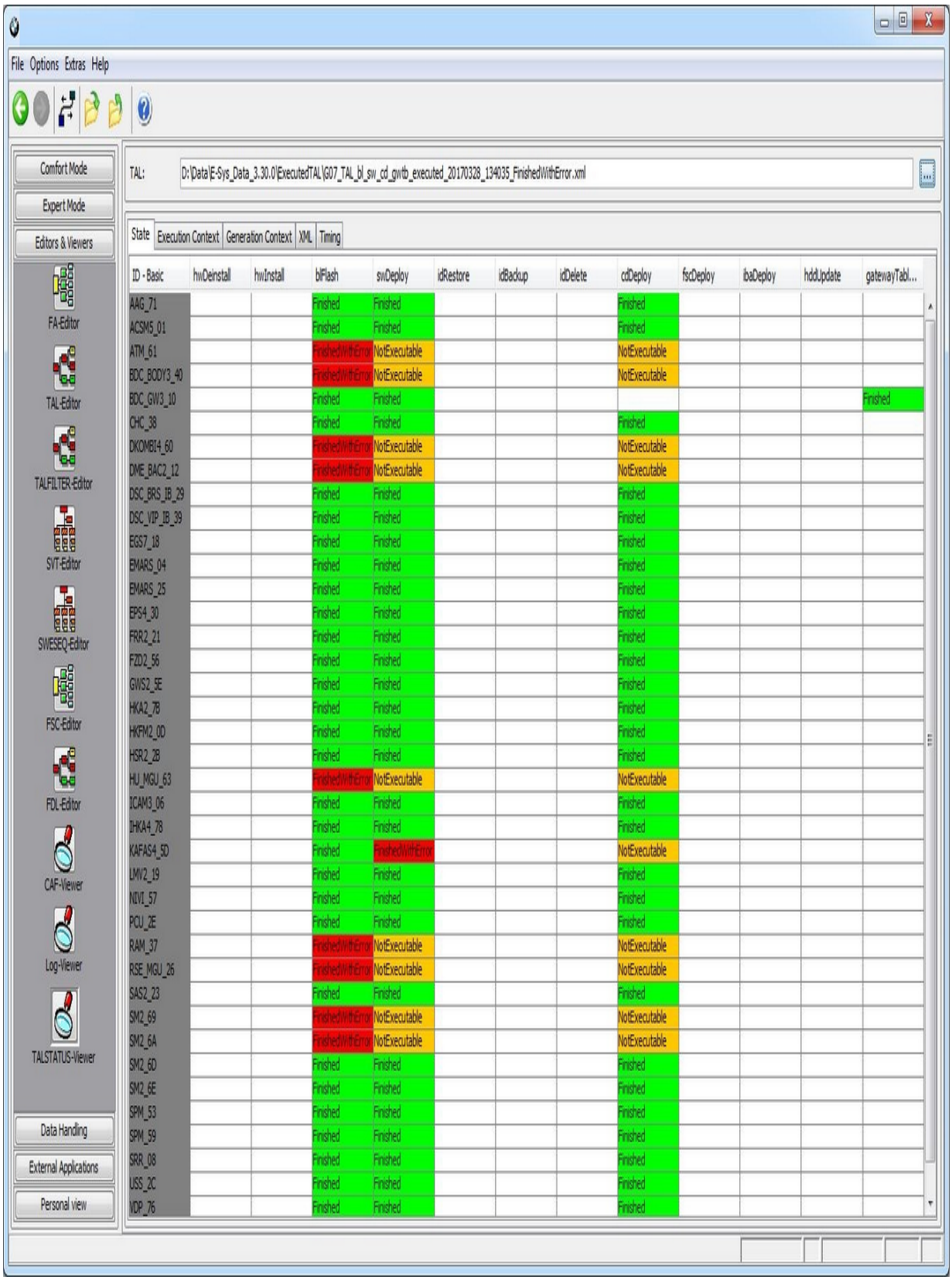
TALSTATUS-Viewer

Note:

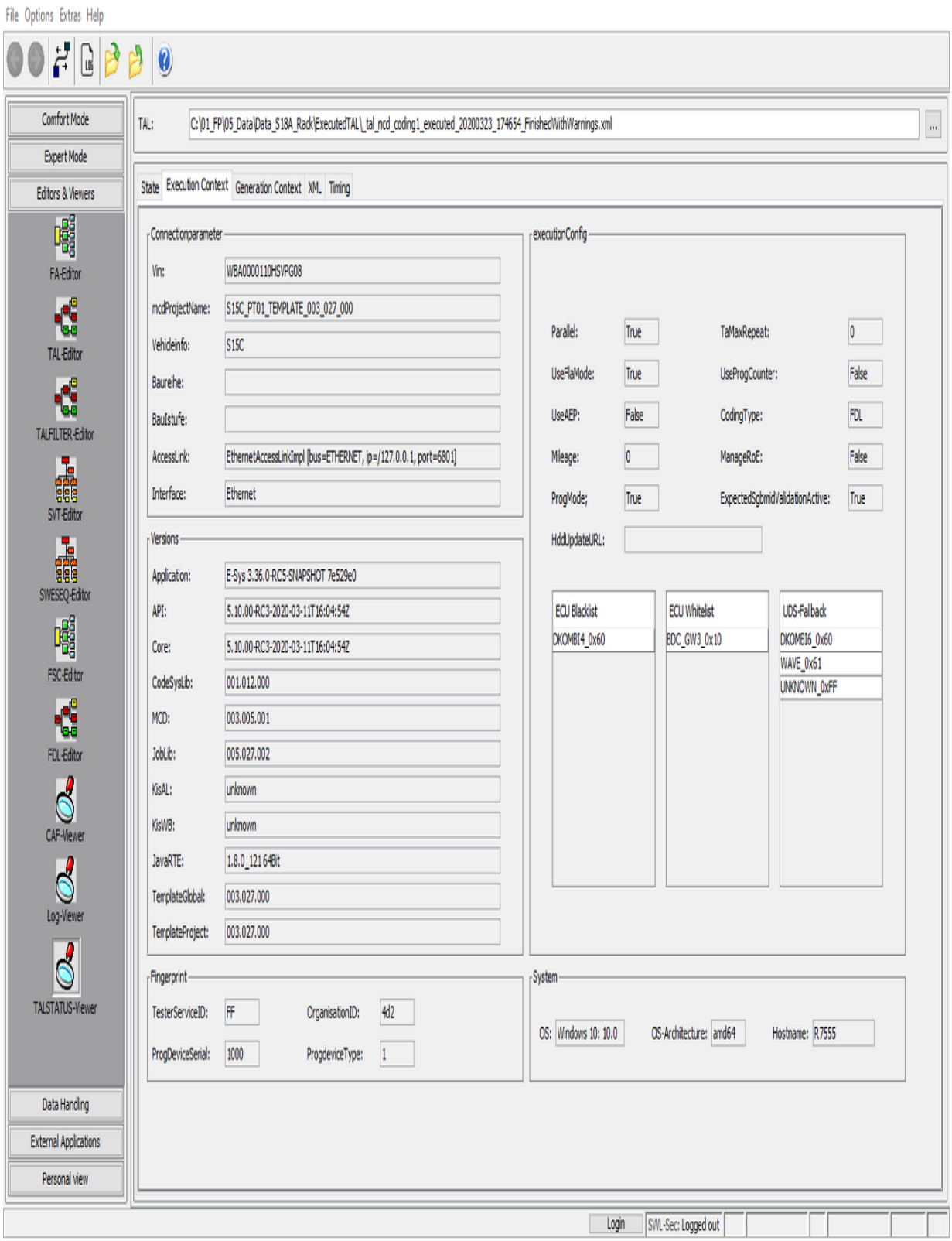
For Executed TALs that contain Mirror Deploy transactions, no graphical representation of the TAL execution process is currently possible.

Module TALSTATUS-Viewer offers the view of an executed TAL in a table with colored status

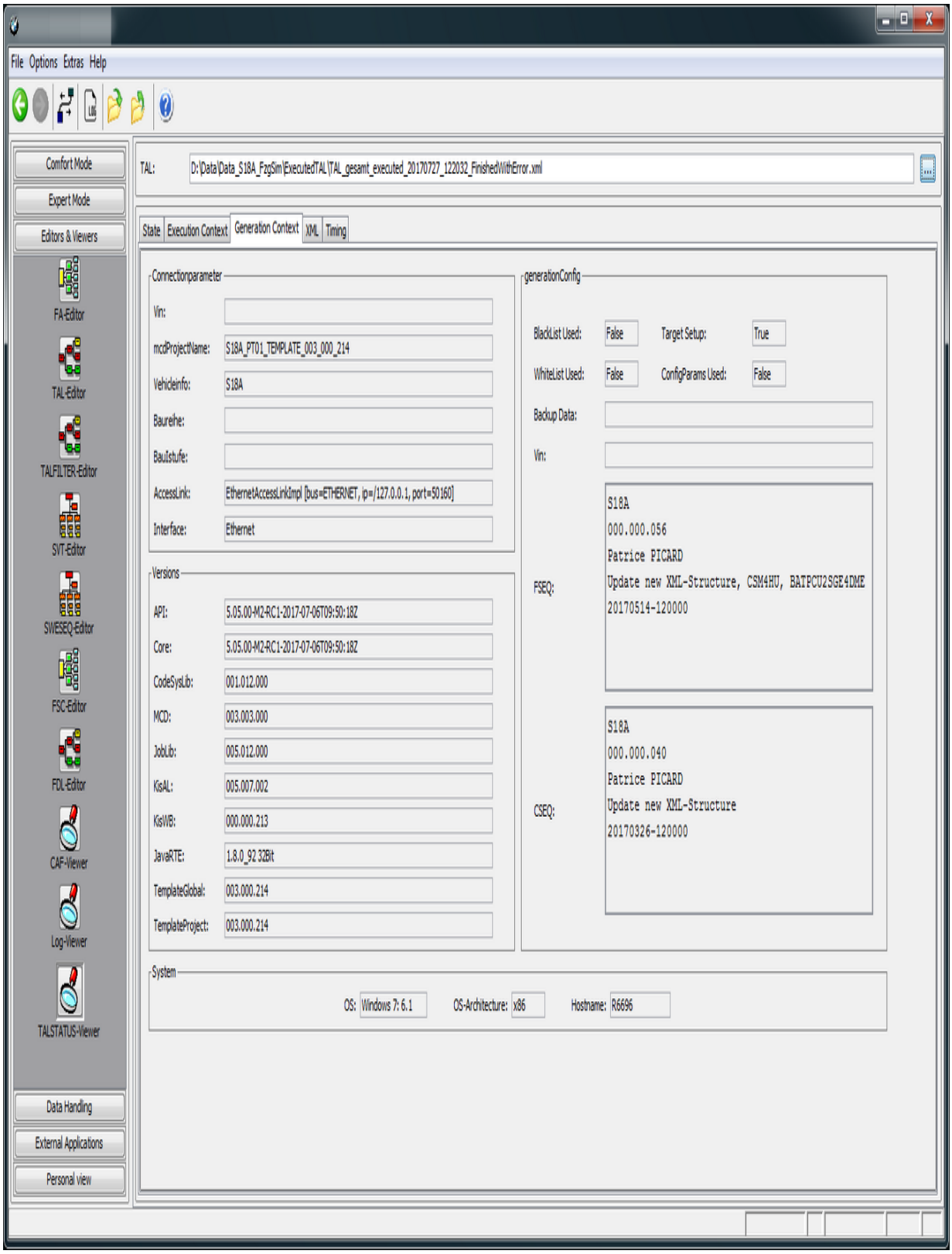
By double clicking on colored status field of the table, the executed TAL is displayed in an editor pane and the corresponding TAL line is marked.


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
TAL: D:\Data\E-Sys_Data_3.30.0\ExecutedTAL\G07_TAL_bl_sw_cd_gwtb_executed_20170328_134035_FinishedWithError.xml
State
Execution Context
Generation Context
XML
Timing
ID - Basic
hwDeinstall
hwInstall
blFlash
swDeploy
idRestore
idBackup
idDelete
cdDeploy
fscDeploy
ibaDeploy
hddUpdate
gatewayTabl...
AAG_71 Finished Finished Finished
ACSM5_01 Finished Finished Finished
ATM_61 FinishedWithError NotExecutable NotExecutable
BDC_BODY3_40 FinishedWithError NotExecutable NotExecutable
BDC_GW3_10 Finished Finished Finished
CHC_38 Finished Finished Finished
DKOMBI4_60 FinishedWithError NotExecutable NotExecutable
DME_BAC2_12 FinishedWithError NotExecutable NotExecutable
DSC_BRS_IB_29 Finished Finished Finished
DSC_VIP_IB_39 Finished Finished Finished
EGS7_18 Finished Finished Finished
EMARS_04 Finished Finished Finished
EMARS_25 Finished Finished Finished
EPS4_30 Finished Finished Finished
FRR2_21 Finished Finished Finished
FZD2_56 Finished Finished Finished
GWS2_5E Finished Finished Finished
HKA2_7B Finished Finished Finished
HKFM2_0D Finished Finished Finished
HSR2_2B Finished Finished Finished
HU_MGU_63 FinishedWithError NotExecutable NotExecutable
ICAM3_06 Finished Finished Finished
IHKA4_7B Finished Finished Finished
KAFAS4_5D Finished FinishedWithError NotExecutable
LMV2_19 Finished Finished Finished
NIVI_57 Finished Finished Finished
PCU_2E Finished Finished Finished
RAM_37 FinishedWithError NotExecutable NotExecutable
RSE_MGU_26 FinishedWithError NotExecutable NotExecutable
SAS2_23 Finished Finished Finished
SM2_69 FinishedWithError NotExecutable NotExecutable
SM2_6A FinishedWithError NotExecutable NotExecutable
SM2_6D Finished Finished Finished
SM2_6E Finished Finished Finished
SPM_53 Finished Finished Finished
SPM_59 Finished Finished Finished
SRR_08 Finished Finished Finished
USS_2C Finished Finished Finished
VDP_76 Finished Finished Finished

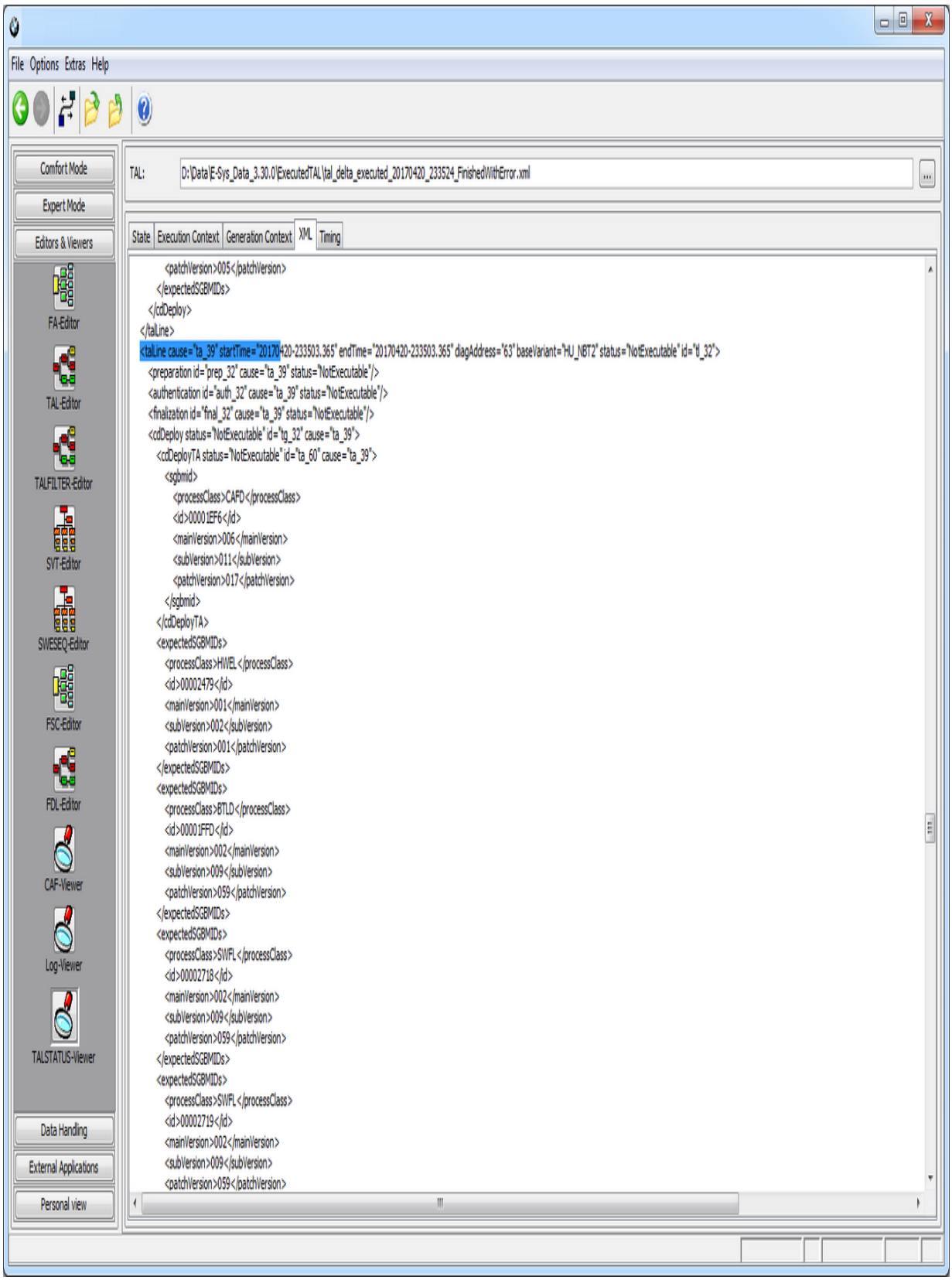
The execution context tab provides information about the tal execution environment, including connection parameters, versions, execution parameters, fingerprint information and operating system information.


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
TAL:
C:\01_FP\05_Data\Data_S18A_Rack\ExecutedTAL_tal_ncd_coding1_executed_20200323_174654_FinishedWithWarnings.xml
State
Execution Context
Generation Context
XML
Timing
Connectionparameter
Vin:
WBA0000110HSVPG08
mcdProjectName:
S15C_PT01_TEMPLATE_003_027_000
VehicleInfo:
S15C
Baureihe:
BauIstufe:
AccessLink:
EthernetAccessLinkImpl [bus=ETHERNET, ip=/127.0.0.1, port=6801]
Interface:
Ethernet
Versions
Application:
E-Sys 3.36.0-RC5-SNAPSHOT 7e529e0
API:
5.10.00-RC3-2020-03-11T16:04:54Z
Core:
5.10.00-RC3-2020-03-11T16:04:54Z
CodeSysLib:
001.012.000
MCD:
003.005.001
JobLib:
005.027.002
KisAL:
unknown
KisVB:
unknown
JavaRTE:
1.8.0_121 64Bit
TemplateGlobal:
003.027.000
TemplateProject:
003.027.000
executionConfig
Parallel:
True
TaMaxRepeat:
0
UseFlaMode:
True
UseProgCounter:
False
UseAEP:
False
CodingType:
FDL
Mileage:
0
ManageRoE:
False
ProgMode;
True
ExpectedSgbmidValidationActive:
True
HddUpdateURL:
ECU Blacklist
DKOMBI4_0x60
ECU Whitelist
BDC_GW3_0x10
UDS-Fallback
DKOMBI6_0x60
WAVE_0x61
UNKNOWN_0xFF
Fingerprint
TesterServiceID:
FF
OrganisationID:
4d2
ProgDeviceSerial:
1000
ProgdeviceType:
1
System
OS:
Windows 10: 10.0
OS-Architecture:
amd64
Hostname:
R7555
Login
SWL-Sec: Logged out

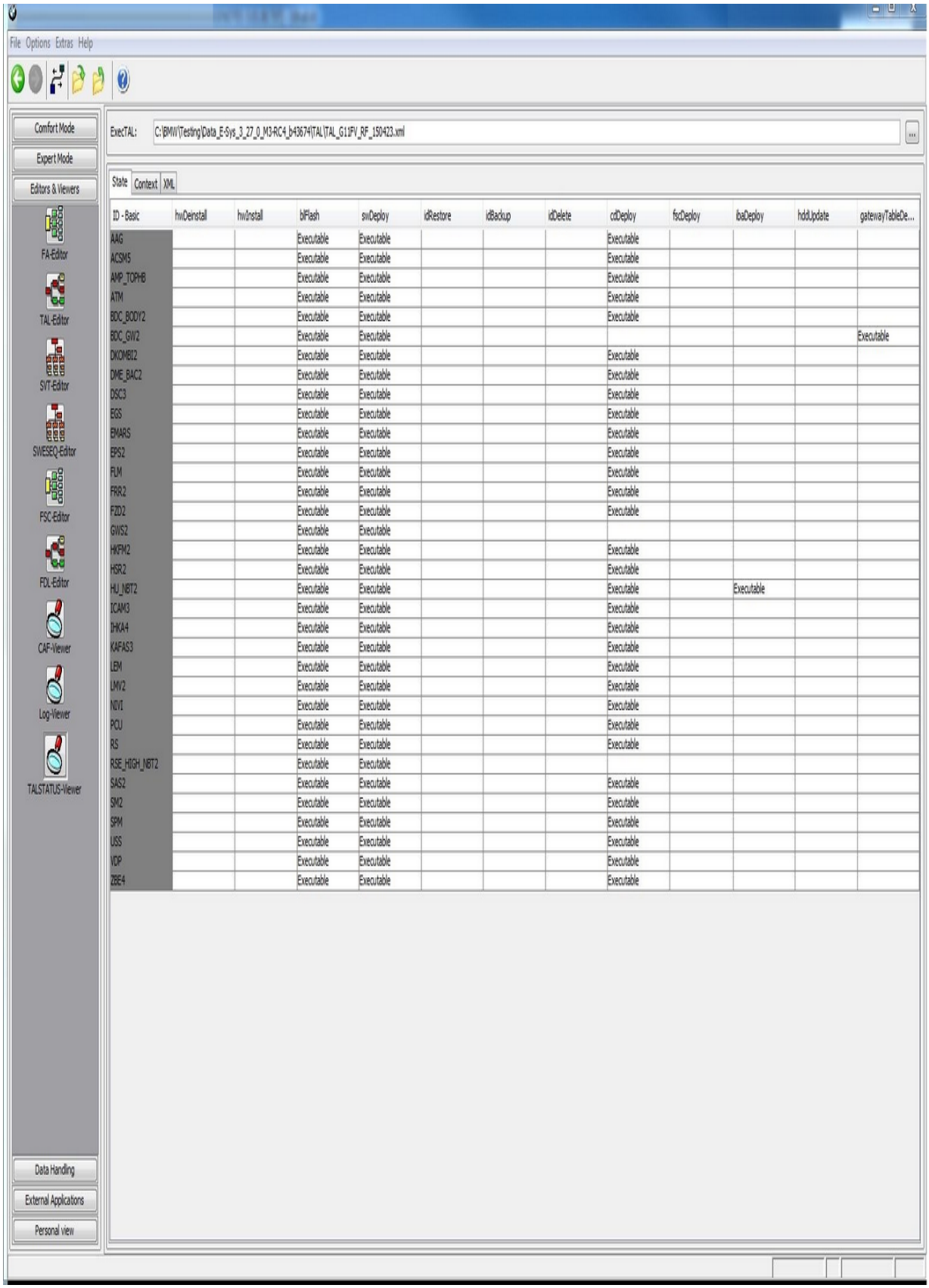
The generation context tab provides information about the tal generation environment, including connection parameters, versions, generation parameters and operating system information.


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
TAL: D:\Data\Data_S18A_FzgSim\ExecutedTAL\TAL_gesamt_executed_20170727_122032_FinishedWithError.xml
State
Execution Context
Generation Context
XML
Timing
Connectionparameter
Vin:
mcdProjectName: S18A_PT01_TEMPLATE_003_000_214
VehicleInfo: S18A
Baureihe:
Bauistufe:
AccessLink: EthernetAccessLinkImpl [bus=ETHERNET, ip=/127.0.0.1, port=50160]
Interface: Ethernet
Versions
API: 5.05.00-M2-RC1-2017-07-06T09:50:18Z
Core: 5.05.00-M2-RC1-2017-07-06T09:50:18Z
CodeSysLib: 001.012.000
MCD: 003.003.000
JobLib: 005.012.000
KisAL: 005.007.002
KisWB: 000.000.213
JavaRTE: 1.8.0_92 32Bit
TemplateGlobal: 003.000.214
TemplateProject: 003.000.214
generationConfig
BlackList Used: False
Target Setup: True
WhiteList Used: False
ConfigParams Used: False
Backup Data:
Vin:
FSEQ:
S18A
000.000.056
Patrice PICARD
Update new XML-Structure, CSM4HU, BATPCU2SGE4DME
20170514-120000
CSEQ:
S18A
000.000.040
Patrice PICARD
Update new XML-Structure
20170326-120000
System
OS: Windows 7: 6.1
OS-Architecture: x86
Hostname: R6696

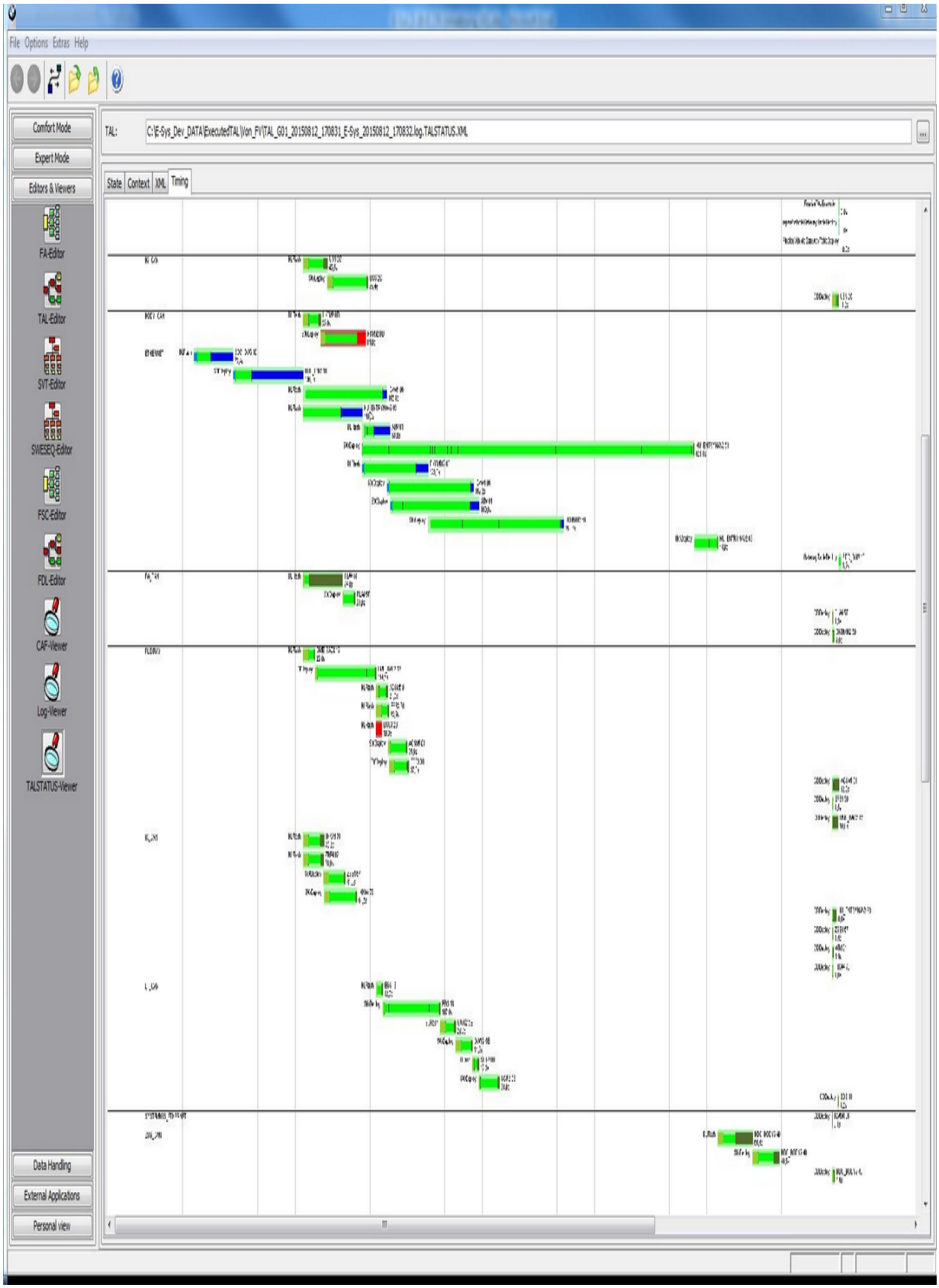
By double clicking on colored status field of the table, the executed TAL is displayed in an editor pane and the corresponding TAL line is marked.


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
TALFILTER-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
TAL:
D:\Data\E-Sys_Data_3.30.0\ExecutedTAL\tal_delta_executed_20170420_233524_FinishedWithError.xml
State
Execution Context
Generation Context
XML
Timing
<patchVersion>005</patchVersion>
</expectedSGBMIDs>
</cdDeploy>
</taLine>
<taLine cause="ta_39" startTime="20170420-233503.365" endTime="20170420-233503.365" diagAddress="63" baseVariant="HU_NBT2" status="NotExecutable" id="tl_32">
<preparation id="prep_32" cause="ta_39" status="NotExecutable"/>
<authentication id="auth_32" cause="ta_39" status="NotExecutable"/>
<finalization id="final_32" cause="ta_39" status="NotExecutable"/>
<cdDeploy status="NotExecutable" id="tg_32" cause="ta_39">
<cdDeployTA status="NotExecutable" id="ta_60" cause="ta_39">
<sgbmid>
<processClass>CAFD</processClass>
<id>00001EF6</id>
<mainVersion>006</mainVersion>
<subVersion>011</subVersion>
<patchVersion>017</patchVersion>
</sgbmid>
</cdDeployTA>
<expectedSGBMIDs>
<processClass>HWEL</processClass>
<id>00002479</id>
<mainVersion>001</mainVersion>
<subVersion>002</subVersion>
<patchVersion>001</patchVersion>
</expectedSGBMIDs>
<expectedSGBMIDs>
<processClass>BTLD</processClass>
<id>00001FFD</id>
<mainVersion>002</mainVersion>
<subVersion>009</subVersion>
<patchVersion>059</patchVersion>
</expectedSGBMIDs>
<expectedSGBMIDs>
<processClass>SWFL</processClass>
<id>00002718</id>
<mainVersion>002</mainVersion>
<subVersion>009</subVersion>
<patchVersion>059</patchVersion>
</expectedSGBMIDs>
<expectedSGBMIDs>
<processClass>SWFL</processClass>
<id>00002719</id>
<mainVersion>002</mainVersion>
<subVersion>009</subVersion>
<patchVersion>059</patchVersion>

Instead of an executed tal also a tal can be opened in order to get a quick overview about the transactions which will be executed.


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
ExecTAL:
C:\BMW\Testing\Data_E-Sys_3_27_0_M3-RC4_b43674\TAL\TAL_G11FV_RF_150423.xml
State
Context
XML
ID - Basic
hwDeinstall
hwInstall
blFlash
swDeploy
idRestore
idBackup
idDelete
cdDeploy
fscDeploy
ibaDeploy
hddUpdate
gatewayTableDe...
AAG Executable Executable Executable
ACSM5 Executable Executable Executable
AMP_TOPHB Executable Executable Executable
ATM Executable Executable Executable
BDC_BODY2 Executable Executable Executable
BDC_GW2 Executable Executable Executable
DKOMBI2 Executable Executable Executable
DME_BAC2 Executable Executable Executable
DSC3 Executable Executable Executable
EGS Executable Executable Executable
EMARS Executable Executable Executable
EPS2 Executable Executable Executable
FLM Executable Executable Executable
FRR2 Executable Executable Executable
FZD2 Executable Executable Executable
GWS2 Executable Executable
HKFM2 Executable Executable Executable
HSR2 Executable Executable Executable
HU_NBT2 Executable Executable Executable Executable
ICAM3 Executable Executable Executable
IHKA4 Executable Executable Executable
KAFAS3 Executable Executable Executable
LEM Executable Executable Executable
LMV2 Executable Executable Executable
NIVI Executable Executable Executable
PCU Executable Executable Executable
RS Executable Executable Executable
RSE_HIGH_NBT2 Executable Executable
SAS2 Executable Executable Executable
SM2 Executable Executable Executable
SPM Executable Executable Executable
USS Executable Executable Executable
VDP Executable Executable Executable
ZBE4 Executable Executable Executable

In the Timing tab the timing sequence is displayed. If the executed tal is loaded for the first time in the viewer, a jpg file is generated and displayed. If the jpg file is already in the same directory as the executed tal, it is displayed only. With the mouse wheel one can zoom the picture.


File Options Extras Help
Comfort Mode
Expert Mode
Editors & Viewers
FA-Editor
TAL-Editor
SVT-Editor
SWESEQ-Editor
FSC-Editor
FDL-Editor
CAF-Viewer
Log-Viewer
TALSTATUS-Viewer
Data Handling
External Applications
Personal view
TAL:
C:\E-Sys_Dev_DATA\ExecutedTAL\Von_FV\TAL_G01_20150812_170831_E-Sys_20150812_170832.log.TALSTATUS.XML
State
Context
XML
Timing

Expert Mode

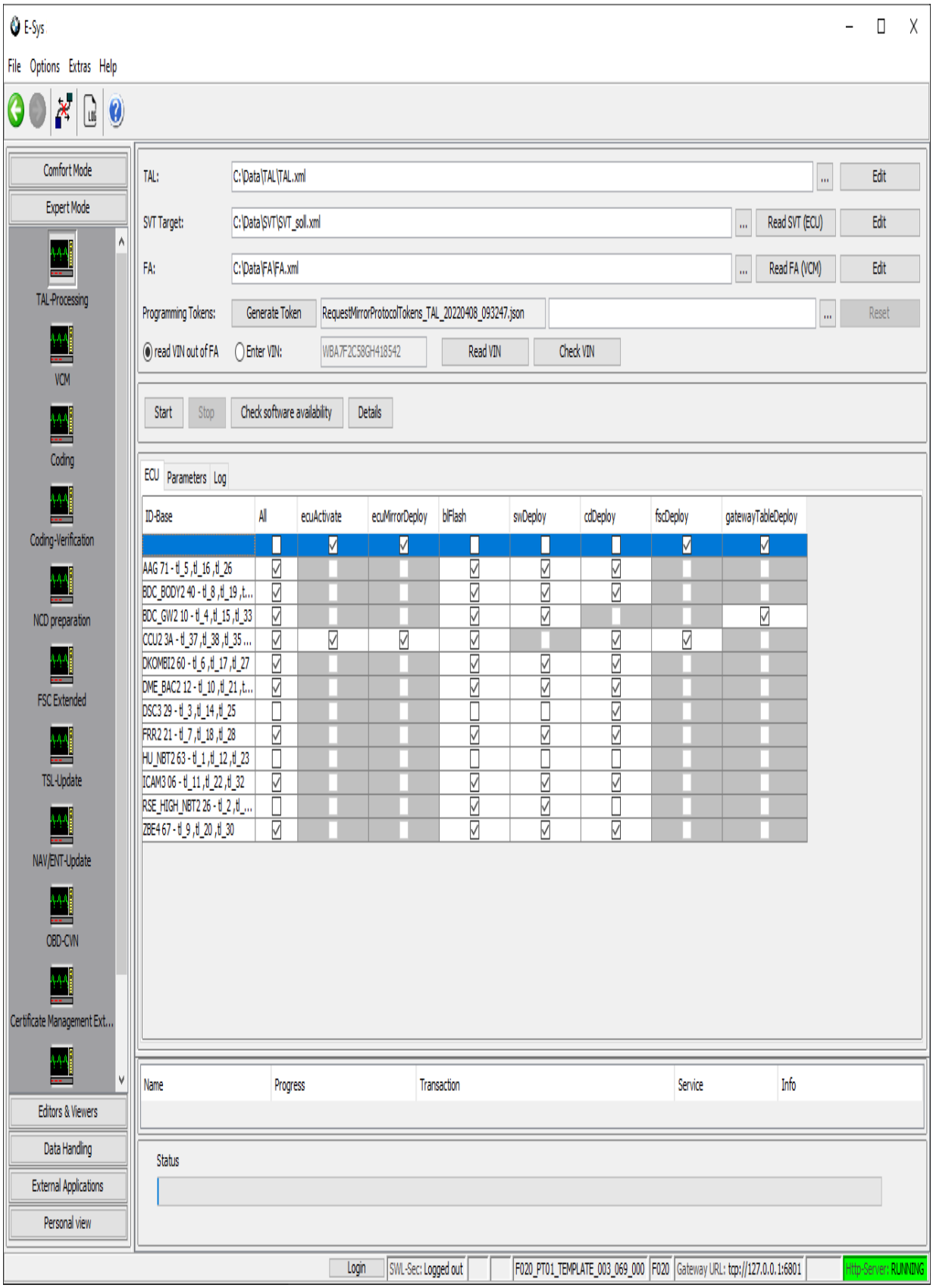
TAL-Processing
VCM
Coding
Coding-Verification
NCD-Preparation
FSC-Extended
TSL Update
NAV/ENT Update
OBD-CVN
Certificate-Management
SFA / ECU-Mode-Extended
KDS-Extended

TAL-Processing

In the module TAL-Processing a TAL is processed.
A TAL is a model year plan with actions that are to be processed on a single ECU, a system of Emus or the entire vehicle. It doesn't count where the TAL is obtained from: The TAL can be loaded from the file system or adopted from another E-Sys module.
A TAL consists at least of one TAL line. There is maximal one TAL line for every ECU and transaction category.

If you try to execute a TAL with assembly or disassembly of hardware a message pops up and you can operate to save the information of the ECU by the menu Extras->Individualdatenrettung that is specific for this individual ECU. Information that can't be retrieved on the HO network nor from other Emus of the same type.
These files are saved in a backup directory. Now the HW can be exchanged and the information added to the TAL in Calculation of TAL mask.
See there further information.

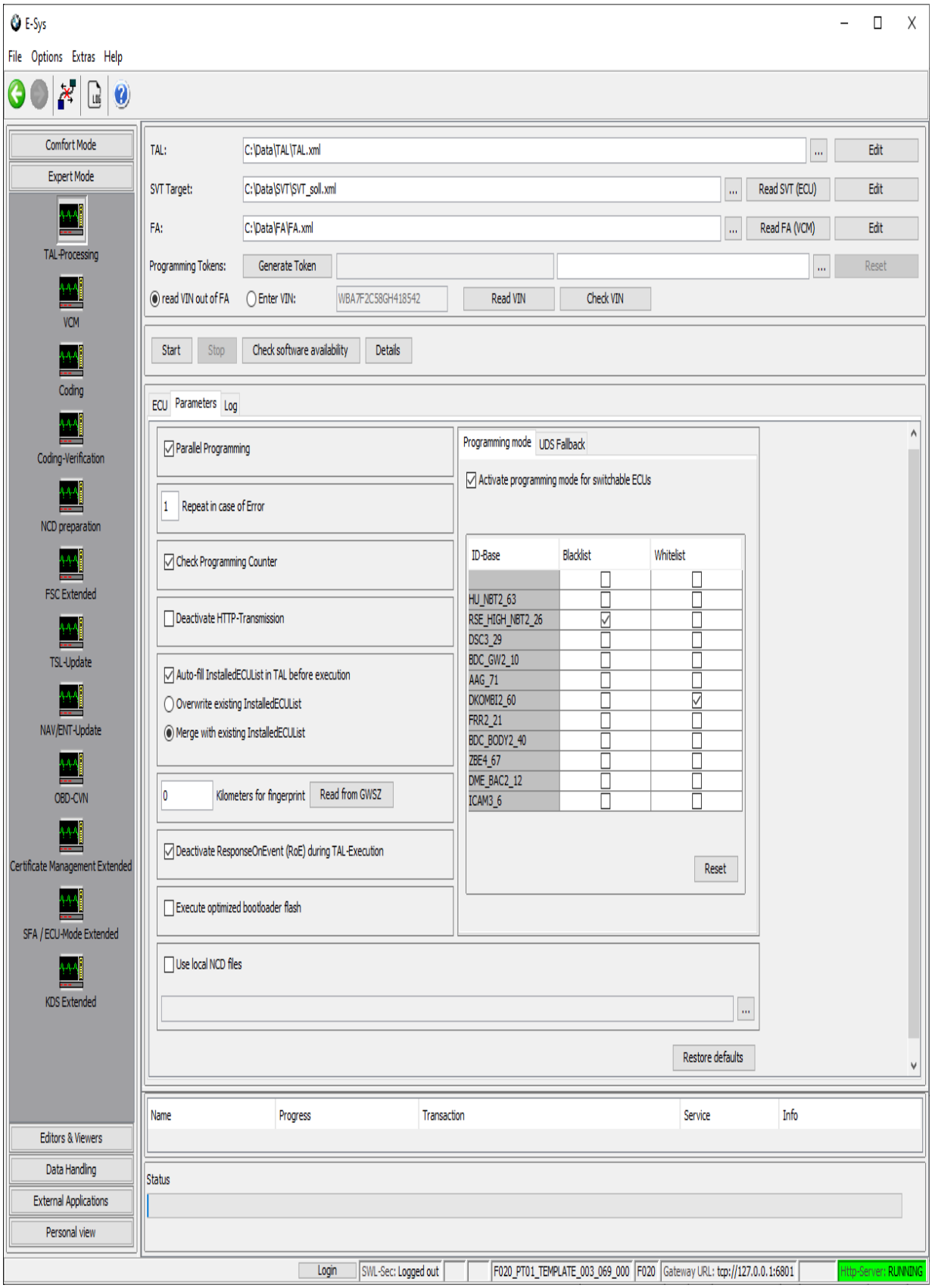
Note:
Before starting the TAL processing, a BAT/HAF verification is carried out.
It will be checked whether a BAT/HAF control unit is installed. If this is the case, the status of switch S1 is determined.
When the switch is open, a message appears informing the user about possible risks for the TAL-execution process.

