E-Sys

Coding
and
Vehicle Configuration Management

User Manual

Version:	1.4
Valid as of E-Sys Version:	3.0.2
Date:	30.07.2007
Authors:	Markus Wolf, EG-75 (Coding)
	Dr. Christian Süß, EG-75 (VCM)
	Hendrik Spila, Bertrandt for EG-75 (Tutorial)
Number of pages	51

Contents

1 Overview

This Manual describes the procedures for handling all tool components of E-Sys required for coding control units and entire vehicle electrical systems. These tool components include:

- Expert Mode Coding for writing (coding), reading and modifying coding data of electronic control units (ECUs) or vehicles.
- Function Data List Editor (FDL Editor) for modifying coding data in control units
- CAF Viewer for viewing Coding Application Files
- Expert Mode Coding Verification for virtual verification (without control units) of Coding Application Files (CAFs) and for generating system variant tables from KIS
- Vehicle Order Editor for creating and editing vehicle orders
- Expert Mode Vehicle Configuration Management (VCM) for reading and writing vehicle configuration data.

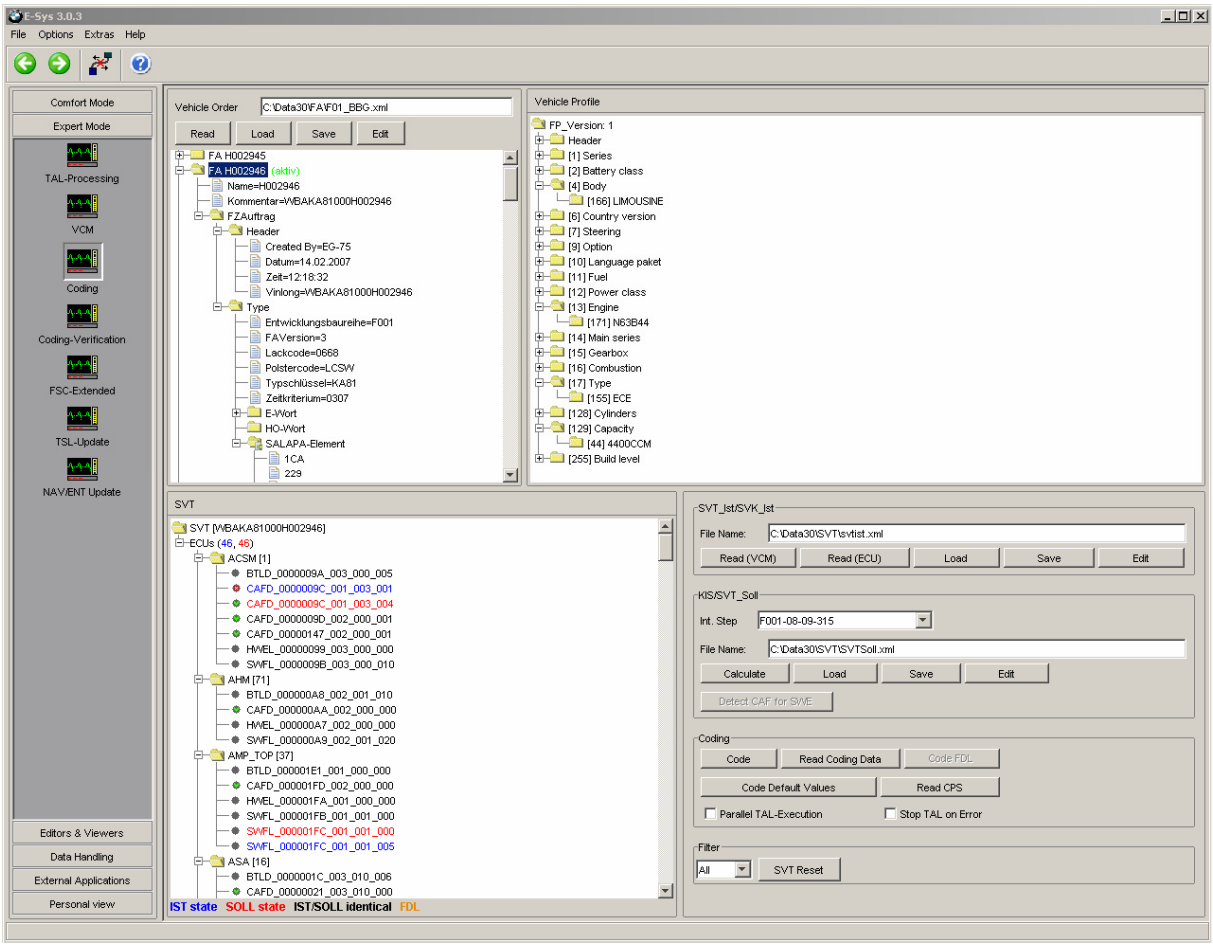
In addition, the Appendix contains a tutorial illustrating the individual steps required for coding an uncoded control unit without KIS and using the FDL functionality.
This document does not discuss the option of coding control units in pure TAL mode.
E-Sys is used both to code control units developed in accordance with the Coding Specifications SAP No. 10001491 or that use Standard Core 6.xx and to code carry over control units for the BN2000 environment in a BN2010 vehicle bus system.

Note: It is advisable to use a colour printer for printed copies of this User Manual since various E-Sys displays are colour displays and are unsuitable for monochrome printouts.

2 Expert Mode Coding

2.1 Graphical User Interface (GUI)

The expert GUI for coding is shown in the diagram below:



The display area uses frames to allow the size of the displays for FA, FP and SVT to be matched individually

Expert Mode Coding view contains the following components:

- Vehicle data
- SVT
- SVT/SVKactual
- KIS
- Coding
- Filter

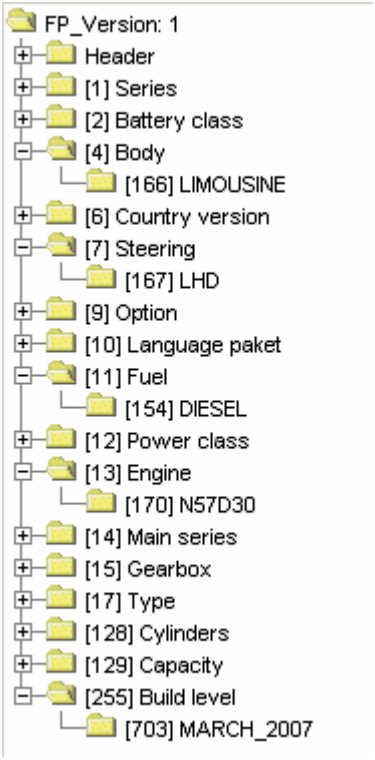
The individual components are explained in further detail in the sections that follow.

2.1.1 Vehicle data

The *Vehicle Data* view consists of the Vehicle Order (Fahrzeugauftrag FA) and Vehicle Profile (FP) displays. Buttons allow you to read (Button *read*) vehicle orders from vehicles (more precisely VCM, see Chapter *6 Expert Mode Vehicle Configuration Management*) or load (Button *load*) them from a file. E-Sys basically operates with FA lists that may contain several FAs. If an FA is read from the vehicle, a list is created containing only this FA. The FA is then displayed as a tree structure. The long VIN number (Vinlong) is always read from the VIN master (e.g. CAS/ZGW) or loaded from a file together with the FA. The illustration below shows the FA tree:

If an FA list has been read or loaded, the user must initially select an FA and activate it for further actions. This is done using the context menu (right mouse button) or by double-clicking (FA entry is marked active). The vehicle profile is then determined for the activated FA and it is updated in the tree at the right. At the same time, the FA and the VIN are checked for validity. If the checksum of the VIN at the 9th position is incorrect, work on the selection cannot continue.
Only one FA may be active at any one time. If an FA is activated and another FA entry from the list is activated, the currently activated FA is deactivated and the other FA is activated.
The vehicle profile is calculated with the aid of an assignment table (FA2FP) that is a part of the PDX package. This also includes a check of the activated FA with respect to the FA2FP. If the FA contains elements (SAs or model codes etc.) that are not contained in the FA2FP, the selected FA is invalid and work cannot continue. In this case, the errored elements must be corrected with the FA Editor (see Chapter *5 FA Editor*).

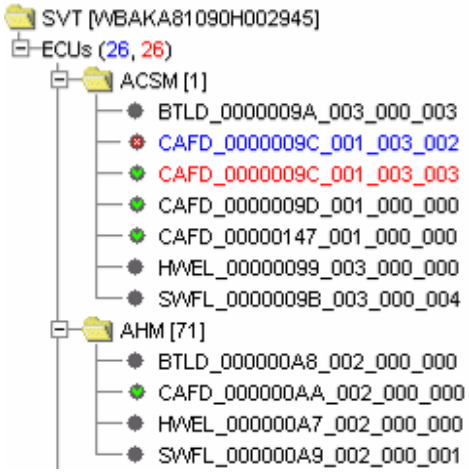
The FP tree is shown below:



The vehicle profile is extracted from the FA. It contains all optional extras and country spec packages (SALAs) and other information derived from model code, build level, supplementary and trade organisation words (E words and HO words) and their readable meaning. This information is broken down within permanently predefined categories as criteria [see also Performance Specification VCM SAP No.10000908]. For example, this includes the following in the vehicle profile shown:
Drive-side category with criterion LL for left-hand drive or
Engine category with criterion N57D30 or
Fuel category with criterion DIESEL etc.
The numerical values in the square brackets represent the machine-readable values of the categories and criteria.
The vehicle profile is used when coding for evaluation of the Coding Application Files (CAFs). The function conditions in CAF files are based on Boolean logic operations of vehicle profile elements and, consequently, only indirectly on vehicle order elements.
The *save* button can be used to save the displayed FA list as an XML file. The *edit* button opens the displayed FA list automatically in the FA Editor (see Chapter *5 FA Editor*) for editing. The vehicle profile can not be saved in this view (see Chapter *6 Expert Mode Vehicle Configuration Management (VCM)* for saving).

2.1.2 Software variant table tree

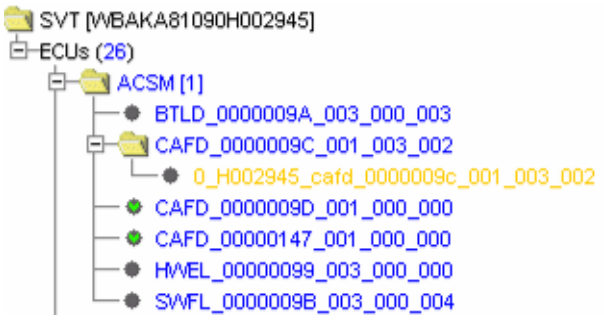
The software variant table tree (SVT tree) in this view does not contain all elements of an SVT but only the number logistics of vehicle electrical system users (SGBM IDs, SGBM = control unit description model), grouped on the basis of basic variant designations of the control units. The diagnostic addresses can be found in hexadecimal notation after the basic variant designators in square brackets. In addition, the root element of the tree shows the long VIN number of the activated vehicle order. The complete SVT can be displayed using the SVT Editor of E-Sys (not a part of this document). Both actual statuses of SGBM IDs of control units and their target statuses can be shown merged in the tree. The differences are distinguished by means of colour. Actual statuses are shown in font colour blue and target statuses are shown in font colour red. Font colour black identifies correspondence between target and actual configurations. The number of the basic variants in the relevant SVT is shown in brackets after the element *ECUs* (blue font colour SVTactual, red font colour SVTtarget).



CAF identifiers (process class CAFD) are also marked with a symbol providing information on codability of a control unit in relation to the current configuration. CAFD entries with a green check mark can be used to code a control unit in the current configuration. Entries with a red cross cannot. The buttons in the Coding frame can also be used accordingly. Determining this codability status for coding or read-out of a control unit is based on the following rules:

1. A control unit can always be coded/read out if CAFD entries are present in the actual configuration (SVTactual), unless all non-coding process classes, i.e. all other classes apart from CAFD, are identical and the target configuration (SVTtarget) prescribes a different CAFD, e.g. with a higher version.
2. A control unit can always be coded/read out if all non-coding process classes in target and actual configuration are identical.
3. A control unit can never be coded if a basic variant occurs only in the target configuration (control unit not fitted).
4. A control unit cannot be coded if at least one non-coding process class has differences in target and actual configuration and if there is no CAFD entry in the actual configuration. In this case, the control unit must first be set to the required software build by flash-programming.

One other special feature of the SVT view in the Coding Expert GUI is fitting function data lists (FDLs) beneath CAFD elements. FDLs contain the interpreted, read-out coding data of a control unit in machine-readable form (see also Chapter *2.2.3 Coding control unit with FDL*).



FDLs can be fitted in the SVT tree in two ways:

- Read-out of a control unit → FDL is fitted automatically by E-Sys (see also Chapter *2.2.2 Reading out control units*)
- Using the context menu *New→FDL* on a CAFD entry and selecting an existing FDL using the File dialog box

2.1.3 SVT/SVKactual

Actual configurations can be read from control units or loaded from a file in section *SVT/SVKist* (*SVT/SVKactual*). If a Vehicle Configuration Management (VCM) is available, the *read (VCM)* button can be used to read the entire SVT of a vehicle from its VCM. Identification of individual control units is performed with *read (ECU)*. In this case, E-Sys sends functional request *read_SVK*. The button *load* can be used to load an SVTactual from a file. The result is displayed in the SVT tree in all cases (in blue font colour unless an SVTtarget was already present). The *edit* button can be used to edit the SVTactual with the SVT Editor, and *save* can be used to save the SVTactual to a file.

2.1.4 KIS

Target configurations for control units or control unit networks (vehicle) can be calculated on the basis of the vehicle order (FA) and a target integration step or can be loaded from a file in section *KIS*. The result is shown in the SVT tree in both cases (in font colour red unless an SVTactual was already present). The *edit* button can be used to edit the SVTtarget with the SVT Editor, and *save* can be used to save the SVTtarget to a file.
Alternatively, there is the option of determining the CAF for individual software units of a basic variant without specifying an integration step directly using the *detect CAF for SWE* function. This function can be applied to SWFL entries or basic variants in the SVT tree.

Note: All KIS functions can be used only if a KIS knowledge base is available. The KIS knowledge base is installed only on PDX import of an integration step container.

2.1.5 Coding

The buttons in the Coding frame are described in *2.2 Operation*.

2.1.6 Filter

The filter can be used to filter the SVT tree by process classes of the SGBM. For example, only entries with coding data process class CAFD can be displayed. Button *SVT Reset* deletes all contents of the SVT tree.

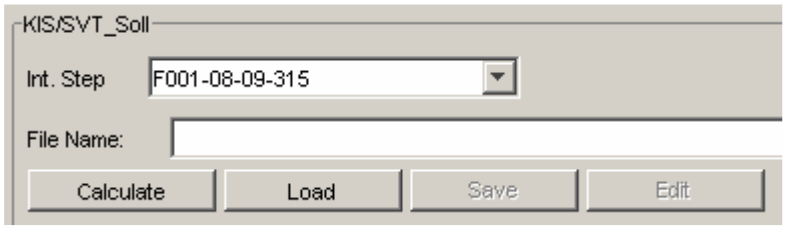
2.2 Operation

Expert Mode Coding basically offers many options for coding control units or entire vehicle electrical systems. The GUI offers major flexibility depending on application (control unit development or vehicle commissioning etc.) but also poses risks if operated incorrectly.

