ETAS ODX-LINK V7.5

User Guide

www.etas.com

Copyright

ODX-LINK V7.5 I User Guide R02 EN - 06.2024

Content

1 Introduction

This manual contains a description of the INCA add-on ODX-LINK for ODX-based ECU diagnostics.

1.1 Intended Use

INCA and INCA add-ons are developed and approved for automotive applications and procedures as described in the user documentation for INCA and INCA add-ons.

ODX-Link adds diagnostic functionality to INCA and makes diagnostic data available as measurement signals in INCA. ODX-Link supports various "Protocols " on page 12.

INCA and the INCA add-ons are intended to be used in industrial labs and in test vehicles.

ETAS GmbH cannot be made liable for damage that is caused by incorrect use and not adhering to the "Safety Information" on the next page.

1.2 Target Group

This software product and this user guide address qualified personnel working in the fields of automotive ECU development and calibration, as well as system administrators and users with administrator privileges who install, maintain, or uninstall software. Specialized knowledge in the areas of measurement and ECU technology is required.

1.3 Classification of Safety Messages

Safety messages warn of dangers that can lead to personal injury or damage to property:

⚠ DANGER

DANGER indicates a hazardous situation that, if not avoided, will result in death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation that, if not avoided, could result in death or serious injury.

⚠ **CAUTION**

CAUTION indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

NOTICE

NOTICE indicates a situation that, if not avoided, could result in damage to property.

1.4 Safety Information

Observe the following safety information when working with INCA and INCA add-ons:

⚠ **WARNING**

Risk of unexpected vehicle behavior

Calibration activities influence the behavior of the ECU and the systems that are connected to the ECU.

This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.

Only perform calibration activities if you are trained in using the product and can assess the possible reactions of the connected systems.

⚠ **WARNING**

Risk of unexpected vehicle behavior

Sending messages via bus systems, such as CAN, LIN, FlexRay, or Ethernet, influences the behavior of the systems connected to it.

This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.

Only perform the sending of messages via a bus system if you have sufficient knowledge in using the respective bus system and can assess the possible reactions of the connected systems.

⚠ WARNING

Risk of unexpected vehicle behavior

Invoking diagnostic functions or sending diagnostic messages affects the behavior of the ECU and the systems connected to it.

This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.

Only invoke diagnostic functions or send diagnostic messages if you have sufficient knowledge in using diagnostic functions.

⚠ WARNING

Risk of unexpected vehicle behavior

ODX consistency errors in the ODX files can result in INCA runtime errors.

This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.

Do not use the respective ODX files if relevant diagnostic functions are affected by inconsistencies.

Adhere to the instructions in the ETAS Safety Advice and the safety information given in the online help and user guides.

Open the ETAS Safety Advice in the INCA help menu **?** > **Safety Advice**.

1.5 Data Protection

If the product contains functions that process personal data, legal requirements of data protection and data privacy laws shall be complied with by the customer. As the data controller, the customer usually designs subsequent processing. Therefore, he must check if the protective measures are sufficient.

1.6 Data and Information Security

To securely handle data in the context of this product, see the INCA Help section “Data and Information Security”.

2 About ODX-LINK

This chapter contains a general introduction to ECUs and the ODX standard. The characteristics of ODX-LINK are also described.

2.1 Tasks of an ECU

Nowadays a modern car without electronic components is unthinkable. Every unit of a motor vehicle contains electronics: from the lights through electrically propelled starters to completely modern components like ABS and ESP.

Today's vehicles have a large number of ECUs – virtually every electronic function has its own ECU or uses an ECU. Every ECU executes different tasks. These functions can be divided into three categories:

- Controlling tasks
- Communication with other ECUs
- Diagnostics

Controlling Tasks

Every ECU has its own specific function area, for example gearbox, engine, or door ECU. The data the ECUs need to fulfil their tasks is received from a large number of analog and digital sensors. The ECUs receive additional data via the vehicle's bus systems (onboard communication). For example, the gearbox sends information about the gear currently selected to the engine ECU so that the latter can calculate the correct ignition time.

In addition to the actual acquisition of data (sensor system), actuating elements play an extremely important role. Target values calculated by the ECU are converted to physical values (voltage, pressure etc.) using various actuators.

Communication Tasks

Normally an ECU requires information from other ECUs to carry out its controlling tasks. The resulting communication with other ECUs is referred to as onboard communication. This involves an ECU sending information on the bus system and all other ECUs checking to see whether they need this information.

Diagnostic Tasks

The third category of task of an ECU, after the controlling tasks and communication, is diagnostics. The increased volume of electronic components inside the vehicle is providing more and more diagnostic options. Communication between ECUs and diagnostic devices is referred to as offboard as the diagnostic system is not part of the vehicle. The diagnostic system provides developers and technicians with a lot of failure-specific and vehicle information which helps them to solve vehicle problems and optimize vehicle performance.

2.1.1 Diagnostics

Development engineers and technicians use the diagnostic functions to exchange data with the ECU. This makes it easier to detect faults and to optimize vehicle performance. When a fault occurs, for example wrong ignition timing, all relevant parameters are stored in the fault memory of the ECU. The technician reads this information using the diagnostic functions to find the cause of the fault.

Developers can also use the diagnostic functions to acquire up-to-date information on the behavior of the ECU under normal operating conditions.

Diagnostic functions can also be used during production. Internal test results can be checked or the serial number of the ECU is read to guarantee a faultless device.

2.2 The ODX Standard

The ODX standard (ASAM MCD 2D) standardizes the formal description of information in the vehicle and ECU diagnostic sector – ODX stands for "Open Diagnostic Data Exchange".

A vehicle-external repair shop tester is connected to the network of these ECUs via a special data interface, called the diagnostic interface. The tester exchanges information with the ECUs and uses message-oriented protocols for this purpose. These protocols are usually standardized (KWP2000 in acc. with ISO14230, UDS in acc. with ISO14229 etc.).

The ODX specification contains the data model for describing all diagnostic data of a vehicle/ECU. ODX is described with UML diagrams (Unified Modelling Language) – the format for data exchange is XML (eXtensible Markup Language).

The ODX specification enables the following:

- The transfer of diagnostic data and program data between system suppliers, vehicle manufacturers and repair shops in accordance with the single-source principle.
- The communication of an offboard tester with the ECUs and the subsequent interpretation of the data contained in the messages without test equipment having to be specifically programmed for this purpose.

The following figure shows an overview of the ISO standards and ASAM specifications for MCD systems.

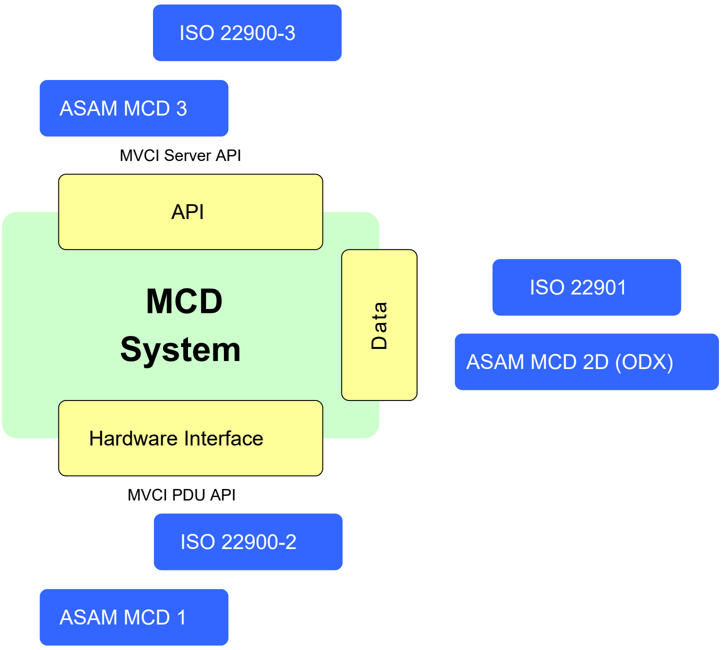


Fig. 2-1: ISO Standards and ASAM Specifications

Using the standardization

- of the API to the vehicle hardware interface (ISO 22900-2: D-PDU API),
- the diagnostic data model (ISO 22901-1 ODX) and
- the interface between the runtime system and the test application (ISO/CD 22900-3: D-Server API),

it is possible, depending on the application, to combine the best hardware with the desired runtime system and the most suitable application.

2.3 ODX-LINK

ODX-LINK adds diagnostic functionality to INCA – once ODX-LINK has been installed, there are additional dialog boxes available for ECU diagnosis in the INCA Experiment Environment. ODX-LINK makes it possible for you to access the fault memory, for example, during calibration and display the diagnostic information in plain text.

ODX-LINK processes diagnostic data using the ODX description – the user can select and run every diagnostic service described in ODX.

ODX-LINK determines all information necessary for sending the service, sends this to the ECU using the connected hardware, receives the response of the ECU with the same hardware and then decodes the response for the user.

ODX Standard

ODX-LINK supports ODX files (including PDX = Packaged ODX) and the binary databases created with DTS-Venice which correspond to the "ASAM MCD 2D (ODX) V2.0.1 and V2.2" specification. This thus guarantees the use of single-source databases throughout the entire development cycle – new implementation of diagnostic information, which is both expensive and prone to errors, is thus no longer necessary.

Protocols

ODX-LINK is available as an add-on to INCA – together with INCA, ODX-LINK supports the following protocols for the serial ECU interfaces:

- UDSonCAN (ISO 14229-1 / ISO 15765-3 on ISO 15765-2)
- KWPonCAN (ISO 14230-3 on ISO 15765-2)
- OBDonCAN (SAE J1979 / ISO 15031-5 on ISO 15765-4)
- OBDonUDS (SAE J1979-2 on ISO 15765-4)
- ZEVonUDS (J1979-3 on ISO 15765-4)
- DoIP (ISO 14229-5 on ISO 13400-2)

Hardware Support

Amongst others[1)], ODX-LINK supports the following ETAS measure and calibration hardware:

- ES59x Interface Modules
- ES58x
- ES8xx
- ES910 Rapid Prototyping Module / ES921 CAN Module

By using ETAS hardware, measure, calibration and diagnostic access to the ECU are possible both from a user interface and via a hardware interface, e.g. UDS on CAN. This means software and hardware costs can be saved and valuable time gained to spend on development.

ODX-LINK and INCA

But ODX-LINK offers a great deal more than the special dialog boxes for querying diagnostic data: The diagnostic data that can be queried by the ECU (via the diagnostic interface) can be used as normal measurement signals in INCA. This means that all INCA functions that are available for measurement signals can also be used for diagnostic data:

- configuration of measure windows with diagnostic signals in the experiment via the variable selection
- definition of trigger conditions based on diagnostic signals
- definition of calculated signals
- recording of diagnostic data in INCA measure files

[1)] Generally, ODX-LINK supports all INCA hardware with a CAN port – the following is simply a list of a few examples.

Diagnostic data can thus be measured together with standard INCA measure data acquired address-based via measure interfaces such as, for example, ETK, CCP and XCP, and recorded and analyzed in a common measure file. Among other things, this makes it possible to validate and evaluate diagnostic data more precisely and efficiently than before.

2.4 Working with ODX Projects

The ODX project is part of the INCA workspace – the ODX project and the logical link are assigned in the Hardware Configuration Editor and saved with the workspace.

This standardizes the operation of ODX-LINK and experiments of one workspace can use different ODX-LINK windows and settings.

Quick Access to Example Data Bases

You can import example data bases via `*.exp` files from the following folder:

`..\ETASData\INCA7.5\ODX`

Quick Access to ODX Project Files

You can find `*.pdx` files according to different versions of the OBD SAEJ1979 standard in the following folder:

`..\ETASData\ODX7.5\Projects`

2.4.1 Procedure

This section takes a brief look at the procedure in INCA – for more details, refer to the tutorial:

"ODX-LINK Tutorial" on page 72.

- Create a **main directory** and **workspace.**
- Add an **ECU project** if necessary.

  This step can be skipped (see "Diagnostics without an A2L File" on the next page).
- Read in the **ODX project.**

  For ODX projects, you can use the following file formats: `*.pdx`, `*.odx-*`, `*.xprj`, `*.prj`.

  With ODX V2.2, a consistency check is executed after the import of the project. The consistency check report of the last imported ODX V2.2 project can be found in \ETAS\LogFiles\ODX.

  For a ODX V2.0.1 project, no consistency check is performed. Therefore, an imported ODX project of the INCA database can have three different consistency states. When you click on the project, the state is displayed:

- consistent: ODX projects without consistency errors
- not consistent: ODX projects with consistency errors
- consistency state unknown: for ODX V2.0.1 projects that have not been checked

⚠ WARNING

Risk of unexpected vehicle behavior

ODX consistency errors in the ODX files can result in INCA runtime errors.

This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.

Do not use the respective ODX files if relevant diagnostic functions are affected by inconsistencies.

- Add a **hardware configuration** (see "Automatic Search for and Configuration of OBDonCAN Devices and OBDonUDS Devices" on the next page).

  In the Hardware Configuration Editor: **Device** > **Add**.
- **ODX configuration**

  In the Hardware Configuration Editor: **Hardware** > **Configure ODX**. Select the ODX project.

  Logical link mapping (assignment of a logical link of the ODX project to the INCA device).
- Add **experiment** and assign **workspace**.

2.4.2 Diagnostics without an A2L File

Communication with the connected ECUs is established during hardware -initialization. The communication parameters necessary for this can be defined in A2L or ODX files and must correspond to the ECUs.

A different set of parameters is also required for each protocol (KWP2000, UDS, CCP, XCP, etc.). A2L files normally contain parameters for a specific protocol and a specific bus system and cannot be used for other protocols and buses.

In earlier versions of ODX-LINK, KWP2000 and UDS devices could only be used if they were assigned an A2L file with the corresponding communication parameters. If there was no such file, a dummy file had to be created containing these KWP2000 or UDS parameters.

ODX-LINK can also be used without an ECU project (in the form of an A2L file). The communication parameters for UDS or KWP2000 devices are then determined from the assigned logical link of the ODX project.

Note

The ODX communication parameters must comply with the ISO 22900-1 specification for ODX V2.0.1 or V2.2 (as far as parameter names, values, units, etc. are concerned)!

An ECU project can still be assigned to UDS or KWP2000 devices – in this case, the A2L file is the "master" and ODX parameters are ignored.

If you use a UDS or KWP2000 device in the hardware configuration, you can simply skip assigning an A2L file – after assigning an ODX project and logical link for the device, the communication parameters are read out of ODX and used during hardware initialization.

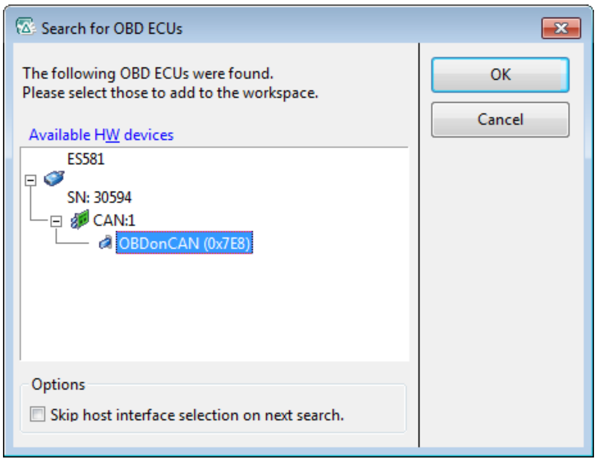
Note

If hardware initialization via ODX communication parameters does not work, you have either assigned the wrong logical link or the ODX data does not correspond to the ECU.

2.4.3 Automatic Search for and Configuration of OBDonCAN Devices and OBDonUDS Devices

In ODX-LINK, it is possible to search for all connected devices that support OBDonCAN or OBDonUDS. To do so, select **Hardware** > **Search For OBD ECUs** in the Hardware Configuration Editor.

An OBDonCAN or an OBDonUDS device with the correct OBD parameters (baud rate and CAN-ID) is automatically added to the hardware configuration for every ECU found.



Search for OBD ECUs

The following OBD ECUs were found.
Please select those to add to the workspace.

Available HW devices

- ES581 SN: 30594
  - CAN:1
    - OBDonCAN (0x7E8)

Options

□ Skip host interface selection on next search.

For more details, refer to the corresponding tutorial chapters in the ODX-LINK User Guide:

- "Using OBDonCAN (SAE J1979) with ODX-LINK" on page 87
- "Using OBDonUDS (SAE J1979-2) with ODX-LINK" on page 92

3 Installation

This chapter contains tips on installing the add-on ODX-LINK.

3.1 System Requirements

The hardware and software requirements for working with INCA V7.5 (or higher) are also sufficient for the add-on ODX-LINK – they are described in the manual "INCA V7.5 - Installation Guide".

3.2 Installing

This section describes the installation of ODX-LINK.

Certain system requirements according to the ETAS INCA 7.5 Installation Guide must be met to install the product. Make sure that these system requirements are met before starting the installation.

To download the INCA installation package

1. On the ETAS homepage, click **Download Center**.
2. Search for **INCA** > **INCA V7.5** > **Software**.
3. Download the Installation Package (*.zip).
4. In the Windows File Explorer, select the downloaded ZIP file, right-click and select **Properties**.
5. On the **General** tab, at the Security option disable the **Unblock** button.
6. Extract the complete structure of the ZIP file.

> **Note**
>
> The fully qualified file name of all components of the setup and the directory name are subjects of a restriction and must fall below a certain character length. The character length is calculated individually.
>
> The path length of the folder where you save the `Setup_ServicePack.exe` shall not exceed 80 characters.
>
> Do not change the folder structure, the folder-names, or any name of installation executables of the installation package.

To install the software

1. Close all open ETAS software.

   > **Note**
   >
   > Do not execute other software updates in parallel, for example an update of the operating system. Wait until all updates are installed and restart the computer before starting the installation.

2. Execute the `Setup_ServicePack.exe` file from your installation directory.

   The "Service Pack Installer" window opens.

3. In the **Install** column, enable your desired software products and add-ons.

   To select all, enable the corresponding package on top level.

4. Read the license agreement and enable **I read and accept the end user license agreement** option.

5. Select your preferred setup language.

   > **Note**
   >
   > The language you select changes the user interface language for the Service Pack Installer and the language for newly installed and for all already installed INCA and INCA add-on products.

6. Click on **Install**.

   The installation process is initiated.

7. Click on **Restart Options**.

   The "Restart" dialog box opens.

   > **Note**
   >
   > It is recommended to restart your system after the installation is fnished.
   >
   > Some installations require a restart in between the installation routine. In this case, a warning sign is displayed in the Status column. After a restart, the installation will continue automatically.

8. Select the desired restart options.

9. Click **Ok**.

All your desired software products and add-ons will be installed in silent mode. This means no additional dialog windows are displayed during installation.

3.3 Licensing

A valid license is required to use the software. You can obtain a license in one of the following ways:

- from your tool coordinator
- via the self-service portal on the ETAS website at www.etas.com/support/licensing
- via the ETAS License Manager

To activate the license, you must enter the Activation ID that you received from ETAS during the ordering process.

For more information about ETAS license management, see the ETAS License Management FAQ or the ETAS License Manager help.

To open the ETAS License Manager help

The ETAS License Manager is available on your computer after the installation of any ETAS software.

1. From the Windows Start menu, select **E** > **ETAS** > **ETAS License Manager**.

   The ETAS License Manager opens.

2. Click in the ETAS License Manager window and press F1.

   The ETAS License Manager help opens.

4 ODX-LINK Menus and Functions

If you have installed ODX-LINK on your system, the **ODX** menu is displayed in the INCA menu bar of the experiment window.

You can use this menu to

- work with the GUIs to run ODX functions (see "User Views" below)
- define how the snapshot data is saved (see "Data Logging Configuration " on page 57)
- configure and trigger snapshot functionality (see "Snapshots" on page 60)

This chapter also contains information on the subject "Diagnostic -Signals in The INCA Variable Selection" (see section "Diagnostic Signals in the INCA Variable Selection" on page 63).

4.1 User Views

Generally ODX-LINK can open all ODX projects which comply with the ODX standard in Version 2.0.1 and 2.2.

These projects describe the diagnostic services, with their parameters, which can be queried via the ODX-LINK user views. In addition, the ODX project contains decoding information for responses from the ECU, i.e. information about how the data from the ECU is to be interpreted.

The ODX-LINK user views output this decoded information (or parts thereof) in the windows.

With the help of several ODX-LINK user views (e.g. the "Diagnostic Services" window), the user can select and parameterize any service – once the service request has been sent, the complete, decoded response is shown in the user view. In this case, the user must have detailed knowledge of the different services and of the data transported.

Other ODX-LINK user views, on the other hand, have been developed for very special requirements, such as the display of data for Service Inspector ("Service Inspector" window). In these cases, the user gets the required result at the simple push of a button.

The ODX-LINK user views do not depend on whether INCA measuring is taking place or not, i.e.:

- even if there is no current INCA measuring, the windows for querying diagnostic data can be used,
- the display of data in the ODX-LINK user views is not controlled via the starting or stopping of INCA measuring but exclusively via manual user operation (using the relevant **Read ...** buttons),
- the displayed diagnostic data of the ODX-LINK windows is not recorded in an INCA measure file. To record diagnostic data, corresponding diagnostic signals must be configured for INCA measuring in the INCA variable selection.

Use **ODX** > **User views** to open the dialog windows for running your diagnostic tasks. The menu contains the following items:

- "Diagnostic Services" on page 22
- Service Inspector
- "Hex Service" on page 28
- "OBD" on page 29

ⓘ **Note**

The configuration of most ODX-LINK user views (apart from "Diagnostic Services" and "Hex Services") must correspond to the ODX diagnostic database read in.
ODX-LINK makes ODX configurations (ODX databases and the corresponding default window configurations) available to available standard diagnostic protocols (OBDonCAN and OBDonUDS) in the form of INCA export files (to be found in `ETASData\INCA7.5\export\ODX`).
Please note, however, that these ODX configurations may not actually suit all diagnostic services of the ECU used, as the diagnostic services of many ECUs deviate from the relevant standard diagnostic protocol. ECU-specific fault memory entries and environment data in particular are thus not part of these sample configurations.

The "OBD" User View

Unlike other diagnostic protocols, the OBD standard defines which services there are, how they are to be parameterized and which responses they supply.

This is why it is possible in the application case "OBD" to use a specific ODX project and preconfigure the "OBD" user view entirely for this project.

When the "OBD" user view is opened, a check is made to see whether all the necessary services and parameters etc. exist in the ODX project currently being used by ODX-LINK. If this is not the case, an error message is issued. In this case, the "OBD" user view is opened but cannot be used with the current database.

ⓘ **Note**

The "OBD" User View can only be used with the ETAS OBD-ODX database `ETASData\ODX7.5\Projects\OBDonCAN_ETAS_SAEJ1979<Version>.pdx` and `OBDonUDS_ETAS_SAEJ1979<Version>.pdx`. The database is also contained in the INCA export file:
`ETASData\INCA7.5\export\ODX\OBDonCan.exp64` and `ETASData\INCA7.4\Export\ODX\ODXTestDevice_OBDonUDS.exp64`.

Default Configurations

The configuration of a user view can be saved as a default setting by selecting the **Save as default** button in the relevant window.

When you have made a configuration for the first time, you are prompted to specify whether you want to save these changes as the default setting when you close the user view. These settings are saved as part of the ODX configuration and are valid for all GUIs of one type.

This guarantees that ODX project-specific settings in the relevant GUIs do not have to be started from scratch every time.

Snapshots

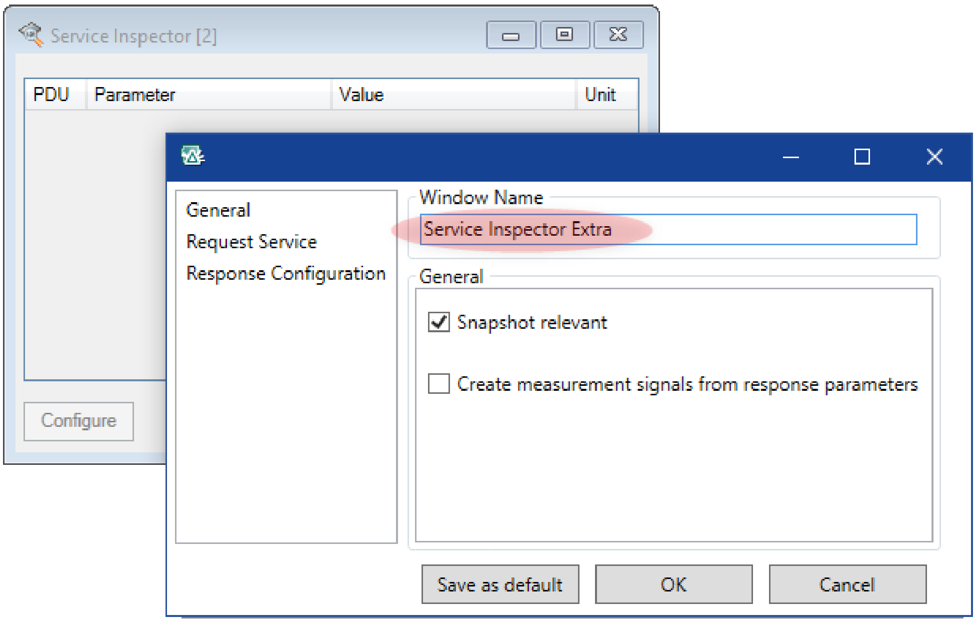
All current results of the service requests which you define with user views, can be saved via a Snapshot function. You decide for each individual user view whether its result should be included in the snapshot file. The relevant information can be found in the configuration settings of the individual user views.

The snapshot icon – a small camera – at the very left of the title bar of each user view window shows that the results of this user view are recorded in the snapshot. For more details on the snapshot function, refer to the section "Snapshots" on page 60.

Names of User Views

If several user views of the same type but with different configurations are used in an experiment, they can be given different names (with the exception of "Hex Service").

To do this, select **Configure** and enter the desired name in the "Window Name"box.



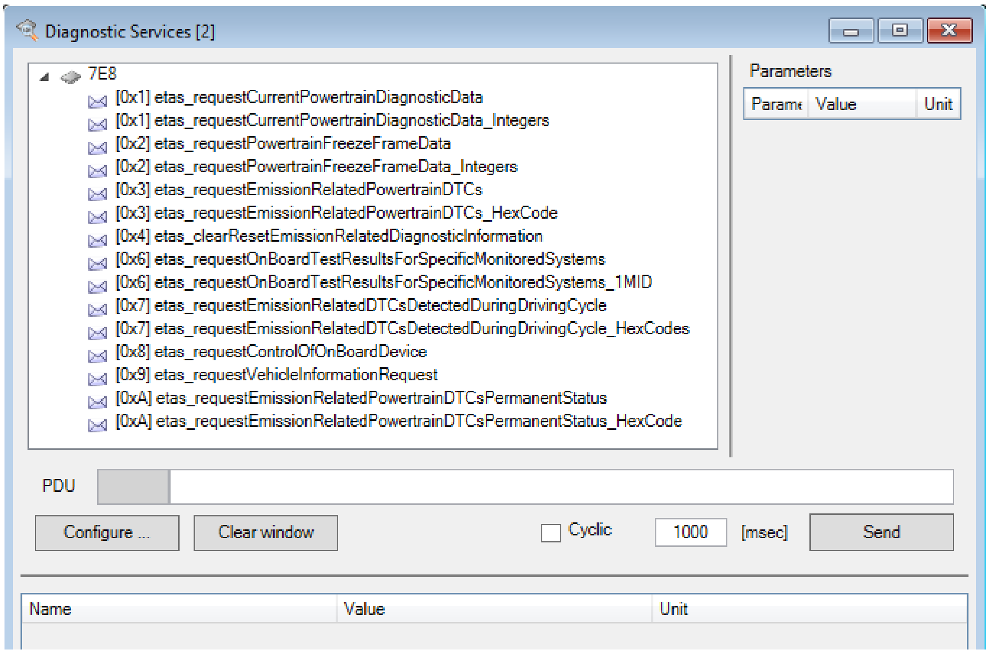
Window Name: Service Inspector Extra

☑ Snapshot relevant

□ Create measurement signals from response parameters

Showing the Services in Function Classes

The services can either be displayed as a list (see left of the figure) or in functional classes (shown by a folder) in a user view window (e.g. Diagnostic Services):



Diagnostic Services [2]

7E8

- [0x1] etas_requestCurrentPowertrainDiagnosticData
- [0x1] etas_requestCurrentPowertrainDiagnosticData_Integers
- [0x2] etas_requestPowertrainFreezeFrameData
- [0x2] etas_requestPowertrainFreezeFrameData_Integers
- [0x3] etas_requestEmissionRelatedPowertrainDTCs
- [0x3] etas_requestEmissionRelatedPowertrainDTCs_HexCode
- [0x4] etas_clearResetEmissionRelatedDiagnosticInformation
- [0x6] etas_requestOnBoardTestResultsForSpecificMonitoredSystems
- [0x6] etas_requestOnBoardTestResultsForSpecificMonitoredSystems_1MID
- [0x7] etas_requestEmissionRelatedDTCsDetectedDuringDrivingCycle
- [0x7] etas_requestEmissionRelatedDTCsDetectedDuringDrivingCycle_HexCodes
- [0x8] etas_requestControlOfOnBoardDevice
- [0x9] etas_requestVehicleInformationRequest
- [0xA] etas_requestEmissionRelatedPowertrainDTCsPermanentStatus
- [0xA] etas_requestEmissionRelatedPowertrainDTCsPermanentStatus_HexCode

Parameters

Parame	Value	Unit

PDU

□ Cyclic 1000 [msec]

Name	Value	Unit

To configure service display

1. Select **Options** > **Users Options** > **Open** from the INCA main menu.

   The window for setting user options opens.
2. Select the "ODX" tab.
3. Select "Yes" or "No" with the option "Show Functional Classes".

A service is assigned to a functional group by the author of the ODX project data.

4.1.1 Diagnostic Services

The **Diagnostic Services** function allows you to send a service request to the ECU. The service request must be defined in the diagnostic database. The service request and its result can be shown both in clear text and in hexadecimal notation. All clear text must be defined in the diagnostic database. The display of the service request and the results can be configured.

The request can be executed either just once or recur periodically. The recurrence rate can be configured.

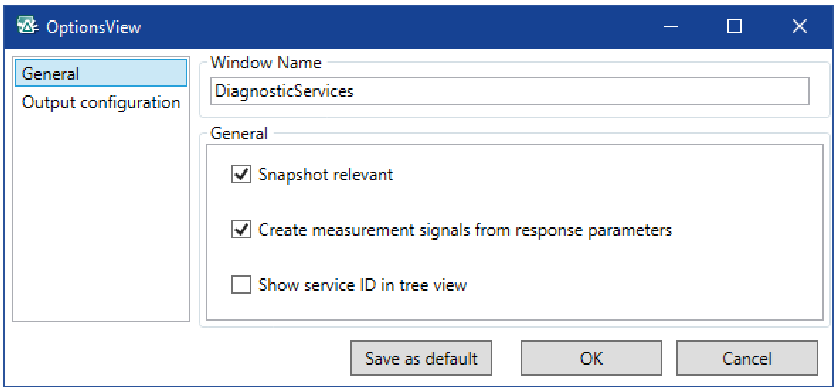
If the cyclical send timeframe you defined cannot be adhered to, for example because the bandwidth of the interface is not sufficient, it is adjusted automatically. In this case, the adjusted cycle time is shown with a red background.

To configure a diagnostic service

1. Select **ODX** > **User views** > **Diagnostic Services**.

   The dialog box for selecting a service is displayed.
2. Click **Configure**..

3. Select "General" in the left-hand window.



Window Name

DiagnosticServices

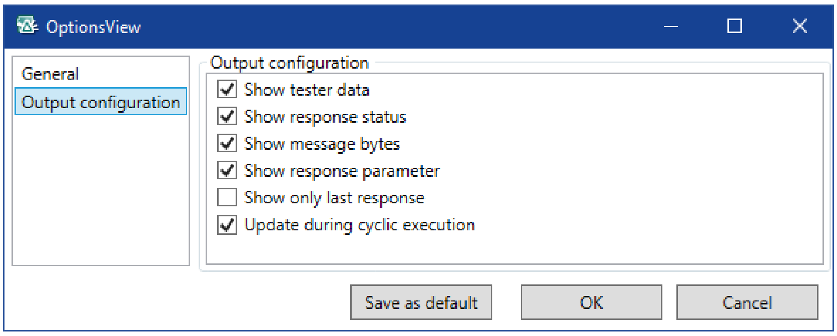
General

☑ Snapshot relevant

☑ Create measurement signals from response parameters

□ Show service ID in tree view

4. Activate the check box "Snapshot relevant" if the results of the service request are to be recorded in the snapshot (see the section "Snapshots" on page 60).
5. Activate the check box "Show service ID in tree view" if you want the ID of the Diagnostic Service to be displayed next to the name.
6. Activate the option "Create Measurement Signals from Response Parameters" if you want to add the signal measured subsequently to the signals available in the INCA variable selection (see"Diagnostic Signals in the INCA Variable Selection" on page 63).
7. Pressing the **Save as default** button makes any changes you have made become the default setting.
8. Select "Output Configuration" in the left-hand window.



Output configuration

☑ Show tester data

☑ Show response status

☑ Show message bytes

☑ Show response parameter

□ Show only last response

☑ Update during cyclic execution

9. Activate the required options.
10. Click **OK**.

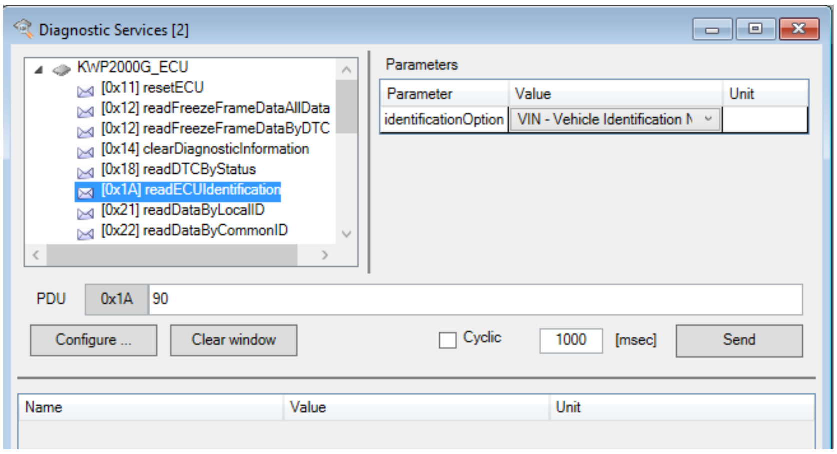
Refer to the following table for details of the significance of the options.

Option	Meaning
Show tester data	Service request and relevant parameters which were sent to the ECU
Show response status	Type of response of the ECU, response status
Show message bytes	Response of the ECU in hexadecimal notation
Show response parameter	ASAM MCD2D-interpreted response of the ECU
Show only last response	Shows only the data of the last service request. The data of previous service requests is deleted
Update during cyclic execution	If the cyclic repetation of the service request is activated, the display is updated in every cycle

To execute a diagnostic service

1. Choose **ODX** > **User views** > **Diagnostic Services**.

   The dialog window for selecting the service will be displayed.



KWP2000G_ECU
- [0x11] resetECU
- [0x12] readFreezeFrameDataAllData
- [0x12] readFreezeFrameDataByDTC
- [0x14] clearDiagnosticInformation
- [0x18] readDTCByStatus
- [0x1A] readECUIdentification
- [0x21] readDataByLocalID
- [0x22] readDataByCommonID

Parameters

Parameter	Value	Unit
identificationOption	VIN - Vehicle Identification N	

PDU 0x1A 90

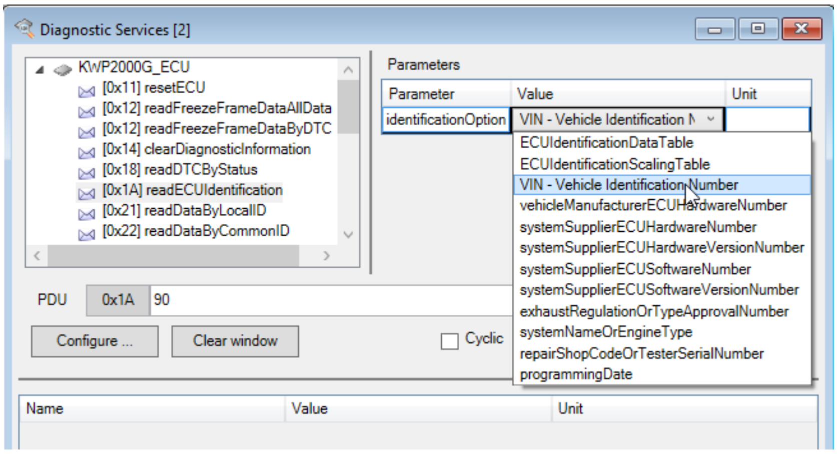
□ Cyclic 1000 [msec]

Name	Value	Unit

2. Select the service you want to execute from the top left section of the window.

   The top right section of the window will show the parameters for the service you selected.

3. In the "Values" column, select the parameter values for the service.


Diagnostic Services [2]
KWP2000G_ECU
[0x11] resetECU
[0x12] readFreezeFrameDataAllData
[0x12] readFreezeFrameDataByDTC
[0x14] clearDiagnosticInformation
[0x18] readDTCByStatus
[0x1A] readECUIdentification
[0x21] readDataByLocalID
[0x22] readDataByCommonID
ECUIdentificationDataTable
ECUIdentificationScalingTable
VIN - Vehicle Identification Number
vehicleManufacturerECUHardwareNumber
systemSupplierECUHardwareNumber
systemSupplierECUHardwareVersionNumber
systemSupplierECUSoftwareNumber
systemSupplierECUSoftwareVersionNumber
exhaustRegulationOrTypeApprovalNumber
systemNameOrEngineType
repairShopCodeOrTesterSerialNumber
programmingDate
Configure ...
Clear window

Parameters

Parameter	Value	Unit
identificationOption	VIN - Vehicle Identification N	

PDU 0x1A 90

☐ Cyclic

Name	Value	Unit

4. If the service is to be repeated on a regular basis, select the "Cyclic" check box and enter the cycle intervals.
5. Click **Send**.

   The service request is sent to the control unit and the response from the ECU will show up in the bottom section of the window.

The data (PDU, Protocol Data Units) sent to the ECU after the service ID has been entered, can be modified manually. To do this, use hexadecimal notation to enter the data in the "PDU" field. Please note that afterward, you can no longer select the parameters for this service.

Java Jobs

The ODX data model makes it possible to run Java code. These Java jobs are handled like diagnostic services - in particular, intermediate results can be issued while they are run.