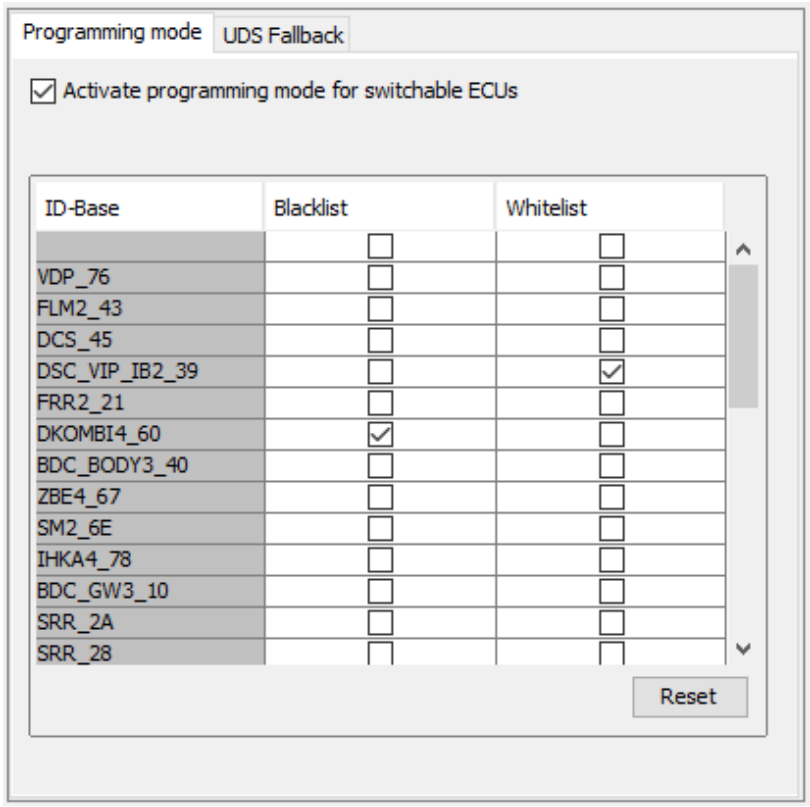

E-Sys
File Options Extras Help
Comfort Mode
Expert Mode
TAL-Processing
VCM
Coding
Coding-Verification
NCD preparation
FSC Extended
TSL-Update
NAV/ENT-Update
OBD-CVN
Certificate Management Ext...
Editors & Viewers
Data Handling
External Applications
Personal view
TAL: C:\Data\TAL\TAL.xml ... Edit
SVT Target: C:\Data\SVT\SVT_soll.xml ... Read SVT (ECU) Edit
FA: C:\Data\FA\FA.xml ... Read FA (VCM) Edit
Programming Tokens: Generate Token RequestMirrorProtocolTokens_TAL_20220408_093247.json ... Reset
read VIN out of FA Enter VIN: WBA7F2C58GH418542 Read VIN Check VIN
Start Stop Check software availability Details
ECU Parameters Log
ID-Base All ecuActivate ecuMirrorDeploy blFlash swDeploy cdDeploy fscDeploy gatewayTableDeploy
AAG 71 - tl_5 ,tl_16 ,tl_26
BDC_BODY2 40 - tl_8 ,tl_19 ,t...
BDC_GW2 10 - tl_4 ,tl_15 ,tl_33
CCU2 3A - tl_37 ,tl_38 ,tl_35 ...
DKOMBI2 60 - tl_6 ,tl_17 ,tl_27
DME_BAC2 12 - tl_10 ,tl_21 ,t...
DSC3 29 - tl_3 ,tl_14 ,tl_25
FRR2 21 - tl_7 ,tl_18 ,tl_28
HU_NBT2 63 - tl_1 ,tl_12 ,tl_23
ICAM3 06 - tl_11 ,tl_22 ,tl_32
RSE_HIGH_NBT2 26 - tl_2 ,tl_...
ZBE4 67 - tl_9 ,tl_20 ,tl_30
Name Progress Transaction Service Info
Status
Login SWL-Sec: Logged out F020_PT01_TEMPLATE_003_069_000 F020 Gateway URL: tcp://127.0.0.1:6801 Http-Server: RUNNING

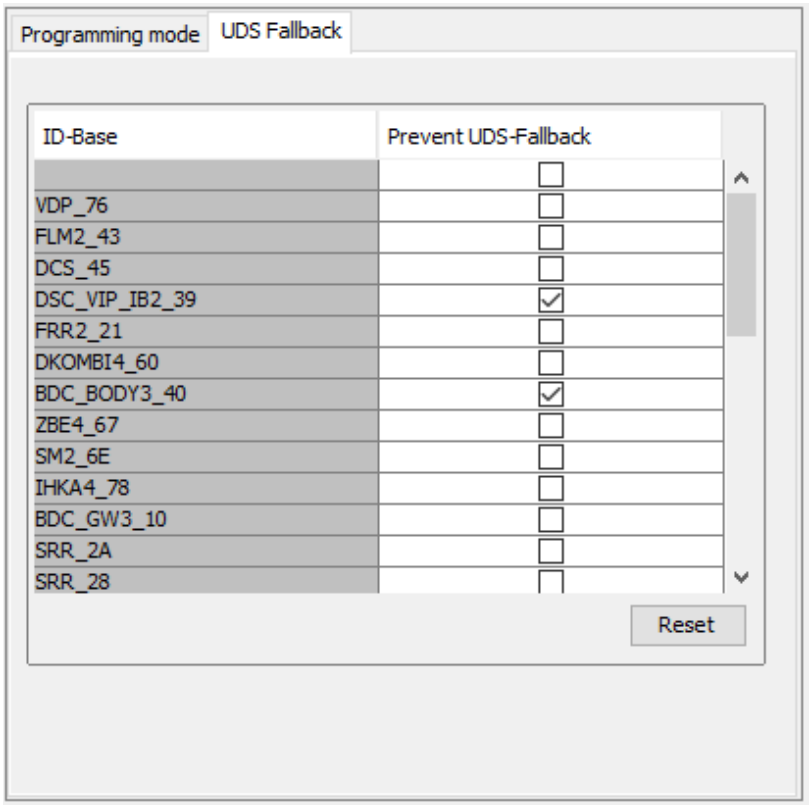
GUI Elements

TAL	Using the three-dot-button a file dialog is started to choose a TAL file. The file path will be shown in the text field. With the "Edit" button you can change to the TAL-Editor to make some changes to the TAL.
SVT	Using the three-dot-button a file dialog is started to choose a target SVT file. The file path will be shown in the text field. With the "Edit" button you can change to the SVT-Editor to make some changes to the SVT. The SVT will be written to the VCM before the flashing the Emus. Pressing button "Read SVT (ECU)" the SVT will be generated by reading the content of the connected Emus. The SVT is optional.
FA	Using the three-dot-button a file dialog is started to choose a vehicle order file. The file path will be shown in the text field. With the "Edit" button you can change to the FA-Editor to make some changes to the vehicle order. Pressing the "Read FA (VCM)" button the vehicle order will be read from the ECU. An FA is optional for flashing but mandatory for coding.
Programming Tokens	Using the button "Generate token" a file with unsigned programming tokens will be created. The file name is displayed in the text field. This file must be signed outside of E-Sys. Using the three-dot-button the file dialog is started to choose a file with signed Programming Tokens. The file path is displayed in the right text field. A signed Programming Token is required to execute a TAL with Mirror-Depoy actions!

read VIN out of FA	If selected the VIN will be taken from the vehicle order for the TAL execution.
Enter VIN	if the selected the VIN will be taken from the text field for the TAL execution.
Read VIN	Read the VIN.
Start	Start of TAL processing. This button turns to "Pause", after another click (disabled for parallel programming) the processing pauses and button turns to "Resume". The next click of this button continues the processing of the TAL.
Stop	After the current processed TALline the processing is aborted.
Check software availability	Checks if all software required for TAL execution is available.
ECU	
rows	List the existing transaction categories for the loaded TAL.
columns	List all transactions for each control unit. If a control unit does not have a transaction for a category, the corresponding cell is displayed in gray. If the executionStatus of the transaction differs from "executable", the corresponding executionStatus is displayed. An executable transaction can be activated or deactivated by checkbox.


E-Sys
File Options Extras Help
Comfort Mode
Expert Mode
TAL-Processing
VCM
Coding
Coding-Verification
NCD preparation
FSC Extended
TSL-Update
NAV/ENT-Update
OBD-CVN
Certificate Management Extended
SFA / ECU-Mode Extended
KDS Extended
Editors & Viewers
Data Handling
External Applications
Personal view
TAL: C:\Data\TAL\TAL.xml ... Edit
SVT Target: C:\Data\SVT\SVT_soll.xml ... Read SVT (ECU) Edit
FA: C:\Data\FA\FA.xml ... Read FA (VCM) Edit
Programming Tokens: Generate Token ... Reset
read VIN out of FA Enter VIN: WBA7F2C58GH418542 Read VIN Check VIN
Start Stop Check software availability Details
ECU Parameters Log
Parallel Programming
1 Repeat in case of Error
Check Programming Counter
Deactivate HTTP-Transmission
Auto-fill InstalledECUList in TAL before execution
Overwrite existing InstalledECUList
Merge with existing InstalledECUList
0 Kilometers for fingerprint Read from GWSZ
Deactivate ResponseOnEvent (RoE) during TAL-Execution
Execute optimized bootloader flash
Programming mode UDS Fallback
Activate programming mode for switchable ECUs
ID-Base Blacklist Whitelist
HU_NBT2_63
RSE_HIGH_NBT2_26
DSC3_29
BDC_GW2_10
AAG_71
DKOMBI2_60
FRR2_21
BDC_BODY2_40
ZBE4_67
DME_BAC2_12
ICAM3_6
Reset
Use local NCD files
Restore defaults
Name Progress Transaction Service Info
Status
Login SWL-Sec: Logged out F020_PT01_TEMPLATE_003_069_000 F020 Gateway URL: tcp://127.0.0.1:6801 Http-Server: RUNNING





Parameters	Parameters for selection of special features for

	TAL processing.
Parallel Programming	Specification if the Emus should be flashed parallel.
Repeat	Definition of the number of repetitions in case of error.
Programming Counter	Specification if the Programming Counter should be activated.
HTTP-Transmission	Specification if the HTTP-Transmission should be activated.
InstalledECUList	Specification if the InstalledECUList in TAL should be auto-filled before execution.
Kilometers	Definition of the Mileage in Kilometers for fingerprint.
ResponseOnEvent	Specification if the ResponseOnEvent during TAL-Execution should be deactivated. This setting is not active for a DIRECT connection.
Optimizable bootloader flash	Specification if the optimizable bootloader flash should be performed or is not allowed. If activated the system is performing an check which of the software (SWFL, SWFK) that has to be falshed, based on TAL, is after the bootloader-flash already installed. After these check only the needed, not already installed, software is going to be flashed.
Use local NCD files	Specification if to use the pre-signed and locally available NCD data. Notes: - in this directory a subfolder is searched with the VIN as name - NCD files must be signed with the VIN from the used FA
Tab "Programming mode"	Definition of a black / white list for the mode switching of individual control units. Whitelist -> Control unit must be switched. Blacklist -> Control unit must not be switched.
Tab	If the HTTP data transfer of the software

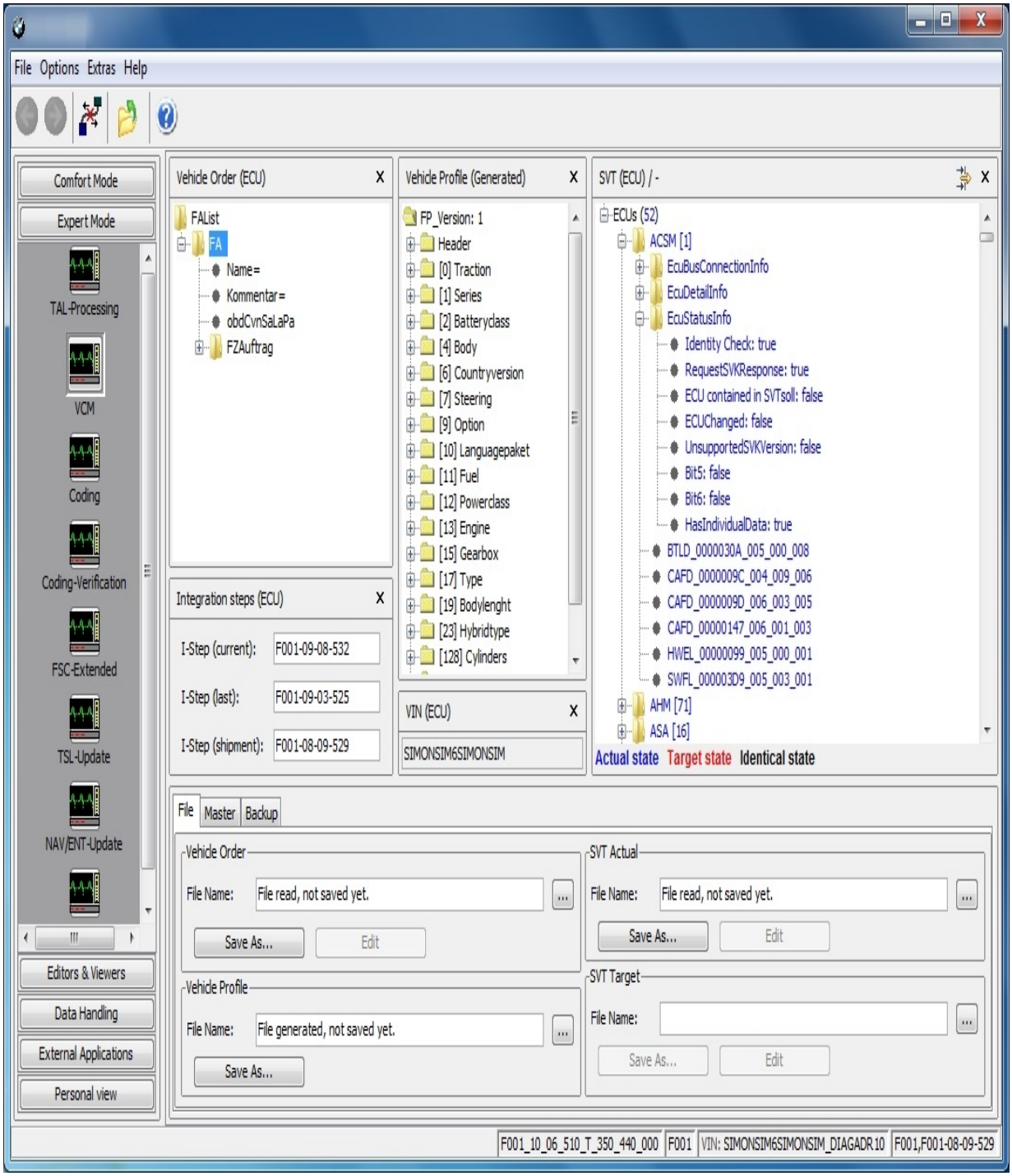
"Programming mode"	packages to the ECU fails, the UDS protocol is used as fallback. In this table, the fallback mechanism can be deactivated for the selected ECU. If an error now occurs in the HTTP transfer, the flash process for the selected ECU is aborted with error.
Log	
Clear	Clear the log window.
Events	Specification if the events should be displayed in the logging.
EventType	Type of the logged events are specified.
Progress	
Table	Visualization of the currently executed transactions and services for the appropriate ECU (Name).
Progress Bar	Visualization of progress in TAL processing.

Reading Vehicle Order and VIN

After reading the vehicle order the VIN will always be read too. Also the VIN could be read independently by pressing the button "Read VIN". The vehicle order and the VIN will be read from the master ECU. If an error occurs they will be read from the backup ECU. In the status bar is shown where the vehicle order and the VIN were actually read from.

VCM

The Module VCM provides functions of the VCM (Vehicle Configuration Management). The VCM is a component in the vehicle that records discrete states and information of the vehicle. The VCM is no ECU but a function of the vehicle. VCM's functions are addressed by a special (reduced) vehicle info specification thus during the begin of a session the application communicates first with VCM and afterwards addresses the Emus. The module VCM has been extended significantly in its functionality for integration step 3.0.0. Besides the present reliable functions now also control functions to backup partner CAS are possible. The VCM functionality itself is provided actually by the ZGW ECU.



GUI Elements

Vehicle Order (source)	Display window for FA-List with specification of source.
×	Clear display window for the Vehicle Order (FA) and delete correspondent data in the

	RAM.
Vehicle Profile (source)	Display window for Vehicle Profile (FP) with specification of source.
×	Clear display window for FP
SVT (source)/(source)	Display window for SVT Actual and SVT Target with specification of sources.
×	Clear display window for SVTs.
Integration Steps	Read/write elements show data for current, last and shipment.
×	Clear display window for the integration steps.
VIN (source)	Display of VIN
×	Clear display window for the VIN.
	Provide masking discrete elements of the displayed SVT in display window.

Control Elements in "File" tag

Three-dot-Button	Load respective data (FA, FP or SVT) from file system.
Save as	Save respective data (FA, FP or SVT) into file system.
Edit	Open the respective editor and load current data (FA or SVT) to provide it for display.



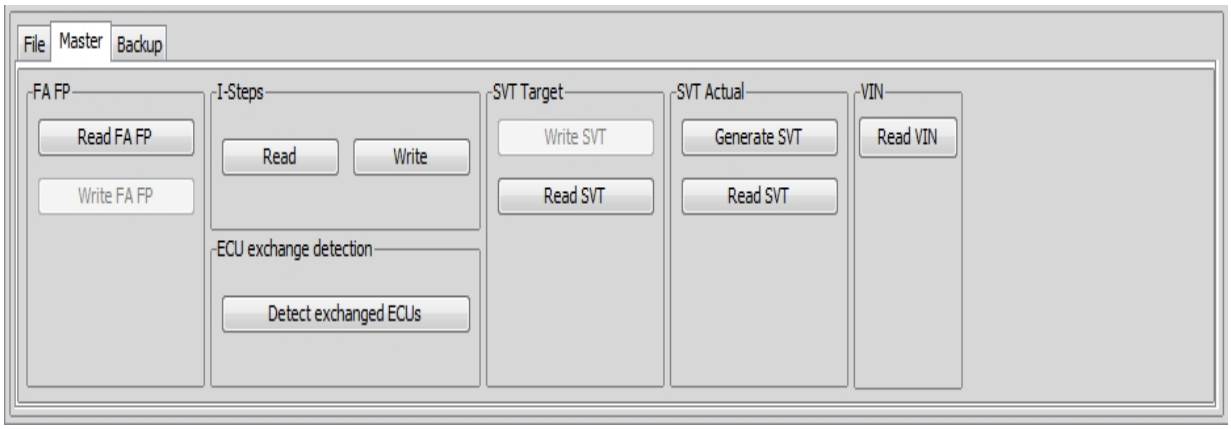
Control Elements in Sub mask "SVT Filter"

check box "Only SVK"	Determines if filter "Only SVK" is set. If filter is set only the type of SVKs displayed in the combo box, are displayed in the SVK window - but not further information like ECU info or BUS info.
belonging combo box	Contains all SVKs and the entry "all".
check box "Ecu Info"	Determines if filter "Ecu Info" is set.
belonging combo box	Contains all ECU Info (like e.g. ActiveNotifyEnabled, IdentityCheck or UnsupportedSVKVersion)
check box "Bus Info"	Determines if filter "Bus Info" is set.
belonging combo box	Contains all bus variants of LIN, MOST, CAN, Flexray.

Control Elements in "Master" tag

FA FP: Read FA FP	Read FA, FP and VIN from VCM and display information.
FA FP: Write FA FP	Write current displayed FA and FP to VCM.
Update VIN	Write VIN displayed in input form to VCM.
I-Steps: Read	Read I-step current, last and shipment from VCM and display information.
I-Steps: Write	Write I-steps displayed in the mask (current, last and shipment) to VCM.
ECU exchanged detection: Detect exchanged Emus	Start routine for detection of exchanged Emus. The result are the bold marked Emus.
SVT Target: Write SVT	Write current SVT target to VCM.

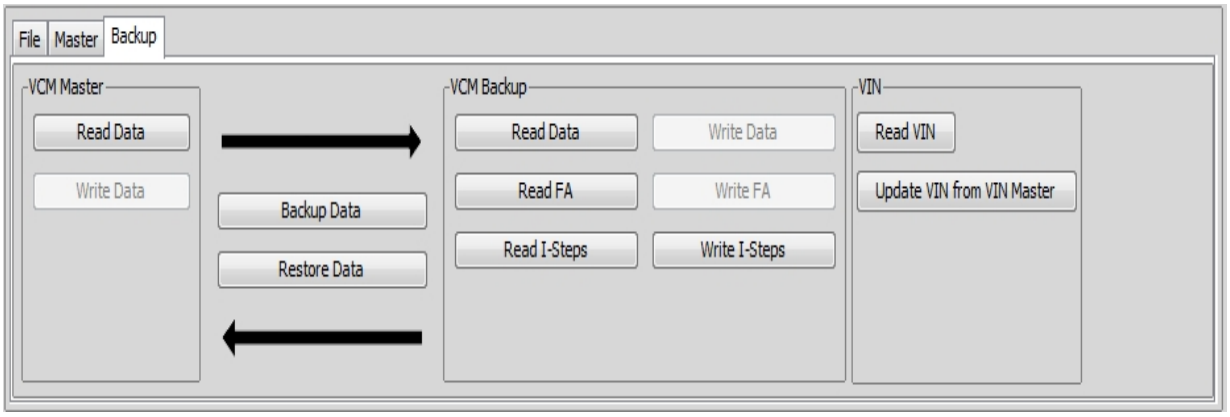
Read SVT	Read SVT target from VCM and display it.
SVT Actual: Generate SVT	Determine SVT from returned SVKs of the discrete Emus.
SVT Actual: Read SVT	Read SVT actual from VCM and display it..
VIN: read VIN	Read VIN from master.



Control Elements in "Backup" tag

VCM Master: Read Data	Read and display all backup relevant data (FA, FP, SVT Target, I-Steps) from VCM.
VCM Master: Write Data	Write all displayed data (FA, FP, SVT Target, I-Steps) to VCM.
Backup Data	Master backup function: Read and display all backup relevant data (FA, FP, SVT Target, I-Steps) from VCM and write it to Backup Partner (normally CAS).
Restore Data	Master backup function: Read and display FA, FP, SVT target and I-Steps from CAS and write to VCM.
VCM Backup: Read Data	Read and display FA, FP, SVT

	Target and I-Steps from CAS.
VCM Backup: Write Data	all backup relevant and displayed data (FA, FP, SVT target, I-Stufen) are written to the backup partner (usually CAS)
VCM Backup: Read FA	Read and display FA and VIN from CAS.
VCM Backup: Write FA	Write displayed FA to CAS.
VCM Backup: Read I-Steps	Display I-Steps read from Backup Partner (normally CAS).
VCM Backup: Write I-Steps	Write displayed I-Steps to Backup Partner (normally CAS).
VIN: Read VIN	Read VIN from backup.
VIN: Update VIN from VIN Master	Update VCM backup VIN with VCM master VIN.

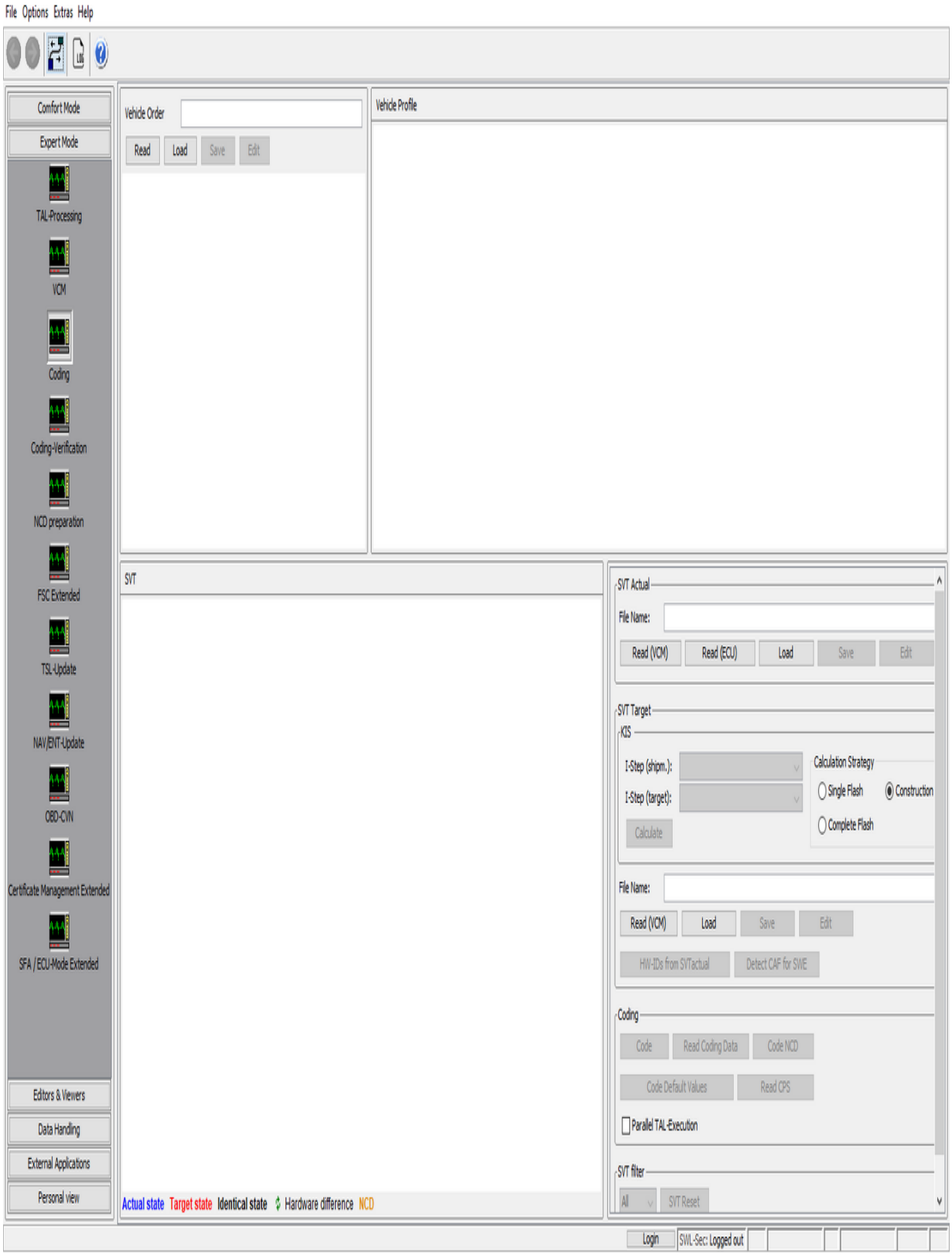


Special Topics

- ECU exchange detection: A routine compares actual serial numbers of the connected Emus with a reference list. Differing serial numbers belong to exchanged Emus. These are marked as bold.
- Interplay ZGW (VCM) and Backup Partner (CAS)

Coding

With module Coding it's possible to select one or multiple Emus for coding without creating a TAL manually for this purpose.


File Options Extras Help
Comfort Mode
Expert Mode
TAL-Processing
VCM
Coding
Coding-Verification
NCD preparation
FSC Extended
TSL-Update
NAV/ENT-Update
OBD-CVN
Certificate Management Extended
SFA / ECU-Mode Extended
Editors & Viewers
Data Handling
External Applications
Personal view
Vehicle Order
Read
Load
Save
Edit
Vehicle Profile
SVT
Actual state
Target state
Identical state
Hardware difference
NCD
SVT Actual
File Name:
Read (VCM)
Read (ECU)
Load
Save
Edit
SVT Target
KIS
I-Step (shipm.):
I-Step (target):
Calculate
Calculation Strategy
Single Flash
Construction
Complete Flash
File Name:
Read (VCM)
Load
Save
Edit
HW-IDs from SVTactual
Detect CAF for SWE
Coding
Code
Read Coding Data
Code NCD
Code Default Values
Read CPS
Parallel TAL-Execution
SVT filter
All
SVT Reset
Login
SWL-Sec: Logged out

The SVTs recorded in red letters in the picture above indicate the reference system state of an ECU. The blue ones indicate the actual state of the ECU. Black entries indicate SVTs with identical reference and actual state. If the hardware of an ECU is different, this is indicated by the arrow symbol.

GUI Elements

Vehicle Data	
Vehicle Order	File containing a list of vehicle orders. For selecting a single vehicle order, this order needs to be activated by using the context menu item 'Activate FA'. If the list of vehicle orders contains only one element, this element will be automatically activated as soon as the file is loaded.
Read	Read out the actual vehicle order from the vehicle.
Load	Open a vehicle order from the file system.
Save	Opens a file dialog to save the vehicle order.
Edit	Change to the FA-Editor and open the vehicle order file.
Vehicle Profile	The Vehicle profile will be automatically generated from an activated Vehicle Order and displayed in a tree structure.
SVT Actual	
Read (VCM)	read out the vehicle order from the vehicle.
Read (ECU)	read out the vehicle order functionally
Load	Open a vehicle order from the file system.
Save	Opens an file dialog to save the SVT.
Edit	Change to the SVT-Editor and open the SVT
KIS/SVT target	
I-Step (shipm.)	Available shipment integration steps
I-Step (target)	Available target integration steps
Calculation strategy	Possible strategy to calculate the SVT (Single Flash | Complete Flash | Construction Progress).
File Name	The path of the SVT file.
Calculate	Calculates an SVT-Target by integration step and

	vehicle order (and SVT-Actual if exists).
Read (VCM)	Read SVT-Target from VCM
Load, Save	Load/save an SVT-Target from/to file system.
Edit	Edit in the corresponding editor.
HW-IDs from SVTactual	HWEL, HWAP und HWFR elements from SVTactual will be copied to the Emus in the target SVT.
Detect CAF for SWE	*not supported yet*
Coding	
Code	Code the ECU.
Read Coding Data	Read the coding data from ECU.
Code NCD	Code the ECU with the selected NCD.
Code Default Values	Code the ECU to the default state.
Read CPS	Read out the CPS.
Parallel TAL-Execution	If checked coding will be executed parallel.
Stop TAL on error	if checked the coding will stop if one error occurs.
Filter	
Filter	filter SVT tree by selected process class
SVT Reset	reset SVT/SVK-Ist/Soll and delete SVT tree

Reading Vehicle Order and VIN

After reading the vehicle order the VIN will always be read too and put into the vehicle order (FAList/FA/FZAuftrag/Header/Vinlong). The vehicle order and the VIN will be read from the master ECU. If an error occurs they will be read from the backup ECU. In the status bar is shown where the vehicle order and the VIN were actually read from.

Note:

Before starting the coding process (Code, Code NCD or Code Default Values), a BAT/HAF verification is carried out.

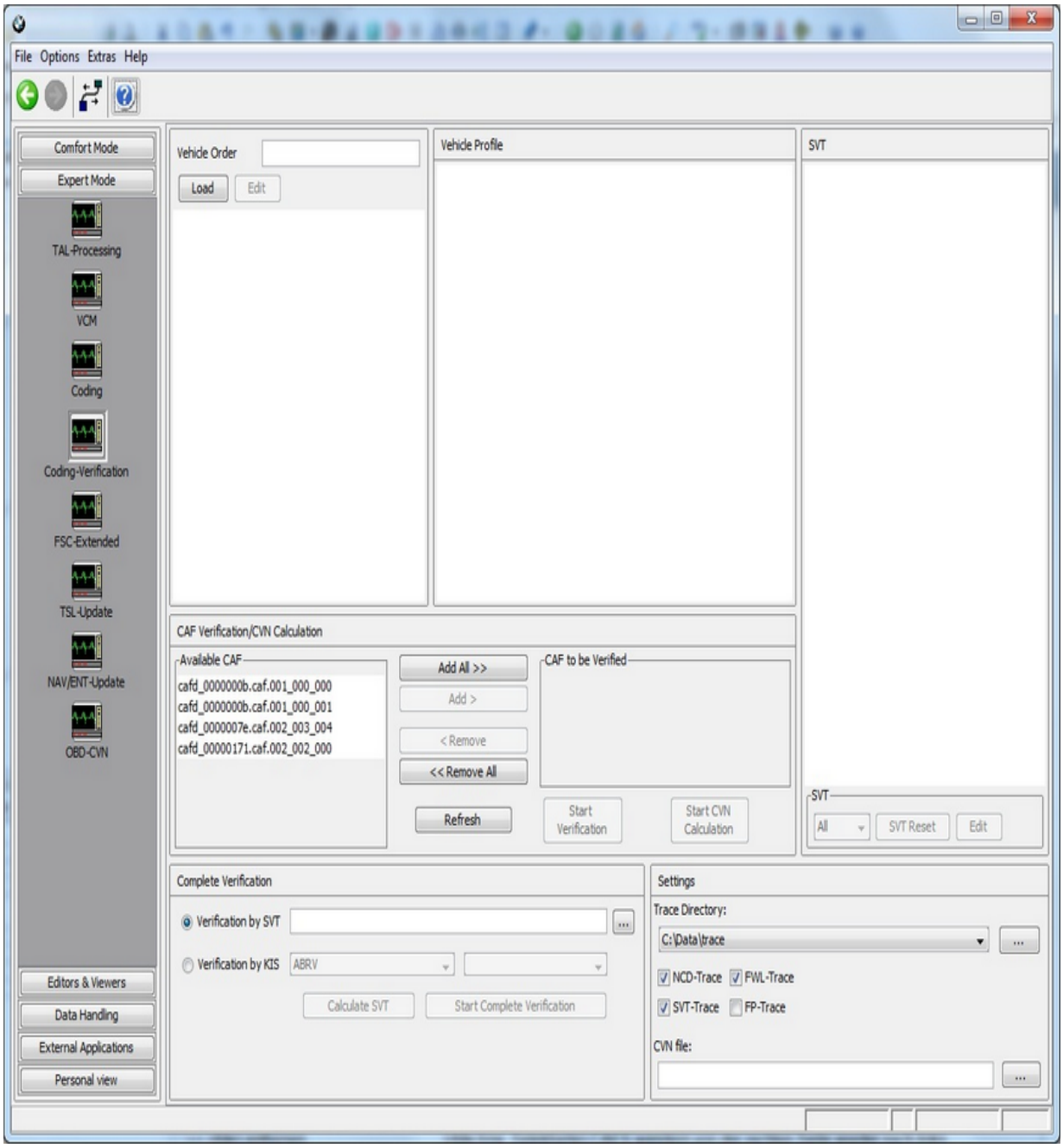
It will be checked whether a BAT/HAF control unit is installed. If this is the case, the status of switch S1 is determined.

When the switch is open, a message appears informing the user about possible risks for the coding process.

Coding-Verification

In module Coding-Verification is checked offline, if the discrete parts (FA, CAF, SVT) match together. In detail there are verification of CAF, entire verification by SVT and by KIS (in the latter also an SVT is created).

The CVN calculation computes the Sub CVNs of CAF files and saves the result in a file.



GUI Elements

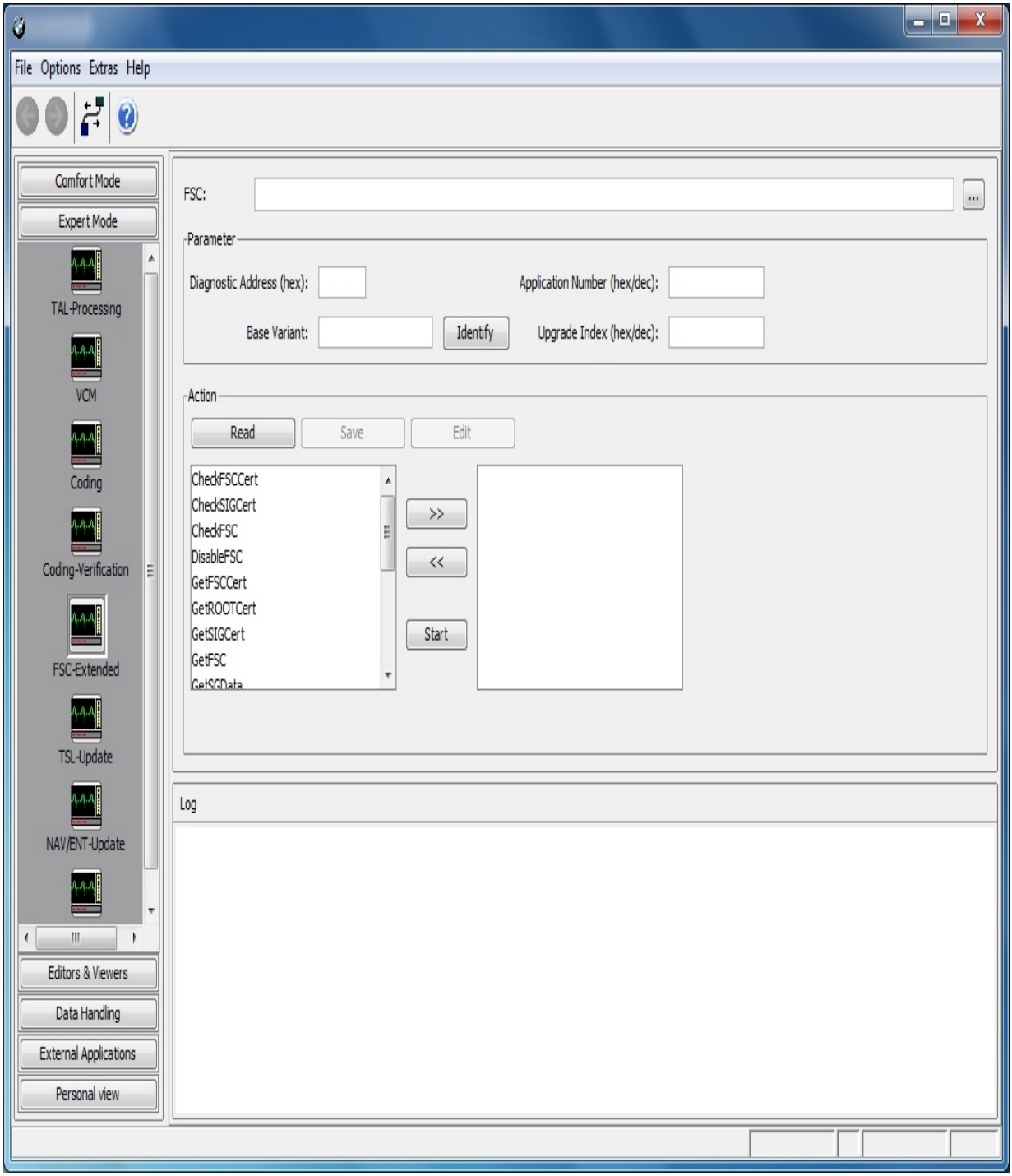
Vehicle Data	
Vehicle Order	File containing a list of vehicle orders. For selecting a single vehicle order, this order needs to be activated by using the context menu item ‘Activate FA’. If the list of vehicle orders contains only one element, this element will be activated by default as soon as the file is loaded.
Vehicle Profile	The Vehicle profile will be automatically generated from an activated Vehicle Order and displayed in a tree structure.
Load	Load file with list of vehicle orders from file system.
Edit	Load current Vehicle Order to FA/FP editor
CAF-Verification	
Available CAFs	This window displays the available CAFs in the proper file name format.
CAFs to be verified	These transferred CAFs will be verified when the "Start Verification" button is pressed.
Add (all) >>	Transfer selected CAFs from the left hand side to the right hand side respectively all of them.
<< Remove (all)	Remove items on the right hand side to the left side.
Refresh	Refresh the two CAF windows.
Start verification	Start the process of the CAFs' verificatiion.
Entire Verification	
Radio button "Verification by SVT" and "Verification by KIS"	In "Verification by KIS" additionally an SVT is calculated and afterwards verified
"..."-Button	Load SVT from the file sytem for verification by SVT.
Combo box	For selecting the integration step for calculating

	of SVT.
Start Entire Verification	This button starts "Verification by SVT" or "Verification by KIS" depending on the state of radio button.
Start CVN Calculation	Starts the CVN calculation an writes the result to the file selected by "Settings/CVN file"
Calculate SVT	Calculate SVT from integration step and further data.
Settings	
Trace Directory	Specification of an existing trace path (by combo box) or input of a new path by three-dot-button-dialogue.
NCD, FWL, SVT, FP trace	The discrete types of traces can be selected in any combination. For "Verification by KIS" the SVT trace is strictly recommended (due to SVT creation).
CVN file	Target file for CVN calculation in ObdCvnFaList format.

FSC-Extended

The module FSC Extended in expert mode serves for read out, check and sign of FSC. In addition to that single jobs can be executed in batch mode.

Corresponding to the topic FSC there are other masks in E-Sys. The module **Freischaltcode** (FSC) serves for inquiring the FSC State, for writing and (de-)activating of FSC. FSC's can be displayed and edited in **FSC-Editor** easily.



GUI Elements

FSC File	The FSC to be written
Key	The key corresponding to the FSC
Diagnosis Address	of the ECU

Software-ID	the software identifier
Base Variant	Name of ECU variant
Upgrade Index	of FSC
Read FSC	The FSC in the ECU is read.
Edit FSC	Forward FSC to module FSC editor
Check FSC	Check FSC (using public key)
Sign FSC	The current FSC is signed using the specified (private) key.
">>", "<<"	The jobs in the left hand window, selected by the mouse are copied to the right hand window.
Start	The jobs residing in the right hand window are started.
Log	The log window displays the state of the currently processed action.

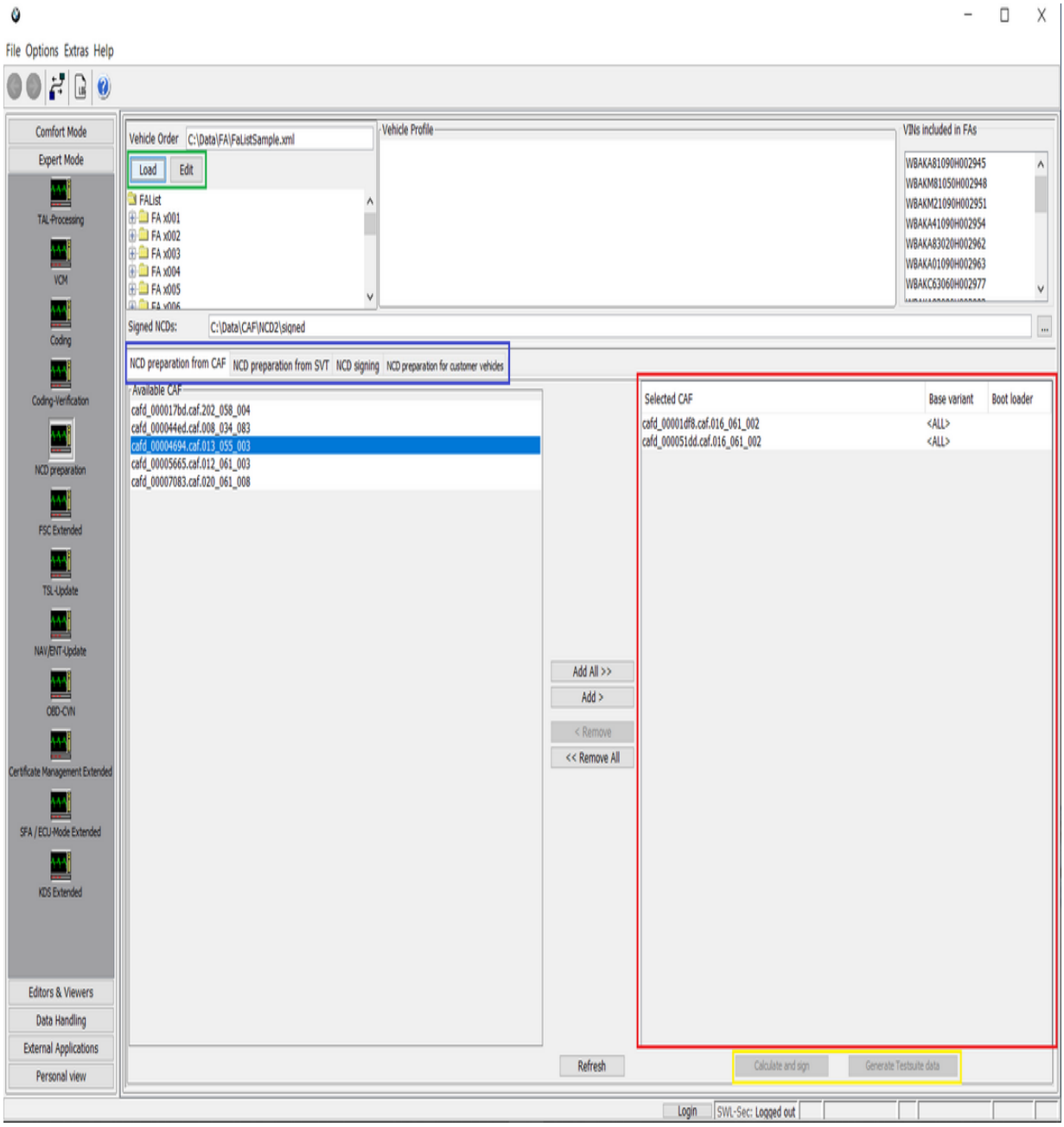
NCD-Preparation

The new coding concept (secure coding) required, that all needed NCDs have to be signed before they can be transferred to the ECU.

Therefore you have now with these new module the possibility to signing the NCDs.

Attention:

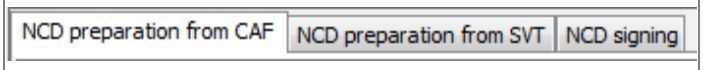
If no pdx-container is imported, the user has to perform an container import and restart e-sys afterwards! Otherwise the user is not able to use these module correctly!



You have the options to sign already local available NCDs (tap "NCD signing") or E-Sys has to calculate the NCDs before they can be signed (tap "preparation from SVT" or "preparation from CAFD").

