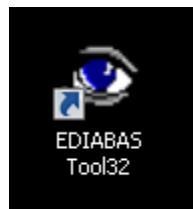


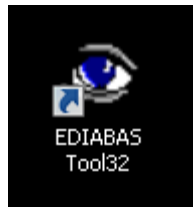
How2EDIABAS TOOL 32



04.06.2019

altran

Inhalt



1 Steuergeräte-Beschreibungsdatei
(SGBD)

2 Fehlerspeicher Lesen & Löschen

3 Steuergeräteverbaukennung
svk_lesen

4 Steuern

5 Info-Speicher

6 Konfiguration

7 Jobs

8 Test erstellen

9 Fenster

10 Hilfe & Shortcuts

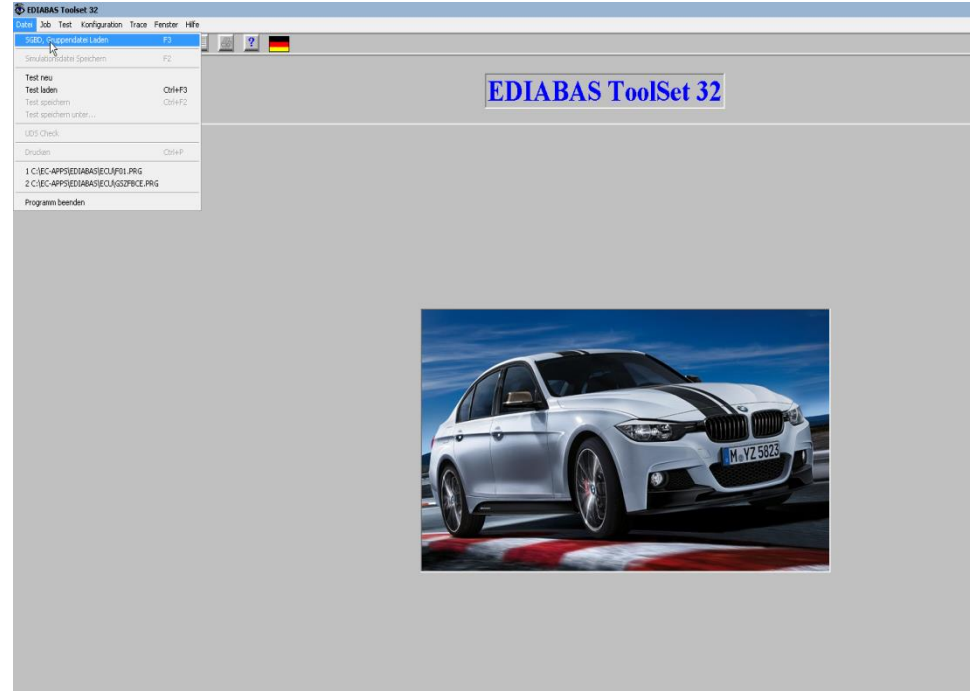
1.

Steuergeräte-Beschreibungsdatei
(SGBD)

1. SGBD

Die SteuerGeräteBeschreibungsDateien(SGBD) beinhalten das Steuergerätspezifische Wissen. Jede SGBD enthält die Auftragstelegramme des betreffenden Steuergeräts und decodiert dessen Antworttelegramme. Dabei werden die Rohdaten (Bytes) indirekt verwendbare Größen (Datentypen: int, long, real, string) umgesetzt d.h der Fehlercode wird in Fehlertext umgewandelt.

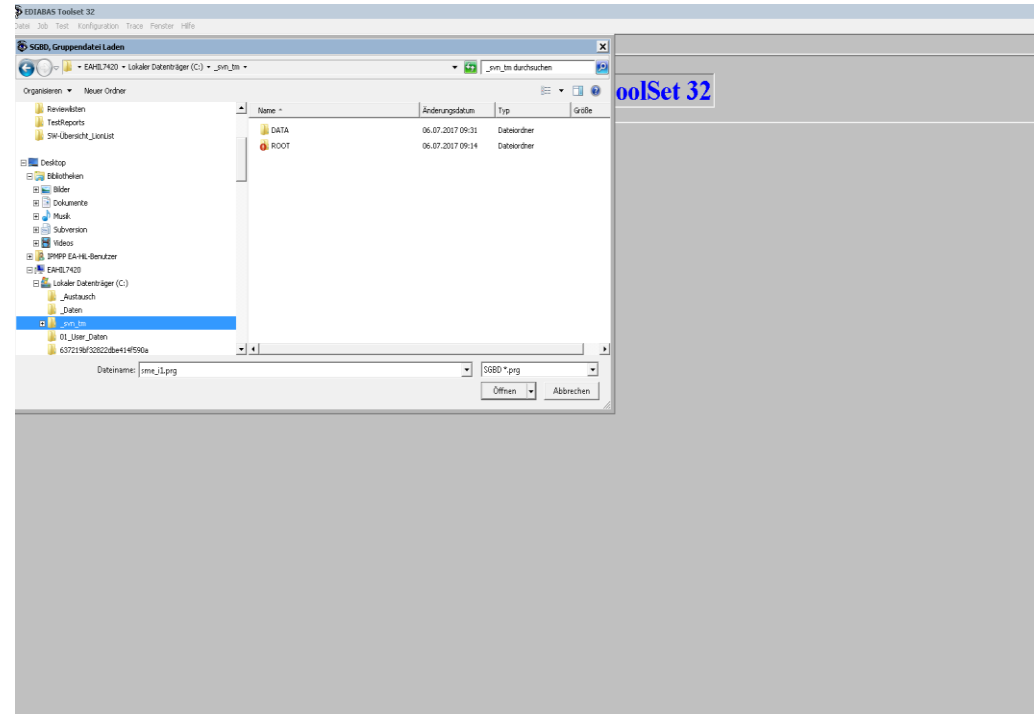
Für jede Steuergerätvariante gibt es **GENAU** eine variantenspezifische SGBD



1.1 SGBD (Beispiel)

Die Seite 5 bis 10 illustriert wie man der SGBD zu jedem Steuergerät findet.

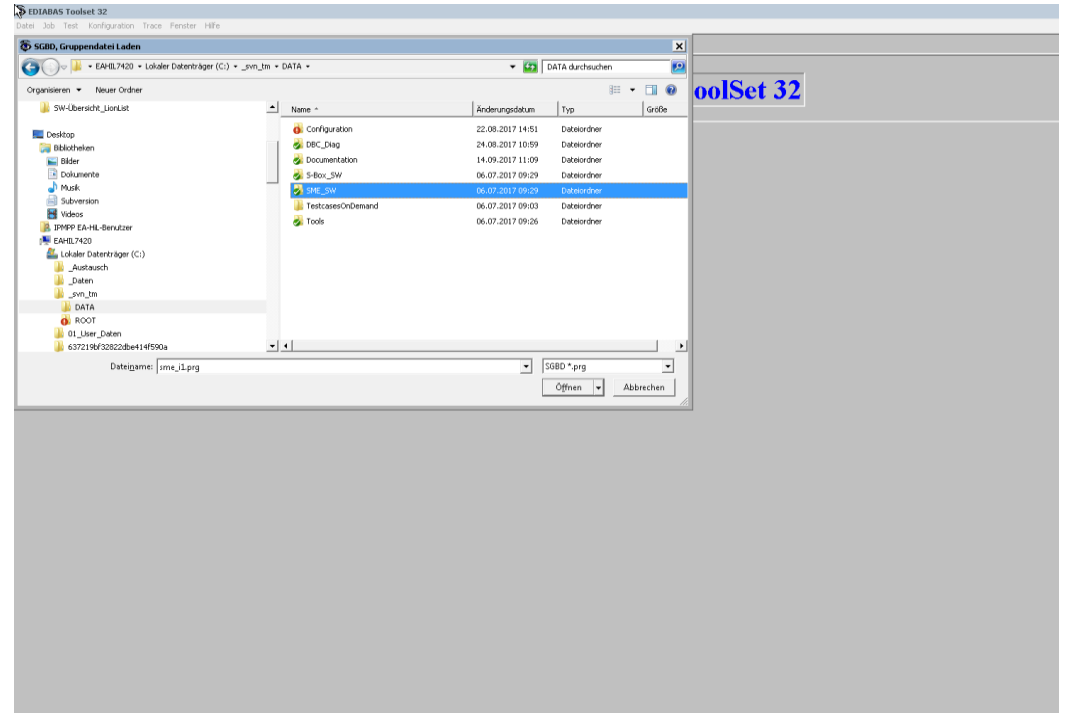
1. Subversion (SVN) klicken.