2.2.1 Coding control unit/vehicle with vehicle order

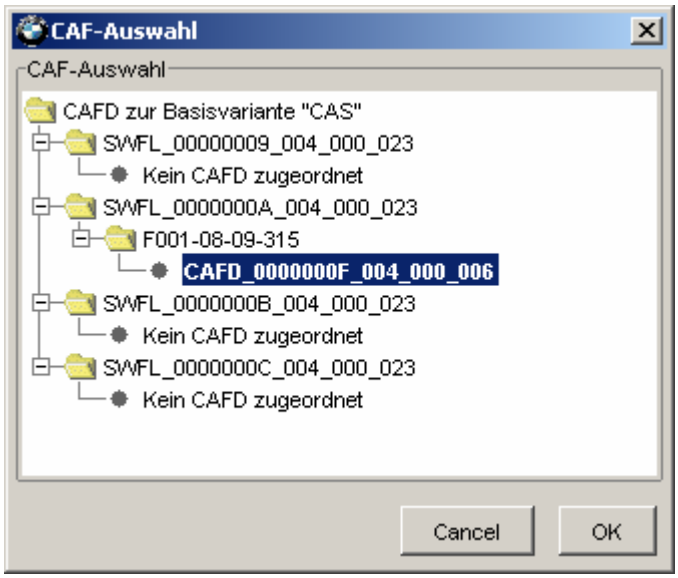
In order to code a control unit by vehicle order, it is initially necessary to load a valid vehicle order, activate it (see *Chapter 2.1.1 Vehicle data*) and establish a link to the central gateway of a vehicle (or test bench) or to a control unit (e.g. via CAN). Since the buttons for coding can be applied only to contents in the SVT tree, the SVT tree must be populated in the next step. There are several options for populating the SVT tree:

1. Read SVTactual from connected control units functionally (button *read (ECU)*) or VCM (button *read (VCM)*)
   or
   load SVTactual from file: This method is suitable only for test purposes and should normally not be applied.
   If the connected control unit is already coded (CAFD entry present in SVTactual), it can be coded again with the CAF specified in the SVTactual (but, for instance, with a different vehicle order).
2. Determine SVTtarget from KIS with button *Calculate* specifying a target integration Step (possible only if an integration step container with a KIS knowledge base has been imported)



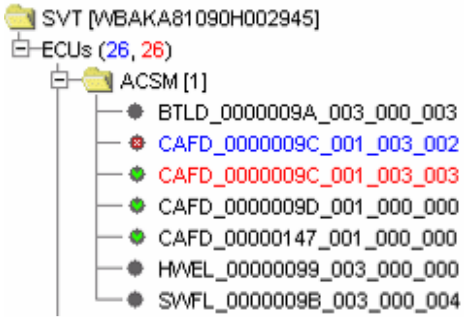
   or
   load SVTtarget from file (button *load*). This is possible only if the required SVTtarget has been created beforehand with the SVT Editor.
   or
   detect CAF for SWE via the KIS function. For this purpose, an SVTactual must be read or loaded. The assigned CAFs can be determined by marking the basic variant or individual SVT tree contents of process class SWFL by pressing button *detect CAF for SWE*. The in-

formation is displayed as a selection tree in a separate window, ordered on the basis of software units SWFL and integration Steps:

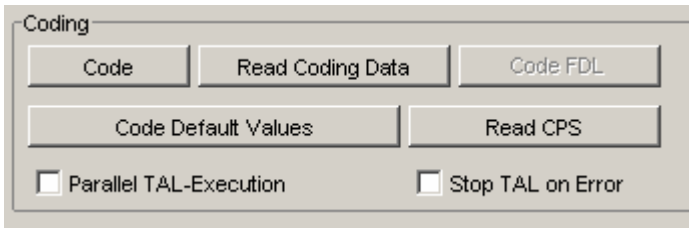


In the example, SWFL 0000000A_004_000_023 is assigned CAFD 0000000F_004_000_006 at integration Step F001_08_09_315. No other SWFLs are assigned a CAF.
The selection is imported to the SVT tree as SVTtarget entry by marking the required CAFD entry and confirming with *OK*.

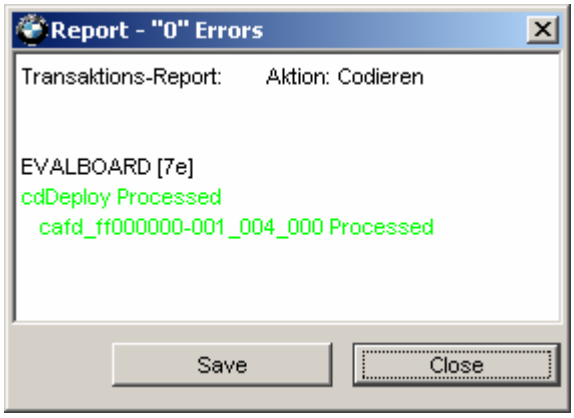
If both SVTactual and SVTtarget have been determined, E-Sys displays both SVTs in the SVT view merged:



In the case shown, only the CAF from the SVTtarget (red entry) can be used to code the control unit (green check mark). The control unit is now coded by marking the CAFD entry and then clicking on button *Code* (or with *Code* in the context menu). E-Sys then automatically generates the required transaction list (TAL) and initiates processing of this list.

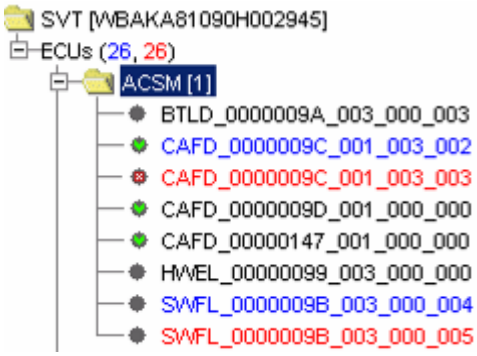


A status report on success of the action is displayed at the end of the coding operation:



The report is generated separately for each basic variant. Green font colour shows that the coding operation was successful and red font colour shows that the coding operation was errored. In the example, basic variant EVALBOARD at address 7E was coded successfully with cafd-0000006e-001-004-000.

Error:
The illustration below shows that the control unit can be coded only with the actual version of the CAF in accordance with the rules from Chapter *2.1.2 Software variant table tree*. If the target version is to be coded, the control unit must first be programmed with the target software build (SWFL_0000009B_003_000_005).



Coding of individual control units, control unit networks or in the total vehicle is controlled via marking in the SVT view. The actions triggered with button *Code* relate to

- a marked CAFD entry
- a single control unit (i.e. also all CAFs for this control unit) if a basic variant is marked
- all marked basic variants (and automatically their assigned CAFs) in the case of multi-selection
- the total combination shown in the SVT if level *ECUs* or *SVT* is marked

Parallel TAL execution and stopping of TAL execution in the event of an error:
If switch *Parallel TAL-execution* has been set when coding at level ECUs (total coding-related content of the SVT tree) or in the case of multi-selected basic variants, coding of all selected control units is performed in parallel.

Switch *Stop TAL on Error* allows you to force an abort of TAL processing after the first error. Otherwise, errors that occur during the sequence are not displayed in the report until the end of TAL processing.

2.2.2 Reading out control units

Button *Read coding data* can be used to read out coding data from control units. As with the *Code* functionality, multiple selection of control units to be read out is possible here as well.

- Selection of individual CAFD entries
  → Control unit is read out with the selected CAF
- Selections at Basic Variants level
  → Control units are read out with all CADs listed below the selected basic variants
- Selection at ECU or SVT level
  → All control units with valid CAFD entries with green check mark in the SVT view are read out.

The precondition for read-out of control units, as is also the case with coding, is that the CAFs required in accordance with the SVT view be present in the file system.
The result of read-out operations is a net-coded data file (NCD) and a function data list file (FDL) per CAF. After read-out, both files are saved automatically to the file system. The FDLs are also appended in the SVT view as sub-elements of the related CAFD entries in font colour orange.
The FDL generated can then be opened in the FDL Editor and edited using the context menu *Edit FDL* (see Chapter *3 FDL Editor*)

2.2.2.1 Net coding data

The NCD file with the net coding data read out is saved to E-Sys data directory ..\CAF. The file name of an NCD file corresponds to the name of the CAF used for read-out of a control unit, but with file extension *ncd* (e.g. CAFD_0005638F_003_025.001.ncd). The net coding data is written to the file in Motorola S-Record format, e.g.:

S00A0000FF000000010401F0
S206300000663E25
S208300100ACE80F0F14
S20C30210005FF00000001040198
S2243022000000020D62448E9626B1CD69C941DF7470ED8A011063F88DF0ACCAD6933FFA5EE
S224302220C62ADCC6C7AB18801BC853AABD3D9B94152DEF5495CA0A76C0C9A5C08D2601AC12
S2243022403F5B71E2B3356EE9BFA5D77C7F4CAA93E04A94C379200D60DF59DC736A76C6E9C1

S224302260D037FD2F4A0699F61530A6EEEF59C297EDD0E6347F4DDF82B36568C16928CE04F5
S2083022802BB83841C9
S503000EEE

Note: A detailed description of the S-Record can be found, for instance, at
http://www.amelek.gda.pl/avr/uisp/srecord.htm

2.2.2.2 Function data list

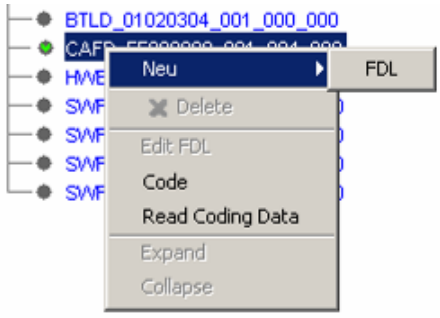
An FDL's structure and content correspond to those of a CAF, but extended with the net coding data read out from the control unit and interpretation thereof in function-value and coding-value pairs. The FDL is saved to E-Sys data directory ..\FDL. The file name of an FDL file corresponds to the name of the CAF used for read-out of a control unit, but with file extension *fdl* (e.g. CAFD_0005638F_003_025.001.fdl).

2.2.3 Coding control unit with FDL

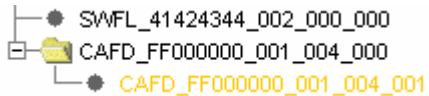
In Expert Mode Coding, E-Sys offers the option of changing and coding individual control unit functions in targeted manner. This is done with the aid of an FDL. A personal Developer Soft Token (EST) and the related PIN that you can apply for from the Automotive Security Backup Center (ASBC) (see also Tutorial) are required for running this function. The EST file must be specified under Options → Settings → EST.
EST is available for the following roles:

Role	Description
BASIC	Is required in order to open CAFs with read access with the Caf Editor
ADVANCED	Is required in order to allow coding with FDLs in E-Sys
EXPERT	Is required in order to edit CAFs with the Caf Editor

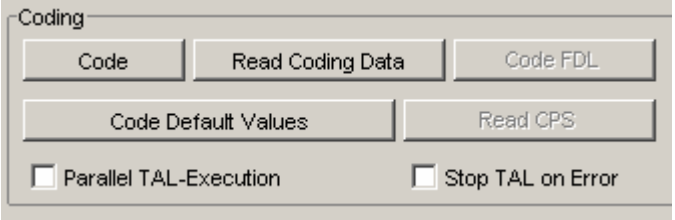
In order to be able to code a control unit with an FDL, it must initially be created. This is done either by read-out of the control unit (see Chapter *2.2.2 Reading out control units*) or using the context menu in the SVT view with the *New→FDL* function with an existing FDL file. Both procedures can be applied only to individual CAFD entries in the SVT view.



The FDL is displayed as a sub-element of the relevant CAFD entry in font colour orange after the action has been executed.



The FDL can be opened and edited in the FDL Editor using the context menu *edit FDL* (see Chapter *3 FDL Editor and CAF Viewer*).
In order to code a control unit with FDL, mark the FDL entry beneath the CAFD entry in the SVT view and use the *code FDL* button to trigger the action. If this function has been called for the first time after E-Sys is launched, you are prompted to enter your personal PIN.

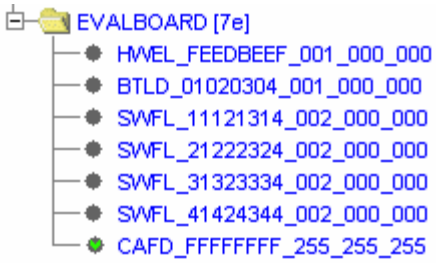


When coding a control unit with FDL, only coding data described in the FDL under the elements *Ausgelesen/Werte* (*Read-out/Values*) are written (see Chapter *3 FDL Editor and CAF Viewer*).

Notes: A vehicle order with valid VIN number must be loaded and activated in order to allow you to write a valid coding check stamp for the *code FDL* function. The vehicle order itself is not used for coding in this case.
Coding of a control unit with FDL works only if the valid CAF on which the FDL is based is found in the E-Sys data path (identical SGBM ID and versions).

2.2.4 Coding control unit to as-delivered condition

The function for coding a control unit to as-delivered condition is primarily used in verification for control unit coding. A control unit is coded with this function independently of the vehicle order to the as-delivered values specified in the CAF. Since the SGBM ID of the CAF used is also set to the as-delivered condition (CAFD_FF…), read-out of the SVK supplies process classes CAFD with as-delivered values (see illustration) after successfully running this function.



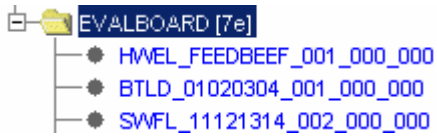
Function *Code Default Values* can be applied only to individual CAFD entries in the SVT tree.

Note: The CPS is not set to as-delivered condition (0xFF) when this function is run.

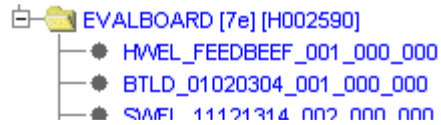
2.2.5 Read CPS

The coding proof stamp (CPS) serves the purpose of documentation of coding operations on codable control units. The CPS value corresponds to the last seven digits of the VIN of the vehicle order used to code a control unit and is written automatically to a control unit at the end of each coding operation.

Function *read CPS* reads the CPS from control units. The SVT tree must be populated and at least one control unit in it must be marked at Basic Variants level for this:



The result of the read-out operation is displayed as ASCII string (e.g. [RH09567]) behind the diagnostic address of the basic variant in square brackets.



Alternatively, this function can also be applied to a total vehicle electrical system. For this purpose, mark level *ECU* in the SVT tree and then trigger the CPS determination process with button *read CPS*. Only control units having the SGBM process class CAFD for coding in the SVT tree are read out.

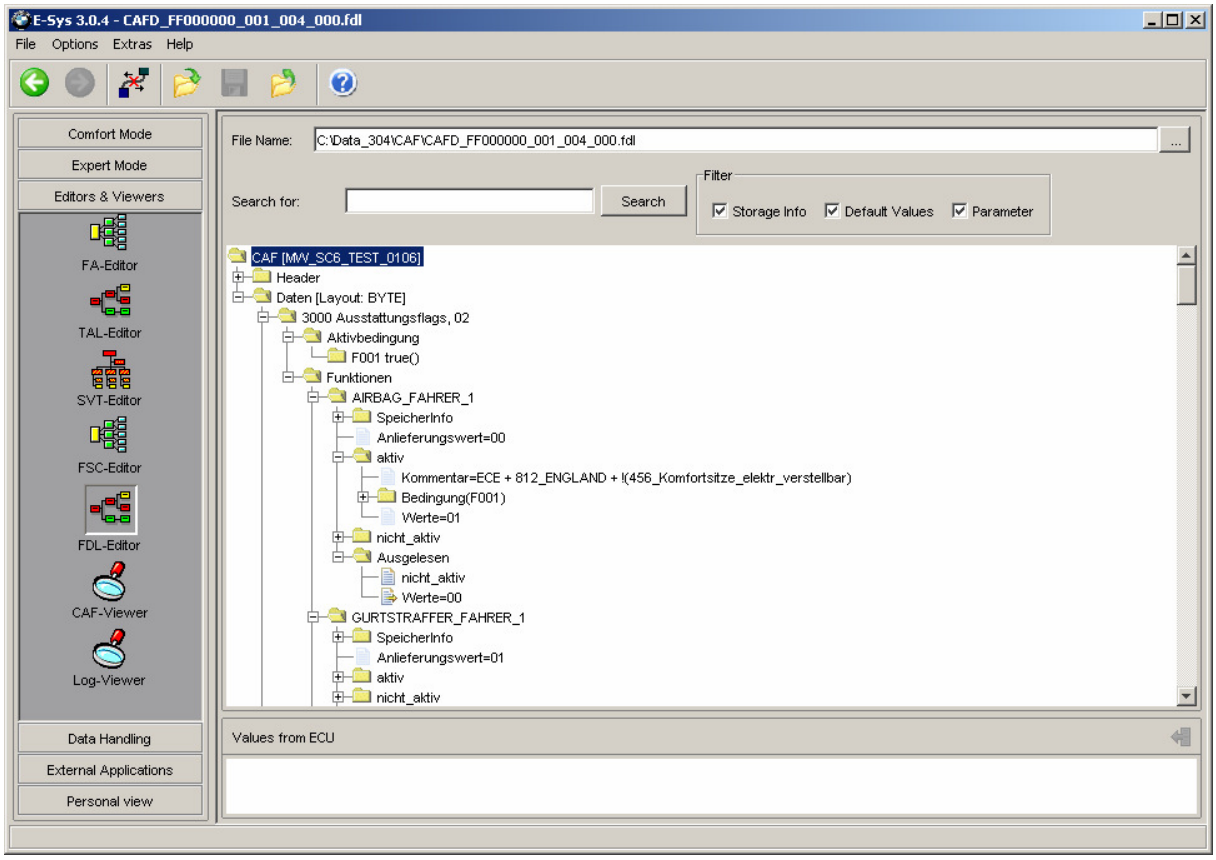
If all CPSs have been read out, the user can check whether all codable control units have been coded to match the vehicle in which they are fitted as well. An automatic check for identical CPSs is not supported by E-Sys.