Java jobs are indicated with a Java icon in the list of services - when selected in the list, both job parameters and diagnostic service parameters are shown on the right in the window and can also be edited there.

Java jobs are started using the **Send** button - depending on how complex they are these may take several seconds or even minutes to run.

Although not permissible in the ODX data model, working with Java GUIs is technically possible (see the example "JobDemo_JavaGUIs").

> **Note**
>
> The first time they are run, the Java windows may remain in the background! The windows are only in the foreground when they are run again.

4.1.2 Service Inspector

This function allows you to inspect all diagnostic data of an ECU or vehicle for a specific diagnostic request. I.e. the function iterates over all requested parameters of a service request as defined in the diagnostic database, then sends the requests and displays all returned results. The result for the service request will be displayed in clear text. The service requests and all clear text must be defined in the diagnostic database.

Note

In former INCA versions, the "Service Inspector" dialog box was named "ECUIdentification".

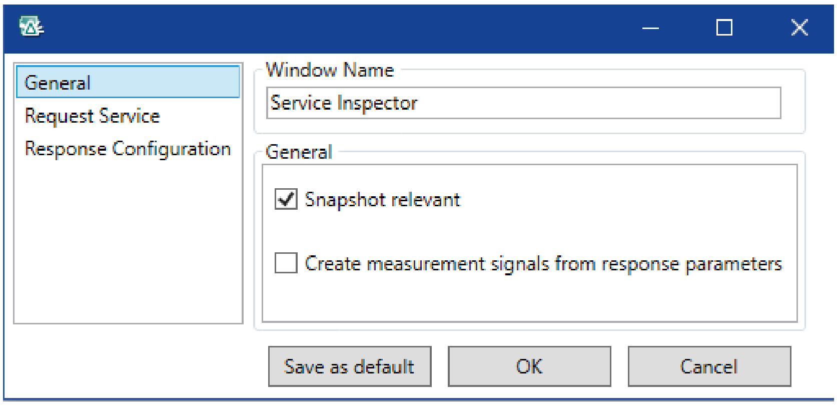
This function also allows you to configure the service request and the display of the results.

The window can also be used to generate diagnostic signals for the INCA variable selection (see "Diagnostic Signals in the INCA Variable Selection" on page 63).

To configure the Service Inspector

1. Select **ODX** > **User views** > **Service Inspector**.
2. Click **Configure**.

   The service request configuration dialog box is displayed.



Window Name: Service Inspector

General

☑ Snapshot relevant

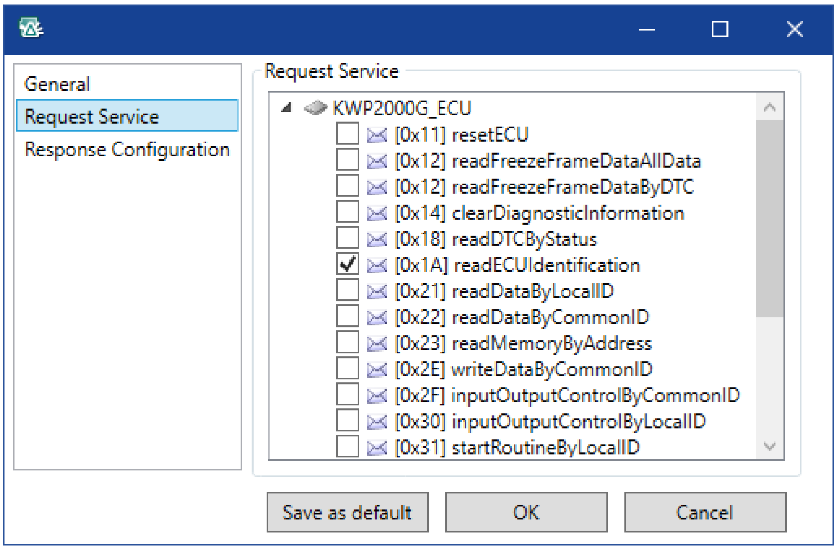
☐ Create measurement signals from response parameters

3. Click "General" in the left-hand window.
4. Activate the "Snapshot relevant" check box if the result of this service request is to be recorded in the snapshot (see the section "Snapshots" on page 60).
5. Activate the option "Create Measurement Signals from Response Parameters" if you want to add the signal measured subsequently to the diagnostic signal list (see "Diagnostic Signals in the INCA Variable Selection" on page 63).

   The configuration is complete.

or

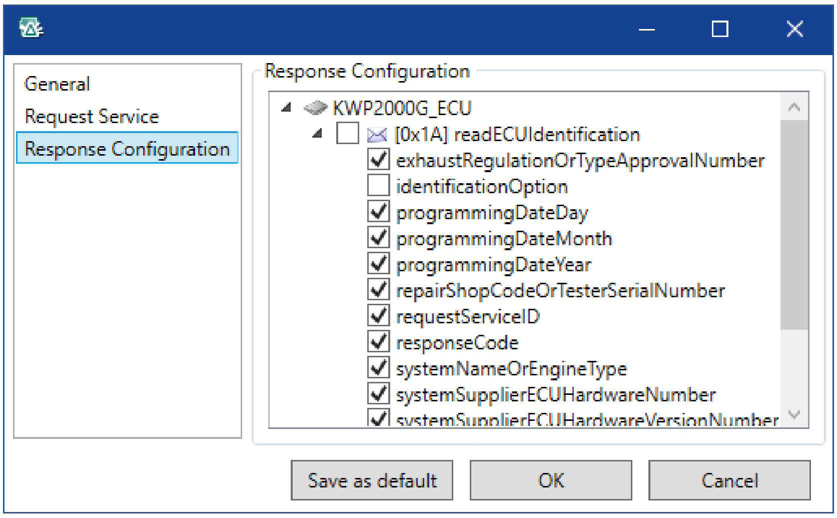
Click "Request Service" in the left-hand window.



Request Service

- KWP2000G_ECU
  - ☐ [0x11] resetECU
  - ☐ [0x12] readFreezeFrameDataAllData
  - ☐ [0x12] readFreezeFrameDataByDTC
  - ☐ [0x14] clearDiagnosticInformation
  - ☐ [0x18] readDTCByStatus
  - ☑ [0x1A] readECUIdentification
  - ☐ [0x21] readDataByLocalID
  - ☐ [0x22] readDataByCommonID
  - ☐ [0x23] readMemoryByAddress
  - ☐ [0x2E] writeDataByCommonID
  - ☐ [0x2F] inputOutputControlByCommonID
  - ☐ [0x30] inputOutputControlByLocalID
  - ☐ [0x31] startRoutineByLocalID

6. Select the service you want to assign to this function.
7. Click "Response Configuration" in the left-hand window.



Response Configuration

- KWP2000G_ECU
  - ☐ [0x1A] readECUIdentification
    - ☑ exhaustRegulationOrTypeApprovalNumber
    - ☐ identificationOption
    - ☑ programmingDateDay
    - ☑ programmingDateMonth
    - ☑ programmingDateYear
    - ☑ repairShopCodeOrTesterSerialNumber
    - ☑ requestServiceID
    - ☑ responseCode
    - ☑ systemNameOrEngineType
    - ☑ systemSupplierECUHardwareNumber
    - ☑ systemSupplierECUHardwareVersionNumber

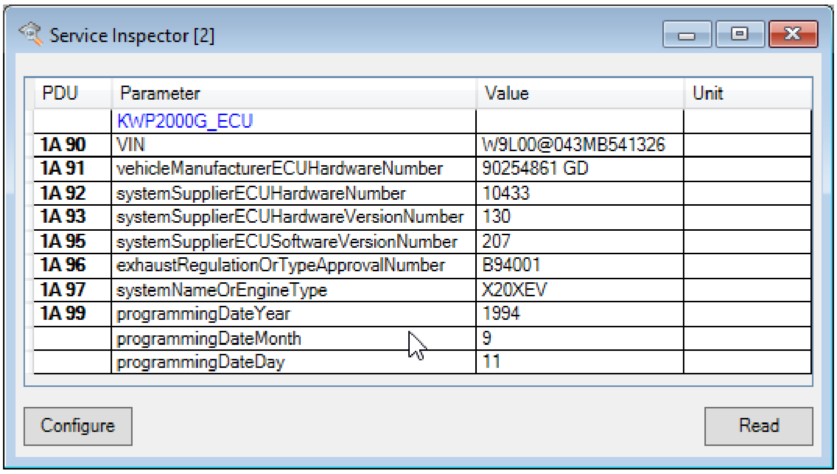
8. Select those response parameters which are to be displayed in the display of results of the service request.
9. Click **OK**.

To execute an Service Inspector

1. Select **ODX** > **User views** > **Service Inspector**.
2. Click **Read**.

   The service is executed for all service parameters defined in ODX and the

ECU responses will be displayed.



PDU	Parameter	Value	Unit
	KWP2000G_ECU		
1A 90	VIN	W9L00@043MB541326	
1A 91	vehicleManufacturerECUHardwareNumber	90254861 GD	
1A 92	systemSupplierECUHardwareNumber	10433	
1A 93	systemSupplierECUHardwareVersionNumber	130	
1A 95	systemSupplierECUSoftwareVersionNumber	207	
1A 96	exhaustRegulationOrTypeApprovalNumber	B94001	
1A 97	systemNameOrEngineType	X20XEV	
1A 99	programmingDateYear	1994	
	programmingDateMonth	9	
	programmingDateDay	11	

3. Click **Read** to execute the service again.

4.1.3 Hex Service

The **Hex Service** view allows you to send any data to the ECU through one of the defined interfaces. You select the diagnostic database for each ECU defined.

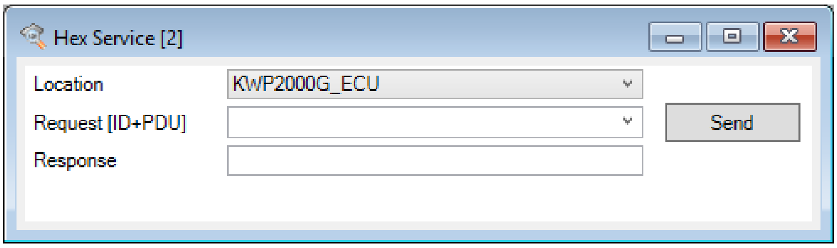
The response from the ECU is shown in hexadecimal notation.

The data you enter is stored in a history. You can use data entered within a session as many times as required.

To execute a hex service

1. Choose **ODX** > **User views** > **Hex Service**.

   The dialog window for specifying the request will be displayed.



Location: KWP2000G_ECU

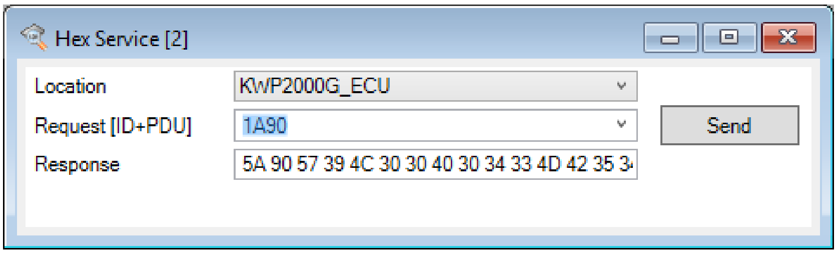
Request [ID+PDU]:

Response:

2. Select a logical link in the "Location" field.
3. In the "Request" field, enter the data (service ID and PDU) for the request that is to be sent to the ECU in hexadecimal notation.

or

   Select an existing entry from the selection list of the "Request" field.
4. Click **Send**.



Location	KWP2000G_ECU
Request [ID+PDU]	1A90
Response	5A 90 57 39 4C 30 30 40 30 34 33 4D 42 35 3

The response from the ECU will be displayed in the "Response" field in hexadecimal notation.

4.1.4 OBD

This user view is used to query and display OBD-relevant data. See also the sections "User Views" on page 19 and "The "OBD" User View" on page 20 for more general information.

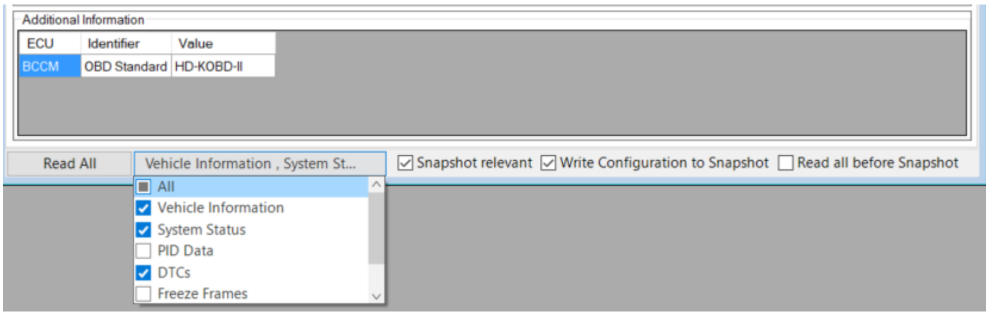
The OBD window provides two different views, one for OBDonCAN (SAE J1979 / ISO 15031-5 on ISO 15765-4) and another one for OBDonUDS (SAE J1979-2 on ISO 15765-4) including ZEVonUDS (J1979-3 on ISO 15765-4).

> **Note**
>
> Service requests for ZEVonUDS are also covered by the OBDonUDS view because the ZEVonUDS services are a subset of the OBDonUDS services.

To open the OBD window, click **ODX** › **User Views** › **OBD**. Depending on the selected ODX project for OBDonCAN or OBDonUDS, the OBD window opens in the corresponding view.

The global settings that you can make on the bottom of the OBD window are the same for both views.

Global Settings



Additional Information

ECU	Identifier	Value
BCCM	OBD Standard	HD-KOBD-II

☑ Snapshot relevant ☑ Write Configuration to Snapshot □ Read all before Snapshot

■ All
☑ Vehicle Information
☑ System Status
□ PID Data
☑ DTCs
□ Freeze Frames

- **Snapshot relevant**

  Activate this option if you want information from this user view to be included in the snapshot. A separate section is created in the snapshot file for every tab.

- **Write Configuration to Snapshot**

  Activate this option if you also want the relevant configuration settings to be included in the snapshot.

- **Read All before Snapshot**

  Activate this option if you want the **Read All** function to be executed before a snapshot. This ensures that the snapshot contains up-to-date ECU data.

- **The "Read All" Button and data selection**

  If all OBD data is to be read at the same time, click **Read All** at the bottom edge of the "OBD" window. Next to the **Read All** button, you can select for which tabs you want to perform the query. All selected data for PIDs, OBD-MIDs, monitors, vehicle information data, DTCs etc. that are supported by the connected ECUs are then read automatically according to the current configuration settings in each tab.

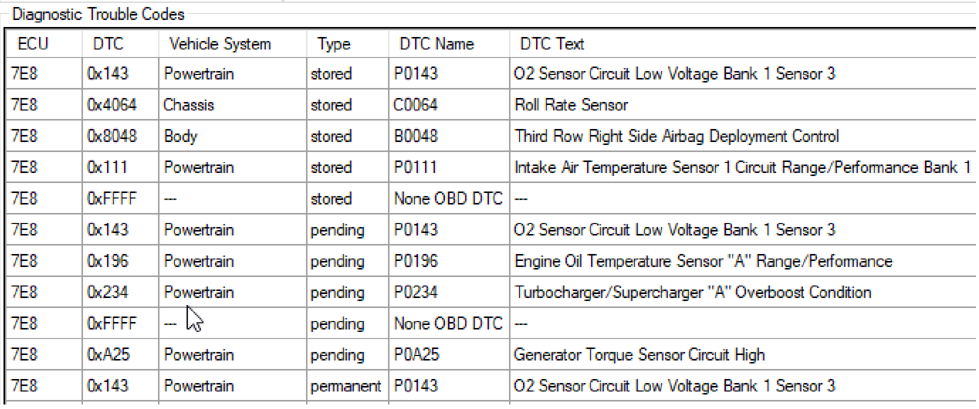
  As it can take some time to read all OBD data – depending on the quantity of supported data and the number of supported ECUs – progress is shown in a separate window.

Saved Settings

All global settings and also all settings that you can make in the tabs of the OBD window are saved as part of the Experiment configuration.

Displaying the Results

The results of the service request are shown in the individual tabs in the form of tables.



Diagnostic Trouble Codes

ECU	DTC	Vehicle System	Type	DTC Name	DTC Text
7E8	0x143	Powertrain	stored	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x4064	Chassis	stored	C0064	Roll Rate Sensor
7E8	0x8048	Body	stored	B0048	Third Row Right Side Airbag Deployment Control
7E8	0x111	Powertrain	stored	P0111	Intake Air Temperature Sensor 1 Circuit Range/Perfomance Bank 1
7E8	0xFFFF	---	stored	None OBD DTC	---
7E8	0x143	Powertrain	pending	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x196	Powertrain	pending	P0196	Engine Oil Temperature Sensor "A" Range/Perfomance
7E8	0x234	Powertrain	pending	P0234	Turbocharger/Supercharger "A" Overboost Condition
7E8	0xFFFF	---	pending	None OBD DTC	---
7E8	0xA25	Powertrain	pending	P0A25	Generator Torque Sensor Circuit High
7E8	0x143	Powertrain	permanent	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3

The individual columns of this table can be moved by Drag&Drop – the lines can also be sorted in ascending or descending order of entries by clicking a column heading (in the above figure, sorting takes place in ascending order in accordance with the content of the "DTC" column).

For information on the detailed description of the individual user views, see the following sections:

- "OBDonCAN (SAE J1979 / ISO 15031-5 on ISO 15765-4)" on the next page
- "OBDonUDS (SAE J1979-2 / SAE J1979-3)" on page 43

4.1.4.1 OBDonCAN (SAE J1979 / ISO 15031-5 on ISO 15765-4)

This user view is used to query and display OBD-relevant data according to SAE J1979 / ISO 15031-5 on ISO 15765-4.

See also the sections "User Views" on page 19 and "The "OBD" User View" on page 20 for more general information.

Emission-Related Diagnostic Services (SAE J1979 / ISO15031-5)

The services $01...$0A are reserved to acquire emission-related diagnostic services:

- **Service $01**

  Request current powertrain diagnostic data
- **Service $02**

  Request powertrain freeze frame data
- **Service $03**

  Show stored (emission-related) diagnostic trouble codes ("stored DTCs")
- **Service $04**

  Clear/reset emission-related diagnostic information
- **Service $06**

  Request on-board monitoring test results for specific monitored systems
- **Service $07**

  Request emission-related diagnostic trouble codes detected during current or last completed driving cycle ("pending DTCs")
- **Service $08**

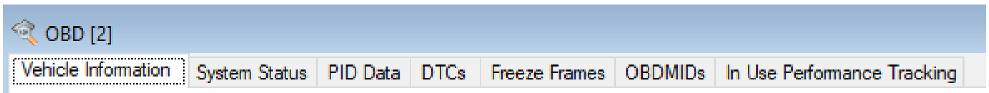
  Request control of on-board system, test or component
- **Service $09**

  Request vehicle information and information from In-Use Performance Tracking.
- **Service $0A**

  Request permanently stored trouble codes ("permanent DTCs").These emission-related trouble codes have a "permanent" status and cannot be deleted.

Grouping into Different Tabs

The information in the "OBD" user view is distributed to different tabs for reasons of clarity. This division does not, however, take place strictly in accordance with the functionality of the individual services, but is user-oriented.



These are:

- "OBDonCAN "Vehicle Information" Tab" below

  Information from service $09 is contained in this tab.
- "OBDonCAN "System Status" Tab" on the next page

  Information from service $01 is contained in this tab.
- "OBDonCAN "PID Data" Tab" on page 34

  Information from sercive $01 is contained in this tab, i.e. current diagnostic data from the powertrain.
- "OBDonCAN "DTCs" Tab" on page 36

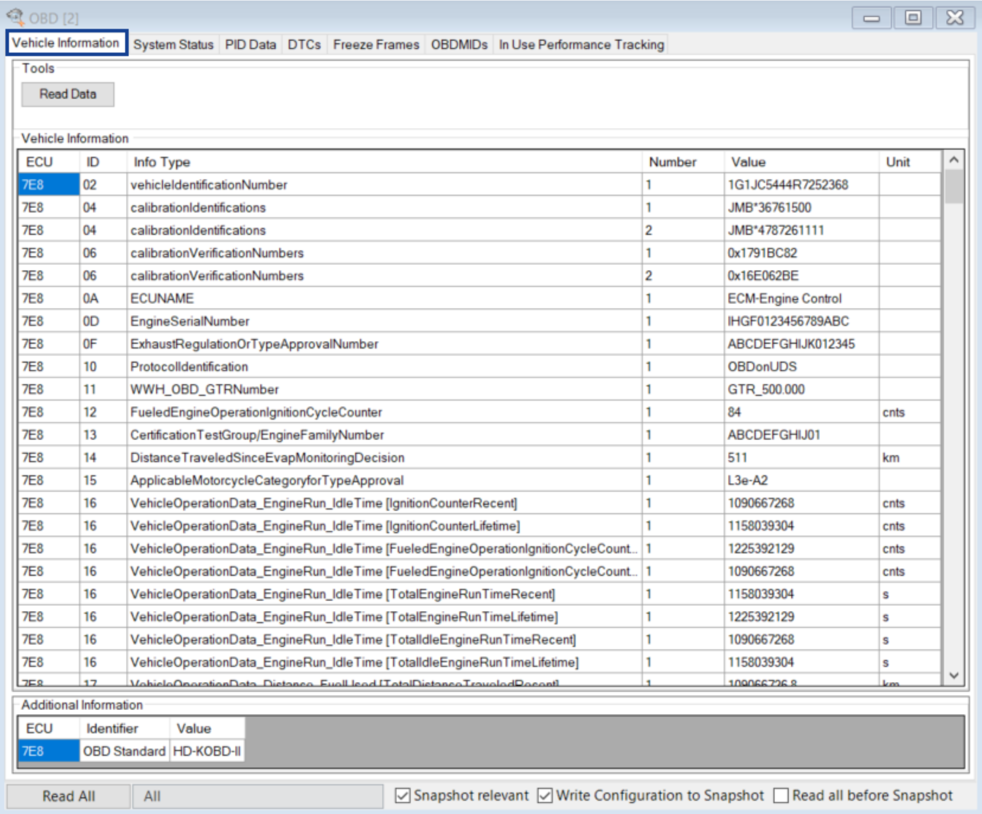
  Information from services $03, $04, $07, amd $0A is contained in this tab.
- "OBDonCAN "Freeze Frames" Tab" on page 38

  Information from service $02 is contained in this tab.
- "OBDonCAN "OBDMIDS" Tab" on page 39

  The OBD Monitor IDs of specially monitored systems (service $06) are queried in this tab.
- "OBDonCAN "In Use Performance Tracking" Tab" on page 42

OBDonCAN "**Vehicle Information**" Tab

Information from service $09 is contained in this tab.



ECU	ID	Info Type	Number	Value	Unit
7E8	02	vehicleIdentificationNumber	1	1G1JC5444R7252368	
7E8	04	calibrationIdentifications	1	JMB*36761500	
7E8	04	calibrationIdentifications	2	JMB*4787261111	
7E8	06	calibrationVerificationNumbers	1	0x1791BC82	
7E8	06	calibrationVerificationNumbers	2	0x16E062BE	
7E8	0A	ECUNAME	1	ECM-Engine Control	
7E8	0D	EngineSerialNumber	1	IHGF0123456789ABC	
7E8	0F	ExhaustRegulationOrTypeApprovalNumber	1	ABCDEFGHIJK012345	
7E8	10	ProtocolIdentification	1	OBDonUDS	
7E8	11	WWH_OBD_GTRNumber	1	GTR_500.000	
7E8	12	FueledEngineOperationIgnitionCycleCounter	1	84	cnts
7E8	13	CertificationTestGroup/EngineFamilyNumber	1	ABCDEFGHIJ01	
7E8	14	DistanceTraveledSinceEvapMonitoringDecision	1	511	km
7E8	15	ApplicableMotorcycleCategoryforTypeApproval	1	L3e-A2	
7E8	16	VehicleOperationData_EngineRun_IdleTime [IgnitionCounterRecent]	1	1090667268	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [IgnitionCounterLifetime]	1	1158039304	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [FueledEngineOperationIgnitionCycleCount...	1	1225392129	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [FueledEngineOperationIgnitionCycleCount...	1	1090667268	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalEngineRunTimeRecent]	1	1158039304	s
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalEngineRunTimeLifetime]	1	1225392129	s
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalIdleEngineRunTimeRecent]	1	1090667268	s
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalIdleEngineRunTimeLifetime]	1	1158039304	s
7E8	17	VehicleOperationData_Distance_FuelUsed [TotalDistanceTraveledRecent]	1	109066726.8	km

Additional Information

ECU	Identifier	Value
7E8	OBD Standard	HD-KOBD-II

☑ Snapshot relevant ☑ Write Configuration to Snapshot □ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

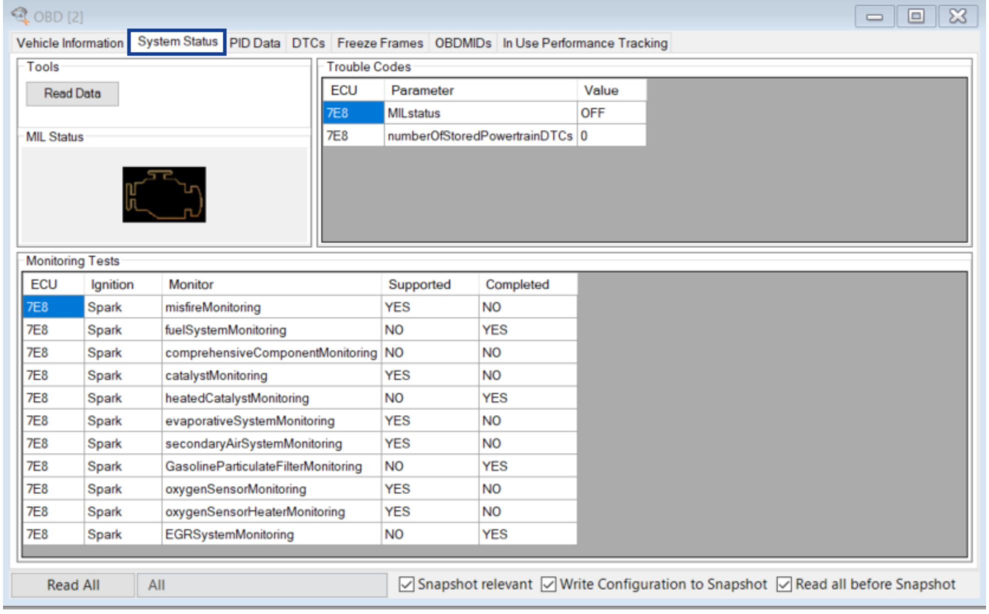
Tools: This field is used to read the data – click **Read Data** for this purpose.

ODX Database Vehicle Information: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
ID	Identifier of the Info Type
Info Type	InfoType for scaling and defining service $09
Number	ID (if INFOTYPE contains several pieces of information)
Value	Physical value of INFOTYPE

OBDonCAN "**System Status**" Tab

Information from service $01 is contained in this tab.



ECU	Parameter	Value
7E8	MILstatus	OFF
7E8	numberOfStoredPowertrainDTCs	0

ECU	Ignition	Monitor	Supported	Completed
7E8	Spark	misfireMonitoring	YES	NO
7E8	Spark	fuelSystemMonitoring	NO	YES
7E8	Spark	comprehensiveComponentMonitoring	NO	NO
7E8	Spark	catalystMonitoring	YES	NO
7E8	Spark	heatedCatalystMonitoring	NO	YES
7E8	Spark	evaporativeSystemMonitoring	YES	NO
7E8	Spark	secondaryAirSystemMonitoring	YES	NO
7E8	Spark	GasolineParticulateFilterMonitoring	NO	YES
7E8	Spark	oxygenSensorMonitoring	YES	NO
7E8	Spark	oxygenSensorHeaterMonitoring	YES	NO
7E8	Spark	EGRSystemMonitoring	NO	YES

The individual fields of the GUI contain the following functions and information:

Tools: This field is used to read the data – click **Read Data** for this purpose.

MIL Status: The icon of the MIL (Malfunction Indicator Lamp) is shown in this field.

Trouble Codes: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Parameter	Name of the parameter
Value	Physical value of the parameter

Monitoring Tests: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Monitor	Monitor
Supported	Is the monitor supported?
Completed	Was the monitor ended?

Note

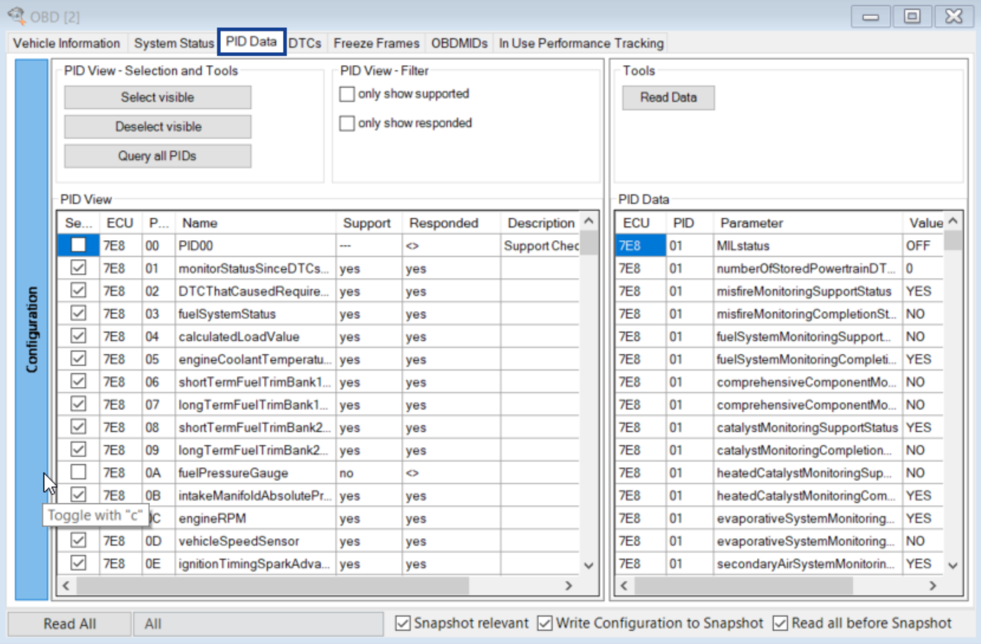
If an ECU identifies itself via service $01, PID01 as a diesel ECU, the monitors relevant for diesel ECUs are displayed, otherwise the monitors for gasoline ECUs.

OBDonCAN "**PID Data**" Tab

Information from service $01 is contained in this tab, i.e. current diagnostic data from the powertrain.

PIDs (parameter identifiers) are the identifiers for the information supported by the engine ECU.

To select the PIDs to be queried, click **Configuration** – the window then shows additional fields (shown below with an *).



☐ only show supported
☐ only show responded

Se...	ECU	P...	Name	Support	Responded	Description
☐	7E8	00	PID00	---	<>	Support Chec
☑	7E8	01	monitorStatusSinceDTCs...	yes	yes	
☑	7E8	02	DTCThatCausedRequire...	yes	yes	
☑	7E8	03	fuelSystemStatus	yes	yes	
☑	7E8	04	calculatedLoadValue	yes	yes	
☑	7E8	05	engineCoolantTemperatu...	yes	yes	
☑	7E8	06	shortTermFuelTrimBank1...	yes	yes	
☑	7E8	07	longTermFuelTrimBank1...	yes	yes	
☑	7E8	08	shortTermFuelTrimBank2...	yes	yes	
☑	7E8	09	longTermFuelTrimBank2...	yes	yes	
☐	7E8	0A	fuelPressureGauge	no	<>	
☑	7E8	0B	intakeManifoldAbsolutePr...	yes	yes	
		C	engineRPM	yes	yes	
☑	7E8	0D	vehicleSpeedSensor	yes	yes	
☑	7E8	0E	ignitionTimingSparkAdva...	yes	yes	

ECU	PID	Parameter	Value
7E8	01	MILstatus	OFF
7E8	01	numberOfStoredPowertrainDT...	0
7E8	01	misfireMonitoringSupportStatus	YES
7E8	01	misfireMonitoringCompletionSt...	NO
7E8	01	fuelSystemMonitoringSupport...	NO
7E8	01	fuelSystemMonitoringCompleti...	YES
7E8	01	comprehensiveComponentMo...	NO
7E8	01	comprehensiveComponentMo...	NO
7E8	01	catalystMonitoringSupportStatus	YES
7E8	01	catalystMonitoringCompletion...	NO
7E8	01	heatedCatalystMonitoringSup...	NO
7E8	01	heatedCatalystMonitoringCom...	YES
7E8	01	evaporativeSystemMonitoring...	YES
7E8	01	evaporativeSystemMonitoring...	NO
7E8	01	secondaryAirSystemMonitorin...	YES

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

***PID View - Selection and Tools**: Using these buttons, you can make a kind of global selection of the PIDs to be queried in the "PID View" list.

- **Select visible**

  Selects all PIDs visible in the "PID View" list (see "OBDonCAN (SAE J1979 / ISO 15031-5 on ISO 15765-4)" on page 31)

- **Deselect visible**

  The selection of visible PIDs is undone

- **Query all PIDs**

  Each individual PID is addressed and then checked to see if a response is returned

***PID View - Filter**: Uses filter criteria with regard to the display in the "PID View" list. The following options are available:

- **only show supported**

  If this option is selected, only the PIDs supported by the ECU are made available for selection in the "PID View" list.

- **only show responded**

  If this option is selected, only the PIDs answered by the ECU after **Query all PIDs** (see above) are made available for selection in the "PID View" list.

Tools: This field is used to read the data – click **Read Data** for this purpose.

***PID View**: This table displays the selected PIDs. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
Select	Selection of the PID
ECU	Name of the logical link (ECU) from hardware configuration
PID	PID
Name	Explicit name of the PID
Support	Is this PID supported? (queried from ECU)
Responded	Was the query of this PID answered (via **Query all PIDs**)?
Description	Explanatory text (if in the database)

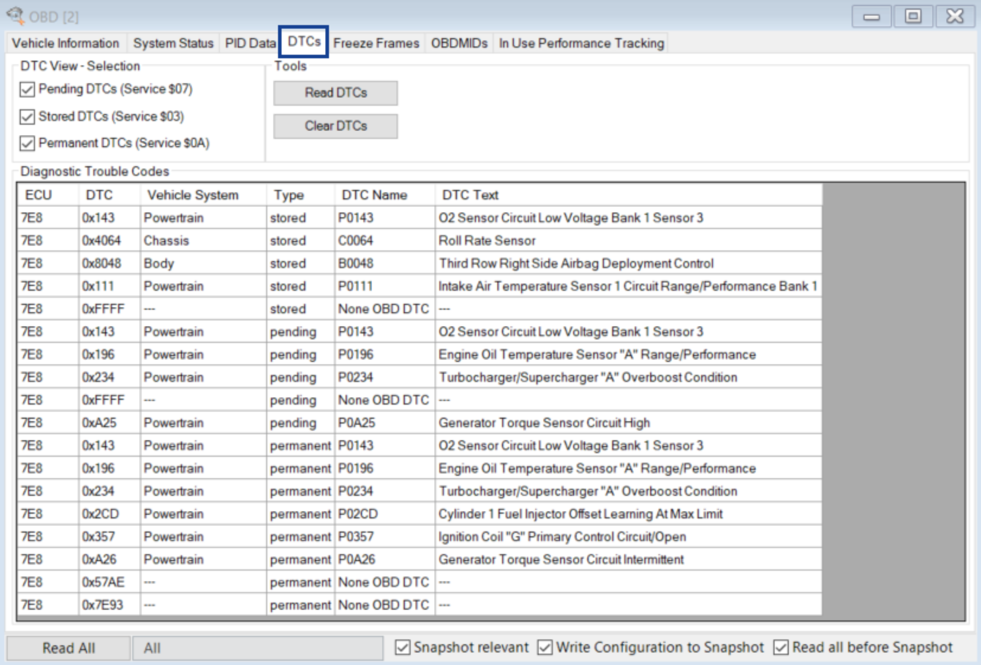
PID Data

This table displays the results of the query/queries. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
PID	PID
Parameter	Explicit name as one PID can consist of several pieces of information
Value	Physical value of the parameter
Unit	Unit of PID (if available)

OBDonCAN "**DTCs**" Tab

Information from services $03, $04, $07, and $0A is contained in this tab.



DTC View - Selection

☑ Pending DTCs (Service $07)

☑ Stored DTCs (Service $03)

☑ Permanent DTCs (Service $0A)

Diagnostic Trouble Codes

ECU	DTC	Vehicle System	Type	DTC Name	DTC Text
7E8	0x143	Powertrain	stored	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x4064	Chassis	stored	C0064	Roll Rate Sensor
7E8	0x8048	Body	stored	B0048	Third Row Right Side Airbag Deployment Control
7E8	0x111	Powertrain	stored	P0111	Intake Air Temperature Sensor 1 Circuit Range/Performance Bank 1
7E8	0xFFFF	---	stored	None OBD DTC	---
7E8	0x143	Powertrain	pending	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x196	Powertrain	pending	P0196	Engine Oil Temperature Sensor "A" Range/Performance
7E8	0x234	Powertrain	pending	P0234	Turbocharger/Supercharger "A" Overboost Condition
7E8	0xFFFF	---	pending	None OBD DTC	---
7E8	0xA25	Powertrain	pending	P0A25	Generator Torque Sensor Circuit High
7E8	0x143	Powertrain	permanent	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x196	Powertrain	permanent	P0196	Engine Oil Temperature Sensor "A" Range/Performance
7E8	0x234	Powertrain	permanent	P0234	Turbocharger/Supercharger "A" Overboost Condition
7E8	0x2CD	Powertrain	permanent	P02CD	Cylinder 1 Fuel Injector Offset Learning At Max Limit
7E8	0x357	Powertrain	permanent	P0357	Ignition Coil "G" Primary Control Circuit/Open
7E8	0xA26	Powertrain	permanent	P0A26	Generator Torque Sensor Circuit Intermittent
7E8	0x57AE	---	permanent	None OBD DTC	---
7E8	0x7E93	---	permanent	None OBD DTC	---

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

DTC View Selection: Selection options for the display of the DTCs in the "Diagnostic Trouble Codes" list. The following options are available:

- Pending DTCs

  All pending DTCs (service $07) are displayed

- Stored DTCs

  All stored DTCs (service $03) are displayed

- Permanent DTCs

  All permanent DTCs (service $0A) are displayed

Tools: This field is used to read the Diagnostic Trouble Codes – click **Read DTCs** for this purpose.