1.2 SGBD (Beispiel)

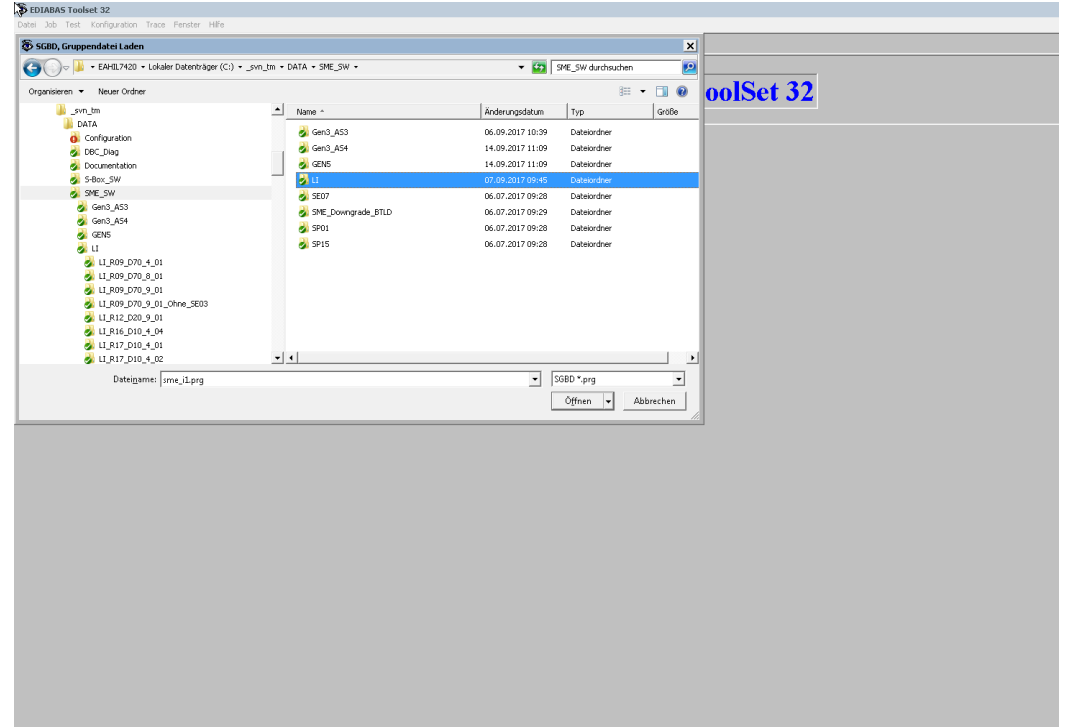
2. SME_SW

In der SME_SW sind alle Projekte zu finden



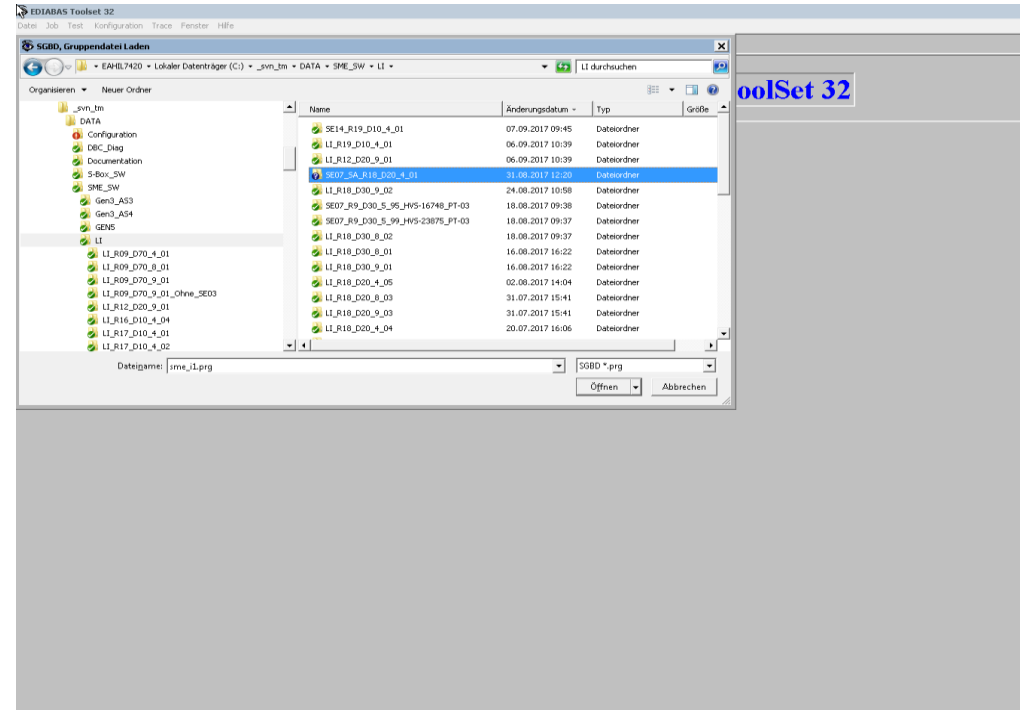
1.3 SGBD (Beispiel)

3. Spezifisch zu diesem Beispiel wird aus der LI Projekt die SGBD ausgewählt



1.4 SGBD (Beispiel)

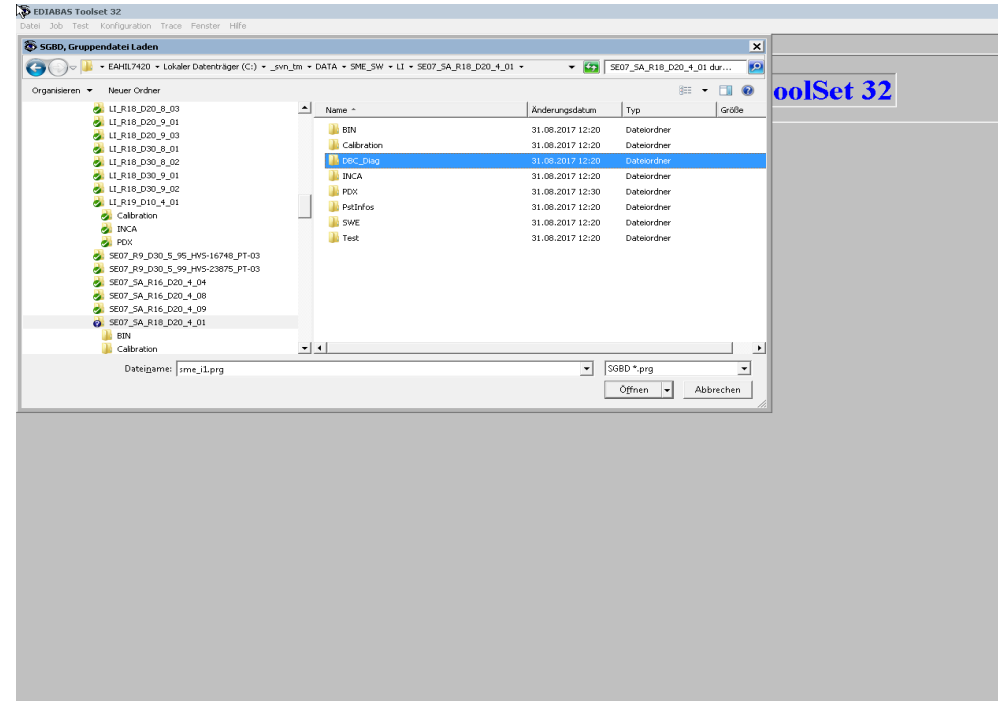
4. Projekt SE07_SA_R18_D30_4_01
wird gewählt.