GUI-Elements

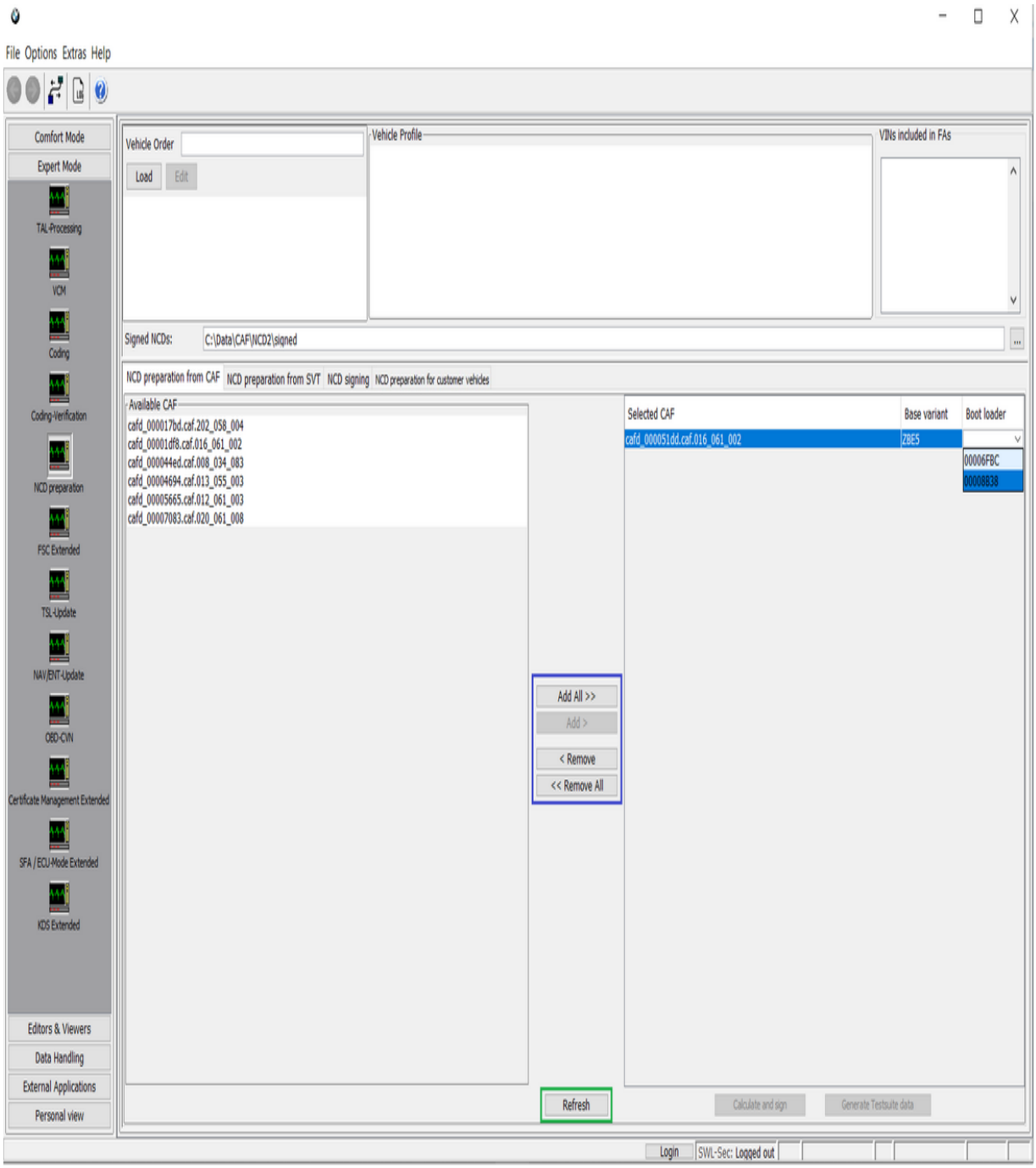
Vehicle-order (FA) (green marking)	Load an existing FA. You can also edit it, in these case you

	are going to switch the module automatically. As soon as a FA is loaded, all VIN-entries included in the vehicle order, will appear in the upper right section of the display. (only if tap "NCD preparation from CAF" or "NCD preparation from SVT" are selected).
Vehicle-profile (FP)	The FP is shown as soon as the FA has been activated. If more than one FA-entry are selected, no FP will be displayed.
Choose the location for the signed NCDs	If you are going to edit the signedNCD location the option-setup is going to appear. After you changed the location click the Button "OK" and the new location will be visible in these module.
Tap for the three different ways to sign NCDs (blue marking) 	Depending on the tap the NCDs are going to be calculated and signed • "Preperation form SVT": NCDs are going to be calculated and signed • "NCD singing": only perform the signing for local NCDS • "Preperation from CAFD": NCDs are going to be calculated and signed

	In all three tabs, NCDs are calculated and signed NCDs for shipment in last step.
Backend-Communication (secure coding)	Call the backend (SWL-Sec2) and set the backend connection options
Selected Files (CAF/NCD) (red marking)	As soon as a CAF or NCD was added, the user can define a bootloader id for each entry.
Calculate and sign (yellow marking)	E-Sys is trying to perform the signing of the NCDs locally (offline). If the local singing is not able the online signing process is automatically performed. Therefore it is mandatory to be logged in at the Backend.
Generate Testsuite Data (yellow marking)	Generation and signing of NCD test sets as preparation for FAT or AmTS coding tests. A ZIP archive is created containing the following data: For Coding 2 • FA used • NCDs signed with FAT-VIN • NCDs signed with FA-VIN • shipping NCDs signed with FA-VIN • If the CAF contains ASIL functions, an NCD with a modified coding value is calculated for each safety-

	relevant function and signed with FA-VIN. <u>For Coding 3</u> • FAs and FPs used • 4 NCD variants that have been calculated and signed differently • shipping NCDs signed with Default-VIN • If the CAF contains ASIL functions, an NCD with a modified coding value is calculated for each safety-relevant function and signed with FA-VIN.

<u>NCD preparation form CAFD</u>



The available CAFDs are located at the used psdzdata.

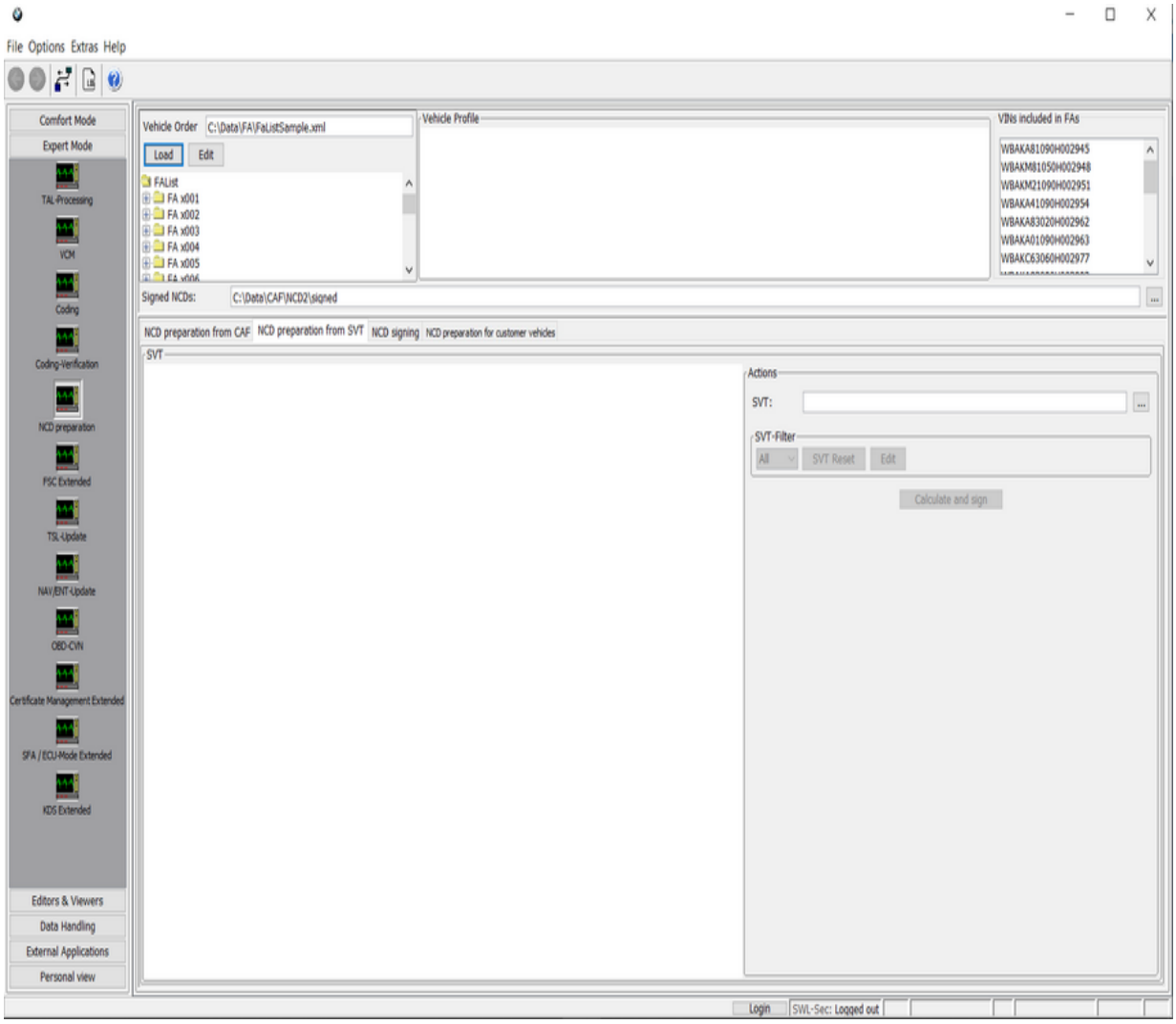
By the action buttons (blue marked) can the CAFDs be added or removed from the list of CAFDs, which are going to be signed.
The refresh button (green marking) is reloading the filed location of the calculated CAFDs. It has the side-effect that all CAFDs, which are

added and located in these order are been removed automatically.

Following steps have to be fulfilled for activating the "Calculate and sign" button:
- A vehicle-order file with only one entry has to be loaded. This entry will be automatically activated when loading the file.
- As an alternative for loading a single vehicle-order file, one or more vehicle-orders have to be selected in the tree view.
- All added CAFDs have a bootloader-ID.

The bootloader-ID can either be entered manually, in these case the base variant 'All' has to be chosen (default value). Or the specific base variant can be selected in the drop down list, in these case are only the BTLD-ID chose able which are defined for the chosen base variant.
The signing is going to be performed offline/online and saved at the defined location.
Subsequently, NCDs for shipment are calculated and signed and saved as well.

NCD preparation form SVT



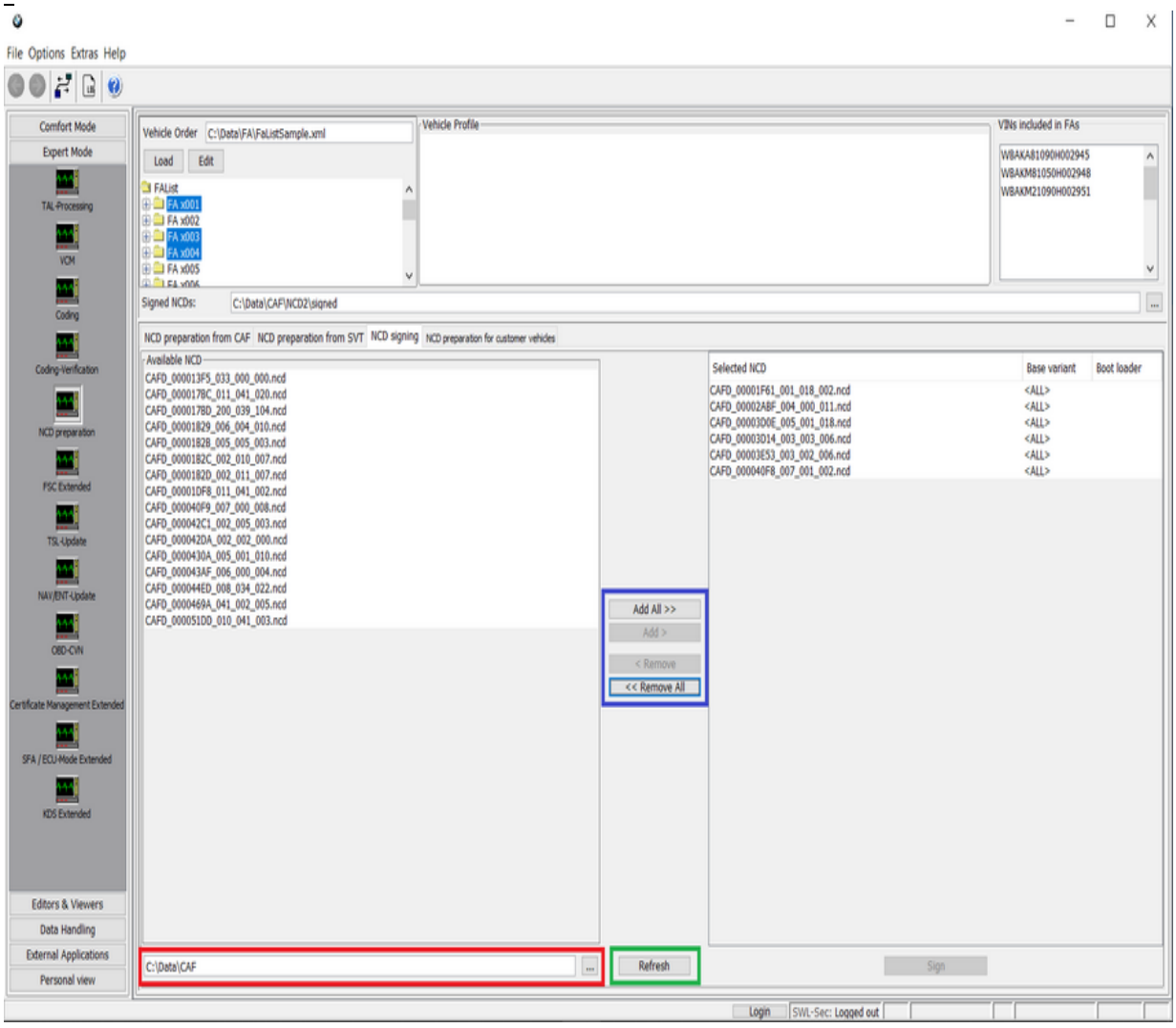
After a SVT is loaded has the user the options to mark multiple Emus or the root-element.

Following steps have to be fulfilled for activating the "Calculate and sign" button:
- A vehicle-order file with only one entry has to be loaded. This entry will be automatically activated when loading the file.
- As an alternative for loading a single vehicle-order file, one or more vehicle-orders have to be selected in the tree view.
- At least one ECU has been selected.

E-Sys is now performing the calculation and signing of the chosen NCDs.

Attention:
The NCDs can only be calculated, if the ECU has a valid CAFD (CAFD_ffffffff_255_255_255 is in these aspect a non valid CAFD). The CAFD is required to be located at the used psdzdata. Subsequently, NCDs for shipment are calculated and signed.

NCD signing



In the red marked row can the user set the filed location of the already calculated NCDs (NCDs can be calculated at Coding-Verification).

By the action buttons (blue marked) can the NCDs be added or removed from the list of NCDs, which are going to be signed.
The refresh button (green marking) is reloading the filed location of the calculated NCDs. It has the side-effect that all NCDs, which are added and located in these order are been removed automatically.

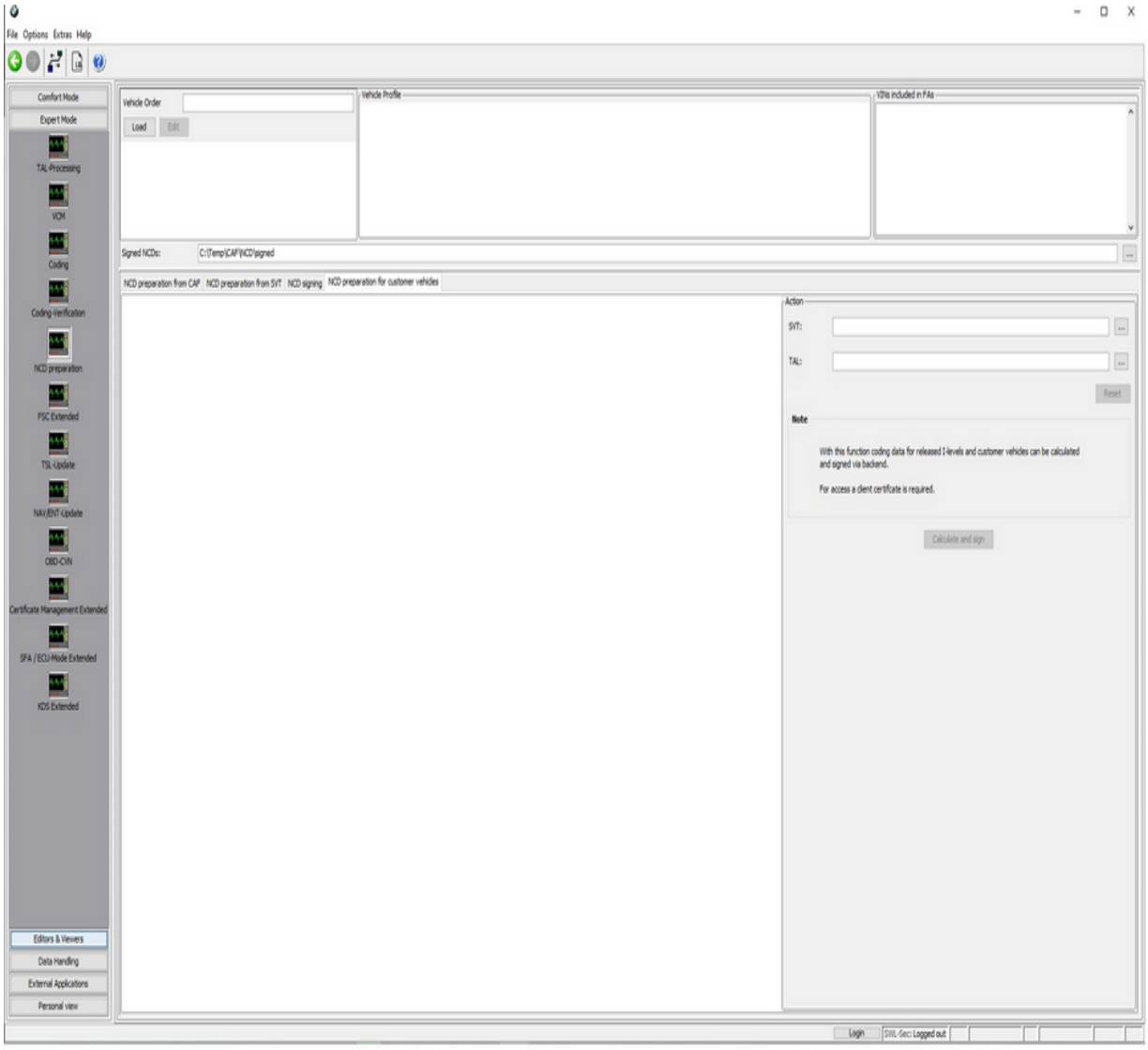
The signing button can only be activated, if following preconditions are fulfilled:
- A vehicle-order file with only one entry has to be loaded. This entry will be automatically activated when loading the file.
- As an alternative for loading a single vehicle-order file, one or more vehicle-orders have to be selected in the tree view.
- At least one ECU has been selected.

The bootloader-ID can either be entered manually or it can be selected from the list after selecting the basis variant name.
The signing is going to be performing (offline/online) depends if all necessary data are local available.
Subsequently, NCDs for shipment are calculated and signed.

Hint:

- Currently, NCD-data for Coding 2 and Coding 3 can be signed in this module. NCDs for Coding 1 are not supported.

NCD preparation for customer vehicles



Following steps have to be fulfilled for activating the "Calculate and sign" button:
- A vehicle-order file with only one entry has to be loaded. This entry will be automatically activated when loading the file.
- As an alternative for loading a single vehicle-order file, one or more vehicle-orders have to be selected in the tree view.
- A SVT or a TAL have to be selected.

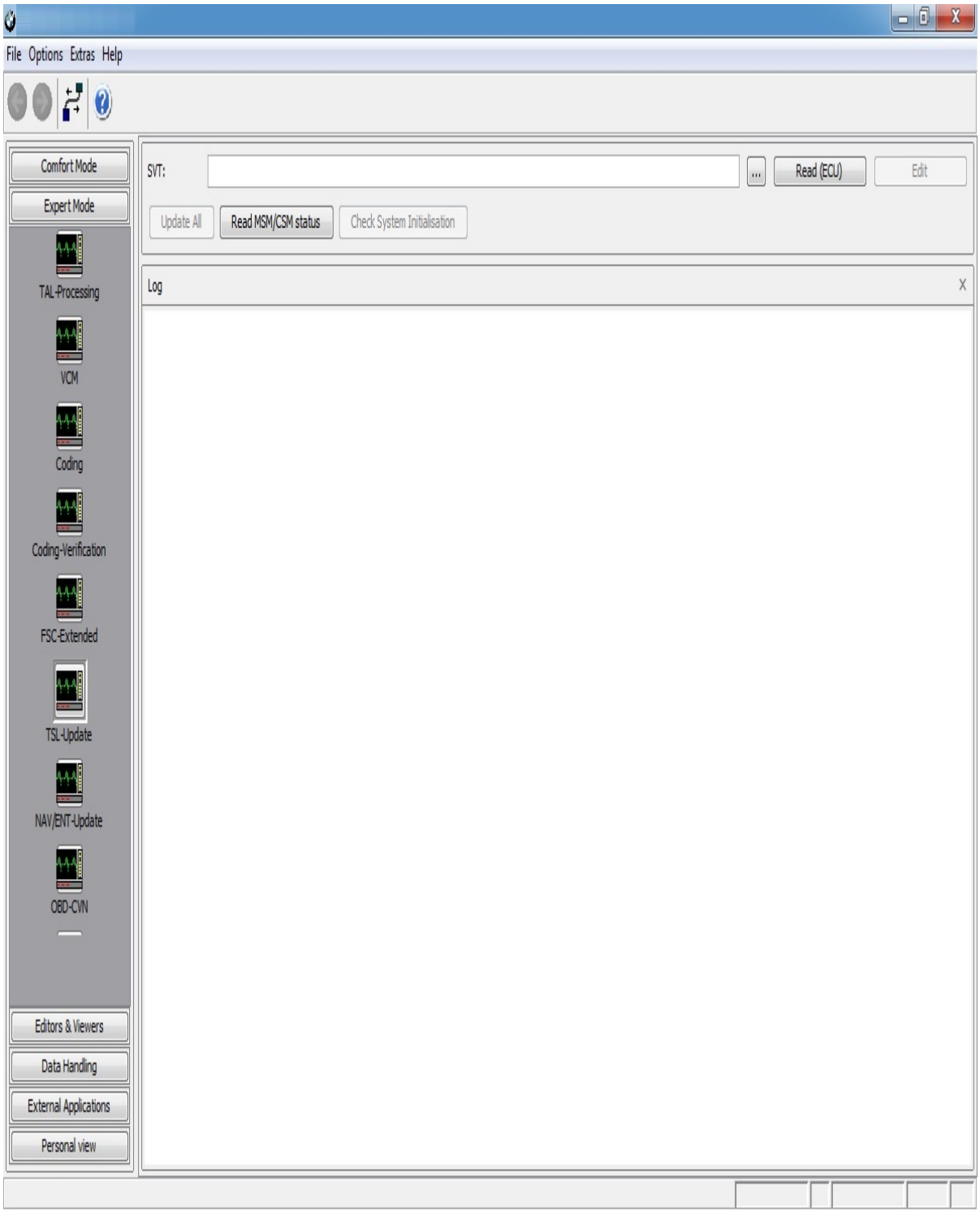
E-Sys is now performing the calculation and signing of all NCDs for the CAFS contained in the SVT or the TAL..

Hint:

- The calculation and signing of the NCDs will be done in the SCB backend
- As a necessary precondition for the access to the SCB backend, a TIS-client Certificate has to be imported first
- The URLs used for connecting to the SCB have to be defined in the esys.properties file by using the property key `scb.backend.url`
- In cases where more than one URLs are available, the complete list of URLs has to be assigned to the property key in a comma separated manner

TSL Update

In a vehicle there is a MSM (Master Security Module) and several Emus with CSM (Client Security Module). Putting or updating of transport key into MSM using PSdZ is called transport key update (Transportschlüssel-Update (**TSU**)). This is the job of the present mask.



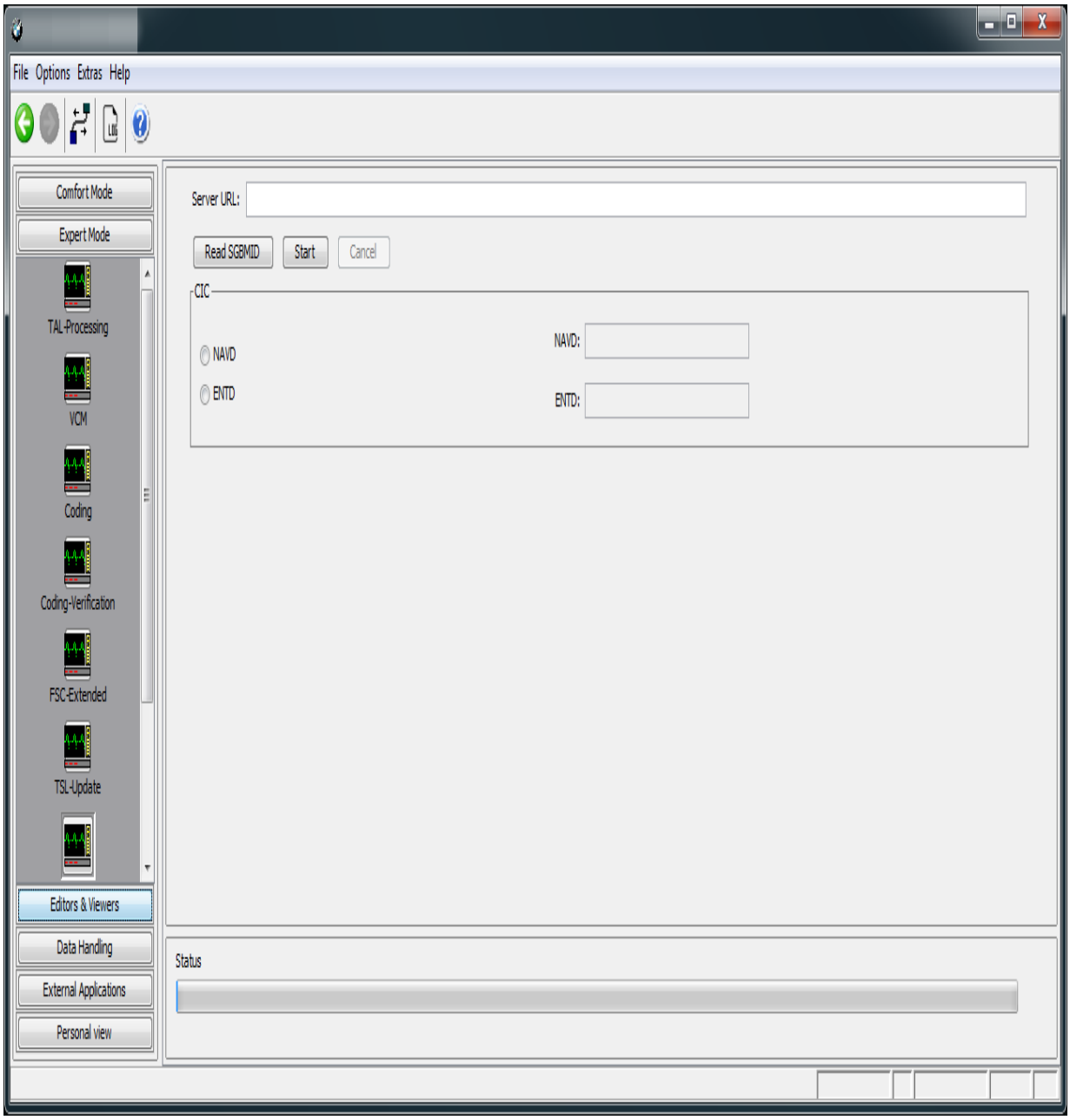
GUI Elements

SVT	The actual system installation table to be loaded by three-dot-button-dialogue.

Read (ECU)	Functional read out of the discrete Emus.
Edit	Change to SVT-Editor and show the opened SVT
TSL-Update	Execute the TSL update.
Read MSM/CSM status	Reads the status and writes the result (or error messages) to the Log view.
Check System Initialization	Execute PSdZ method `msm_F25_checkMsmSystemInitialisation.`

NAV/ENT Update

Navigation and entertainment data (NAV, ENT) are placed on a hard disk drive (HDD) in the Head unit. These data can be updated using the module NAV/ENT Update.



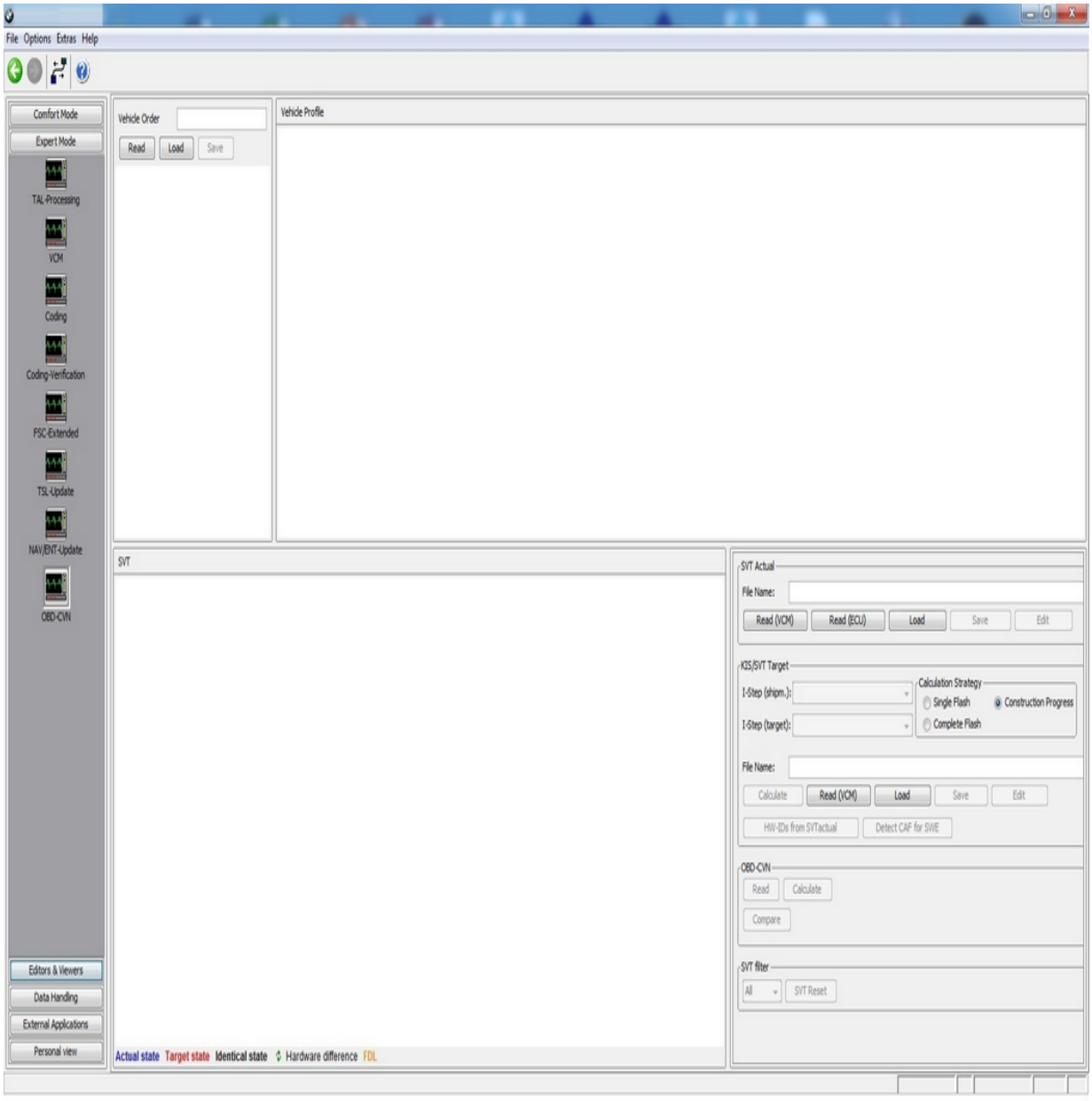
Current Data

By pressing "Read SGBMID" you can see the versions of the current navigation and entertainment data. Therefore the current SVT is read from VCM and the SGBMIDs with process classes NAVD and ENTD are shown.

Update
After writing the address of the server, where the newest navigation and entertainment data are stored, into the appropriate text field, the update process can be started by pressing the "Start" button. But the data will be updated only, if the appropriate process class is selected. By pressing the "Cancel" button the update process could be aborted.

OBD-CVN

The module OBD-CVN reads CVNs (Calibration Verification Number) from Emus, calculates CVNs over all OBD (On-Board Diagnostics) relevant SWEs.



The SVTs recorded in red letters in the picture above indicate the reference system state of an ECU. The blue ones indicate the actual

state of the ECU. Green entries indicate SVTs with identical reference and actual state.

GUI Elements

Vehicle Data	
Vehicle Order	File containing a list of vehicle orders. For selecting a single vehicle order, this order needs to be activated by using the context menu item ‘Activate FA’. If the list of vehicle orders contains only one element, this element will be automatically activated as soon as the file is loaded.
Read	Read out the actual vehicle order from the vehicle.
Load	Open a vehicle order from the file system.
Save	Opens a file dialog to save the vehicle order.
Vehicle Profile	The Vehicle profile will be automatically generated from an activated Vehicle Order and displayed in a tree structure.
SVT Actual	
Read (VCM)	read out the vehicle order from the vehicle.
Read (ECU)	read out the vehicle order functionally
Load	Open a vehicle order from the file system.
Save	Opens an file dialog to save the SVT.
Edit	Change to the SVT-Editor and open the SVT
KIS/SVT target	
I-Step (shipm.)	Available shipment integration steps
I-Step (target)	Available target integration steps
Calculation strategy	Possible strategy to calculate the SVT (Single Flash | Complete Flash | Construction Progress).
File Name	The path of the SVT file.
Calculate	Calculates an SVT-Target by integration step and vehicle order (and SVT-Actual if exists).
Read (VCM)	Read SVT-Target from VCM
Load, Save	Load/save an SVT-Target from/to file system.
Edit	Edit in the corresponding editor.

HW-IDs from SVTactual	HWEL, HWAP und HWFR elements from SVTactual will be copied to the Emus in the target SVT.
~~Detect CAF for SWE~~	*not supported yet*
Coding	
Read	Read CVNs of selected Emus.
Calculate	Calculate CVNs from the Sub-CVNs of SWEs which are OBD relevant.
Compare	Compare calculated and read CVNs.
Filter	
Filter	filter SVT tree by selected process class
SVT Reset	reset SVT/SVK-Ist/-Soll and delete SVT tree

Reading Vehicle Order and VIN

After reading the vehicle order the VIN will always be read too and put into the vehicle order (FAList/FA/FZAuftrag/Header/Vinlong). The vehicle order and the VIN will be read from the master ECU. If an error occurs they will be read from the backup ECU. In the status bar is shown where the vehicle order and the VIN were actually read from.

ATTENTION

The functionalities provided in this module which include direct server access require to have several preconditions to be fulfilled:

- The system which runs E-Sys must be in a BMW network segment in which the backend systems (CBB) are available.
- There must be a valid E-Sys client Certificate imported into E-Sys.
- At least one URL of a valid backend system (CBB) must be configured in E-Sys.

A more detailed set of instructions can be found here: HOWTO (only available within BMW network).
For further questions please contact the support service: support-security-systemfunktionen@bmw.de

Introduction:

As an enabler for new functions and to increase the security of existing functions, individual Certificates are required in the treatment of vehicles. From SP2018, individual Certificates and their binding to the respective vehicle must be installed both in the factory and in the service. In addition, bindings between ECU devices must be distributed. This functionality is provided by this module.

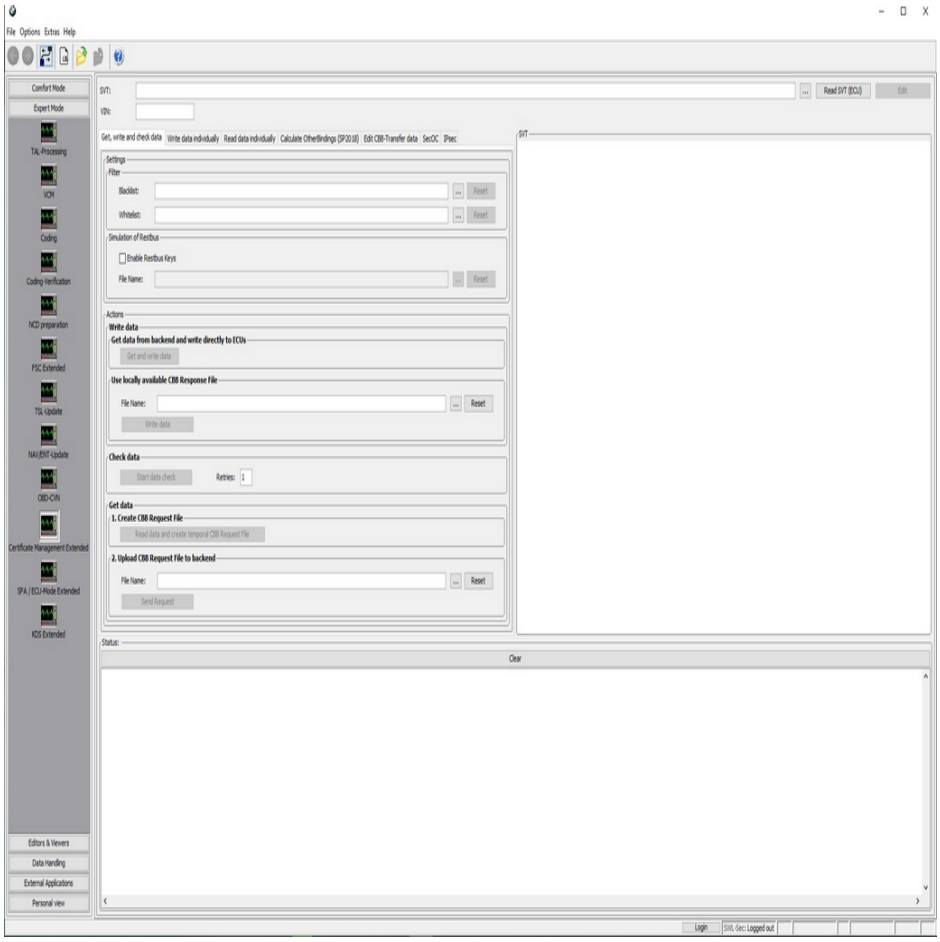
Overview Certificate Management Module:

The Certificate management module has 5 tabs. The individual functions are explained in the following sections:
1) Get, write and check data
2) Write data individually
3) Read data individually
4) Calculate otherBindings
5) Edit CBB-Transfer data
6) SecOC
7) IPSec

1) Get, write and check data:

Supports beside the functionality of the Comfort Mode the possibility of using black- / white-lists.

Format of the blacklist:	0x63,0x61,0x60,... (comma-separated diagnostic addresses in hex format)
File-Type:	.txt (e.g. blacklist.txt)



It is possible to use either a loaded file as white-list or just click at the Emus you want to treat (in the svt view on the right).
ATTENTION: It is not possible to have a loaded filter file and selected Emus in the svt view.

If no filter is selected all Emus will be treated.

Simulation of residual bus:

Checkbox 'Enable residual bus Keys' allows the storage of symmetrical keys for the ECUs in plain text.

With this option selected, the keys will be stored into XML-files.

Name and path for the XML-files can by defined by the user.

Start data check:

Performs the Certificates check of the Emus. The result is logged in a local result-file as well as displayed in the svt view.

Create a local CSR-file:

Generates an [JSON] request-file that can either uploaded in the B2B-portal to get the response or send with E-Sys to the backend (E-Sys only possible in BMW-Intranet).

2) Write data individually

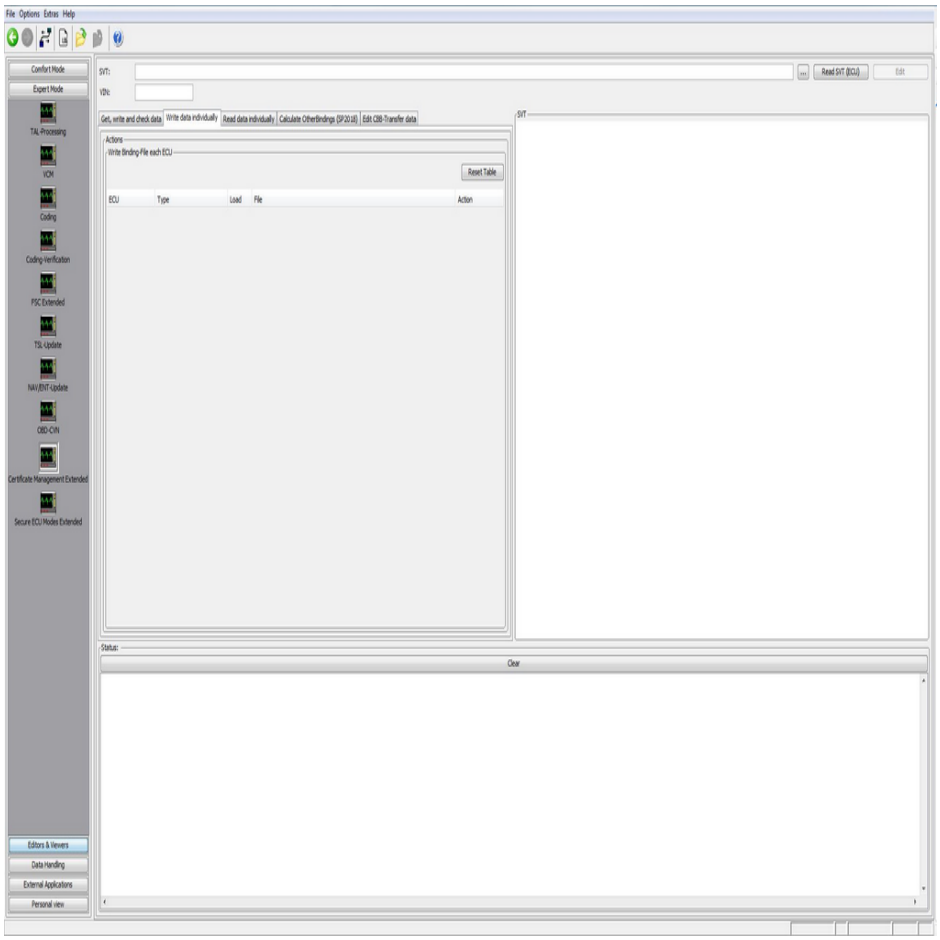
The user has the possibility to write one memory type in particular.
In order to use the function, a valid XML file containing the type of memory object selected in the table must be specified for the respective ECU in the table.

ATTENTION: The data format of the loaded object is xml and not JSON. The response file can't be used in this slide.

Each ECU can only be written with one memory type (Type1, 2, 5) at once.
In the table on the left side has the user the possibility to

Description of types:

Certificates / CSRs (Type 1) : 0
Binding (Type 1) : 1
OtherBindings (Type 1) : 2
Certificates / CSRs (Type 2) : 3
Binding (Type 2) : 4
Sec_Oc_Keylist: 5



3) Read data individually:

Reads the selected Certificate type from the configured ECU devices and displays the result in the status window.

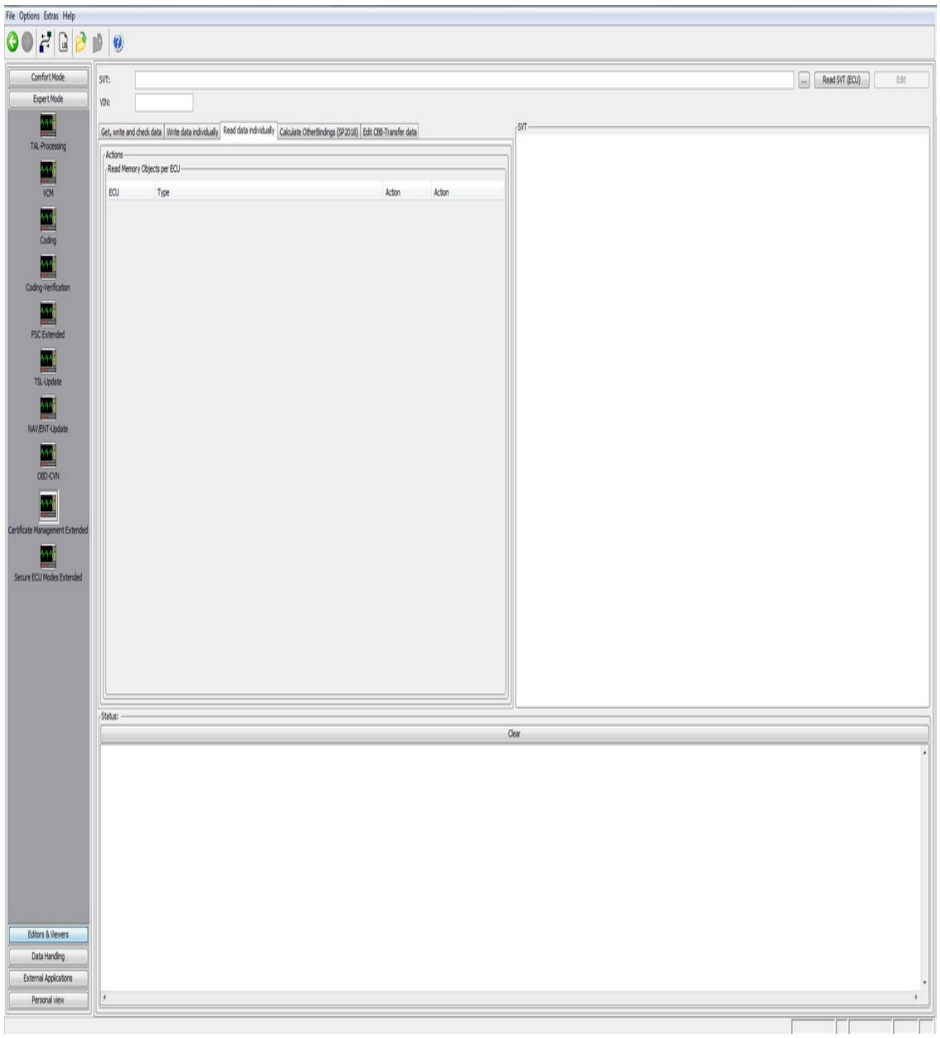
Generates for each ECU to be handled a file in XML format, which contains the entire Certificate container of the respective ECU device. Default storage location is the CERT directory in the E-Sys Data directory.
In the selected storage location, one or more XML files are now dependent on the selected memory object type
For each ECU can only be read out on memory type at the time.