Diagnostic Trouble Codes: This table displays the queried DTCs.

The meaning of each individual entry in the list is described in the following table:

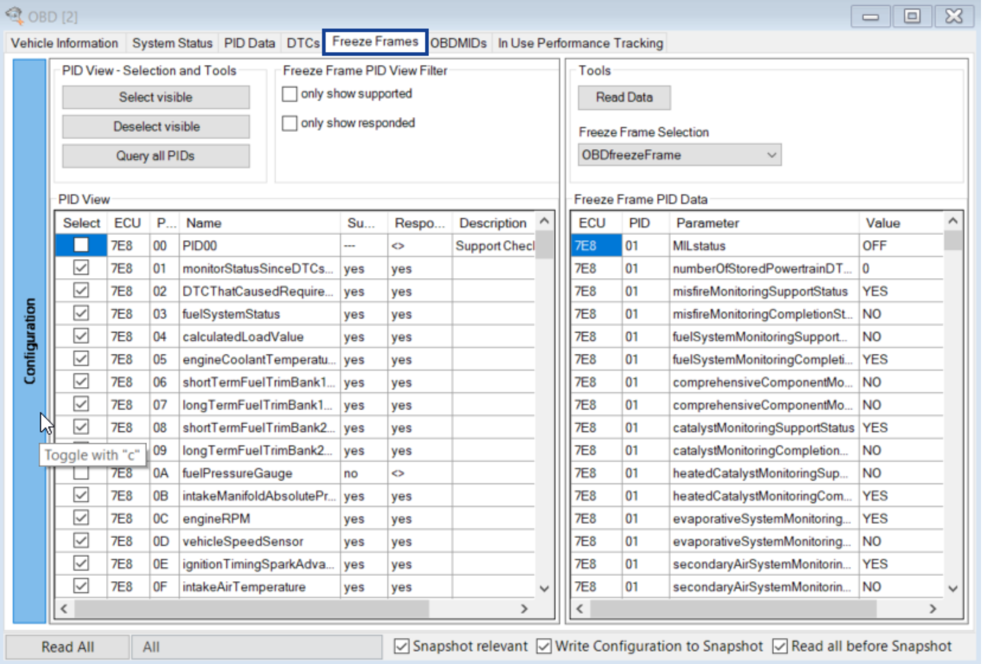
Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
DTC	DTC in hex notation (e.g. 0x143)
Vehicle System	Vehicle system, e.g. powertrain, body, chassis

Column	Meaning
Type	"Pending", "Stored", or "Permanent"
DTC Name	Name of the DTC (e.g. P0143)
DTC Text	Explanatory text on the DTC (e.g. "O2 Sensor Circuit Low Voltage, Bank 1 Sensor 3")

OBDonCAN "**Freeze Frames**" Tab

Information from service $02 is contained in this tab.

To select the freeze-frame data to be displayed, click **Configuration** – the window then shows additional fields (shown below with an *).



□ only show supported

□ only show responded

Freeze Frame Selection: OBDfreezeFrame

Select	ECU	P...	Name	Su...	Respo...	Description
□	7E8	00	PID00	---	<>	Support Checl
☑	7E8	01	monitorStatusSinceDTCs...	yes	yes	
☑	7E8	02	DTCThatCausedRequire...	yes	yes	
☑	7E8	03	fuelSystemStatus	yes	yes	
☑	7E8	04	calculatedLoadValue	yes	yes	
☑	7E8	05	engineCoolantTemperatu...	yes	yes	
☑	7E8	06	shortTermFuelTrimBank1...	yes	yes	
☑	7E8	07	longTermFuelTrimBank1...	yes	yes	
☑	7E8	08	shortTermFuelTrimBank2...	yes	yes	
		09	longTermFuelTrimBank2...	yes	yes	
□	7E8	0A	fuelPressureGauge	no	<>	
☑	7E8	0B	intakeManifoldAbsolutePr...	yes	yes	
☑	7E8	0C	engineRPM	yes	yes	
☑	7E8	0D	vehicleSpeedSensor	yes	yes	
☑	7E8	0E	ignitionTimingSparkAdva...	yes	yes	
☑	7E8	0F	intakeAirTemperature	yes	yes	

ECU	PID	Parameter	Value
7E8	01	MILstatus	OFF
7E8	01	numberOfStoredPowertrainDT...	0
7E8	01	misfireMonitoringSupportStatus	YES
7E8	01	misfireMonitoringCompletionSt...	NO
7E8	01	fuelSystemMonitoringSupport...	NO
7E8	01	fuelSystemMonitoringCompleti...	YES
7E8	01	comprehensiveComponentMo...	NO
7E8	01	comprehensiveComponentMo...	NO
7E8	01	catalystMonitoringSupportStatus	YES
7E8	01	catalystMonitoringCompletion...	NO
7E8	01	heatedCatalystMonitoringSup...	NO
7E8	01	heatedCatalystMonitoringCom...	YES
7E8	01	evaporativeSystemMonitoring...	YES
7E8	01	evaporativeSystemMonitoring...	NO
7E8	01	secondaryAirSystemMonitorin...	YES
7E8	01	secondaryAirSystemMonitorin...	NO

All

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

***PID View - Selection and Tools**: Using these buttons, you can make a kind of global selection of the freeze frame PIDs to be queried in the "PID View" list.

- **Select visible**

  Selects all PIDs visible in the "PID View" list (see "OBDonCAN (SAE J1979 / ISO 15031-5 on ISO 15765-4)" on page 31)

- **Deselect visible**

  The selection of visible PIDs is undone

- **Query all PIDs**

  Each individual freeze frame PID is addressed and then checked to see if a response is returned

***Freeze Frame PID View Filter**:Uses filter criteria with regard to the display in the „PID View" list. The following options are available:

- **Only show supported**

  If this option is selected, only those PIDs supported by the ECU are made available for selection in the „PID View" list.

- **Only show responded**

  If this option is selected, only the PIDs answered by the ECU after **Query all PIDs** (see above) are made available for selection in the „PID View" list.

Tools: This field is used to read the data – click **Read Data** for this purpose.

In the "PID View" field, you can choose between general OBD Freeze Frames and manufacturer-specific Freeze Frames.

***PID View**: In this table, the Freeze Frame PIDs to be queried are selected. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
Select	Selection of the PID
ECU	Name of the logical link (ECU) from hardware configuration
PID	PID
Name	Explicit name of the PID
Support	Is this PID supported?
Responded	Was the query of this PID answered (via **Query all PIDs**)?
Description	Explanatory text (if in the database)

Freeze Frame PID Data: This table displays the results of the query/queries. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
PID	PID
Parameter	Explicit name as one PID can consist of several pieces of information
Value	Physical value of the parameter

OBDonCAN "**OBDMIDS**" Tab

The OBD Monitor IDs of specially monitored systems (service $06) are queried in this tab.

For reasons of clarity, this window is divided into several tabs:

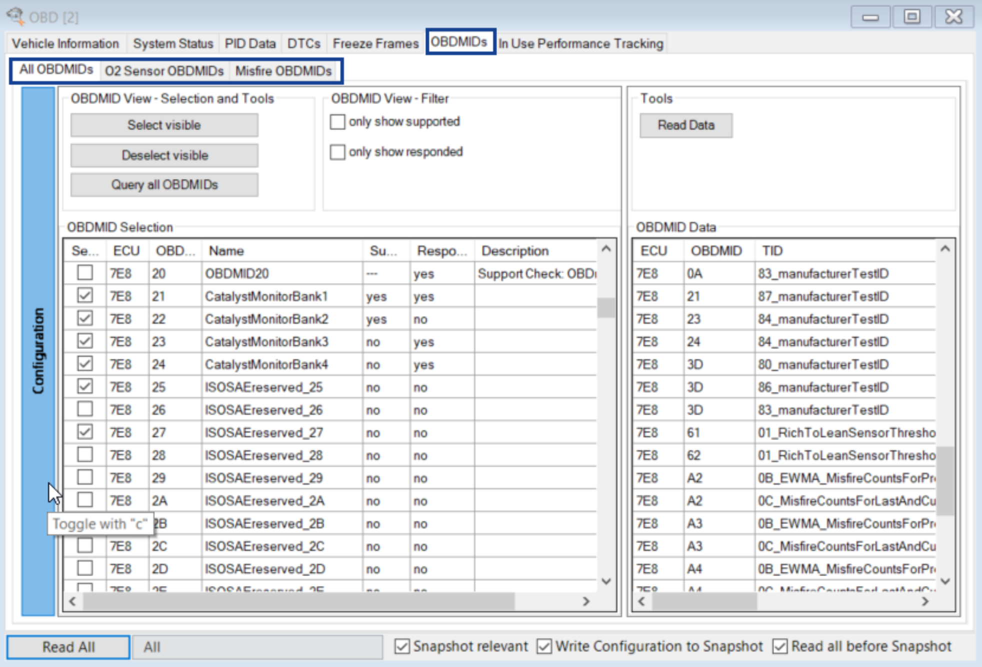
- "All OBDMIDs" tab

  All OBDMIDs are displayed in this tab
- "O2 Sensor OBDMIDs" tab

  All OBDMIDs connected with the O2 sensor monitor are displayed in this tab
- "Misfire OBDMIDs" tab

  All OBDMIDs connected with the misfire monitor are displayed in this tab

To select the OBDMIDs, click **Configuration** – the window then shows additional fields (shown below with an *).



OBDMID View - Filter

□ only show supported

□ only show responded

OBDMID Selection

Se...	ECU	OBD...	Name	Su...	Respo...	Description
□	7E8	20	OBDMID20	---	yes	Support Check: OBD
☑	7E8	21	CatalystMonitorBank1	yes	yes	
☑	7E8	22	CatalystMonitorBank2	yes	no	
☑	7E8	23	CatalystMonitorBank3	no	yes	
☑	7E8	24	CatalystMonitorBank4	no	yes	
☑	7E8	25	ISOSAEreserved_25	no	no	
□	7E8	26	ISOSAEreserved_26	no	no	
☑	7E8	27	ISOSAEreserved_27	no	no	
□	7E8	28	ISOSAEreserved_28	no	no	
□	7E8	29	ISOSAEreserved_29	no	no	
□	7E8	2A	ISOSAEreserved_2A	no	no	
		2B	ISOSAEreserved_2B	no	no	
□	7E8	2C	ISOSAEreserved_2C	no	no	
□	7E8	2D	ISOSAEreserved_2D	no	no	

OBDMID Data

ECU	OBDMID	TID
7E8	0A	83_manufacturerTestID
7E8	21	87_manufacturerTestID
7E8	23	84_manufacturerTestID
7E8	24	84_manufacturerTestID
7E8	3D	80_manufacturerTestID
7E8	3D	86_manufacturerTestID
7E8	3D	83_manufacturerTestID
7E8	61	01_RichToLeanSensorThresho
7E8	62	01_RichToLeanSensorThresho
7E8	A2	0B_EWMA_MisfireCountsForPr
7E8	A2	0C_MisfireCountsForLastAndCu
7E8	A3	0B_EWMA_MisfireCountsForPr
7E8	A3	0C_MisfireCountsForLastAndCu
7E8	A4	0B_EWMA_MisfireCountsForPr

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

***OBDMID View - Selection Tools**: Using these buttons, you can make a kind of global selection of the OBDMIDs to be queried in the "OBDMID Selection" list.

- **Select visible**

  Selects all OBDMIDs visible in the "OBDMID Selection" list (see "OBDonCAN (SAE J1979 / ISO 15031-5 on ISO 15765-4)" on page 31).
- **Deselect visible**

  The selection of visible OBDMIDs is undone
- **Query all OBDMIDs**

  Each individual OBDMID is addressed and then checked to see if a response is returned

***OBDMID View - Filter**: Uses filter criteria with regard to the display in the "OBDMID Selection" list. The following options are available:

- **only show supported**

  If this option is selected, only those OBDMIDs supported by the ECU are made available for selection in the "OBDMID Selection" list.

- **only show responded**

  If this option is selected, only the OBDMIDs answered by the ECU after **Query all OBDMIDs** (see above) are made available for selection in the "OBDMID Selection" list.

Tools: This field is used to read the data – click **Read Data** for this purpose.

***OBDMID Selection**: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
Select	Selection of the OBDMID
ECU	Name of the logical link (ECU) from hardware configuration
OBDMID	On-Board Diagnostic Monitor ID
Name	Explicit name of the ODBMID
Support	Is this OBDMID supported?
Responded	Was the query of this OBDMID answered (via **Query all OBDMIDs**)?
Description	Explanatory text (if in the database)

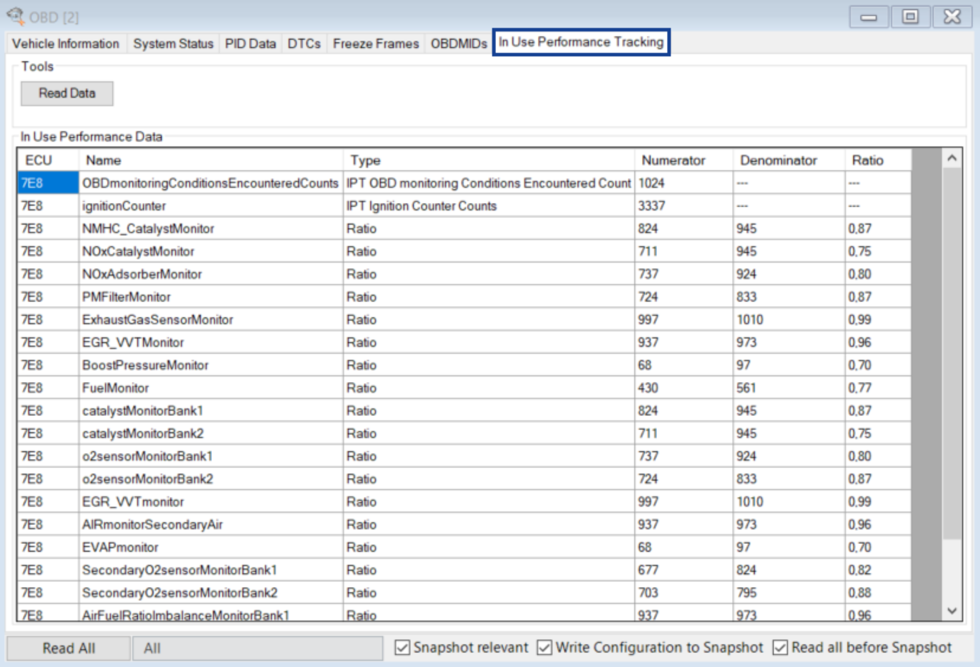
OBDMIM Data: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
OBDMID	On-Board Diagnostic Monitor ID
TID	Test ID of service 08
UnitAndScalingID	Unit and Scaling ID (1 Byte)
Test Value	Value read from the ECU
Min Test Limit	Minimum test limit

Column	Meaning
Max Test Limit	Maximum test limit
OBDMID Name	Explicit name of the ODBMID
Description	Explanatory text (if in the database)

OBDonCAN "**In Use Performance Tracking**" Tab

In this tab the data of the In-Use Performance Tracking of service $09 is displayed.



ECU	Name	Type	Numerator	Denominator	Ratio
7E8	OBDmonitoringConditionsEncounteredCounts	IPT OBD monitoring Conditions Encountered Count	1024	---	---
7E8	ignitionCounter	IPT Ignition Counter Counts	3337	---	---
7E8	NMHC_CatalystMonitor	Ratio	824	945	0,87
7E8	NOxCatalystMonitor	Ratio	711	945	0,75
7E8	NOxAdsorberMonitor	Ratio	737	924	0,80
7E8	PMFilterMonitor	Ratio	724	833	0,87
7E8	ExhaustGasSensorMonitor	Ratio	997	1010	0,99
7E8	EGR_VVTMonitor	Ratio	937	973	0,96
7E8	BoostPressureMonitor	Ratio	68	97	0,70
7E8	FuelMonitor	Ratio	430	561	0,77
7E8	catalystMonitorBank1	Ratio	824	945	0,87
7E8	catalystMonitorBank2	Ratio	711	945	0,75
7E8	o2sensorMonitorBank1	Ratio	737	924	0,80
7E8	o2sensorMonitorBank2	Ratio	724	833	0,87
7E8	EGR_VVTmonitor	Ratio	997	1010	0,99
7E8	AIRmonitorSecondaryAir	Ratio	937	973	0,96
7E8	EVAPmonitor	Ratio	68	97	0,70
7E8	SecondaryO2sensorMonitorBank1	Ratio	677	824	0,82
7E8	SecondaryO2sensorMonitorBank2	Ratio	703	795	0,88
7E8	AirFuelRatioImbalanceMonitorBank1	Ratio	937	973	0,96

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

Tools: This field is used to read the data – click **Read Data**.

In Use Performance Data: This table displays the queried information of service $09. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Name	Name of the parameter
Type	"General Denominator", "Counter", "Denominator" or "Numerator"

Column	Meaning
Numerator	Tracks the number of times that the vehicle has been operated in the specified conditions. These conditions are specified for each monitored component or system.
Denominator	Tracks the number of times that all conditions necessary for a specific monitor to detect a malfunction have been encountered
Ratio	Ratio of the values above

4.1.4.2 OBDonUDS (SAE J1979-2 / SAE J1979-3)

This user view is used to query and display OBD-relevant data.

See also the sections "User Views" on page 19 and "The "OBD" User View" on page 20 for more general information.

Diagnostic Services (SAE J1979-2 / SAE J1979-3)

The OBDonUDS standard SAE J1979-2 and ZEVonUDS standard SAE J1979-3 make use of UDS protocol services as defined in ISO14229-1.

The following UDS services are used by the OBD user view to acquire diagnostic data:

- **Service $14**

  Clear/Reset Emission-Related Diagnostic Information
- **Service $19**

  Request Emission-Related Diagnostic Trouble Codes with Confirmed Status, Subfunction $42

  Request Emission-Related Diagnostic Trouble Codes with Pending Status, Subfunction $42

  Request Emission-Related Diagnostic Trouble Codes with Permanent Status, Subfunction $55

  Request Powertrain Freeze Frame Data, Subfunction $04

  Request Supported DTCExtendedRecord Information, Subfunction $1A

  Request DTCExtendedDataRecord, Subfunction $06

  Request DTCs for a ReadinessGroup, Subfunction $56
- **Service $22**

  Request Current Powertrain Diagnostic Data (PIDs $F400 - $F5FF)

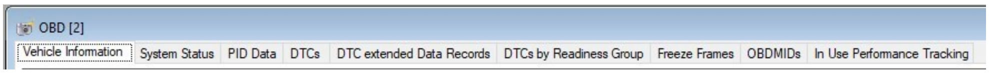
  Request On-Board Monitoring Test Results for Specific Monitored Systems (MIDs $F600 - $F6FF)

  Request Vehicle Information (ITIDs $F800 - $F8FF)

The additional OBDonUDS service $31 “Request Control of On-Board System, Test, or Component Service” is not provided by the OBD user view but can be used in the Diagnostic Services user view.

Grouping into Different Tabs

The information in the "OBD" user view is distributed to different tabs for reasons of clarity. This division does not, however, take place strictly in accordance with the functionality of the individual services, but is user-oriented.



These are:

- "OBDonUDS "Vehicle Information" Tab" on the next page

  Information from service $22 for Vehicle Information ITIDs $F800 - $F8FF is contained in this tab.

- "OBDonUDS "System Status" Tab" on the next page

  Information from service $22 for Current Powertrain Diagnostic Data PID $F501 is contained in this tab.

- "OBDonUDS "PID Data" Tab" on page 47

  Information from service $22 for Current Powertrain Diagnostic Data PIDs $F400 - $F5FF is contained in this tab.

- "OBDonUDS "DTCs" Tab" on page 48

  Information from services $19, subfunctions $42 and $55 with Emission-Related Diagnostic Trouble Codes with Confirmed, Pending and Permanent Status is contained in this tab.

- "OBDonUDS "DTC extended Data Records" Tab" on page 50

  Information from services $19, subfunctions $1A and $06 with DTCExtendedDataRecord information for all supported DTCExtendedRecords is contained in this tab.

- "OBDonUDS "DTCs by Readiness Group" Tab" on page 51

  Information from services $19, subfunction $56 with DTCs for all ReadinessGroups is contained in this tab.

- "OBDonUDS "Freeze Frames" Tab" on page 52

  Information from service $22, subfunction $04 with Powertrain Freeze Frame Data for pending and confirmed DTCs is contained in this tab.

- "OBDonUDS "OBDMIDS" Tab" on page 53

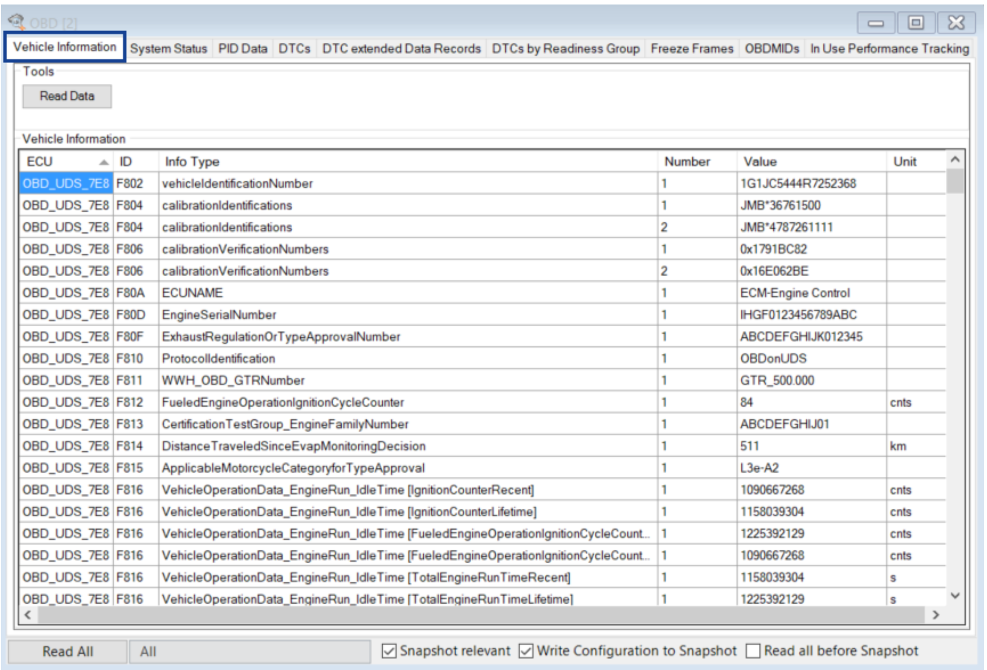
  Information from service $22 for On-Board Monitoring Test Results for Specific Monitored Systems MIDs $F600 - $F6FF is contained in this tab.

- "OBDonUDS "In Use Performance Tracking" Tab" on page 55

  In this tab the data of the In Use Performance Tracking of service $22 is dis-played for ITIDs $F808, $F80B.

OBDonUDS "**Vehicle Information**" Tab

Information from service $22 for Vehicle Information ITIDs $F800 - $F8FF is contained in this tab.



ECU	ID	Info Type	Number	Value	Unit
OBD_UDS_7E8	F802	vehicleIdentificationNumber	1	1G1JC5444R7252368	
OBD_UDS_7E8	F804	calibrationIdentifications	1	JMB*36761500	
OBD_UDS_7E8	F804	calibrationIdentifications	2	JMB*4787261111	
OBD_UDS_7E8	F806	calibrationVerificationNumbers	1	0x1791BC82	
OBD_UDS_7E8	F806	calibrationVerificationNumbers	2	0x16E062BE	
OBD_UDS_7E8	F80A	ECUNAME	1	ECM-Engine Control	
OBD_UDS_7E8	F80D	EngineSerialNumber	1	IHGF0123456789ABC	
OBD_UDS_7E8	F80F	ExhaustRegulationOrTypeApprovalNumber	1	ABCDEFGHIJK012345	
OBD_UDS_7E8	F810	ProtocolIdentification	1	OBDonUDS	
OBD_UDS_7E8	F811	WWH_OBD_GTRNumber	1	GTR_500.000	
OBD_UDS_7E8	F812	FueledEngineOperationIgnitionCycleCounter	1	84	cnts
OBD_UDS_7E8	F813	CertificationTestGroup_EngineFamilyNumber	1	ABCDEFGHIJ01	
OBD_UDS_7E8	F814	DistanceTraveledSinceEvapMonitoringDecision	1	511	km
OBD_UDS_7E8	F815	ApplicableMotorcycleCategoryforTypeApproval	1	L3e-A2	
OBD_UDS_7E8	F816	VehicleOperationData_EngineRun_IdleTime [IgnitionCounterRecent]	1	1090667268	cnts
OBD_UDS_7E8	F816	VehicleOperationData_EngineRun_IdleTime [IgnitionCounterLifetime]	1	1158039304	cnts
OBD_UDS_7E8	F816	VehicleOperationData_EngineRun_IdleTime [FueledEngineOperationIgnitionCycleCount...	1	1225392129	cnts
OBD_UDS_7E8	F816	VehicleOperationData_EngineRun_IdleTime [FueledEngineOperationIgnitionCycleCount...	1	1090667268	cnts
OBD_UDS_7E8	F816	VehicleOperationData_EngineRun_IdleTime [TotalEngineRunTimeRecent]	1	1158039304	s
OBD_UDS_7E8	F816	VehicleOperationData_EngineRun_IdleTime [TotalEngineRunTimeLifetime]	1	1225392129	s

☑ Snapshot relevant ☑ Write Configuration to Snapshot □ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

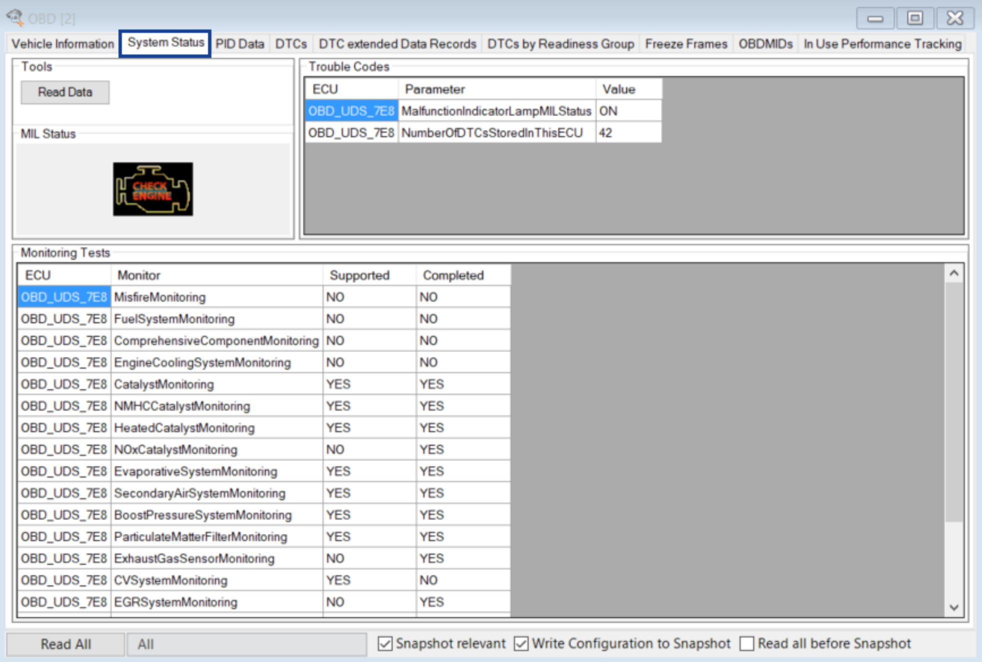
Tools: This field is used to read the data – click **Read Data** for this purpose.

Vehicle Information: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
ID	Identifier for vehicle information
Info Type	InfoType parameter name
Number	Number of vehicle information (>1 if the information occurs more than one time)
Value	Physical value of INFOTYPE

OBDonUDS "**System Status**" Tab

Information from service $22 for Current Powertrain Diagnostic Data PID $F501 is contained in this tab.



ECU	Parameter	Value
OBD_UDS_7E8	MalfunctionIndicatorLampMILStatus	ON
OBD_UDS_7E8	NumberOfDTCsStoredInThisECU	42

ECU	Monitor	Supported	Completed
OBD_UDS_7E8	MisfireMonitoring	NO	NO
OBD_UDS_7E8	FuelSystemMonitoring	NO	NO
OBD_UDS_7E8	ComprehensiveComponentMonitoring	NO	NO
OBD_UDS_7E8	EngineCoolingSystemMonitoring	NO	NO
OBD_UDS_7E8	CatalystMonitoring	YES	YES
OBD_UDS_7E8	NMHCCatalystMonitoring	YES	YES
OBD_UDS_7E8	HeatedCatalystMonitoring	YES	YES
OBD_UDS_7E8	NOxCatalystMonitoring	NO	YES
OBD_UDS_7E8	EvaporativeSystemMonitoring	YES	YES
OBD_UDS_7E8	SecondaryAirSystemMonitoring	YES	YES
OBD_UDS_7E8	BoostPressureSystemMonitoring	YES	YES
OBD_UDS_7E8	ParticulateMatterFilterMonitoring	YES	YES
OBD_UDS_7E8	ExhaustGasSensorMonitoring	NO	YES
OBD_UDS_7E8	CVSystemMonitoring	YES	NO
OBD_UDS_7E8	EGRSystemMonitoring	NO	YES

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☐ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

Tools: This field is used to read the data – click **Read Data** for this purpose.

MIL Status: The icon of the MIL (Malfunction Indicator Lamp) is shown in this field.

Trouble Codes: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Parameter	Name of the parameter
Value	Physical value of the parameter

Monitoring Tests: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Monitor	Monitor
Supported	Is the monitor supported?
Completed	Was the monitor completed?

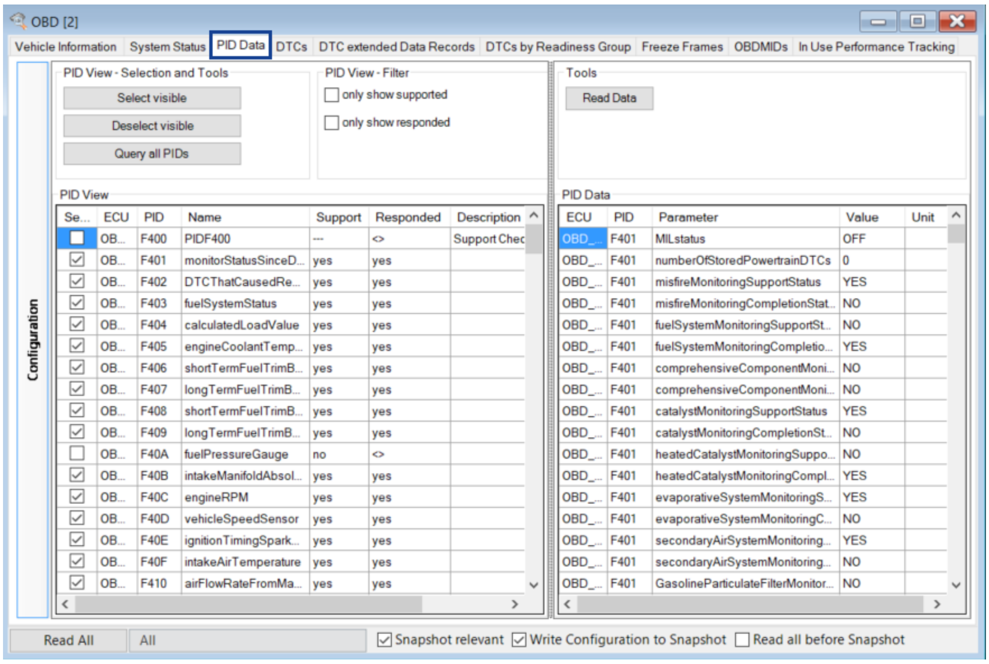
Note

If an ECU identifies itself via Service $22, PID $F501 as a diesel ECU, the monitors relevant for diesel ECUs are displayed, otherwise the monitors for gasoline ECUs.

OBDonUDS "**PID Data**" Tab

Information from service $22 for Current Powertrain Diagnostic Data PIDs $F400 - $F5FF is contained in this tab.

PIDs (parameter identifiers) are the indentifiers for the information supported by the engine ECU.



PID View - Filter

□ only show supported

□ only show responded

PID View

Se...	ECU	PID	Name	Support	Responded	Description
□	OB...	F400	PIDF400	---	<>	Support Chec
☑	OB...	F401	monitorStatusSinceD...	yes	yes	
☑	OB...	F402	DTCThatCausedRe...	yes	yes	
☑	OB...	F403	fuelSystemStatus	yes	yes	
☑	OB...	F404	calculatedLoadValue	yes	yes	
☑	OB...	F405	engineCoolantTemp...	yes	yes	
☑	OB...	F406	shortTermFuelTrimB...	yes	yes	
☑	OB...	F407	longTermFuelTrimB...	yes	yes	
☑	OB...	F408	shortTermFuelTrimB...	yes	yes	
☑	OB...	F409	longTermFuelTrimB...	yes	yes	
□	OB...	F40A	fuelPressureGauge	no	<>	
☑	OB...	F40B	intakeManifoldAbsol...	yes	yes	
☑	OB...	F40C	engineRPM	yes	yes	
☑	OB...	F40D	vehicleSpeedSensor	yes	yes	
☑	OB...	F40E	ignitionTimingSpark...	yes	yes	
☑	OB...	F40F	intakeAirTemperature	yes	yes	
☑	OB...	F410	airFlowRateFromMa...	yes	yes	

PID Data

ECU	PID	Parameter	Value	Unit
OBD_...	F401	MILstatus	OFF	
OBD_...	F401	numberOfStoredPowertrainDTCs	0	
OBD_...	F401	misfireMonitoringSupportStatus	YES	
OBD_...	F401	misfireMonitoringCompletionStat...	NO	
OBD_...	F401	fuelSystemMonitoringSupportSt...	NO	
OBD_...	F401	fuelSystemMonitoringCompletio...	YES	
OBD_...	F401	comprehensiveComponentMoni...	NO	
OBD_...	F401	comprehensiveComponentMoni...	NO	
OBD_...	F401	catalystMonitoringSupportStatus	YES	
OBD_...	F401	catalystMonitoringCompletionSt...	NO	
OBD_...	F401	heatedCatalystMonitoringSuppo...	NO	
OBD_...	F401	heatedCatalystMonitoringCompl...	YES	
OBD_...	F401	evaporativeSystemMonitoringS...	YES	
OBD_...	F401	evaporativeSystemMonitoringC...	NO	
OBD_...	F401	secondaryAirSystemMonitoring...	YES	
OBD_...	F401	secondaryAirSystemMonitoring...	NO	
OBD_...	F401	GasolineParticulateFilterMonitor...	NO	

☑ Snapshot relevant ☑ Write Configuration to Snapshot □ Read all before Snapshot

To select the PIDs to be queried, click **Configuration** – the window then shows additional fields (shown below with an *).

The individual fields of the GUI contain the following functions and information:

***PID View - Selection and Tools**: Using these buttons, you can make a kind of global selection of the PIDs to be queried in the "PID View" list.

- **Select visible**

  Selects all PIDs visible in the "PID View" list (see "OBDonUDS (SAE J1979-2 / SAE J1979-3)" on page 43)

- **Deselect visible**

  The selection of visible PIDs is undone

- **Query all PIDs**

  Each individual PID is addressed and then checked to see if a response is returned

***PID View - Filter**: Uses filter criteria with regard to the display in the "PID View" list. The following options are available:

- **only show supported**

  If this option is selected, only the PIDs supported by the ECU are made available for selection in the "PID View" list.

- **only show responded**

  If this option is selected, only the PIDs answered by the ECU after **Query all PIDs** (see above) are made available for selection in the "PID View" list.

Tools: This field is used to read the data – click **Read Data** for this purpose.

***PID View**: This table displays the selected PIDs. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
Select	Selection of the PID
ECU	Name of the logical link (ECU) from hardware configuration
PID	PID
Name	Explicit name of the PID
Support	Is this PID supported? (queried from ECU)
Responded	Was the query of this PID answered (via **Query all PIDs**)?
Description	Explanatory text (if in the database)

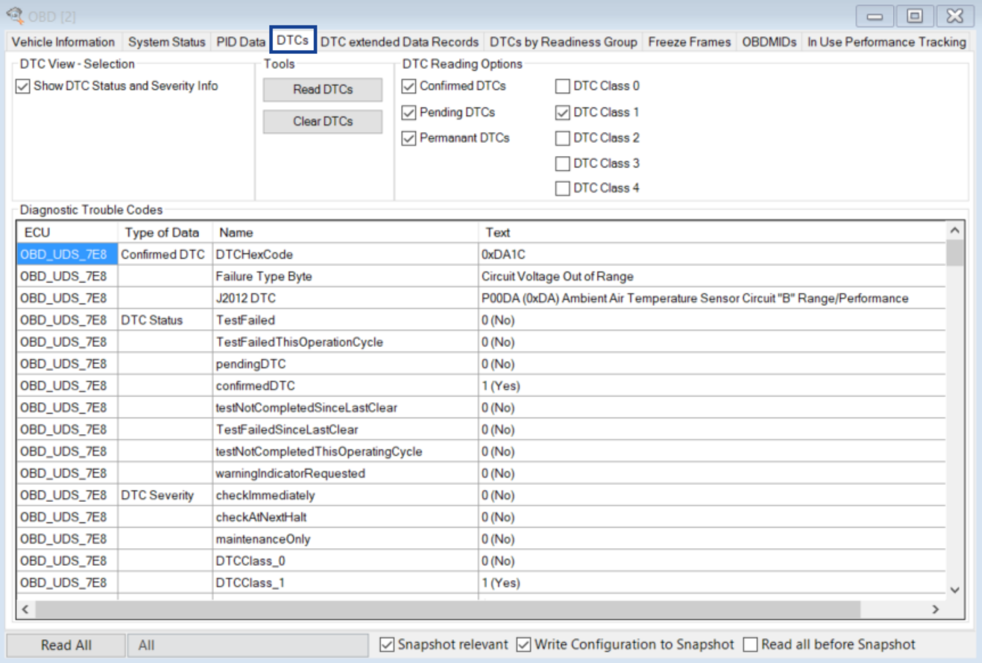
PID Data

This table displays the results of the query/queries. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
PID	PID
Parameter	Explicit name as one PID can consist of several pieces of information
Value	Physical value of the parameter
Unit	Unit of PID (if available)

OBDonUDS "**DTCs**" Tab

Information from services $19, subfunctions $42 and $55 with Emission-Related Diagnostic Trouble Codes with Confirmed, Pending and Permanent Status is contained in this tab.