1.5 SGBD (Beispiel)

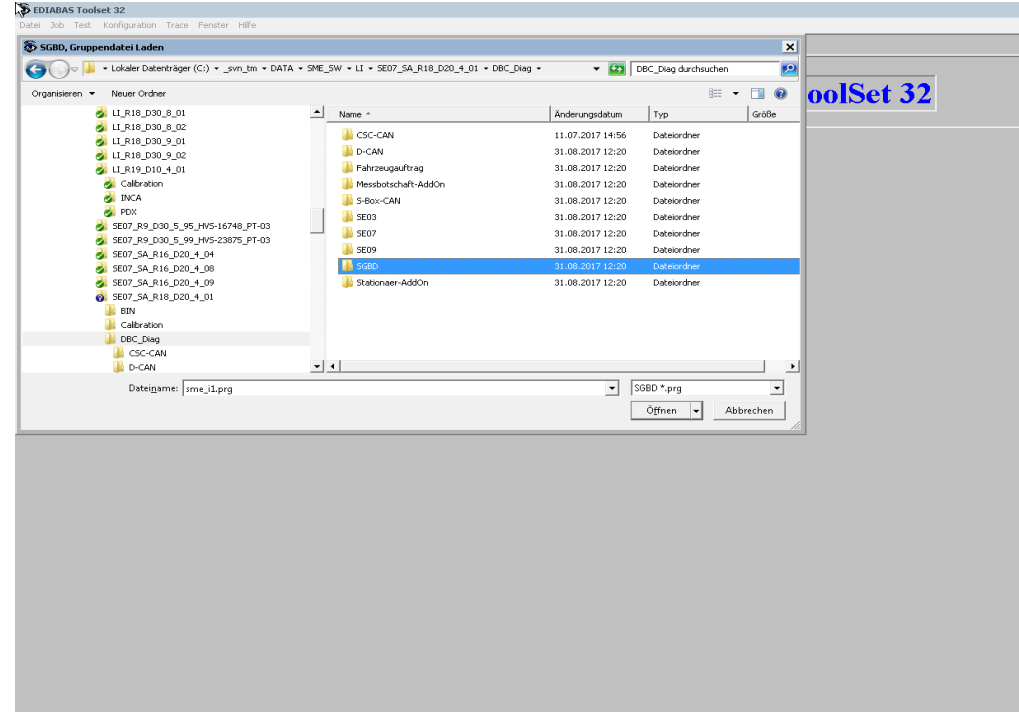
5. DBC_Diag

In der DBC_Diag(Data Bus Controller) liegt nicht nur der SGBD Ordner sondern auch die 4-CAN's und der Fahrzeugauftrag



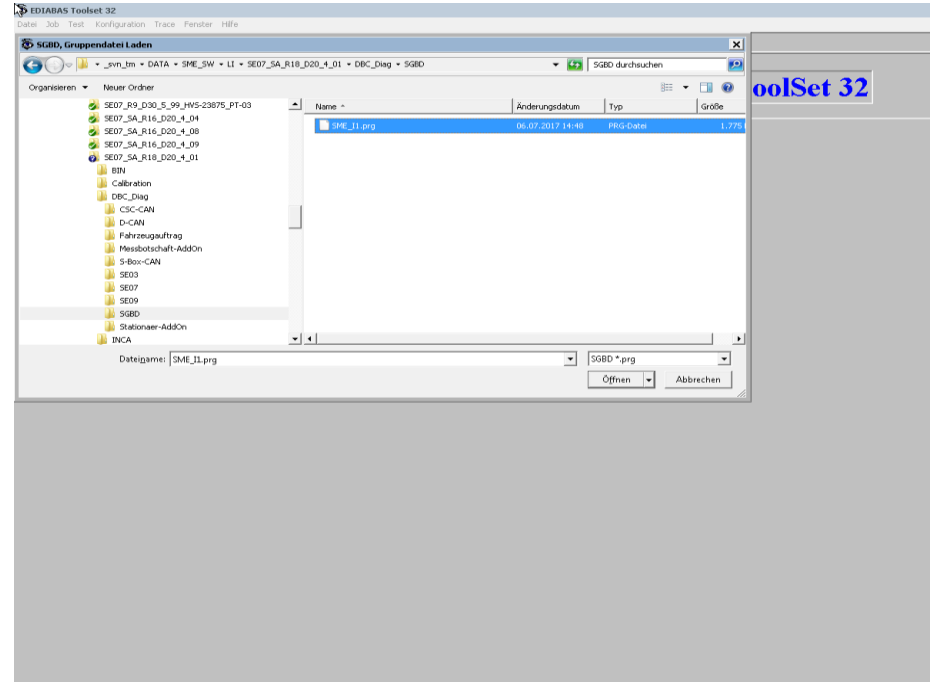
1.6 SGBD (Beispiel)

6. SGBD Ordner



1.7 SGBD (Beispiel)

In der SGBD-Ordner liegt dann der Datei in PRG Format.



2.

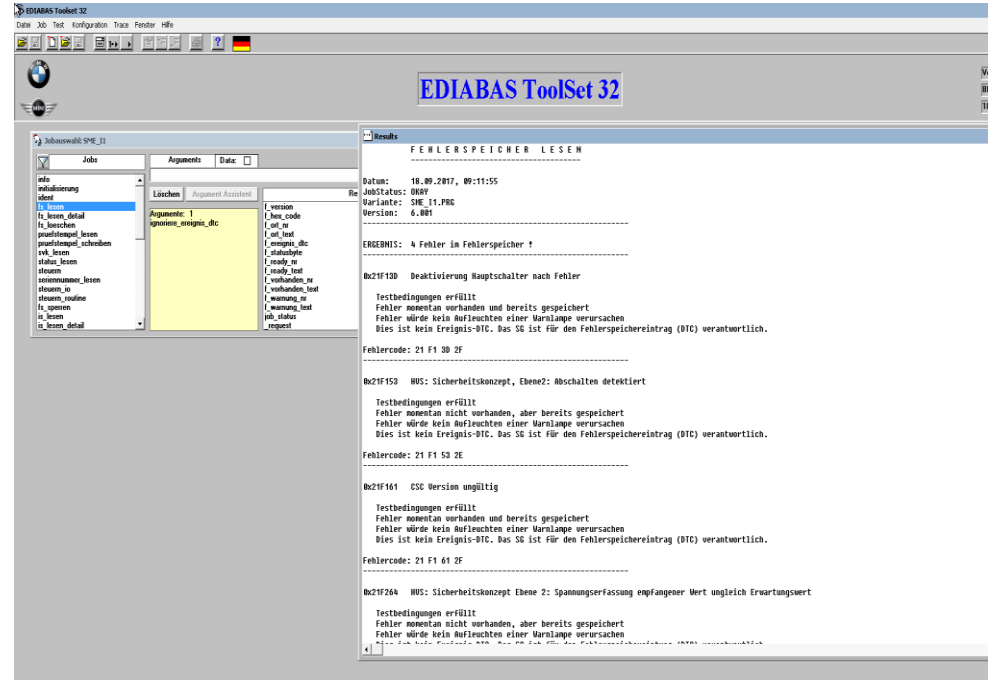
Fehlerspeicher Lesen und Löschen

2.1 Fehlerspeicher Lesen

Mit dem Job **fs_lesen** hat man die Möglichkeit, mit der Hilfe von EDIABAS, Fehler vom SG zu lesen. Die Fehlern sind in der Regel sehr detailliert beschrieben.