Example of stored XMLs:

" BindingContainer_<Basisvariante>_<DiagAdr[Hex]>_fromVehicle_<Zeitstempel>.xml
"

"

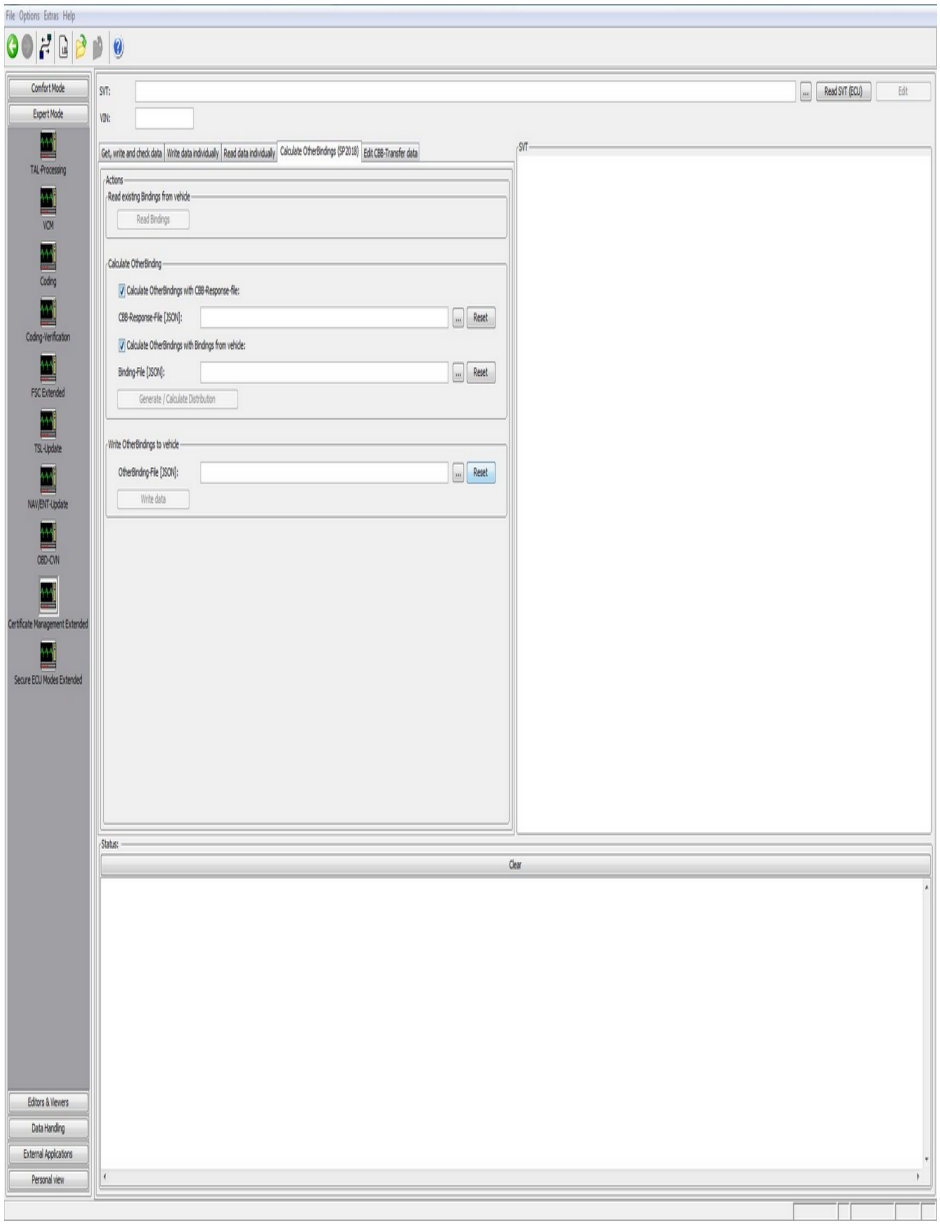
CertificateContainer_<Basisvariante>_<DiagAdr[Hex]>_fromVehicle_<Zeitstempel>.xml
"



4) Binding Distribution: (only SP2018 Emus)

A binding distribution can be created in this tab.

The Binding Distribution contains the information about which Emus are allowed to establish a secure communication with each other.



The "Calculate Distribution" button calculates the OtherBindings, which contain the information which ECU devices may communicate with each other in the vehicle.
For this purpose, the user selects a location for the BindingDistribution file. Default storage location is the CERT directory in the E-Sys Data directory.
Here, the user has the possibility to influence the calculation of the distribution with the check boxes to the right of the files to be loaded.

Both check boxes active: PSdZ calculates the OtherBindings by using the bindings of the backend

with the bindings from the vehicle for calculation.

Checkbox "from backend" active and "from vehicle" inactive: PSdZ does not get any readout binding set from the vehicle to the calculation.

Checkbox "from vehicle" active and "from backend" inactive: PSdZ will only allow the bindings from the vehicle to be included in the calculation.

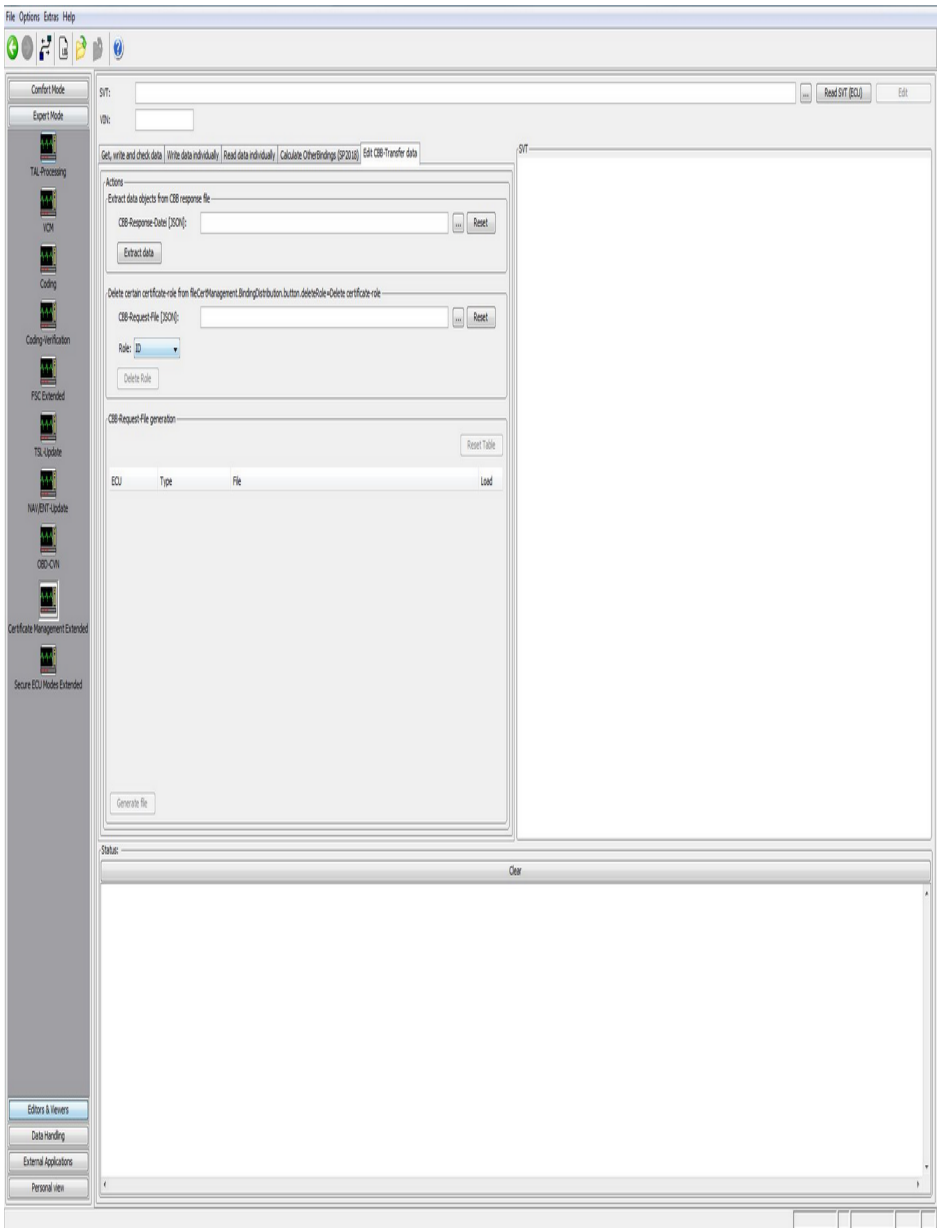
Both check boxes inactive: This combination does not provide a result since no bindings are passed to PSdZ.

[Note: If BindingContainer is available from the CBB box and the vehicle for the same control unit, the newly calculated container from the CBB-Box is always used again. The container determined from the vehicle is discarded. **]**

5) Edit CBB-Transfer data:

The user has the possibility:

- extract individual memory objects of the loaded JSON file and store them in a xml file
- deleting individual Certificate roles from a CSR file
- generates the request file with the loaded xml files



Extract xml files of a JSON:

1. Load JSON file
2. Button Extract data

Delete role IDs:

1. Load request file [JSON].
2. Chose role that you want to delete.
3. Button Delete-Role

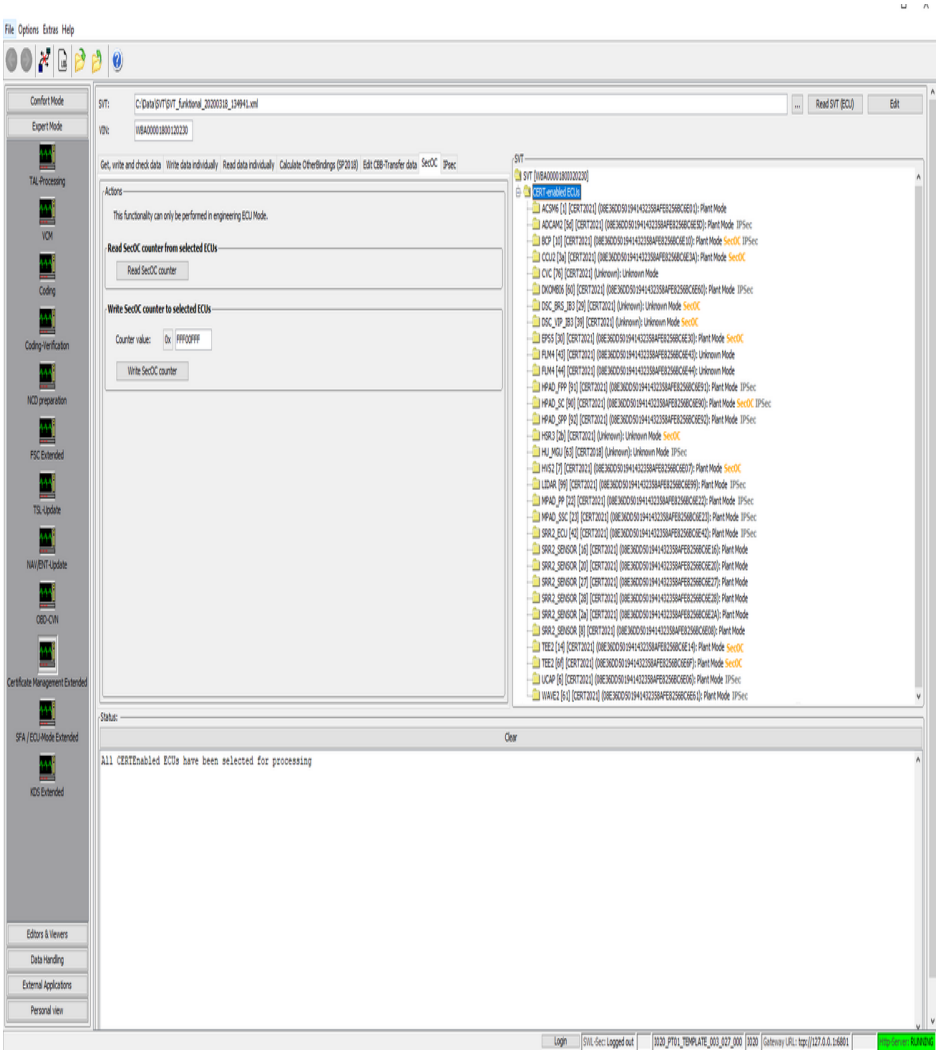
Creating a CSR file:

1. Load ore read svt.
2. Load all the memory object [xml] in the table
5. The "Create file"

6) SecOC:

The user has the possibility:

- read out the current SecOC counter value of all selected Emus
- set a new SecOC counter value to all selected Emus

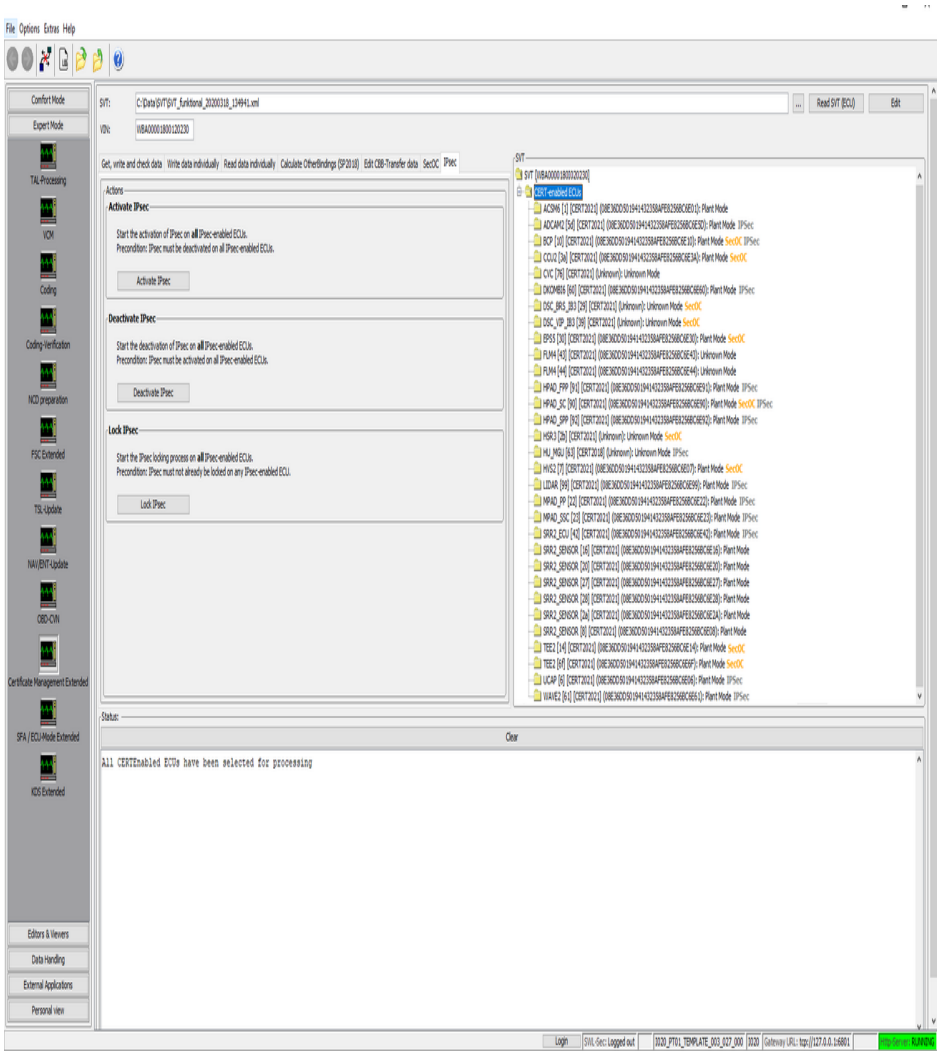


Notes:

- The following command can only be performed on Emus that support this functionality.
- The Emus must be in engineering ECU Mode
- Text field accepts counter values in HEX in range from 0x0 until 0xFFFFFFFF only

7) IPSec:

IPSec of all IPsec-enabled Emus can be activated, deactivated or locked in this tab.



Notes:

- The following command can only be performed on Emus that support this functionality.
- For each action, all IPsec-enabled Emus are handled. Selection in the SVT tree is not considered.

SFA / LCS / Secure ECU Modes - Extended

Introduction:

The functionalities provided in this module which include direct server access require to have several preconditions to be fulfilled:

- The system which runs E-Sys must be in a BMW network segment in which the backend systems (SFA-Backend) are available.
- There must be a valid E-Sys client Certificate imported into E-Sys.
- At least one URL of a valid backend system (SFA-Backend) must be configured in E-Sys.

A more detailed set of instructions can be found here: HOWTO (only available within BMW network).

The module "SFA / Secure ECU Mode" empowers the user to write a secure token set, switch ECUs, which supports Secure ECU Modes, between the three states (Engineering, Plant, Field), generates a response-file, delete or clear features, verify tokens with VIN, and read the current SFA-State or ECU-Mode.



Difference to the Comfort Mode:

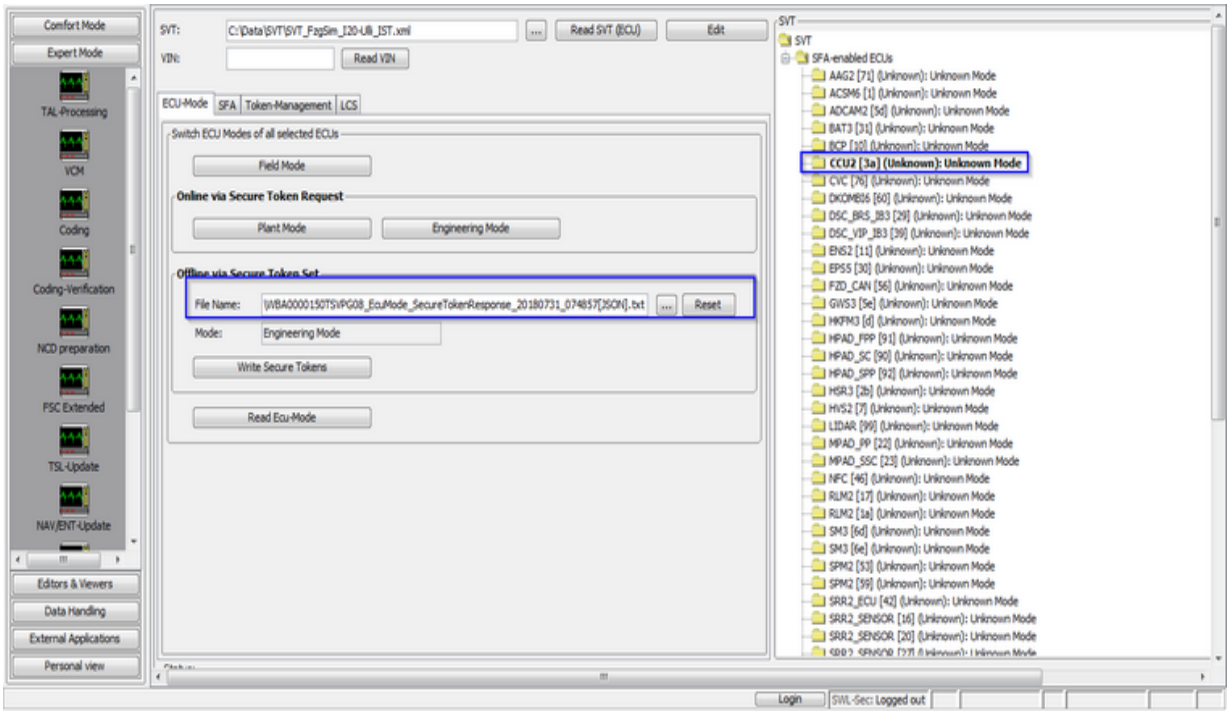
In the extended mode the ECUs which shall be treated can be selected in the tree view. There is single or multiple selection of Emus possible.
To switch back to a treatment for the entire vehicle, just click on the root element in the tree-view.

1) ECU-Mode:

The Modes Plant and Engineering are only accessible with a valid token an the ECU has to be in den Field-Mode.

To switch back into the Field-Mode is no token necessary.

All ECUs that are contained in the svt are visualized on the right side in the svt-tree-view.



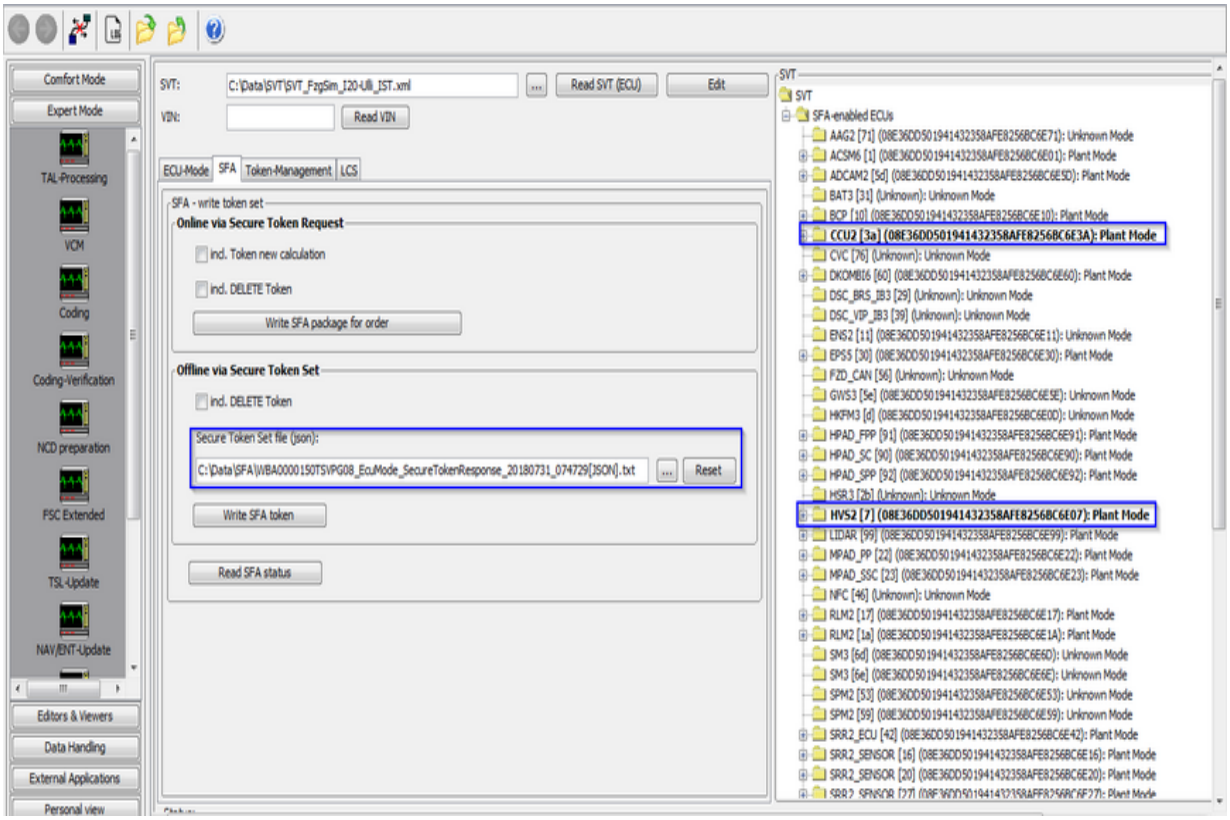
Use Cases:

Available only inside BMW network		
	Switch Mode with Token Request: Plant Mode	Generates a Plant-Mode-Token for all Emus, that provides ECU-Mode, sending the request to the Backend and writing the response at the Emus. The Emus are finally in the PlantMode
	Switch Mode with Token Request:	Generates an Engineering-Mode-Token for all Emus, that provides ECU-Mode, sending the request to

	Engineering Mode	the Backend and writing the response at the Emus. The Emus are finally in the EngineeringMode
	Upload Secure Token Set Request File	Sending the request to the backend and store the response at "DATA\SFA"
Available outside BMW network		
	Generate Request file	Generates a request-file for the chosen ECU-Mode for all Emus of the svt.
	Delete Token	Deactivate the chosen ECU-Mode token. ATTENTION: The token can't be used a second time.
	Clear Feature	Delete the chosen ECU-Mode token from all Emus. ATTENTION: The Emus have to be in the Engineering-Mode. After the process the deleted token can be used a second time.
	Write Secure Token Set	Write all secure tokens the loaded response file, all Emus that are part of these file are marked in the tree-view (blue marked at the illustration). Depends on the ECU-Mode in the chosen file Engineering Mode, Plant Mode, Mixed (Engineering and Plant tokens are included) or Undef / unknown (beside the ECU-Mode-tokens are also other feature-tokens part of the file) are visible.
	Switch into Field Mode	Sends the job to switch into Field Mode to all Emus.

2) SFA:

SFA is the replacement of SWT from Service-Pack 2021 on.



Use-Case:

Available only inside BMW network		
	Write SFA Target (package for order)	Generates at the backend a new secure token set. In case of new calculation is active the tokens for the already active features, that should be active based on the VIN, are going also to be generated. If the user wants only the features active, that are calculated to be active by the backup-System, "incl. DELETE Token" has be active. These means that it is possible that features which was

		active before are now deleted. To reactivate these features a new generated token is required.
Available outside BMW network		
	Write a local secure token set	Write the loaded secure token set. Attention if the inclusion of surplus tokens is active. With these option all features, that are not part of the token set or the target status are going to be deleted. To reactivate these features a new generated token is required. An example of on surplus toke is an active ecu-mode token. All Emus that are part of the token set are marked in the SVT-treeview. (blue marked in the above illustration)
	Generate SFA-production-status Request	Generates a SFA (secure Token-Request set) for the SFA-production status.
	Read SFA-Status	Reading the SFA-status of the choose Emus or the entire vehicle.

Color representation of SFA features in the SVT tree view:
In order to determine the status of all SFA features (activated, deactivated etc.) for each ECU selected in the SVT tree view, the button “Read SFA status” must be activated.
This triggers communication with each selected SFA-capable ECU and its features are read out.
The SVT tree structure is then updated in such a way that each ECU node can be opened up and all available features can be displayed. Each feature is highlighted in color in the SVT tree view.

Color markings mean:

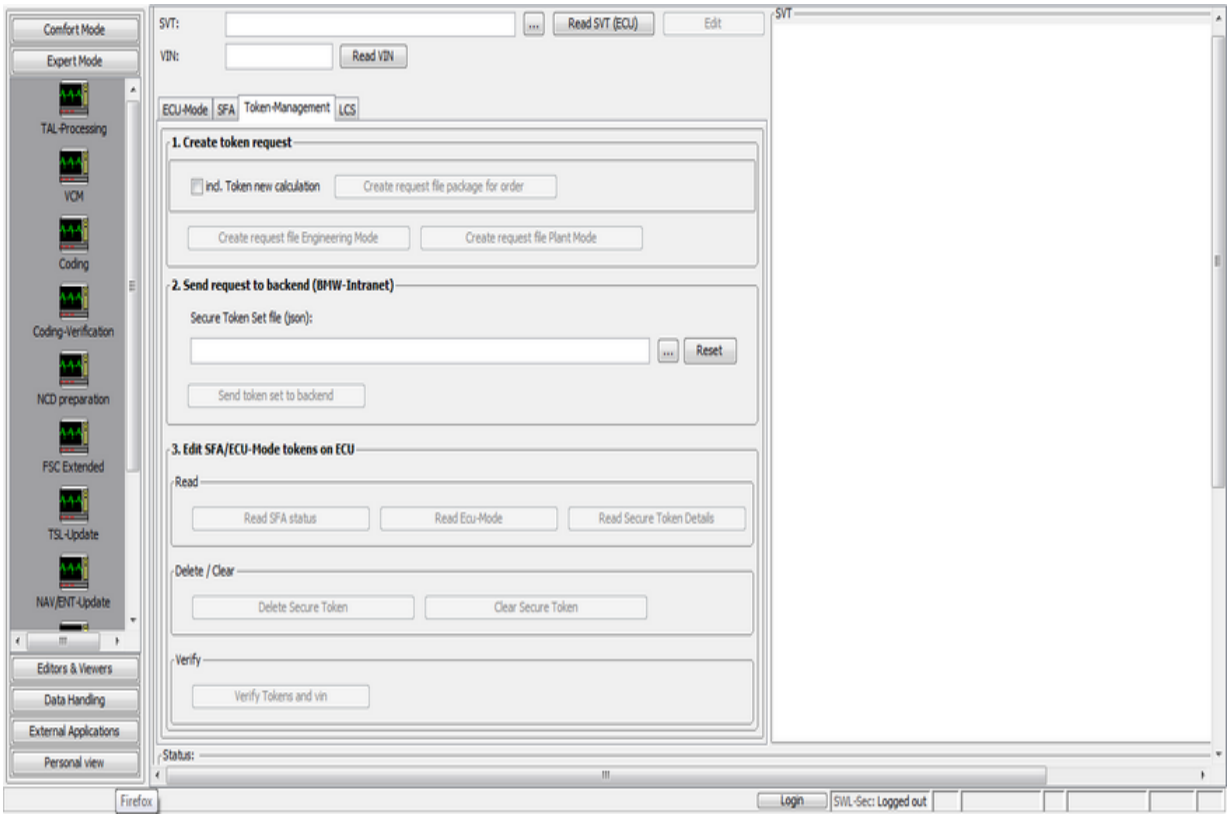
Green - the Feature_ID is **activated**

Orange - the Feature_ID is **deactivated** or **expired**

Red - the Feature_ID is **incorrect**

Black - the Feature_ID is initially **not active**, i.e. no secure token available

3) Token-Management:



Use-Case:

Available only inside BMW network		
	Upload Secure Token Set Request File	Sending the request, depends on the request-type (ECU-Mode, package for order, newest package), to the backend and store the response at "DATA\SFA". A "newest package" request has to

		have "newest" a part of its file-name.
Available outside BMW network		
	Generate ECU-Mode Request file	Generates a request-file for the chosen ECU-Mode for all Emus of the svt.
	Generate SFA-production-status request	Generates a request-file (secure Token-Request set) for the SFA-production-status.
	Delete Token	Deactivate the chosen ECU-Mode token. ATTENTION: The token can't be used a second time.
	Clear Feature	Delete the chosen ECU-Mode token from all Emus. ATTENTION: The Emus have to be in the Engineering-Mode. After the process the deleted token can be used a second time.
	Read Status / Read ECU-Mode	Reading the SFA-/ECU-mode status of the choose Emus or the entire vehicle.
	Verify Token and VIN	The system verify if all token are compatible with the VIN of the vehicle.

4) LCS:

LCS stand for "Locking Configuration Switch". In this tab the user has the option to read and change the value of a specific configuration of the chosen ECU.
The user has either the option to choose the configuration by the drop-down menu or enter the HEX-Value manually.

LCS #0 - #99 are reserved for internal ECU functions. LCS #100 - #255 are reserved for ECU-specific functions.

Currently LCS values are mapped to the following scheme:

LCS Value	LCS Name	Description
LCS#0 (0x00)	SP Switch	LCS #0 is provided for switching Service Packs (SP2018/SP2021)
LCS#1 (0x01)	SecOC Bypass	Activation is required, for example, if a vehicle has mixed ECUs with and without secure onboard communication.
LCS#2 (0x02)	TimeSupreme	System time used in diagnostics. At Release 23/03, the system time is migrated from "Kombi" ECU to the main ECU ("BCP"). That enables the switchover to the AUTOSAR time sync mechanism. The value 2 here corresponds to the switchover to the "BCP".

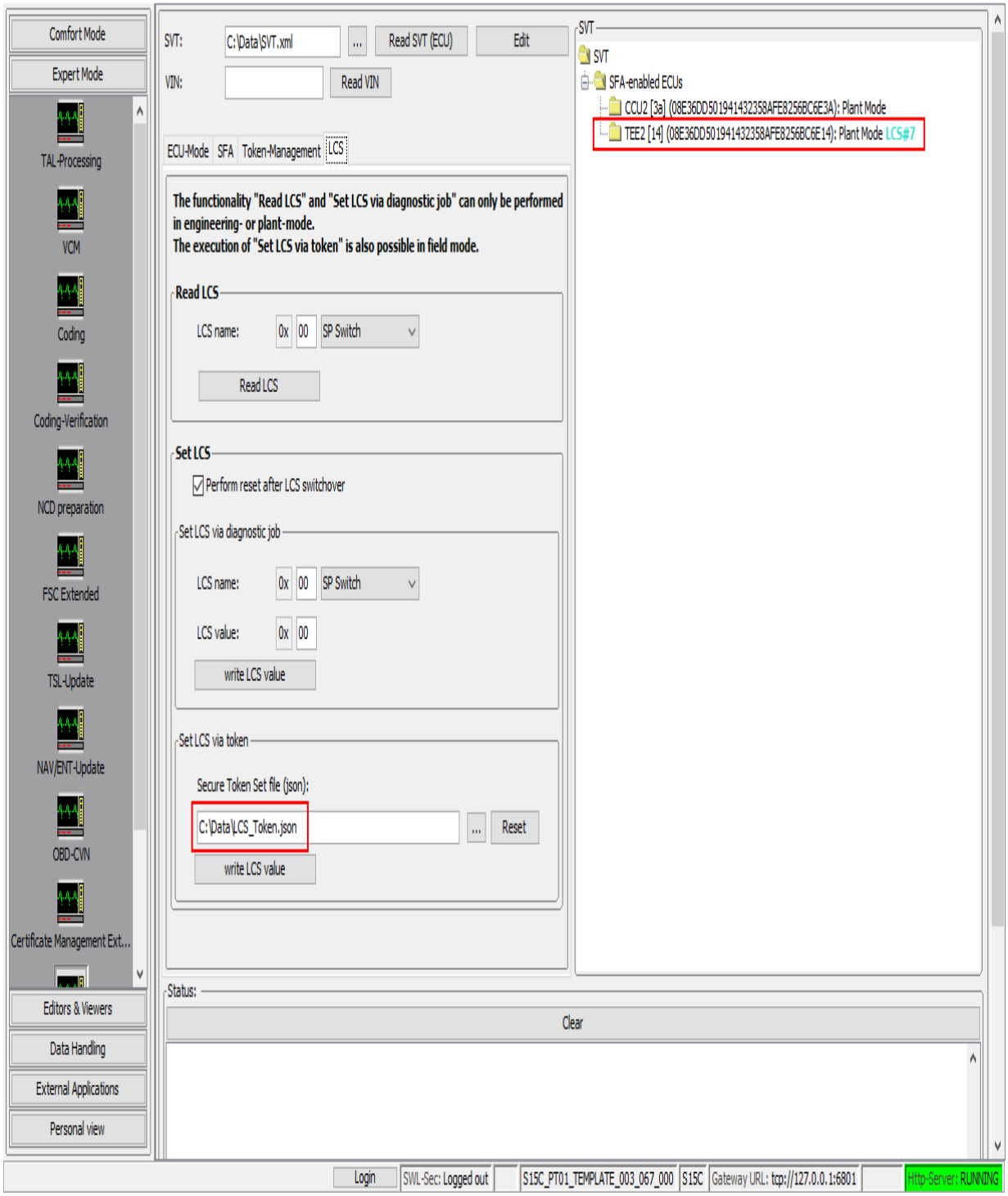
All other LCS values (0x03-0xFF) can be entered manually, but are currently not defined. For this reason, the LCS Name is displayed as "Undefined" in the drop-down menu.

Alternatively, a file containing LCS tokens can be used to perform the LCS switch over.

Notes:

- If the loaded file contains a LCS token that matches an ECU from the loaded SVT, this is marked in the SVT tree structure by a colored text (see figure below).

- After a LCS#01 switch over, an ECU reset is to be performed so that the new LCS#01 value is also implemented by the ECU.



KDS - Extended

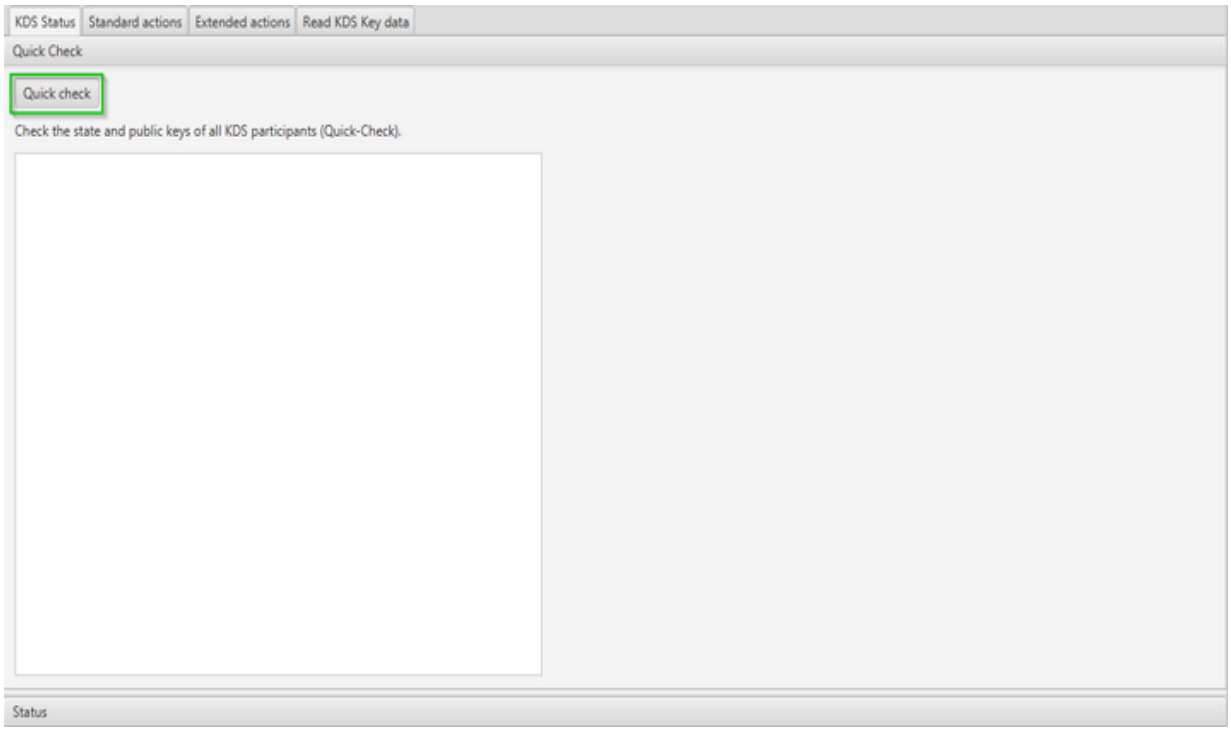
Introduction:

KDS (anti-theft protection 2.0) based on the secure token sets (see SFA / Secure ECU-Mode).
The functionalities provided in this module which include direct server access require to have several preconditions to be fulfilled:

- The system which runs E-Sys must be in a BMW network segment in which the backend systems (SFA-Backend) are available.
- There must be a valid E-Sys client Certificate imported into E-Sys.
- At least one URL of a valid backend system (SFA-Backend) must be configured in E-Sys.

A more detailed set of instructions can be found here: HOWTO (only available within BMW network).

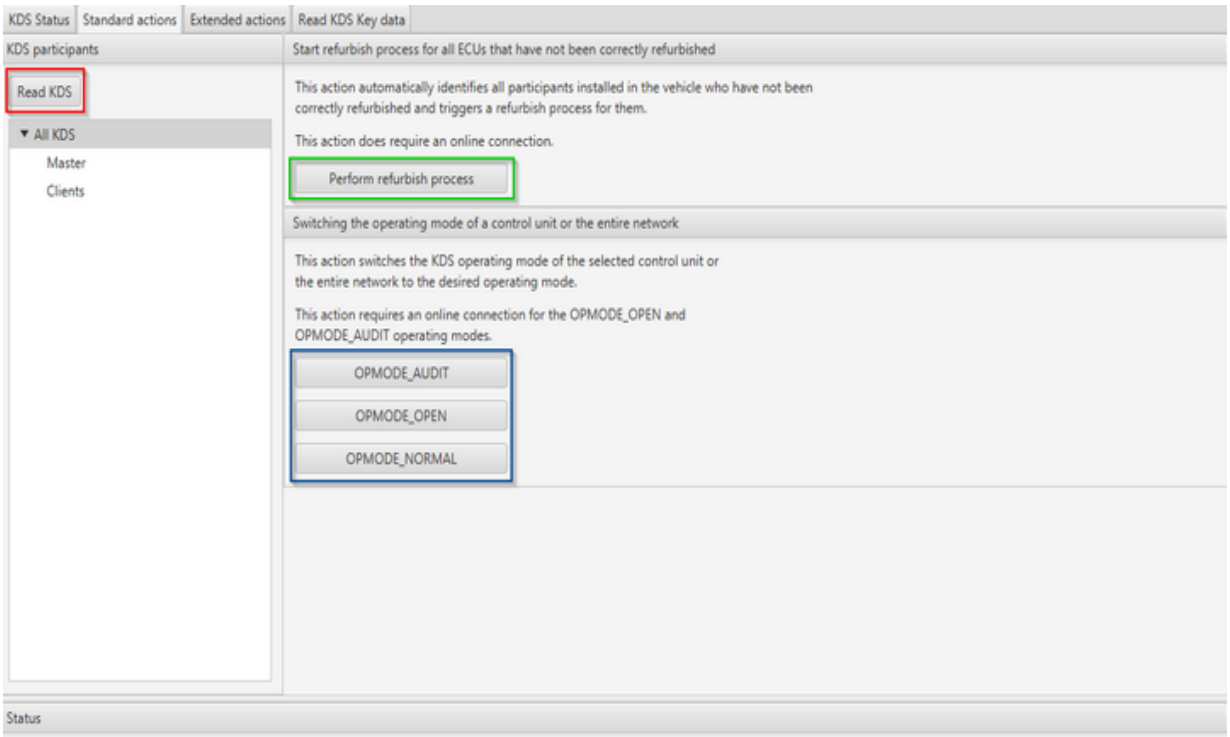
KDS status:



The button "Quick check" (marked green) starts with a KDS system check. The result is visualized in the GUI and has one of the following values:

KDS quick check results
MASTER_OK_CLIENT_OK
MASTER_OK_CLIENT_INVALID
MASTER_INVALID_CLIENT_OK
MASTER_INVALID_CLIENT_INVALID
ERROR_CLIENT_NOT_PAIRED

Standard actions:

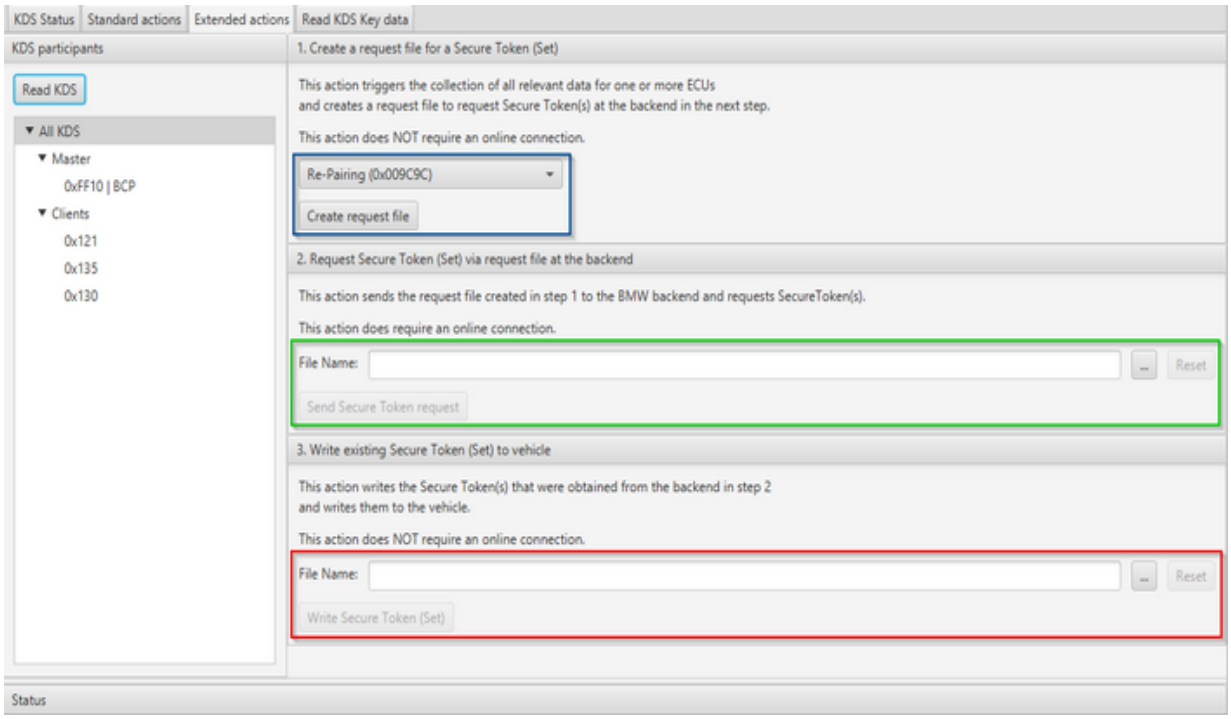


These tab can visualize the KDS system. It is separated in the KDS-master and KDS-clients. All KDS system members are by the KDS-ID.