3 FDL Editor and CAF Viewer

Both Coding Application Files and function data lists are XML files and are based on the same XML schema. An FDL is the result of a read-out operation of coding data of a control unit and always corresponds to the CAF that was used to read out the coding data but extended with the values read out and their interpretation in the logical coding data layout.
FDL Editor and CAF Viewer differ only in respect of the content of the tree view (FDL has additional sub-branches with coded values read out) and are described together under a general heading at this point.

3.1 GUI

The FDL Editor is used to edit coding data of control units without reference to a vehicle order. The CAF Viewer is used to view CAF files. Both modules can be found in the module bar at the left beneath *Editors and viewers*. The view of the FDL Editor or CAF Viewer is shown in the illustration below.

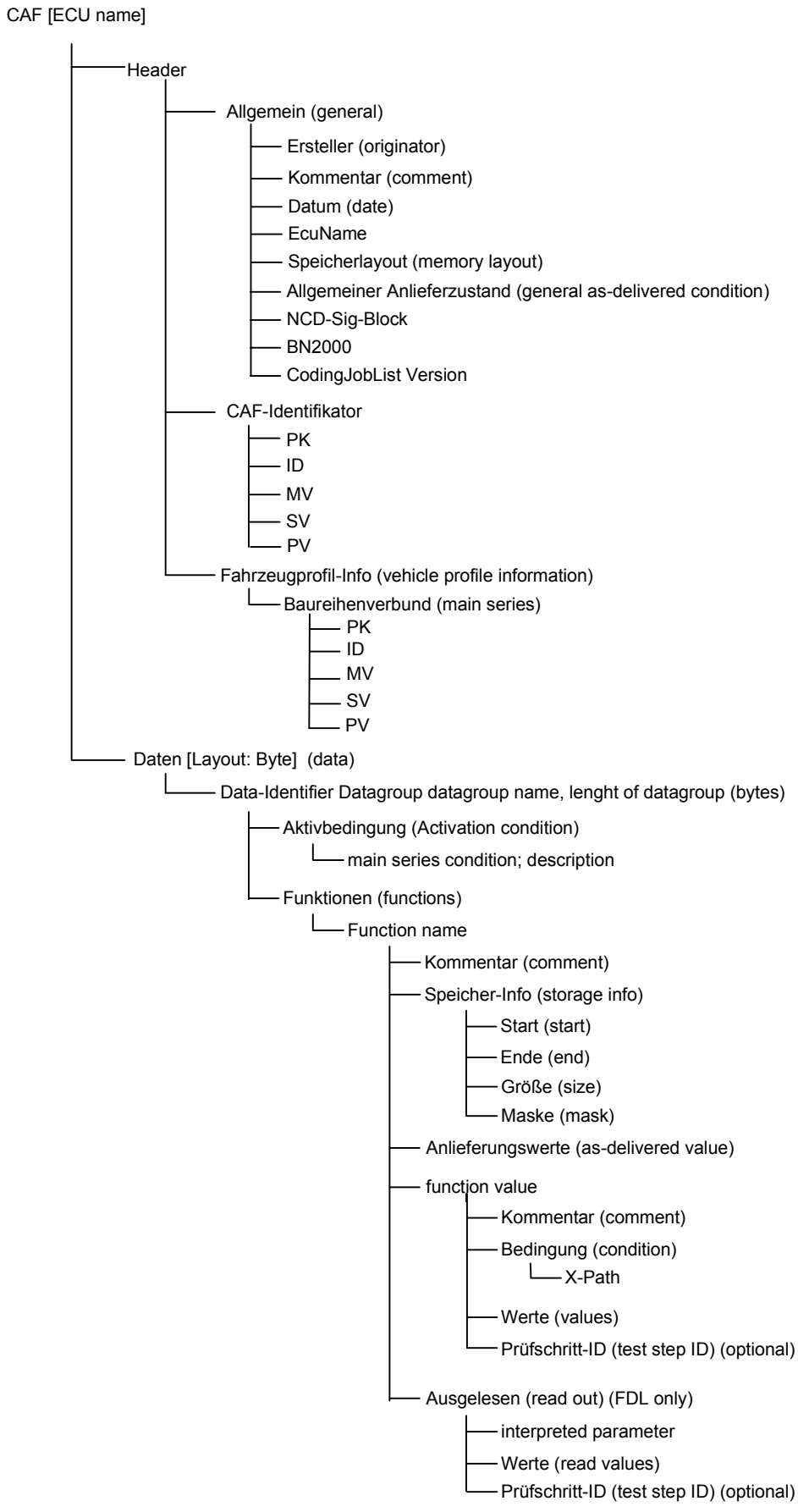


FDL Editor and CAF Viewer have the following controls and display elements:

- File Name text field and button for opening an FDL or a CAF from the file system with the File dialog box
- Text field and button for searching for codable function names in an FDL or CAF
- Filter unit for controlling display of FDL/CAF contents
- Tree view for editing or displaying the contents of FDLs or CAFs

3.2 Tree structure and tree contents

The contents of CAFs or FDLs are shown in a tree structure as follows:



The following definitions apply to the structure of CAFs and FDLs:

- Codable function: the term *Codable Function* means a control unit function that, if changed by coding, influences the behaviour of the control unit.

- Function value: a codable function has one or more *Function Values* precisely one of which is assigned to it by coding. This assignment can be made by selecting a *Function Value* from a set of possible *Function Values* in accordance with a coding condition or by calculating the *Function Value* with a calculation job (e.g. active, not_active etc.).

- Coding value: each function value is represented in the control unit by a *Coding Value*. This may be an individual numeric expression or a set of numeric expressions that are actually written to a control unit (e.g. 0xFF, 0x10 etc.)

Description of the tree elements:

- Header/general
  - Originator: name of the originator of the CAF
  - Comment: free comment on the CAF
  - Date: creation date of the CAF
  - EcuName: name of the control unit for which this CAF is destined.
  - Memory layout: memory organisation of the coding description (currently only BYTE)
  - General as-delivered condition: general coding value of codable functions in uncoded state (as-delivered condition). Is used by E-Sys for application “Code as-delivered condition” if no specific as-delivered state value is specified for the corresponding codable functions
  - NCD Sig. Block: data identifier with which the signature of the net coding data is transferred to control unit
  - BN2000: attribute specifying whether the CAF applies to a BN2000 control unit (value = true) or to a BN2010 control unit (value = false)
  - CodingJobList Version: version for creation of the CodingJobList for calculated coding values of a codable function, taken as a basis for creation of the CAF

- Header/CAF identifier
  - PK: process class for coding data (CAFD)
  - ID: SGBM ID of the CAF
  - MV: main version of the CAF
  - SV: secondary version of the CAF
  - PV: patch version of the CAF

- Header/vehicle profile information
  - Main series of the FA2FP mapping table
    - PK: process class of the FA2FP mapping table (FAFP)
    - ID: SGBM ID of the FA2FP used to create the CAF
    - MV: main version of the FA2FP
    - SV: secondary version of the FA2FP
    - PV: patch version of the FA2FP

- Data
  - Data identifier of the data group with which the coding data of this data group is transferred to the control unit
  - Active condition, main series: contains a Boolean expression indicating whether the data group for a main series is to be transferred to the control unit or not. The expression may be vehicle profile-dependent or job-dependent
  - Functions: bracket element around all codable functions of a data group
  - Function: designation of the codable function
  - Comment: comment on the codable function
  - Storage information: information on the position of the codable function in the memory layout
    - Start: byte number of the start of the codable function in a data group
    - End: byte number of the end of the codable function in a data group
    - Size: describes the ratio of memory mapping of a codable function to a memory unit (1 byte). This may either be less than, greater than the memory unit or precisely one memory unit (*less than unit, greater than unit, equal to unit*)
    - Mask: describes position and size of the codable function in a memory unit
  - As-delivered values: this describes the specific coding values of a codable function used by E-Sys for functionality “Code as-delivered condition”.
  - Function value: designation of a function value of a codable function
    - Comment: free comment on a function value
    - Condition: description of the dependences of the function value on the vehicle profile
      - X-Path: Boolean expression from vehicle profile elements in X-Path notation
    - Coding values: summary of the numerical (hexadecimal) values that a codable function for the function value may assume.
    - Test step ID: reference ID for the function checklist, electrical system.
  - Read out: element for accommodating the coding values of the codable function read out (only in the case of FDL).
    - Interpreted function value: name of the function value determined on the basis of the coding value read out for a codable function value (assumes value UNKNOWN if no correspondence is found between coding values in CAF and coding values read out from the control unit).
    - Read coding values: the numeric coding values actually read out from the control unit
    - Test step ID: reference ID for the function checklist, electrical system (as above)

Example of an FDL in the tree view of the FDL Editor:

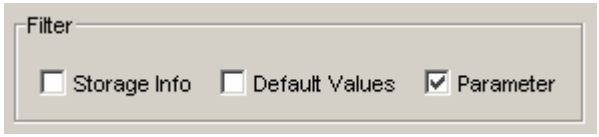
3.3 Operation

 or  Open existing CAF/FDL via File dialog box

 Save changed FDL (CAF files cannot be saved!)

 Close open CAF/FDL

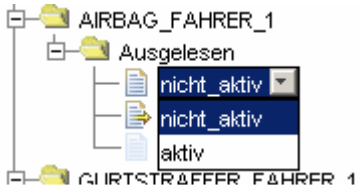
The tree view is populated with the CAF or FDL data after a CAD/FDL is opened using the File dialog box or from the context menu in Expert Mode Coding (*edit FDL*).
By default, when an FDL is opened, only the values read out from the control unit are shown beneath element “Read out”. If necessary, the filter can be used to show the remaining information from the underlying CAF.

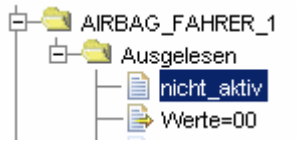


You can search by function names in the FDL using the search mask (option for sub-text search).

3.3.1 Editing function values

All function values present in accordance with CAF can be displayed and edited per codable function in order to edit coding data beneath element “Read out”. If you click on *Edit* in the context menu on the required function value, a drop-down list is displayed showing existing function values. After you select a function value, the corresponding coding value for the selected function value (in accordance with underlying CAF) is automatically entered in the FDL beneath element *Werte* (*Values*).



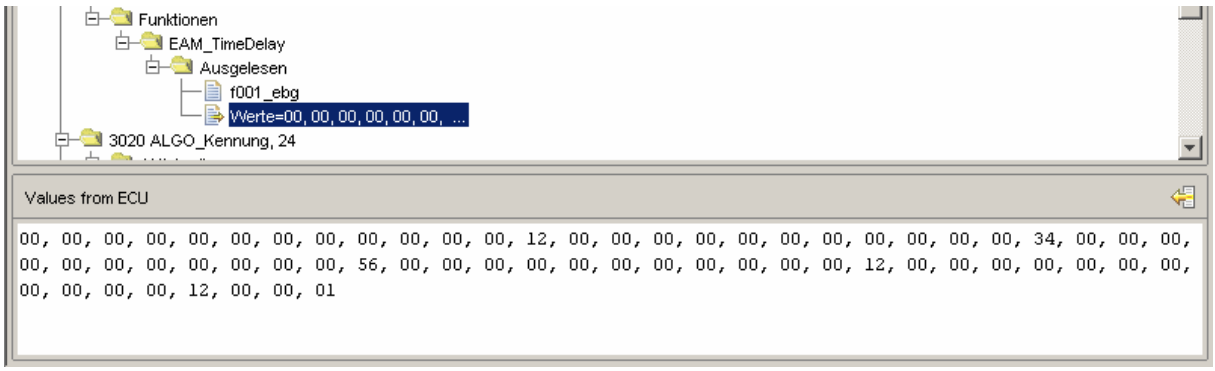


One exception relates to so-called *calculated functions* whose coding values are only determined by the system during runtime (e.g. coding date and paint code etc.) and cannot be changed.

Only function value *calculated value* is displayed in the drop-down list of these functions.

3.3.2 Editing coding values

The coding values of the codable function can also be edited as an alternative to editing coding data by selecting function values. Mark branch *Werte* (*Values*) beneath the required function for this. The existing coding values are then displayed in the text field in the bottom part of the window and the coding values contained can be edited. When doing this, note that several coding values must be separated by semicolons, followed by a space. Errored or incomplete entries are indicated by a red border around the text field.



This procedure can also be used to edit coding values of *calculated functions*.

Note: Editing coding values should be left to experts with adequate knowledge of a control unit since this could otherwise allow invalid coding values to be written to a control unit and, under certain circumstances, this could lead to damage the control unit software. Also, when entering coding values, ensure that the number or quantity of coding values match the function mask. E-Sys does not conduct a consistency check in respect of the function mask. This is done only during the coding operation by the programming system and leads to coding abort.

The changes can be accepted in the tree view with icon  in the bottom right-hand section of the window.