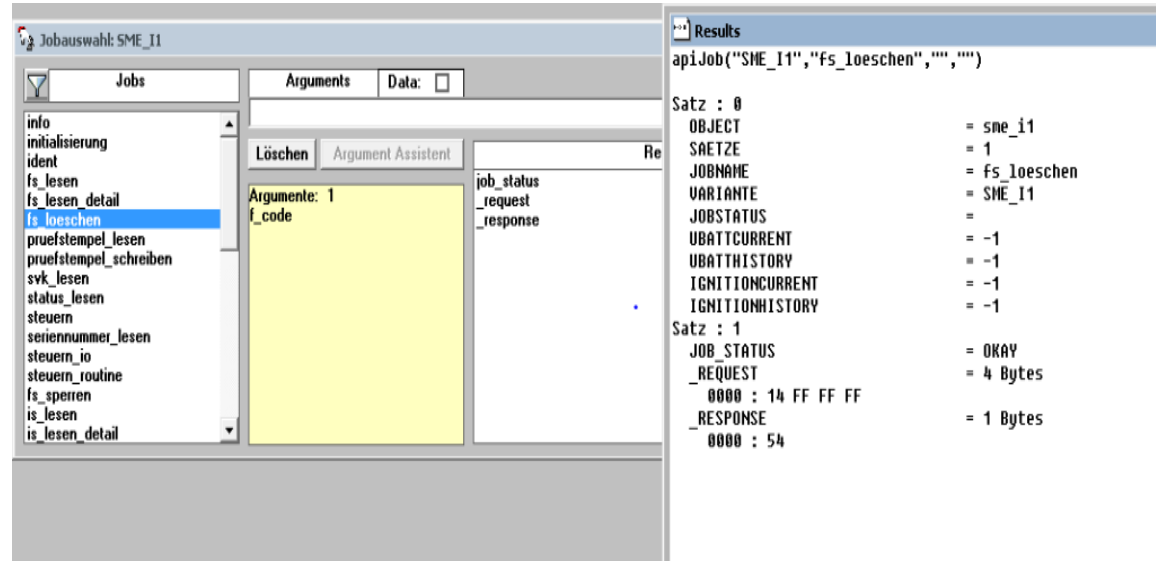
Mit Hilfe der Fehlercodes kann man ebenso – im Ordner „Calibration“ -> DTC-ERRCAT- Fehler über den HEX-Code entschlüsseln.

Fehlerkategorie	Botschaft	Botschaft-ID	Signal (A-CAN)	Signalbelegung	Reaktion
CAT0				00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	Error memory entry
CAT1	ST_HVSTO_1	506(1FAh)	ST_PRCNG_LCKD	00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	Precharge locked
CAT2	ST_HVSTO_1	506(1FAh)	ST_EMODD_HVSTO	00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	CAN error (limp home)
CAT3	ST_HVSTO_1	506(1FAh)	RQ_SER_HVSTO	00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	service request
CAT4				00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	
CAT5	STAT_HVSTO_2	274(112h)	RQ_OPC_CHG_HVSTO	00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	zero current request
CAT6	STAT_HVSTO_2	274(112h)	RQ_OPN_DCSW_HVSTO_FAST	00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	fast open request: waiting for t1 = (adjustable) 2 s
CAT7	STAT_HVSTO_2	274(112h)	RQ_OPN_DCSW_HVSTO_ILY	00 = Keine Aussage möglich 01 = nicht aktiv 10 = aktiv 11 = Signal ungültig	immediate open request: Open within t4 = (adjustable) 140ms



2.2 Fehlerspeicher Löschen

Mit dem Job **fs_loeschen** kann man, mit der Hilfe von EDIABAS, Fehler vom SG löschen.



The screenshot shows the Ediabas Job Selection window with the job **fs_loeschen** selected. The 'Arguments' tab is active, showing a single argument **f_code**. The 'Results' window on the right displays the output of the job, including the API call and the status of the operation.

Jobs

- info
- initialisierung
- ident
- fs_lesen
- fs_lesen_detail
- fs_loeschen**
- pruefstempel_lesen
- pruefstempel_schreiben
- svk_lesen
- status_lesen
- steuern
- seriennummer_lesen
- steuern_io
- steuern_routine
- fs_sperren
- is_lesen
- is_lesen_detail

Arguments

Argumente: 1
f_code

Results

```
apiJob("SME_I1","fs_loeschen","","")
```

Satz : 0

OBJECT	= sme_i1
SAETZE	= 1
JOBNAME	= fs_loeschen
VARIANTE	= SME_I1
JOBSTATUS	=
UBATTCURRENT	= -1
UBATTHISTORY	= -1
IGNITIONCURRENT	= -1
IGNITIONHISTORY	= -1

Satz : 1

JOB_STATUS	= OKAY
_REQUEST	= 4 Bytes
0000 : 14 FF FF FF	
_RESPONSE	= 1 Bytes
0000 : 54	

3.

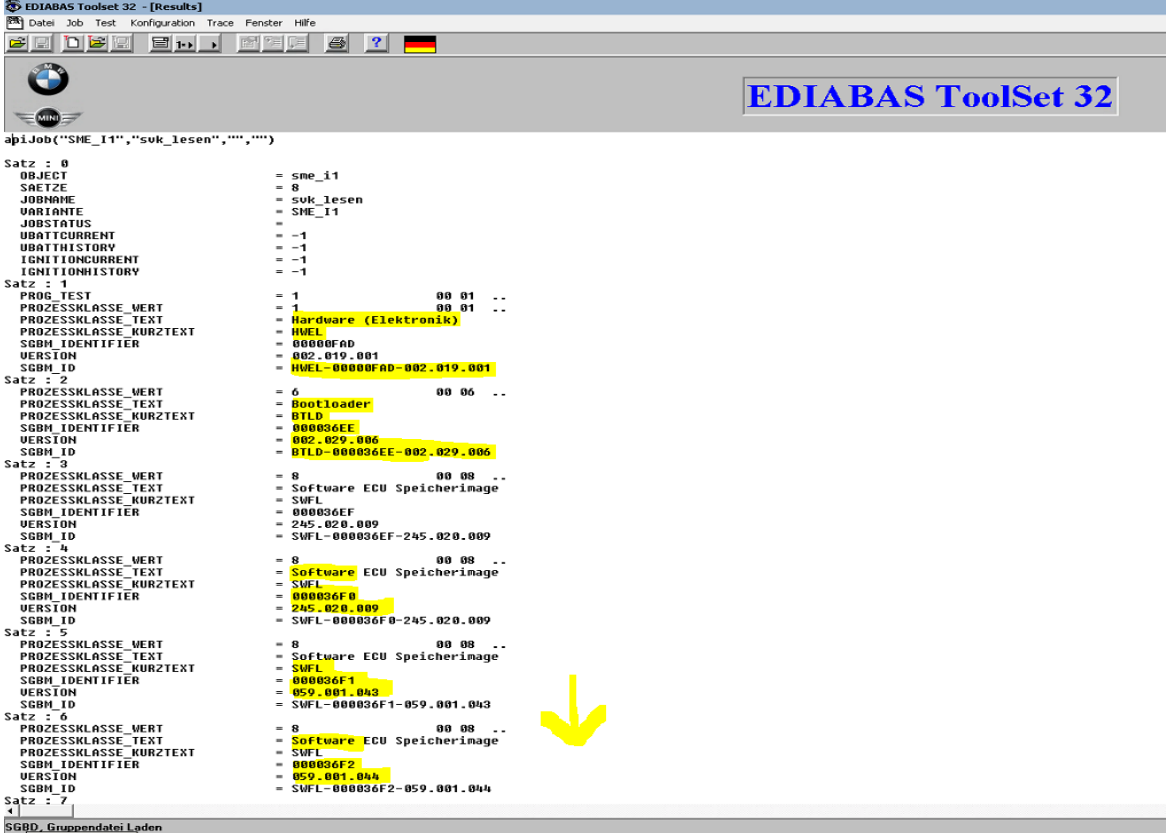
Steuergeräteverbaukennung (SVK Lesen)