- Read KDS (marked red): Read the entire KDS system.
- Refurbish process (marked green): **[Only with BMW-Intranet connection possible]** Identifies all participants in the KDS system, that have not been correctly paired and is pairing the entire KDS system.
- Switching operation mode: Change the operation mode for the

chose Emus. If no ECU is selected the entire KDS system is going to switch. The switch into the modes AUDIT and OPEN requires a BMW-Intranet connection and the valid client-backend-Certificate (E-Sys options / authentication)

Extended actions:



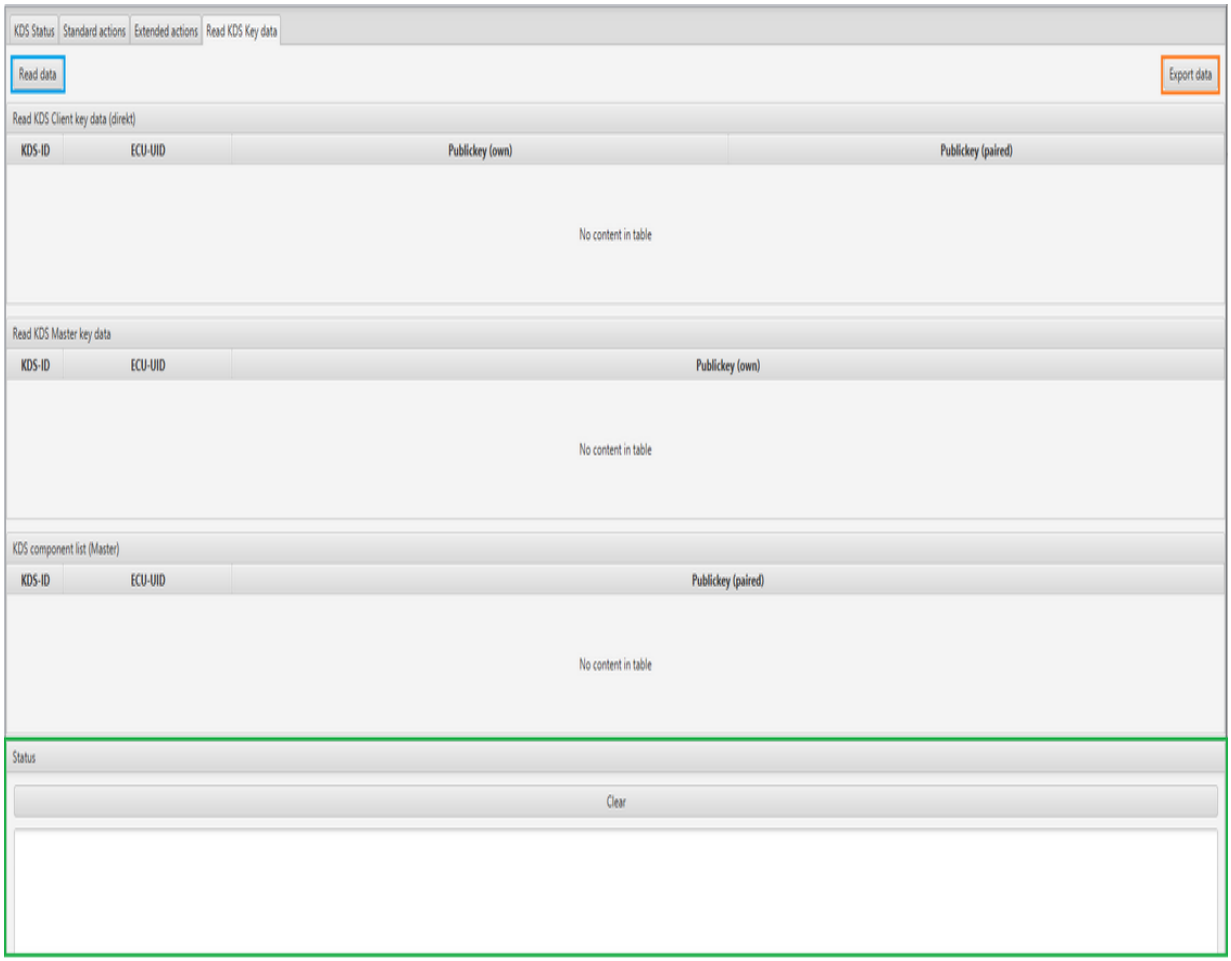
- Generates a token request file for the chosen Emus or the entire KDS-system. The request-type can be defined by the drop down menu, behind the name of the is the feature-ID displayed. (marked blue)

Type of the secure token request file	Short description
Re-Pairing (0x009C9C)	Generates a token for a re-paring (like refurbish process).
Remove client pairing (0x00DC9D)	Resets the pairing status of KDS-clients.
Remove master pairing (0x00DCC9)	Resets the component list at the KDS-master for the chosen

	KDS-clients.
OPMODE_AUDIT (0x00AAFC)	Create a request token for the selected Emus to change into the operation mode AUDIT.
OPMODE_OPEN (0x00AAFC)	Create a request token for the selected Emus to change into the operation mode OPEN.

- [**BMW-Intranet necessary**] Sending the loaded request-file to the SFA-Backend and automatically save the response (secure token set). (marked green).
- Write the active KDS secure token set (marked red)

Read KDS key data:



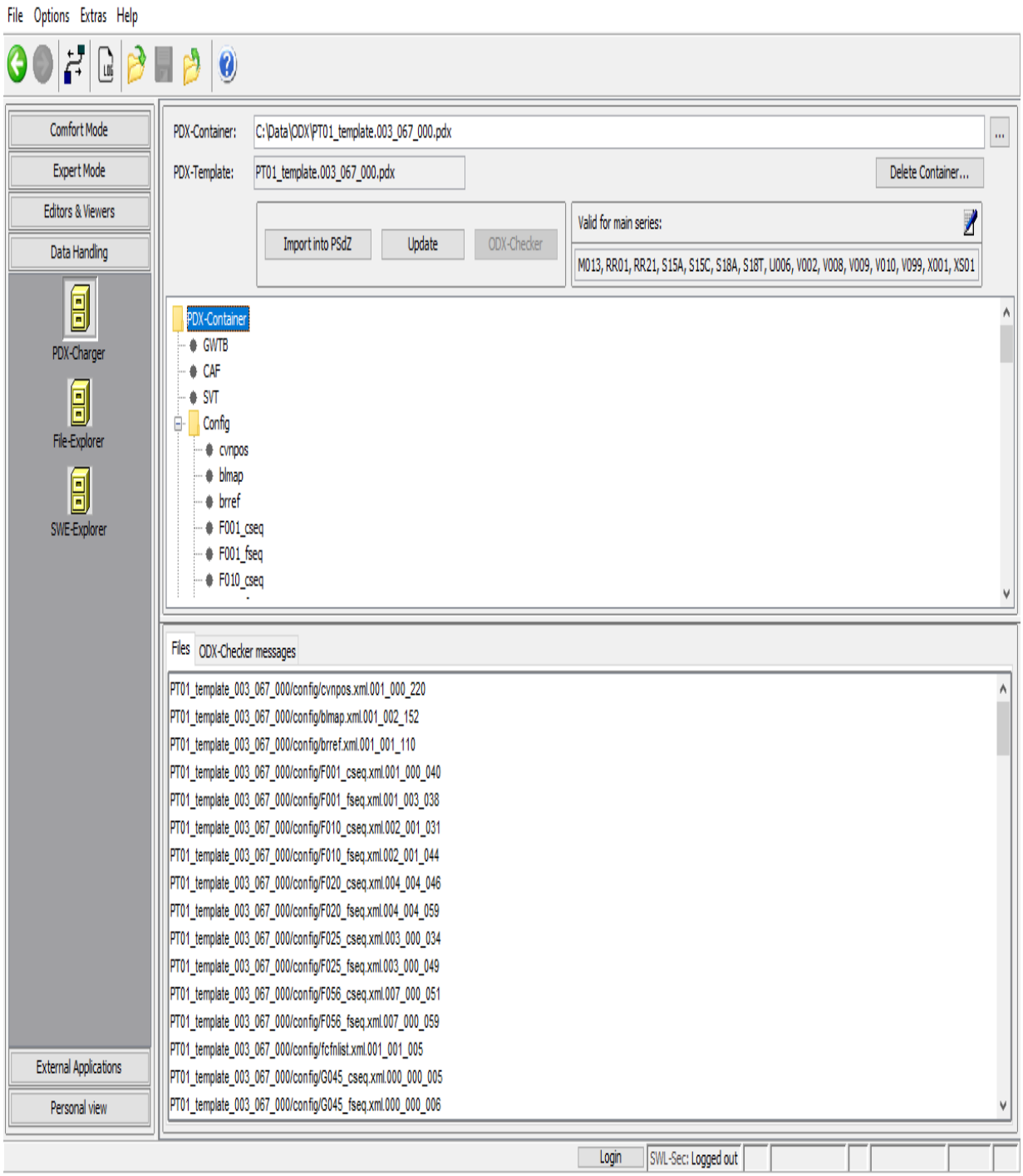
- Read data (marked blue): Visualize all KDS-ID with the corresponding publickey.
- Export data (marked orange): The results can be exported as a csv-file. The location where the cvs-file is stored can be seen in the log-view (marked green); the location is the KDS sub folder in at E-Sys_Data.

Data Handling

PDX-Charger
PDX-Update
PDX-Delete
ODX-Checker
File-Explorer
SWE-Explorer

PDX-Charger

To process a vehicle flash or ECU programming the PSdZ (or the more basic runtime system beneath) needs a couple of data files e.g. SWEs, program jobs, programming procedure, vehicle information etc. All this information are transported in MCD-2D (ODX) containers and passed.



Therefore an existing container is read from disk into E-Sys by the "tree-dot"-button.
The displayed design of the container is as follows:

Root element "PDX-	By context menu a new ECU can be

Container":	inserted here
CAF:	A (further) CAF file can be added..
SVT or SWE-Sequent:	A (further) SVT or. SWE sequence can be added.

Already existing elements of the upper parts can be removed by context menu.
After adding a ECU it's possible to append one or more SWEs to its substructure.
To an ECU exactly one base variant can be assigned. And to every base variant you can
specify bootID and diagnosis address. Furthermore it's possible to deactivate a base variant.

Ecus are displayed with blue letters if of L4 and with black letters if it is an L6 ECU.

The container's file versions are displayed in the lower window. Differently to the display in the above window (i.e. swfl_11111114) the file version gets displayed, too (swfl_11111114_001_001_002). For swe files it is possible to delete single file versions, but solely if 2 swe file versions for the same identifier exist.

After all modification is done the PDX container can be imported into PSdZ unless there is no connection to an ECU.
So in the mask above you have to disconnect before trying to import into PSdZ.

GUI-Elements

"Import into PSdZ"	Import current container into PSdZ
"..." button	load existing PDX container
"Update" button	PDX update, see correspondent help page
"ODX-Checker"	Starts the ODX-Checker, see correspondent help page
"Delete	PDX delete, see correspondent help page

Container..."	

Further Information:

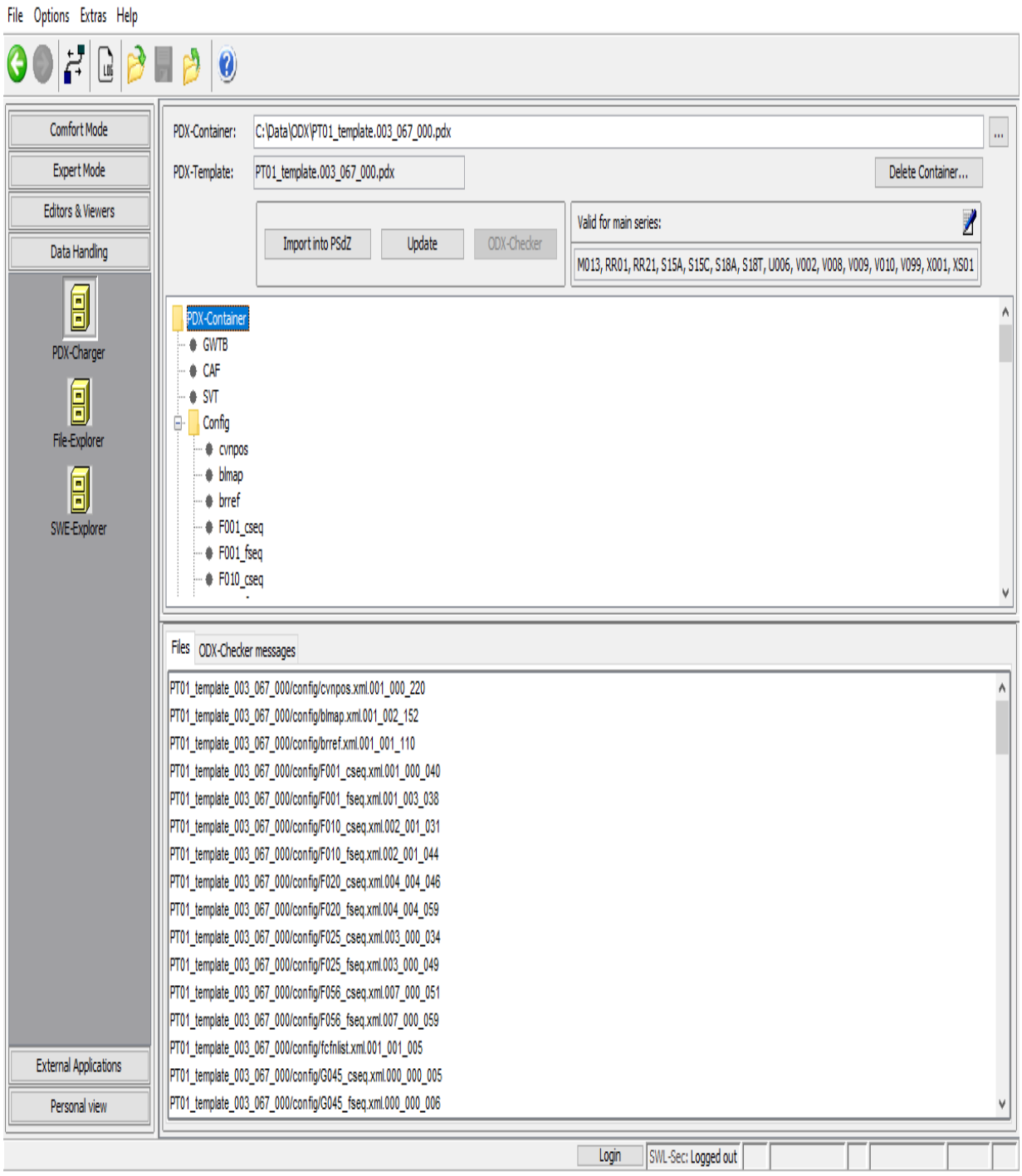
ODX (**O**pen **D**iagnostic Data E**x**change) is a standardized exchange format based on XML for data and information related to diagnosis. PDX (Packed ODX) is a zipped ODX. The PDX file only contains information. The real data (SWEs) are contained in separate files and are only referenced from the ODX file. In ODX multiple layers are defined which contain the different types of information:
The ODX-Charger performs the task to integrate the particular layers to one ODX-Container. Hereby the XML files for the layers Diag-Layer-Container, Comparam-Spec, Vehicle-Info-Spec and Multiple-Ecu-Job-Spec exist already in a file system or are delivered completed.

The flash layer is assembled by the ODX-Charger from a couple of selected SWEs.
So far the SWEs were included into den ODX-Container as MSR files. Hence the new version 2.0.0 of E-Sys also BSW files are supported as SWE. This BSW is a ZIP archive that contains the binary flash data and an XML file containing the header information.

Even if the ODX-Container could consist from a single large file it's recommended for the sake of clearness to leave the distinct layers in it's files and to associate them in the ODX file.

PDX-Update

Up to now it was only possible to import a delivered or modified (by E-Sys) PDX container into PSdZ.
In practice on-site it's common to start with a discrete HW/SW configuration and modify and expand it in the course of time of progression (I-Stufe) in the way that when new data is available still there is a wish to preserve the old data.
An amplification of the previous container by the new contents of the new container is desired.
This can be realized now by the update functionality in E-Sys.



During start of mask the update button is still disabled. Primal a container is loaded both button "Import into PSdZ" and "Update" are enabled.

As the first container the 'old' container to be updated is selected.

After pressing the update button a file select box opens and the PDX container (with the new data) used for the update can be specified.

Now E-Sys calculates the set union of the two containers and displays it. Afterwards the modified container can be saved and imported into PSdZ.

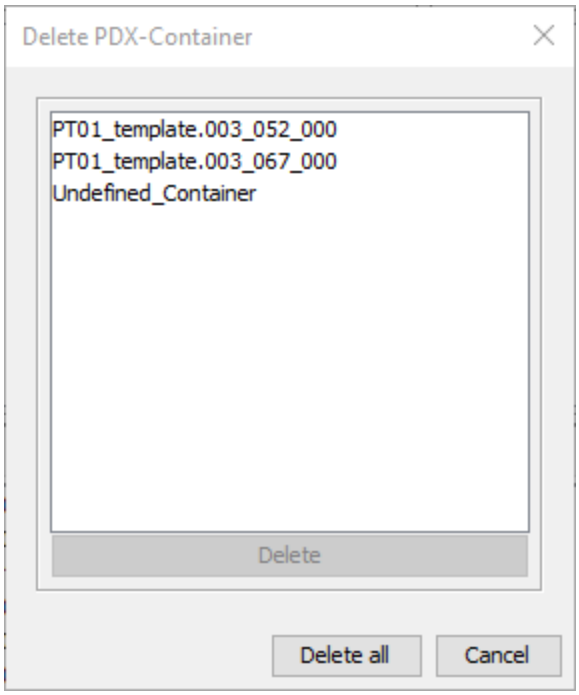
PDX-Delete

Starting with E-Sys 3.31, an index file is introduced to allow the deletion of PDX containers. This index file is updated during each import and delete process.

In the dialog 'Delete PDX-Container' the user is offered a selection menu of the PDX containers that have been imported so far.
Either individual container files or all containers can be deleted.

The deletion process is liwithed to entries in the subdirectory ***"..\psdzdata\swe"*** and ***"..\psdzdata\mainseries"***.
Only those files that are not needed by any other PDX container are deleted.

To delete individual container contents, the corresponding PDX containers have to marked in the selection list and the button 'Delete' has to be pressed.
Deleting all containers is done with the 'Delete All' button:

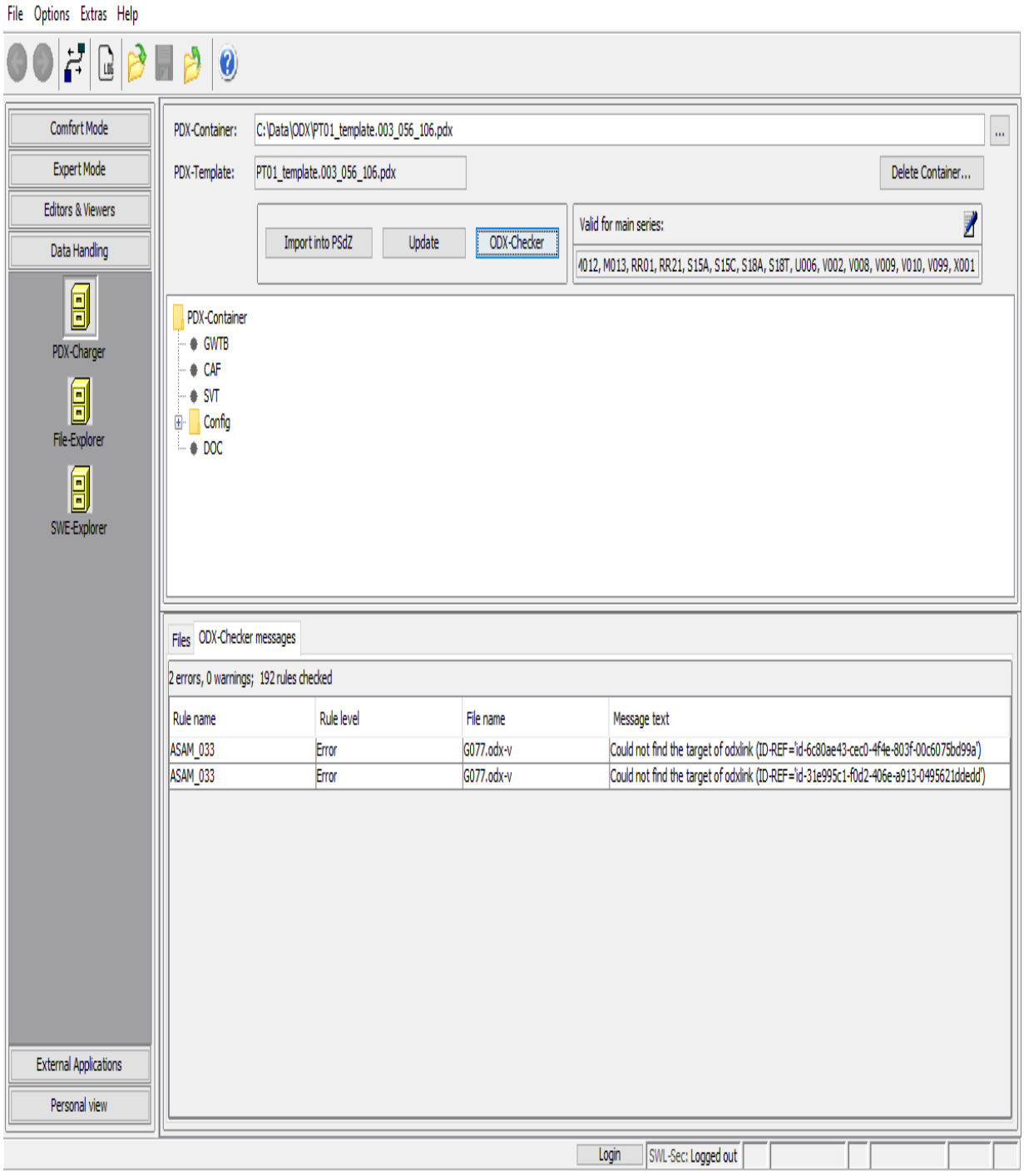


Specials:

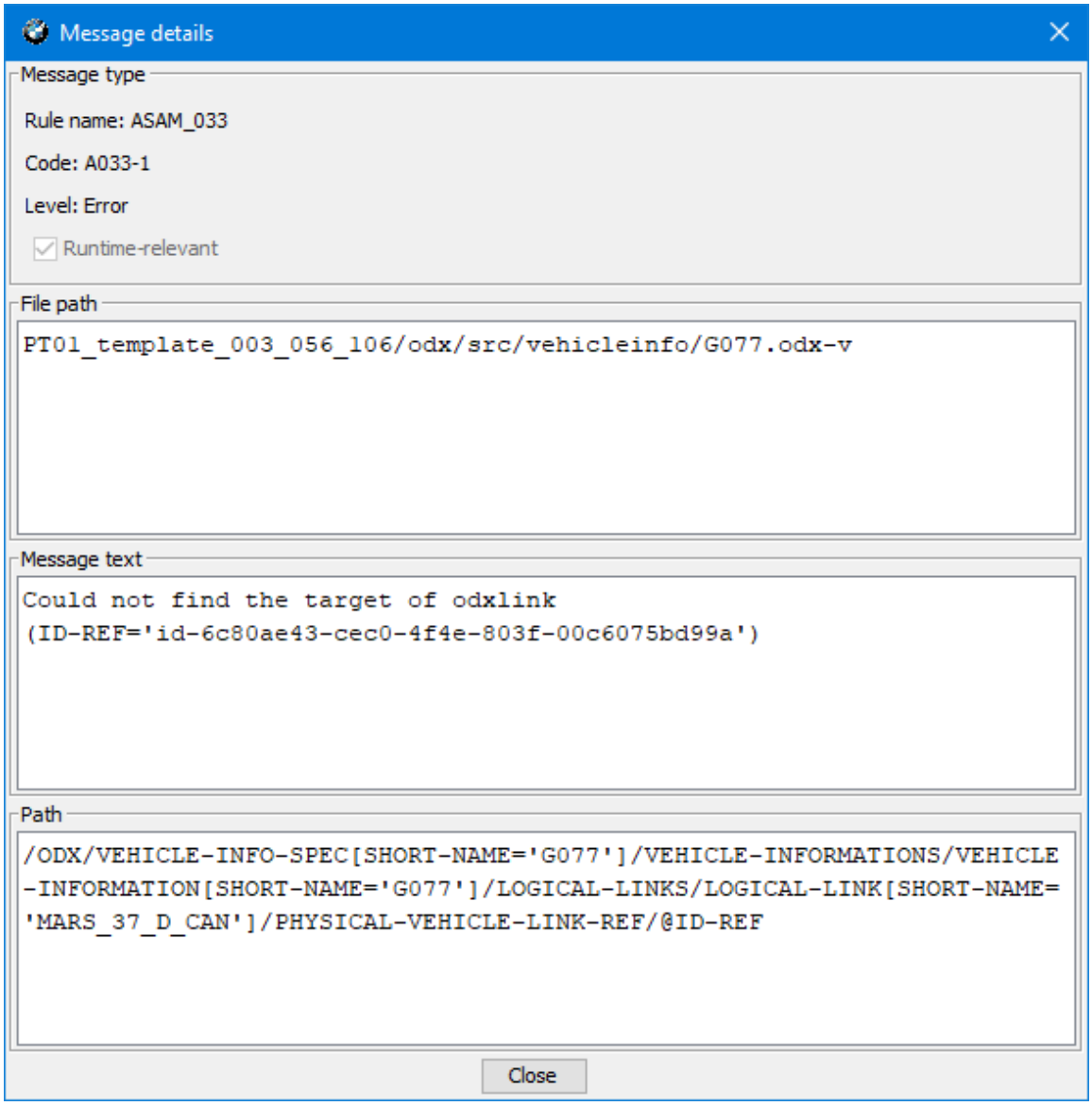
- Deletion Container is available only if there is no connection to the ECU.
- In the case of a missing or incorrect index file a new one index file will be created. All existing SWEs files and project names (TargetSelectors) are written in this file as "Undefined_Container".
- By Import of the same container more times under different project names, all linked project names (Target Selectors) are to deleted by deleting of this container.

ODX-Checker

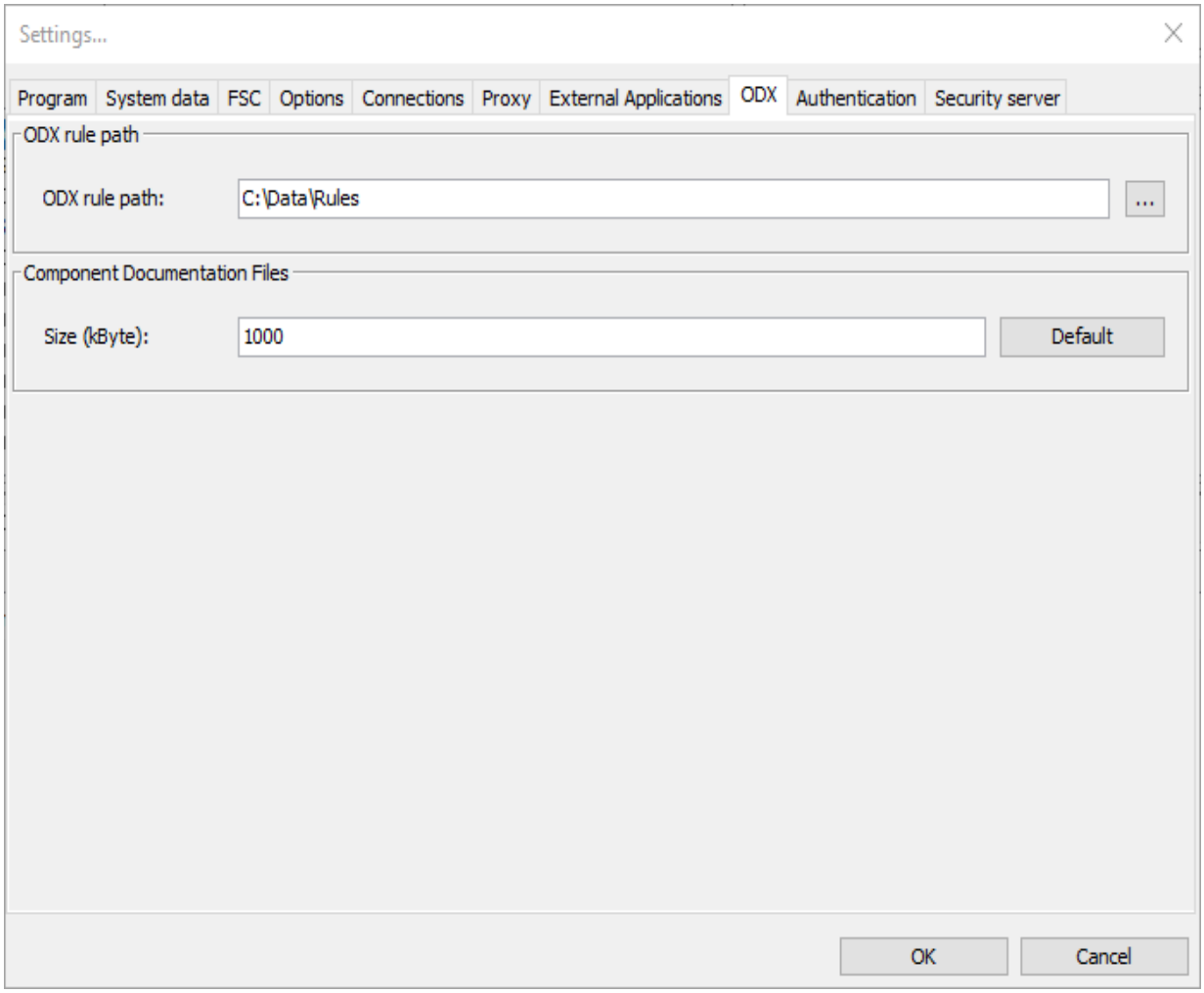
The PDX container check could be started with the button "ODX checker".
The result is presented in the "ODX checker messages" table.



Message details can be open with a double click on a line in the table.

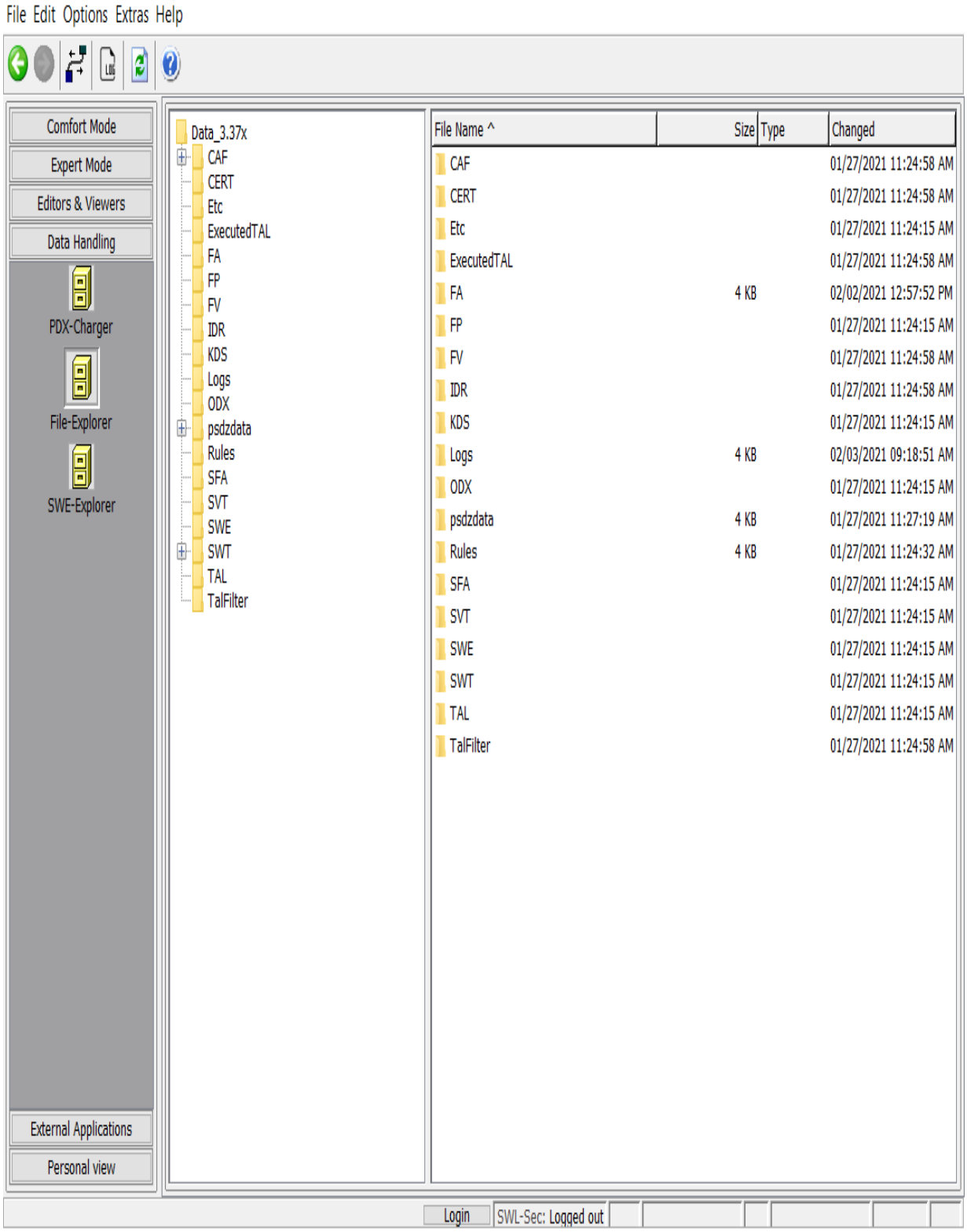


The ODX checker rules could be stored in a directory. The path for this directory could be set in menu "Options / Settings..." in the "ODX" tab:

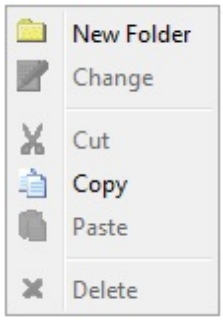

Settings...
Program
System data
FSC
Options
Connections
Proxy
External Applications
ODX
Authentication
Security server
ODX rule path
ODX rule path:
C:\Data\Rules
...
Component Documentation Files
Size (kByte):
1000
Default
OK
Cancel

File-Explorer

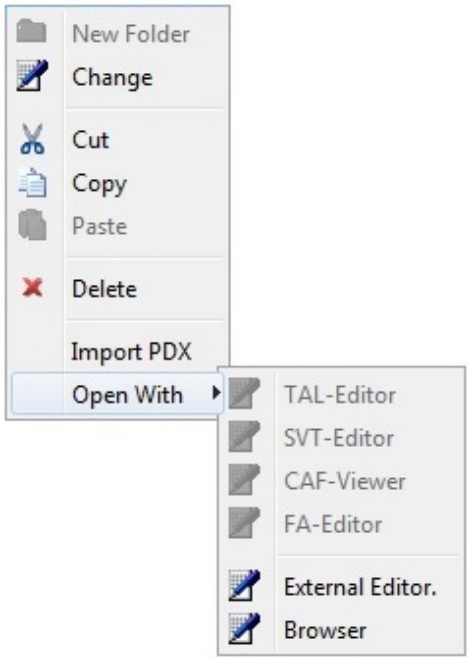
Using the file explorer you can manage the elements of the data directory. The files or directories can be copied, moved, deleted and created newly. In addition to that also an import into PSdZ (in case of ODX files) or passing the file to an external editor is possible.



Context Menu (left hand side)



Context Menu (right hand side)

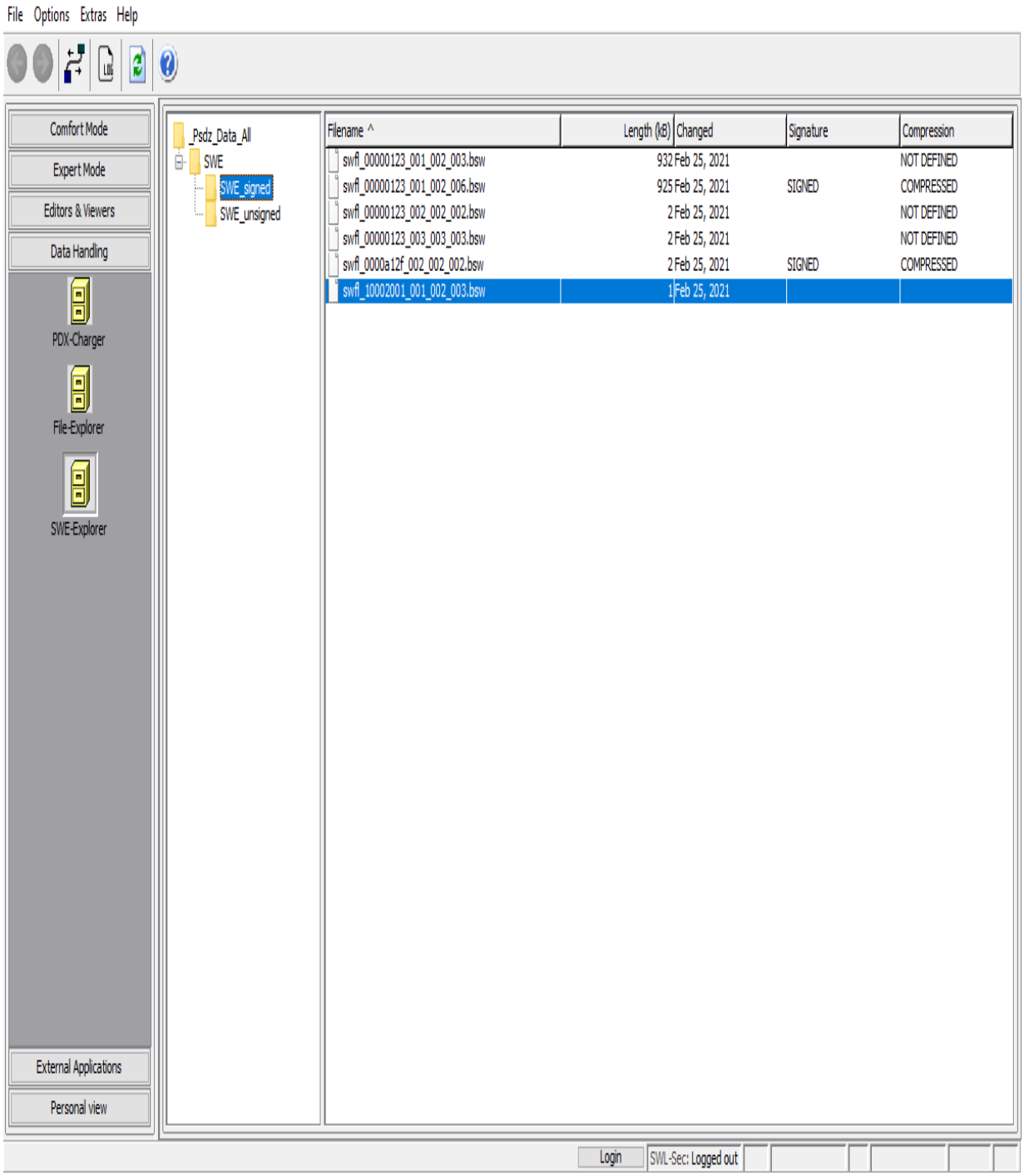


Specials:

- In the right hand display you have many options of editing using the context menu.
- "ImportPDX" only is available in subdirectory ODX, and even then only if there is no connection to the ECU.
- With the item "ImportPDX", it is possible to import single PDX-Container as well as a set of containers simultaneously.
- Using "Open With" menu let you jump directly to the suitable E-Sys editor corresponding to the data structure. In this editor the desired file is already displayed.
- The other context menu items are to be used in common manner.

SWE-Explorer

You can watch the both subdirectories SWE_signed and SWE_unsigned using this explorer.



Context Menu

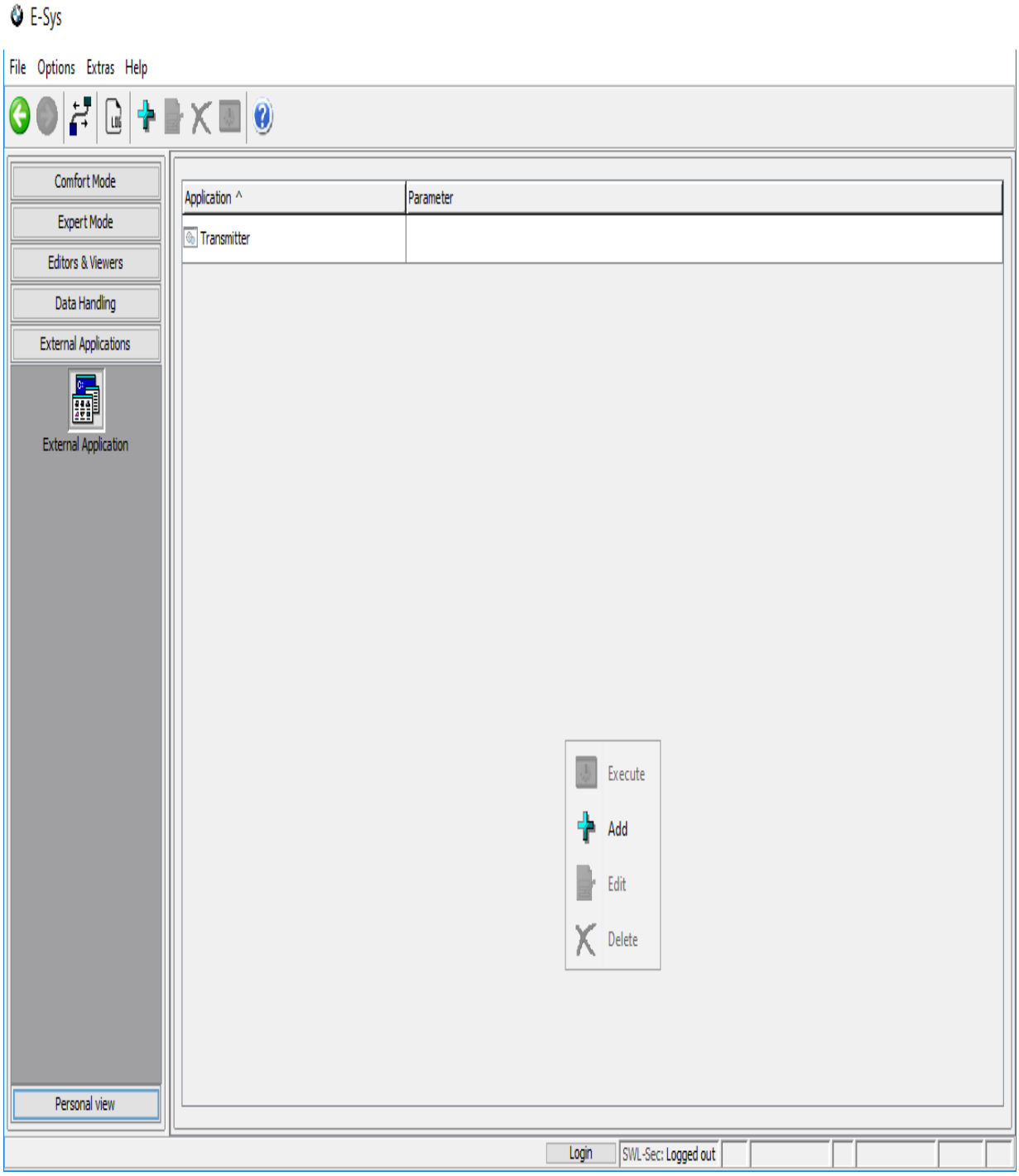
doesn't exist. This is no software bug.

Externe Application

Applications

Applications

In this module you can start any application. Therefore you must add a application by using the add button in the tool bar or in the context menu.



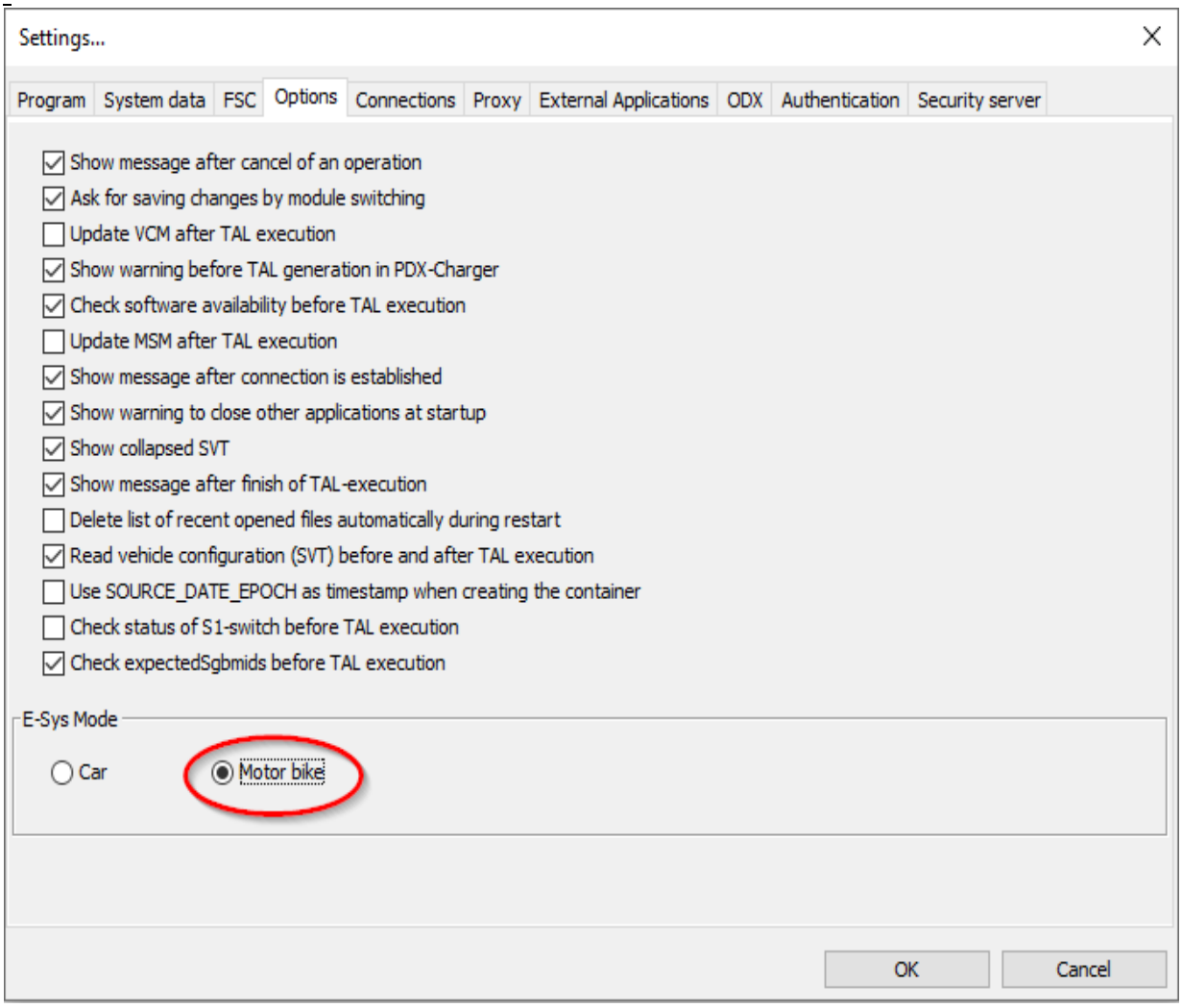
Instructions

Motor Bike
Establish connection
Batch

Motor Bike

With E-Sys also motor bikes can be programmed and coded. Therefore E-Sys mode must be changed to "Motor bike". See the menu Options | Settings... | Options.