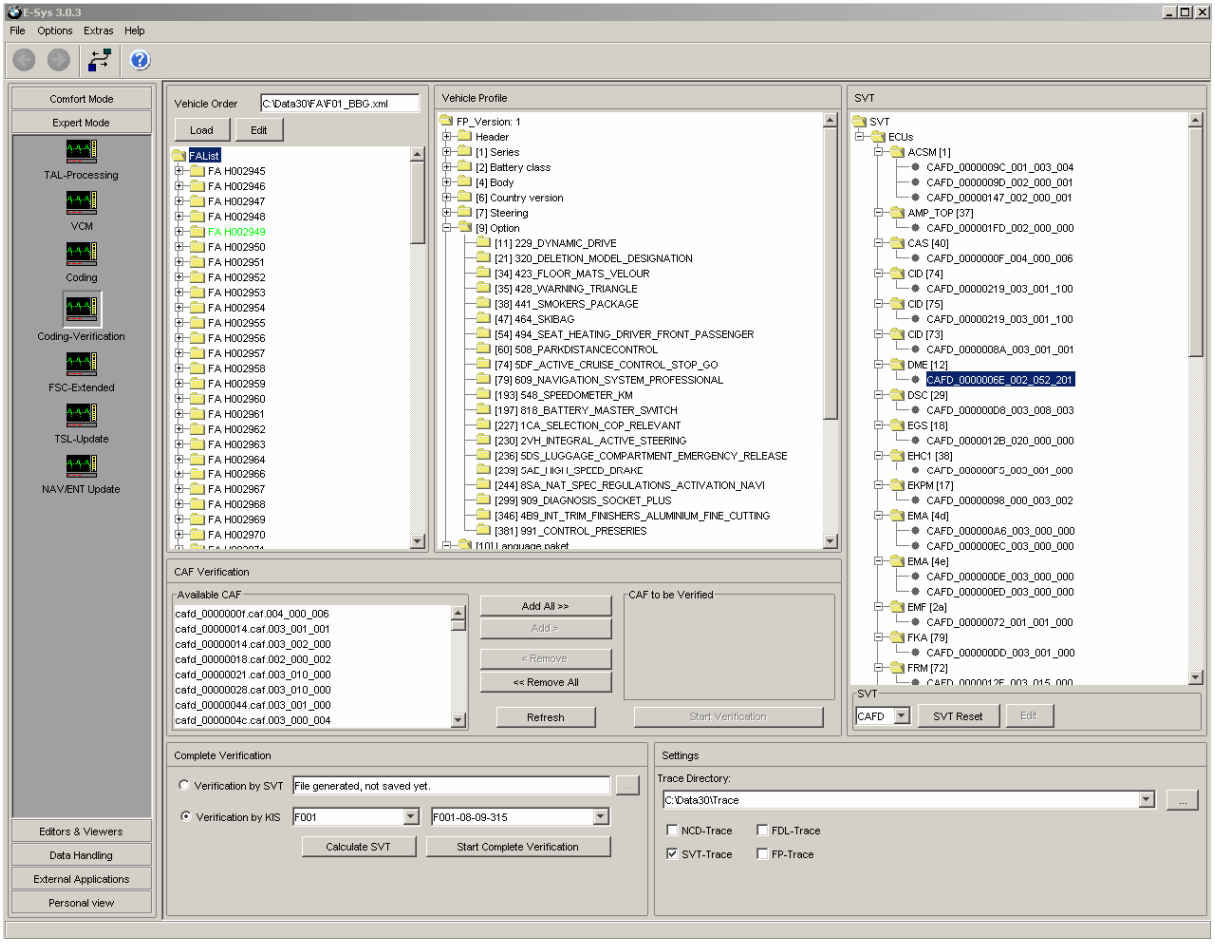
The FDL must be saved (do not save it under a different name!) after the required changes are complete. You can then switch back to *Expert Mode Coding* view and send the changes to the control unit with the *code FDL* function (see Chapter *2.2.3 Coding control unit with FDL*).

4 Expert Mode Coding Verification

Expert Mode *Coding Verification* is primarily used to verify CAF files without access to control units (“Simulation mode”). Target users of these functionalities are primarily specialist departments performing a corroboration function and control unit developers.
In Verification mode net coding data, function data lists and vehicle profiles for various vehicle orders can be generated automatically and saved in Trace files for further use. In addition, it is also possible to generate and save system variant tables for various vehicle orders from KIS.

4.1 GUI

The verification module for coding can be found in the toolbar at the left under *Expert Mode*. The GUI for coding verification is shown below:



The verification module contains the following controls and display elements:

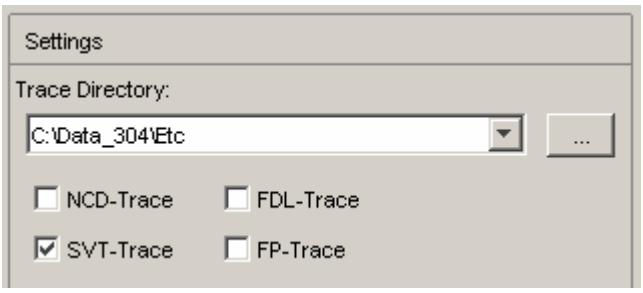
- Vehicle Data view for opening, viewing and editing vehicle orders
- SVT tree for viewing logistic data in system variant tables with the option of filtering for process classes and editing the SVT in the SVT Editor
- CAF Verification view for verifying individual CAFs with vehicle orders
- Total Verification view for verifying total I-Step content in relation to coding (net coding data, SWE-CAF assignments in system variant tables)
- Settings view for configuring the Trace options

4.2 Verification of individual Coding Application Files

In order to verify individual CAFs, they must be present in the data directory of E-Sys (..\psdzdata\swe\cafd). CAF files, like all other programming files, are also saved to the data directory by PDX import. The currently available CAF files are listed in List view *Available CAF*. Button *Refresh* is used to generate the contents of the CAF list anew after a PDX import for instance. Buttons *Add/Remove* can also be used to make a selection of the CAF files to be verified, and these files are then shown in list *CAF to be verified*.

The next step is to load an FA list with at least one vehicle order in view *Vehicle Data* (button *load FA*). *Edit FA* can be used to open the loaded vehicle order list in the Vehicle Order Editor if necessary for revision. The vehicle profile can be populated and displayed for individual vehicle orders via the context menu (click with the right mouse button on FA entry → calculate FP) (the related vehicle order entry is marked in green).
Verification of CAFs refers to marked vehicle orders in the FA tree. Multi-selection (mouse click + Ctrl/Shift key) allows any vehicle orders to be marked. If all vehicle orders of the loaded list are to be used for verification, the root element in the FA tree (*FAList*) must be selected.

If all boundary conditions are met (CAF files to be verified have been selected and vehicle order entries have been marked), verification can be started with button *Start Verification*. All specified CAFs with all marked vehicle orders are verified.
The scope of checks conducted when verifying CAFs is dependent on the Trace settings.

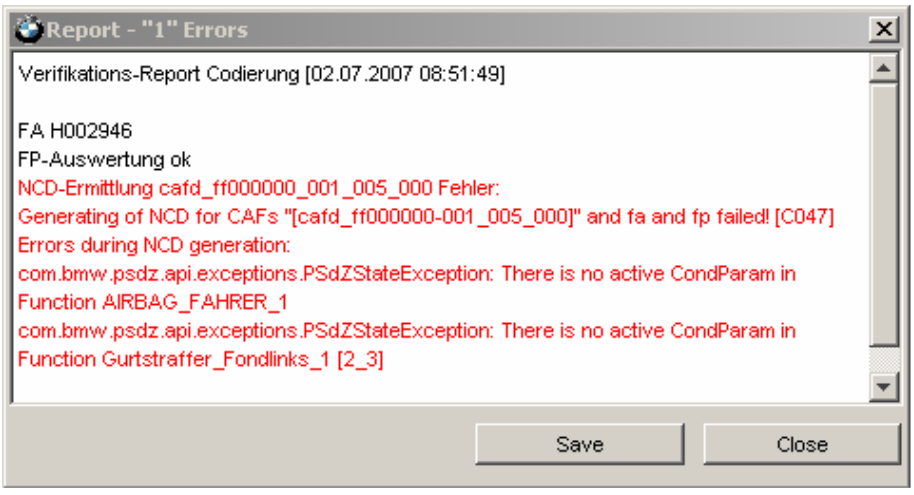


- No Trace recoding activated (no checkbox selected): the vehicle profile and the net coding data without recording are calculated (suitable for short checks).

- NCD Trace activated: the generated net coding data is saved as a file in S-Record format in the path (as specified under *Trace Director*).
  Naming convention:
  [Absolute position of the FA in the list with which the CAF was evaluated]_[FAName]_[SGBMID].ncd
  Example: 13_Testfaxyz_cafd_01020304_001_000_002.ncd
  FDL Trace activated: the generated net coding data is back-interpreted to an FDL and saved as a file in the path (as specified under *Trace Directory*).
  Naming convention:
  [Absolute position of the FA in the list with which the CAF was evaluated]_[FAName]_[SGBMID].fdl
  Example: 13_Testfaxyz_cafd_01020304_001_000_002.fdl
- FP Trace activated: the vehicle profile taken as a basis for generation of the net coding data is saved as a file in the path (as specified under *Trace Directory*).
  Naming convention:
  [Absolute position of the FA in the list with which the FP was generated]_[FAName].fpl
  Example: 1_Testfaxyz.fpl
- Setting SVT Trace does not influence verification of individual CAF files (see Chapter *4.3 Verification by SVT/KIS*).

Note: If at least one Trace recording is activated, a Trace directory must be specified to allow verification to be started.

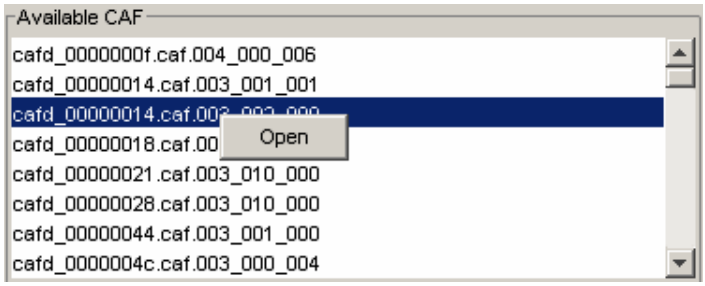
The result of a verification run is displayed in the form of a report at the end. If at least one Trace recording is activated, the report is also saved as a text file in the Trace directory (VReport_Date_Time.txt)
The illustration below shows a verification report:



Determining net coding data in the dry run serves firstly to reveal zero hits and double hits in the function conditions of a CAF and secondly to check correctness of the conditions used. A CAF can be used successfully to code a control unit only if precisely one function value and, consequently, one coding value exists for each function occurring in a CAF after evaluation via a vehicle order. All other results, such as multiple function values (double hits) or no function value (zero hits) lead to abort of coding of the control unit.

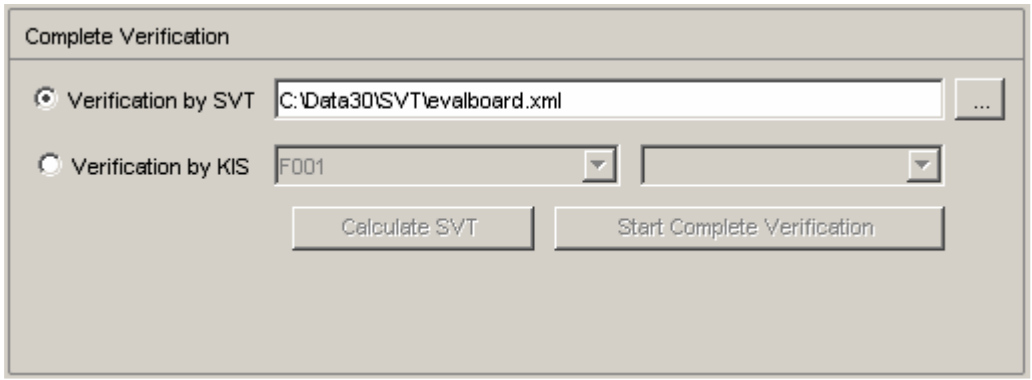
Consequently, the informative value of verification greatly depends on the input variable *Vehicle Order (FA).* The more FA elements (or FP elements) are used for control of coding of a control unit, the greater will be the number of differing vehicle orders required to cover the combinatorial analysis. Ideally, one vehicle order should be created and used for verification for each vehicle profile element used in the CAF.
In the illustration above showing a verification report, the process of determining the parameter to be coded for function *GURTSTRAFFER_BEIFAHRER_1* has failed since no parameter was found for the underlying vehicle order.

The CAF files in list *Available CAF* can be displayed using the context menu →*Open* in the CAF Viewer.



4.3 Verification by SVT/KIS

View *Complete Verification* provides the option of dry verification of CAF files contained in SVTs.



4.3.1 Verification by SVT

Option *Verification by SVT* allows you to load an SVT file in the file system (button ...) and trigger verification for the CAFs that it contains. The SVT is displayed at the top right in the SVT tree after it has been opened. If necessary, you can filter by process classes or open the SVT for editing in the SVT Editor (Process Classes selection box or *Edit*). Button *SVT Reset* deletes the contents of the SVT tree. If all boundary conditions are met (SVT has been loaded, FALactual has been loaded and at least 1 FA has been marked), the total verification can be started with button *Start Complete Verification*. E-Sys then searches all basic variants contained in the SVT and performs the actions as described in Chapter *4.2 Verification of individual Coding Application Files* for any CAFD entries assigned thereunder.
In principle, this type of verification is a string of individual CAF verifications.

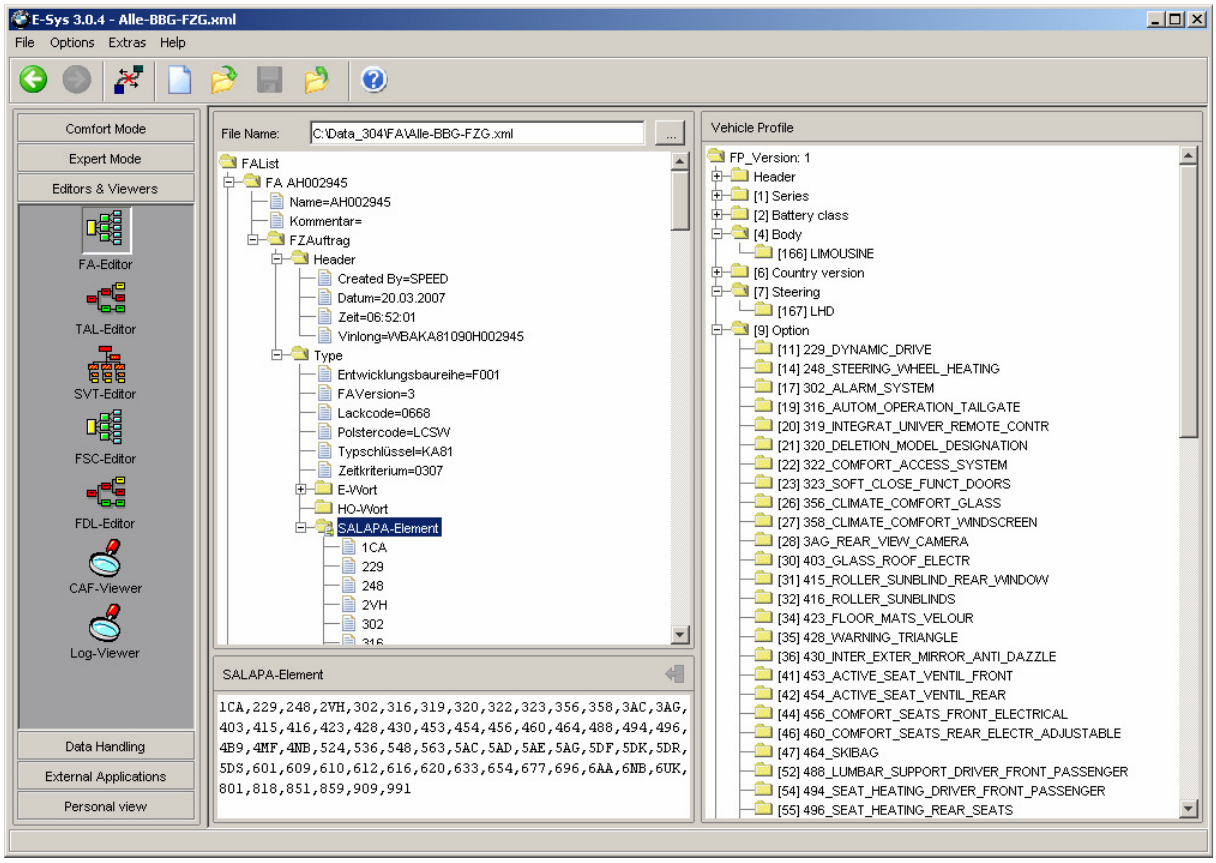
4.3.2 Verification by KIS

Verification by KIS allows total verifications to be conducted on the basis of KIS information. The required KIS knowledge base and the evaluation of logic must be available in the data directory for this purpose (KIS shares are available after importing the corresponding I-Step container). Select a main series in the selection box at the left, and the selection box at the right is then automatically populated with the I-Steps supported by the KIS knowledge base for the main series. After selecting the I-Step to be verified, the next step is to load an FA list, analogously to individual CAF verification, and mark the required vehicle orders (see Chapter *4.2 Verification of individual Coding Application Files*). After you click on the *Start Complete Verification* button, the SVTtarget for the selected I-Step is calculated from KIS for each marked vehicle order, and each software unit of process class CAFD contained therein is checked in an individual CAF verification. If Trace option *SVT Trace* is selected, the calculated SVTtarget is saved in the specified Trace directory for each vehicle order. The nomenclature of the SVT Traces is as follows:
[Absolute position of the FA in the list with which the SVT was generated]_[FAName abbreviated to 15 letters]_[I-Step].svt.
Example: 24_F02US12Zyl_F001-08-09-300.svt
All other Trace options also apply with this verification mode and are described in Chapter *4.2 Verification of individual Coding Application Files*.
In principle, *Verification by KIS* is an automatic string of individual SVT verifications.