3.1 SVK Lesen

Mit dem Job **svk_lesen** hat man die Möglichkeit sich über den Softwarestand, SGBM_ID und Bootloader zu informieren.

z.B. vergleich mit SW-Liste:

- *STSP*: TAL vergleichen
- *Automotive*: Excel Tabelle vergleichen
- HWEL 002.019.00x → Serien SG
→ kein SME-D-CAN & INCA Zugriff
- HWEL 002.012/002.013.00x →
APPL SG



```
apiJob("SME_I1","svk_lesen","", "")

Satz : 0
OBJECT = sme_i1
SAETZE = 8
JOBNAME = svk_lesen
VARIANTE = SME_I1
JOBSTATUS =
UBATTCURRENT = -1
UBATTISTORY = -1
IGNITIONCURRENT = -1
IGNITIONHISTORY = -1

Satz : 1
PROG_TEST = 1 00 01 ..
PROZESSKLASSE_WERT = 1 00 01 ..
PROZESSKLASSE_TEXT = Hardware (Elektronik)
PROZESSKLASSE_KURZTEXT = HWEL
SGBM_IDENTIFIER = 00000FAD
VERSION = 002.019.001
SGBM_ID = HWEL-00000FAD-002.019.001

Satz : 2
PROZESSKLASSE_WERT = 6 00 06 ..
PROZESSKLASSE_TEXT = Bootloader
PROZESSKLASSE_KURZTEXT = BTLD
SGBM_IDENTIFIER = 000036EE
VERSION = 002.029.006
SGBM_ID = BTLD-000036EE-002.029.006

Satz : 3
PROZESSKLASSE_WERT = 8 00 08 ..
PROZESSKLASSE_TEXT = Software ECU Speicherimage
PROZESSKLASSE_KURZTEXT = SWFL
SGBM_IDENTIFIER = 000036EF
VERSION = 245.020.009
SGBM_ID = SWFL-000036EF-245.020.009

Satz : 4
PROZESSKLASSE_WERT = 8 00 08 ..
PROZESSKLASSE_TEXT = Software ECU Speicherimage
PROZESSKLASSE_KURZTEXT = SWFL
SGBM_IDENTIFIER = 000036F0
VERSION = 245.020.009
SGBM_ID = SWFL-000036F0-245.020.009

Satz : 5
PROZESSKLASSE_WERT = 8 00 08 ..
PROZESSKLASSE_TEXT = Software ECU Speicherimage
PROZESSKLASSE_KURZTEXT = SWFL
SGBM_IDENTIFIER = 000036F1
VERSION = 059.001.003
SGBM_ID = SWFL-000036F1-059.001.003

Satz : 6
PROZESSKLASSE_WERT = 8 00 08 ..
PROZESSKLASSE_TEXT = Software ECU Speicherimage
PROZESSKLASSE_KURZTEXT = SWFL
SGBM_IDENTIFIER = 000036F2
VERSION = 059.001.004
SGBM_ID = SWFL-000036F2-059.001.004

Satz : 7
SGRD_Gruppendatei Laden
```


3.2 SVK vergleichen

- Mit dem Job **svk_lesen** hat man die Möglichkeit sich über den Softwarestand, SGBM_ID und Bootloader zu informieren.
- Dieser kann auch direkt mit der .xml-Datei (=TAL) im Softwarestand verglichen werden, um zu überprüfen ob die Bootloader kompatibel sind, oder ob der entsprechende SW-Stand schon geflasht wurde
- Die .xml-Datei liegt im Ordner: _svn_tm\SME_SW*SW-Stand*\PDX*.xml-Datei* und kann mit Doppelklick im Explorer geöffnet werden
- Immer Bootloader vor Flash checken!! -> meist kein „runterflashen“ möglich

4.

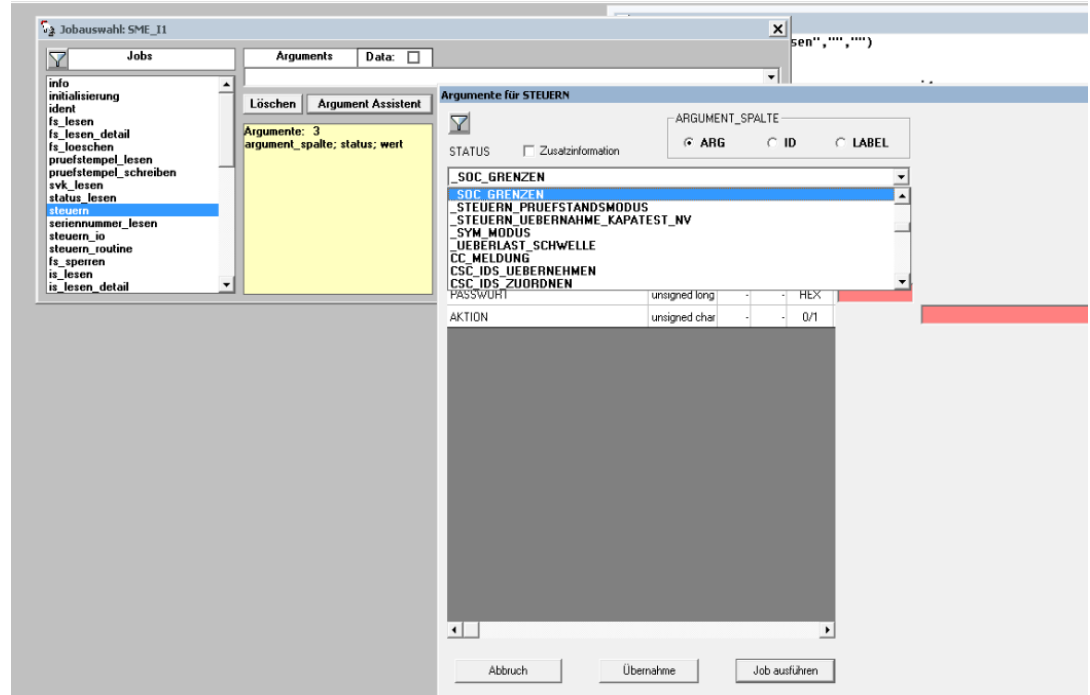
Steuern

4.1 Steuern I

Mit dem Job **steuern** kann man viele Manipulation durchführen.

Einige Argumente der Steuern-Jobs sind Passwortgeschützt, damit nicht jeder etwas ändern.