Options



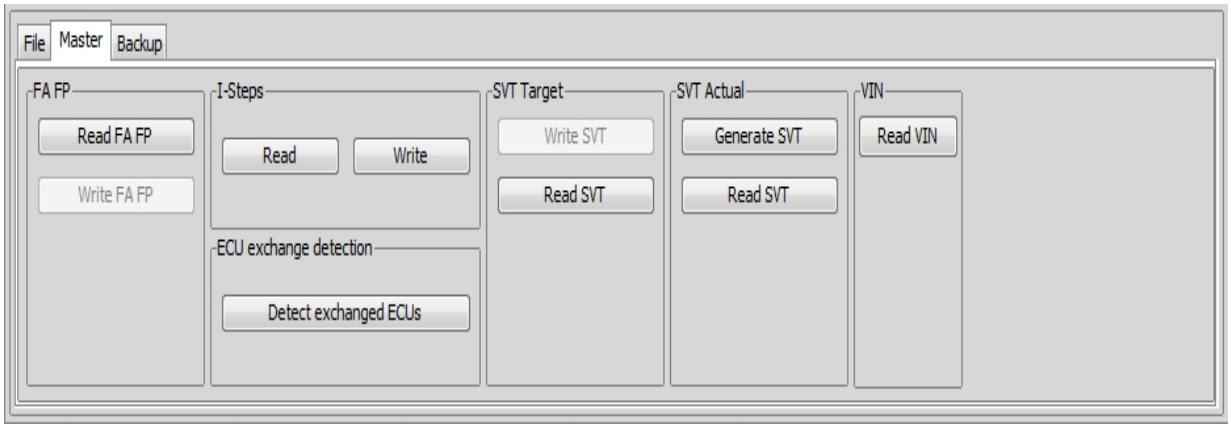
In mode motor bike the option "Update MSM after TAL execution" must be turned off, because of the missing MSM in motor bikes.

Disabled Actions

Several Emus of a car are not built in motor bikes. Thus some actions from E-Sys mode car are not available in mode motor bike. In the VCM Master tab of the VCM module the following actions are disabled:

- FA VIN -- Update VIN
- ECU exchange detection -- Detect exchanged Emus

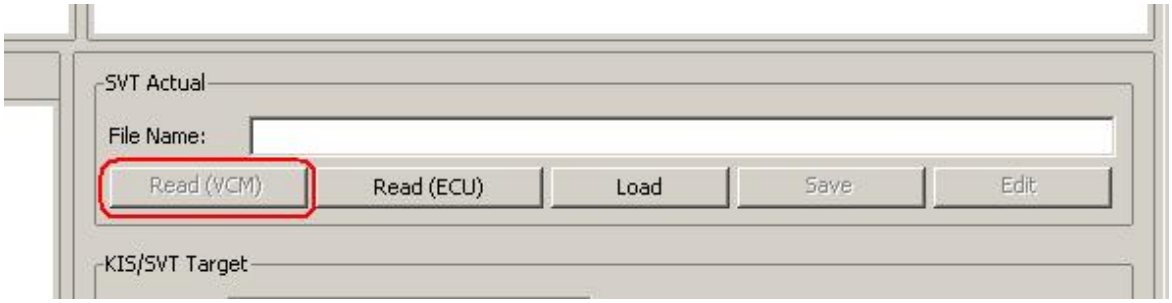
- SVT Target -- Write SVT, Read SVT
- SVT Actual -- Generate SVT, Read SVT



In the TAL-Calculation module and the Coding module the action

- SVT Actual -- Read (VCM)

is disabled.



Changed Functionality

In Modula VCM several actions read, write, or generate a vehicle profile (FP). This functionality is not available for motor bikes and thus will be owithted in the mode motor bike for the following actions:

- VCM Update after TAL execution
- VCM Master: FA VIN -- Read FA VIN
- VCM Master: FA VIN -- Write FA
- VCM Backup: VCM -- Write Data
- VCM Backup: Restore Data

Beside this, the actions will be executed the same as in mode car.

Currently not used

TSL-Update is currently not used in mode motor bike.

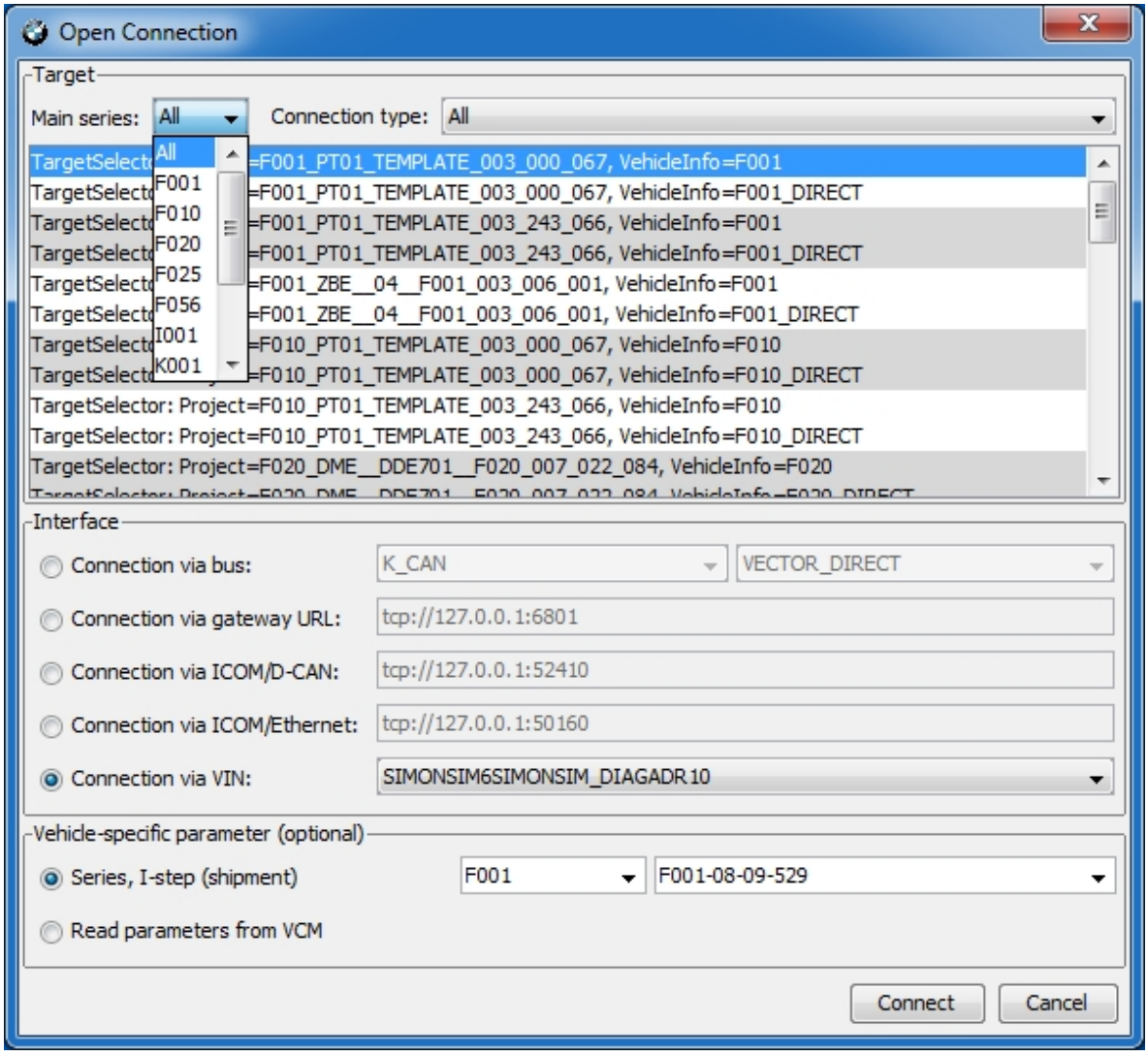
NAV/ENT Update is currently not used in mode motor bike.

Establish connection

Some functionalities of E-Sys need a established connection with the PDX date or the vehicle. Subsequent the process for establishing a connection with the vehicle is described shortly.



In the symbol bar you find the icon for starting/stopping a connection. After the click on "establish connection" subsequent dialog is displayed:



In the window "Target" you find all available projects and vehicle information. These could be filtered by the main series. With the second filter selects either gateway or direct ECU connection.
After the selection of a target selector and vehicle info the connection could be established by clicking on the "Connect" button.
If the connections was established successfully in the status bar you see the selected target.

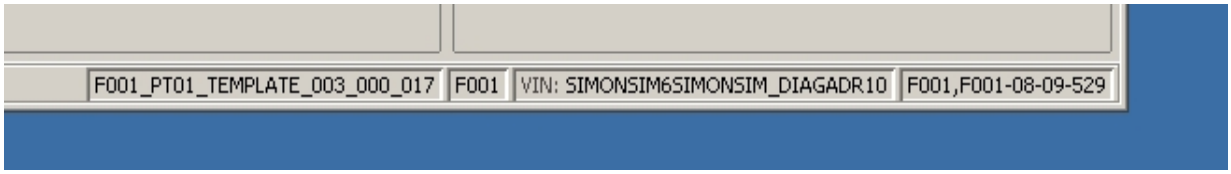
The type of the connection could be chosen in the "Interface" part of the dialog:

- For a connection via bus the bus type and a interface must be selected.
- For a connection via gateway URL the appropriate URL with port must be defined.
- For a connection via ICOM/CAN the URL of the ICOM with the appropriate port must be defined.
- For a connection via ICOM/ETHERNET the URL of the ICOM with the appropriate port must be defined. After an initial connection the port will be calculated from the base port taken from the settings.
- For a connection via VIN a vehicle could be chosen.

In the part "Vehicle-specific parameters" the series and I-step could be set.

- Enter series and/or I-step (both fields could be left blank also)
- Read series and I-step from VCM and comwith or change the values in the following dialog.

Information about the established connection are shown on the right hand side in the status bar.



Secure coding

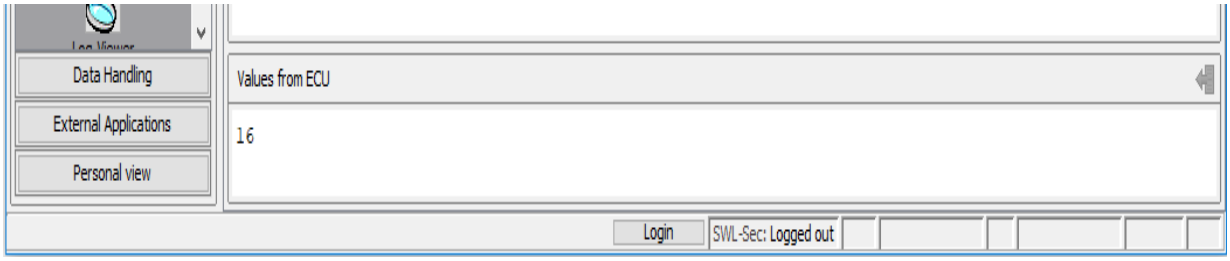
To code an ECU signed NCDs (Netto-Coding-Files) are used.
Based of secure aspects the lettering of the NCDs is performed in backend-system (outside of E-Sys).
Therefore the user has to be applied at the backend-system as well as to establish a backend connection.

Backend-connection for the secure Coding

To perform a TAL, that includes coding actions a backend-connection (to SWL-Sec2) is needed.
If you did not activated the backend-connection already the system is asking you to do it now. Therefore a popup appear (screenshot below) where the username, password and access point (BMW-Intranet or Internet) can be set.

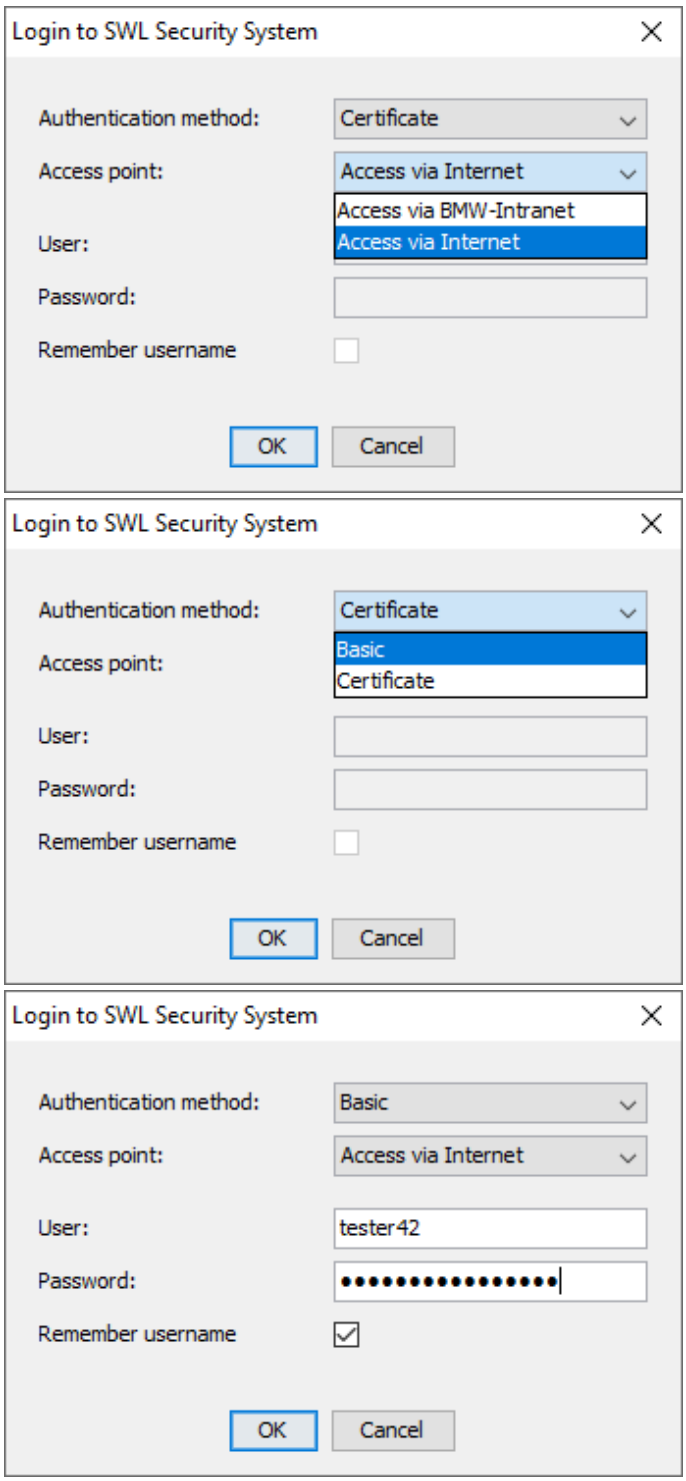
Two authentication methods are supported:

- username/password
- Certificate-based authentication



The button to perform the backend-connection can be find at the status bar .

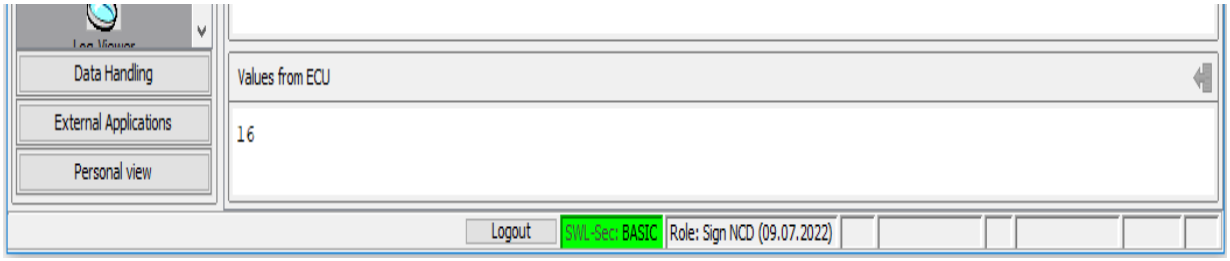
After the E-Sys user has clicked on the button, a pop-up window is going to appear. You have now two different possible configuration for the backend-system (*BMW-Intranet / Internet*).



If you choose *BMW-Intranet* your computer must be in the BMW IT-network.

The status of the authentication is indicated in the status bar. In addition to the selected authentication method, the activated role and

the validity of the role are displayed. In the case of successful Certificate-based authentication, the quota and the Certificate's usage time are also displayed.





Batch

General information on batch operation

General overview of commands

Anflash

Coding (NCD, verification, back-end authentication, etc.)

CVN / OBD

Read data

ECU Modes / SFA (Secure Feature Activation)

Server-Client operation

FA convert

FV ExecuteFV

PDX Containers

Proxy

FSC / SWT (sweeping Technoligie)

TAL processing

TAL calculation

TSL

VCM (Vehicle Configuration Management)

Certificate Management (SP18, SP21, etc.)

ConnectionParameter

General information on batch operation

General:

In addition to GUI mode, E-Sys can also be run in batch mode. For this purpose, the file `E-Sys.bat` with various parameters is started in the E-Sys installation directory (default: `C:\EC\Apps\ESG\E-Sys`). Depending on the specified parameter, an appropriate action is performed.

In addition, 2 operating modes are available to the user in batch modes.
One is client batch mode and other is client-server batch mode.

What is client operation?

In client mode, the following steps are always followed during TAL processing in batch mode:

1. Start E-Sys batch instance
2. Establish connection to vehicle
3. Execute TAL
4. Close connection to vehicle
5. Close E-Sys-Batch Instance

If you want to execute multiple TALs with the same vehicle connection, opening and closing the vehicle connection and starting and stopping an E-Sys batch instance means unnecessary overhead.
On the production-line, fast coding times are required, so a client-server-based batch mode has been introduced.
Thus, it is possible to start an E-Sys server instance, which is controlled with the help of a client.
In this way, individual E-Sys functions can be offered in batch.

How client-server batch mode works:

An E-Sys batch server is started, waiting for client requests. To check if an E-Sys batch server is running, the `-check` command is offered (see Server Client Operation).
A client can be started by specifying the commands with the batch parameter `-server` in another command line window (see overview of commands).
The client connects to a running E-Sys batch server, transfers the parameters specified on the command line to the server, and waits for a response.
The server performs the appropriate action and reports the result to the client. The client then exits with the appropriate return value.
The E-Sys batch server runs until it exits.

Example of a server-client operation:

1. Start E-Sys batch instance
2. Establish connection to vehicle
3. Read SVT
4. Close connection to vehicle
5. Import PDX Container
6. Establish connection to vehicle
7. Calculate TAL
8. Execute TAL
9. Close connection to vehicle
10. Read SVT
11. Close E-Sys-Batch Instance

Outputs to the console

When an action is written to the console, messages are output to the client console.

Logging

The log behavior can be set in the E-Sys.properties.

Default behavior: (log.file.client = false or not present in E-Sys.properties)

The client does not create a log file. This is not necessary because the client is only used to start an action on the server.
This means that the complete business logic runs on the server and is also logged there.

Log-File Splitting in Server Client Mode: (log.file.client = true in E-Sys.properties)

This setting creates a new log file for each client request for the server and client.
The file name shows which client request is involved.

Example:

E-Sys_20160203_150924_client_openconnection.log
E-Sys_20160203_150924_server_openconnection.log
E-Sys_20160203_151145_client_closeconnection.log
E-Sys_20160203_151145_server_closeconnection.log

Client-Server Connection

A TCP connection is established between the client and the server via port 30303.
If the port is not available, another port can be configured in the E-Sys.properties file (BatchServer.port) (see `-prop <E-Sys.properties>`).
Connections between client and server go through "localhost", i.e. the client and server must always be started on the same machine.

Restrictions

Only one E-Sys instance can run on a computer at a time.
This means that either the GUI mode, the E-Sys client or the E-Sys batch server must be started.

Only one client can be connected to the server at a time.
Note: TAL processing in client-server batch mode While the client that started the TAL processing is still waiting for the result, another client can request a cancellation of the TAL processing.

Legend:

[ ].....optional, can be specified, but does not have to
( ).....Selection, must be specified
| ....."or", is used in a selection

Anflash:

```
E-Sys.bat           -anflash <config file> [-ignoreBATHAF]
E-Sys.bat  -server  -anflash <config file> [-ignoreBATHAF]
```

Coding (NCD, verification, back-end authentication, etc.):

```
E-Sys.bat  -server  -authenticationCoding  -connection
<bmw_intranet|internet>  (-user <username>|-useSwlSecCertificate)
 [-dialog]
E-Sys.bat  -server  -backendCheckSigning  -vin <vin file>  -sgbmno
<SGBM file>
```
~~E-Sys.bat -fwl2ncd <config file>~~ **(Veraltet, bitte fwl2Ncd verwenden)**
~~E-Sys.bat -server -fwl2ncd <config file>~~ **(Veraltet, bitte fwl2Ncd verwenden)**
```
E-Sys.bat           -fwl2Ncd <config file>
E-Sys.bat  -server  -fwl2Ncd <config file>
```
~~E-Sys.bat -generatencd <FA file> [-caf <CAF name>] -trace <trace directory>~~ **(Veraltet, bitte generateNcd verwenden)**
~~E-Sys.bat -server -generatencd <FA file> [-caf <CAF name>] -trace <trace directory>~~ **(Veraltet, bitte generateNcd verwenden)**
```
E-Sys.bat           -generateNcd <FA file>  [-caf <CAF name>]  -
trace <trace directory>
E-Sys.bat  -server  -generateNcd <FA file>  [-caf <CAF name>]  -
trace <trace directory>
```
~~E-Sys.bat -ncdsign <config file>~~ **(Veraltet, bitte signNcd verwenden)**
~~E-Sys.bat -server -ncdsign <config file>~~ **(Veraltet, bitte signNcd verwenden)**
```
E-Sys.bat           -signNcd <config file>
E-Sys.bat  -server  -signNcd <config file>
```
~~E-Sys.bat -readncd <SVT file> -connection <config file> -out <NCD directory>~~ **(Veraltet, bitte readNcd verwenden)**
~~E-Sys.bat -server -readncd <SVT file> -connection <config file> -out <NCD directory>~~ **(Veraltet, bitte readNcd verwenden)**
```
E-Sys.bat           -readNcd <SVT file>  -connection <config file>
 -out <NCD directory>
```

```
E-Sys.bat  -server  -readNcd <SVT file>  -connection <config file>
 -out <NCD directory>
E-Sys.bat           -importSwlSecCertificate <P12 file>  [-dialog]
E-Sys.bat  -server  -importSwlSecCertificate <P12 file>  [-dialog]
E-Sys.bat           -generateTestNcds <config file>
E-Sys.bat  -server  -generateTestNcds <config file>
E-Sys.bat           -createCustomerNcd <config file>
E-Sys.bat  -server  -createCustomerNcd <config file>
```

CVN / OBD:

```
E-Sys.bat           -comparecvn  -connection <config file>  [-sg
<ecu list>]  (-svt <SVT file>|-svtvcmist|-svtvcmsoll|-svtecu)  -
trace <trace directory>  [-fa <FA file>]  [-xmlreport]
E-Sys.bat  -server  -comparecvn  -connection <config file>  [-sg
<ecu list>]  (-svt <SVT file>|-svtvcmist|-svtvcmsoll|-svtecu)  -
trace <trace directory>  [-fa <FA file>]  [-xmlreport]
E-Sys.bat           -cvncalculation <FA file>  -out <OBD-CVN-FA-
List file>  [-caf <CAF name>]
E-Sys.bat  -server  -cvncalculation <FA file>  -out <OBD-CVN-FA-
List file>  [-caf <CAF name>]
```

Read data:

```
E-Sys.bat           -getbrv <series>
E-Sys.bat  -server  -getbrv <series>
E-Sys.bat           -help
E-Sys.bat  -server  -help
E-Sys.bat           -readfa  -connection <config file>  -out <FA
directory>
E-Sys.bat  -server  -readfa  -connection <config file>  -out <FA
directory>
E-Sys.bat           -readsvt  -connection <config file>  -out <SVT
directory>
E-Sys.bat  -server  -readsvt  -connection <config file>  -out <SVT
directory>
E-Sys.bat           -version
E-Sys.bat  -server  -version
```

ECU Modes / SFA / LCS:

```
E-Sys.bat           -clearSecureToken  -connection <config file>  -
featureid <FeatureID>  (-diagaddress <diagnosis address>|-svt <SVT
file>)
E-Sys.bat  -server  -clearSecureToken  -connection <config file>  -
featureid <FeatureID>  (-diagaddress <diagnosis address>|-svt <SVT
```

```
file>)
E-Sys.bat           -createRequestEcuMode  -connection <config
file>  (-plant|-engineering) [-vin <VIN17>]  [-svt <SVT file>]  [-
blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -createRequestEcuMode  -connection <config
file>  (-plant|-engineering) [-vin <VIN17>]  [-svt <SVT file>]  [-
blacklist <file>]  [-whitelist <file>]
E-Sys.bat           -createRequestNewestPackage  -vin <VIN17>  [-
whitelist <file>]  [-rebuild]
E-Sys.bat  -server  -createRequestNewestPackage  -vin <VIN17>  [-
whitelist <file>]  [-rebuild]
E-Sys.bat           -createRequestPackageForOrder  -vin <VIN17>  [-
rebuild]
E-Sys.bat  -server  -createRequestPackageForOrder  -vin <VIN17>  [-
rebuild]
E-Sys.bat           -deleteSecureToken  -connection <config file>
 -featureid <FeatureID>  -diagaddress <diagnosis address>
E-Sys.bat  -server  -deleteSecureToken  -connection <config file>
 -featureid <FeatureID>  -diagaddress <diagnosis address>
E-Sys.bat           -discoverAllFeatureStatus  -connection <config
file>  [-svt <SVT file>]
E-Sys.bat  -server  -discoverAllFeatureStatus  -connection <config
file>  [-svt <SVT file>]
E-Sys.bat           -getResponseFromRequest  [-out <response file>]
 -request <request file>  -svt <SVT file>
E-Sys.bat  -server  -getResponseFromRequest  [-out <response file>]
 -request <request file>  -svt <SVT file>
E-Sys.bat           -readShortStatus  -connection <config file>  -
featureid <FeatureID>  -diagaddress <diagnosis address>
E-Sys.bat  -server  -readShortStatus  -connection <config file>  -
featureid <FeatureID>  -diagaddress <diagnosis address>
E-Sys.bat           -readSoftwareVersion  -connection <config file>
 -diagaddress <diagnosis address>
E-Sys.bat  -server  -readSoftwareVersion  -connection <config file>
 -diagaddress <diagnosis address>
E-Sys.bat           -switchEcuMode  -connection <config file>  -
tokenset <SecureToken file>  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]
E-Sys.bat  -server  -switchEcuMode  -connection <config file>  -
tokenset <SecureToken file>  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]
E-Sys.bat           -switchEcuModeField  -connection <config file>
 [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -switchEcuModeField  -connection <config file>
 [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat           -switchEcuModeOnline  -connection <config file>
 (-plant|-engineering)  [-vin <VIN17>]  [-svt <SVT file>]  [-
blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -switchEcuModeOnline  -connection <config file>
 (-plant|-engineering)  [-vin <VIN17>]  [-svt <SVT file>]  [-
```

```
blacklist <file>]  [-whitelist <file>]
E-Sys.bat           -verifySecureTokens  -connection <config file>
 [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -verifySecureTokens  -connection <config file>
 [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat           -verifySecureTokensVehicle  -connection <config
file>  [-svt <SVT file>]
E-Sys.bat  -server  -verifySecureTokensVehicle  -connection <config
file>  [-svt <SVT file>]
E-Sys.bat           -writeNewestPackageOnline  -connection <config
file>  [-disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]  [-
blacklist <file>]  [-whitelist <file>]  [-rebuild]
E-Sys.bat  -server  -writeNewestPackageOnline  -connection <config
file>  [-disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]  [-
blacklist <file>]  [-whitelist <file>]  [-rebuild]
E-Sys.bat           -writePackageForOrderOnline  -connection
<config file>  [-disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]
 [-blacklist <file>]  [-whitelist <file>]  [-rebuild]
E-Sys.bat  -server  -writePackageForOrderOnline  -connection
<config file>  [-disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]
 [-blacklist <file>]  [-whitelist <file>]  [-rebuild]
E-Sys.bat           -writeSecureTokens  -connection <config file>
 -tokenset <SecureToken file>  [-svt <SVT file>]  [-disableDelete]
 [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -writeSecureTokens  -connection <config file>
 -tokenset <SecureToken file>  [-svt <SVT file>]  [-disableDelete]
 [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat           -readLcs  -connection <config file>  [-lcsName
<LCS name>]  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist
<file>]
E-Sys.bat  -server  -readLcs  -connection <config file>  [-lcsName
<LCS name>]  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist
<file>]
E-Sys.bat           -setLcs  -connection <config file>  -lcsName
<LCS name>  -lcsValue <LCS value>  [-noreset]  [-svt <SVT file>]
 [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -setLcs  -connection <config file>  -lcsName
<LCS name>  -lcsValue <LCS value>  [-noreset]  [-svt <SVT file>]
 [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat           -writeLcsToken  -connection <config file>  -
tokenset <SecureToken file>  [-noreset]  [-svt <SVT file>]  [-
blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -writeLcsToken  -connection <config file>  -
tokenset <SecureToken file>  [-noreset]  [-svt <SVT file>]  [-
blacklist <file>]  [-whitelist <file>]
```

Server-Client operation:

```
E-Sys.bat  -server  -check
```

```
E-Sys.bat  -server  -stop
E-Sys.bat  -server  -openconnection <config file>
E-Sys.bat  -server  -closeconnection
E-Sys.bat  -server  -aborttalexecution
E-Sys.bat  -startserver
E-Sys.bat  -startserver [-prop <property file>] [-setProxy <config
file> ] [-authenticationCoding ( -user <Benutzername> | -
useSwlSecCertificate ) -connection <bmw_intranet|internet>] [-
dialog]
```

FA convert:

```
E-Sys.bat           -convertorderxml <FO directory>  -out <FA
directory>
E-Sys.bat  -server  -convertorderxml <FO directory>  -out <FA
directory>
```

FV ExecuteFV:

```
E-Sys.bat           -executeFv <config file> [-ignoreBATHAF]
E-Sys.bat  -server  -executeFv <config file> [-ignoreBATHAF]
```

PDX Containers:

```
E-Sys.bat           -mergepdxcontainer <PDX container definition
file>  -out <PDX container file>
E-Sys.bat  -server  -mergepdxcontainer <PDX container definition
file>  -out <PDX container file>
E-Sys.bat           -pdxcontainer <PDX container definition file>
 -out <PDX container file>
E-Sys.bat  -server  -pdxcontainer <PDX container definition file>
 -out <PDX container file>
E-Sys.bat           -pdximport <PDX container file>  -project
<project name>
E-Sys.bat  -server  -pdximport <PDX container file>  -project
<project name>
E-Sys.bat           -pdxupdate <PDX container file>  [-out <PDX
container file>]  -template <PDX template file>
E-Sys.bat  -server  -pdxupdate <PDX container file>  [-out <PDX
container file>]  -template <PDX template file>
```

Proxy:

```
E-Sys.bat           -readStatusProxy
E-Sys.bat  -server  -readStatusProxy
```

```
E-Sys.bat  -server  -removeProxy
E-Sys.bat  -server  -setProxy <config file>  [-dialog]
```

SWT:

```
E-Sys.bat           -swtaction <config file>
E-Sys.bat  -server  -swtaction <config file>
E-Sys.bat           -swtstatus  -project <project name>  -
diagaddress <diagnosis address>  -vehicleinfo <vehicle info>  -
basevariant <basevariant>
E-Sys.bat  -server  -swtstatus  -project <project name>  -
diagaddress <diagnosis address>  -vehicleinfo <vehicle info>  -
basevariant <basevariant>
```

TAL processing:

```
E-Sys.bat           -talexecution <config file> [-ignoreBATHAF]
E-Sys.bat  -server  -talexecution <config file> [-ignoreBATHAF]
```

TAL calculation:

```
E-Sys.bat           -talcalculation <config file>
E-Sys.bat  -server  -talcalculation <config file>
```

TSL:

```
E-Sys.bat           -tslstatus  -connection <config file>
E-Sys.bat  -server  -tslstatus  -connection <config file>
E-Sys.bat           -updatetsl <SVT filename>  -connection <config
file>
E-Sys.bat  -server  -updatetsl <SVT filename>  -connection <config
file>
```

Vehicle Configuration Management (VCM):

```
E-Sys.bat           -readVcmBackup <FA|ISTUFEN>  [-connection
<config file>]  -out <target directory>
E-Sys.bat  -server  -readVcmBackup <FA|ISTUFEN>  [-connection
<config file>]  -out <target directory>
E-Sys.bat           -readVcmMaster <FA|FP|ISTUFEN|SVTSOLL>  [-
connection <config file>]  -out <target directory>
E-Sys.bat  -server  -readVcmMaster <FA|FP|ISTUFEN|SVTSOLL>  [-
connection <config file>]  -out <target directory>
E-Sys.bat           -readVinFromBackup  [-connection <config file>]
```

```
E-Sys.bat  -server  -readVinFromBackup  [-connection <config file>]
E-Sys.bat           -readVinFromMaster  [-connection <config file>]
E-Sys.bat  -server  -readVinFromMaster  [-connection <config file>]
E-Sys.bat           -writeVcmBackup <FA|ISTUFEN>  [-connection
<config file>]  -in <source file>
E-Sys.bat  -server  -writeVcmBackup <FA|ISTUFEN>  [-connection
<config file>]  -in <source file>
E-Sys.bat           -writeVcmMaster <FA|ISTUFEN|SVTSOLL>  [-
connection <config file>]  -in <source file>
E-Sys.bat  -server  -writeVcmMaster <FA|ISTUFEN|SVTSOLL>  [-
connection <config file>]  -in <source file>
```

Certificate Management:

```
E-Sys.bat           -activateIPsec  -connection <config file>
E-Sys.bat  -server  -activateIPsec  -connection <config file>
E-Sys.bat           -certexecution <config file>
E-Sys.bat  -server  -certexecution <config file>
E-Sys.bat           -checkCERT  -connection <config file>  -retries
<value>  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist
<file>]
E-Sys.bat  -server  -checkCERT  -connection <config file>  -retries
<value>  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist
<file>]
E-Sys.bat           -deactivateIPsec  -connection <config file>
E-Sys.bat  -server  -deactivateIPsec  -connection <config file>
E-Sys.bat           -generateCSR  -connection <config file>  -out
<target file>  [-secOCKeys]  [-vin <VIN17>]  [-blacklist <file>]
 [-whitelist <file>]
E-Sys.bat  -server  -generateCSR  -connection <config file>  -out
<target file>  [-secOCKeys]  [-vin <VIN17>]  [-blacklist <file>]
 [-whitelist <file>]
E-Sys.bat           -getCbbResponseFromRequest <cbb request file>
[-out <target directory>]
E-Sys.bat  -server  -getCbbResponseFromRequest <cbb request file>
[-out <target directory>]
E-Sys.bat           -lockIPsec  -connection <config file>
E-Sys.bat  -server  -lockIPsec  -connection <config file>
E-Sys.bat           -readCERT  -connection <config file>  -type
<certificate type>  [-svt <SVT file>]  [-blacklist <file>]  [-
whitelist <file>]
E-Sys.bat  -server  -readCERT  -connection <config file>  -type
<certificate type>  [-svt <SVT file>]  [-blacklist <file>]  [-
whitelist <file>]
E-Sys.bat           -setSecOCCounter  -connection <config file>  -
value <hex value>  [-blacklist <file>]  [-whitelist <file>]
E-Sys.bat  -server  -setSecOCCounter  -connection <config file>  -
value <hex value>  [-blacklist <file>]  [-whitelist <file>]
```

```
E-Sys.bat           -writeBindings  -connection <config file>  -in
<CBB response file>  [-secOCKeys]  [-svt <SVT file>]  [-
secOCKeysPath <SecOCKeyPack file>]
E-Sys.bat  -server  -writeBindings  -connection <config file>  -in
<CBB response file>  [-secOCKeys]  [-svt <SVT file>]  [-
secOCKeysPath <SecOCKeyPack file>]
```

Others:

```
E-Sys.bat           [-prop <property file>]
```

General:

The Anflash module offers the possibility to perform a complete flash cycle fully automated.
The exact structure of the required configuration files is described in the Anflash manual.

[ ].....optional, can be specified, but does not have to
( ).....Selection, must be specified
| ....."or", is used in a selection

Batch-command Anflash

Command in client batch mode:

```
E-Sys.bat           -anflash <config file> [-ignoreBATHAF]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -anflash <config file> [-ignoreBATHAF]
```

Declaration:

`-anflash <config file>`	Absolute path to Anflash-Configuration-file (siehe Anflash-Handlungsanweisung).
`-ignoreBATHAF`	Optional parameter. Ignores the S1-switch.

Example:

```
E-Sys.bat -anflash C:\data\executeAnflash.config -ignoreBATHAF
```

Information:

- Information about the batch command `-anflash` is to be obtained from the "Anflash Manual".
- If a BAT/HAF test reveals that switch S1 is open, the further execution of the command depends on parameter `-ignoreBATHAF`.

`-ignoreBATHAF` has been set:

The flash sequence continues. However, the user must be aware that the execution can lead to interruptions. In this case, the TAL-processing cannot be ended with value 0 (OK), but at best with value 4 (with warning).

`-ignoreBATHAF` not set:
The execution of the command `-talexecution` is canceled.

General:

For NCDs, the following commands are offered in the batch.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Backend authentication for Coding

Log in to the Coding-Backend (necessary to sign the NCDs)

Command in client batch mode:

```
---only applicable in server-client operation---
```

Command in client-server batch mode:

```
E-Sys.bat -server -authenticationCoding -connection <bmw_intranet|internet> (-user <username>|-useS
dialog]
```

Declaration:

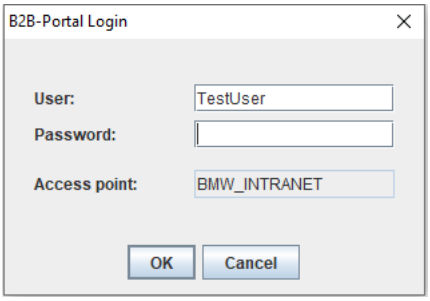
`-authenticationCoding`	command.
`-connection <bmw_intranet|internet>`	Login via `bmw_intranet` (inside of BMW-network) or `internet` (outsid
`-user <username>`	Backend-Login via username (example: max.mustermann).
`-useSwlSecCertificate`	Backend-Login via Certificate.
`-dialog`	Backend-Login with GUI-dialog.

Example:

```
E-Sys.bat -server -authenticationCoding -user Max.Mustermann -connection intranet
E-Sys.bat -server -authenticationCoding -user Max.Mustermann -connection bmw_intranet -dialog
E-Sys.bat -server -authenticationCoding -useSwlSecCertificate -connection bmw_intranet
```

Information:

- The `-connection` parameter determines whether the back-end system, which is addressed for the signin reached via the BMW intranet or over the Internet.
- Either the parameter `-user <username>` for Basic Authentication or `-useSwlSecCertificate` for Certificate-b be specified.
- In a basic authentication, the password must be entered manually in the command line of the **E-Sys se**
- If the switch `-dialog` is specified, a GUI dialog is displayed in which login data must be entered. The Login dialog looks like this:



- For Certificate-based authentication, the required client Certificate must first be imported using the "`-im`

batch command or in GUI mode.

- Whether logging in to the back-end system was successful can be seen in the command line of the E-S
  Basic authentication displays both the authorization and the expiration date of the authorization.
  In the case of successful Certificate-based authentication, the quota and the Certificate's usage time ar

Importing PKCS-12 Containers with Client Certificate for SWL Security System (SWL-SEC)

This command can be used to import client Certificates that are required to communicate with the BMW b
With a successfully imported client Certificate, Certificate-based authentication to the SWL Security Syste

Command in client batch mode:

```
E-Sys.bat         -importSwlSecCertificate <P12 file>  [-dialog]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -importSwlSecCertificate <P12 file>  [-dialog]
```

Declaration:

`-importSwlSecCertificate <P12 file>`	Absolute path to PKCS12 container with client Certificate.
`-dialog`	Backend-Login with GUI-dialog.

Example:

```
E-Sys.bat -importSwlSecCertificate C:\CERT\000_USR_Max.Mustermann@partner.bmw.de.p12
E-Sys.bat -importSwlSecCertificate C:\CERT\000_USR_Max.Mustermann@partner.bmw.de.p12 -dialog
```

Information:

- If the switch `-dialog` is defined, a GUI dialog is displayed in which authentication data must be entered.
- The command is supported in both client- and server-client-based batch operation.
- The required Certificate for SWL Security System must be obtained separately via the B2B portal.

Authorization check for NCD signing in the backend

This command checks whether the user is authorized to sign at the backend for the VINs or CAFD-SGBM

Command in client batch mode:

```
---only applicable in server-client operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -backendCheckSigning  -vin <vin file>  -sgbmno <SGBM file>
```

Declaration:

`-backendCheckSigning`	command.
`-vin <vin file>`	Absolute path to VIN-file.
`-sgbmno <SGBM file>`	Absolute path to SGBMNo-file.

Example:

```
E-Sys.bat -server -backendCheckSigning -vin C:\NCD\vinsToCheck.txt -sgbmno C:\NCD\sgbmnrToCheck.txt
```

Information:

- The command is only supported in server-client-mode and only with a BASIC authentication to backen
  The user must first login in to the backend using the "`-authenticationCoding`" command.
  A '\' may not be used in the configuration file. Instead, use '/' or '\\' for path information.

- The format of the configuration files to pass is defined as follows:
  Example for a VIN-file:
  ```
  # VIN17,VIN17,VIN17
  TESTVIN1234567890,VINTEST1234567890,1234567890TEST,TEST1234567890VIN
  ```

  Example for a SGBMNo-file:
  ```
  # SGBMNo,SGBMNo,SGBMNo
  00001234,00021000,00004321
  ```

Working with NCD

Creating NCD Files from FWL Files

NCD files are created from a list of FWL-files to be specified as a list in a configuration file.
The location for the generated NCD files is also set in this configuration file.
In addition, it contains the specification of a VIN, which is required to calculate the CPS (Coding Proof Sta

Command in client batch mode:

~~E-Sys.bat -fwl2ncd <config file>~~ **(deprecated)**
```
E-Sys.bat -fwl2Ncd <config file>
```

Command in client-server batch mode:

~~E-Sys.bat -server -fwl2ncd <config file>~~ **(deprecated)**
```
E-Sys.bat -server -fwl2Ncd <config file>
```

Declaration:

`-fwl2Ncd <config file>`	Absolute path to configuration file.

`<config file>`...Absolute path to configuration file

Example:
```
E-Sys.bat -fwl2Ncd C:\conf\fwl.properties
```

Information:

The `<config file>` must contain the following entries:

```
# Absolute path of an FA file. E.g. FA = C:/Data/fa.xml
FA = <FA file>

# e.g. NCD_DIR = C:/NCD
NCD_DIR = <NCD destination folder>

# e.g. FWL_LIST = C:/FWL/flw_1.fwl;C:/FWL/flw_2.fwl;
FWL_LIST = <path to fwl1>; <path to fwl2>; ....
```

Reading of Netto Coding Data

The Netto Coding Data of the ECUs is read out for all CAFs present in the SVT and stored as FWL and N
In addition, the VIN is determined and used to determine the file name.

Command in client batch mode:

~~E-Sys.bat -readncd <SVT> -connection <config file> -out <NCD directory>~~ **(deprecated)**
```
E-Sys.bat -readNcd <SVT file> -connection <config file> -out <NCD directory>
```

Command in client-server batch mode:

~~E-Sys.bat -server -readncd <SVT> -connection <config file> -out <NCD directory>~~ **(deprecated)**
```
E-Sys.bat -server -readNcd <SVT file> -connection <config file> -out <NCD directory>
```

Declaration:

`-readNcd <SVT file>`	Absolute path to SVT-file.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see
`-out <NCD directory>`	Absolute path to output-directory.