5 FA Editor

5.1 GUI

Vehicle orders can be edited with the Vehicle Order Editor (FA Editor). The FA Editor can be found in the toolbar at the left under *Editors and Viewers*. The GUI of the Editor is shown in the illustration below:



The FA Editor has the following controls:

- Toolbar for opening, closing, saving and creating vehicle orders
- Text field and button for displaying and opening FA list files
- Tree view with context menu for display of an FA list
- Text entry field for editing SALAPA elements
- Tree view for displaying a vehicle profile

5.2 Operation

New FA lists can be created and saved and existing FA lists can be opened and closed via the toolbar.

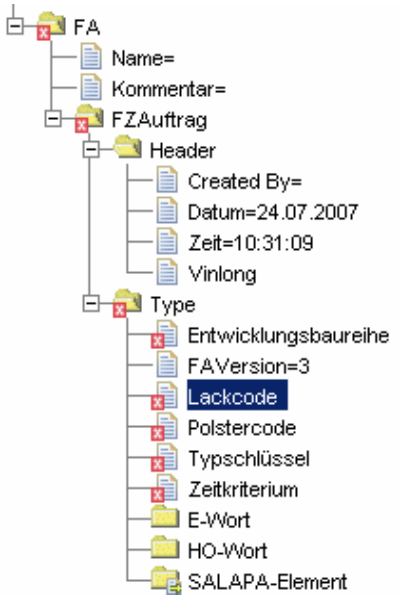
 Create new FA list

 or  Open existing FA list via File dialog box

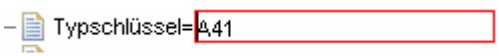
 Save edited FA list

 Close open FA list

If creating a new FA list ("Sheet of paper" symbol), a blank FA framework is created automatically, and mandatory entries to be made in it are marked with a red cross.

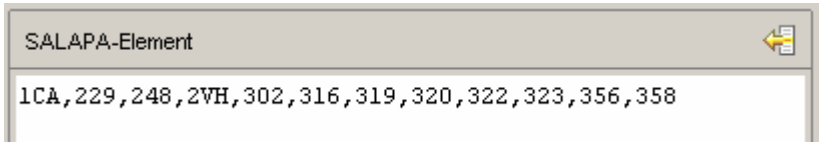


The relevant fields can be populated with the context menu *Edit*. When doing this, note the required length of the relevant entry. To assist you, E-Sys shows the entry fields bordered in red until the entry is complete.

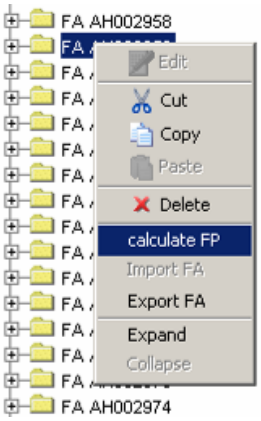


For example, field *Typschlüssel (Model Code)* requires precisely 4 alphanumeric characters.
The VIN number must be entered, including checksum, in the Header field *Vinlong*. However, E-Sys does not offer the option of calculating the checksum!
The SALAPA contents can be entered either in the entry text field *SALAPA-Element* beneath the FA tree or via the context menu *New→SALAPA-Element*. The individual entries must comprise precisely 3 alphanumeric characters. If the SALAPA Elements entry text field be-

neath the FA tree is used, the contents must be separated by commas. Incorrect entries or incomplete entries are shown by a red border around the entry field.



The entry can be transferred to the FA by clicking on icon  .
An analogous procedure is used to edit existing vehicle order lists. Entire vehicle orders can be deleted from the FA list with the context menu at FA level.
Individual vehicle orders (to be more precise: FA lists with only one FA entry) can be imported from the file system to the existing FA list and exported with the Import/Export function. This functionality is required primarily for Expert Mode *TAL processing* since it allows only FA lists with only one vehicle order to be processed.
After all required FA elements have been entered, the vehicle profile can be calculated via the context menu at FA level *calculate FP*, and the vehicle order can consequently be checked for validity.



If a valid vehicle order exists, the vehicle profile is shown in the vehicle profile tree in the right-hand section of the window. Errored entries (unknown SAs, model codes or build level) are displayed as an error message.

Note: Vehicle orders can be edited and new vehicle orders can be created conveniently with tool *FA-Generator* (*FA Generator*) that is also distributed through GIS since all FA elements of a main series including description are available in this case.

6 Expert Mode Vehicle Configuration Management (VCM)

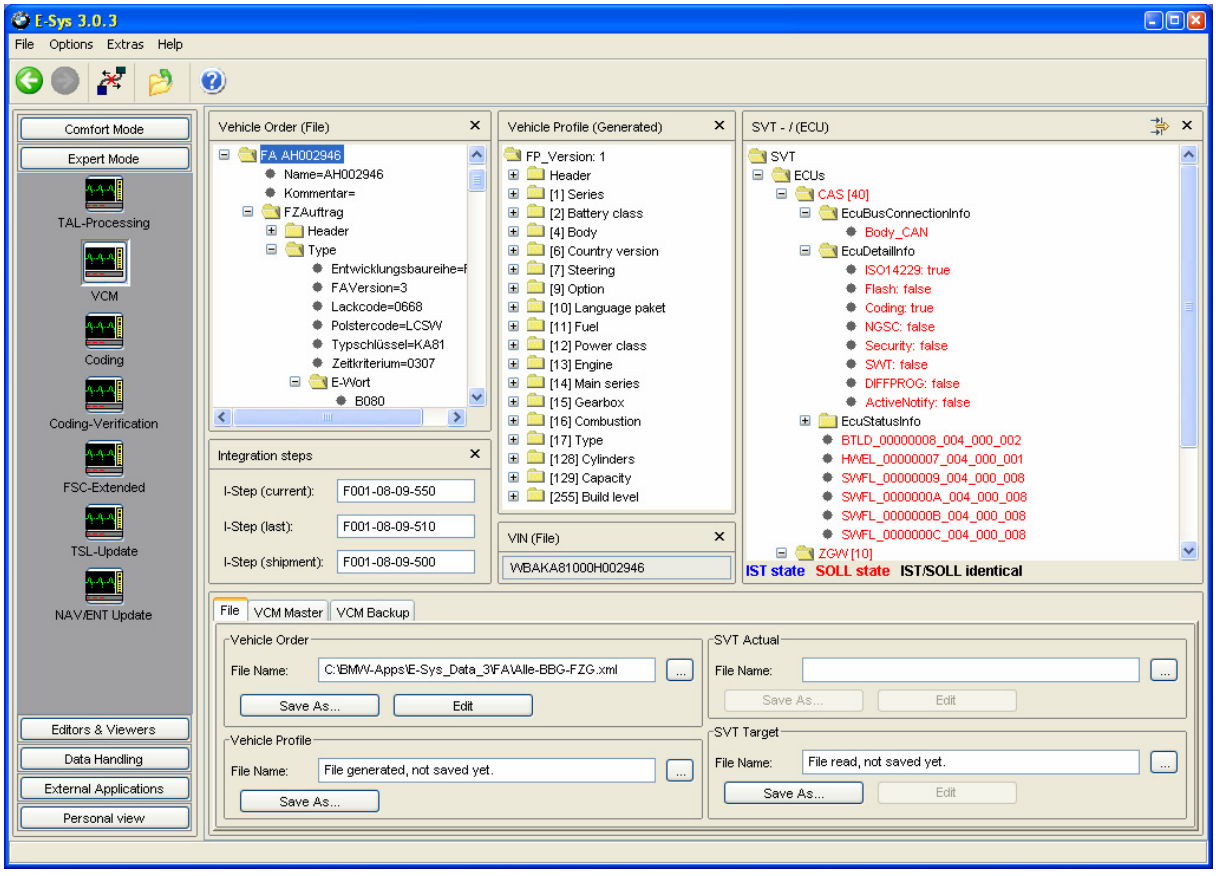
The VCM Expert view is used to read, display and edit the VCM data from the VCM in the vehicle or from files and write the VCM data in the vehicle or to files. The VCM data includes: vehicle order (FA), vehicle profile (FP), system variant table target status (SVTtarget), SVTactual and I-Steps.

The VIN is not a part of the actual data stock of the VCM. It is saved in the ZGW primarily for vehicle recognition via Ethernet. The Master VIN of the vehicle is saved in the VIN Master control unit (CAS). The ZGW automatically fetches the Master VIN in the case of each SwitchToPower, consequently overwriting the ZGW VIN. A diagnostic service can be used to prompt the ZGW to update its VIN with the Master VIN.

The VCM data has already essentially been described in the section on vehicle data. In addition, some of the functions are identical or similar to the functions of Expert Mode Coding and are described in more detail there.

6.1 GUI

The VCM GUI is subdivided into a display area and a control area. The display area is located in the upper section of the VCM GUI, and the control area is located on several tabs beneath.

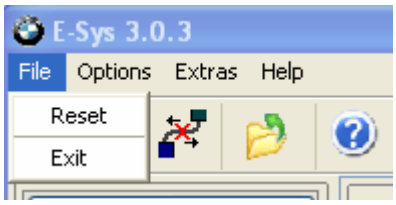


The controls in the control area are arranged so that they correspond to the arrangement of the related data display in the display area.

Display area
The lettering above the displays for FA, FP, SVT and VIN indicates the origin of the data displayed:

- From a file (FA, FP, SVT)
- From the VCM (FA, FP, SVT, I-Steps, VIN)
- From the VCM backup partner (FA, I-Steps, VIN)
- From a calculation (FP)

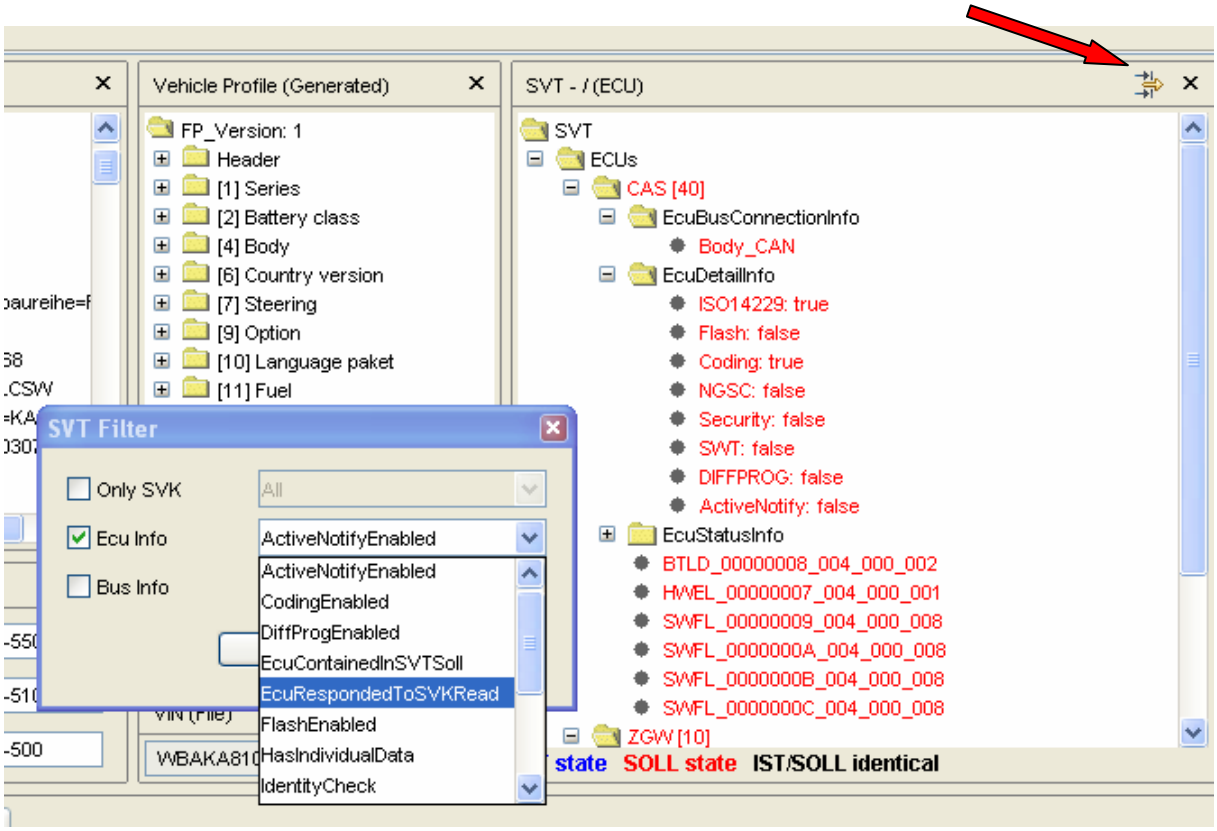
The display area is structured with views so that the sizes of the displays for FA, FP and SVT can be adapted individually.
There is a cross at the top right for each display for rejecting the currently loaded data and deleting the displays.
The File menu, Reset menu entry, can be used to reject all data and delete the contents of all views.



If the FA file has been edited in the FA Editor, the current file is opened, the current FP is reset and the FP display is cleared when you switch back to the VCM Expert view.

If the SVT file has been edited, the current file is opened and, consequently, the SVT display in the VCM Expert view is updated as soon as you switch to the VCM Expert view. An SVT may also contain control unit identification blocks of BN2000 control units besides the SVKs of the BN2010 control units. The PSdZ maps these identification blocks over the structure of an SVK. These control units can be seen in an SVT by the fact that bit ISO14229 is not set in the “ecuDetailInfo” byte. These control units are marked specially in the SVT. An FA from the FA list can be activated via a context menu. It makes no difference in this case from where the FA was loaded/read (file, ECU).
The SVT display allows two SVTs to be loaded simultaneously and compared. An assignment is always made in this case to SVTtarget (red display) and SVTactual (blue display). The elements that are identical in SVTtarget and SVTactual are shown in black.
If a SVTtarget is read from the VCM, the assignment to the SVTtarget in the SVT display is made automatically. The same applies accordingly to the SVTactual. There is no automatic assignment to SVTtarget and SVTactual in the case of SVTs saved in files. In this case, the assignment is made indirectly by opening the file in the corresponding field SVTtarget or SVTactual respectively. In this way, it is then also possible to compare two SVTtargets respectively two SVTactuals with each other.

The SVT display offers a filter function for evaluating the displayed SVTs. The filter settings can be opened with the  icon in the top right-hand corner of the SVT display.