DTC View - Selection

☑ Show DTC Status and Severity Info

DTC Reading Options

☑ Confirmed DTCs
☑ Pending DTCs
☑ Permanant DTCs

☐ DTC Class 0
☑ DTC Class 1
☐ DTC Class 2
☐ DTC Class 3
☐ DTC Class 4

Diagnostic Trouble Codes

ECU	Type of Data	Name	Text
OBD_UDS_7E8	Confirmed DTC	DTCHexCode	0xDA1C
OBD_UDS_7E8		Failure Type Byte	Circuit Voltage Out of Range
OBD_UDS_7E8		J2012 DTC	P00DA (0xDA) Ambient Air Temperature Sensor Circuit "B" Range/Performance
OBD_UDS_7E8	DTC Status	TestFailed	0 (No)
OBD_UDS_7E8		TestFailedThisOperationCycle	0 (No)
OBD_UDS_7E8		pendingDTC	0 (No)
OBD_UDS_7E8		confirmedDTC	1 (Yes)
OBD_UDS_7E8		testNotCompletedSinceLastClear	0 (No)
OBD_UDS_7E8		TestFailedSinceLastClear	0 (No)
OBD_UDS_7E8		testNotCompletedThisOperatingCycle	0 (No)
OBD_UDS_7E8		warningIndicatorRequested	0 (No)
OBD_UDS_7E8	DTC Severity	checkImmediately	0 (No)
OBD_UDS_7E8		checkAtNextHalt	0 (No)
OBD_UDS_7E8		maintenanceOnly	0 (No)
OBD_UDS_7E8		DTCClass_0	0 (No)
OBD_UDS_7E8		DTCClass_1	1 (Yes)

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☐ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

DTC View - Selection:

- **Show DTC Status and Severity Info**

  If selected, all rows with DTC Status and Severity information are displayed. Otherwise, this information is hidden.

Tools:

- **Read DTCs**

  If selected, the Diagnostic Trouble Codes are read as configured by the DTC Reading Options.

- **Clear DTCs**

  If selected, all OBD DTCs in the vehicle will be cleared.

DTC Reading Options:

Select if confirmed, pending, and/or permanent DTCs shall be read.:

- **Confirmed DTCs**
- **Pending DTCs**
- **Permanent DTCs**

Select which DTC class shall be read. Class 1 is the default for OBDonUDS:

- **DTC Class 0**
- **DTC Class 1**

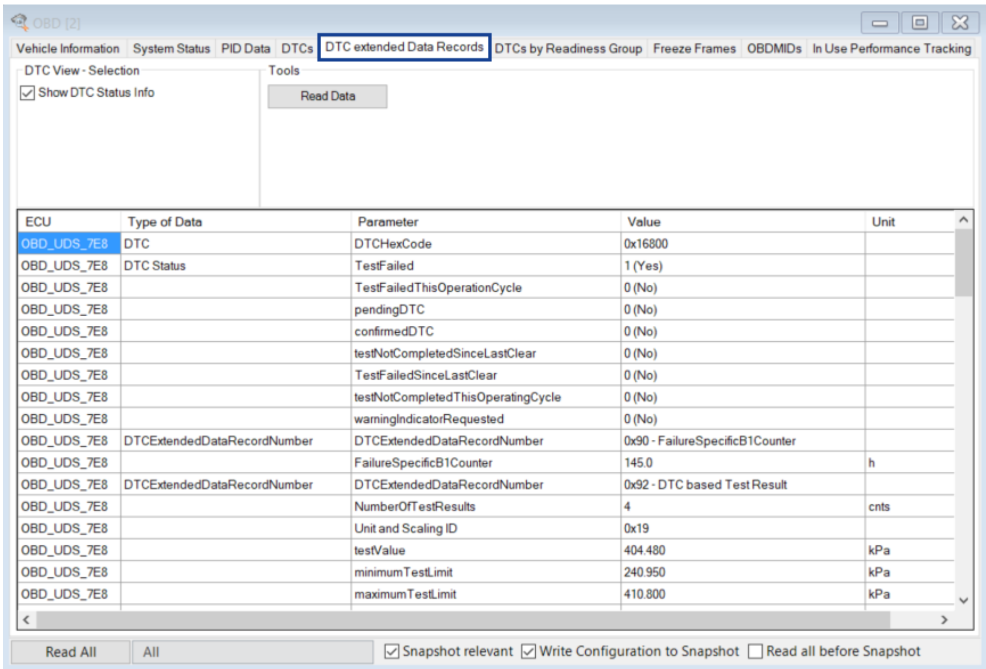
- **DTC Class 2**
- **DTC Class 3**
- **DTC Class 4**

The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Type of Data	Type of information that is displayed in this row and succeeding rows below
Name	Name of the parameter that is displayed
Text	Value of the parameter

OBDonUDS "**DTC extended Data Records**" Tab

Information from services $19, subfunctions $1A and $06 with DTCExtendedDataRecord information for all supported DTCExtendedRecords is contained in this tab.



☑ Show DTC Status Info

ECU	Type of Data	Parameter	Value	Unit
OBD_UDS_7E8	DTC	DTCHexCode	0x16800	
OBD_UDS_7E8	DTC Status	TestFailed	1 (Yes)	
OBD_UDS_7E8		TestFailedThisOperationCycle	0 (No)	
OBD_UDS_7E8		pendingDTC	0 (No)	
OBD_UDS_7E8		confirmedDTC	0 (No)	
OBD_UDS_7E8		testNotCompletedSinceLastClear	0 (No)	
OBD_UDS_7E8		TestFailedSinceLastClear	0 (No)	
OBD_UDS_7E8		testNotCompletedThisOperatingCycle	0 (No)	
OBD_UDS_7E8		warningIndicatorRequested	0 (No)	
OBD_UDS_7E8	DTCExtendedDataRecordNumber	DTCExtendedDataRecordNumber	0x90 - FailureSpecificB1Counter	
OBD_UDS_7E8		FailureSpecificB1Counter	145.0	h
OBD_UDS_7E8	DTCExtendedDataRecordNumber	DTCExtendedDataRecordNumber	0x92 - DTC based Test Result	
OBD_UDS_7E8		NumberOfTestResults	4	cnts
OBD_UDS_7E8		Unit and Scaling ID	0x19	
OBD_UDS_7E8		testValue	404.480	kPa
OBD_UDS_7E8		minimumTestLimit	240.950	kPa
OBD_UDS_7E8		maximumTestLimit	410.800	kPa

☑ Snapshot relevant ☑ Write Configuration to Snapshot □ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

DTC View - Selection

- **Show DTC Status Info**

  If selected, all rows with DTC Status and Severity information are displayed. Otherwise this information is hidden.

Tools

- **Read Data**

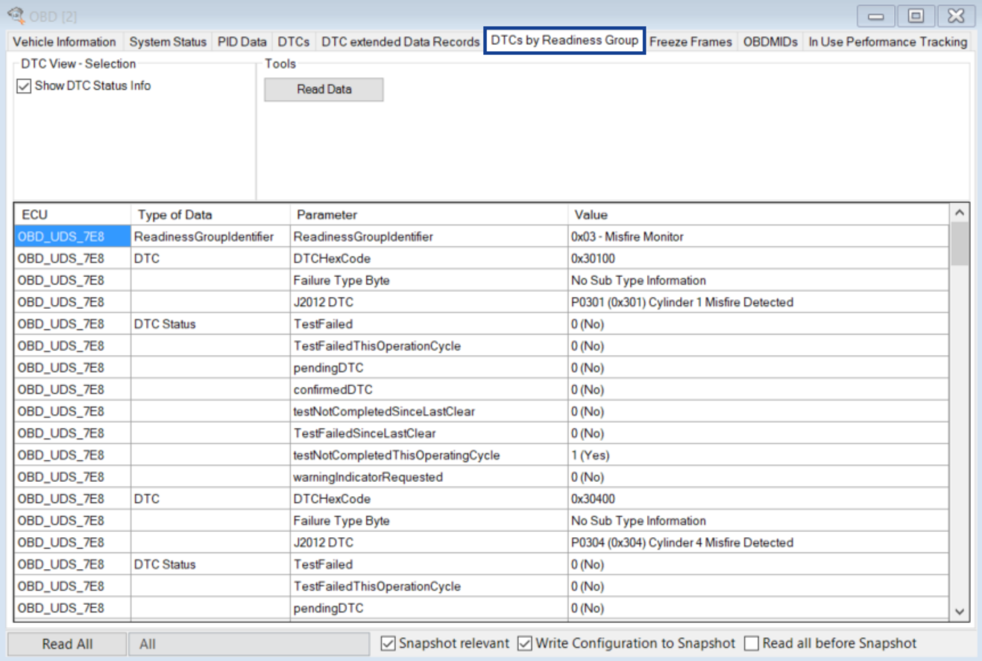
  To read all supported extended data records, click **Read Data**.

The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Type of Data	Type of information that is displayed in this row and succeeding rows below
Parameter	Name of the parameter
Value	Value of the parameter
Unit	Unit of the parameter (if available)

OBDonUDS "**DTCs by Readiness Group**" Tab

Information from services $19, subfunction $56 with DTCs for all ReadinessGroups is contained in this tab.



ECU	Type of Data	Parameter	Value
OBD_UDS_7E8	ReadinessGroupIdentifier	ReadinessGroupIdentifier	0x03 - Misfire Monitor
OBD_UDS_7E8	DTC	DTCHexCode	0x30100
OBD_UDS_7E8		Failure Type Byte	No Sub Type Information
OBD_UDS_7E8		J2012 DTC	P0301 (0x301) Cylinder 1 Misfire Detected
OBD_UDS_7E8	DTC Status	TestFailed	0 (No)
OBD_UDS_7E8		TestFailedThisOperationCycle	0 (No)
OBD_UDS_7E8		pendingDTC	0 (No)
OBD_UDS_7E8		confirmedDTC	0 (No)
OBD_UDS_7E8		testNotCompletedSinceLastClear	0 (No)
OBD_UDS_7E8		TestFailedSinceLastClear	0 (No)
OBD_UDS_7E8		testNotCompletedThisOperatingCycle	1 (Yes)
OBD_UDS_7E8		warningIndicatorRequested	0 (No)
OBD_UDS_7E8	DTC	DTCHexCode	0x30400
OBD_UDS_7E8		Failure Type Byte	No Sub Type Information
OBD_UDS_7E8		J2012 DTC	P0304 (0x304) Cylinder 4 Misfire Detected
OBD_UDS_7E8	DTC Status	TestFailed	0 (No)
OBD_UDS_7E8		TestFailedThisOperationCycle	0 (No)
OBD_UDS_7E8		pendingDTC	0 (No)

The individual fields of the GUI contain the following functions and information:

DTC View - Selection

- **Show DTC Status Info**

  If selected, all rows with DTC Status information are displayed. Otherwise, this information is hidden.

Tools

- **Read Data**

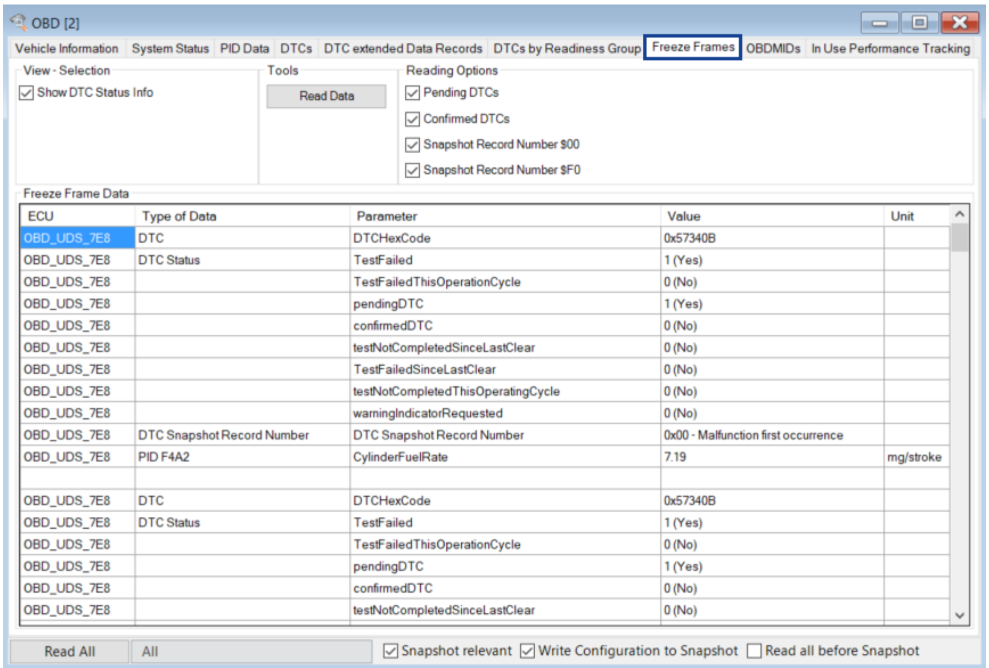
  To read the DTCs by Readiness Group, click **Read Data**.

The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Type of Data	Type of information that is displayed in this row and succeeding rows below
Parameter	Name of the parameter
Value	Value of the parameter

OBDonUDS "**Freeze Frames**" Tab

Information from service $22, subfunction $04 with Powertrain Freeze Frame Data for pending and confirmed DTCs is contained in this tab.



☑ Show DTC Status Info

☑ Pending DTCs
☑ Confirmed DTCs
☑ Snapshot Record Number $00
☑ Snapshot Record Number $F0

ECU	Type of Data	Parameter	Value	Unit
OBD_UDS_7E8	DTC	DTCHexCode	0x57340B	
OBD_UDS_7E8	DTC Status	TestFailed	1 (Yes)	
OBD_UDS_7E8		TestFailedThisOperationCycle	0 (No)	
OBD_UDS_7E8		pendingDTC	1 (Yes)	
OBD_UDS_7E8		confirmedDTC	0 (No)	
OBD_UDS_7E8		testNotCompletedSinceLastClear	0 (No)	
OBD_UDS_7E8		TestFailedSinceLastClear	0 (No)	
OBD_UDS_7E8		testNotCompletedThisOperatingCycle	0 (No)	
OBD_UDS_7E8		warningIndicatorRequested	0 (No)	
OBD_UDS_7E8	DTC Snapshot Record Number	DTC Snapshot Record Number	0x00 - Malfunction first occurrence	
OBD_UDS_7E8	PID F4A2	CylinderFuelRate	7.19	mg/stroke
OBD_UDS_7E8	DTC	DTCHexCode	0x57340B	
OBD_UDS_7E8	DTC Status	TestFailed	1 (Yes)	
OBD_UDS_7E8		TestFailedThisOperationCycle	0 (No)	
OBD_UDS_7E8		pendingDTC	1 (Yes)	
OBD_UDS_7E8		confirmedDTC	0 (No)	
OBD_UDS_7E8		testNotCompletedSinceLastClear	0 (No)	

☑ Snapshot relevant ☑ Write Configuration to Snapshot □ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

View - Selection

- **Show DTC Status Info**

Tools

- **Read Data**

  to read the Freeze Frame data as configured with the reading options - click **Read Data**.

Reading Options

Select if Freeze Frame data shall be read for pending and/or confirmed DTCs:

- **Pending DTCs**
- **Confirmed DTCs**

Select if Freeze Frame snapshot record number $00 and/or $F0 shall be read:

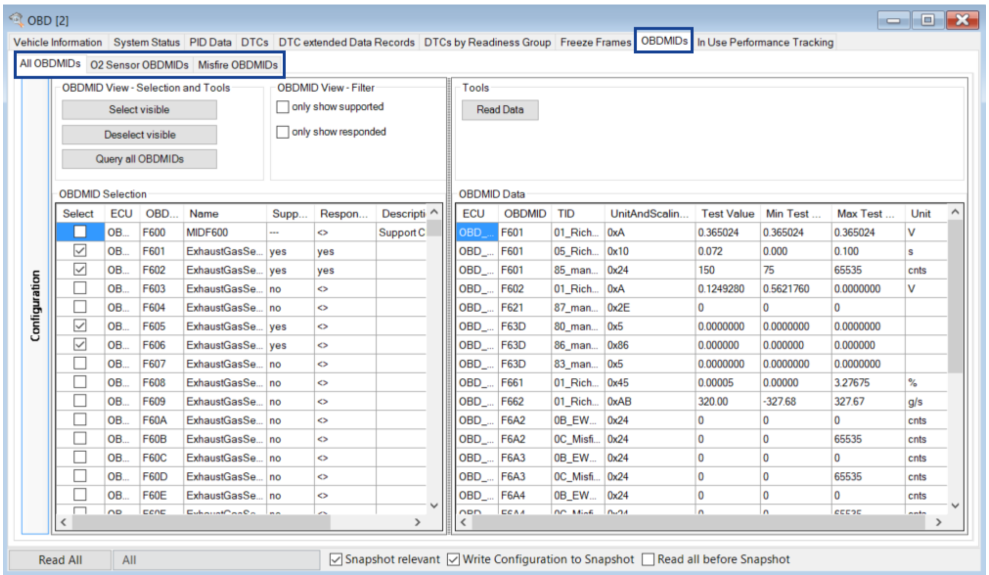
- **Snapshot Record Number $100**
- **Snapshot Record Number $F0**

The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Type of Data	Type of information that is displayed in this row and succeeding rows below
Parameter	Name of the parameter
Value	Value of the parameter
Unit	Unit of the parameter (if available)

OBDonUDS "**OBDMIDS**" Tab

Information from service $22 for On-Board Monitoring Test Results for Specific Monitored Systems MIDs $F600 - $F6FF is contained in this tab.



□ only show supported

□ only show responded

Select	ECU	OBD...	Name	Supp...	Respon...	Descripti
□	OB...	F600	MIDF600	---	<>	Support C
☑	OB...	F601	ExhaustGasSe...	yes	yes	
☑	OB...	F602	ExhaustGasSe...	yes	yes	
□	OB...	F603	ExhaustGasSe...	no	<>	
□	OB...	F604	ExhaustGasSe...	no	<>	
☑	OB...	F605	ExhaustGasSe...	yes	<>	
☑	OB...	F606	ExhaustGasSe...	yes	<>	
□	OB...	F607	ExhaustGasSe...	no	<>	
□	OB...	F608	ExhaustGasSe...	no	<>	
□	OB...	F609	ExhaustGasSe...	no	<>	
□	OB...	F60A	ExhaustGasSe...	no	<>	
□	OB...	F60B	ExhaustGasSe...	no	<>	
□	OB...	F60C	ExhaustGasSe...	no	<>	
□	OB...	F60D	ExhaustGasSe...	no	<>	
□	OB...	F60E	ExhaustGasSe...	no	<>	
□	OB...	F60F	ExhaustGasSe...	no	<>	

ECU	OBDMID	TID	UnitAndScalin...	Test Value	Min Test ...	Max Test ...	Unit
OBD_...	F601	01_Rich...	0xA	0.365024	0.365024	0.365024	V
OBD_...	F601	05_Rich...	0x10	0.072	0.000	0.100	s
OBD_...	F601	85_man...	0x24	150	75	65535	cnts
OBD_...	F602	01_Rich...	0xA	0.1249280	0.5621760	0.0000000	V
OBD_...	F621	87_man...	0x2E	0	0	0	
OBD_...	F63D	80_man...	0x5	0.0000000	0.0000000	0.0000000	
OBD_...	F63D	86_man...	0x86	0.000000	0.000000	0.000000	
OBD_...	F63D	83_man...	0x5	0.0000000	0.0000000	0.0000000	
OBD_...	F661	01_Rich...	0x45	0.00005	0.00000	3.27675	%
OBD_...	F662	01_Rich...	0xAB	320.00	-327.68	327.67	g/s
OBD_...	F6A2	0B_EW...	0x24	0	0	0	cnts
OBD_...	F6A2	0C_Misfi...	0x24	0	0	65535	cnts
OBD_...	F6A3	0B_EW...	0x24	0	0	0	cnts
OBD_...	F6A3	0C_Misfi...	0x24	0	0	65535	cnts
OBD_...	F6A4	0B_EW...	0x24	0	0	0	cnts
OBD_...	F6A4	0C_Misfi...	0x24	0	0	65535	cnts

☑ Snapshot relevant ☑ Write Configuration to Snapshot □ Read all before Snapshot

For reasons of clarity, this window is divided into several tabs:

- **All OBDMIDs tab**

  All OBDMIDs are displayed in this tab

- **O2 Sensor OBDMIDs** tab

  All OBDMIDs connected with the O2 sensor monitor are displayed in this tab
- **Misfire OBDMIDs** tab

  All OBDMIDs connected with the misfire monitor are displayed in this tab

To select the OBDMIDs, click **Configuration** – the window then shows additional fields (shown below with an *).

The individual fields of the GUI contain the following functions and information:

***OBDMID View - Selection Tools**: Using these buttons, you can make a kind of global selection of the OBDMIDs to be queried in the "OBDMID Selection" list.

- **Select visible**

  Selects all OBDMIDs visible in the "OBDMID Selection" list (see "OBDonUDS (SAE J1979-2 / SAE J1979-3)" on page 43).
- **Deselect visible**

  The selection of visible OBDMIDs is undone
- **Query all OBDMIDs**

  Each individual OBDMID is addressed and then checked to see if a response is returned

***OBDMID View - Filter**: Uses filter criteria with regard to the display in the "OBDMID Selection" list. The following options are available:

- **only show supported**

  If this option is selected, only those OBDMIDs supported by the ECU are made available for selection in the "OBDMID Selection" list.
- **only show responded**

  If this option is selected, only the OBDMIDs answered by the ECU after **Query all OBDMIDs** (see above) are made available for selection in the "OBDMID Selection" list.

Tools: This field is used to read the data – click **Read Data** for this purpose.

***OBDMID Selection**: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
Select	Selection of the OBDMID
ECU	Name of the logical link (ECU) from hardware configuration
OBDMID	On-Board Diagnostic Monitor ID
Name	Explicit name of the ODBMID

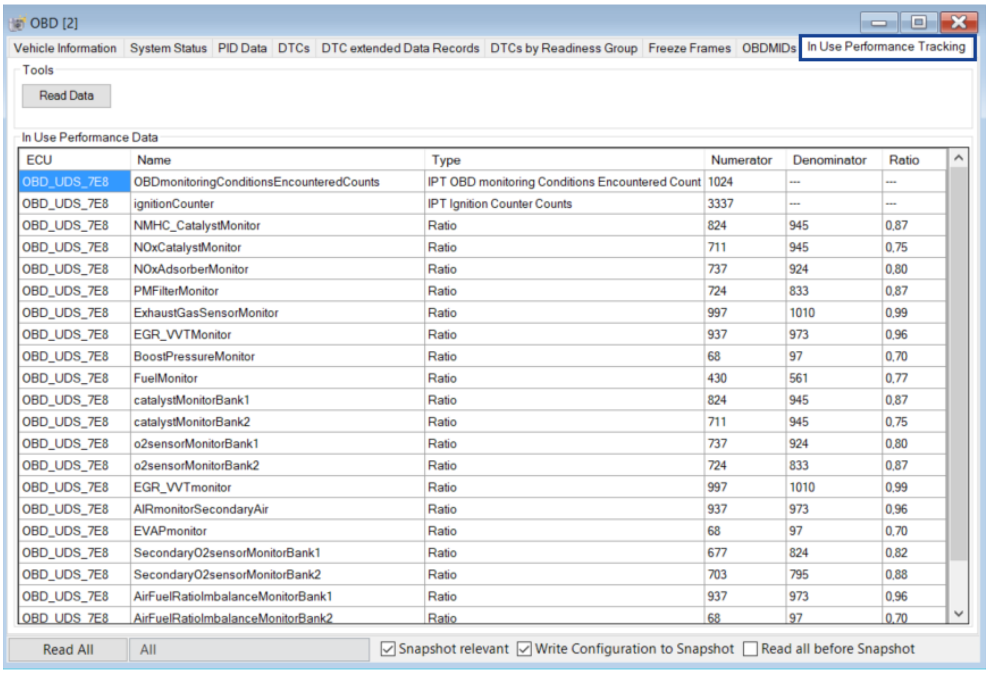
Column	Meaning
Support	Is this OBDMID supported?
Responded	Was the query of this OBDMID answered (via **Query all OBDMIDs**)?
Description	Explanatory text (if in the database)

OBDMID Data: The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
OBDMID	On-Board Diagnostic Monitor ID
TID	Test ID of service 08
UnitAndScalingID	Unit and Scaling ID (1 Byte)
Test Value	Value read from the ECU
Min Test Limit	Minimum test limit
Max Test Limit	Maximum test limit
Unit	Unit of the physical value

OBDonUDS "**In Use Performance Tracking**" Tab

In this tab the data of the In Use Performance Tracking of service $22 is displayed for ITIDs $F808, $F80B.



ECU	Name	Type	Numerator	Denominator	Ratio
OBD_UDS_7E8	OBDmonitoringConditionsEncounteredCounts	IPT OBD monitoring Conditions Encountered Count	1024	---	---
OBD_UDS_7E8	ignitionCounter	IPT Ignition Counter Counts	3337	---	---
OBD_UDS_7E8	NMHC_CatalystMonitor	Ratio	824	945	0,87
OBD_UDS_7E8	NOxCatalystMonitor	Ratio	711	945	0,75
OBD_UDS_7E8	NOxAdsorberMonitor	Ratio	737	924	0,80
OBD_UDS_7E8	PMFilterMonitor	Ratio	724	833	0,87
OBD_UDS_7E8	ExhaustGasSensorMonitor	Ratio	997	1010	0,99
OBD_UDS_7E8	EGR_VVTMonitor	Ratio	937	973	0,96
OBD_UDS_7E8	BoostPressureMonitor	Ratio	68	97	0,70
OBD_UDS_7E8	FuelMonitor	Ratio	430	561	0,77
OBD_UDS_7E8	catalystMonitorBank1	Ratio	824	945	0,87
OBD_UDS_7E8	catalystMonitorBank2	Ratio	711	945	0,75
OBD_UDS_7E8	o2sensorMonitorBank1	Ratio	737	924	0,80
OBD_UDS_7E8	o2sensorMonitorBank2	Ratio	724	833	0,87
OBD_UDS_7E8	EGR_VVTmonitor	Ratio	997	1010	0,99
OBD_UDS_7E8	AIRmonitorSecondaryAir	Ratio	937	973	0,96
OBD_UDS_7E8	EVAPmonitor	Ratio	68	97	0,70
OBD_UDS_7E8	SecondaryO2sensorMonitorBank1	Ratio	677	824	0,82
OBD_UDS_7E8	SecondaryO2sensorMonitorBank2	Ratio	703	795	0,88
OBD_UDS_7E8	AirFuelRatioImbalanceMonitorBank1	Ratio	937	973	0,96
OBD_UDS_7E8	AirFuelRatioImbalanceMonitorBank2	Ratio	68	97	0,70

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☐ Read all before Snapshot

The individual fields of the GUI contain the following functions and information:

Tools: This field is used to read the data – click **Read Data** for this purpose.

In Use Performance Data: This table displays the queried information of service $22 for the ITIDs $F808 and $F80B. The meaning of each individual entry in the list is described in the following table:

Column	Meaning
ECU	Name of the logical link (ECU) from hardware configuration
Name	Name of the parameter
Type	"General Denominator", "Counter", "Denominator" or "Numerator"
Numerator	Tracks the number of times that the vehicle has been operated in the specified conditions. These conditions are specified for each monitored component or system.
Denominator	Tracks the number of times that all conditions necessary for a specific monitor to detect a malfunction have been encountered
Ratio	Ratio of the values above

4.2 Data Logging Configuration

The "Data Logging Configuration" (**ODX** > **Data Logging Configuration**) is used to determine how snapshot data is saved. For more details on the snapshot function, refer to the sections "User Views" on page 19 and "Snapshots" on page 60.

> ⓘ **Note**
>
> Please note that only the data of user views configured accordingly are recorded in the snapshots. The snapshot icon at the extreme left of the title bar of the user views indicates whether the data of the relevant user view is recorded in the snapshot.

The settings which you specified can be exported to a file to be used later. Settings exported previously can be reimported from files exported previously.

The "DataLoggingConfig" window contains two tabs:

- **File**

  settings on file name and storage location of the snapshot
- **Header**

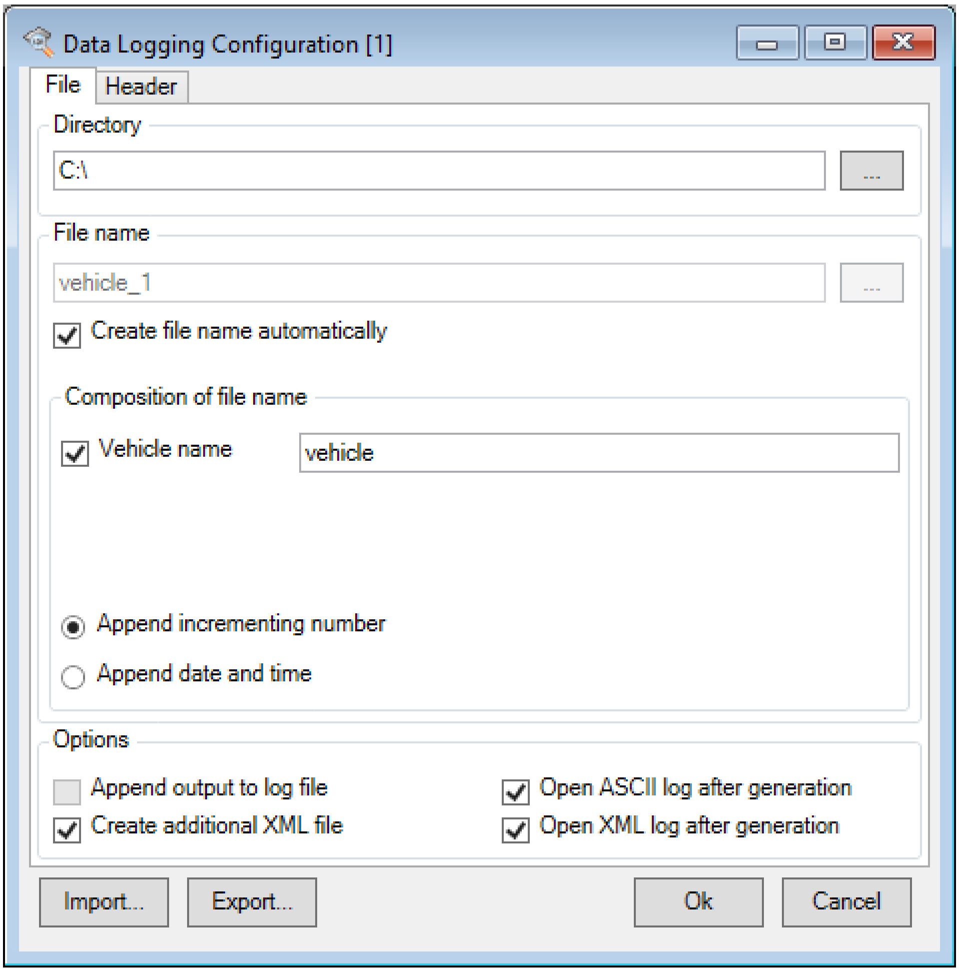
  meta information added to the snapshot data

To configure Data Logging

1. Select **ODX** > **Data Logging** > **Configuration**.
2. Set the configuration to correspond to your requirements. The following sections explain the significance of the input fields and options.
3. Click **OK**.

"File" Tab

This is where you specify in which directory and with which file name the snapshot data is to be saved.



Directory: C:\

File name: vehicle_1

☑ Create file name automatically

☑ Vehicle name: vehicle

● Append incrementing number

○ Append date and time

□ Append output to log file

☑ Create additional XML file

☑ Open ASCII log after generation

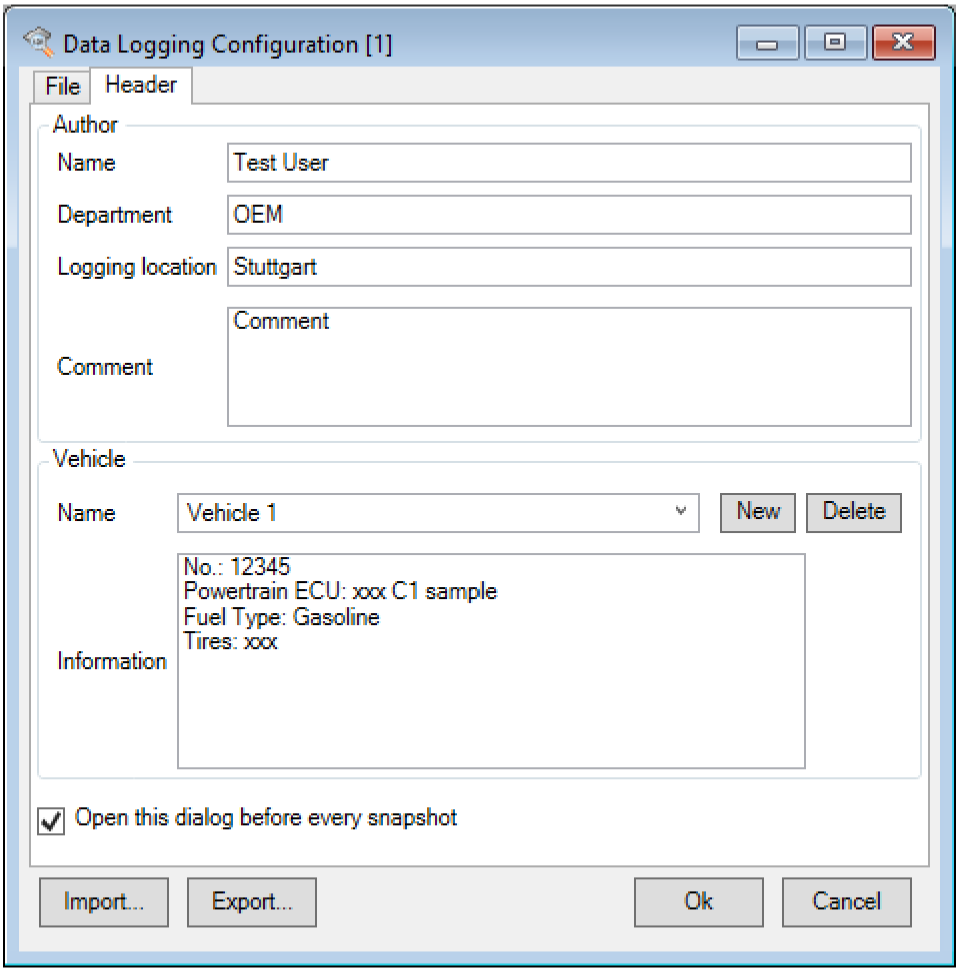
☑ Open XML log after generation

Input Field or Option	Meaning
Directory	Directory in which the snapshot files are saved
File name	File name for the snapshot files. If you activated the option "Create file name automatically", you cannot enter a file name here.
Create file name automatically	The file name is created automatically
Composition of file name	This is where you specify the individual parts and composition of the file name
Vehicle name	Activate this option to include the vehicle name from the input field in the file name created automatically.
Append incrementing number	An automatically incrementing number is added to the file name.

Input Field or Option	Meaning
Append date and time	The current date and the current time (format: "yyyymmdd_hhmmss") are added to the file name
Append output to log file	The snapshot data is appended to the existing data in the log file. This option is only available when the file name is generated manually.
Open ASCII log after generation	The snapshot file is opened automatically
Create additional XML file	An additional XML file is created with the snapshot data
Open log file after generation	The XML file is opened automatically as HTML in your standard browser.

"Header" Tab

This is where you specify which meta information should be added to the snapshot data.



Data Logging Configuration [1]

File | Header

Author

Name: Test User

Department: OEM

Logging location: Stuttgart

Comment: Comment

Vehicle

Name: Vehicle 1

Information:
No.: 12345
Powertrain ECU: xxx C1 sample
Fuel Type: Gasoline
Tires: xxx

☑ Open this dialog before every snapshot

Input Field or Option	Meaning
Name	Your name
Department	Your department within your company
Logging location	The location of the recorded data
Comment	Notes on data recording
Vehicle	List with vehicle names
Name	Name of vehicle
Information	Notes on vehicle
Open this dialog before every snapshot	Activate this option to open this dialog box before the snapshot data is saved to adapt the meta information for the snapshot.

Exporting and Importing Configurations

All configuration data can be exported as XML files and then imported again. Proceed as follows:

To export Data Logging Configuration

1. Select **ODX** > **Data Logging Configuration...**.
2. The "DataLoggingConfig" window opens.
3. Click **Export...**.

   A file selector window opens.
4. Enter the required file name and click **Save**.

To import Data Logging Configuration

1. Select **ODX** > **Data Logging Configuration...**.
2. The "DataLoggingConfig" window opens.
3. Click **Import...**.

   A file selector window opens.
4. Enter the name of the file to be imported and click **Open**.

4.3 Snapshots

With the snapshot function (**ODX** > **Snapshot**), it is possible to save the current data of ODX-LINK windows into a file.

Note

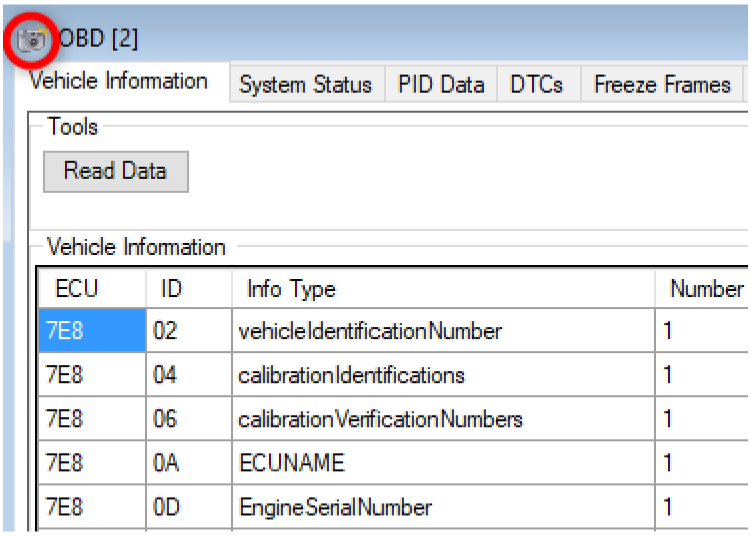
A snapshot saves the diagnostic data currently stored in all snapshot-relevant ODX-LINK user views. Please ensure that the windows are updated manually before the snapshot function is triggered!

Meta data you can configure is added to the data read from the ECU (see the section "Data Logging Configuration " on page 57).

You define which data is to be recorded in a snapshot (option "Snapshot relevant"). The data of the following "user views" can be recorded in snapshots:

- Diagnostic Services (see "Diagnostic Services" on page 22)
- Service Inspector (see "Service Inspector" on page 25)
- OBD (see "OBD" on page 29)

The Snapshot icon - a small camera - at the left of the title bar of the relevant dialog window indicates whether the data of a user view is recorded in the snapshot.