Manche Manipulationen können gefährlich sein z.B. am Prüfstand.

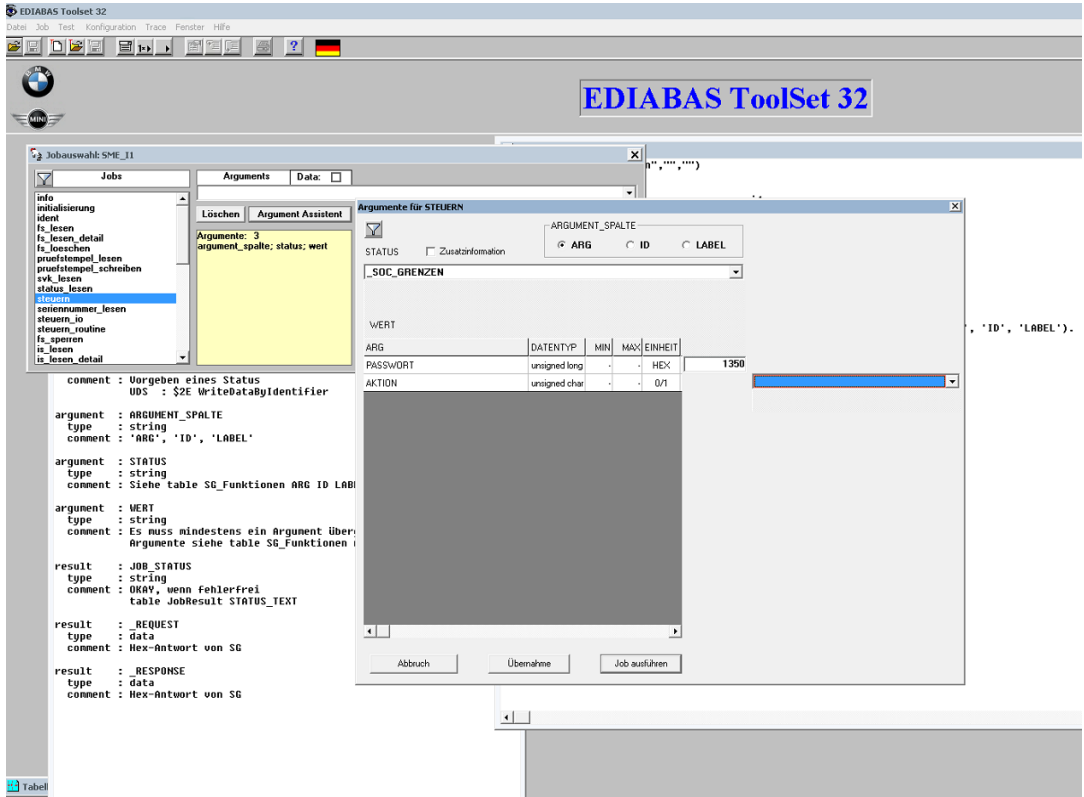


4.2 Steuern II

Steuern -> Argument Assistent ->
Steuern-Jobs wählen.

Für die SOC-Grenze zum Beispiel
ist das Passwort = **1350**

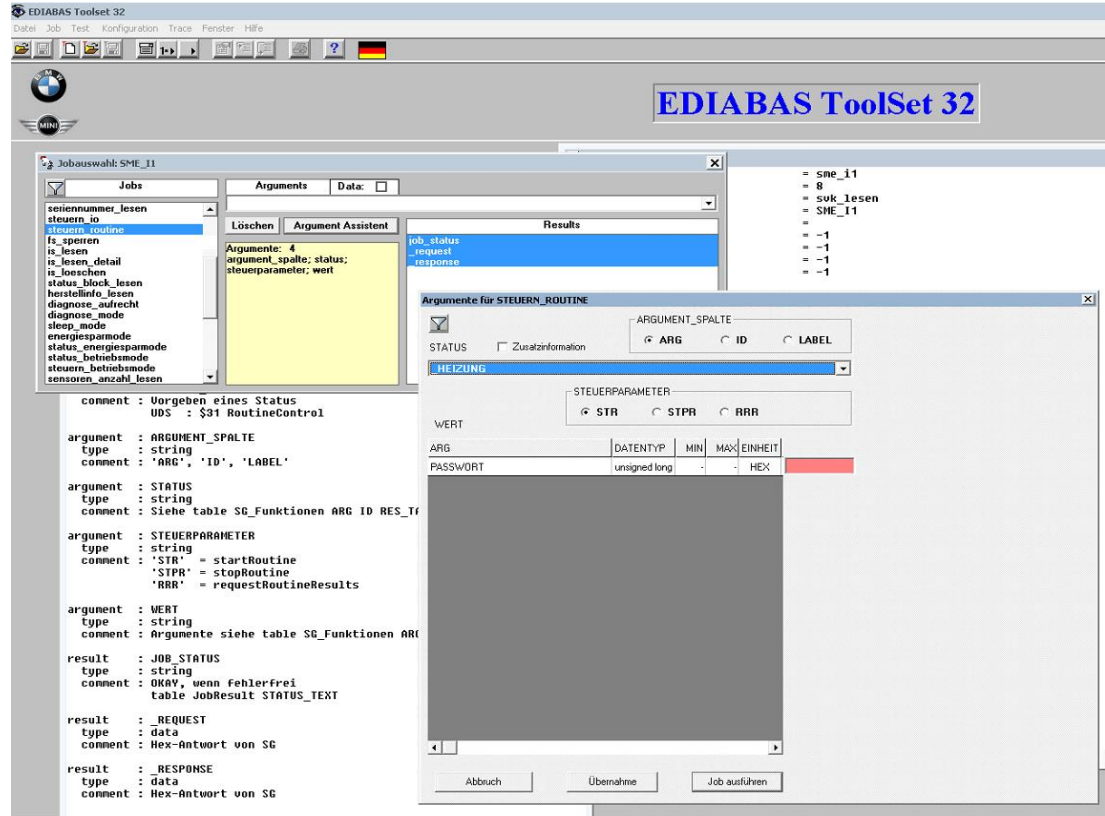
Danach kann die gewünschte
SOC-Grenzwert eingegeben
werden. Nachdem kann den Job
ausgeführt und übernommen
werden.



4.3 Steuern_Routine

Steuern_routine -> Argument
Assistent -> Argumente für
Steuern_Routine

Hier kann man eine Kombination
aus Steuern und status lesen
durchführen.
Es ist auch Passwortgeschützt.



5.