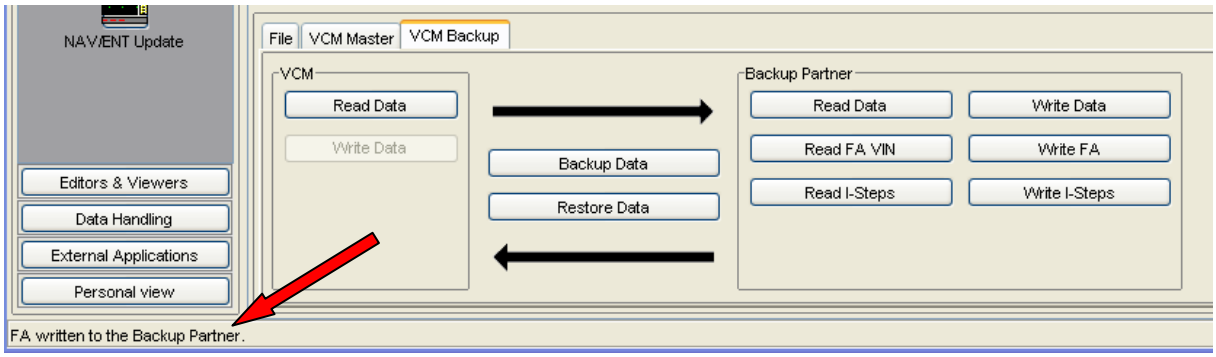


There are three primary filters that are each mutually AND-ed and that can be activated individually via a checkbox.
Filter “Only SVK” offers the option of displaying only a selected process class for a control unit.
“Ecu Info” allows only the control units on which a specific “EcuDetailInfo” respectively “EcuStatusInfo” bit is displayed. The control units contained in the SVTactual but not in the SVTtarget can be shown with setting “NotInSvtSoll_EcuRespToSVK”.
A third “Bus Info” filter offers the option of displaying only control units of a specific communication bus.

Note: The filter settings are not reset by clearing the display.

Activating an FA automatically calculates the related FP and displays it. The text field in which the VIN is displayed cannot be edited.

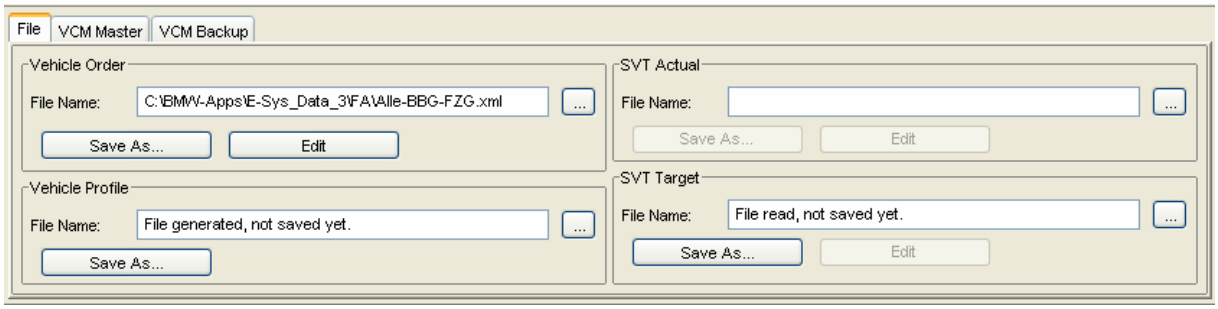
A related status is shown in the status bar for each of the actions executed.



6.2 Operation

6.2.1 File operations

The illustrations below shows the *File* tab:



This tab contains the buttons for loading and saving data from/to a file. The data loaded is shown in the display area.

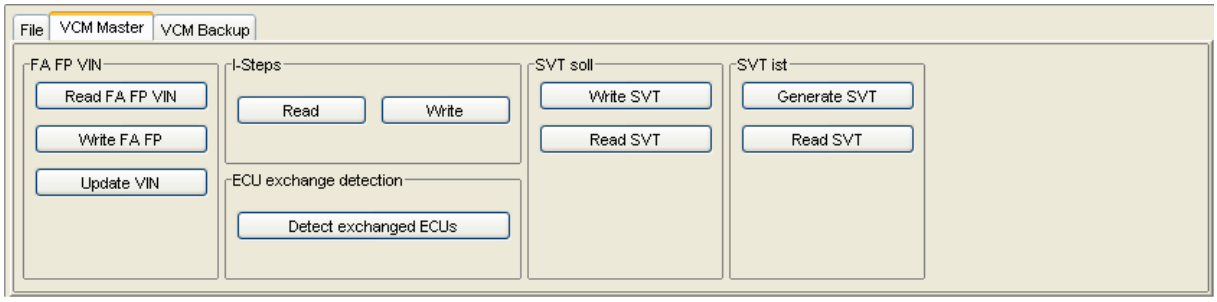
Vehicle order (FA)
A current FA (FA list) can be loaded from a file. The current FA is always saved to a file as a single element of an FA list using *Save As…*. The VIN from the text field of the display area is saved in the FA. The current, saved FA can be edited directly in the FA Editor with *Edit*. If an FA has been read from an ECU beforehand but not yet saved, a corresponding warning message is displayed.

Vehicle profile (FP)
The FP cannot be loaded from a file but can only be calculated from an FA or loaded from the VCM. The current FP is always saved to a file with *Save As…*.

System variant table (SVT)
The current SVT is always saved to a file with *Save As…*.
The current, saved SVT can be edited directly in the SVT Editor with *Edit*. If an SVT has been read from an ECU beforehand but not yet saved, a corresponding warning message is displayed. The assignment to SVTtarget respectively SVTactual is lost when an SVT is saved to a file. When a SVT is loaded from a file, the assignment to SVTtarget respectively SVTactual is made by opening the file in the corresponding SVT target respectively SVT actual area.

6.2.2 VCM Master operations

The illustration below shows the *VCM Master* tab:



This tab contains the buttons for reading and writing data from/to the VCM. The data read is shown in the display area.

FA, FP, VIN
The FA and FP are read from the VCM and the VIN is read from the ZGW with *Read FA FP VIN*. The FA is shown as an individual FA of an FA list. The VIN is not part of the FA from the ECU. However, it is assigned in E-Sys directly to the FA as additional information, i.e. the VIN is also always read when reading the FA from the ZGW/VCM or from an FA file and shown in the display area. The current FA and the current FP can be written to the VCM with W*rite FA FP*. However, only a valid FA-FP combination can be written in this case, i.e. the FP must either have been calculated from the FA or have been read as a valid combination from the VCM and may no longer have been changed.
The VIN can be transferred from the VIN Master (CAS) to the ZGW with *Update VIN*. After this routine has been run, the VIN from the ZGW is displayed in the VIN display. If no CAS is available, a corresponding error message is displayed.

I-Steps
Read reads the entire I-Step data record from the VCM.
Write writes the I-Step data record displayed/entered in the display area to the VCM.

Control unit exchange identification
Detect exchanged ECUs starts a routine in the VCM that polls the serial numbers of all control units and compares them with the serial number reference list. The result of the control unit exchange identification (list with diagnostic addresses of the control units that have been exchanged) is displayed in the SVT display. However, this does pre-require that the SVTtarget have been read out and displayed successfully beforehand. This means that a control unit exchange identification can be executed only if the SVTtarget has been read beforehand from the control unit. If this routine fails, E-Sys displays an error message. Another condition for successful execution of the control unit exchange identification is that a serial number reference list has been created in the VCM. This is done automatically following writing of a SVTtarget to the VCM.

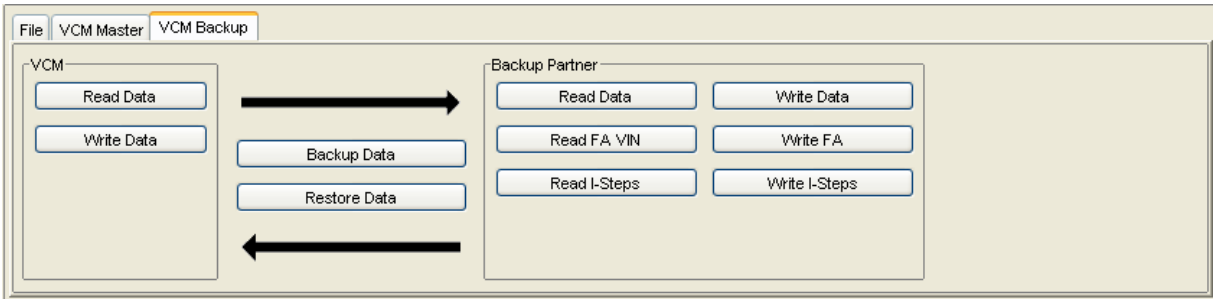
SVTtarget
Read SVT reads the SVTtarget from the VCM and displays it in the SVT display.
Write SVT writes the current SVTtarget to the VCM. If an attempt is made to write a SVTtarget with a wrong version to the VCM (this is checked by PSdZ and prevented), a corresponding error message is displayed in E-Sys. Directly following writing of the SVTtarget, the PSdZ triggers initialisation of the serial number reference list in the VCM. If one of these procedures fails, a corresponding error message is displayed in E-Sys. If an attempt is made to read the SVTtarget if no SVTtarget is saved in the VCM, the query is acknowledged with a negative response. A corresponding error message with a reference to the fact that there is no SVTtarget is displayed in E-Sys.

SVTactual
Read SVT reads the SVTactual from the VCM and displays it in the SVT display.
Generate SVT starts a routine in the VCM that generates the SVTactual from the SVKs of the individual control units signalled back. The result of the SVTactual generation respectively the differences between SVTtarget and SVTactual can be shown in the SVT display and analysed via the filter function.
If an attempt is made to read the SVTactual if there is no SVTactual saved in the VCM, the query is acknowledged with a negative response. A corresponding error message with reference to the fact that there is no SVTactual is displayed in E-Sys.

Note: The control unit list respectively a SVTactual may also contain control units that are not contained in the SVTtarget.

6.2.3 VCM backup operations

The illustration below shows the *VCM Backup* tab:



Tab *VCM Backup* offers the option of saving VCM data (FA, I-Steps) in a backup partner control unit (CAS) and restoring it from the backup partner back in the VCM. This tab contains the buttons for reading and writing data from/to the VCM and from/to the backup partner CAS.

VCM data
Read Data reads all backup-related data (FA, I-Steps) from the VCM and displays it in the display area. The VIN is also read indirectly and assigned to the FA.
Write Data writes all loaded/displayed, backup-related data (FA, calculated FP, I-Steps) to the VCM. The VCM data cannot be written until after a valid FA-FP combination has been created, i.e. until after a FP has been calculated for an FA read from the backup partner control unit.

Note: If applicable, a SVTtarget appropriate for the data restored from the backup must be created separately and written to the VCM.

Backup partner data

Read Data reads all backup-related data (FA, I-Steps) from the backup partner and displays it in the display area. The VIN is also read indirectly and assigned to the FA and is also shown in the VIN display.

Write Data writes all loaded/displayed, backup-related data (FA, I-Steps) to the backup partner.

Read FA VIN reads the FA from the backup partner and displays it in the FA display area. *Write FA* writes the loaded/displayed FA to the backup partner. *Read I-Steps* reads the I-Steps from the backup partner and displays them in the I-Steps display area. *Write I-Steps* writes the loaded/displayed I-Steps to the backup partner.

Master Backup functions

Back up data reads all backup-related data (FA, I-Steps) from the VCM, displays it in the display area and writes it to the backup partner. The VIN is also read indirectly in this case with the FA but is not written to the backup partner.

Restore Data reads all backup-related data (FA, I-Steps) from the backup partner, displays it in the display area and writes it to the VCM. The VIN is also read from the backup partner and assigned to the FA. After the data has been read, a valid FP is calculated for the FA and is then written together with the FA and the I-Steps to the VCM.

7 List of abbreviations

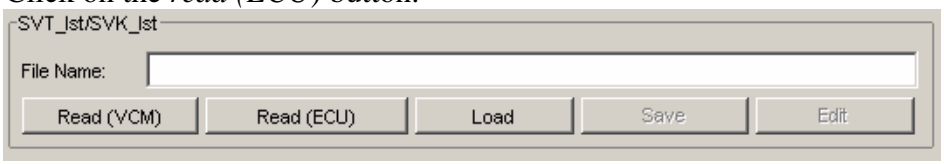
Abbreviation	Significance
ASBC	Automotive Security Backup Center
CAF	Coding Application File
CAFD	Process class designation of the CAF
EST	Developer Soft Token
FA	Fahrzeugauftrag = Vehicle order
FA2FP	Assignment table of vehicle order to vehicle profile
FDL	Function data list
FP	Vehicle profile
I-Step	Integration Step for software updates
KIS	Configuration information system
NCD	Net coding data
ODX	Open Diagnostic Data Exchange
PDX	Packaged ODX
SALAPA	Optional extra, country spec. package/(equipment) packages
SGBM-ID	Control unit description model identification number
SVK	Software variant code
SVT	System variant table
TAL	Transaction list
VCM	Vehicle Configuration Management
VIN	Vehicle Identification Number (VIN number)

A Coding Tutorial

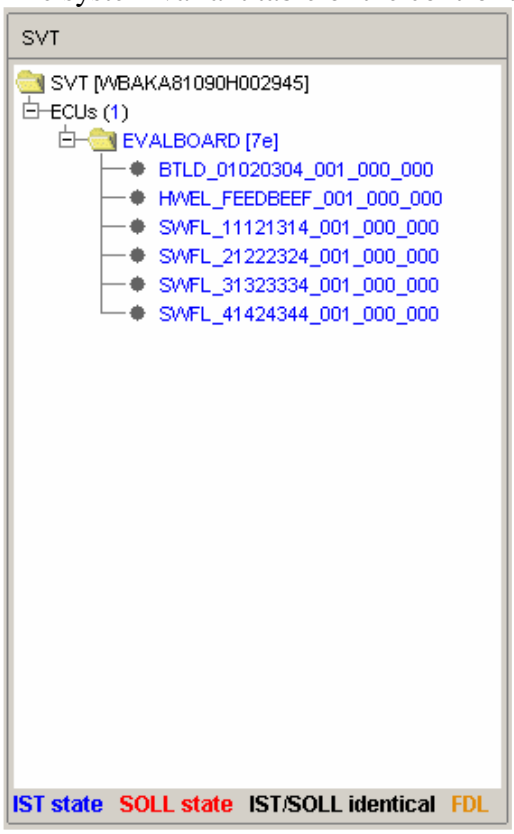
It is assumed that a connection has been set up to a control unit/vehicle electrical system for the procedures described below.

A.1 Coding uncoded control unit without KIS support

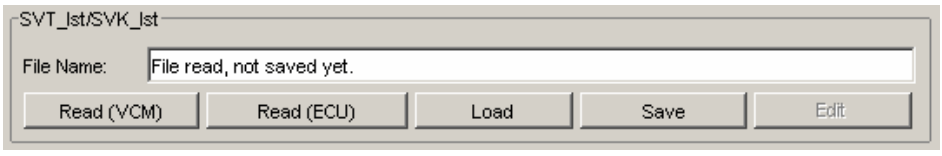
- Switch to Expert Mode *Coding*.
- Click on the *read (ECU)* button.



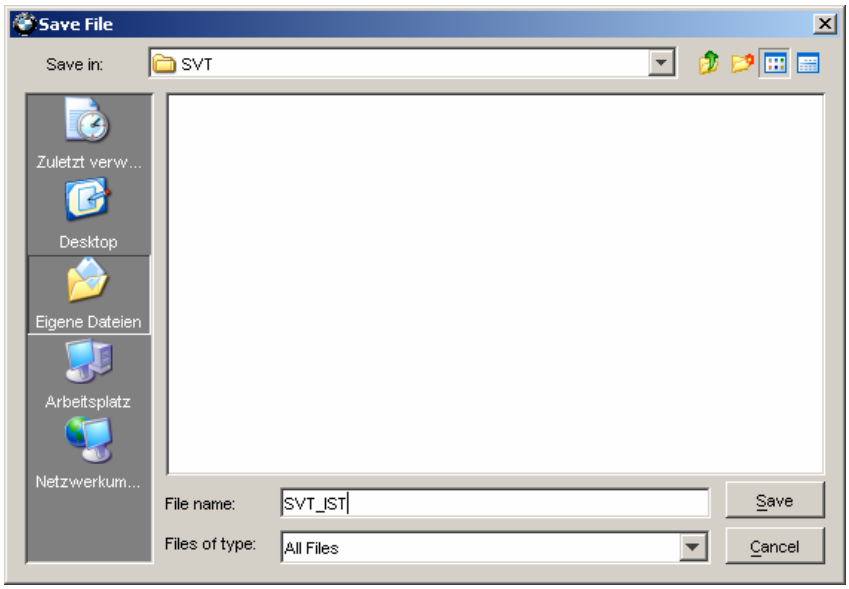
- The system variant table of the control unit is displayed.