ECU	ID	Info Type	Number
7E8	02	vehicleIdentificationNumber	1
7E8	04	calibrationIdentifications	1
7E8	06	calibrationVerificationNumbers	1
7E8	0A	ECUNAME	1
7E8	0D	EngineSerialNumber	1

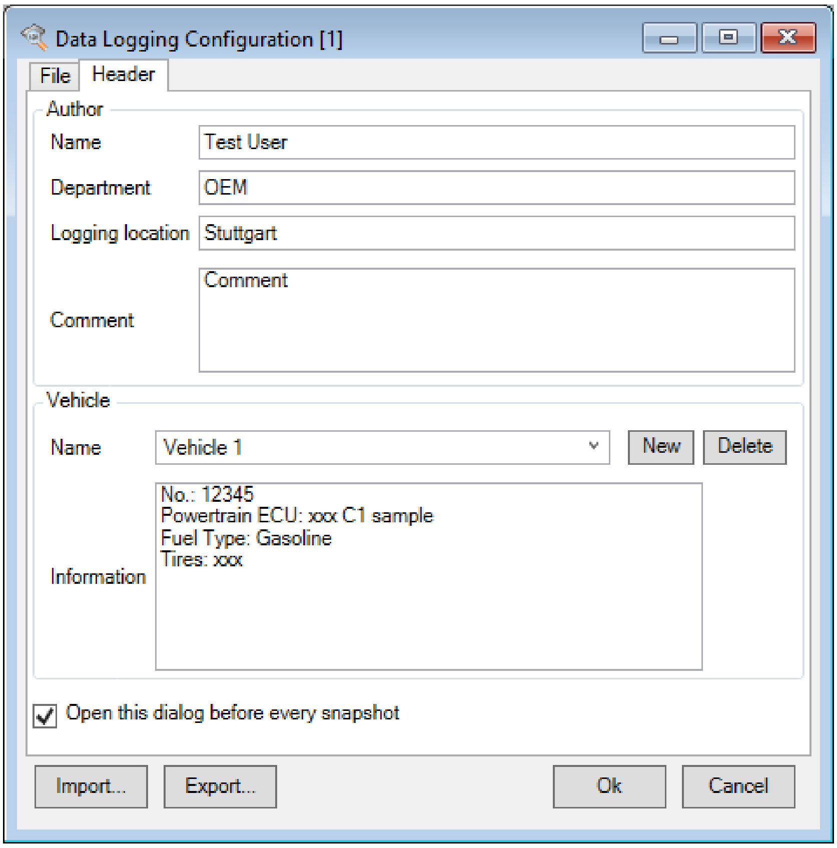
To record a snapshot

1. Click **Snapshot** from the toolbar of the INCA experiment environment.

or

Select **ODX** > **Snapshot**.

If you activated the relevant option in the "Data Logging Configuration", a dialog box for the snapshot meta data is displayed. If you did not activate the option, the data is written immediately.



Author

Name: Test User

Department: OEM

Logging location: Stuttgart

Comment: Comment

Vehicle

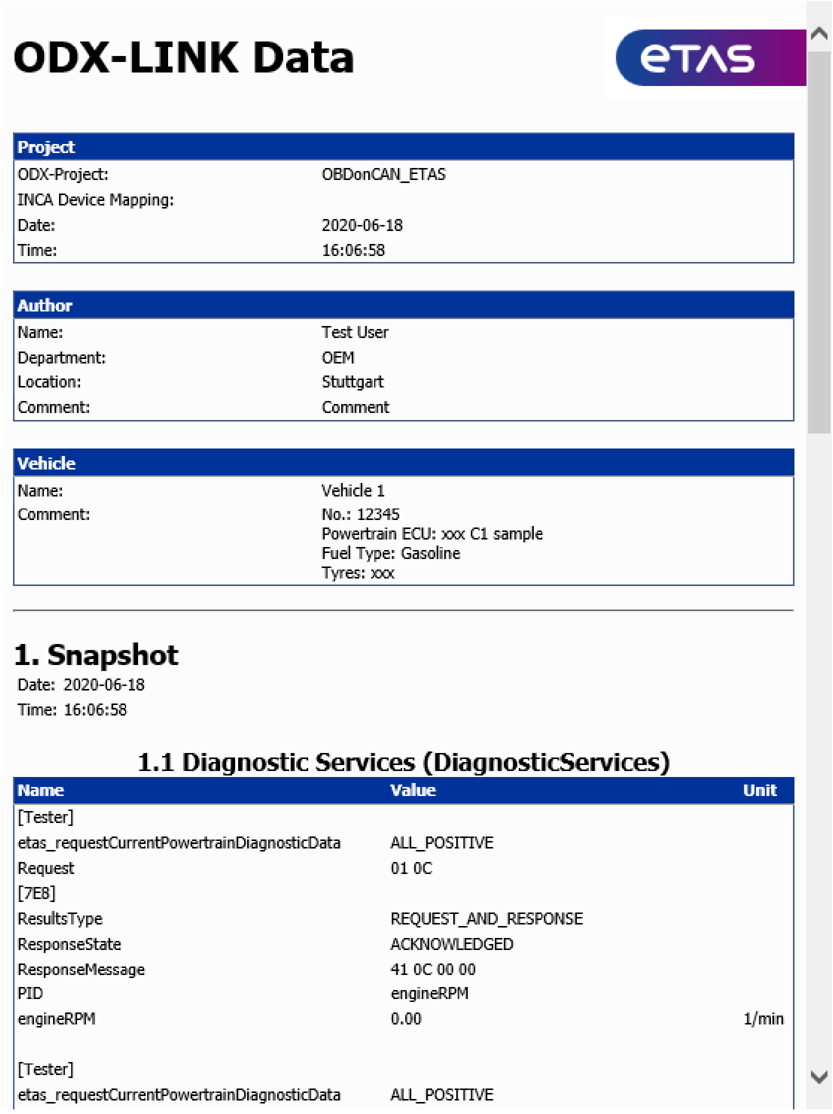
Name: Vehicle 1

Information:
No.: 12345
Powertrain ECU: xxx C1 sample
Fuel Type: Gasoline
Tires: xxx

☑ Open this dialog before every snapshot

2. Enter the relevant meta data. For further details on "Data Logging Configuration", refer to the section "Data Logging Configuration " on page 57.
3. Click **OK**.

   The snapshot data is written. If you have set the "Data Logging Configuration" accordingly, the data written is displayed accordingly.



ODX-LINK Data

Project	
ODX-Project:	OBDonCAN_ETAS
INCA Device Mapping:	
Date:	2020-06-18
Time:	16:06:58

Author	
Name:	Test User
Department:	OEM
Location:	Stuttgart
Comment:	Comment

Vehicle	
Name:	Vehicle 1
Comment:	No.: 12345 Powertrain ECU: xxx C1 sample Fuel Type: Gasoline Tyres: xxx

1. Snapshot

Date: 2020-06-18

Time: 16:06:58

1.1 Diagnostic Services (DiagnosticServices)

Name	Value	Unit
[Tester]		
etas_requestCurrentPowertrainDiagnosticData	ALL_POSITIVE	
Request	01 0C	
[7E8]		
ResultsType	REQUEST_AND_RESPONSE	
ResponseState	ACKNOWLEDGED	
ResponseMessage	41 0C 00 00	
PID	engineRPM	
engineRPM	0.00	1/min
[Tester]		
etas_requestCurrentPowertrainDiagnosticData	ALL_POSITIVE	

4.4 Diagnostic Signals in the INCA Variable Selection

The diagnostic data that can be read about the ODX project at the diagnostic interface of an ECU can also be measured and recorded in a measure file with INCA and ODX-LINK. The prerequisite for this is the definition of diagnostic signals in the ODX project.

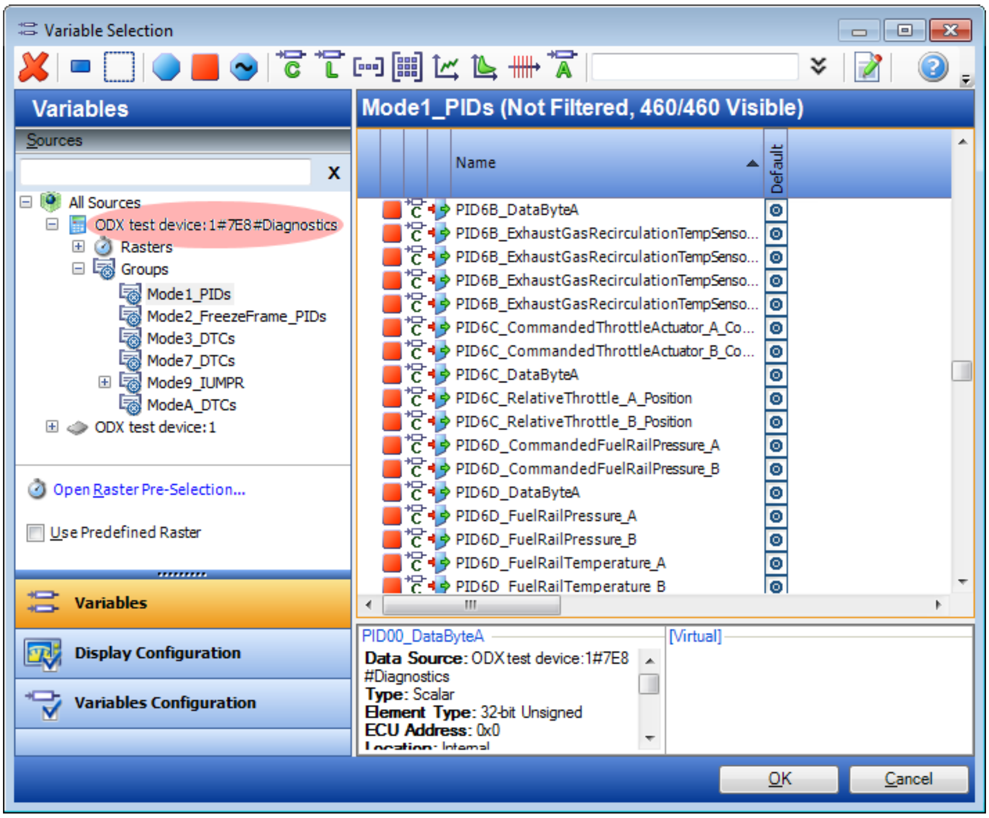
The diagnostic signals of the ODX project are described in the diagnostic signal list (DSL) – the signals defined there can be used in the same way as standard measure variables in INCA's "Variable Selection" window.

If users want to work with their own ODX data, it is possible to generate such signals in the INCA experiment and add them to the ODX project. This function is available for the user views "Diagnostic Services" and "Service Inspector" (see "Adding Diagnostic Signals" on page 65).

When you install ODX-LINK, the ODX projects "`OBDonCAN_ETAS_SAEJ1979<Version>.pdx`" and "`OBDonUDS_ETAS_SAEJ1979<Version>.pdx`" already contains one DSL file. When using this ODX project, the "Variable Selection" window in INCA features an additional diagnostic device with the diagnostic signals of the `OBDSignalList.dsl` file contained.

The diagnostic device is named in accordance with the following conventions:

<Device name>#<Logical Link>#Diagnostics


Variable Selection
Variables
Sources
X
All Sources
ODX test device:1#7E8#Diagnostics
Rasters
Groups
Mode1_PIDs
Mode2_FreezeFrame_PIDs
Mode3_DTCs
Mode7_DTCs
Mode9_IUMPR
ModeA_DTCs
ODX test device:1
Open Raster Pre-Selection...
Use Predefined Raster
Variables
Display Configuration
Variables Configuration
Mode1_PIDs (Not Filtered, 460/460 Visible)
Name
Default
PID6B_DataByteA
PID6B_ExhaustGasRecirculationTempSenso...
PID6B_ExhaustGasRecirculationTempSenso...
PID6B_ExhaustGasRecirculationTempSenso...
PID6B_ExhaustGasRecirculationTempSenso...
PID6C_CommandedThrottleActuator_A_Co...
PID6C_CommandedThrottleActuator_B_Co...
PID6C_DataByteA
PID6C_RelativeThrottle_A_Position
PID6C_RelativeThrottle_B_Position
PID6D_CommandedFuelRailPressure_A
PID6D_CommandedFuelRailPressure_B
PID6D_DataByteA
PID6D_FuelRailPressure_A
PID6D_FuelRailPressure_B
PID6D_FuelRailTemperature_A
PID6D_FuelRailTemperature_B
PID00_DataByteA
[Virtual]
OK
Cancel

Data Source: ODX test device:1#7E8 #Diagnostics
Type: Scalar
Element Type: 32-bit Unsigned
ECU Address: 0x0
Location: Internal

4.4.1 Definitions in the Variable Selection Dialog Box

The diagnostic signals in the Variable Selection dialog box are divided into the following signal groups in the `OBDonCAN_ETAS_SAEJ1979<Version>.pdx` project (with the included signal list):

- **Mode1_PIDs**

  In the signal group "Mode1_PIDs", you find information from service $01 (see "OBDonCAN "PID Data" Tab" on page 34), i.e. up-to-date diagnostic data from the powertrain.

- **Mode2_FreezeFrame_PIDs**

  In the signal group "Mode2_FreezeFrame_PIDs", you find information from service $02 (see "OBDonCAN "Freeze Frames" Tab" on page 38).

- **Mode3_DTCs**

In the signal group "Mode3_DTCs", you find signals that are saved and can be deleted again with service $03 (see "OBDonCAN "DTCs" Tab" on page 36).

- **Mode7_DTCs**

  In the signal group "Mode7_DTCs", you find trouble codes discovered during the current or last completed driving cycle (see "OBDonCAN "DTCs" Tab" on page 36).

- **Mode9_IUMPR**

  Mode9_IUMPR_compression

  Mode9_IUMPR_spark

  In the signal group "Mode9_IUMPR", you find signals from In-Use Performance Tracking (see "OBDonCAN "In Use Performance Tracking" Tab" on page 42).

- **ModeA_DTCs**

  In the signal group "ModeA_DTCs", you find signals that have the status "permanent" and cannot be deleted (see "OBDonCAN "DTCs" Tab" on page 36).

Note

In generated signal lists, the functional groups are called "ModeX" (with the ETAS OBD project) or "ServiceXY" (with other ODX projects).

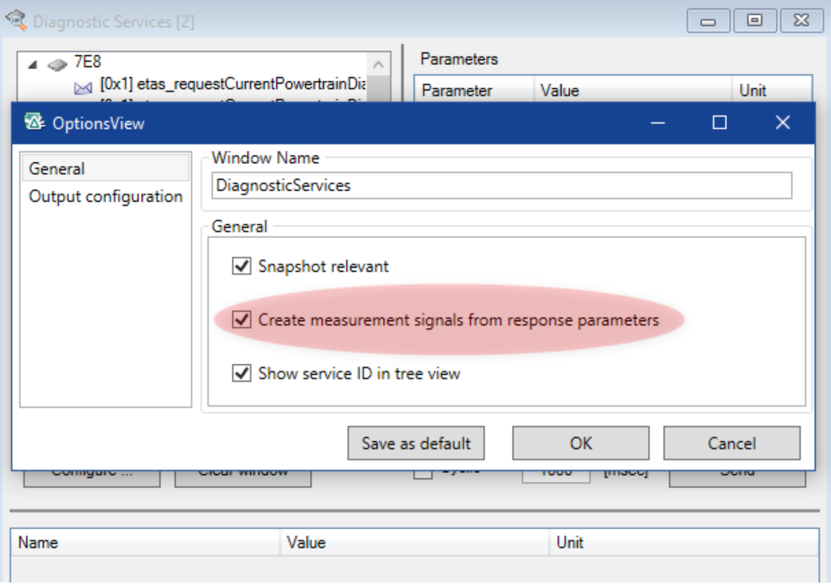
4.4.2 Adding Diagnostic Signals

To make diagnostic signals available for use in the Variable Selection dialog box, proceed as follows:

To add diagnostic signals

1. Open the experiment.
2. Select **ODX** > **User views** > **Diagnostic Services**.

   The "Diagnostic Services" window opens.
3. Select **Configure**.
4. Select the option "Create Measurement Signals from Response Parameters".



Window Name: DiagnosticServices

General

☑ Snapshot relevant

☑ Create measurement signals from response parameters

☑ Show service ID in tree view

5. Click **OK**.

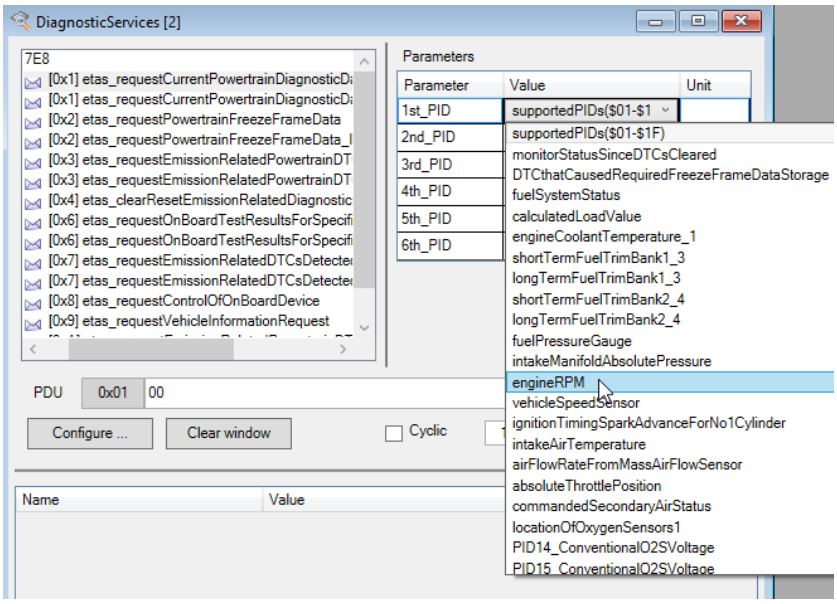
Note

Signals are created for the logical link (and thus also only for the corresponding device in the Variable Selection dialog box) for which you have selected the service!

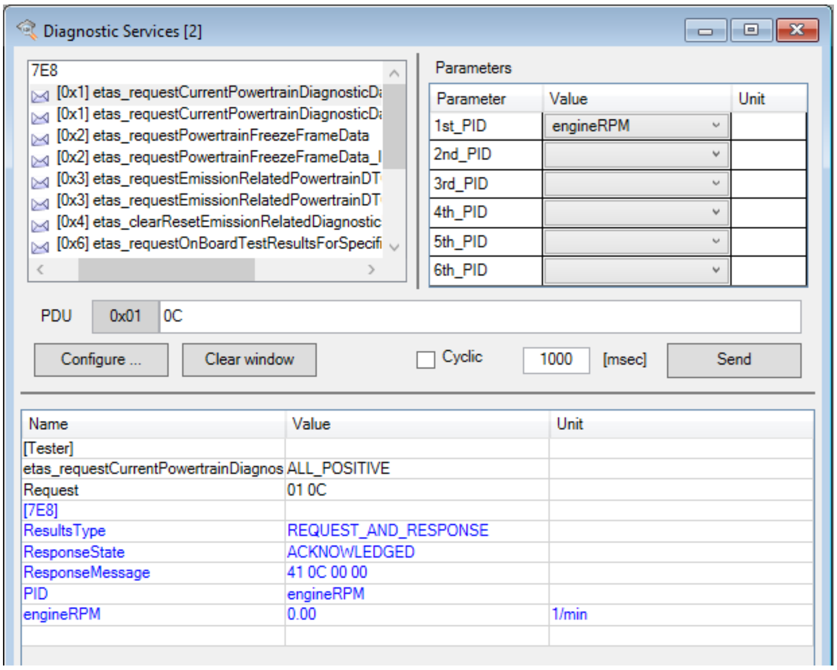
Note

The selection of this option only applies for the services sent subsequently within the current experiment session and is **not** saved in **Save as default**!

6. Select the service and the parameter you want to measure (and thus add to the DSL).


DiagnosticServices [2]
7E8
[0x1] etas_requestCurrentPowertrainDiagnosticD
[0x1] etas_requestCurrentPowertrainDiagnosticD
[0x2] etas_requestPowertrainFreezeFrameData
[0x2] etas_requestPowertrainFreezeFrameData_I
[0x3] etas_requestEmissionRelatedPowertrainDT
[0x3] etas_requestEmissionRelatedPowertrainDT
[0x4] etas_clearResetEmissionRelatedDiagnostic
[0x6] etas_requestOnBoardTestResultsForSpecifi
[0x6] etas_requestOnBoardTestResultsForSpecifi
[0x7] etas_requestEmissionRelatedDTCsDetecte
[0x7] etas_requestEmissionRelatedDTCsDetecte
[0x8] etas_requestControlOfOnBoardDevice
[0x9] etas_requestVehicleInformationRequest
Parameters
Parameter
Value
Unit
1st_PID
supportedPIDs($01-$1
2nd_PID
3rd_PID
4th_PID
5th_PID
6th_PID
supportedPIDs($01-$1F)
monitorStatusSinceDTCsCleared
DTCthatCausedRequiredFreezeFrameDataStorage
fuelSystemStatus
calculatedLoadValue
engineCoolantTemperature_1
shortTermFuelTrimBank1_3
longTermFuelTrimBank1_3
shortTermFuelTrimBank2_4
longTermFuelTrimBank2_4
fuelPressureGauge
intakeManifoldAbsolutePressure
engineRPM
vehicleSpeedSensor
ignitionTimingSparkAdvanceForNo1Cylinder
intakeAirTemperature
airFlowRateFromMassAirFlowSensor
absoluteThrottlePosition
commandedSecondaryAirStatus
locationOfOxygenSensors1
PID14_ConventionalO2SVoltage
PID15_ConventionalO2SVoltage
PDU
0x01
00
Configure ...
Clear window
Cyclic
Name
Value

7. Click **Send**.


Diagnostic Services [2]
7E8
[0x1] etas_requestCurrentPowertrainDiagnosticD
[0x1] etas_requestCurrentPowertrainDiagnosticD
[0x2] etas_requestPowertrainFreezeFrameData
[0x2] etas_requestPowertrainFreezeFrameData_I
[0x3] etas_requestEmissionRelatedPowertrainDT
[0x3] etas_requestEmissionRelatedPowertrainDT
[0x4] etas_clearResetEmissionRelatedDiagnostic
[0x6] etas_requestOnBoardTestResultsForSpecifi
Configure ...
Clear window
Send

Parameters

Parameter	Value	Unit
1st_PID	engineRPM	
2nd_PID		
3rd_PID		
4th_PID		
5th_PID		
6th_PID		

PDU 0x01 0C

□ Cyclic 1000 [msec]

Name	Value	Unit
[Tester]		
etas_requestCurrentPowertrainDiagnos	ALL_POSITIVE	
Request	01 0C	
[7E8]		
ResultsType	REQUEST_AND_RESPONSE	
ResponseState	ACKNOWLEDGED	
ResponseMessage	41 0C 00 00	
PID	engineRPM	
engineRPM	0.00	1/min

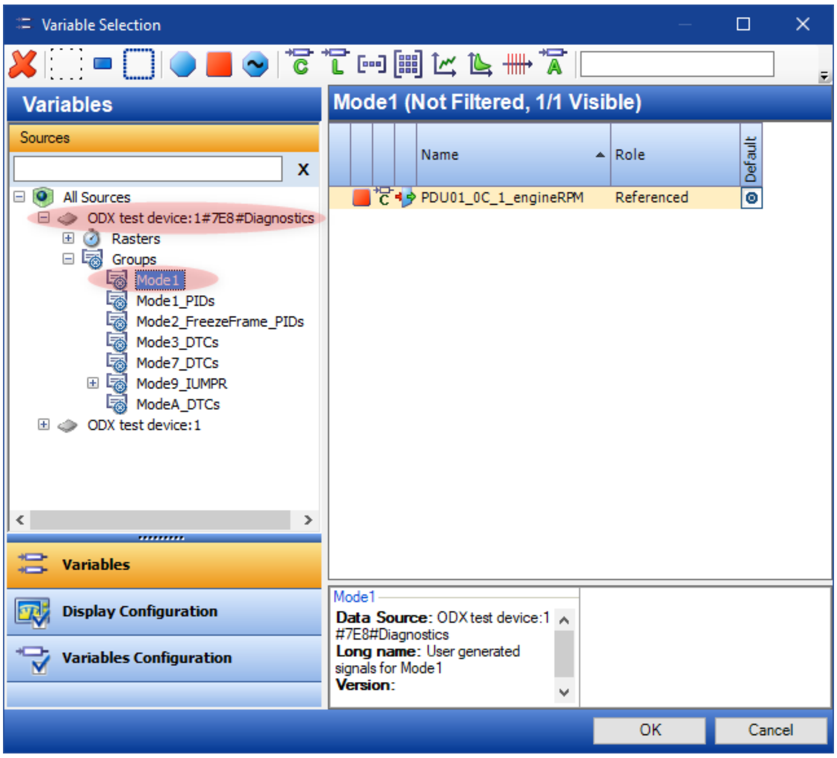
One signal is added to the DSL for every numeric value (Integer/Float) in the response.

8. Close the experiment.

   A note is displayed explaining that the ODX project has been modified (as new signals have been defined) (the diagnostic signal list is saved with the ODX project in the INCA database) and asking whether you want to save the ODX project.

9. Select **Yes** to save the new signals.
10. Open the experiment again.
11. Select **Variables** > **Variable Selection**.

The added diagnostic signals are displayed (here in group "Mode1").

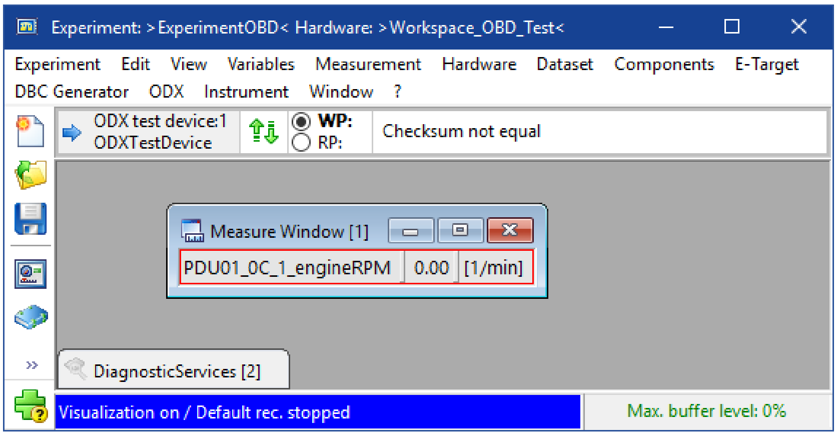


Name	Role
PDU01_0C_1_engineRPM	Referenced

Mode1
Data Source: ODX test device:1 #7E8#Diagnostics
Long name: User generated signals for Mode1
Version:

12. Select the signal in the list and right-click it.
13. Select **Add to** > **Layer_1** > **New** > **Measure Window** to display the signals and click **OK**.
14. Select **Measurement** > **Start Visualization**.

    The measured diagnostic signals are displayed.



Checksum not equal

PDU01_0C_1_engineRPM 0.00 [1/min]

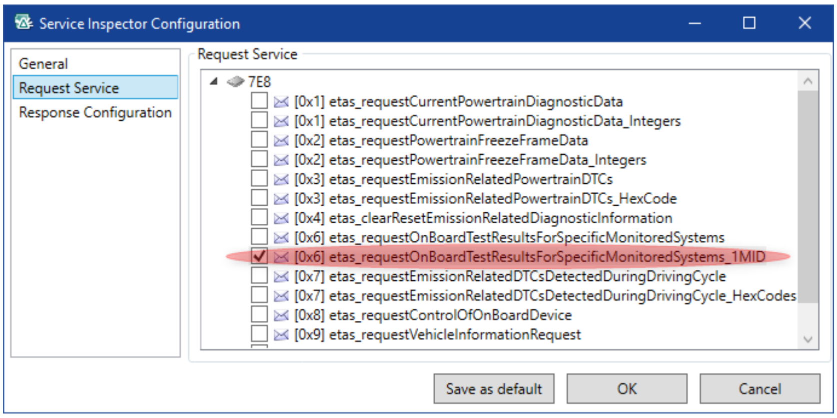
Visualization on / Default rec. stopped

Max. buffer level: 0%

15. Stop the measurement.
16. To include signals for all data that can be called up using this service in the list of diagnostic signals, select **ODX** > **User views** > **Service Inspector**.

    The "Service Inspector" window opens.
17. Click **Configure**.
18. Here too, activate the option "Create Measurement Signals from Response Parameters".

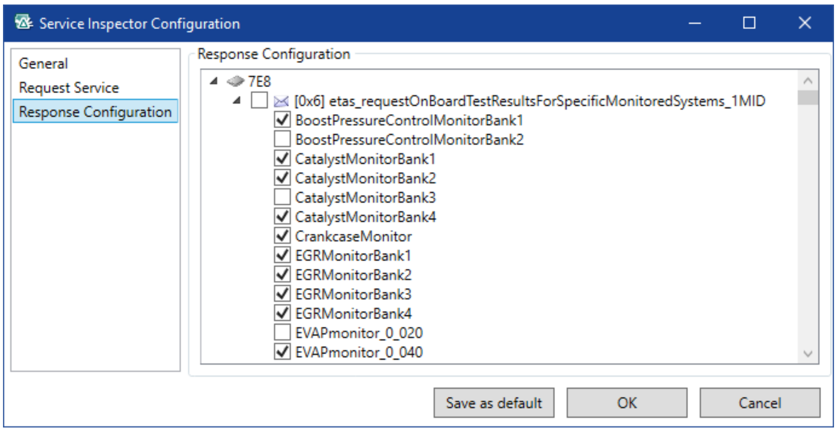
19. Select (on the left-hand side of the window) "Request Service" and then the required service on the right-hand side of the window.



Request Service

- 7E8
  - □ [0x1] etas_requestCurrentPowertrainDiagnosticData
  - □ [0x1] etas_requestCurrentPowertrainDiagnosticData_Integers
  - □ [0x2] etas_requestPowertrainFreezeFrameData
  - □ [0x2] etas_requestPowertrainFreezeFrameData_Integers
  - □ [0x3] etas_requestEmissionRelatedPowertrainDTCs
  - □ [0x3] etas_requestEmissionRelatedPowertrainDTCs_HexCode
  - □ [0x4] etas_clearResetEmissionRelatedDiagnosticInformation
  - □ [0x6] etas_requestOnBoardTestResultsForSpecificMonitoredSystems
  - ☑ [0x6] etas_requestOnBoardTestResultsForSpecificMonitoredSystems_1MID
  - □ [0x7] etas_requestEmissionRelatedDTCsDetectedDuringDrivingCycle
  - □ [0x7] etas_requestEmissionRelatedDTCsDetectedDuringDrivingCycle_HexCodes
  - □ [0x8] etas_requestControlOfOnBoardDevice
  - □ [0x9] etas_requestVehicleInformationRequest

20. If you need to configure the response, select "Response Configuration" and configure the response accordingly.

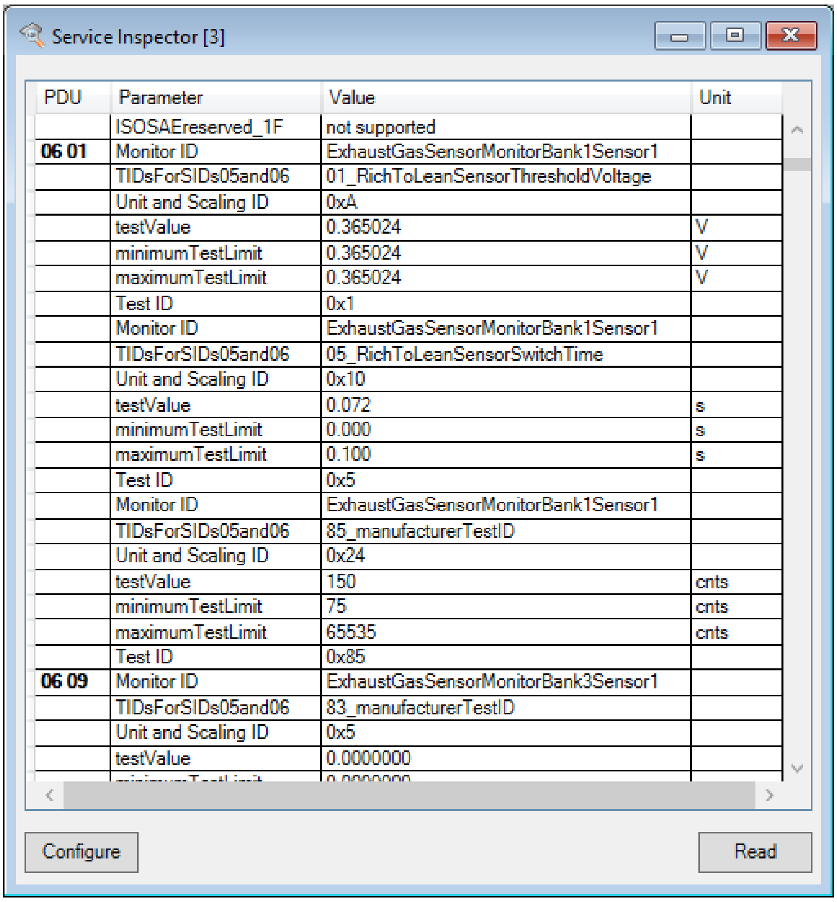


Response Configuration

- 7E8
  - □ [0x6] etas_requestOnBoardTestResultsForSpecificMonitoredSystems_1MID
    - ☑ BoostPressureControlMonitorBank1
    - □ BoostPressureControlMonitorBank2
    - ☑ CatalystMonitorBank1
    - ☑ CatalystMonitorBank2
    - □ CatalystMonitorBank3
    - ☑ CatalystMonitorBank4
    - ☑ CrankcaseMonitor
    - ☑ EGRMonitorBank1
    - ☑ EGRMonitorBank2
    - ☑ EGRMonitorBank3
    - ☑ EGRMonitorBank4
    - □ EVAPmonitor_0_020
    - ☑ EVAPmonitor_0_040

21. Click **OK**.

22. Click **Read** in the "Service Inspector" window.



PDU	Parameter	Value	Unit
	ISOSAEreserved_1F	not supported	
06 01	Monitor ID	ExhaustGasSensorMonitorBank1Sensor1	
	TIDsForSIDs05and06	01_RichToLeanSensorThresholdVoltage	
	Unit and Scaling ID	0xA	
	testValue	0.365024	V
	minimumTestLimit	0.365024	V
	maximumTestLimit	0.365024	V
	Test ID	0x1	
	Monitor ID	ExhaustGasSensorMonitorBank1Sensor1	
	TIDsForSIDs05and06	05_RichToLeanSensorSwitchTime	
	Unit and Scaling ID	0x10	
	testValue	0.072	s
	minimumTestLimit	0.000	s
	maximumTestLimit	0.100	s
	Test ID	0x5	
	Monitor ID	ExhaustGasSensorMonitorBank1Sensor1	
	TIDsForSIDs05and06	85_manufacturerTestID	
	Unit and Scaling ID	0x24	
	testValue	150	cnts
	minimumTestLimit	75	cnts
	maximumTestLimit	65535	cnts
	Test ID	0x85	
06 09	Monitor ID	ExhaustGasSensorMonitorBank3Sensor1	
	TIDsForSIDs05and06	83_manufacturerTestID	
	Unit and Scaling ID	0x5	
	testValue	0.0000000	
	minimumTestLimit	0.0000000	

One signal is added to the DSL for every numeric value (Integer/Float) in the response.

23. Close the experiment.

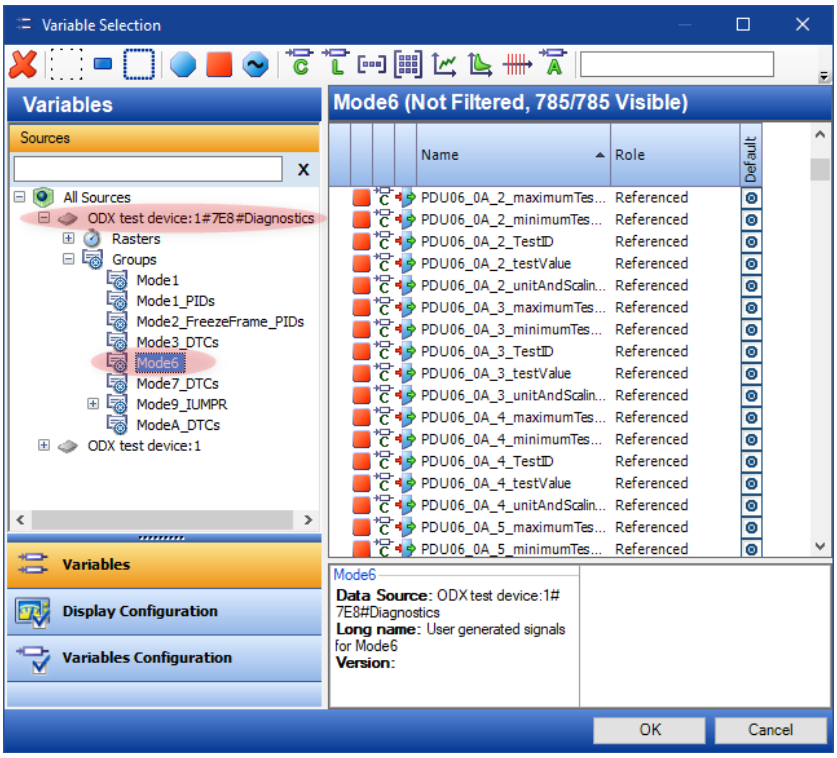
    A note is displayed explaining that the ODX project has been modified (as new signals have been defined) (the diagnostic signal list is saved with the ODX project in the INCA database) and asking whether you want to save the ODX project.

24. Select **Yes** to save the new signals.

25. Open the experiment again.

26. Select **Variables** > **Variable Selection**.

    The added diagnostic signals are displayed (here in group "Mode6").



Name	Role
PDU06_0A_2_maximumTes...	Referenced
PDU06_0A_2_minimumTes...	Referenced
PDU06_0A_2_TestID	Referenced
PDU06_0A_2_testValue	Referenced
PDU06_0A_2_unitAndScalin...	Referenced
PDU06_0A_3_maximumTes...	Referenced
PDU06_0A_3_minimumTes...	Referenced
PDU06_0A_3_TestID	Referenced
PDU06_0A_3_testValue	Referenced
PDU06_0A_3_unitAndScalin...	Referenced
PDU06_0A_4_maximumTes...	Referenced
PDU06_0A_4_minimumTes...	Referenced
PDU06_0A_4_TestID	Referenced
PDU06_0A_4_testValue	Referenced
PDU06_0A_4_unitAndScalin...	Referenced
PDU06_0A_5_maximumTes...	Referenced
PDU06_0A_5_minimumTes...	Referenced

Mode6
Data Source: ODX test device:1#7E8#Diagnostics
Long name: User generated signals for Mode6
Version:

27. Select the signals in the list, add them (as above) to a measure window and start the measurement.

4.4.3 Storage Location of the DSL File

The DSL file is stored in the INCA database in the ODX project. This means the signals are available in all workspaces/experiments that use this ODX project.