Example:

```
E-Sys.bat -readNcd C:\Data\SVT -connection C:\conf\connection.properties -out C:\Data\CAF
```

Information:

- The file name of the FWL file is formed according to the following pattern: "<VIN>_<CAF>_<BV>.fwl".
  VIN.....the last 7 digits of the VIN from the VIN-Master
  CAF.....SGBM-number of the CAF used
  BV.....Name of the basic variant

- The file name of the NCD-file is formed according to the following pattern: "<SGBMID>.caf.<MV> _<SV
  SGBMID.....SGBM-ID of the CAF used
  MV.....Main version of the CAF used
  SV.....Sub version of the CAF used
  PV.....Patch version of the CAF used

- The following return codes are returned:
  0 - The processing was completed successfully.
  1 - Errors have occurred.

Signing of Netto Coding Data

This command signs a list of NCD-files for the corresponding boot loaders..
The NCDs must be complete (incl. CPS). Validation or modification of the NCD-files is no longer possible

Command in client batch mode:

```
~~E-Sys.bat -ncdsign <config file>~~ (deprecated)
E-Sys.bat -signNcd <config file>
```

Command in client-server batch mode:

```
~~E-Sys.bat -server -ncdsign <config file>~~ (deprecated)
E-Sys.bat -server -signNcd <config file>
```

Declaration:

`-signNcd <config file>`	Absolute path to configuration file.

Example:

```
E-Sys.bat -server -signNcd C:\NCD\ncdSign.config
```

Information:

- In client-mode, the command is supported only in combination with Certificate-based-authentication.
- In server-client-mode, the user must first log in to the backend using the "`-authenticationCoding`" comm
- Required parameters are `FA` and `NCD_LIST_1`.
- Optional parameters are `BACKEND_SIGNATURE`, `SHIPMENT_NCD_GENERATION`, `SIGNED_NCD_DIR`, and `SHIPMENT_NCD_D` information from the `E-Sys.properties`-file is read or default settings are used.
- Depending on the `BACKEND_SIGNATURE`-setting (`ALLOW` - allow, `MUST_NOT` - prohibit, `FORCE` - enforce) the NCDs backend or local.
- The `SHIPMENT_NCD_GENERATION`-setting controls the calculation and signing of NCDs for delivery status. If c

delivery status are then calculated and signed.

- If multiple NCD lists are defined, the `NCD_LIST_x`-key must be unique.
- If multiple NCD lists are defined, the bootloader-number `<BTLD-Nummer-x>` must be unique.
- A '\' may not be used in the configuration file. Instead, use '/' or '\\' for path information.

The format of the configuration file to pass is defined as follows:

```
# Absolute path of an FA file. E.g. FA = C:/Data/fa.xml
FA = <FA file>

# Format MUST_NOT | FORCE | ALLOW
BACKEND_SIGNATURE = <MUST_NOT | FORCE | ALLOW>

# Generate shipment NCDs. DEFAULT = on
# Format: on | off
SHIPMENT_NCD_GENERATION = <on|off>

# z.B. SIGNED_NCD_DIR = C:/NCD/signed
SIGNED_NCD_DIR = <Destination directory for signed NCD>

# z.B. SHIPMENT_NCD_DIR = C:/NCD/shipment
SHIPMENT_NCD_DIR = <Destination directory for signed NCD>

# Format <BTLD-Number>;<NCD-Pfad> , z.B. "00001234;C:/NCD/CAFD_0000DDDD_026_000_002.ncd;C:/NCD/CAFD_
NCD_LIST_1 = <BTLD-Nummer-1>;<NCD-1>;<NCD-2>;<NCD-3>;<NCD-4>

# Optional parameter
# If specified, only client-batch-mode is considered
# Default value = off
USE_SWL_SEC_CERTIFICATE = <on|off>

# Optional parameter.
# Considered only in client-batch-mode
# If the parameter is owithted SWL_SEC_CONNECTION, the value from Esys.properties is used.
SWL_SEC_CONNECTION = <bmw_intranet|internet>
```

Example for a configuration file:

```
# Absolute path of an FA file. E.g. FA = C:/Data/fa.xml
FA = <FA file>

# Format MUST_NOT | FORCE | ALLOW
BACKEND_SIGNATURE = ALLOW

# Generate shipment NCDs. DEFAULT = on
# Format: on | off
SHIPMENT_NCD_GENERATION = off

# z.B. SIGNED_NCD_DIR = C:/NCDs/signed
SIGNED_NCD_DIR = C:/NCD/signed

# z.B. SHIPMENT_NCD_DIR = C:/NCD/shipment
SHIPMENT_NCD_DIR = C:/NCD/shipment

# Format <BTLD-Nummer>;<NCD path>
NCD_LIST_1 = 00001ffd;C:/NCD/CAFD_0000AAAA_006_011_017.ncd;C:/NCD/CAFD_0000BBBB_006_011_008.ncd;
NCD_LIST_2 = 00001c44;C:/NCD/CAFD_0000CCCC_026_000_000.ncd;C:/NCD/CAFD_0000DDDD_026_000_002.ncd;
```

Calculate Netto Coding Data (NCD)

Similar to the GUI module coding verification, a verification can be performed in batch mode after specifyi
and thus Netto Coding Data (NCD) can be calculated.

During verification, NCD- and FP-traces as well as a verification report are generated and stored in the tra
Verification is performed on all FAs at the FA-list and all imported CAFs.
The optional parameter `[-caf]` allows verification to be restricted to individual (imported) CAFs.
If multiple CAFs are specified, the names are separated by commas (",").

Command in client batch mode:

```
~~E-Sys.bat -generatencd <FA file> [-caf <CAF name>] -trace <trace directory>~~ (deprecated)
E-Sys.bat -generateNcd <FA file> [-caf <CAF name>] -trace <trace directory>
```

Command in client-server batch mode:

```
~~E-Sys.bat -server -generatencd <FA file> [-caf <CAF name>] -trace <trace directory>~~ (deprecated)
E-Sys.bat -server -generateNcd <FA file> [-caf <CAF name>] -trace <trace directory>
```

Declaration:

`-generateNcd <FA file>`	Absolute path of an FA- or FA-list-file.
`-caf <CAF name>`	Name of the CAF to be verified.
`-trace <trace directory>`	Absolute path of the directory to which the data to be generated is to

Example:

```
E-Sys.bat -generateNcd C:\Data\FA\F020.xml -trace C:\Data\Trace -caf cafd_0000815.caf.001_001_003
```

Information:

- No check will be performed to verify that the trace directory is empty. This means that existing files in th overwritten without warning.

- The following codes are returned:
  0 - All NCDs and FPs were generated without errors.
  1 - Errors occurred, so not all NCDs or FPs could be generated.

Calculate NCD-Testdata

NCD test data will be created and signed based on a list of CAF files, which have to be specified in a con

In addition to the CAFs, a vehicle order (file name + absolute path) must be specified in the configuration
It is also mandatory to provide a BTLD SGBM number, as this is required for signing the NCDs.

Optionally, the storage location for the generated NCD files can also be specified in the configuration file.

Command in client batch mode:

```
E-Sys.bat -generateTestNcds <config file>
```

Command in client-server batch mode:

```
E-Sys.bat -server -generateTestNcds <config file>
```

Declaration:

`-generateTestNcds <configuration file>`	Absolute path to configuration file.

Example:

```
E-Sys.bat -generateTestNcds  C:\conf\MyConfigForTestdata.cfg
```

Information:

- **In client mode, the command is only supported in combination with Certificate-based authentica**
- **In client-server mode, the user must first log in with a separate command to the backend.**
- If the BACKEND_SIGNATURE parameter is owithted, the value from *Esys.properties* is used.
- If the parameter NCD_TEST_DATA_DIR is owithted, then the E-Sys data directory is read from Esys.pr path is constructed to the folder in which the target folder "NCD test data" is created.
- Generation of test NCDs for Coding 2:
- FA file must contain exactly 1 FA
- Generation of test NCDs for Coding 3:
- FA file must contain exactly 2 FAs
- FAs must include Test-VINs (11. char of Long-VIN is 'H')
- FAs must not be identical, i.e. they must differ in at least one of the following points: colour code, fabric entries, E entries, K entries

The format of the configuration file is defined as follows:

```
# e.g. FA = C:/FA/fa_I020.xml
FA = <Vehicle order with absolute path>

# e.g. BTLD = 00001234
BTLD = <BTLD SGBM-Number>

# e.g. CAF_LIST = "cafd_0000DDDD.caf.026_000_002;cafd_0000BBBB.caf.006_011_008"
CAF_LIST = <CAF_file_name_1>;<CAF_file_name_2>;<CAF_file_name_3>;..

# Optional parameter
# e.g. NCD_TEST_DATA_DIR = C:/Testdata
NCD_TEST_DATA_DIR = <target folder for the ZIP-file containing the NCD-Testdata>

# Optional parameter
# Format MUST_NOT | FORCE | ALLOW
BACKEND_SIGNATURE = <MUST_NOT | FORCE | ALLOW>

# Optional parameter. Will be only considered in the Client Batch-Mode.
# If the parameter SWL_SEC_CONNECTION is missing, the value from Esys.properties will be used instea
SWL_SEC_CONNECTION = <bmw_intranet|internet>
```

At the end of the execution a generation report will be created. After the NCD-Calculation this report file v the generated ZIP-Archive.

Netto Coding Data (NCD) for customer vehicles

Similar to the GUI module 'NCD-Preparation', it is also possible to request NCDs in the SCB backend in b For this process, a list of vehicle orders (FA_LIST) is required, which must contain at least one entry.
In addition to the FA_LIST, either SVT or TAL must be defined.

The generated NCD files are stored in a directory after the command has been executed. This storage lo defined in the configuration file if required (SIGNED_NCD_DIR).
If the specification of an alternative storage location is missing, then the default directory for signed NCDs used.

This command can be used only in the BMW environment, because there is a communication with the Se (SCB).

Command in client batch mode:

```
E-Sys.bat         -createCustomerNcd <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -createCustomerNcd <config file>
```

Declaration:

`-createCustomerNcd <configuration file>`	Absolute path to configuration file.

Example:

```
E-Sys.bat -createCustomerNcd C:\conf\configForSCB.cfg
```

Information:

- The calculation and signing of the NCDs will be done in the SCB backend
- As a necessary precondition for the access to the SCB backend, a TIS-client Certificate has to be impo
- The URLs used for connecting to the SCB have to be defined in the esys.properties file by using the pr `scb.backend.url`
- Required parameters are `FA_LIST` und either `SVT` or `TAL`.
- The configuration file can contain either an `SVT` or a `TAL`, but not both.
- Optional parameter is `SIGNED_NCD_DIR`. If owithted, the information from the `E-Sys.properties`-file is read o used.

The format of the configuration file is defined as follows:

```
# e.g. FA_LIST = C:/Data/FA/fa_1.xml;C:/Data/FA/fa_2.xml;
FA_LIST = <Absolute path FA1>;<Absolute path FA2>;<Absolute path FA3>

# e.g. SVT = C:/Data/SVT/svt.xml
SVT = <Absolute path SVT>

# e.g. TAL = C:/Data/SVT/tal.xml
TAL = <Absolute path TAL>

# Optional parameter.
# e.g. SIGNED_NCD_DIR = C:/Data/NCD
SIGNED_NCD_DIR = <target folder for the NCD data>
```

- The following codes are returned:
  0 - All NCDs were generated without errors.
  1 - Errors occurred, so not all NCDs could be generated.

General:

CVNs (Calibration Verification Number) are calculated via the OBD-relevant (On-Board Diagnostics) SWEn for a control unit and compared with the CVNs that are read from the ECUs. The ECUs and SWEs required for the calculation are read from an SVT document.
The CVN values and the comparison results are written to a report file as text or in XML format (`-xmlreport`).
The SVT document can be read from the ECU, from the VCM, or from a file to be specified.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Calculate CVN

The following command can be used to calculate CVN values.

Command in client batch mode:

```
E-Sys.bat           -cvncalculation <FA file>  -out <OBD-CVN-FA-List file>  [-caf <CAF name>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -cvncalculation <FA file>  -out <OBD-CVN-FA-List file>  [-caf <CAF name>]
```

Declaration:

`-cvncalculation <FA file>`	Absolute path of an FA- or FA-list-file.
`-out <OBD-CVN-FA-List file>`	Absolute path of the output-file.
`-caf <CAF name>`	Names of the CAFs to use.

Example:

```
E-Sys.bat -cvncalculation
C:\Data_3_39_1_09f71b0_64bit_JDK11\FA\FaListSample.xml  -out
C:\Data_3_39_1_09f71b0_64bit_JDK11\Etc\CVN.xml  -caf
cafd_000000ae.caf.001_002_003
```

Information

Compare CVN

Command in client batch mode:

```
E-Sys.bat           -comparecvn  -connection <config file>  [-sg
<ecu list>]  (-svt <SVT file>|-svtvcmist|-svtvcmsoll|-svtecu)  -
trace <trace directory>  [-fa <FA file>]  [-xmlreport]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -comparecvn  -connection <config file>  [-sg
<ecu list>]  (-svt <SVT file>|-svtvcmist|-svtvcmsoll|-svtecu)  -
trace <trace directory>  [-fa <FA file>]  [-xmlreport]
```

Declaration:

`-comparecvn`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-sg <ecu list>`	ECU-filters. If the parameter is specified, only ECUs from the list are processed. The ECUs are separated by commas in the list.
`-svt <SVT file>`	Absolute path to SVT-file.
`-svtvcmist`	The SVTist is read from the VCM.
`-svtvcmsoll`	The SVTsoll is read from the VCM.
`-svtecu`	The SVT is read from the ECU.
`-trace <trace directory>`	Absolute path to the directory for the report-file.
`-fa <FA file>`	Absolute path to the FA-List-file. If the

	parameter "`-fa`" is not specified, the FA is read out of the vehicle.
`-xmlreport`	Erstellen einer Report-file im XML-Format.

Example:

```
E-Sys.bat -comparecvn -trace C:\Data\trace -svtecu -fa
C:\Data\FA\FaList.xml -sg ZBE,ZGW -connection
C:\conf\connection.properties
```

Information

- The following codes are returned:
  0 - The comparison was completed successfully.
  1 - The comparison could not be performed due to errors. The errors were written to the report file.

Read data

General:

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

read out the E-Sys-Version

With this command, the E-Sys-Version will be read out.

Command in client batch mode:

```
E-Sys.bat           -version
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -version
```

Declaration:

`-version`	command.

Example:

```
E-Sys.bat -version
E-Sys.bat -server -version
```

Information:

- The E-Sys-Version is output to stdout.

read BRV:

From the development-series, the name of the series-network is determined and output on the console.

Command in client batch mode:

```
E-Sys.bat           -getbrv <series>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -getbrv <series>
```

Declaration:

`-getbrv <series>`	Name of the development Series / Derivative / Serie.

Example:

```
E-Sys.bat -getbrv F003
E-Sys.bat -getbrv F44
E-Sys.bat -getbrv RR25
```

Information:

- The following codes are returned: 0 - The development series has been correctly determined.
  1 - Errors have occurred.

read FA:

The vehicle-order is read out of the vehicle and stored in a file.
If only one directory is specified, the VIN is also determined and used to determine the file-name.
Otherwise, the specified file will be used. If it already exists, it will be overwritten.

Command in client batch mode:

```
E-Sys.bat           -readfa  -connection <config file>  -out <FA directory>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readfa  -connection <config file>  -out <FA directory>
```

Declaration:

`-readfa`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-out <FA directory>`	Absolute path for the output directory with or without file-specification.

Example:

```
E-Sys.bat -readfa -out C:\Data\FA  -connection C:\conf\connection.properties
E-Sys.bat -readfa -out C:\Data\FA\FA_123.xml  -connection C:\conf\connection.properties
```

Information:

- If no file-name is specified, it is determined after the pattern "FA_<VIN>_<Timestamp>.xml".
  VIN..... the last 7 digits of the VIN from the VIN-Master
  Timestamp..... Timestamp according to the pattern ddmmyyyyhhmm

- The following codes are returned:
  0 - The processing was completed successfully.
  1 - Errors have occurred.

read SVT:

The SVT is read by the vehicle and stored in a file.
If only one directory is specified, the VIN is also determined and used to determine the file-name.
Otherwise, the specified file will be used. If it already exists, it will be overwritten.

Command in client batch mode:

```
E-Sys.bat           -readsvt  -connection <config file>  -out <SVT directory>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readsvt  -connection <config file>  -out <SVT directory>
```

Declaration:

`-readsvt`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-out <SVT directory>`	Absolute path for the output directory with or without file specification.

Example:

```
E-Sys.bat -readsvt -out C:\Data\SVT -connection C:\conf\connection.properties
E-Sys.bat -readsvt -out C:\Data\SVT\SVT_123.xml -connection C:\conf\connection.properties
```

Information:

- If no file name is specified, it is determined after the pattern "SVT_<VIN>_<Timestamp>.xml".
  Vin..... the last 7 digits of the VIN from the VIN-Master
  Timestamp..... Time stamp according to the pattern yyyymmddhhmm

- The following codes are returned:
  0 - The processing was completed successfully.
  1 - Errors have occurred.

ECU-Modes / SFA (Secure Feature Activation) / LCS (Locking Configuration Switch)

General:

For Secure ECU-Modes / Secure Feature Activation, the following commands are offered in Batch.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Switch ECU-Modes / write tokens (with backend-connection)

Switching to plant- / engineering-mode

Command in client batch mode:

```
E-Sys.bat  -switchEcuModeOnline  -connection <config file>  (-plant|-engineering)  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -switchEcuModeOnline  -connection <config file>  (-plant|-engineering)  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Declaration:

`-switchEcuModeOnline`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-plant`	Argument.
`-engineering`	Argument.
`-vin <VIN17>`	17-digit VIN.
`-svt <SVT file>`	Absolute path to The SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -switchEcuModeOnline -connection C:\conf\connection.properties -plant -vin WBA11CF070H123456 -blacklist C:\Daten\blacklist.txt
```

```
E-Sys.bat -switchEcuModeOnline -connection C:\conf\connection.properties -
engineering -svt C:\SVT\SVT_I20.xml -whitelist C:\Daten\whitelist.txt
```

Information:

- The accessibility of the BMW-backend is assumed.
- In order to be able to use the required services of the BMW-backend, a corresponding configuration of the client and server including authentication in the options is required.

Write the latest secure-tokens

Command in client batch mode:

```
E-Sys.bat           -writeNewestPackageOnline  -connection <config file>  [-
disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]  [-
whitelist <file>]  [-rebuild]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -writeNewestPackageOnline  -connection <config file>  [-
disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]  [-
whitelist <file>]  [-rebuild]
```

Declaration:

`-writeNewestPackageOnline`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-disableDelete`	Do not delete Secure Features.
`-vin <VIN17>`	17-digit VIN.
`-svt <SVT file>`	Absolute path to SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.
`-rebuild`	Recalculation of tokens.

Example:

```
E-Sys.bat -writeNewestPackageOnline -connection
C:\conf\connection.properties -disableDelete -vin WBA11CF070H123456 -svt -
svt C:\SVT\SVT_I20.xml -blacklist C:\Daten\blacklist.txt
E-Sys.bat -writeNewestPackageOnline -connection
C:\conf\connection.properties -vin WBA11CF070H123456 -svt <SVT file> -
whitelist C:\Daten\whitelist.txt -rebuild
```

Information:

- The accessibility of the BMW-backend is assumed.
- In order to be able to use the required services of the BMW-backend, a corresponding configuration of the client and server including authentication in the options is required.

Write secure-tokens for factory-state

Command in client batch mode:

```
E-Sys.bat           -writePackageForOrderOnline  -connection <config file>
 [-disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]  [-rebuild]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -writePackageForOrderOnline  -connection <config file>
 [-disableDelete]  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]  [-rebuild]
```

Declaration:

`-writePackageForOrderOnline`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-disableDelete`	Do not delete Secure Features.
`-vin <VIN17>`	17-digit VIN.
`-svt <SVT file>`	Absolute path to SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.
`-rebuild`	Recalculation of tokens.

Example:

```
E-Sys.bat -writePackageForOrderOnline -connection
C:\conf\connection.properties -disableDelete -vin WBA11CF070H123456 -svt -
svt C:\SVT\SVT_I20.xml -blacklist C:\Daten\blacklist.txt
E-Sys.bat -writePackageForOrderOnline -connection
C:\conf\connection.properties -vin WBA11CF070H123456 -svt <SVT file> -
whitelist C:\Daten\whitelist.txt -rebuild
```

Information:

- The accessibility of the BMW-backend is assumed.
- In order to be able to use the required services of the BMW-backend, a corresponding configuration of the client and server including authentication in the options is required.

Create request-file (without backend-connection)

Create request-file for switching to plant- / enginieering-mode

Command in client batch mode:

```
E-Sys.bat           -createRequestEcuMode  -connection <config file>  (-
plant|-engineering)  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -createRequestEcuMode  -connection <config file>  (-
plant|-engineering)  [-vin <VIN17>]  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]
```

Declaration:

`-createRequestEcuMode`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-plant`	Argument.
`-engineering`	Argument.
`-vin <VIN17>`	17-digit VIN.
`-svt <SVT file>`	Absolute path to SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -createRequestEcuMode -connection C:\conf\connection.properties -
plant -vin WBA11CF070H123456 -blacklist C:\Daten\blacklist.txt
E-Sys.bat -createRequestEcuMode -connection C:\conf\connection.properties -
engineering -svt C:\SVT\SVT_I20.xml -whitelist C:\Daten\whitelist.txt
```

Information:

Create a request-file for the latest secure-tokens

Command in client batch mode:

```
E-Sys.bat           -createRequestNewestPackage  -vin <VIN17>  [-whitelist
<file>]  [-rebuild]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -createRequestNewestPackage  -vin <VIN17>  [-whitelist
<file>]  [-rebuild]
```

Declaration:

`-createRequestNewestPackage`	command.
`-vin <VIN17>`	17-digit VIN.
`-whitelist <file>`	Absolute path to Whitelist-file.
`-rebuild`	Recalculation of tokens.

Example:

```
E-Sys.bat -createRequestNewestPackage -vin WBA11CF070H123456 -blacklist
C:\Daten\blacklist.txt
E-Sys.bat -createRequestNewestPackage -vin WBA11CF070H123456 -whitelist
C:\Daten\whitelist.txt -rebuild
```

Information:

Create a request-file for factory-state secure-tokens

Command in client batch mode:

```
E-Sys.bat           -createRequestPackageForOrder  -vin <VIN17>  [-rebuild]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -createRequestPackageForOrder  -vin <VIN17>  [-rebuild]
```

Declaration:

`-createRequestPackageForOrder`	command.
`-vin <VIN17>`	17-digit VIN.

-rebuild	Recalculation of tokens.

Example:

```
E-Sys.bat -createRequestPackageForOrder -vin WBA11CF070H123456 -rebuild
```

Information:

Request-token-set-file from a request-file in the BMW backend

Request-token-set-file for ECU-Mode or SFA from a request-file in the BMW backend

Command in client batch mode:

```
E-Sys.bat           -getResponseFromRequest  [-out <response file>]  -
request <request file>  -svt <SVT file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -getResponseFromRequest  [-out <response file>]  -
request <request file>  -svt <SVT file>
```

Declaration:

-getResponseFromRequest	command.
-out <response file>	Absolute path for the output-directory with file-specification (response-file [JSON]).
-request <request file>	Absolute path to request-file [JSON].
-svt <SVT file>	Absolute path to SVT-file.

Example:

```
E-Sys.bat -getResponseFromRequest -request
C:\Daten\WBA11CF070H123456_Plant_Mode_Request_20200528_131607[JSON].txt -svt
C:\SVT\SVT_I20.xml
E-Sys.bat -getResponseFromRequest -out <response file> -request
C:\Daten\WBA11CF070H123456_Plant_Mode_Request_20200528_131607[JSON].txt -svt
C:\SVT\SVT_I20.xml
```

Information:

- The accessibility of the BMW-backend is assumed.
- In order to be able to use the required services of the BMW-backend, a

corresponding configuration of the client and server including authentication in the options is required.

- All four request-file-types can be handled (EcuMode, PackageForOrder, NewestPackageForVehicle, Newest PackageForEcu).
- NewestPackage-request-files must contain "Newest" or they will be interpreted as PackageForOrder-request.

Write tokens from existing token-sets

Write ECU-mode-token from existing token-set

Command in client batch mode:

```
E-Sys.bat           -switchEcuMode  -connection <config file>  -tokenset
<SecureToken file>  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist
<file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -switchEcuMode  -connection <config file>  -tokenset
<SecureToken file>  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist
<file>]
```

Declaration:

`-switchEcuMode`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-tokenset <SecureToken file>`	Absolute path to ECU-mode-token-set (response-file [JSON]).
`-svt <SVT file>`	Absolute path to SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -switchEcuMode -connection C:\conf\connection.properties -tokenset
C:\SecureTokenRequests\SecureTokenResponse_ADCAM2_ACSM6_Engineering_Mode.txt
-blacklist C:\Daten\blacklist.txt
E-Sys.bat -switchEcuMode -connection C:\conf\connection.properties -tokenset
C:\SecureTokenRequests\SecureTokenResponse_ADCAM2_ACSM6_Engineering_Mode.txt
-svt C:\SVT\SVT_I20.xml -whitelist C:\Daten\whitelist.txt
```

Information:

Write secure-tokens from existing token-set

Command in client batch mode:

```
E-Sys.bat           -writeSecureTokens  -connection <config file>  -tokenset
<SecureToken file>  [-svt <SVT file>]  [-disableDelete]  [-blacklist <file>]
 [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -writeSecureTokens  -connection <config file>  -tokenset
<SecureToken file>  [-svt <SVT file>]  [-disableDelete]  [-blacklist <file>]
 [-whitelist <file>]
```

Declaration:

`-writeSecureTokens`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-tokenset <SecureToken file>`	Absolute path to secure-token-set (response-file [JSON]).
`-svt <SVT file>`	Absolute path to SVT-file.
`-disableDelete`	Do not delete secure-features.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -writeSecureTokens -connection C:\conf\connection.properties -
tokenset C:\Daten\SecureTokenResponse_ADCAM2_ACSM6_Engineering_Mode.txt -svt
C:\SVT\SVT_I20.xml -disableDelete
E-Sys.bat -writeSecureTokens -connection C:\conf\connection.properties -
tokenset C:\Daten\SecureTokenResponse_ADCAM2_ACSM6_Engineering_Mode.txt -
blacklist C:\Daten\blacklist.txt
E-Sys.bat -writeSecureTokens -connection C:\conf\connection.properties -
tokenset C:\Daten\SecureTokenResponse_ADCAM2_ACSM6_Engineering_Mode.txt -
whitelist C:\Daten\whitelist.txt
```

Information:

Switch ECUs to field-mode

Switch ECUs to field-mode

Command in client batch mode:

```
E-Sys.bat           -switchEcuModeField  -connection <config file>  [-svt
<SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -switchEcuModeField  -connection <config file>  [-svt
<SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Declaration:

`-switchEcuModeField`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-svt <SVT file>`	Absolute path to SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -switchEcuModeField -connection C:\conf\connection.properties -svt
C:\SVT\SVT_I20.xml -blacklist C:\Daten\blacklist.txt
E-Sys.bat -switchEcuModeField -connection C:\conf\connection.properties -
whitelist C:\Daten\whitelist.txt
```

Information:

Clean / delete secure-tokens from ECUs

Clean up secure-tokens from ECUs (CLEAR_FEATURE --> secure-tokens can be reused)

Command in client batch mode:

```
E-Sys.bat           -clearSecureToken  -connection <config file>  -featureid
<FeatureID>  (-diagaddress <diagnosis address>|-svt <SVT file>)
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -clearSecureToken  -connection <config file>  -featureid
<FeatureID>  (-diagaddress <diagnosis address>|-svt <SVT file>)
```

Declaration:

`-clearSecureToken`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-featureid <FeatureID>`	FeatureID in Hex, e.g. 0x23.
`-diagaddress <diagnosis address>`	Diagnostic address of the ECU in hex.
`-svt <SVT file>`	Absolute path to SVT-file.

Example:

Information:

- All ECUs must be in engineering mode.

Delete secure-tokens from ECUs (DELETE --> secure-tokens cannot be reused)

Command in client batch mode:

```
E-Sys.bat          -deleteSecureToken  -connection <config file>  -
featureid <FeatureID>  -diagaddress <diagnosis address>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -deleteSecureToken  -connection <config file>  -
featureid <FeatureID>  -diagaddress <diagnosis address>
```

Declaration:

`-deleteSecureToken`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-featureid <FeatureID>`	FeatureID in Hex, e.g. 0xa23456.
`-diagaddress <diagnosis`	

address>	Diagnostic address of the ECU in hex.

Example:

```
E-Sys.bat -deleteSecureToken -connection C:\conf\connection.properties -
featureid 0xa23456 -diagaddress 0x5d
```

Information:

Query SFA status

Status query of all ECUs per FeatureID

Command in client batch mode:

```
E-Sys.bat          -discoverAllFeatureStatus  -connection <config file>  [-
svt <SVT file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -discoverAllFeatureStatus  -connection <config file>  [-
svt <SVT file>]
```

Declaration:

-discoverAllFeatureStatus	command.
-connection <config file>	Absolute path to configuration file with the connection parameter (see Connection Parameter).
-svt <SVT file>	Absolute path to SVT-file.

Example:

```
E-Sys.bat -discoverAllFeatureStatus -connection
C:\conf\connection.properties
E-Sys.bat -discoverAllFeatureStatus -connection
C:\conf\connection.properties -svt C:\SVT\SVT_I20.xml
```

Information:

Status query of a featureID of a ECU

Command in client batch mode:

```
E-Sys.bat           -readShortStatus  -connection <config file>  -featureid
<FeatureID>  -diagaddress <diagnosis address>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readShortStatus  -connection <config file>  -featureid
<FeatureID>  -diagaddress <diagnosis address>
```

Declaration:

`-readShortStatus`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (seeConnection Parameter).
`-featureid <FeatureID>`	FeatureID in Hex, e.g. 0xa23456.
`-diagaddress <diagnosis address>`	Diagnostic address of the ECU in hex.

Example:

```
E-Sys.bat -readShortStatus -connection C:\conf\connection.properties -
featureid 0xa23456 -diagaddress 0x5d
```

Information:

Check the validity of the secure-tokens (for a list of ECUs)

Command in client batch mode:

```
E-Sys.bat           -verifySecureTokens  -connection <config file>  [-svt
<SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -verifySecureTokens  -connection <config file>  [-svt
<SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Declaration:

`-verifySecureTokens`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).

-svt <SVT file>	Absolute path to SVT-file.
-blacklist <file>	Absolute path to Blacklist-file.
-whitelist <file>	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -verifySecureTokens -connection C:\conf\connection.properties -
blacklist C:\Daten\blacklist.txt
E-Sys.bat -verifySecureTokens -connection C:\conf\connection.properties -svt
C:\SVT\SVT_I20.xml -whitelist C:\Daten\whitelist.txt
```

Information:

Check the validity of the secure-tokens (for all ECUs of a vehicle)

Command in client batch mode:

```
E-Sys.bat           -verifySecureTokensVehicle  -connection <config file>
 [-svt <SVT file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -verifySecureTokensVehicle  -connection <config file>
 [-svt <SVT file>]
```

Declaration:

-verifySecureTokensVehicle	command.
-connection <config file>	Absolute path to configuration file with the connection parameter (see Connection Parameter).
-svt <SVT file>	Absolute path to SVT-file.

Example:

```
E-Sys.bat -verifySecureTokensVehicle -connection
C:\conf\connection.properties
E-Sys.bat -verifySecureTokensVehicle -connection
C:\conf\connection.properties -svt C:\SVT\SVT_I20.xml
```

Information:

Read SFA-software-version from ECU

Command in client batch mode:

```
E-Sys.bat           -readSoftwareVersion  -connection <config file>  -
diagaddress <diagnosis address>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readSoftwareVersion  -connection <config file>  -
diagaddress <diagnosis address>
```

Declaration:

`-readSoftwareVersion`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).
`-diagaddress <diagnosis address>`	Diagnostic address of the ECU in hex.

Example:

```
E-Sys.bat -readSoftwareVersion -connection C:\conf\connection.properties -
diagaddress 0x5d
```

Information:

Read or switch over Locking Configuration Switch (LCS)

Read LCS values

Command in client batch mode:

```
E-Sys.bat           -readLcs  -connection <config file>  [-lcsName <LCS
name>]  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readLcs  -connection <config file>  [-lcsName <LCS
name>]  [-svt <SVT file>]  [-blacklist <file>]  [-whitelist <file>]
```

Declaration:

`-readLcs`	

	command.
-connection <config file>	Absolute path to configuration file with the connection parameter (see Connection Parameter).
-lcsName <LCS name>	LCS-function in hex (e.g.: 0x01).
-svt <SVT file>	Absolute path to SVT-file.
-blacklist <file>	Absolute path to Blacklist-file.
-whitelist <file>	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -readLcs -connection C:\conf\connection.properties -whitelist
C:\Daten\whitelist.txt
E-Sys.bat -readLcs -connection C:\conf\connection.properties -lcsName 0x01 -
svt C:\SVT\SVT_I20.xml
E-Sys.bat -readLcs -connection C:\conf\connection.properties -lcsName 0x01 -
svt C:\SVT\SVT_I20.xml -blacklist C:\Daten\blacklist.txt
```

Information:

see LCS section

Set a new LCS value via diagnostic job

Command in client batch mode:

```
E-Sys.bat           -setLcs  -connection <config file>  -lcsName <LCS name>
 -lcsValue <LCS value>  [-noreset]  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -setLcs  -connection <config file>  -lcsName <LCS name>
 -lcsValue <LCS value>  [-noreset]  [-svt <SVT file>]  [-blacklist <file>]
 [-whitelist <file>]
```

Declaration:

-setLcs	command.
-connection <config file>	Absolute path to configuration file with the connection parameter (see Connection Parameter).
-lcsName <LCS name>	LCS-function in hex (e.g.: 0x01).
-lcsValue <LCS value>	LCS-function-value in hex (e.g.: 0x01).
-noreset	Do not perform an ECU reset after switching an LCS.
-svt <SVT file>	Absolute path to SVT-file.
-blacklist <file>	Absolute path to Blacklist-file.

-whitelist <file>	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -setLcs -connection C:\conf\connection.properties -lcsName 0x01 -
lcsValue 0x01 -noreset -svt C:\SVT\SVT_I20.xml -blacklist
C:\Daten\blacklist.txt
E-Sys.bat -setLcs -connection C:\conf\connection.properties -lcsName 0x01 -
lcsValue 0x01 -noreset -whitelist C:\Daten\whitelist.txt
E-Sys.bat -setLcs -connection C:\conf\connection.properties -lcsName 0x01 -
lcsValue 0x01 -svt C:\SVT\SVT_I20.xml
```

Information:

see LCS section

Set a new LCS value via secure token

Command in client batch mode:

```
E-Sys.bat           -writeLcsToken  -connection <config file>  -tokenset
<SecureToken file>  [-noreset]  [-svt <SVT file>]  [-blacklist <file>]  [-
whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -writeLcsToken  -connection <config file>  -tokenset
<SecureToken file>  [-noreset]  [-svt <SVT file>]  [-blacklist <file>]  [-
whitelist <file>]
```

Declaration:

-writeLcsToken	command.
-connection <config file>	Absolute path to configuration file with the connection parameter (see Connection Parameter.
-tokenset <SecureToken file>	Absolute path to secure-token-set (response-file [JSON]).
-noreset	Do not perform an ECU reset after switching an LCS.
-svt <SVT file>	Absolute path to SVT-file.
-blacklist <file>	Absolute path to Blacklist-file.
-whitelist <file>	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -writeLcsToken -connection C:\conf\connection.properties -tokenset
C:\Daten\token-3730977-000101-000102030405060708090A0B0C0D0EEE.json -noreset
```

```
-whitelist C:\Daten\whitelist.txt
E-Sys.bat -writeLcsToken -connection C:\conf\connection.properties -tokenset
C:\Daten\token-3730977-000101-000102030405060708090A0B0C0D0EEE.json -svt
C:\SVT\SVT_I20.xml
E-Sys.bat -writeLcsToken -connection C:\conf\connection.properties -tokenset
C:\Daten\token-3730977-000101-000102030405060708090A0B0C0D0EEE.json -noreset
-svt C:\SVT\SVT_I20.xml -blacklist C:\Daten\blacklist.txt
```

Information:

see LCS section

Server-client batch mode

General:

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Start E-Sys-Batch-Server

Command to start an E-Sys-Batch-Server.

Command in client batch mode:

```
E-Sys.bat  -startserver
or
E-Sys.bat  -startserver [-prop <property file>] [-setProxy
<configuration file> ] [-authenticationCoding ( -user <username> |
-useSwlSecCertificate ) -connection <bmw_intranet|internet>] [-
dialog]
```

Command in client-server batch mode:

```
---Only applicable in client-operation---
```

Declaration:

`-startserver`	command um den Server zu starten.
`-prop <property file>`	Absolute path to the E-Sys-property-file to be used.
`-setProxy <config file>`	Absolute path to Proxy-Configuration-file.
`-authenticationCoding`	Argument.
`-user <username>`	Backend-Login via username (example: max.mustermann).
`-useSwlSecCertificate`	Backend-Login via Certificate.
`-connection <bmw_intranet|internet>`	Login via `bmw_intranet` (inside BMW-network) or `internet` (outside BMW-network).
`-dialog`	Backend-Login with GUI-dialog.

Example:

```
E-Sys.bat -startserver
E-Sys.bat –startserver -prop C:\config\Esys.properties -
authenticationCoding -user max.mustermann -connection intranet -
dialog
E-Sys.bat –startserver -prop C:\config\Esys.properties -setProxy
C:\data\proxyConfig.config -authenticationCoding -
useSwlSecCertificate -connection internet
```

Information:

- If the parameter `-authenticationCoding` is set, the E-Sys server is only started if the SWL-Sec2 is successfully authenticated. See the details of the parameter below.
- If the parameter `-setProxy` is set, the E-Sys-server is only started if SWL-Sec2 can be reached via the proxy-server. For details on the parameter, see Batch/Proxy.
- If the switch `-dialog` is included, a GUI-dialog is displayed in which login data must be entered.
- It makes no sense to specify the property-file (`-prop <property file>`) in connection with a client-request to the server (`-server`) because the property-file is not used by the server. The server uses the property-file loaded at server-startup.

- The following codes are returned: 0 - The action was successful.
  1 - Errors occurred so that the action could not be completed.
  2 - The action was performed with errors.
  3 - Unable to connect to the server.

Stop E-Sys-Batch-Server

Command to exit the E-Sys-Batch-Servers.

Command in client batch mode:

```
------Only applicable in server-client-operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -stop
```

Declaration:

`-server`	command for Server-Client-operation.
`-stop`	command to stop the Server.

Example:

```
E-Sys.bat -server -stop
```

Information:

- The following codes are returned:
  0 - The action was successful.
  1 - Errors occurred so that the action could not be completed.
  2 - The action was performed with errors.
  3 - Unable to connect to the server.

Check if an E-Sys-Batch-Server is active

Command to check if an E-Sys-Batch-Server has been started.

Command in client batch mode:

```
---Only applicable in server-client-operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -check
```

Declaration:

`-server`	command for Server-Client-operation.
`-check`	command to check the Server-Client-operation.

Example:

```
E-Sys.bat -server -check
```

Information:

- The following codes are returned:
  0 - The action was successful.
  1 - Errors occurred so that the action could not be completed.
  2 - The action was performed with errors.
  3 - Unable to connect to the server.

Establishing a vehicle-connection in server-client-operation

If you need a vehicle-connection for an action, it must be established with `-openconnection`.

Command in client batch mode:

```
---Only applicable in server-client-operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -openconnection <config file>
```

Declaration:

`-server`	command for Server-Client-operation.
`-openconnection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).

Example:

```
E-Sys.bat -server -openconnection C:\config\Client-
Server_Server.config
```

Information:

- Not all actions require vehicle-connection-parameters. If any are specified, they will not lead to an error, but will be ignored.

- The following codes are returned:
  0 - The action was successful.
  1 - Errors occurred so that the action could not be completed.

2 - The action was performed with errors.
3 - Unable to connect to the server.

Disconnect vehicle-connection in server-client-operation

Command in client batch mode:

```
---Only applicable in server-client-operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -closeconnection
```

Declaration:

`-server`	command for Server-Client-operation.
`-closeconnection`	command to close the vehicle-connection.

Example:

```
E-Sys.bat -server -closeconnection
```

Information:

- The following return codes are returned:
  0 - The action was successful.
  1 - Errors occurred so that the action could not be completed.
  2 - The action was performed with errors.
  3 - Unable to connect to the server.

Cancel ongoing TAL-processing

Command to cancel a running TAL-processing

Command in client batch mode:

```
---Only applicable in server-client-operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -aborttalexecution
```

Declaration:

`-server`	command für Server-Client-Betrieb.
`-aborttalexecution`	command to cancel a running TAL-processing.

Example:

```
E-Sys.bat -server -aborttalexecution
```

Information:

- The following codes are returned:
  0 - The action was successful.
  1 - Errors occurred so that the action could not be completed.
  2 - The action was performed with errors.
  3 - Unable to connect to the server.

Expiration example for a server-client-operation

```
   E-Sys.bat –startserver
   E-Sys.bat –server -check
   E-Sys.bat -server -authenticationCoding -user <username> -
connection <Backend-Connection> -setProxy <configuration-file> -
dialog
   E-Sys.bat –server -openconnection <configuration-file>
   E-Sys.bat –server –talexecution <configuration-file>
   :::
   E-Sys.bat –server –talexecution <configuration-file>
   E-Sys.bat –server –closeconnection
   E-Sys.bat –server -stop
```

General:

It converts OrderXML-files (vehicle order) from a specified path to FA (vehicle-order-files). The file-names of the OrderXML-files must correspond to pattern A<xxxxxxx>.xml (x = letter or digit).

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Convert OrderXML to FA

Command in client batch mode:

```
E-Sys.bat           -convertorderxml <FO directory>  -out <FA directory>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -convertorderxml <FO directory>  -out <FA directory>
```

Declaration:

`-convertorderxml <FO directory>`	Absolute path to the directory with the FO files.
`-out <FA directory>`	Absolute path to the directory for the storage of the FA files.

Example:

```
E-Sys.bat -convertorderxml c:\Data\FO -out C:\Data\FA
```

Information:

- For each OrderXML-file, an FA-file is created with the file-name pattern "<OrderXML>_FA_<Timestamp>.xml".
  OrderXML..... Name of OrderXML
  Timestamp..... Time stamp according to the pattern ddmmyyyyhhmm

- The following codes are returned:
  0 - The processing was completed successfully.
  1 - Errors have occurred, so no verification was possible.
  2 - Errors have occurred while processing individual files. It is possible that individual files have been converted correctly.

General:

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Batch-command for executeFv

Command in client batch mode:

```
E-Sys.bat           -executeFv <config file> [-ignoreBATHAF]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -executeFv <config file> [-ignoreBATHAF]
```

Declaration:

`-executeFv <config file>`	Absolute path to ecexuteFv-Configuration-file.
`-ignoreBATHAF`	Optional parameter. Ignores the S1-switch.

Example:

```
E-Sys.bat -executeFv C:\data\executeFV.config -ignoreBATHAF
```

Information:

- Information about the batch-command `-executeFv` can be optained from the "executeFv manual".
- If a BAT/HAF test reveals that switch S1 is open, the further execution of the command depends on parameter `-ignoreBATHAF`.

  `-ignoreBATHAF` has been set:
  The flash sequence continues. However, the user must be aware that the execution can lead to interruptions. In this case, the TAL-processing cannot be ended with value 0 (OK), but at best with value 4 (with warning).

`-ignoreBATHAF` not set:
The execution of the command `-talexecution` is canceled.

General:

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

PDX-Import.

This command can be used to import containers (KC, IC or PDX template) into the programming system.

Command in client batch mode:

```
E-Sys.bat           -pdximport <PDX container file>  -project <project name>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -pdximport <PDX container file>  -project <project name>
```

Declaration:

`-pdximport <PDX container file>`	Absolute path of the container to be imported.
`-project <project name>`	Name of the project to import a container into.

Example:

```
E-Sys.bat -pdximport C:\Data\ODX\TEST__03__I020.001_023_015.pdx -project TEST__03__I020_001_023_015
```

Important additional information for partial KCs:

If you want to import a constructed (maybe by yourself) you have to keep in mind, that the referenced software unit(s) (SWE) won't be impor
reference and did not exist as a real data at the Container, so it is not possible to import the real data.
In case that the referenced software unit(s) are going to be needed, they have to imported by an other container, in which the needed softwa

Information:

- The following codes are returned:
  0 - The action was successful.
  1 - Errors occurred so that the action could not be completed.
  2 - The action was performed with errors.
  3 - Unable to connect to the server.
  4 - The action was performed without errors but with warnings.

PDX-Update

This command can be used to update containers with a PDX-template.

Command in client batch mode:

```
E-Sys.bat           -pdxupdate <PDX container file>  [-out <PDX container file>]  -template <PDX templa
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -pdxupdate <PDX container file>  [-out <PDX container file>]  -template <PDX templa
```

Declaration:

`-pdxupdate <PDX container file>`	Absolute path of the container whose template is to be updated.
`-out <PDX container file>`	Absolute path of the container to be created. If the `-out` paramete the same name after the update.
`-template <PDX template file>`	Absolute path of the PDX-template, which the container should c

Example:

```
E-Sys.bat -pdxupdate C:\Data\ODX\TEST__03__I020.001_023_015.pdx -out C:\Data\ODX\TEST__03__I020.001_023
C:\Data\ODX\PT01_template.003_040_000.pdx
```

Information:

- The following codes are returned:
  0 - A PDX container was created without errors.
  1 - Errors occurred so that no PDX container could be created.
  2 - A PDX container with errors was created.