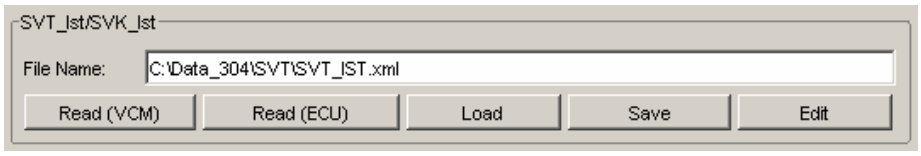
- Click on the *Save* button.



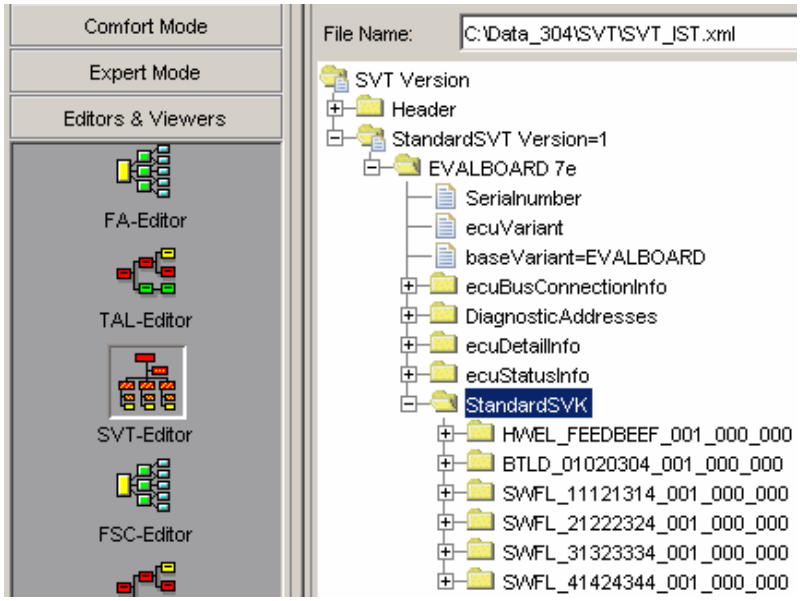
- The *Save File* dialog box is displayed. Enter the file name (e.g. SVT_ACTUAL) and click on the *Save* button.



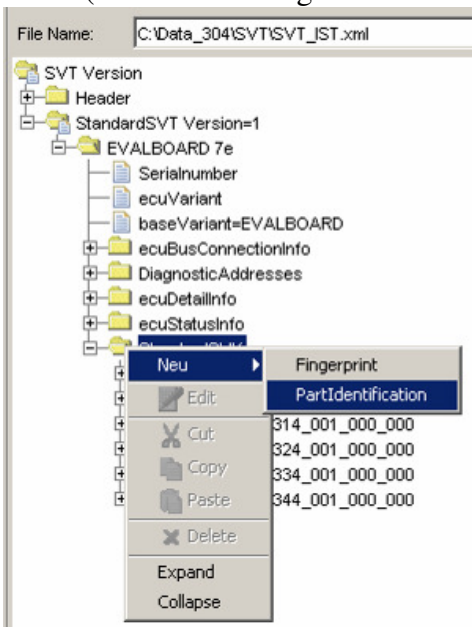
- The system variant table is saved.
- Click on the *Edit* button.



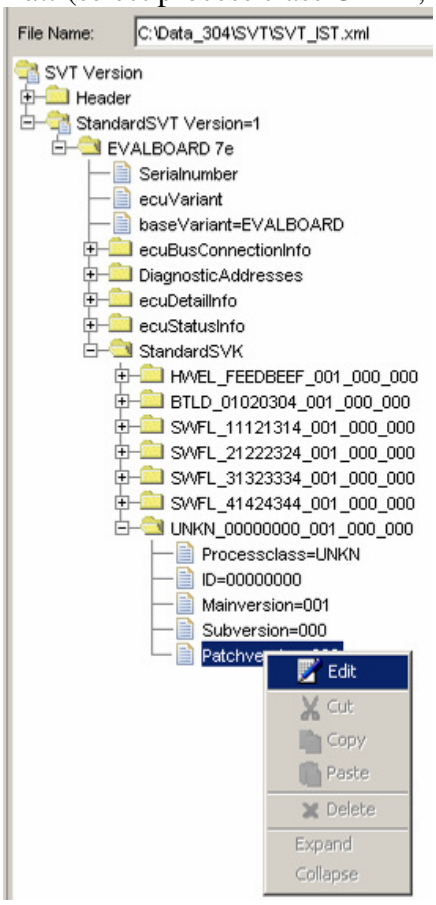
- The system variant table of the control unit is displayed in the SVT Editor. Expand *StandardSVT* and then *StandardSVK* in the tree.



- Mark *StandardSVK* in the tree and select *New* → *PartIdentification* via the context menu (click with the right mouse button).

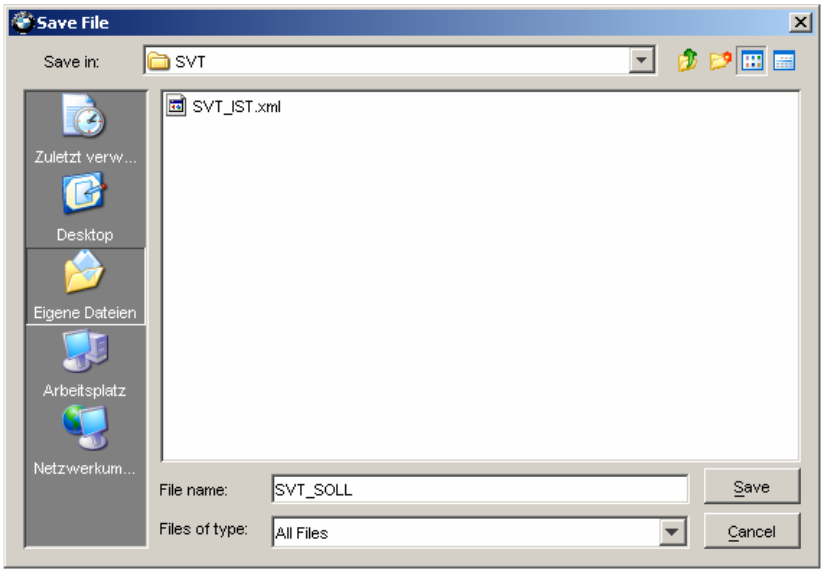


- A new entry *UNKN_00000000_001_000_000* has now been created under *StandardSVK*.
- Expand element *UNKN_00000000_001_000_000* in the tree. Adapt all elements beneath it in line with the CAF to be coded. In order to do this, mark the elements individually and then use the context menu (click with the right mouse button) to select *Edit* (select process class CAFD, enter ID, MV, SV and PV of the required CAF)

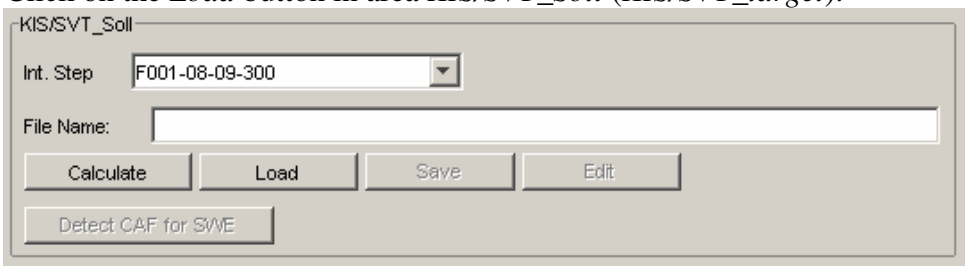


- Select *File* → *Save As* ... in the menu bar.

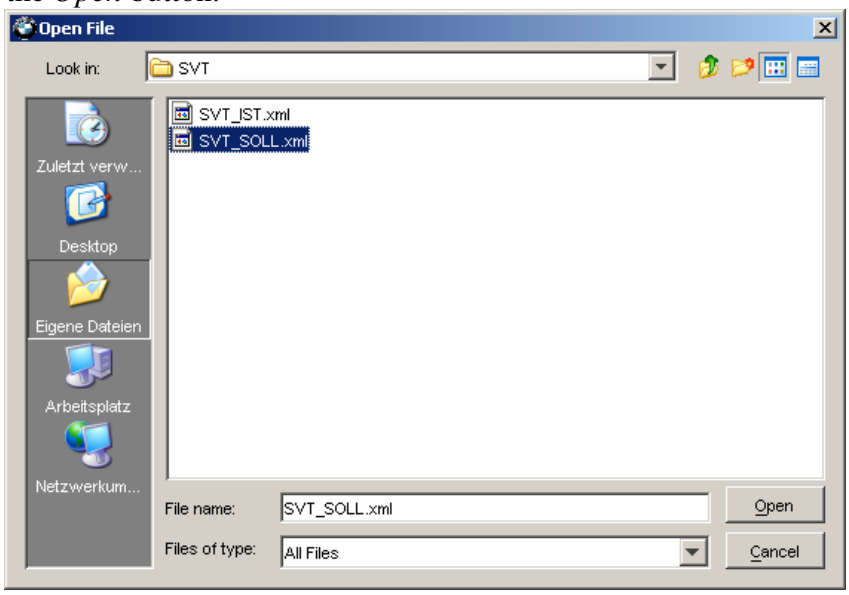
- The *Save File* dialog box is displayed. Enter the file name (e.g. SVT_TARGET) and click on the *Save* button.



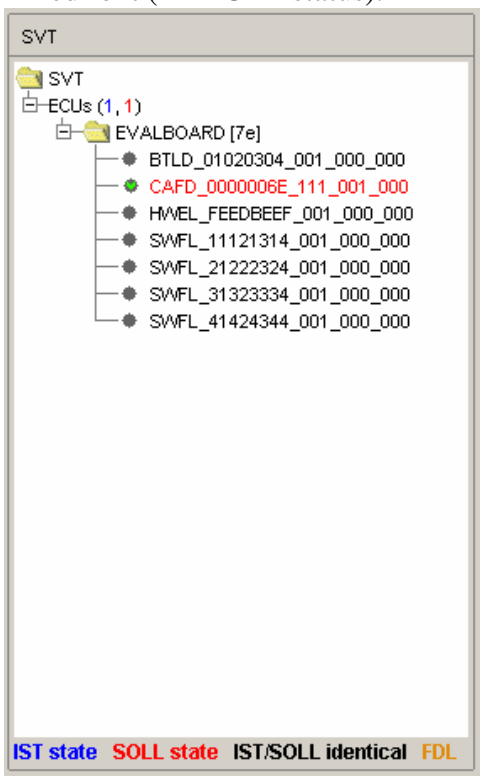
- Switch to Expert Mode *Coding*.
- Click on the *Load* button in area *KIS/SVT_Soll* (*KIS/SVT_target*).



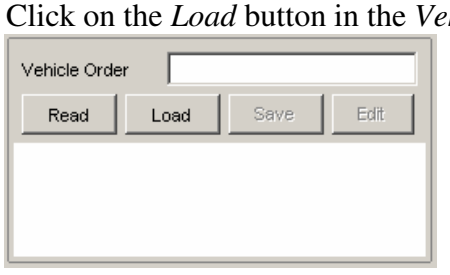
- The *Open File* dialog box is displayed. Select the file (e.g. SVT_SOLL) and click on the *Open* button.



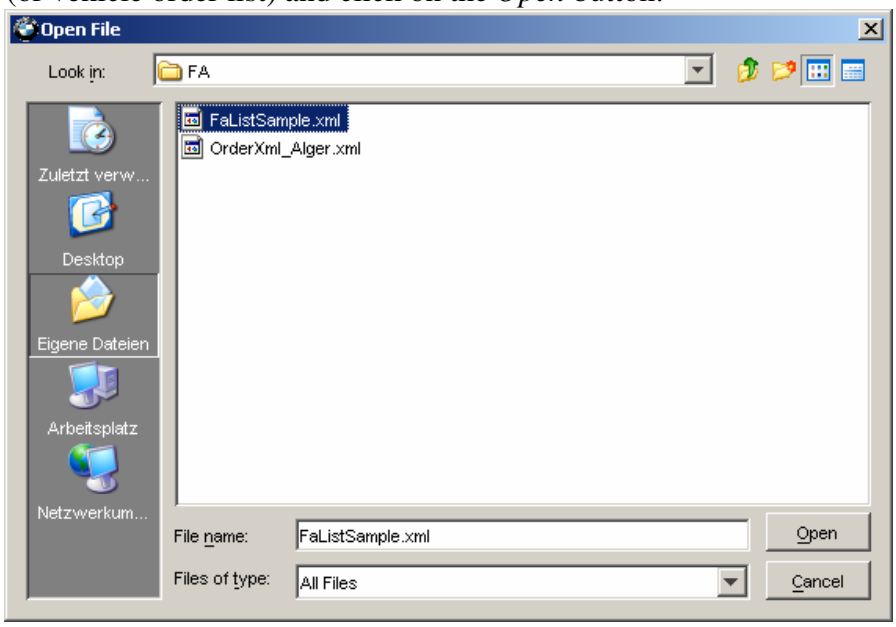
- The display of the system variant table is updated. The CAF to be coded is displayed in red font (TARGET status).



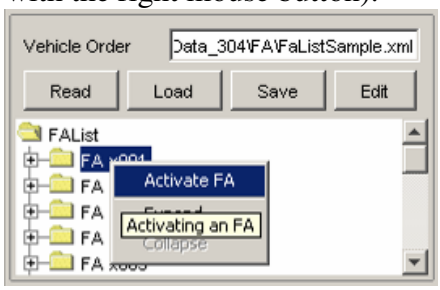
- Click on the *Load* button in the *Vehicle Order* area.



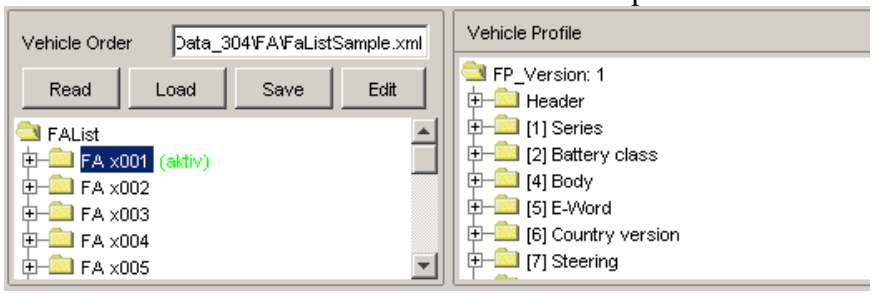
- The *Open File* dialog box is displayed. Select the file with the required vehicle order (or vehicle order list) and click on the *Open* button.



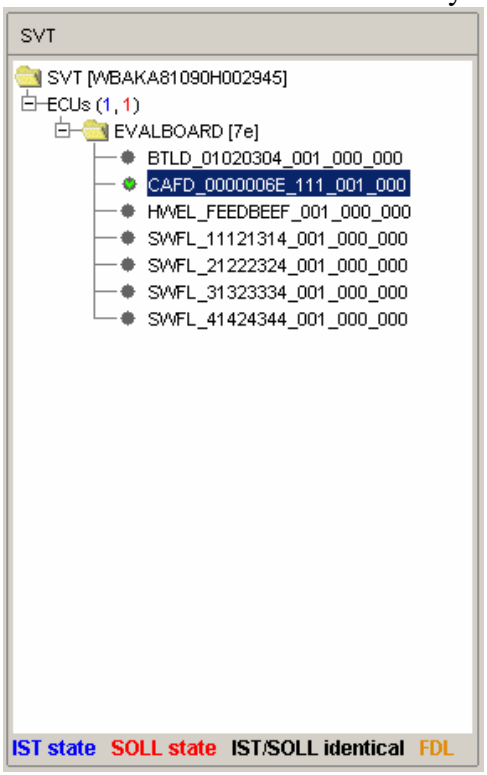
- The vehicle order (or the vehicle order list) is displayed.
- Mark the required vehicle order and select *Activate FA* using the context menu (click with the right mouse button).