These can also be transferred to other INCA databases by exporting/importing.

5 ODX-LINK Tutorial

In this tutorial, you will learn the major operational procedures for ODX-LINK.

> This tutorial assumes that you have installed INCA and ODX-LINK. For information on installing ODX-LINK, see Chapter "Installation " on page 16. This tutorial also assumes that you are familiar with basic INCA operations.

The tutorial consists of the following lessons:

- "Creating an INCA Workspace " on the next page

  In this lesson, you will create a new INCA workspace for working with ODX-LINK.
- "Preparing an INCA Experiment for ODX without Hardware Connection" on page 76

  In this lesson, you will prepare the INCA experiment for using ODX without any external hardware. The ECU will be simulated, as in the previous lesson.
- "Working with ODX User Views " on page 78

  In this lesson, you will use the ODX user views to perform diagnostics and to display the responses from the ECU.
- "Preparing an INCA Experiment for ODX with Real Hardware" on page 82

  In this exercise, you will get an INCA experiment ready for working with your ECU. You need an A2L file and a HEX file for your ECU.
- "Configuration of ODX User Views " on page 85

  In this lesson, you will modify the configurations for Diagnostic Services, Service Inspector, and Diagnostic Trouble Code.
- "Using OBDonCAN (SAE J1979) with ODX-LINK" on page 87

  In this lesson, you will configure INCA for using the OBD protocol SAE J1979 with the OBDonCAN implementation and you will use the ODX-LINK OBD database for OBD diagnostic communication with a real ECU.
- "Using OBDonUDS (SAE J1979-2) with ODX-LINK" on page 92

  In this lesson, you will configure INCA for using the OBD protocol SAE J1979-2 with the OBDonUDS implementation and you will use the ODX-LINK OBD database for OBD diagnostic communication with a real ECU.
- "Working with the "OBD" User View" on page 97

  In this lesson, you will work with the OBD user view.
- "Using Diagnostic Signal in the Experiment" on page 100

  In this lesson you will learn how diagnostic signals can be used in an experiment.

- "Measuring OBD Data on the Vehicle" on page 102

  In this lesson, you measure diagnostic signals on the vehicle and generate additional ECU-specific measurement signals.

5.1 Creating an INCA Workspace

In this lesson, you will create a new INCA workspace for working with ODX-LINK.

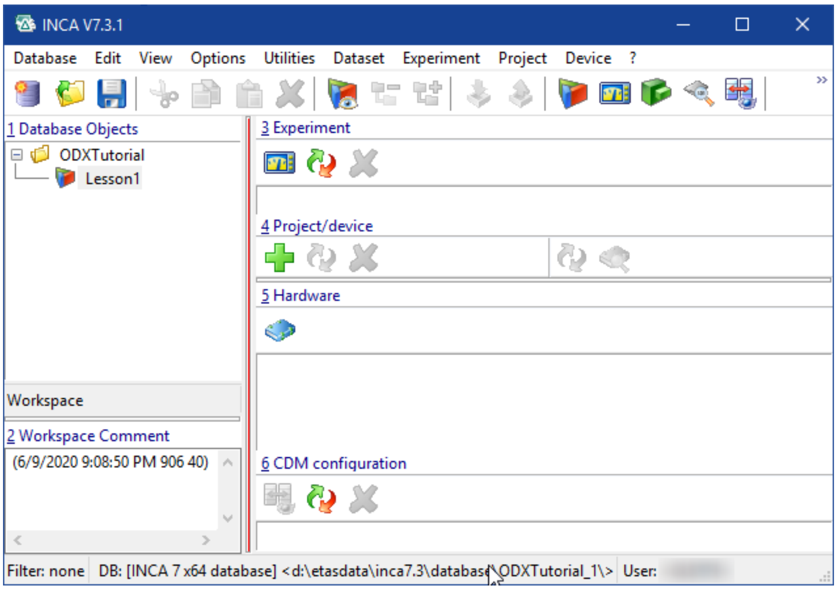
You do not need any additional devices for this lesson. The "K-Line Test Device" will simulate the ECU in this lesson.

To create a top folder

1. Start INCA.
2. Choose **Edit** > **Add** > **Add top folder**.
3. Enter `ODXTutorial` and press ENTER.

To create an INCA workspace

1. Choose **Edit** > **Add** > **Workspace**.
2. Enter `Lesson1` and press ENTER.



(6/9/2020 9:08:50 PM 906 40)

Filter: none DB: [INCA 7 x64 database] <d:\etasdata\inca7.3\database\ODXTutorial_1\> User:

To add an ECU project

1. Select **Edit** > **Add** › **ECU-Project (A2L)...**

   A window opens in which you can select the project file.
2. Select the `\ETASData\INCA7.5\Data\Demo\ODXTestDevice.a2l` file and click **Open**.

   A further window for selecting files opens.

   No ECU program file is required for simulating the ECU.
3. Click **Cancel**.

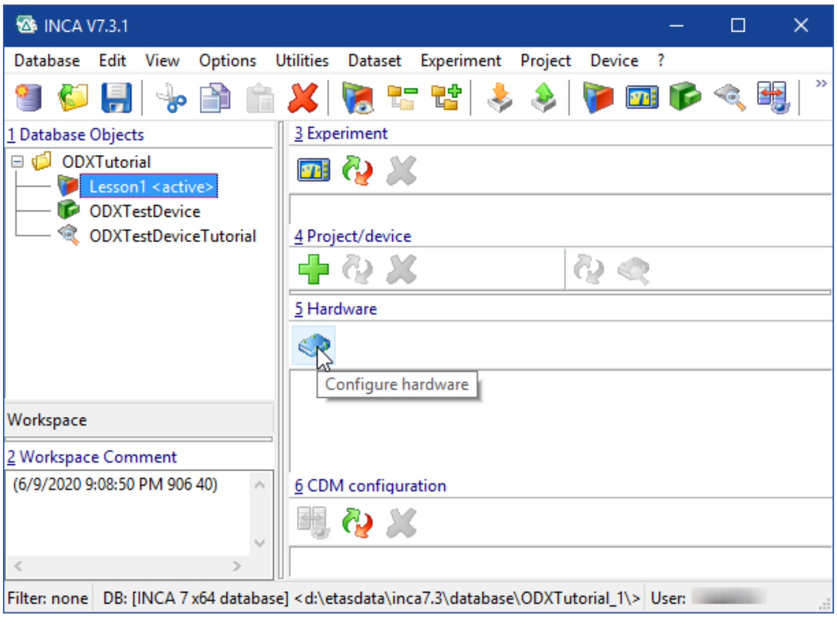
   The ECU project is added to the top folder you selected.

To add an ODX project

1. Change to the INCA main window and select the `ODXTutorial` folder.
2. Select **Edit** > **Add** > **ODX Project...**.

   A file selector window opens.
3. Select the file `\ETASData\ODX7.5\Projects\ODXTestDeviceTutorial.pdx` and click **Open**.

   The ODX project is added.



(6/9/2020 9:08:50 PM 906 40)

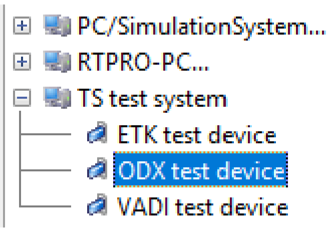
Filter: none DB: [INCA 7 x64 database] <d:\etasdata\inca7.3\database\ODXTutorial_1\> User: [redacted]

To add a new device

1. Select the "Lesson1" workspace.
2. Click the **Configure Hardware** icon.

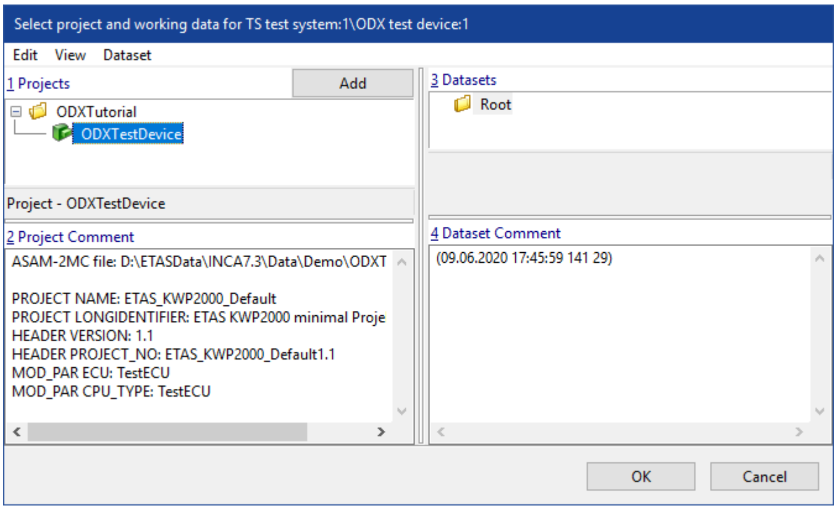
   The hardware configuration window opens.
3. Select **Device** > **Insert**.

   A window for selecting the hardware opens.
4. Extend the `TS test system` item.



5. Select "ODX test device" and click **OK**.

   The window for selecting the project opens.


Select project and working data for TS test system:1\ODX test device:1
Edit
View
Dataset
1 Projects
Add
3 Datasets
Root
OK
Cancel

ODXTutorial
ODXTestDevice

Project - ODXTestDevice

2 Project Comment

ASAM-2MC file: D:\ETASData\INCA7.3\Data\Demo\ODXT

PROJECT NAME: ETAS_KWP2000_Default
PROJECT LONGIDENTIFIER: ETAS KWP2000 minimal Proje
HEADER VERSION: 1.1
HEADER PROJECT_NO: ETAS_KWP2000_Default1.1
MOD_PAR ECU: TestECU
MOD_PAR CPU_TYPE: TestECU

4 Dataset Comment

(09.06.2020 17:45:59 141 29)

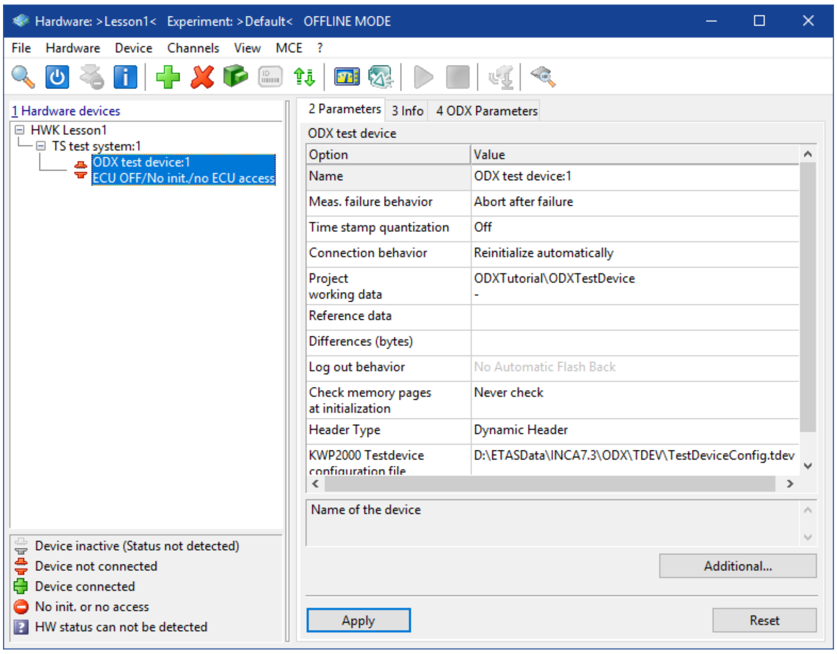
6. Select the "ODXTestDevice" project and click **OK**.

   A file selector window opens.

7. Select the `TestDeviceConfig.tdev` file (in the `ETASData\INCA7.5\ODX\TDEV` folder).

8. Click **Open**.

   This concludes the hardware configuration.



HWK Lesson1
TS test system:1
ODX test device:1
ECU OFF/No init./no ECU access

ODX test device

Option	Value
Name	ODX test device:1
Meas. failure behavior	Abort after failure
Time stamp quantization	Off
Connection behavior	Reinitialize automatically
Project working data	ODXTutorial\ODXTestDevice -
Reference data	
Differences (bytes)	
Log out behavior	No Automatic Flash Back
Check memory pages at initialization	Never check
Header Type	Dynamic Header
KWP2000 Testdevice configuration file	D:\ETASData\INCA7.3\ODX\TDEV\TestDeviceConfig.tdev

Name of the device

Device inactive (Status not detected)
Device not connected
Device connected
No init. or no access
HW status can not be detected

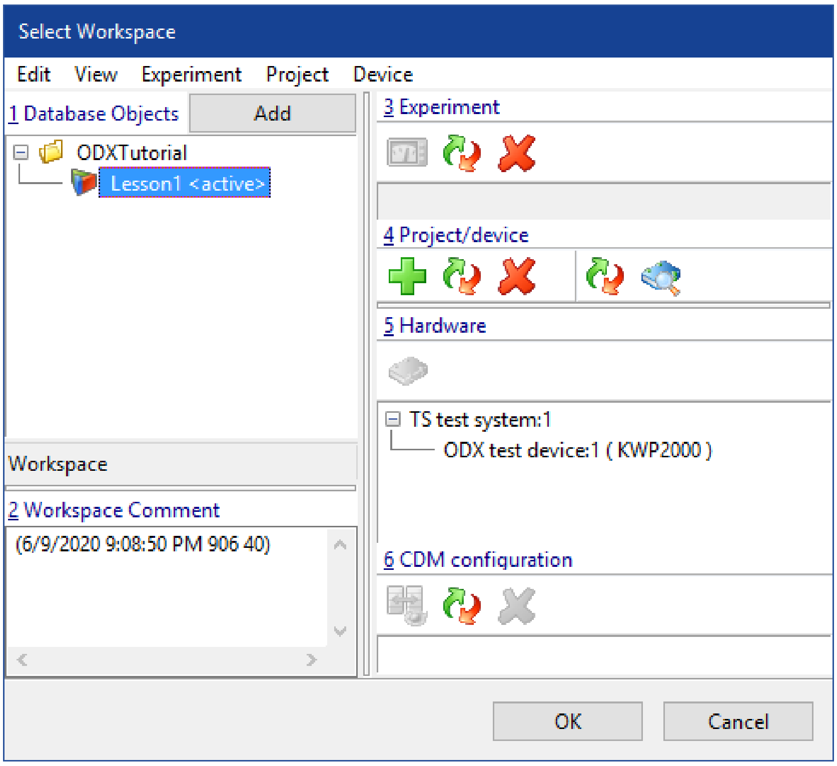
9. Close the "Hardware: >Lesson1<" window.

To add an experiment

1. Choose **Edit** > **Add** > **Experiment**.
2. Enter `Experiment1` and press ENTER.
3. Double-click `Experiment1`.

   Now the dialog window for selecting the workspace will appear.

4. Select `Lesson1` and click **OK**.



TS test system:1
ODX test device:1 (KWP2000)

(6/9/2020 9:08:50 PM 906 40)

This concludes the hardware and workspace configuration.

5. Close the "Experiment ›Experiment1‹" window.

In this lesson, you have created a new INCA workspace, added an ECU project, an ODX project and hardware definition and created an experiment.

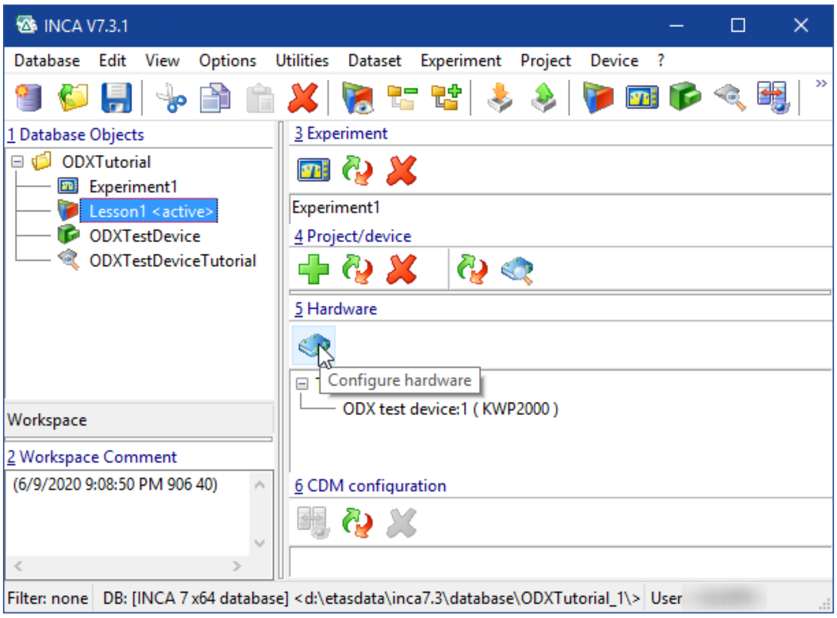
5.2 Preparing an INCA Experiment for ODX without Hardware Connection

In this lesson, you will prepare the INCA experiment for using ODX without any external hardware. The ECU will be simulated, as in the previous lesson.

The settings you created in Lesson "Creating an INCA Workspace " on page 73 are prerequisites for this lesson.

To open an ODX configuration

1. Switch to the INCA main window.
2. Open the `ODXTutorial` folder.
3. Select the workspace `Lesson1`.
4. Click the **Configure Hardware** icon.



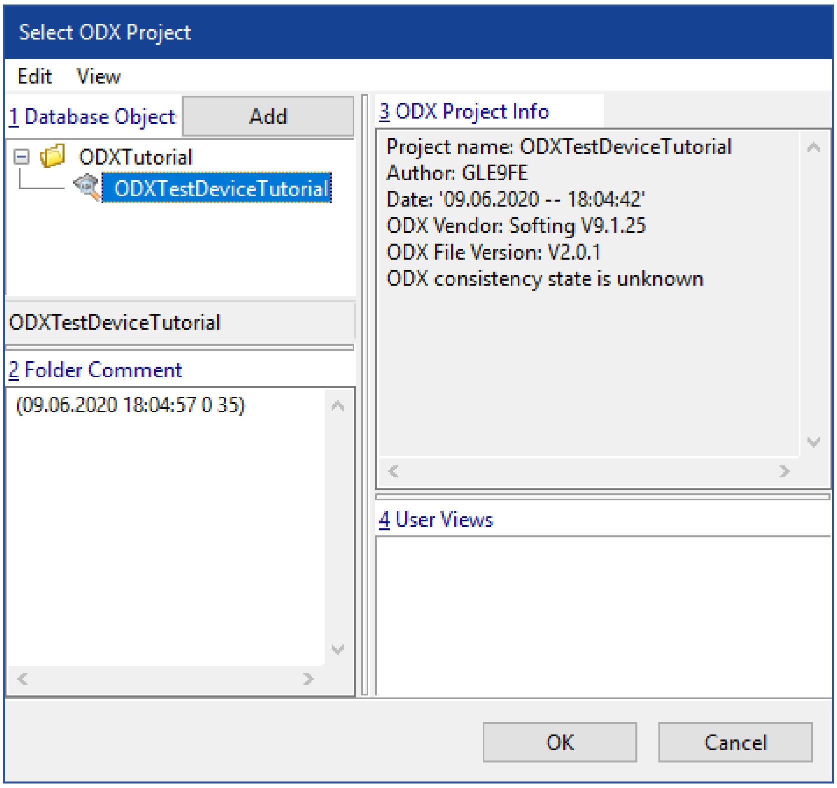
(6/9/2020 9:08:50 PM 906 40)

Filter: none DB: [INCA 7 x64 database] <d:\etasdata\inca7.3\database\ODXTutorial_1\> User [redacted]

The hardware configuration window opens.

5. Choose **Hardware** > **Configure ODX**.

   The dialog window for selecting the ODX configuration is displayed.



(09.06.2020 18:04:57 0 35)

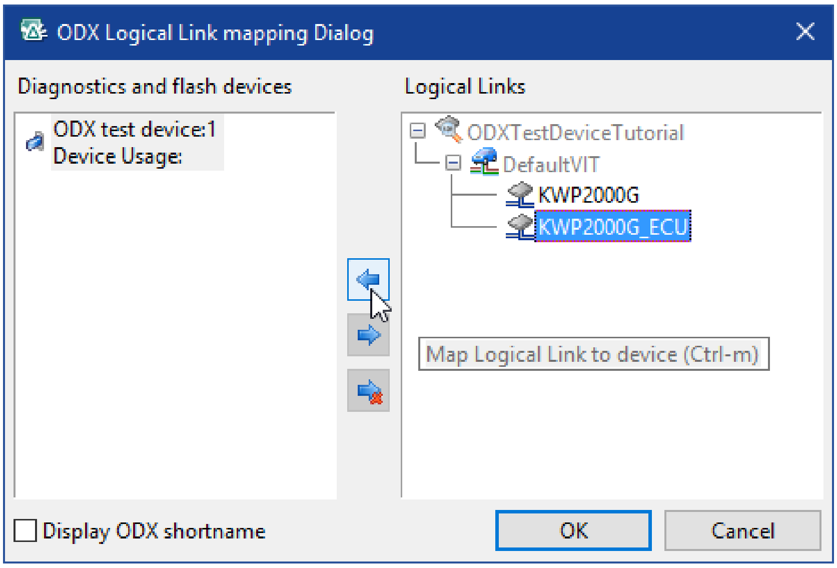
Project name: ODXTestDeviceTutorial
Author: GLE9FE
Date: '09.06.2020 -- 18:04:42'
ODX Vendor: Softing V9.1.25
ODX File Version: V2.0.1
ODX consistency state is unknown

6. Select the `ODXTestDeviceTutorial` project file and click **OK**.

   The dialog window for logical link mapping is displayed.

7. Select "ECU".

8. Click the left arrow.



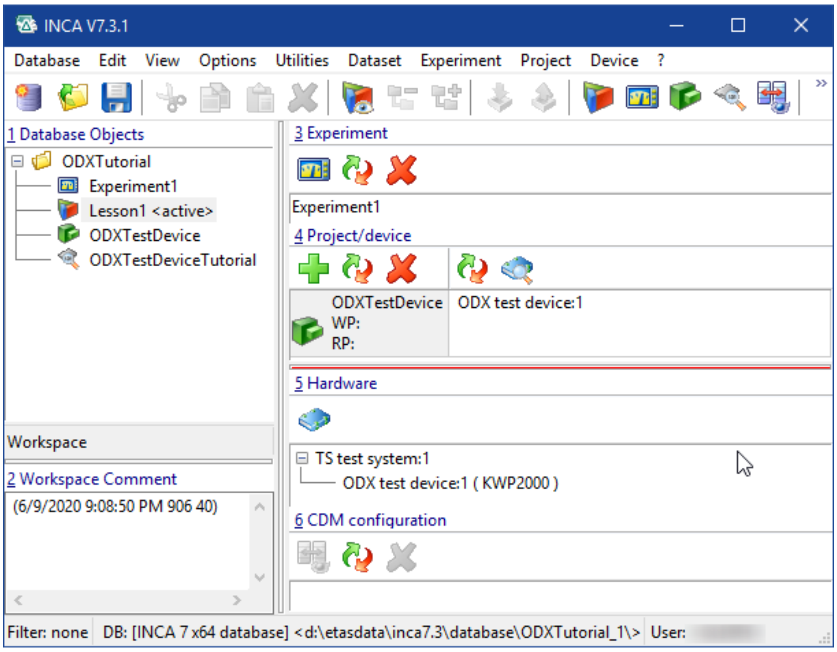
ODX Logical Link mapping Dialog
Diagnostics and flash devices
Logical Links
Map Logical Link to device (Ctrl-m)
Display ODX shortname
OK
Cancel

ODX test device:1
Device Usage:

ODXTestDeviceTutorial
DefaultVIT
KWP2000G
KWP2000G_ECU

□ Display ODX shortname

9. Click **OK**.



ODXTutorial
Experiment1
Lesson1 <active>
ODXTestDevice
ODXTestDeviceTutorial

Experiment1

ODXTestDevice
WP:
RP:
ODX test device:1

TS test system:1
ODX test device:1 (KWP2000)

(6/9/2020 9:08:50 PM 906 40)

Filter: none DB: [INCA 7 x64 database] <d:\etasdata\inca7.3\database\ODXTutorial_1\> User: [redacted]

10. Close the hardware configuration window.

 In this lesson, you have opened an INCA experiment, opened an ODX configuration and assigned a logical link of the ODX project to an INCA device.

5.3 Working with ODX User Views

In this lesson, you will use the ODX user views to perform diagnostics and to display the responses from the ECU.

Since in this lesson you will not be working with external hardware either, all responses from the ECU will be simulated. The simulated ECU responses are defined in the `TestDeviceConfig.tdev` file.

The settings you created in Lesson "Preparing an INCA Experiment for ODX without Hardware Connection" on page 76 are prerequisites for this lesson.

To display the Service Inspector

1. Double-click the workspace "Lesson1".

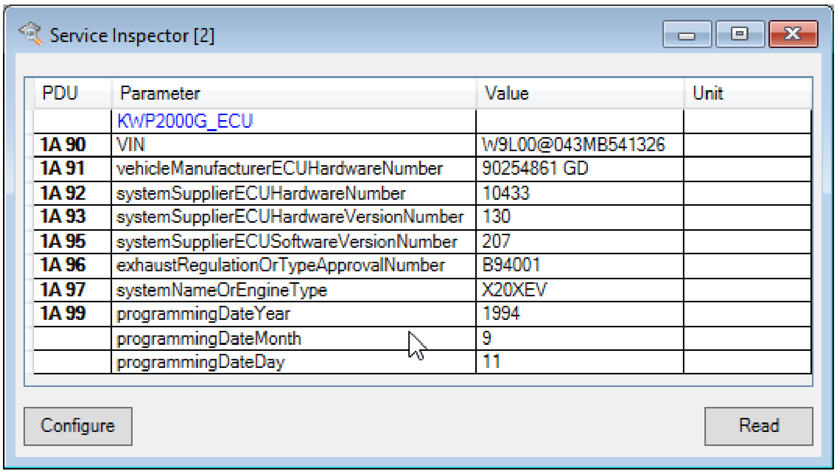
   The experiment "Experiment1" is opened.

2. Choose **ODX** > **User views** > **Service Inspector**.

   The dialog window for the Service Inspector will be displayed.

3. Click **Read**.

   The service request to query the Service Inspector is sent to the ECU. The contents of the service request and the ECU response are displayed in the bottom section of the window.



PDU	Parameter	Value	Unit
	KWP2000G_ECU		
1A 90	VIN	W9L00@043MB541326	
1A 91	vehicleManufacturerECUHardwareNumber	90254861 GD	
1A 92	systemSupplierECUHardwareNumber	10433	
1A 93	systemSupplierECUHardwareVersionNumber	130	
1A 95	systemSupplierECUSoftwareVersionNumber	207	
1A 96	exhaustRegulationOrTypeApprovalNumber	B94001	
1A 97	systemNameOrEngineType	X20XEV	
1A 99	programmingDateYear	1994	
	programmingDateMonth	9	
	programmingDateDay	11	

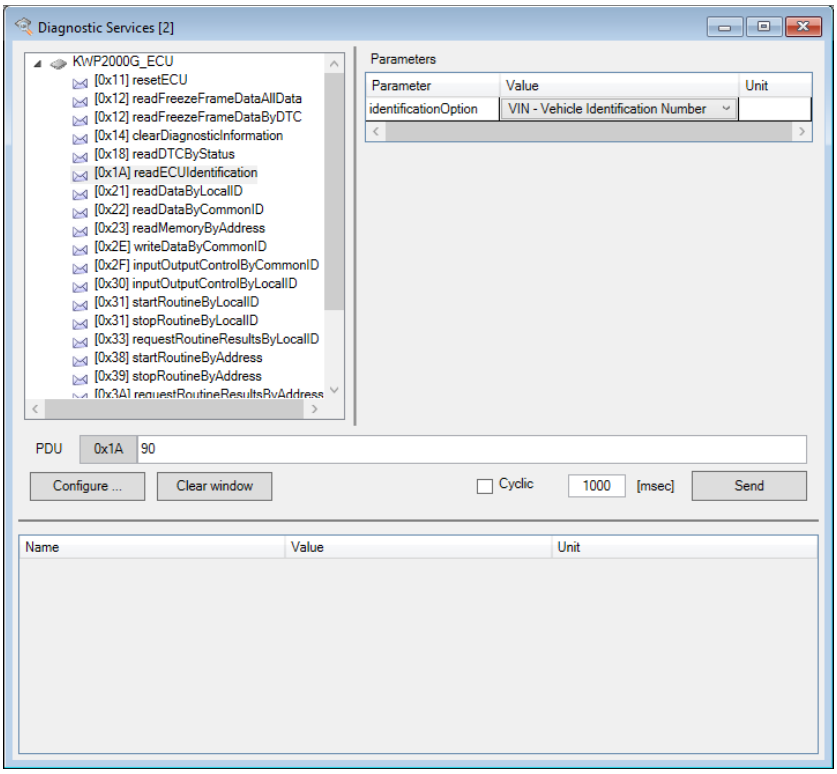
4. Close the dialog window.

> ### Note
>
> After installation, this function is assigned the "[1A] readService Inspector" service identification from the KWP2000 protocol. You can, however, assign any service identification to this function. For further information, see "Configuration of ODX User Views " on page 85.

To execute a diagnostic service

1. Choose **ODX** > **User views** > **DiagnosticServices**.

   The dialog window for the diagnostic services is displayed.


Diagnostic Services [2]
KWP2000G_ECU
[0x11] resetECU
[0x12] readFreezeFrameDataAllData
[0x12] readFreezeFrameDataByDTC
[0x14] clearDiagnosticInformation
[0x18] readDTCByStatus
[0x1A] readECUIdentification
[0x21] readDataByLocalID
[0x22] readDataByCommonID
[0x23] readMemoryByAddress
[0x2E] writeDataByCommonID
[0x2F] inputOutputControlByCommonID
[0x30] inputOutputControlByLocalID
[0x31] startRoutineByLocalID
[0x31] stopRoutineByLocalID
[0x33] requestRoutineResultsByLocalID
[0x38] startRoutineByAddress
[0x39] stopRoutineByAddress
[0x3A] requestRoutineResultsByAddress
Configure ...
Clear window
Send

Parameters

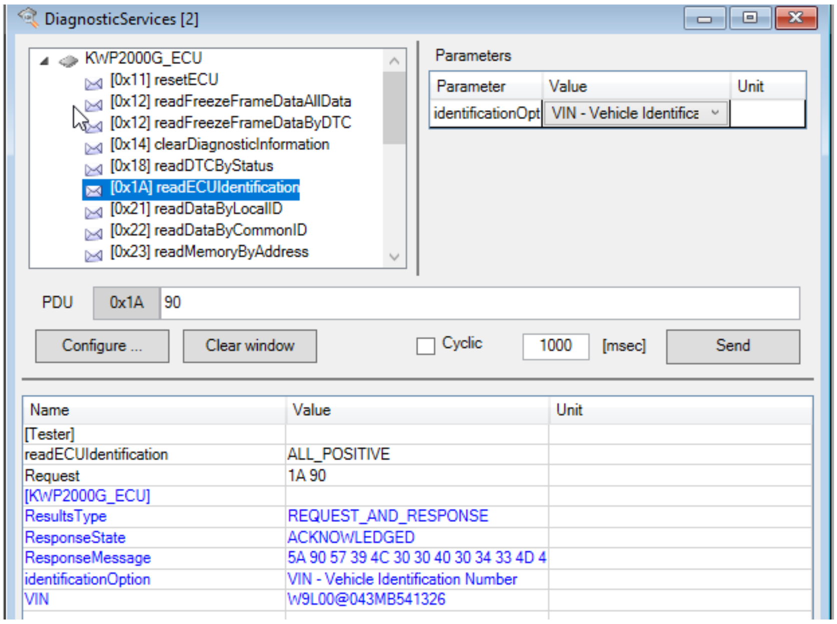
Parameter	Value	Unit
identificationOption	VIN - Vehicle Identification Number	

PDU 0x1A 90

□ Cyclic 1000 [msec]

Name	Value	Unit

2. Select the `[1A] readECU-Identification` service.
3. In the "Value" column, click the first entry in the "Parameters" field. In the selection list, click the "VIN - Vehicle Identification Number" entry.
4. Click **Send**.


DiagnosticServices [2]
KWP2000G_ECU
[0x11] resetECU
[0x12] readFreezeFrameDataAllData
[0x12] readFreezeFrameDataByDTC
[0x14] clearDiagnosticInformation
[0x18] readDTCByStatus
[0x1A] readECUIdentification
[0x21] readDataByLocalID
[0x22] readDataByCommonID
[0x23] readMemoryByAddress
Configure ...
Clear window
Send

Parameters

Parameter	Value	Unit
identificationOpt	VIN - Vehicle Identifica	

PDU 0x1A 90

□ Cyclic 1000 [msec]

Name	Value	Unit
[Tester]		
readECUIdentification	ALL_POSITIVE	
Request	1A 90	
[KWP2000G_ECU]		
ResultsType	REQUEST_AND_RESPONSE	
ResponseState	ACKNOWLEDGED	
ResponseMessage	5A 90 57 39 4C 30 30 40 30 34 33 4D 4	
identificationOption	VIN - Vehicle Identification Number	
VIN	W9L00@043MB541326	

The service request is sent to the simulated ECU. The contents of the service request and the ECU response are displayed in the bottom section of the window.

Note

Some settings for this window are configurable. For further information, see "Configuration of ODX User Views " on page 85.

To execute a diagnostic service on a regular basis

1. Select the "Cyclic" option.
2. Enter `1500` as the cycle time.
3. Click **Send**.

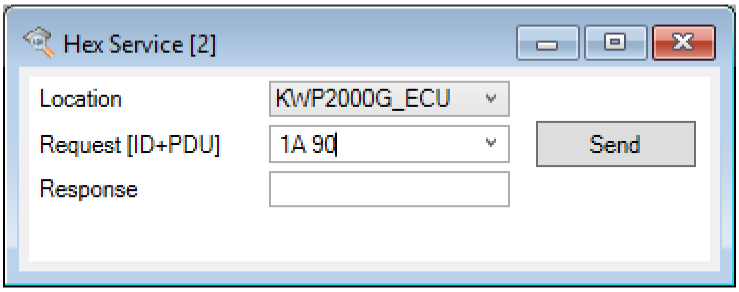
   The service request will be sent to the ECU every 1.5 seconds. The contents of the service request and the ECU response are displayed in the bottom section of the window.
4. Click **STOP**.

   This stops the cyclic sending of the service request.
5. Close the window.

To execute a freely configurable service request

1. Choose **ODX** > **User views** > **Hex Services**.

   The dialog window for the freely configurable diagnostic services is displayed.



Hex Service [2]

Location: KWP2000G_ECU

Request [ID+PDU]: 1A 90

Response:

2. In the "Request" field, enter the service request in hexadecimal notation: Type in `1A 90`.
3. Click **Send**.

   The ECU response is displayed in the "Response" field.
4. Close the dialog window.

In this lesson, you have worked with the user views **Service Inspector**, **Diagnostic Services** and **Hex Service**. You have sent the predefined service request for identifying the ECU, a service request from the ODX database, a freely configurable service request to the ECU, and have monitored the response from the ECU.

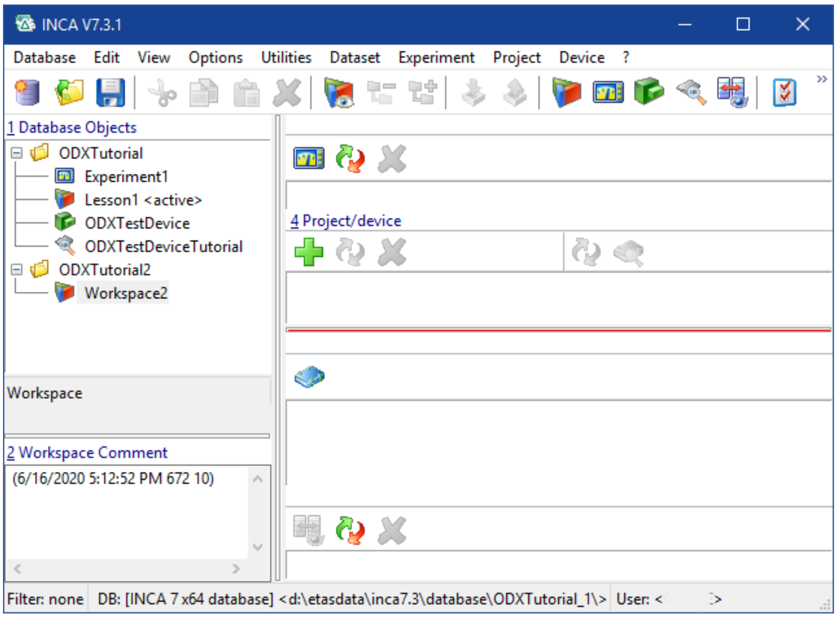
If you have not connected any additional hardware to your system, go on to "Configuration of ODX User Views " on page 85.

5.4 Preparing an INCA Experiment for ODX with Real Hardware

In this exercise, you will get an INCA experiment ready for working with your ECU. You need an A2L file and a HEX file for your ECU.

To create a top folder and INCA workspace

1. Start INCA.
2. Create a new top folder. Choose **Edit** > **Add** > **Add top folder**.
3. Type in `ODXTutorial2` and press ENTER.
4. Add a new workspace. Choose **Edit** > **Add** > **Workspace**.
5. Type in `Workspace2` and press ENTER.