Create PDX-component-containers

This command can be used to create component-containers.

Partial PDX-component-container:
It is also possible to create a partial PDX-component-container. Here, a md5 hash value is used to refere
be assigned to this PDX-component-container. These referenced software-units have already been succe
previous PDX-component-container and are already available there.
Further information can be found in the instructions ESys_Handlungsanweisung_partieller_KC.pdf.

Command in client batch mode:

```
E-Sys.bat           -pdxcontainer <PDX container definition file>  -out <PDX container file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -pdxcontainer <PDX container definition file>  -out <PDX container file>
```

Declaration:

`-pdxcontainer <PDX container definition file>`	Absolute path of the definition-file.
`-out <PDX container file>`	Absolute path of the PDX-container to be generated.

Example:

```
E-Sys.bat -pdxcontainer C:\Data\pdx_def.xml -out C:\Data\ODX\TEST__03__I020.001_023_020.pdx
```

Information:

- The PDX-container-definition-file describes the contents of the component-container. It is read and a co
  and stored below the specified path.
- SWE files (BTLD, SWFL, SWFK, FLSL) of a specific ECU can be defined by reference only (see `SWERefe`
  created container and are not present in the container.
  The software unites, that are listed under the reference (`SWEReferenceList`) are no real existing software uni

- A simple definition-file has the following content:

```
<PDXContainerDefinition xmlns="http://bmw.com/2010/esys.data.pdxcontainerdefinition">
    <PDXTemplate>C:/Data/ODX/F001_template.002_006_009.pdx</PDXTemplate>
    <BRVList>
        <BRV>F001</BRV>
        <BRV>F010</BRV>
    </BRVList>
    <ECUVariantList>
        <ECUVariant name="ECU1">
            <BaseVariant>EVALBOARD</BaseVariant>
            <SWEList> <!--List of software unites, that are part of the container (existing at the container)-->
                <File>C:/Data/SWE/btld_01020304_002_000_000.bsw</File>
            </SWEList>
            <SWEReferenceList> <!--List of software unites that are no part of the container (the hash value reference only
                <TE checksum="735208badabcd1e447c217f30aa621b0" descTableStartAddress ="1">btld_01234567_004_005_006</TE>
                <TE checksum="d8462a1b22db2c88b29bc6dc989b2004" descTableStartAddress ="13FE44">swfl_01234567_007_005_001<
            </SWEReferenceList>
```

```
            </ECUVariant>
        </ECUVariantList>
    </PDXContainerDefinition>
```

- The XML-file is defined by the PDXContainerDefinition.xsd schema (see <E-Sys-Installationsverzeichn elements and describes them in more detail in comentaries that can be used in the definition file.
- As in the command-line, all path information of files and directories must be absolute in the definition-fil
- If errors occur, an error-file (named errors... txt) will be create. It is placed in the directory where the PD error messages of the errors that occurred.

- The following codes are returned:
  0 - A PDX container was created without errors.
  1 - Errors occurred so that no PDX container could be created. An error file has been created.
  2 - A PDX container with errors was created. An error file has been created.

Merge multiple PDX-Container

This command can be used to package multiple component-containers into one component-container.

Command in client batch mode:

```
E-Sys.bat           -mergepdxcontainer <PDX container definition file>  -out <PDX container file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -mergepdxcontainer <PDX container definition file>  -out <PDX container file>
```

Declaration:

`-mergepdxcontainer <PDX container definition file>`	Absolute path of the definition-file.
`-out <PDX container file>`	Absolute path of the PDX-container to be generated.

Example:

```
E-Sys.bat -mergepdxcontainer C:\Data\pdx_merge.xml -out C:\Data\ODX\TEST__03__I020.001_023_020.pdx
```

Information:

- The PDX-container-definition-file describes the contents of the component-container. The file contains t component-container (element `<PDXTemplate>`), and the paths of the to merge PDX-containers (element `<ContainerList>`). The file is read and a corresponding PDX-container is creat specified on the command-line

- A simple definition-file has the following content:

```
<PDXContainerDefinitionMerge xmlns="http://bmw.com/2010/esys.data.pdxcontainerdefinition">
       <PDXTemplate>C:/Data/ODX/F001_template.002_006_009.pdx</PDXTemplate>
       <ContainerList>
              <File>C:/Data/ODX/F001_EKPM__03.001.000.001.pdx</File>
              <File>C:/Data/ODX/F001_DME__MSD85.001.001.002.pdx</File>
              <File>C:/Data/ODX/F001_EGS__GKEB23.002.008.003.pdx</File>
       </ContainerList>
</PDXContainerDefinitionMerge>
```

- The XML-file is defined by the PDXContainerDefinition.xsd schema (see <E-Sys-Installationsverzeichn elements and contains additional detail in the form of comments which can be used in the definition file
- As in the command-line, all path information of files and directories must be absolute in the definition-fil
- If errors occur, an error-file (named errors... txt) will be create. It is placed in the directory where the PD error messages of the errors that occurred.

- The following codes are returned:
  0 - A PDX container was created without errors.
  1 - Errors occurred so that no PDX container could be created. An error file has been created.
  2 - A PDX container with errors was created. An error file has been created.

- Not supported:
  - ECU-Variant-Patterns
  - Documentation files of component containers
  - Merging containers with the same ECU variants

Proxy

General:

If an active proxy has been defined in the properties (proxy.useProxy=**true**), it is taken into account in both client- and server-client-based E-Sys-batch calls, **even if you did not use a -setProxy in the server-client-batch.**
If the proxy-server requires authentication (proxy.authRequired=**true** in the Properties), you are prompted to enter the login and password for the proxy.
Only Basic-authentication is currently supported as an authentication method on the proxy server.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Read Proxy-Status

The current Proxy-Status is output to stdout.

Command in client batch mode:

```
E-Sys.bat           -readStatusProxy
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readStatusProxy
```

Declaration:

`-readStatusProxy`	command.

Example:

```
E-Sys.bat -readStatusProxy
```

Information:

- Here is an example edition:

```
A proxy is used:
```

```
https://web-proxy.de:8080
No proxy-authentication required
Do not use proxy on the following URLs:
https://bmw.de
https://bmwgroup.com
```

Setting a Proxy

The command provides the required proxy information (proxy-host, proxy-port, proxy-authentication, username, password, URL-blacklist) E-Sys.
An active proxy is considered for all backend calls.

Command in client batch mode:

```
---Only applicable in server-client operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -setProxy <configuration file>  [-dialog]
```

Declaration:

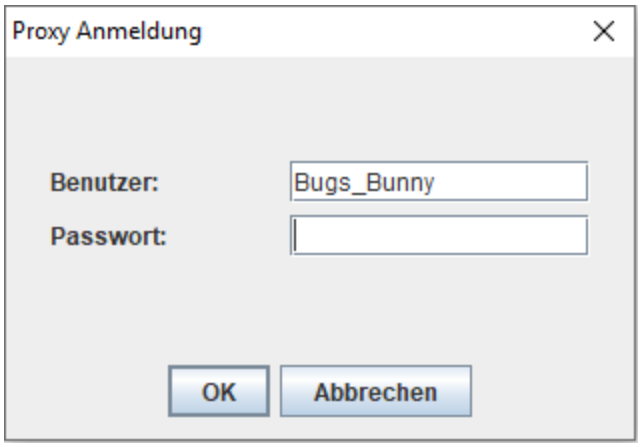
`-setProxy <config file>`	Absolute path to the configuration-file with the proxy-server-parameters.
`-dialog`	Backend-Login with GUI-dialog.

Example:

```
E-Sys.bat -server -setProxy c:\data\proxyConfig.config
E-Sys.bat -server -setProxy c:\data\proxyConfig.config -dialog
```

Information:

- If the switch `-dialog` is defined, a GUI-dialog is displayed in which authentication-data must be entered.
- The authentication-dialog looks like this:



- The format of the configuration-file to pass is defined as follows:

```
# Proxy-URL e.g.: https://web-proxy.de
PROXY_HOST = <URL>

# Proxy-Port z.B. 8080
PROXY_PORT = <Port>

# Proxy authentication required
# DEFAULT = off
PROXY_AUTH = <on|off>

# PROXY_USERNAME is mandatory parameter if PROXY_AUTH = on
PROXY_USERNAME = <Proxy ID>

# PROXY_EXCLUDE_LIST is an optional parameter, but if it is
used, the file must be in the file path
# A list of the beckend-URLs for which an active proxy should
not be used.
PROXY_EXCLUDE_LIST = <file-path to the list>
```

- **The command is only supported in server-client-mode.**
- Necessary parameters are `PROXY_HOST` and `PROXY_PORT`.
- Optional parameters are `PROXY_AUTH`, `PROXY_USERNAME`, and `PROXY_EXCLUDE_LIST`. If owithted, the information from the E-Sys.properties-file is read or default settings are used.
- The set proxy-settings are stored in the properties (except `PROXY_USERNAME`).
- The value of the `PROXY_USERNAME` is used only for the course of the

E-Sys-session. There is no Proxy-ID storage.

- If `PROXY_AUTH = set on`, the password for proxy must be entered manually in the command-line of the E-Sys-server.

Turn off Proxy

Do not use a set Proxy anymore.

Command in client batch mode:

```
---Only applicable in server-client operation---
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -removeProxy
```

Declaration:

`-removeProxy`	command.

Example:

```
E-Sys.bat -server -removeProxy
```

Information:

- **The command is only supported in server-client-mode.**
- In the Config-file, the use of Proxy is disabled, all other proxy settings remain.

General:

The FSC comfort functions are also available in batch mode.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

SWT-Status

This command determines the SWT status.

Command in client batch mode:

```
E-Sys.bat           -swtstatus  -project <project name>  -
diagaddress <diagnosis address>  -vehicleinfo <vehicle info>  -
basevariant <basevariant>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -swtstatus  -project <project name>  -
diagaddress <diagnosis address>  -vehicleinfo <vehicle info>  -
basevariant <basevariant>
```

Declaration:

`-swtstatus`	command.
`-project <project name>`	Project name for the connection.
`-diagaddress <diagnosis address>`	The diagnostic-address of the ECU (optional parameter).
`-vehicleinfo <vehicle info>`	Vehicle for connection.
`-basevariant <basevariant>`	The basevariant of the ECU (optional parameter).

Example:

```
E-Sys.bat -swtstatus -project I020_TEST__03__I020_001_023_015 -
diagaddress 0x33 -vehicleinfo I020 -basevariant TEST
```

Information:

SWT-Aktion

This command can be used to perform an SWT action.

Command in client batch mode:

```
E-Sys.bat           -swtaction <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -swtaction <config file>
```

Declaration:

`-swtaction <config file>`	Absolute path to configuration file.

Example:

```
E-Sys.bat -swtaction C:\conf\connection.properties
```

Information:

- The concrete SWT-action is defined in the configuration-file together with the Connection Parameter and the following parameters.

```
FA = <FA-Filename>
VIN = <VIN-String>

# write_vin|store|update|upgrade|deactivate
SWT_ACTION = <SWT Aktion>

BASE_VARIANT = <Basevariant>

DIAG_ADDRESS = <Diagnoseadress (hex)>

APPLICATION_NUMBER = <Applikationsnumber>

UPGRADE_INDEX = <Upgrade-Index>
```

```
FSC = <FSC-Filename>
```

- No concurrent indication of `FA` and `VIN` is allowed.
- The FSC is only necessary for the `store` and `upgrade` actions.
- For the `write_vin` action, the `VIN` parameter must be defined.
- The `write_vin` action does not take into account the parameters `APPLICATION_NUMBER` and `UPGRADE_INDEX`.
- The `store|update|upgrade|deactivate` actions must always have the parameters `BASE_VARIANT`, `DIAG_ADDRESS`, `APPLICATION_NUMBER`, and `UPGRADE_INDEX` defined.

TAL processing

General:

With this batch-mode you can program and encode ECUs (SWE and CAF).
Before you can start a TAL-processing in the batch, you must first import the appropriate container.
You can import PDX containers in GUI-mode (see PDX-Charger) or batch-mode (see PDX-Container).

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

TAL-processing

This command starts a TAL-processing.

Command in client batch mode:

```
E-Sys.bat           -talexecution <config file> [-ignoreBATHAF]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -talexecution <config file> [-ignoreBATHAF]
```

Declaration:

`-talexecution <config file>`	Absolute path to configuration file.
`-ignoreBATHAF`	Optional parameter. Ignores the S1-switch.

Example:

```
E-Sys.bat -talexecution C:\conf\connection.properties -ignoreBATHAF
```

Information:

- If a BAT/HAF test reveals that switch S1 is open, the further execution of the command depends on pa
  `-ignoreBATHAF` has been set:
  The flash sequence continues. However, the user must be aware that the execution can lead to interru
  be ended with value 0 (OK), but at best with value 4 (with warning).

  `-ignoreBATHAF` not set:
  The execution of the command `-talexecution` is canceled.
- In the configuration file, the following parameters can be defined in addition to the Connection Paramet

```
TAL = <TAL-Filename>
FA = <FA-Filename>
VIN = <VIN-String>

SVT = <SVT-Filename>

# fa|shipment|ncd (Default: fa)
CODING_TYPE = <Coding-Type>

# optional path indication where the NCDs are located
# Format <NCD-path 1>;<NCD-path 2>... , e.g. "C:/NCD/00001234_0000dddd.caf.026_000_002.ncd;C:/NCD/00
# Only considered for coding type "ncd". Caution: only one ECU may be handled during NCD encoding.
NCD_LIST = <NCD-path 1>;

# Perform FA-coding locally
# on|off OR true|false
USE_LOCAL_NCD = on
# Optional path where the NCDs are located, attention: in this directory a subfolder with the VIN as
# If owithted, the NCD directory is used, which is specified in the settings. (stored in properties)
# Only considered for coding type "fa" und "shipment".
PATH_NCD = C:/signedNCD/
```

```
# on|off
PARALLEL_PROGRAMMING = off

# on|off
CHECK_PROGRAMMING_COUNTER = on

REPEATS_ON_ERROR = <number of repetitions in case of failure>

# off|all|progress|transaction|error|tal|aep
EVENT= <Event-Filter for console output>

# on|off
FILL_INSTALLED_ECU_LIST = on

# merge|overwrite
FILL_INSTALLED_ECU_LIST_MODE = <Mode, how to fill the InstalledECUList>

# on|off
MODE_SWITCH_PROGRAMMING = <Mode-Switch during programming on/off>
# List of Gateways (diagnostic addresses in HEX), to be switched to programming mode.
# Without the parameter, all available gateways are switched.
# E.g.: 10,63
GATEWAYS_TO_SWITCH_TO_PROGRAMMING_MODE = <Gateway-List>

# Absolute path to TAL-Filter-file.
TAL_FILTER = <TAL-Filter-Filename>

# Absolute path to a backup-directory with Individual-data.
IDR_BACKUP_PATH = <IDR-Backup-Directory>

# on|off   (If the parameter is on, the mileage is read out before the TAL-processing and written in
READ_MILEAGE_FROM_GWSZ  = off

# Mileage for Fingerprint
# If a valid value is specified, this will be written in Fingerprint, independent of the mileage bei
# Default value '0' is used for Fingerprint, if neither MILEAGE nor READ_MILEAGE_FROM_GWSZ was indic
MILEAGE = <0-655330>

# URL of HDD-Update Servers
HDD_UPDATE_SERVER_URL = <URL for HDD-Update>

# on|off (Response on event during TAL-processing enable/disable)
ROE_DEACTIVATION = off

# on|off (prevent Http-transmission-protocol during TAL-processing)
PREVENT_HTTP_TRANSMISSION = off

# on|off (perform Optimized Bootloader flash on ECUs that support this functionality.
OPTIMIZABLE_BOOTLOADER_FLASH = off

# on|off (check expectedSgbmids)
EXPECTED_SGBMID_VALIDATION = on

# Optional Parameter.
# If specified, only client-batch-mode is considered
# Default value = off
USE_SWL_SEC_CERTIFICATE = <on|off>

# Optional Parameter. Considered only in client-batch-mode.
# If the parameter is owithted SWL_SEC_CONNECTION, the value from Esys.properties is used.
SWL_SEC_CONNECTION = <bmw_intranet|internet>

# Optional Parameter.
# List of ECUs (diagnostic addresses in HEX) for which no UDS fallback may be performed.
# Alternatively, 'all' can be specified to disable the UDS fallback for all ECUs.
# The enumerated diagnostic addresses are only separated by a comma. No space may be inserted betwee
# Without the parameter, the UDS flash is executed in case of an HTTP error.
# E.g.: 10,63
ECUS_TO_PREVENT_UDS_FALLBACK = < Diagnosis adresses in HEX> | all >
```

- Because of "safe coding", a connection to the SWL-Sec2 must be established before executing a TAL \
  This also means that the computer must have two NICs. Since E-Sys potentially has two different conn
  backend and vehicle).
- **A TAL with coding elements can only be run in E-Sys without server-client operation if:**
  - a client Certificate for SWL Security System is imported and the parameter `USE_SWL_SEC_CERTIFICATE` i
  - OR the required test-keys are available locally
  - OR the paramter `USE_LOCAL_NCD` is active (NCDs are signed locally)
  - OR at `CODING_TYPE=ncd` (NCDs are signed locally)
- Necessary parameters are `TAL`, `PROJECT`, `VEHICLEINFO` and `FA` or `VIN`.
- The configuration file can contain either an FA or a VIN, but not both. If an FA is specified, it must conta
- An FA is mandatory to handle a TAL with coding-actions (coding-TAs). Applies to `CODING_TYPE=fa` or `ship`
- CODING_TYPE = ncd allows you to encode an SG with an already existing NCD file. The option NCD_
- With CODING_TYPE = ncd, only one ECU may be encoded during a TAL-processing.
- If the configuration-file contains the parameter HDD_UPDATE_SERVER_URL, an **HDD-update** is perf
  performed.
- For the switches `PARALLEL_PROGRAMMING`, `CHECK_PROGRAMMING_COUNTER` and `FILL_INSTALLED_ECU_LIST` the value
- The default value for the parameters is the GUI values (from the Esys.properties file).
- A '\' may not be used in the configuration file. Instead, use '/' or '\\' for path information.
- The path-information for the parameters `TAL`, `FA` and `TAL_FILTER` can be both, absolute and relative. For r
  the location of the config-file.
- All other E-Sys settings are read from the Esys.Properties file.
- The history of the TAL-processing is stored in the E-Sys log file.

TAL calculation

General:

A TAL is calculated from an SVT-actual- and an SVT-target-file.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

TAL-calculation

This command calculates a TAL

Command in client batch mode:

```
E-Sys.bat           -talcalculation <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -talcalculation <config file>
```

Declaration:

`-talcalculation <config file>`	Absolute path to configuration file.

Example:

```
E-Sys.bat -talcalculation C:\conf\connection.properties
```

Information:

- In the configuration-file, the following parameters can be defined in addition to the Connection Parameter:

```
# Aboslute path of the SVT-Actual file
SVT-Actual = <SVT-Actual-file>

# Aboslute path of the SVT-Target file
SVT-Target = <SVT-Target-file>

# Aboslute path of the generated TAL file
GENERATED_TAL = <TAL-filename>
```

```
# Aboslute path to a TAL-Filter file.
TAL_FILTER = <TAL-Filter-filename>

# normal | idrBackup | idrRestore
TAL_TYPE = <TAL-Type>

# Aboslute path to a Backup-Directory with Individual-data.
IDR_BACKUP_PATH = <IDR-Backup-Directory>
```

- Required parameters are `SVT-Actual`, `SVT-Target`, and `GENERATED_TAL`.
- In standalone mode, additional connection parameters are required (see Connection Parameter)
- In server-client-mode, the server must have a PSdZ-connection open.
- The default value for the parameters is the GUI values (from the `Esys.properties file`).
- A '\' may not be used in the configuration file. Instead, use '/' or '\\' for path information.
- If the parameter `TAL_TYPE` has the value `idrRestore`, the parameter `IDR_BACKUP_PATH` must also be defined.

Example TAL Filter:

```
<?xml version="1.0" encoding="UTF-8"?>
<talfilter>
   <allEcu>
       <blUpdate>empty</blUpdate>
       <cdDeploy>empty</cdDeploy>
       <fscDeploy>empty</fscDeploy>
       <hwDeinstall>empty</hwDeinstall>
       <hwInstall>empty</hwInstall>
       <idBackup>empty</idBackup>
       <idDelete>empty</idDelete>
       <idRestore>empty</idRestore>
       <swDeploy>empty</swDeploy>
       <ibaDeploy>empty</ibaDeploy>
```

```
        <hddUpdate>empty</hddUpdate>
        <gatewayTableDeploy>empty</gatewayTableDeploy>
    </allEcu>
    <ecuFilter>
        <!-- decimal diagnostic address -->
        <diagAdr>10</diagAdr>
        <setting>
            <blUpdate>mustBeTreated</blUpdate>
            <cdDeploy>mustNotBeTreated</cdDeploy>
            <fscDeploy>mustNotBeTreated</fscDeploy>
            <hwDeinstall>allowedToBeTreated</hwDeinstall>
            <hwInstall>mustNotBeTreated</hwInstall>
            <idBackup>mustNotBeTreated</idBackup>
            <idDelete>mustNotBeTreated</idDelete>
            <idRestore>mustNotBeTreated</idRestore>
            <swDeploy>mustNotBeTreated</swDeploy>
            <ibaDeploy>mustNotBeTreated</ibaDeploy>
            <hddUpdate>mustNotBeTreated</hddUpdate>

 <gatewayTableDeploy>mustNotBeTreated</gatewayTableDeploy>
        </setting>
    </ecuFilter>
</talfilter>
```

General:

2 TSL functions are supportet in Batch-Mode.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Read TSL-Status

The TSL-Status is output to stdout

Command in client batch mode:

```
E-Sys.bat           -tslstatus  -connection <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -tslstatus  -connection <config file>
```

Declaration:

`-tslstatus`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).

Example:

```
E-Sys.bat -tslstatus -connection C:\conf\connection.properties
```

Information:

TSL-Update

Command in client batch mode:

```
E-Sys.bat           -updatetsl <SVT filename>  -connection <config
file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -updatetsl <SVT filename>  -connection <config
file>
```

Declaration:

`-updatetsl <SVT filename>`	Absolute path to The SVT-file.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Connection Parameter).

Example:

```
E-Sys.bat -updatetsl C:\data\svt.xml -connection
C:\conf\connection.properties
```

Information:

The following codes are returned:
0 - The processing was completed successfully.
1 - TSL update could not be started
2 - TSL update with errors

VCM (Vehicle Configuration Management)

General:

The following data can be read and written from the VCM (VCM-Master or VCM-Backup): FA, FP, SVT-Ta

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Write VCM-Master (FA, I-Step or SVT-Target)

Command in client batch mode:

```
E-Sys.bat           -writeVcmMaster <FA|ISTUFEN|SVTSOLL>  [-connection <config file>]  -in <source file
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -writeVcmMaster <FA|ISTUFEN|SVTSOLL>  [-connection <config file>]  -in <source file
```

Declaration:

`-writeVcmMaster <FA|ISTUFEN|SVTSOLL>`	Parameters that can take the value `FA` , `ISTUFEN` or `SVTSOLL`.
`-connection <config file>`	Absolute path to configuration file with the connection paran
`-in <source file>`	Absolute path for "FA-file", "text file with I-step" or "SVT-Targ

Example:

```
E-Sys.bat -writeVcmMaster FA -connection C:\conf\connection.properties -in C:\Data\FA\FA_Y120BX.xml
E-Sys.bat -writeVcmMaster ISTUFEN -in C:\Data\istufen_Y120BX.txt
E-Sys.bat -writeVcmMaster SVTSOLL -connection C:\conf\connection.properties -in C:\Data\SVT\SVT_SOLL_Y1
```

Information:

- The I-Steps are to be passed in the "Text-file with I-Steps" in the following format:

```
<Current I-Step>
<Last I-Step>
<Auslieferungs-I-Step>
```

- The I-Steps are specified in the usual format, e.g. `S15A-15-07-500`

Write VCM-Backup (FA or I-Step)

Command in client batch mode:

```
E-Sys.bat           -writeVcmBackup <FA|ISTUFEN>  [-connection <config file>]  -in <source file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -writeVcmBackup <FA|ISTUFEN>  [-connection <config file>]  -in <source file>
```

Declaration:

`-writeVcmBackup <FA|ISTUFEN>`	Parameters that can take the value `FA` or `ISTUFEN`.
`-connection <config file>`	Absolute path to configuration file with the connection paran
`-in <source file>`	Absolute path for "FA file" or "text file with I-Steps".

Example:

```
E-Sys.bat -writeVcmBackup FA -connection C:\conf\connection.properties -in C:\Data\FA\FA_Y120BX.xml
E-Sys.bat -writeVcmBackup ISTUFEN -connection C:\conf\connection.properties -in C:\Data\istufen_Y120BX.
```

Information:

- The I-Steps are to be passed in the "Text-file with I-Steps" in the following format:

```
<Current I-Step>
<Last I-Step>
<Auslieferungs-I-Step>
```

- The I-Steps are specified in the usual format, e.g. `S15A-15-07-500`

Read VCM-Master (FA, FP, I-Steps or SVT-Target)

Command in client batch mode:

```
E-Sys.bat           -readVcmMaster <FA|FP|ISTUFEN|SVTSOLL>  [-connection <config file>]  -out <target d
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readVcmMaster <FA|FP|ISTUFEN|SVTSOLL>  [-connection <config file>]  -out <target d
```

Declaration:

`-readVcmMaster <FA|FP|ISTUFEN|SVTSOLL>`	Parameters that can take the value `FA`, `FP` , `ISTUFEN` or `SVTSOL`
`-connection <config file>`	Absolute path to configuration file with the connection paran
`-out <target directory>`	Absolute path for FA, FP, I-Steps- or SVT-Target-file with or

Example:

```
E-Sys.bat -readVcmMaster FA -connection C:\conf\connection.properties -out C:\Data\FA
E-Sys.bat -readVcmMaster FA -connection C:\conf\connection.properties -out C:\Data\FA_123.xml
or
E-Sys.bat -readVcmMaster FP -connection C:\conf\connection.properties -out C:\Data\FP
E-Sys.bat -readVcmMaster FP -connection C:\conf\connection.properties -out C:\Data\FP\FP_123.xml
or
E-Sys.bat -readVcmMaster ISTUFEN -connection C:\conf\connection.properties -out C:\Data
E-Sys.bat -readVcmMaster ISTUFEN -connection C:\conf\connection.properties -out C:\Data\IS_123.txt
or
E-Sys.bat -readVcmMaster SVTSOLL -connection C:\conf\connection.properties -out C:\Data\SVT -connection
E-Sys.bat -readVcmMaster SVTSOLL -connection C:\conf\connection.properties -out C:\Data\SVT\SVTSOLL_123
```

Information:

- `FA`.....If no file name is specified, the selected FA with the following file name: FA_Master_<
- `FP`.....If no file name is specified, the selected FP with the following file name: FP_Master_<
- `I-Stufe`.....If no file name is specified, the destination directory contains the i-levels read with the following file name: ISTUFEN_Master_<VIN>_<Timestamp>.txt
- `SVT-Target`.....If no file name is specified, the selected SVTSOLL with the following file name: SVTSO destination directory.

Read VCM-Backup (FA or I-Step)

Command in client batch mode:

```
E-Sys.bat           -readVcmBackup <FA|ISTUFEN>  [-connection <config file>]  -out <target directory>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readVcmBackup <FA|ISTUFEN>  [-connection <config file>]  -out <target directory>
```

Declaration:

`-readVcmBackup`	

<FA|ISTUFEN>.....	Parameters that can take the value FA or ISTUFEN.
-connection <config file>.....	Absolute path to configuration file with the connection parar
-out <target directory>.....	Absolute path for FA or I-Step file with or without file-name.

Example:

```
E-Sys.bat -readVcmBackup FA -connection C:\conf\connection.properties -out C:\Data\FA
E-Sys.bat -readVcmBackup FA -connection C:\conf\connection.properties -out C:\Data\FA\FA_123.xml
or
E-Sys.bat -readVcmBackup ISTUFEN -connection C:\conf\connection.properties -out C:\Data
E-Sys.bat -readVcmBackup ISTUFEN -connection C:\conf\connection.properties -out C:\Data\IS_123.txt
```

Information:

- FA. .....If no file name is specified, the selected FA with the following file name: FA_Master_<VI
- I-Stufe. .....If no file name is specified, the destination directory contains the read I levels with the foll

Read VIN from VCM-Master

Command in client batch mode:

```
E-Sys.bat           -readVinFromMaster  [-connection <config file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readVinFromMaster  [-connection <config file>]
```

Declaration:

-readVinFromMaster	command.
-connection <config file>	Absolute path to configuration file with the connection paran

Example:

```
E-Sys.bat -readVinFromMaster -connection C:\conf\connection.properties
```

Information:

- VIN is output to stdout

Read VIN from VCM-Backup

Command in client batch mode:

```
E-Sys.bat           -readVinFromBackup  [-connection <config file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readVinFromBackup  [-connection <config file>]
```

Declaration:

-readVinFromBackup	command.
-connection <config file>	Absolute path to configuration file with the connection paran

Example:

```
E-Sys.bat -readVinFromBackup -connection C:\conf\connection.properties
```

Information:

- VIN is output to stdout

- The following codes are returned: 0 - The processing was completed successfully.
  1 - Errors have occurred.

Certificate Management (SP18, SP21, etc.)

General:

For Certificate Management, batch offers the following commands.

[ ].....optional, can be specified, but does not have to be
( ).....Selection, must be specified
| ....."or", is used in a selection

Read Certificate

With the following batch command, it is possible to read from the ECUs according to the container type pr
Attention!
Not all ECUs support reading of all container types.

Command in client batch mode:

```
E-Sys.bat           -readCERT  -connection <config file>  -type <Certificate type>  [-svt <SVT file>]
<file>]  [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -readCERT  -connection <config file>  -type <Certificate type>  [-svt <SVT file>]
<file>]  [-whitelist <file>]
```

Declaration:

`-readCERT`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see C Parameter).
`-type <Certificate type>`	Type of Certificate to read: `CERTIFICATE`, `BINDING` , `ONLINE_CERTIFICATES_EC SEC_OC_KEYLIST`.
`-svt <SVT file>`	Absolute path to SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -readCERT -connection C:\conf\connection.properties -type CERTIFICATE -blacklist C:\Daten\bla
E-Sys.bat -readCERT -connection C:\conf\connection.properties -type BINDING -svt C:\SVT\SVT_I20.xml -wh
C:\Daten\whitelist.txt
E-Sys.bat -readCERT -connection C:\conf\connection.properties -type ONLINE_CERTIFICATES_ECU -svt C:\SVT
blacklist C:\Daten\blacklist.txt -whitelist C:\Daten\whitelist.txt
E-Sys.bat -readCERT -connection C:\conf\connection.properties -type SEC_OC_KEYLIST
```

Information:

Generating a Certificate request

This command generates a request file to the CBB. This can then be passed on to the corresponding pro
interface, mail).

Command in client batch mode:

```
E-Sys.bat           -generateCSR  -connection <config file>  -out <target file>  [-secOCKeys]  [-vin <V
blacklist <file>]  [-whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -generateCSR  -connection <config file>  -out <target file>  [-secOCKeys]  [-vin <V
blacklist <file>]  [-whitelist <file>]
```

Declaration:

`-generateCSR`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see (Parameter).
`-out <target file>`	Absolute path to the target file.
`-secOCKeys`	Ability residual bus Keys. Extension of CBB request file with a virtual E
`-vin <VIN17>`	17-digit VIN.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -generateCSR -connection C:\conf\connection.properties -out C:\Data\CERT\requestCBB[JSON].txt
C:\Data\CERT\WBA12345671234567_CBBRequest_Restbussim_20210102_131515[JSON].txt
```

Information:

Send a Certificate request to the CBB and save the response (online)

This command sends a request file to the CBB. The obtained result is stored in a CBB response file.
This command can only be used in the BMW environment, as Certificates are obtained from one/more se

Command in client batch mode:

```
E-Sys.bat         -getCbbResponseFromRequest <cbb request file> [-out <target directory>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -getCbbResponseFromRequest <cbb request file> [-out <target directory>]
```

Declaration:

`-getCbbResponseFromRequest <cbb request file>`	Absolute path to CBB-Request-file.
`-out <target directory>`	Absolute path to the target file. If owithted, the default path is used response files.

Example:

```
E-Sys.bat -getCbbResponseFromRequest C:\Daten\WBAAE810X0H123456_CBBRequest.txt -out C:\cert\cert.txt
```

Information:

Writing the bindings from file

This command writes all bindings, Certificates, and bindings from the specified response file from the CBI connected vehicle.
The "Other Bindings" are also generated and also written into the vehicle.

The `-secOCKeys` option also displays a file for residual bus simulation.

Command in client batch mode:

```
E-Sys.bat           -writeBindings  -connection <config file>  -in <CBB response file>  [-secOCKeys]  [
 [-secOCKeysPath <SecOCKeyPack file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -writeBindings  -connection <config file>  -in <CBB response file>  [-secOCKeys]  [
 [-secOCKeysPath <SecOCKeyPack file>]
```

Declaration:

`-writeBindings`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Parameter).
`-in <CBB response file>`	Absolute path to file.
`-secOCKeys`	Ability residual bus Keys. Creation of a file for residual bus simulation. response file is required.
`-svt <SVT file>`	Absolute path to SVT-file.
`-secOCKeysPath <SecOCKeyPack file>`	Absolute path to the target file with symmetric keys. Only connection v considered.

Example:

```
E-Sys.bat -writeBindings -connection C:\conf\connection.properties -in
C:\Daten\WBAAE810X0H123456_CBBResponse_20210331_125817[JSON].txt" -secOCKeysPath
C:\Data\CERT\Keys\20210102_131515_SecOC_KeyPack_plain.xml
```

Information:

- If the parameter `-secOCKeysPath <SecOCKeyPack target file>` is owithted, the default path is used

Checking the Certificates on the Certificate-enabled ECUs

This command can be used to initiate a Certificate check in the ECU.
After the internal inspection is complete, E-Sys waits for the longest returned inspection time of the ECUs
After that, the results are requested and written to result files.
The status is read out in parallel for several ECUs, taking into account the topology.

Command in client batch mode:

```
E-Sys.bat          -checkCERT  -connection <config file>  -retries <value>  [-svt <SVT file>]  [-black
whitelist <file>]
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -checkCERT  -connection <config file>  -retries <value>  [-svt <SVT file>]  [-black
whitelist <file>]
```

Declaration:

`-checkCERT`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see (Parameter).
`-retries <value>`	Number of performed scans (for 1 to a maximum of 10). example: `-ret`
`-svt <SVT file>`	Absolute path to SVT-file.
`-blacklist <file>`	Absolute path to Blacklist-file.
`-whitelist <file>`	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -checkCERT -connection C:\conf\connection.properties -retries 3
E-Sys.bat -checkCERT -connection C:\conf\connection.properties -retries 3 -blacklist C:\Daten\blacklist
E-Sys.bat -checkCERT -connection C:\conf\connection.properties -retries 3 -blacklist C:\Daten\blacklist
C:\Daten\whitelist.txt
```

Information:

Automated Certificate execution with configuration-file

The following command performs an automated Certificate execution. In addition, a file can be created fo
simulation.
This command can only be used in the BMW environment, as Certificates are obtained from one/more se
them to the connected vehicle.

Command in client batch mode:

```
E-Sys.bat           -certexecution <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -certexecution <config file>
```

Declaration:

`-certexecution <config file>`	Absolute path to configuration file.

Example:

```
E-Sys.bat -certexecution C:\Cert\CertExecution.config
```

Information:

- The specific certificate action is defined in the configuration file along with the Connection Parameter a
  parameters:

Parameter	Example value	Mandatory parameters?	Comme
SVT	C:/Data/SVT/SVT-Target.xml	yes	SVT Target
VIN	WBA12345671234567	yes	VIN17
CERT_BLACKLIST	C:/Data/etc/blacklist.txt	no	Indicates ECUs that sh handled
CERT_WHITELIST	C:/Data/etc/whitelist.txt	no	Specifies ECUs to be h exclusively
CERT_SERVER_URL_LIST	https\://www.maxmustermann.de\:<optional-Port-No.>/<Server-path>; https\://www.maxmustermann.com\:<optional-Port-No.>/<Server-path>;	yes	List of Backend-Server
CERT_RETRIES	5	no	Number of connection BMW-Backend; default: 1; maximum va
CERT_TIME_BETWEEN_RETRIES	30000	no	Duration connection att Backend: default: 20000; maximu
SECOC_KEYS	on	no	Ability residual bus Key Possible values: .on|off
SECOC_KEYS_PATH	C:\SecOC_KeyPack_plain.xml	no	Absolute path incl. fi target file with symmetr Only considered for SEC If owithted, the default Example: C:-Data-CERT-Keys-20210102_131515_SecOC_K
PROJECT	S18A_...	yes	Project name in psdz-d

VEHICLEINFO	S18A	yes	Baureihenverbund
CONNECTION	vin	yes	valid values: bus | gateway_url | icom_ icom_ethernet | vin
GATEWAY_VIN	WBA12345671234567_DIAGADR10	ja, if CONNECTION=vin	valid values: <VIN17>_DIAGADR<Gateway AUTODETECT

Setting the Freshness Value of a specific SecOC PDU

This command can be used to write a new SecOC Counter value to the ECU.
Either the black/white mechanism may be used as a parameter (blacklist and/or whitelist) or diagaddres
If neither diagaddress nor blacklist/whitelist is passed, the new SecOc Counter value is set for all SecOc

Command in client batch mode:

```
E-Sys.bat           -setSecOCCounter  -connection <config file>  -value <hex value>  { [-diagaddress <d
address>]  |  {[-blacklist <file>]  [-whitelist <file>] } }
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -setSecOCCounter  -connection <config file>  -value <hex value>  { [-diagaddress <d
address>]  |  {[-blacklist <file>]  [-whitelist <file>] } }
```

Declaration:

-setSecOCCounter	command.
-connection <config file>	Absolute path to configuration file with the connection parameter (see Parameter).
-value <hex value>	Counter-Value as HEX in range of 0x0 to 0xFFFFFFFF.
-diagaddress <diagnosis address>	The diagnostic address of the ECU.
-blacklist <file>	Absolute path to Blacklist-file.
-whitelist <file>	Absolute path to Whitelist-file.

Example:

```
E-Sys.bat -setSecOCCounter -connection C:\conf\connection.properties -value 0x01 -diagaddress 0x33
E-Sys.bat -setSecOCCounter -connection C:\conf\connection.properties -value 0x01 -blacklist C:\Daten\bl
whitelist C:\Daten\whitelist.txt
```

Information:

Activation of IPsec on all IPsec-enabled ECUs

This command enables IPsec to be enabled for all ECUs that support this functionality

Command in client batch mode:

```
E-Sys.bat           -activateIPsec  -connection <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -activateIPsec  -connection <config file>
```

Declaration:

-activateIPsec	command.
-connection <config file>	Absolute path to configuration file with the connection parameter (see Parameter).

Example:

```
E-Sys.bat -activateIPsec -connection C:\conf\connection.properties
```

Information:

Deactivate IPsec on all IPsec-enabled ECUs

This command deactivates IPsec for all ECUs that support this functionality

Command in client batch mode:

```
E-Sys.bat           -deactivateIPsec  -connection <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -deactivateIPsec  -connection <config file>
```

Declaration:

`-deactivateIPsec`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see Parameter).

Example:

```
E-Sys.bat -deactivateIPsec -connection C:\conf\connection.properties
```

Information:

Lock IPsec on all IPsec-enabled ECUs

This command can be used to lock IPsec to all ECUs that support this functionality

Command in client batch mode:

```
E-Sys.bat           -lockIPsec  -connection <config file>
```

Command in client-server batch mode:

```
E-Sys.bat  -server  -lockIPsec  -connection <config file>
```

Declaration:

`-lockIPsec`	command.
`-connection <config file>`	Absolute path to configuration file with the connection parameter (see (Parameter).

Example:

```
E-Sys.bat -lockIPsec -connection C:\conf\connection.properties
```

Information:

Connection Parameter

General:

In order to establish a connection, the following parameters can be defined in a batch configuration file:

```
PROJECT = <project name>
VEHICLEINFO = <vehicle info>

# bus|gateway_url|icom_d_can|icom_ethernet|vin
CONNECTION = <connection type>
```

The parameter `Connection` is optional. Without this parameter the information's are read from file E-Sys.properties. If the parameter is defined, the batch configuration file has to consist the following parameter depending on the connection type:

Connection type `bus`:

```
# z.B. BODY_CAN, A_CAN, D_CAN, ...
BUS_NAME = <bus name>

# VECTOR_DIRECT | VECTOR_DIRECT_FLEXRAY | PASS_THRU
INTERFACE = <interface>
```

Connection type `gateway_url, icom_d_can, icom_ethernet`:

```
# Format <protocoll>://<IP address>:<port>
URL = <URL matching the format above>
```

Connection type `vin`:

```
   # Format <VIN 17-digit>_DIAGADR<gateway diagnostic address
hex>|AUTODETECT
   GATEWAY_VIN = <VIN matching the format above>

   If AUTODETECT is used, E-Sys is creating a list with available
VINs with the following behaviour:
   List = 1 VIN:  this VIN is used for connection
   List = 0 VINs: "No available vehicles were found"
   List > 1VINs:  "More than one available vehicle found: VIN1,
VIN2, ..."
```

Optionally, the following vehicle-specific parameters could be defined.

```
# on|off
READ_VEHICLE_CONNECTION_PARAMETER

# used if READ_VEHICLE_CONNECTION_PARAMETER = off
VEHICLE_CONNECTION_PARAMETER_SERIES = <series>

# used if READ_VEHICLE_CONNECTION_PARAMETER = off
VEHICLE_CONNECTION_PARAMETER_ISTEP_SHIPMENT = <I-step>
```

For the value `on` of parameter `READ_VEHICLE_CONNECTION_PARAMETER`, the series and I-step will be read from VCM. Otherwise the parameters `VEHICLE_CONNECTION_PARAMETER_SERIES` and `VEHICLE_CONNECTION_PARAMETER_ISTEP_SHIPMENT` will be read.

Abbreviations

ASAM	Association for Standardization of Automation and Measuring Systems
AVAKON	Automatic Configuration of Variants
BAF	Boot Sector Exchange Format
BT	Description Table
CAF	Coding Application File
CBD	Coding Description File
COS	Central Operating Services
CVN	Calibration Verification Number
DAF	Data Exchange Format
ECCO-NF	Electric / Electronic Change- & Configuration-Management-Successor
E-Sys	New programming system for department E
FA	Vehicle Order
FFS	Flash File System
FP	Vehicle Programming
FP-Prozess	Base System Process of Vehicle Programming
FSC	Activation Code
FWL	Function value list
FZM	Vehicle State Management
GUI	Graphical User Interface
HO	Dealership Network
IEX	Integration Test Vehicle (new: EBG Development Component)
Intel-Hex	Data Format Intel-Hex
KIS	Compatibility and Information System
KMM	Configuration Management Module
LabCar	Labor Vehicle
LAN	Local Area Network

LBT	Logical Block Table
MCD	Measurement, Calibration and Diagnosis
MCD-2D (ODX)	An ASAM standard, that comprises the exchange format of ECU data and ECU descriptions in conjunction with diagnosis tasks.
MMI	Man Machine Interface
NCD	Netto coding values (Nettocodierdaten)
OBD	On-Board Diagnostics
OffBPS	Offboard Programming System
OnBPS	Onboard Programming System
PAF	Program Exchange Format
PDM	Product Data Management
PDX	Packed ODX
PE	Programming Unity
PEP	Product Manufacturing Process
PIA	Personalization, Individualization, Adoption
PPP	Product-Process-Prototype (new: BBG Actuation Component)
PSdZ	Programming System of the Future
PV	Patch Version
SA	Special Equipment
SALAPA	Special Equipment, Country Equipment, Package
SG	ECU (Electronic Control Unit)
SG-ID	ECU ID
SGBM	Control Unit Description Model
SLdZ	Software Logistic of the Future
SPA	Software Package Administration
SPS	Software Package Storage
SREC	Data Format Motorola SREC
SVK	System Installation Identifier
SVT (Ist)	Actual System Installation Table

SVT (Soll)	Reference System Installation Table
SW	Software (process class)
SWE	Software Unity
SWT	Sweeping Technologies
TAL	Transaction List
TI	Technical Integration
UV	Sub Version
VCM	Vehicle Configuration Management
VIN	Vehicle Identification Number
VS	Sale and Service
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language
XSD	XML Scheme Definition

Link to usefull documentation

Collection of links to more help documents in GIS:

1. User information on handling of inconsistent PDX containers in E-Sys.

2. Information on a programming problem via Ethernet with Windows 7 in Zeroconf network environments.

WAVE-11 - Speciality

To perform a positive flash/coding-action with the e/e-component WAVE-11 it is necessary to set a new parameter at the registry.

Keep in mind, that **local admin rights** are needed to make changes in the registry.

For the WAVE-11 it is necessary to ad the parameter TcpTimeWaitDelay with the value of 30.

The change can be executed automatically with the batch-file "registry-wave11.bat" located in the lib folder (E-Sys\lib\registry-wave11.bat)