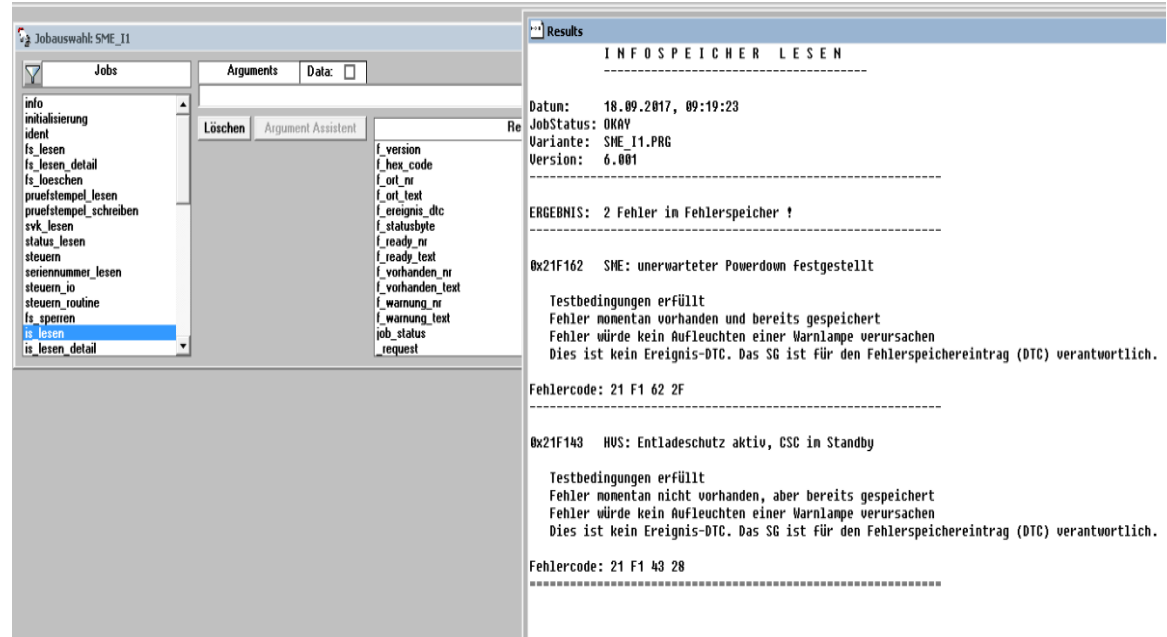
Info-Speicher

5.1 Info-Speicher

is_lesen gibt die Möglichkeit alle detaillierte Infos aus dem Speichern auszulesen.

Führt nicht zu einer KAT-Reaktion, nicht zur CCM, d.h. hat keinen Impact.

Bzw. **is_loeschen** bittet die Möglichkeit alle Infos aus dem Speichern zu löschen.



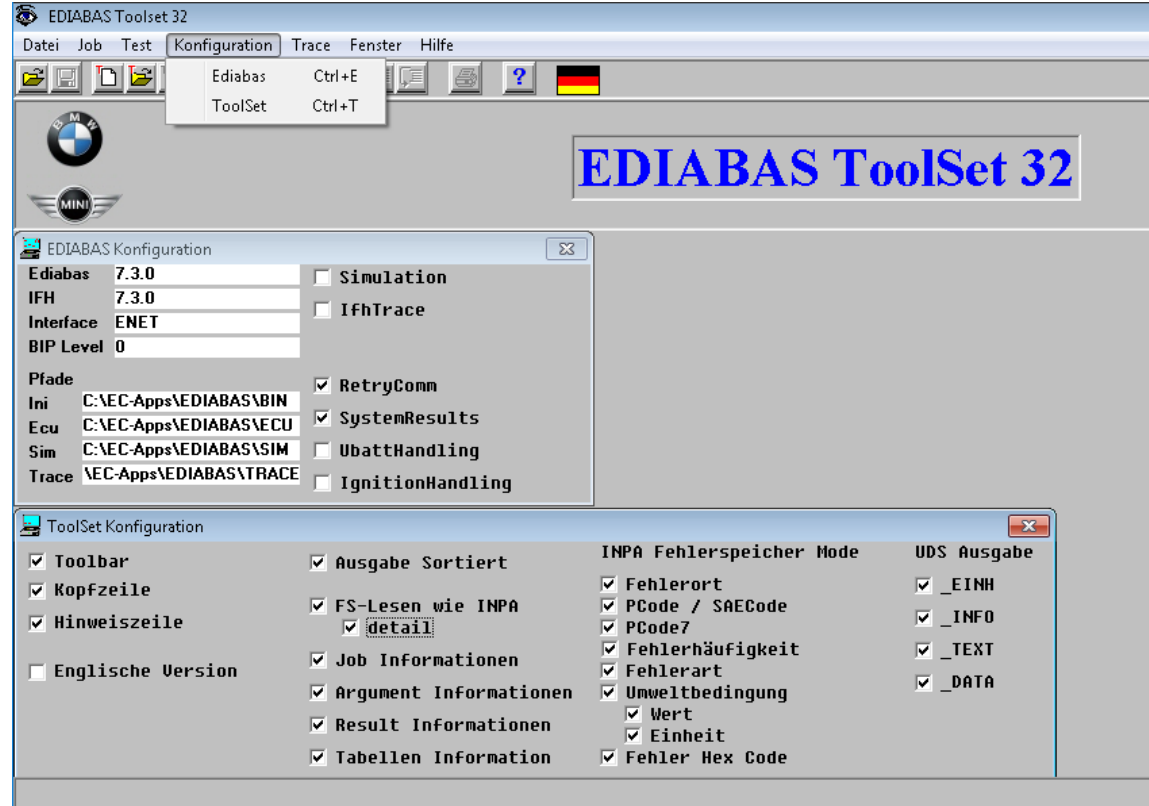
6.

Konfiguration

6.1 Toolset Konfiguration

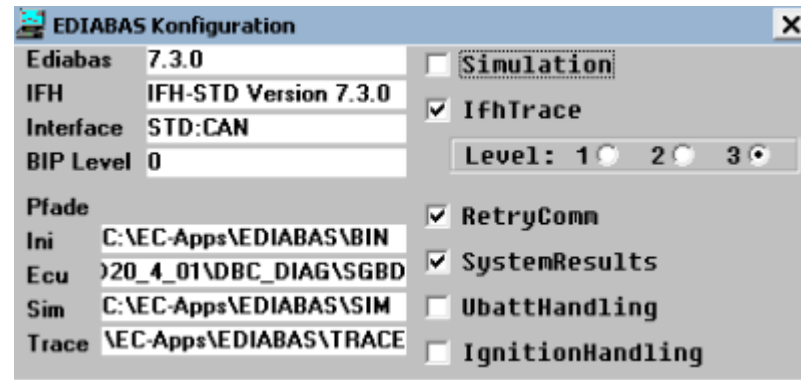
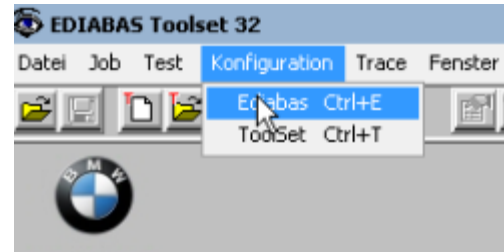
Konfiguration bietet verschiedene ToolSet-Configs an.

z.B. FS-lesen wie Inpa: Detail dadurch werden jeweils erweiterte Informationen wie Umweltdaten pro DTC zurück gegeben



6.2 EDIABAS Konfiguration

Im EDIABAS Konfiguration
befinden sich mehrere
Möglichkeiten



7.

Jobs

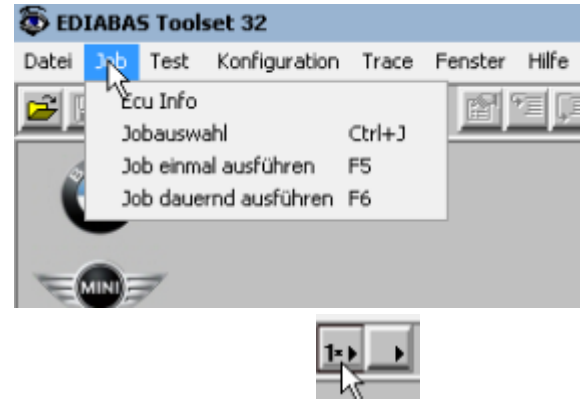
7.1 Jobs

Den Reiter „Job“ bietet verschiedene Möglichkeiten an:

Ecu Info

Jobauswahl

Job einmal oder dauernd ausführen



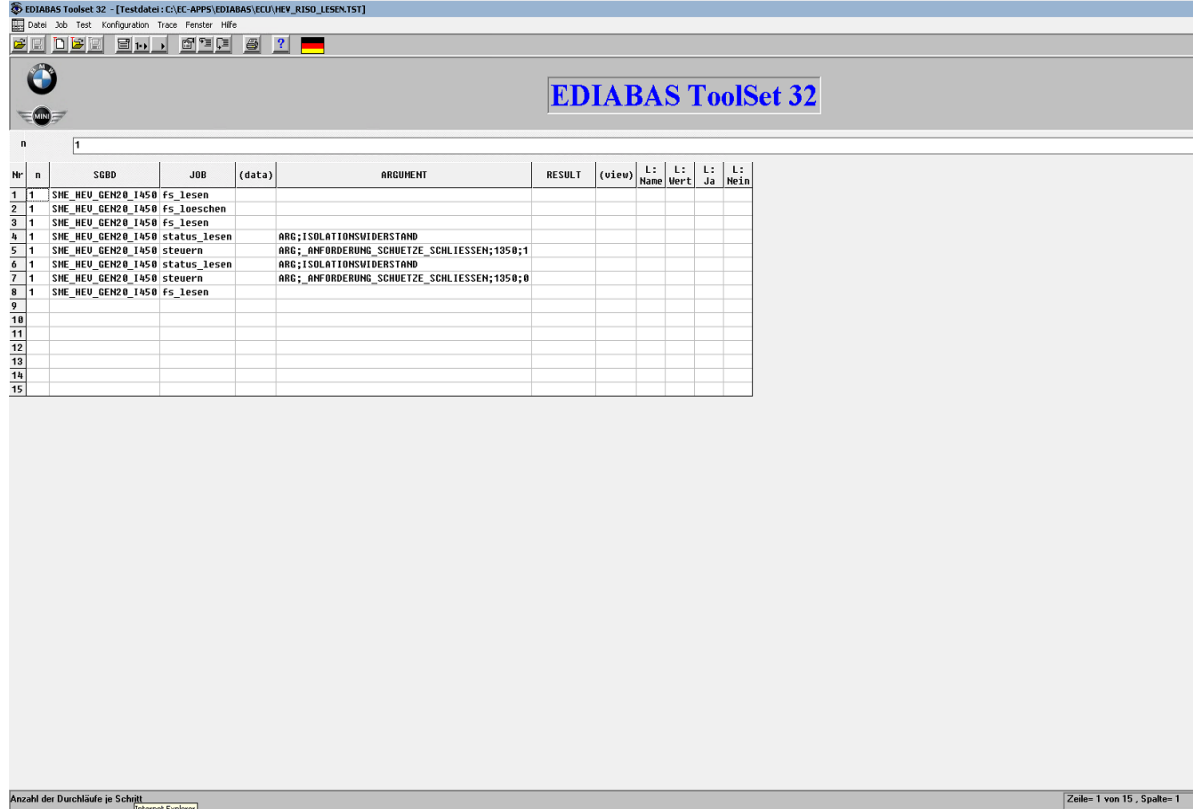
8.

Test erstellen

8.1 Test Erstellung

Datei → Test neu

Mit Ediabas kann man kleine Automatisierung durchführen.



The screenshot shows the EDIABAS ToolSet 32 application window. The title bar indicates the file path: C:\EC-APPS\EDIABAS\ECU\HEV_RISO_I1SEN.TST. The menu bar includes Datei, Job, Test, Konfiguration, Trace, Fenster, and Hilfe. The toolbar contains icons for file operations and execution. The main area displays a table with test configurations. The table has columns for Nr, n, SGBD, JOB, (data), ARGUMENT, RESULT, (view), L: Name, L: Wert, L: Ja, and L: Nein. The table contains 15 rows of test data, including commands like fs_lesen, status_lesen, and steuern, with associated arguments and results.

Nr	n	SGBD	JOB	(data)	ARGUMENT	RESULT	(view)	L: Name	L: Wert	L: Ja	L: Nein
1	1	SHE_HEU_GEN20_1450	fs_lesen								
2	1	SHE_HEU_GEN20_1450	fs_loeschen								
3	1	SHE_HEU_GEN20_1450	fs_lesen								
4	1	SHE_HEU_GEN20_1450	status_lesen		ARG:ISOLATIONSWIDERSTAND						
5	1	SHE_HEU_GEN20_1450	steuern		ARG:_ANFORDERUNG_SCHUETZE_SCHLIESSEN;1950;1						
6	1	SHE_HEU_GEN20_1450	status_lesen		ARG:ISOLATIONSWIDERSTAND						
7	1	SHE_HEU_GEN20_1450	steuern		ARG:_ANFORDERUNG_SCHUETZE_SCHLIESSEN;1950;0						
8	1	SHE_HEU_GEN20_1450	fs_lesen								
9											
10											
11											
12											
13											
14											
15											

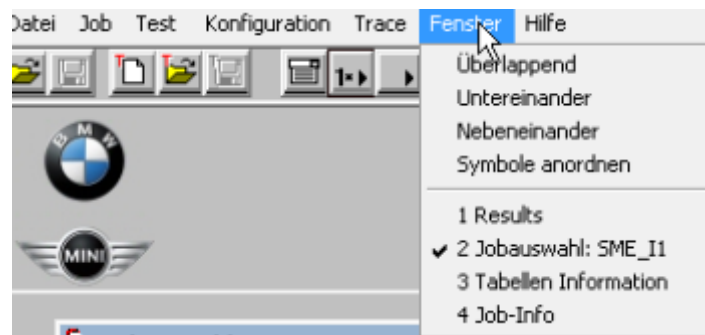
Anzahl der Durchläufe je Schritt: 1
Zellen: 1 von 15, Spalten: 1

9.

Fenster

9.1 Fenster

Fenster „Formatierung“ sowie geöffnet Fenster Bearbeitung möglich.

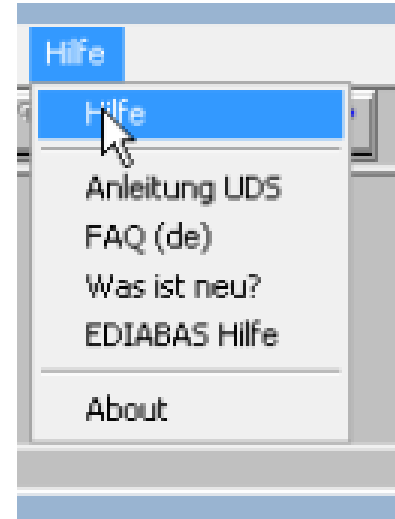


10.

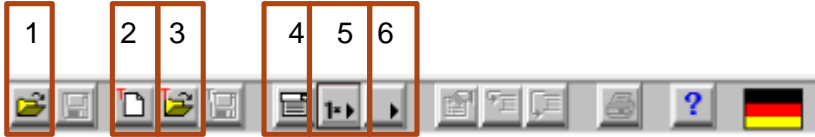
Hilfe & Shortcuts

10.1 Hilfe

In der Hilfe Reite kann man über Anleitung UDS, FAQ, Software Stand, etc...sich informieren



10.2 Allgemein Shortcuts



Allgemein Shortcuts Reihe:

1. SGBD, Gruppendatei Laden
2. Neuer Test
3. Test Laden
4. Jobauswahl
5. Job einmal ausführen (F5)
6. Job dauernd ausführen (F6)

Steuern oder lesen Jobs kann man einmal ausführen F5 oder als Schleife mit F6.

Übung

Aufgabenschritte:

- a. Svk lesen
- b. Fs lesen
- c. Fs löschen
- d. status_lesen (kontinuierliches Auslesen)
- e. steuern
- f. steuern_routine
- g. Testdatei anlegen (*.tst)