(6/16/2020 5:12:52 PM 672 10)

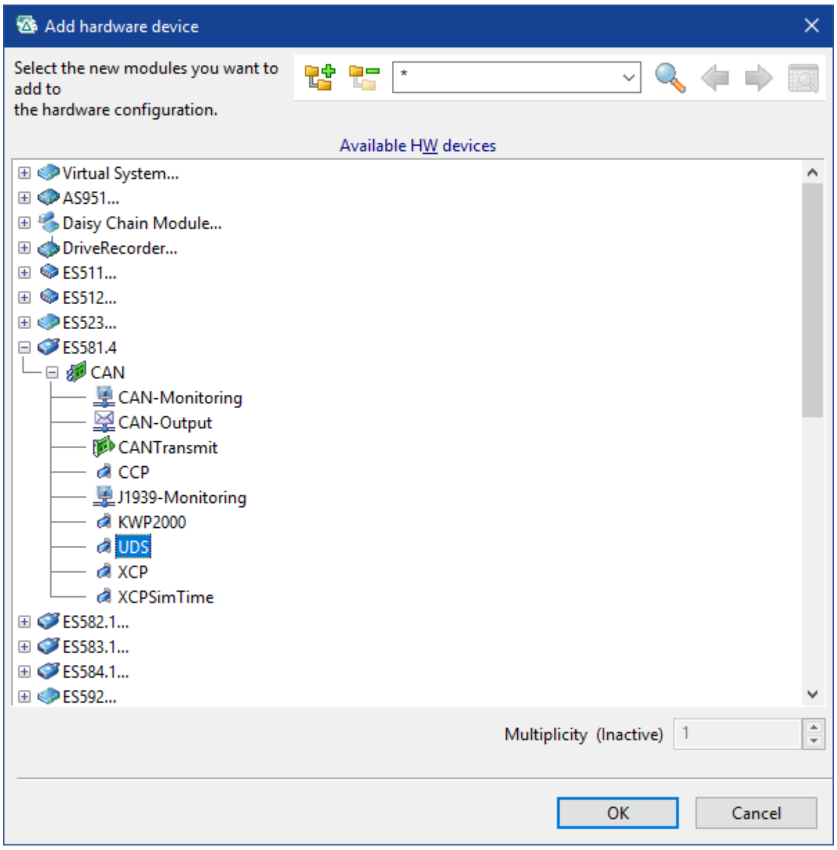
Filter: none DB: [INCA 7 x64 database] <d:\etasdata\inca7.3\database\ODXTutorial_1\> User: < >

To add a new device

1. Choose **Device** > **Configure hardware**.

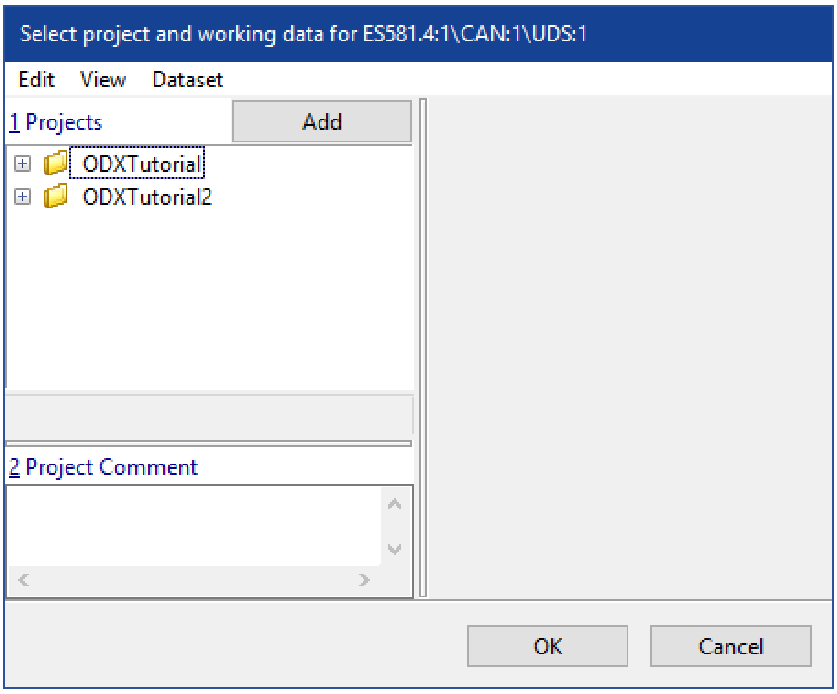
   The Hardware Configuration Editor is opened.
2. Choose **Device** > **Insert** and select the device you are using.

   In the example here, an ES581 with UDS via CAN has been selected. For your own ECU, you may have to select a different device.



3. Click **OK**.

   The dialog window for selecting the project is displayed.



4. Select a project or click **Cancel**.

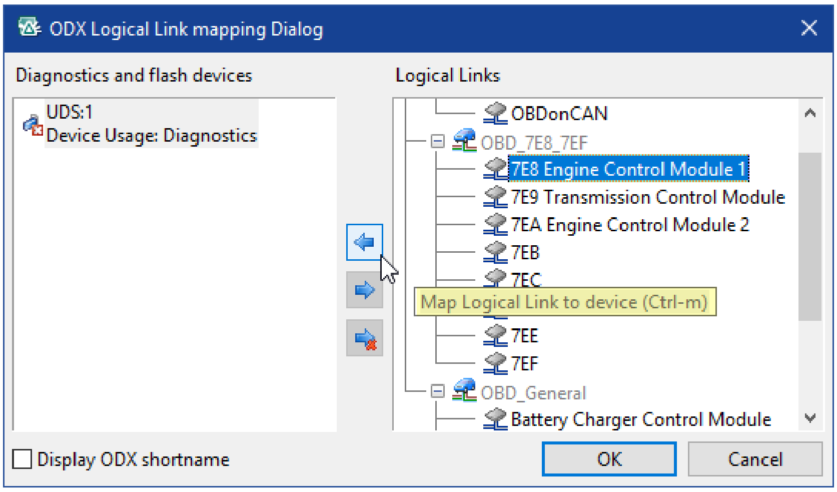
Note

The ODX configuration parameters can be determined from the ODX project - an A2L file is not required!

To configure ODX

1. Select **Hardware** > **Configure ODX** from the Hardware Configuration Editor.
2. Select your ODX project file and click **OK**.

   The ODX project opens.
3. Select the logical link and assign it to the diagnostic device using the arrow button.


ODX Logical Link mapping Dialog
Diagnostics and flash devices
UDS:1
Device Usage: Diagnostics
Logical Links
OBDonCAN
OBD_7E8_7EF
7E8 Engine Control Module 1
7E9 Transmission Control Module
7EA Engine Control Module 2
7EB
7EC
Map Logical Link to device (Ctrl-m)
7EE
7EF
OBD_General
Battery Charger Control Module
☐ Display ODX shortname
OK
Cancel

4. Click **OK**.

   The hardware is now configured.
5. Close the "Hardware: >Workspace2<" window.

To add an experiment

1. Switch to the "INCA" window and select `ODXTutorial2`.
2. Choose **Add** > **Experiment**.
3. Type in `Experiment2` and press ENTER.
4. Double-click `Experiment2`.

   A dialog window for selecting the workspace now appears.
5. Select `Workspace2` and click **OK**.

   The experiment is opened.
6. Click **Close**.

   This concludes the configuration of the hardware and the workspace.
7. Close the "Experiment >Experiment2<" window.

In this exercise, you have created an INCA experiment for working with your own ECU, have thus created a new INCA workspace, added your hardware components and a new experiment.

5.5 Configuration of ODX User Views

In this lesson, you will modify the configurations for Diagnostic Services, Service Inspector, and Diagnostic Trouble Code.

The following lessons are prerequisites for this lesson:

- "Creating an INCA Workspace " on page 73
- "Preparing an INCA Experiment for ODX without Hardware Connection" on page 76
- "Working with ODX User Views " on page 78

To open an INCA experiment

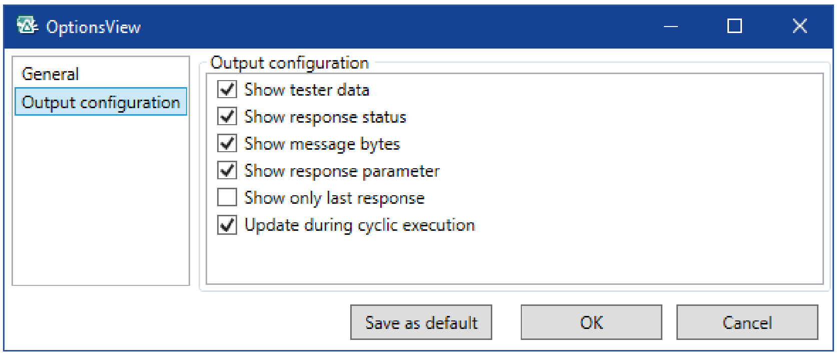
1. Switch to INCA.
2. Open the "ODXTutorial" folder.
3. Double-click "Lesson1".

   The experiment opens.

To configure "Diagnostic Services"

1. Choose **ODX** > **User views** > **Diagnostic Services**.
2. Click **Configure**..
3. Select "Output Configuration".

   The configuration dialog window for the display will open.



Output configuration

☑ Show tester data
☑ Show response status
☑ Show message bytes
☑ Show response parameter
☐ Show only last response
☑ Update during cyclic execution

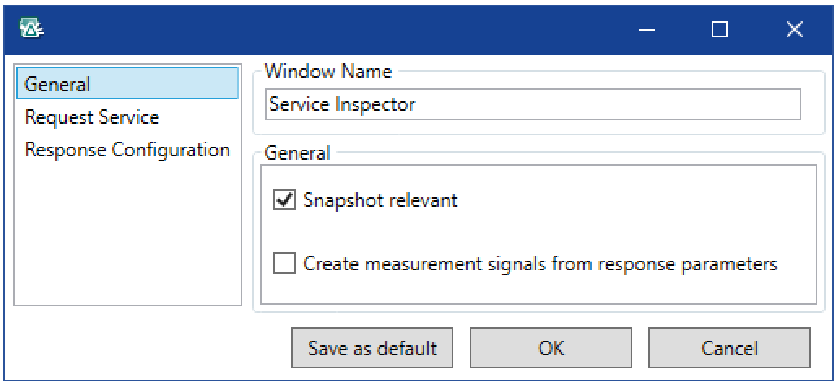
4. If you have not already done so, please make the following settings:
   - Select "Show tester data" to enable the service request to be displayed.
   - Select the "Show response status" check box to enable a plain text display of the ECU response.
   - Select the "Show message bytes" check box to enable the „response messages" to be displayed.
   - Select the "Show response parameter" check box to enable the interpreted response to be displayed.
   - Select "Update during cyclic execution" to refresh the display whenever the service request is periodically repeated.
5. Click **OK**.

   Your settings are saved.

To configure "Service Inspector"

1. Choose **ODX** > **User views** > **Service Inspector**.
2. Click **Configure**.

   The dialog window for configuring the service for identifying the ECU will be displayed.



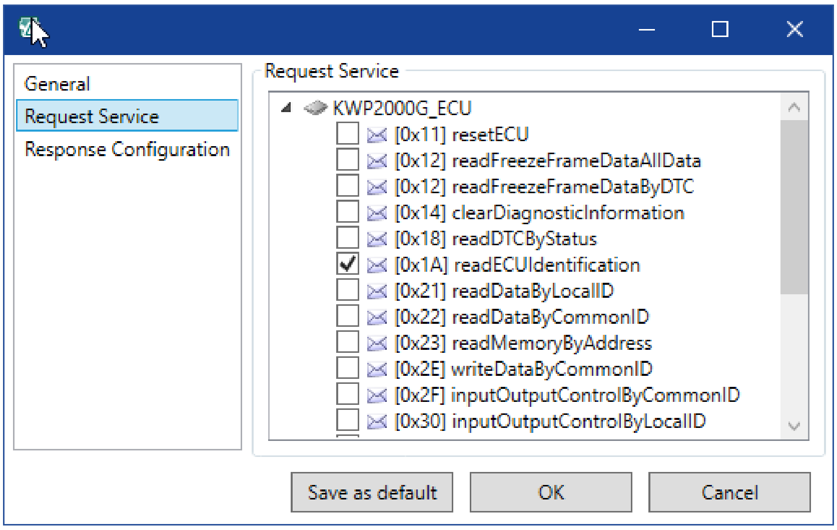
Window Name: Service Inspector

General

☑ Snapshot relevant

☐ Create measurement signals from response parameters

3. Select "Request Service" in the left-hand window.



Request Service

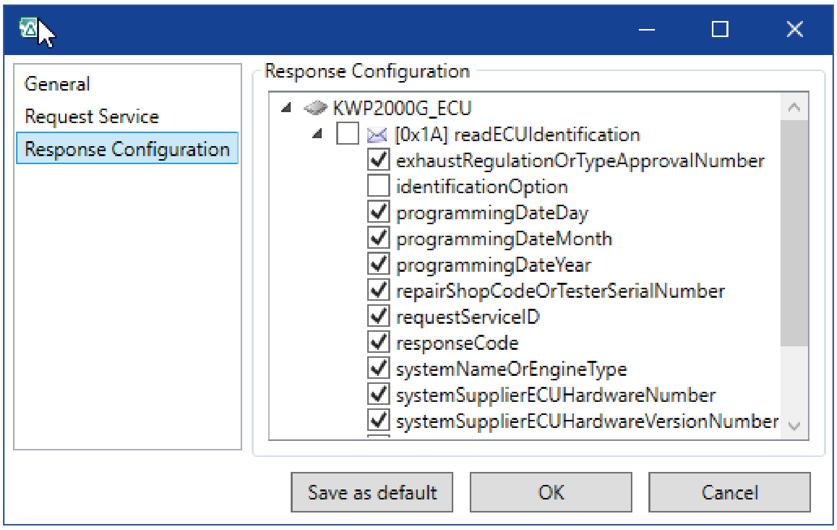
KWP2000G_ECU
- ☐ [0x11] resetECU
- ☐ [0x12] readFreezeFrameDataAllData
- ☐ [0x12] readFreezeFrameDataByDTC
- ☐ [0x14] clearDiagnosticInformation
- ☐ [0x18] readDTCByStatus
- ☑ [0x1A] readECUIdentification
- ☐ [0x21] readDataByLocalID
- ☐ [0x22] readDataByCommonID
- ☐ [0x23] readMemoryByAddress
- ☐ [0x2E] writeDataByCommonID
- ☐ [0x2F] inputOutputControlByCommonID
- ☐ [0x30] inputOutputControlByLocalID

4. Select the service to be used for reading the Service Inspector.

Note

You can as well select any other service – the output always displays the ECU responses of the selected service.

5. Select "Response Configuration" in the left-hand window.



Response Configuration

- KWP2000G_ECU
  - □ [0x1A] readECUIdentification
    - ☑ exhaustRegulationOrTypeApprovalNumber
    - □ identificationOption
    - ☑ programmingDateDay
    - ☑ programmingDateMonth
    - ☑ programmingDateYear
    - ☑ repairShopCodeOrTesterSerialNumber
    - ☑ requestServiceID
    - ☑ responseCode
    - ☑ systemNameOrEngineType
    - ☑ systemSupplierECUHardwareNumber
    - ☑ systemSupplierECUHardwareVersionNumber

6. Select the options you want to read from the ECU.

   Click **OK**.

   Your settings are saved

In this lesson, you have configured the "Diagnostic Services","Service Inspector".

5.6 Using OBDonCAN (SAE J1979) with ODX-LINK

In this lesson, you will configure INCA for using the OBD protocol SAE J1979 with the OBDonCAN implementation and you will use the ODX-LINK OBD database for OBD diagnostic communication with a real ECU.

Note

You can only do this lesson if you have connected one of the hardware components supported by ODX-LINK (e.g. ES592, ES595, ES582, ES910) and an ECU.

For this lesson, you need an ECU which supports OBD communication in accordance with ISO15765-4 on CAN or ISO15031-5/SAE J1979.

To configure INCA for OBDonCAN

1. Start INCA.
2. Create a new top folder with **Edit** > **Add** > **Add top folder**.
3. Enter "OBD Tutorial" as top folder name and press ENTER.
4. Add a new workspace using **Edit** > **Add** > **Workspace**.
5. Enter "OBDLesson" as the name and press ENTER.

To add an ODX project for OBD

1. Change to the "INCA" window and select "OBD Tutorial".
2. Select **Edit** > **Add** > **ODX Project**.
3. A file selector window opens for selecting the project or ODX/PDX file(s).

4. Navigate to the `\ETASData\ODX7.5\Projects` folder.
5. Select the file `OBDonCAN_ETAS_SAEJ1979<Version>.pdx`
6. Click **Open**.

   The ODX project is added.

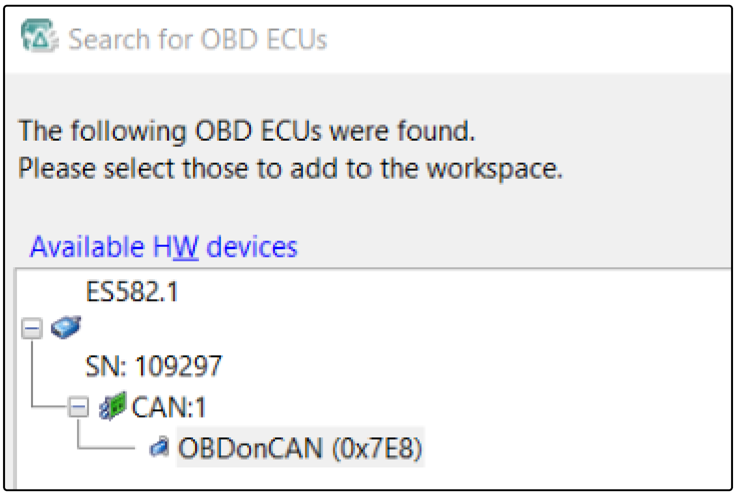
To configure hardware

1. Select the "OBDLesson" workspace.
2. Select **Device** > **Configure Hardware**.

   The hardware configuration window opens.
3. Select **Hardware** > **Search for OBD ECUs**.
4. From the list, select the interface types to be searched for and click **OK**.

   A search takes place for connected hardware with OBDonCAN support.

   The result is shown in the "Search for OBD ECUs" window and can be selected there.



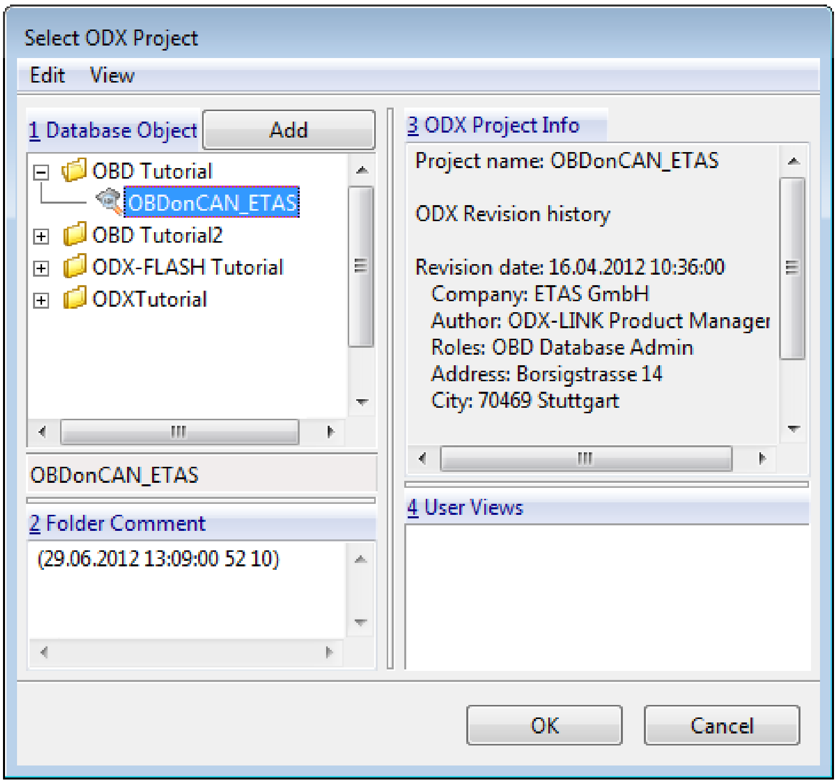
Search for OBD ECUs

The following OBD ECUs were found.
Please select those to add to the workspace.

Available HW devices

- ES582.1
  SN: 109297
  - CAN:1
    - OBDonCAN (0x7E8)

5. Select an OBDonCAN ECU found and click **OK**.
6. Select the corresponding ODX project from the following window and click **OK**.



OBDonCAN_ETAS

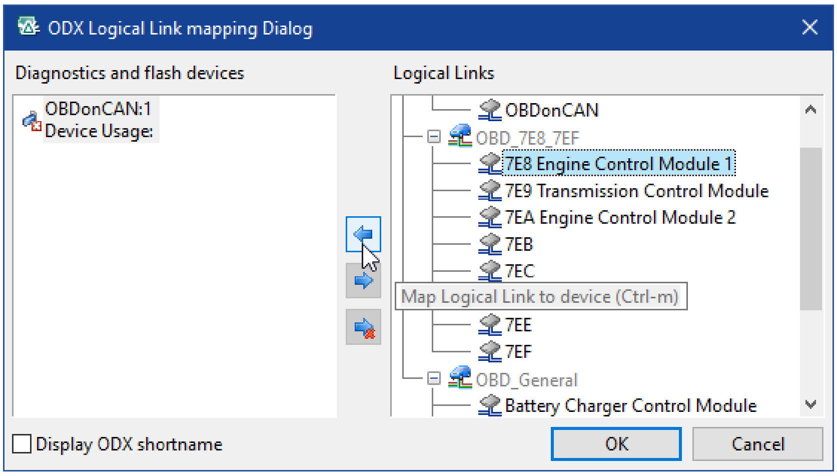
(29.06.2012 13:09:00 52 10)

Project name: OBDonCAN_ETAS

ODX Revision history

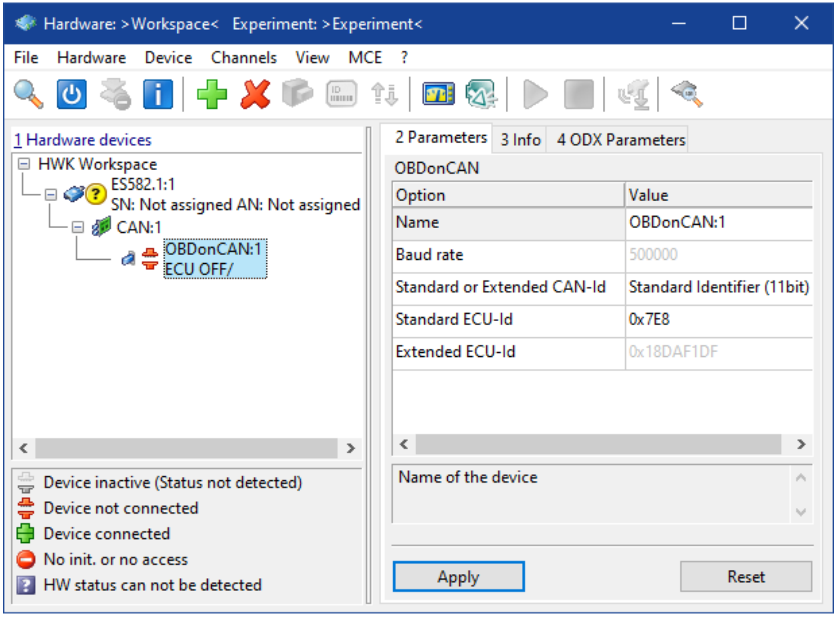
Revision date: 16.04.2012 10:36:00
Company: ETAS GmbH
Author: ODX-LINK Product Manager
Roles: OBD Database Admin
Address: Borsigstrasse 14
City: 70469 Stuttgart

The "ODX Logical Link mapping Dialog" window opens.


ODX Logical Link mapping Dialog
Diagnostics and flash devices
OBDonCAN:1
Device Usage:
Logical Links
OBDonCAN
OBD_7E8_7EF
7E8 Engine Control Module 1
7E9 Transmission Control Module
7EA Engine Control Module 2
7EB
7EC
Map Logical Link to device (Ctrl-m)
7EE
7EF
OBD_General
Battery Charger Control Module
☐ Display ODX shortname
OK
Cancel

7. Select the required logical link and assign it to the diagnostic device by clicking the arrow pointing left.
8. Click **OK**.

Parameters such as CAN-ID (in brackets beside the device name) and baud rate are generated and configured automatically.



Option	Value
Name	OBDonCAN:1
Baud rate	500000
Standard or Extended CAN-Id	Standard Identifier (11bit)
Standard ECU-Id	0x7E8
Extended ECU-Id	0x18DAF1DF

9. Select **Hardware** > **Initialize Hardware** to test the parameters set.

   Once successfully initialized, hardware configuration is completed.

> **Note**
>
> Like every INCA device, an "OBDonCAN" device can only communicate with one ECU at a time. If you need to communicate with several ECUs simultaneously, you have to create and configure an individual OBD device for each one.

10. Close the "Hardware ›OBDLesson‹" window.

To add an experiment

1. Select **Edit** > **Add** > **Experiment**.
2. Enter "OBDExperiment" and press ENTER.
3. Double-click "OBDExperiment".

   A dialog box for selecting the workspace opens.
4. Select "OBDLesson" and click **OK**.

   The INCA experiment window opens.

To send OBD Service requests with ODX-LINK

1. Select **ODX** > **User views** > **Diagnostic Services**.

   The diagnostic services dialog box containing all the OBD services is displayed.
2. Click **Configure...**.

   The "Diagnostic Services Configuration" window for configuring the OBD service IDs display opens.

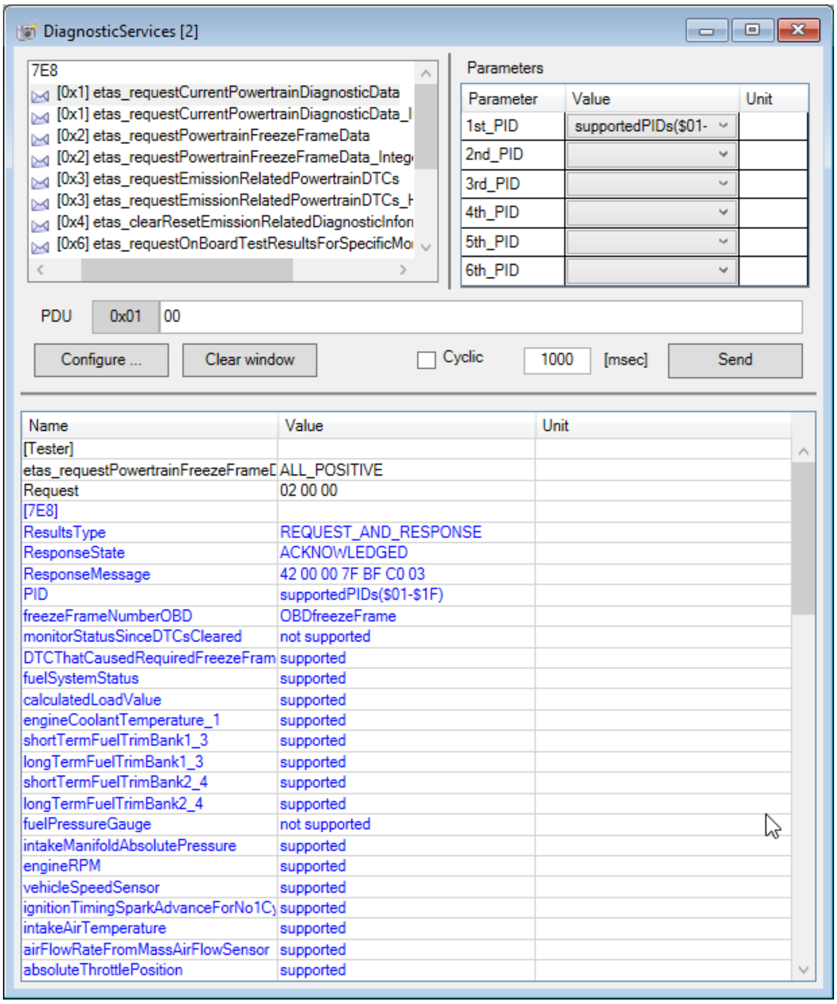
3. Check the "Show service ID in tree view" check box and click **OK**.

   The diagnostic services dialog box displays the service IDs.

4. Select the service "[0x1] etas_requestCurrentPowertrainDiagnosticData".

5. Select the PID "supportedPIDs($01-$1F)" and click **Send**.

   The service is sent and the response of the ECU shown in the output field.



7E8
[0x1] etas_requestCurrentPowertrainDiagnosticData
[0x1] etas_requestCurrentPowertrainDiagnosticData_I
[0x2] etas_requestPowertrainFreezeFrameData
[0x2] etas_requestPowertrainFreezeFrameData_Integ
[0x3] etas_requestEmissionRelatedPowertrainDTCs
[0x3] etas_requestEmissionRelatedPowertrainDTCs_H
[0x4] etas_clearResetEmissionRelatedDiagnosticInfor
[0x6] etas_requestOnBoardTestResultsForSpecificMo

Parameters

Parameter	Value	Unit
1st_PID	supportedPIDs($01-	
2nd_PID		
3rd_PID		
4th_PID		
5th_PID		
6th_PID		

PDU 0x01 00

□ Cyclic 1000 [msec]

Name	Value	Unit
[Tester]		
etas_requestPowertrainFreezeFrameD	ALL_POSITIVE	
Request	02 00 00	
[7E8]		
ResultsType	REQUEST_AND_RESPONSE	
ResponseState	ACKNOWLEDGED	
ResponseMessage	42 00 00 7F BF C0 03	
PID	supportedPIDs($01-$1F)	
freezeFrameNumberOBD	OBDfreezeFrame	
monitorStatusSinceDTCsCleared	not supported	
DTCThatCausedRequiredFreezeFram	supported	
fuelSystemStatus	supported	
calculatedLoadValue	supported	
engineCoolantTemperature_1	supported	
shortTermFuelTrimBank1_3	supported	
longTermFuelTrimBank1_3	supported	
shortTermFuelTrimBank2_4	supported	
longTermFuelTrimBank2_4	supported	
fuelPressureGauge	not supported	
intakeManifoldAbsolutePressure	supported	
engineRPM	supported	
vehicleSpeedSensor	supported	
ignitionTimingSparkAdvanceForNo1Cy	supported	
intakeAirTemperature	supported	
airFlowRateFromMassAirFlowSensor	supported	
absoluteThrottlePosition	supported	

Note

Your ECU may only support some of the possible PIDs and OBD services. If you select a PID or service which is not supported, the ECU may not respond at all resulting in a timeout error message in the INCA Monitor window.

Note

ODX-LINK makes ODX configurations available for OBDonCAN, OBDonUDS, and the ODX test device as part of INCA export files in the *ETASData\INCA7.5\Export\ODX* folder. The ODX-LINK window configurations contained correspond to the relevant diagnostic databases and can be used immediately.

5.7 Using OBDonUDS (SAE J1979-2) with ODX-LINK

In this lesson, you will configure INCA for using the OBD protocol SAE J1979-2 with the OBDonUDS implementation and you will use the ODX-LINK OBD database for OBD diagnostic communication with a real ECU.

> **Note**
>
> You can only do this lesson if you have connected one of the hardware components supported by ODX-LINK (e.g. ES592, ES595, ES582, ES910) and an ECU.

For this lesson, you need an ECU which supports OBD communication in accordance with SAE J1979-2 and ISO15765-4 on CAN..

To configure INCA for OBDonUDS

1. Start INCA.
2. Create a new top folder with **Edit** > **Add** > **Add top folder**.
3. Enter "OBD Tutorial" as top folder name and press ENTER.
4. Add a new workspace using **Edit** > **Add** > **Workspace**.
5. Enter "OBDLesson" as the name and press ENTER.

To add an ODX project for OBD

1. Change to the "INCA" window and select "OBD Tutorial".
2. Select **Edit** > **Add** > **ODX Project**.
3. A file selector window opens for selecting the project or ODX/PDX file(s).
4. Navigate to the `\ETASData\ODX7.5\Projects` folder.
5. Select the file `OBDonUDS_ETAS_SAEJ1979<Version>.pdx`.
6. Click **Open**.

   The ODX project is added.

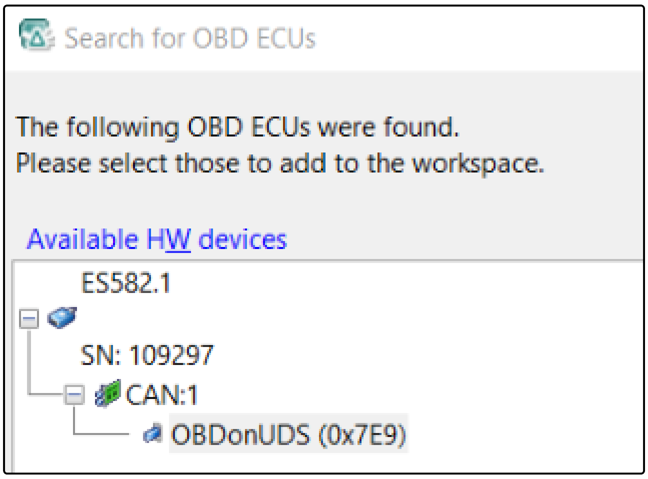
To configure hardware

1. Select the "OBDLesson" workspace.
2. Select **Device** > **Configure Hardware**.

   The hardware configuration window opens.
3. Select **Hardware** > **Search for OBD ECUs**.
4. From the list, select the interface types to be searched for and click **OK**.

   A search takes place for connected hardware with OBDonCAN and OBDonUDS support.

   The result is shown in the "Search for OBD ECUs" window and can be selected there.



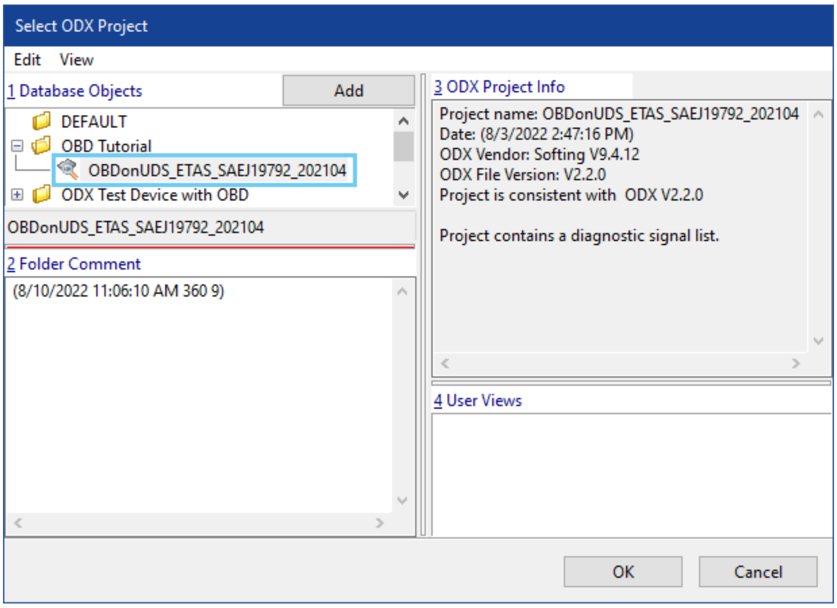
Search for OBD ECUs

The following OBD ECUs were found.
Please select those to add to the workspace.

Available HW devices

ES582.1
SN: 109297
CAN:1
OBDonUDS (0x7E9)

5. Select an ODBonUDS ECU found and click **OK**.
6. Select the corresponding ODX project from the following window and click **OK**.



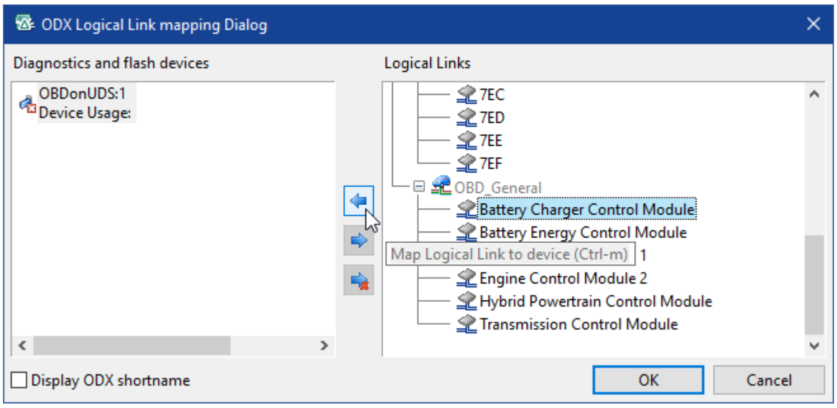
OBDonUDS_ETAS_SAEJ19792_202104

(8/10/2022 11:06:10 AM 360 9)

Project name: OBDonUDS_ETAS_SAEJ19792_202104
Date: (8/3/2022 2:47:16 PM)
ODX Vendor: Softing V9.4.12
ODX File Version: V2.2.0
Project is consistent with ODX V2.2.0

Project contains a diagnostic signal list.

The "ODX Logical Link mapping Dialog" window opens.

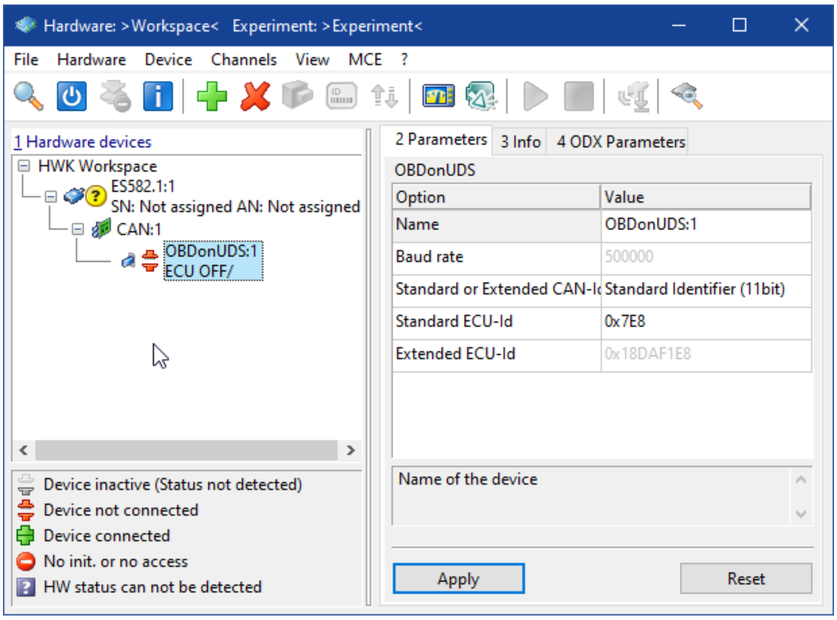


ODX Logical Link mapping Dialog
Diagnostics and flash devices
Logical Links
OBDonUDS:1
Device Usage:
7EC
7ED
7EE
7EF
OBD_General
Battery Charger Control Module
Battery Energy Control Module
Map Logical Link to device (Ctrl-m)
1
Engine Control Module 2
Hybrid Powertrain Control Module
Transmission Control Module
Display ODX shortname
OK
Cancel

7. Select the required logical link and assign it to the diagnostic device by clicking the arrow pointing left.

8. Click **OK**.

   Parameters such as CAN-ID (in brackets beside the device name) and baud rate are generated and configured automatically.