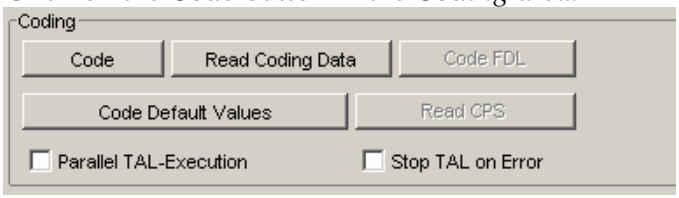
- The vehicle order is set to active and the vehicle profile is calculated and displayed.



- Mark the CAF to be coded in the system variant table.

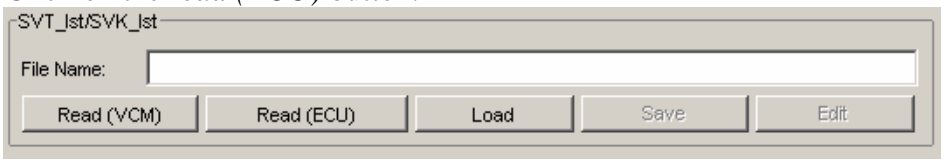


- Click on the *Code* button in the *Coding* area.

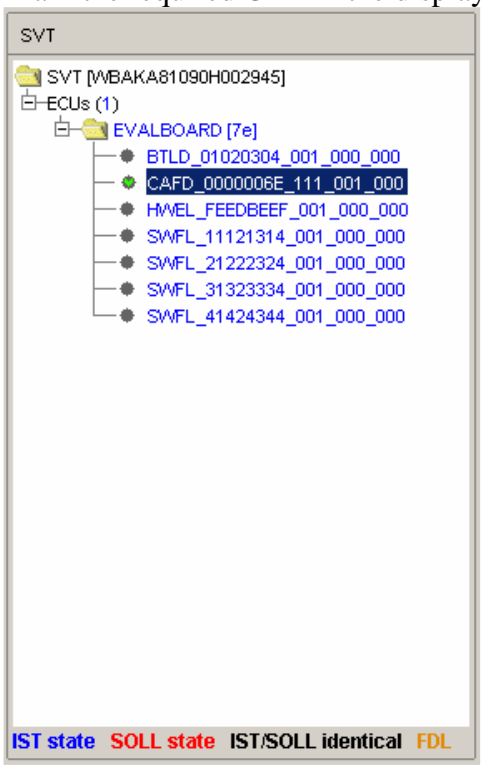


A.2 Creating/editing and coding FDL

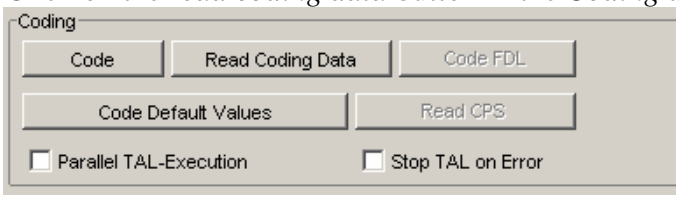
- Switch to Expert Mode *Coding*.
- Click on the *read (ECU)* button.



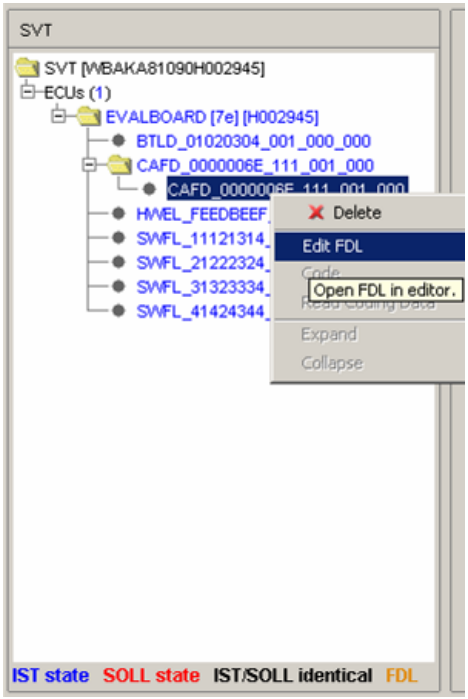
- Mark the required CAF in the displayed system variant table of the control unit.



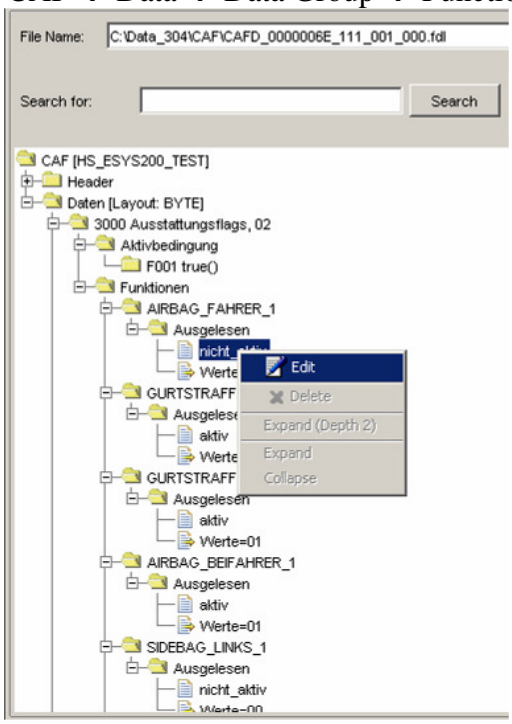
- Click on the *read coding data* button in the *Coding* area.



- An FDL is created in the system variant table beneath the marked CAF. Mark this element and select *edit FDL* via the context menu (click with the right mouse button).

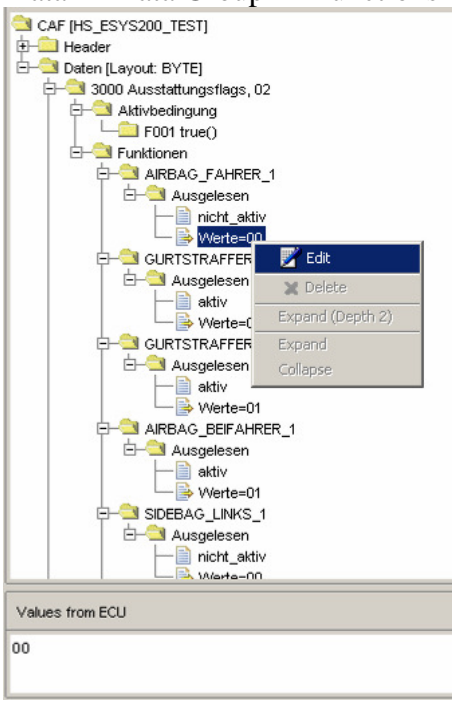


- The FDL is displayed in the FDL Editor.
- Either edit the function values read out.
  - In order to do this, mark the required function value read out and select *Edit* via the context menu (click with the right mouse button) (in the tree beneath CAF → Data → Data Group → Functions → Function Names → Read Out).

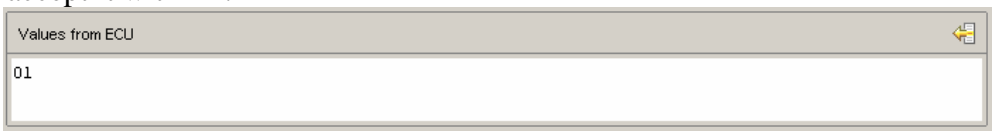


- Select a function value in the selection dialog box displayed. The function values defined in the CAF are available in the selection dialog box.

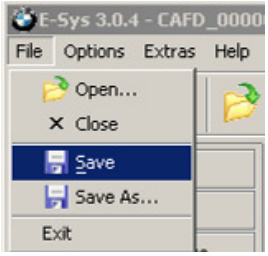
- Or edit values read out.
  - In order to do this, mark the required value read out and select *Edit* via the context menu (click with the right mouse button) (in the tree beneath CAF → Data → Data Group → Functions → Function Names → Read Out).



  - Edit the value in the bottom area of the window under *Values from ECU* and accept it with .

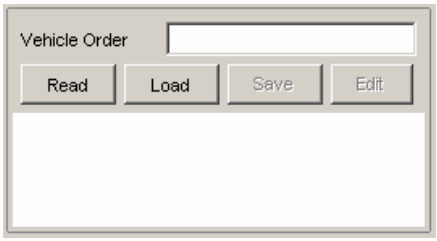


- Select *File* → *Save* in the menu bar.

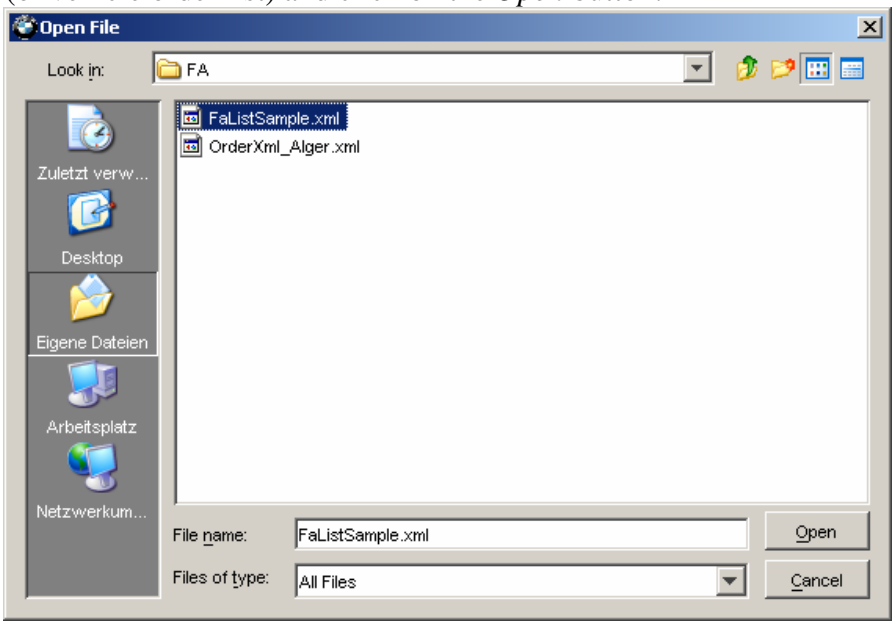


- Switch to Expert Mode *Coding*.

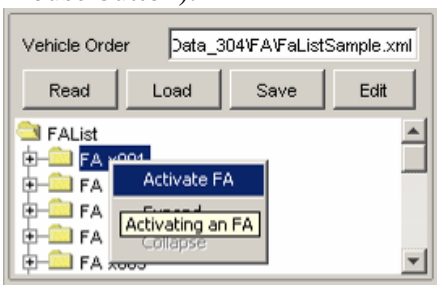
- Click on the *Load* button in the *Vehicle Order* area.



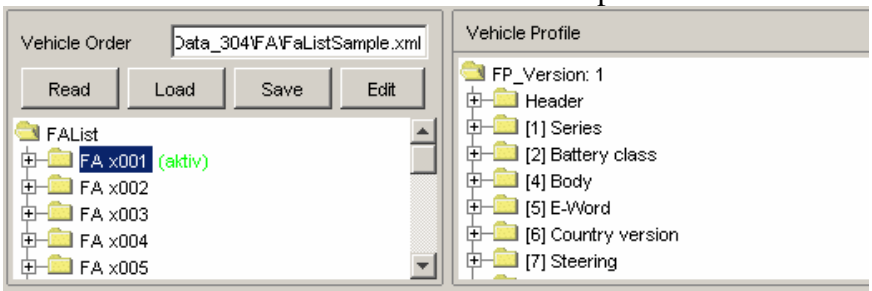
- The *Open File* dialog box is displayed. Select the file with the required vehicle order (or vehicle order list) and click on the *Open* button.



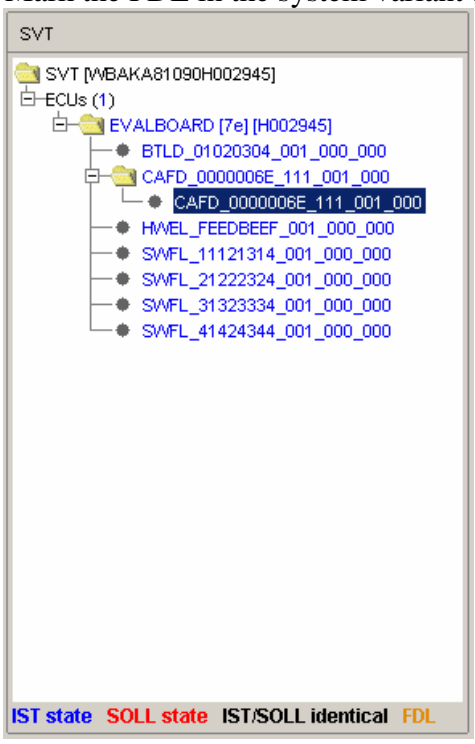
- The vehicle order (or the vehicle order list) is displayed.
- Mark a vehicle order and select *activate FA* via the context menu (click with the right mouse button).



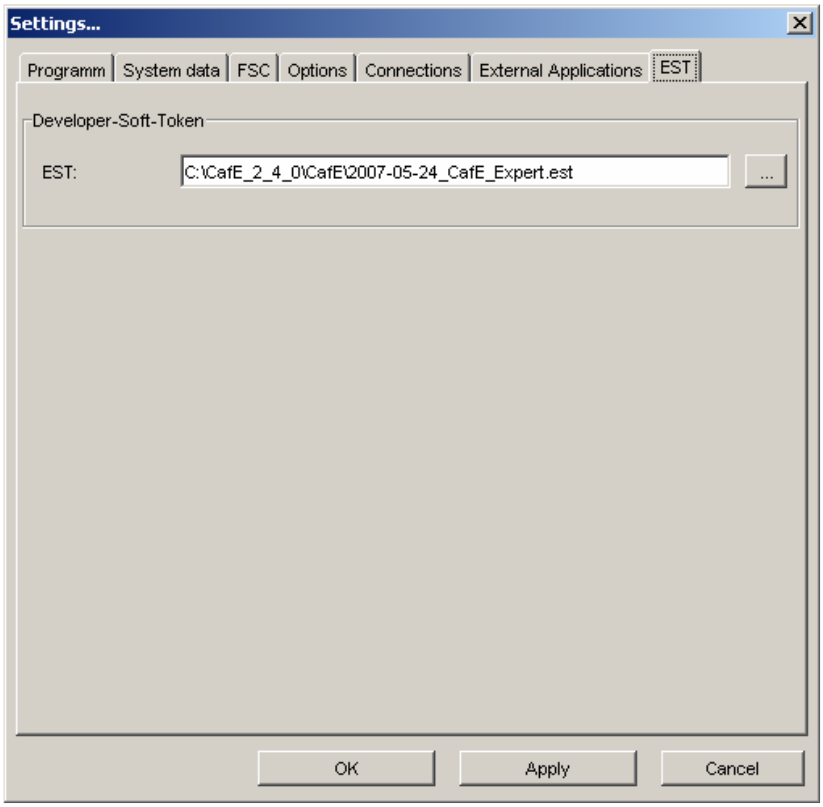
- The vehicle order is set active and the vehicle profile is calculated and displayed.



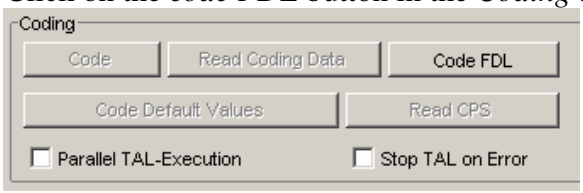
- Mark the FDL in the system variant table.



- If no EST has yet been selected, select *Options* → *Settings...* → *EST* in the menu bar and enter the file location of the EST.



- Click on the *code FDL* button in the *Coding* area.



- Enter the PIN for the personal, loaded EST and confirm with *OK*.