OBDonUDS

Option	Value
Name	OBDonUDS:1
Baud rate	500000
Standard or Extended CAN-Id	Standard Identifier (11bit)
Standard ECU-Id	0x7E8
Extended ECU-Id	0x18DAF1E8

Name of the device

- Device inactive (Status not detected)
- Device not connected
- Device connected
- No init. or no access
- HW status can not be detected

9. Select **Hardware** > **Initialize Hardware** to test the parameters set.

   Once successfully initialized, hardware configuration is completed.

   > **Note**
   >
   > Like every INCA device, an "OBDonCAN" device can only communicate with one ECU at a time. If you need to communicate with several ECUs simultaneously, you have to create and configure an individual OBD device for each one.

10. Close the "Hardware ›OBDLesson‹" window.

To add an experiment

1. Select **Edit** > **Add** > **Experiment**.
2. Enter "OBDExperiment" and press ENTER.
3. Double-click "OBDExperiment".

   A dialog box for selecting the workspace opens.
4. Select "OBDLesson" and click **OK**.

   The INCA experiment window opens.

To send OBD Service requests with ODX-LINK

1. Select **ODX** > **User views** > **Diagnostic Services**.

   The diagnostic services dialog box containing all the OBD services is displayed.
2. Click **Configure...**.

The "Diagnostic Services Configuration" window for configuring the OBD service IDs display opens.

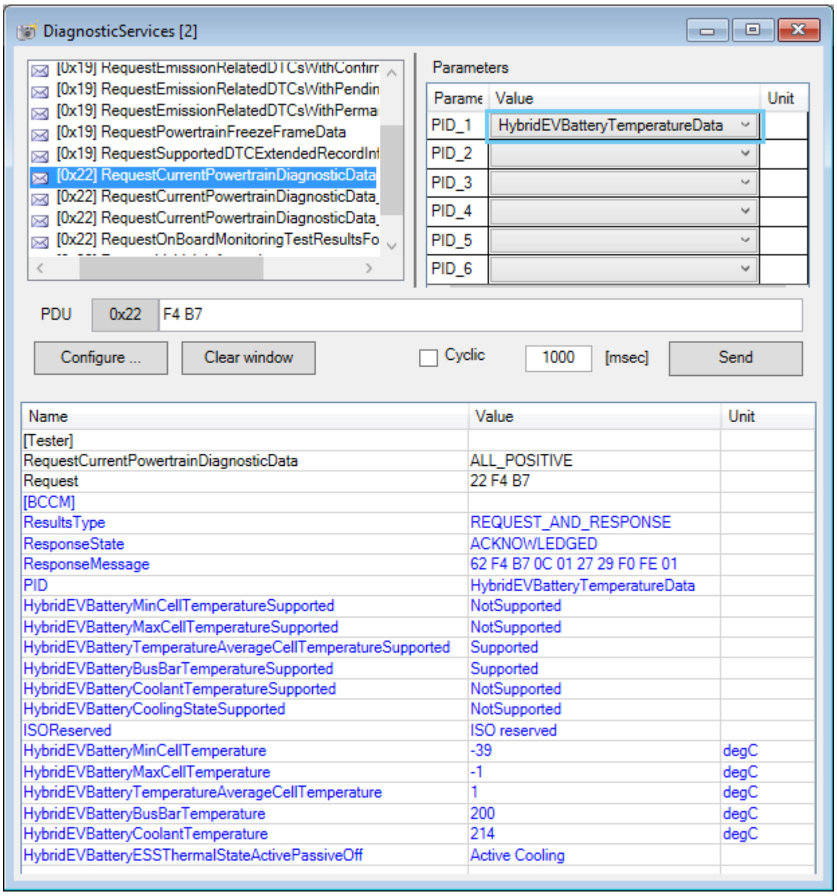
3. Check the "Show service ID in tree view" check box and click **OK**.

   The diagnostic services dialog box displays the service IDs.

4. Select the service "RequestCurrentPowertrainDiagnosticData".

5. Select the PID "HybridEVBatteryTemperatureData" and click **Send**.

   The service is sent and the response of the ECU shown in the output field.



DiagnosticServices [2]

- [0x19] RequestEmissionRelatedDTCsWithConfirr
- [0x19] RequestEmissionRelatedDTCsWithPendin
- [0x19] RequestEmissionRelatedDTCsWithPerma
- [0x19] RequestPowertrainFreezeFrameData
- [0x19] RequestSupportedDTCExtendedRecordInf
- [0x22] RequestCurrentPowertrainDiagnosticData
- [0x22] RequestCurrentPowertrainDiagnosticData_
- [0x22] RequestCurrentPowertrainDiagnosticData_
- [0x22] RequestOnBoardMonitoringTestResultsFo

Parameters

Parame	Value	Unit
PID_1	HybridEVBatteryTemperatureData	
PID_2		
PID_3		
PID_4		
PID_5		
PID_6		

PDU 0x22 F4 B7

Configure ... Clear window □ Cyclic 1000 [msec] Send

Name	Value	Unit
[Tester]		
RequestCurrentPowertrainDiagnosticData	ALL_POSITIVE	
Request	22 F4 B7	
[BCCM]		
ResultsType	REQUEST_AND_RESPONSE	
ResponseState	ACKNOWLEDGED	
ResponseMessage	62 F4 B7 0C 01 27 29 F0 FE 01	
PID	HybridEVBatteryTemperatureData	
HybridEVBatteryMinCellTemperatureSupported	NotSupported	
HybridEVBatteryMaxCellTemperatureSupported	NotSupported	
HybridEVBatteryTemperatureAverageCellTemperatureSupported	Supported	
HybridEVBatteryBusBarTemperatureSupported	Supported	
HybridEVBatteryCoolantTemperatureSupported	NotSupported	
HybridEVBatteryCoolingStateSupported	NotSupported	
ISOReserved	ISO reserved	
HybridEVBatteryMinCellTemperature	-39	degC
HybridEVBatteryMaxCellTemperature	-1	degC
HybridEVBatteryTemperatureAverageCellTemperature	1	degC
HybridEVBatteryBusBarTemperature	200	degC
HybridEVBatteryCoolantTemperature	214	degC
HybridEVBatteryESSThermalStateActivePassiveOff	Active Cooling	

Note

Your ECU may only support some of the possible PIDs and OBD services. If you select a PID or service which is not supported, the ECU may not respond at all resulting in a timeout error message in the INCA Monitor window.

Note

ODX-LINK makes ODX configurations available for OBDonCAN, OBDonUDS, and the ODX test device as part of INCA export files in the `ETASData\INCA7.5\Export\ODX` folder. The ODX-LINK window configurations contained correspond to the relevant diagnostic databases and can be used immediately.

5.8 Working with the "OBD" User View

In this lesson, you will work with the OBD user view.

> ⓘ **Note**
>
> From INCA V7.4.4, you can use the the OBD window also for OBDonUDS and for the ODX projekt `ETASData\ODX7.4\Projects\OBDonUDS_ETAS_SAEJ1979<Version>.pdx`.

You will use the results from lessons "Creating an INCA Workspace " on page 73 and "Preparing an INCA Experiment for ODX without Hardware Connection" on page 76 for this purpose, but have to use a different ODX project (see "User Views" on page 19).

To add an OBD-ODX project

1. Change to INCA.
2. Open the "ODXTutorial" folder.
3. Select **Edit** > **Add** > **ODX Project**.

   A file selector window opens for selecting the project or ODX/PDX file(s).
4. Navigate to the `\ETASData\ODX7.5\Projects` folder.
5. Select the file `OBDonCAN_ETAS_SAEJ1979<Version>.pdx`.
6. Click **Open**.

   The ODX project is added.

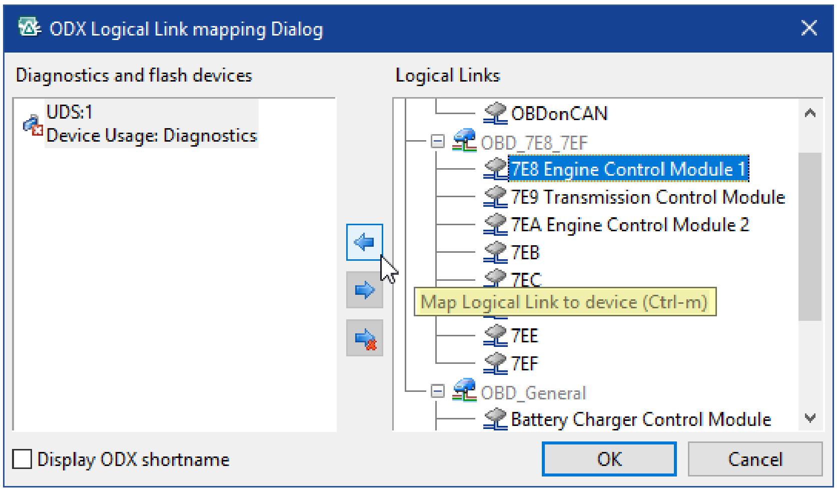
To add an ODX configuration

1. Select the "Lesson1" workspace.
2. Select **Device** > **Configure hardware**.

   The hardware configuration window opens.
3. Select **Hardware** > **Configure ODX**.

   The window for selecting the ODX project opens.
4. Select the ODX project "OBDonCAN_ETAS" and click **OK**.

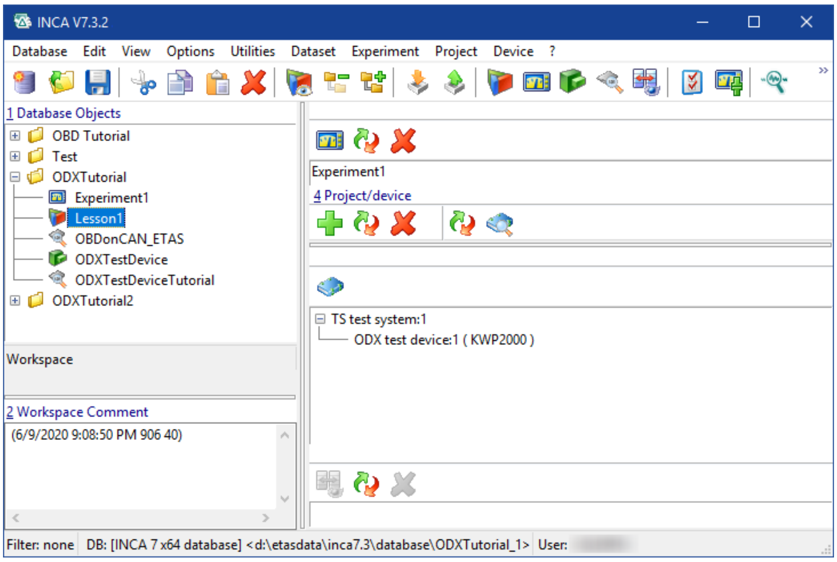
   The logical link mapping window opens.
5. Select the logical link "7E8" and add it to the ODX test device by clicking on the button with the arrow pointing left.


ODX Logical Link mapping Dialog
Diagnostics and flash devices
UDS:1
Device Usage: Diagnostics
Logical Links
OBDonCAN
OBD_7E8_7EF
7E8 Engine Control Module 1
7E9 Transmission Control Module
7EA Engine Control Module 2
7EB
7EC
Map Logical Link to device (Ctrl-m)
7EE
7EF
OBD_General
Battery Charger Control Module
□ Display ODX shortname
OK
Cancel

6. Click **OK**.
7. Close the hardware configuration window.
8. Select **Database** > **Save** from the main INCA window.

To query vehicle information and DTCs

1. Double-click the workspace `Lesson1`.

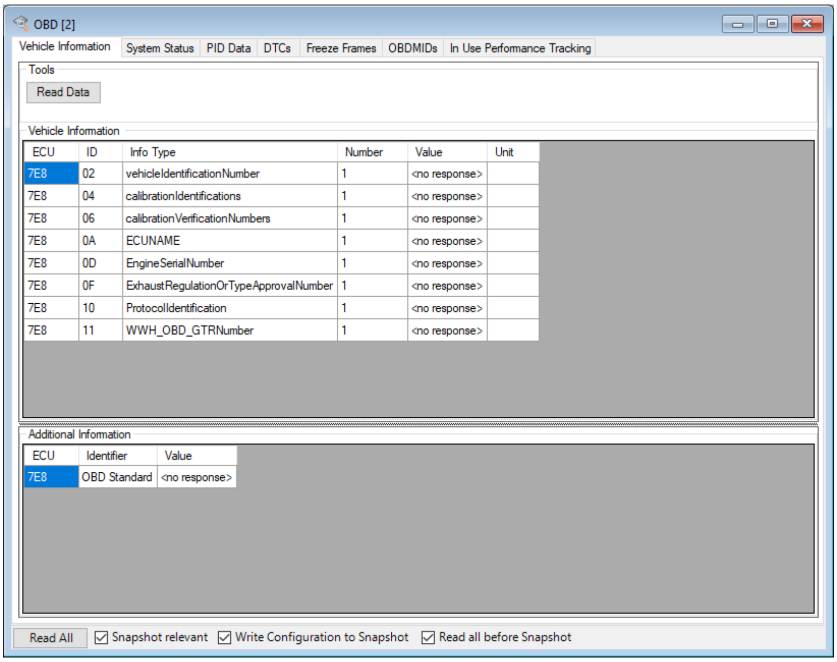

INCA V7.3.2
Database Edit View Options Utilities Dataset Experiment Project Device ?
1 Database Objects
OBD Tutorial
Test
ODXTutorial
Experiment1
Lesson1
OBDonCAN_ETAS
ODXTestDevice
ODXTestDeviceTutorial
ODXTutorial2
Workspace
2 Workspace Comment
(6/9/2020 9:08:50 PM 906 40)
Experiment1
4 Project/device
TS test system:1
ODX test device:1 (KWP2000)
Filter: none DB: [INCA 7 x64 database] <d:\etasdata\inca7.3\database\ODXTutorial_1> User:

The experiment is opened.

2. Select **ODX** > **User views** > **OBD**.

   The "OBD" user view opens.

3. Select the "Vehicle Information" tab.



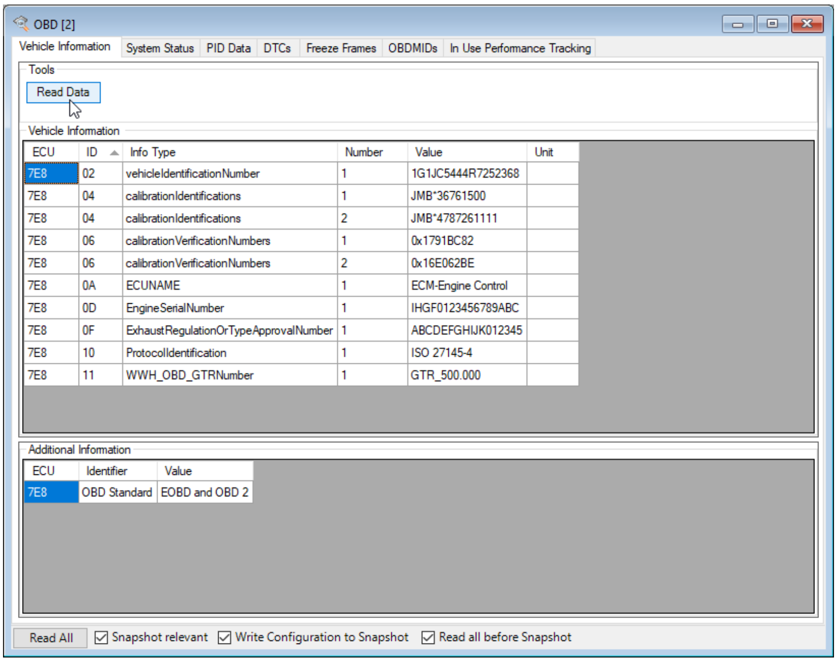
ECU	ID	Info Type	Number	Value	Unit
7E8	02	vehicleIdentificationNumber	1	<no response>	
7E8	04	calibrationIdentifications	1	<no response>	
7E8	06	calibrationVerificationNumbers	1	<no response>	
7E8	0A	ECUNAME	1	<no response>	
7E8	0D	EngineSerialNumber	1	<no response>	
7E8	0F	ExhaustRegulationOrTypeApprovalNumber	1	<no response>	
7E8	10	ProtocolIdentification	1	<no response>	
7E8	11	WWH_OBD_GTRNumber	1	<no response>	

ECU	Identifier	Value
7E8	OBD Standard	<no response>

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

4. Click **Read Data**.

   The information is queried and displayed in the "Vehicle Information" field.



ECU	ID	Info Type	Number	Value	Unit
7E8	02	vehicleIdentificationNumber	1	1G1JC5444R7252368	
7E8	04	calibrationIdentifications	1	JMB*36761500	
7E8	04	calibrationIdentifications	2	JMB*4787261111	
7E8	06	calibrationVerificationNumbers	1	0x1791BC82	
7E8	06	calibrationVerificationNumbers	2	0x16E062BE	
7E8	0A	ECUNAME	1	ECM-Engine Control	
7E8	0D	EngineSerialNumber	1	IHGF0123456789ABC	
7E8	0F	ExhaustRegulationOrTypeApprovalNumber	1	ABCDEFGHIJK012345	
7E8	10	ProtocolIdentification	1	ISO 27145-4	
7E8	11	WWH_OBD_GTRNumber	1	GTR_500.000	

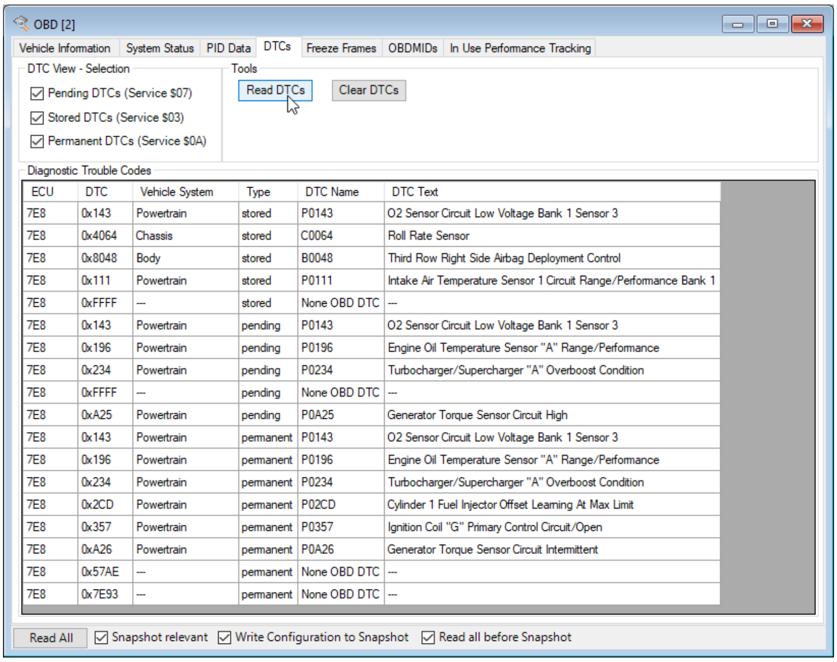
ECU	Identifier	Value
7E8	OBD Standard	EOBD and OBD 2

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

   In the "Additional Information" field, additional information on the OBD requirements of the vehicle (PID $1C) are displayed.

5. Select the "DTCs" tab.
6. Click **Read DTCs**.

   The DTCs are displayed in the "Diagnostic Trouble Codes" field.



☑ Pending DTCs (Service $07)
☑ Stored DTCs (Service $03)
☑ Permanent DTCs (Service $0A)

ECU	DTC	Vehicle System	Type	DTC Name	DTC Text
7E8	0x143	Powertrain	stored	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x4064	Chassis	stored	C0064	Roll Rate Sensor
7E8	0x8048	Body	stored	B0048	Third Row Right Side Airbag Deployment Control
7E8	0x111	Powertrain	stored	P0111	Intake Air Temperature Sensor 1 Circuit Range/Performance Bank 1
7E8	0xFFFF	---	stored	None OBD DTC	---
7E8	0x143	Powertrain	pending	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x196	Powertrain	pending	P0196	Engine Oil Temperature Sensor "A" Range/Performance
7E8	0x234	Powertrain	pending	P0234	Turbocharger/Supercharger "A" Overboost Condition
7E8	0xFFFF	---	pending	None OBD DTC	---
7E8	0xA25	Powertrain	pending	P0A25	Generator Torque Sensor Circuit High
7E8	0x143	Powertrain	permanent	P0143	O2 Sensor Circuit Low Voltage Bank 1 Sensor 3
7E8	0x196	Powertrain	permanent	P0196	Engine Oil Temperature Sensor "A" Range/Performance
7E8	0x234	Powertrain	permanent	P0234	Turbocharger/Supercharger "A" Overboost Condition
7E8	0x2CD	Powertrain	permanent	P02CD	Cylinder 1 Fuel Injector Offset Learning At Max Limit
7E8	0x357	Powertrain	permanent	P0357	Ignition Coil "G" Primary Control Circuit/Open
7E8	0xA26	Powertrain	permanent	P0A26	Generator Torque Sensor Circuit Intermittent
7E8	0x57AE	---	permanent	None OBD DTC	---
7E8	0x7E93	---	permanent	None OBD DTC	---

☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

5.9 Using Diagnostic Signal in the Experiment

In this lesson you will learn how diagnostic signals can be used in an experiment.

The diagnostic signals are selected in the variable selection dialog and are indicated in the experiment in measuring windows; a measurement is recorded and viewed in the MDA. For this, use the results from Lesson "Working with the "OBD" User View" on page 97.

Opening the INCA Experiment

1. Open the folder "ODXTutorial".
2. Double-click the workspace "Lesson1".

   The INCA experiment is opened. In the next step select the diostic signals in the variable selection dialog and assign them to different measuring windows.

Selecting diagnostic signals in the variable selection dialog

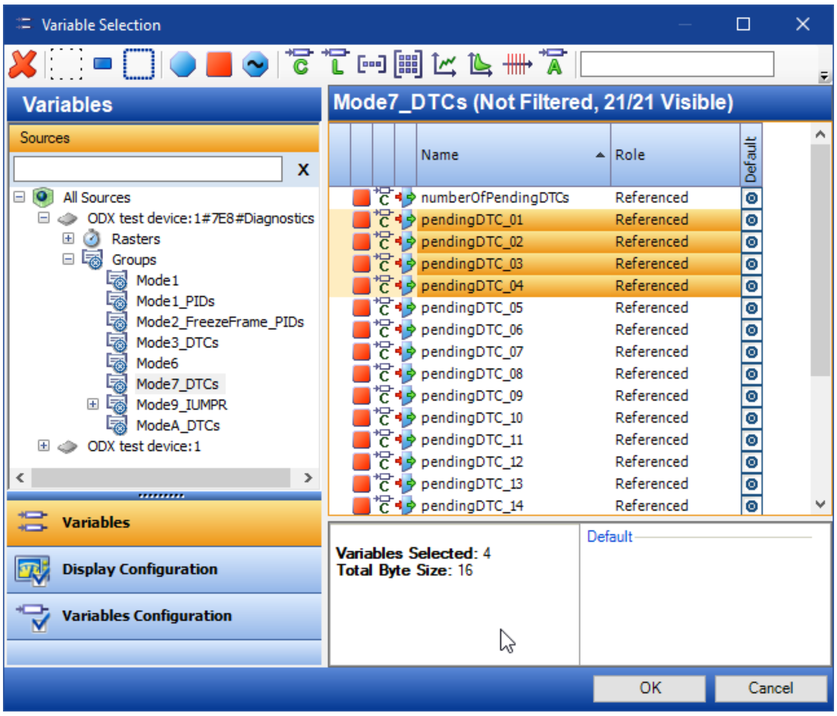
1. Select the **Variables** > **Variable Selection**.

   The variable selection dialog is opened.

2. Select the source
   ODX test device:1#7E8#Diagnostics
   and open the tree structure.
3. Select under *Groups* the group *Mode7_DTCs*.

   The diagnostic signals of the group *Mode7_DTCs* will be displayed.

4. Select the diagnostic signals *pendingDTC_01- pendingDTC_04*.



Name	Role
numberOfPendingDTCs	Referenced
pendingDTC_01	Referenced
pendingDTC_02	Referenced
pendingDTC_03	Referenced
pendingDTC_04	Referenced
pendingDTC_05	Referenced
pendingDTC_06	Referenced
pendingDTC_07	Referenced
pendingDTC_08	Referenced
pendingDTC_09	Referenced
pendingDTC_10	Referenced
pendingDTC_11	Referenced
pendingDTC_12	Referenced
pendingDTC_13	Referenced
pendingDTC_14	Referenced

Variables Selected: 4
Total Byte Size: 16

5. Select in the context menu **Add to** > **Layer_1** > **New** > **Measure Table**.
6. Click **OK**.

   The diagnostic signals are displayed in the experiment in a measuring table.

Indicating diagnostic signals

1. Click the **Start Visualization** symbol. ▶

   The diagnostic signals are displayed in the measuring table.

2. Click the **Stop Measuring** symbol. ■

   The display will be stopped. In the next step, the diagnostic signals are recorded in the experiment.

> **Note**
>
> The indicated values of the diagnostic signals are simulated by the use of the ODX-Test-Device and, hence, remain constant.

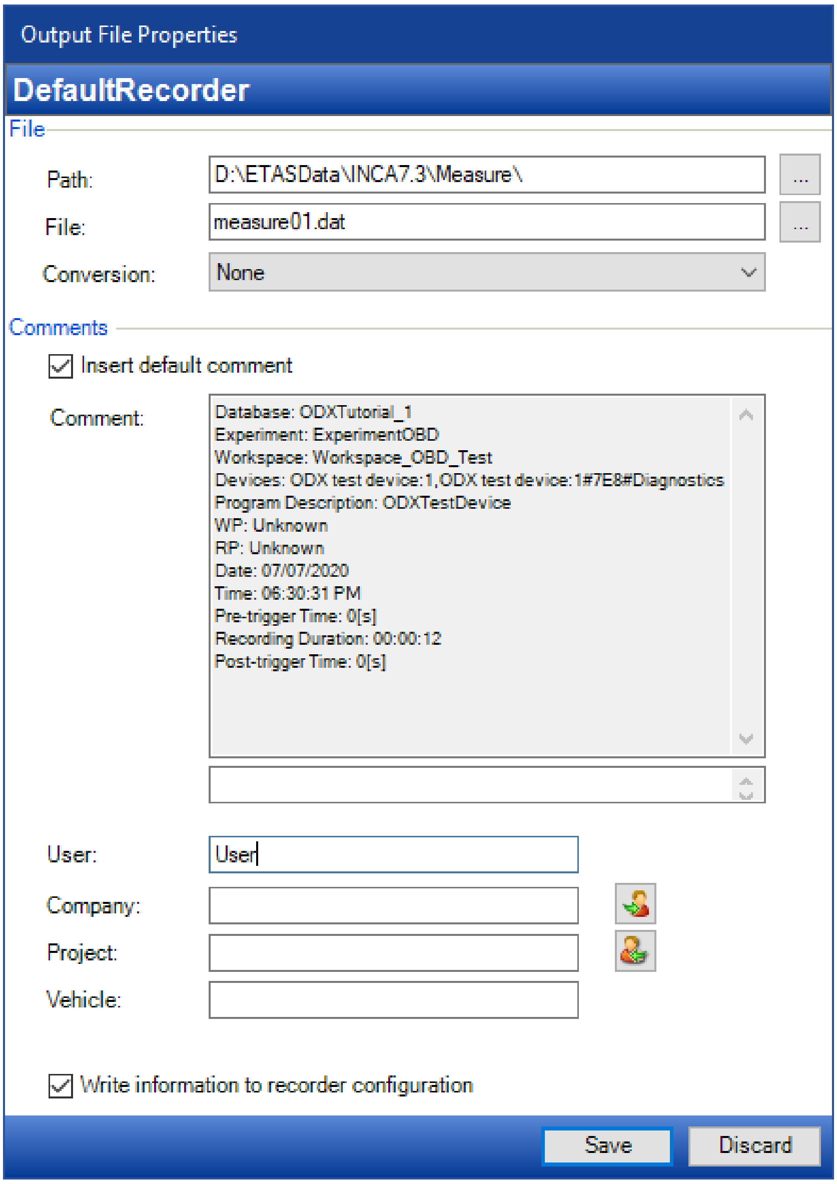
Recording diagnostic signals

1. Click the **StartRecording** symbol. ●

   The measurement will be recorded with a standard recorder.

2. Wait for approx. 10 seconds.
3. Click the **Stop Measuring** symbol. ■

   The dialog window **Output File Properties** is opened.



Path: D:\ETASData\INCA7.3\Measure\

File: measure01.dat

Conversion: None

☑ Insert default comment

Comment:

```
Database: ODXTutorial_1
Experiment: ExperimentOBD
Workspace: Workspace_OBD_Test
Devices: ODX test device:1,ODX test device:1#7E8#Diagnostics
Program Description: ODXTestDevice
WP: Unknown
RP: Unknown
Date: 07/07/2020
Time: 06:30:31 PM
Pre-trigger Time: 0[s]
Recording Duration: 00:00:12
Post-trigger Time: 0[s]
```

User: User

Company:

Project:

Vehicle:

☑ Write information to recorder configuration

4. Click **Save**.

   The measurement file is stored.

5.10 Measuring OBD Data on the Vehicle

In this lesson, you measure diagnostic signals on the vehicle and generate additional ECU-specific measurement signals.

To create a main directory and workspace

1. Create a main directory called "OBD Tutorial2".
2. Create an "OBDLesson2" workspace in this directory.

To import the ODX project

Add a diagnostic project (`\ETASData\ODX7.5\Projects\OBDonCAN_ETAS_SAEJ1979<Version>.pdx`)

To configure the hardware

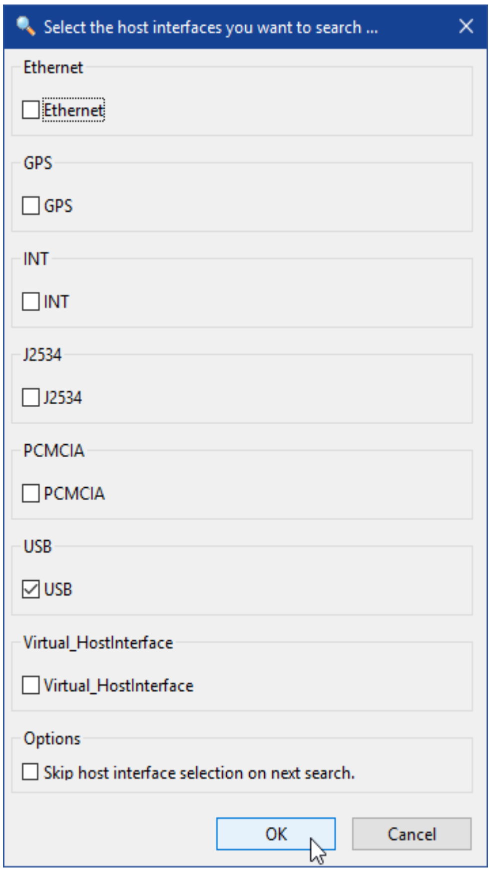
1. Select the "OBDLesson2" workspace.
2. Select **Device** > **Configure hardware**.

   The Hardware Configuration Editor opens.

   Note

   Make sure that your measurement hardware (ES581, ES595, etc.) is connected to the ECU or vehicle via a CAN port!

3. Select **Hardware** > **Search for OBD ECUs**.
4. In the following dialog box, select the interfaces where you want to search for hardware (e.g. Ethernet, USB etc.).



Select the host interfaces you want to search ...

Ethernet
□ Ethernet

GPS
□ GPS

INT
□ INT

J2534
□ J2534

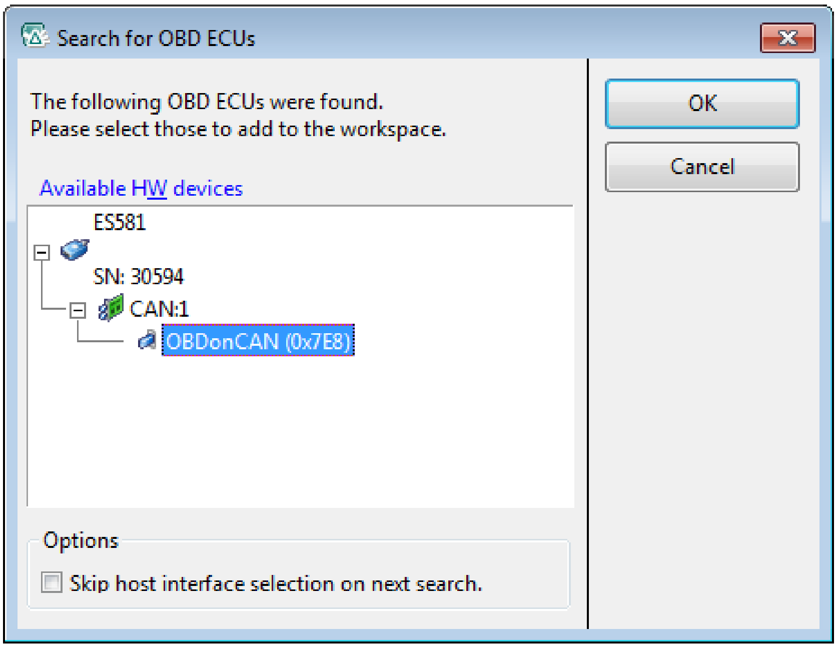
PCMCIA
□ PCMCIA

USB
☑ USB

Virtual_HostInterface
□ Virtual_HostInterface

Options
□ Skip host interface selection on next search.

The list of OBD ECUs found is displayed.



Search for OBD ECUs

The following OBD ECUs were found.
Please select those to add to the workspace.

Available HW devices

- ES581 SN: 30594
  - CAN:1
    - OBDonCAN (0x7E8)

Options

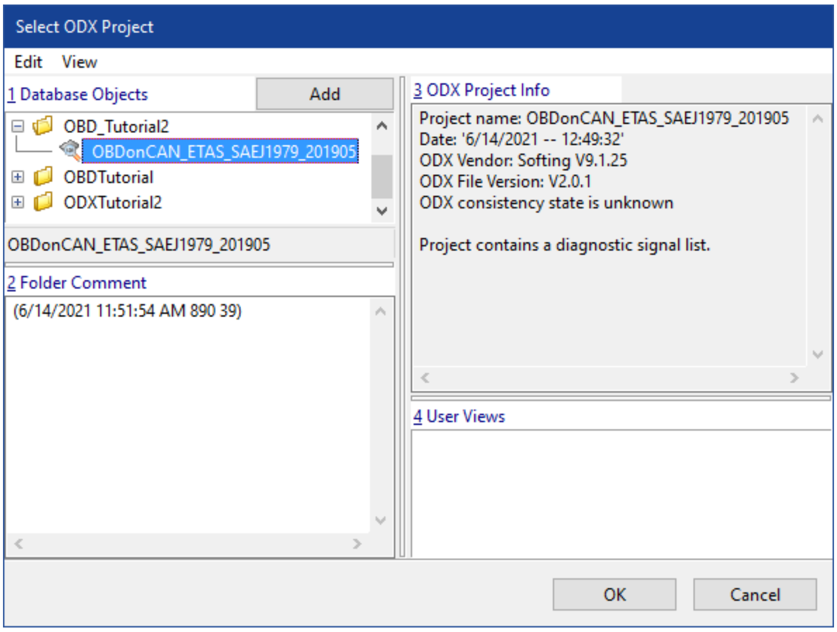
□ Skip host interface selection on next search.

5. Select the OBD ECU that you want to use for hardware configuration and click **OK**.

   The relevant OBDonCAN devices are created and configured automatically.

> **Note**
>
> If no ECU is found, the ECU or the vehicle either does not support OBDonCAN or is not connected correctly with the hardware.

6. Select the ODX project `OBDonCAN_ETAS_SAEJ1979_2021-04`.



Select ODX Project

(6/14/2021 11:51:54 AM 890 39)

Project name: OBDonCAN_ETAS_SAEJ1979_201905
Date: '6/14/2021 -- 12:49:32'
ODX Vendor: Softing V9.1.25
ODX File Version: V2.0.1
ODX consistency state is unknown

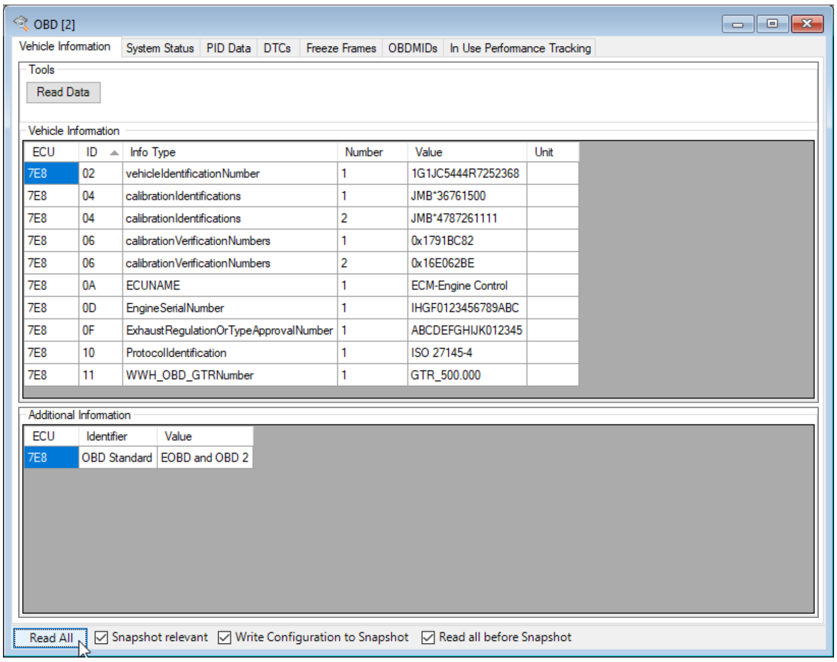
Project contains a diagnostic signal list.

7. In the "ODX Logical Link mapping Dialog" assign every OBDonCAN device from the hardware configuration a corresponding logical link from the ODX project (corresponding to the CAN-ID in the device name and in the name of the logical link) and click **OK**.
8. Close the Hardware Configuration Editor.

To create an experiment and open OBD windows

1. Create an experiment "OBDExperiment2".
2. Open the experiment and select the "OBDLesson2" workspace.
3. Select **ODX** > **User views** > **OBD**.

   The "OBD" user view opens.
4. Click **Read All** to read out all supported OBD data from the connected ECUs once.



Vehicle Information

ECU	ID	Info Type	Number	Value	Unit
7E8	02	vehicleIdentificationNumber	1	1G1JC5444R7252368	
7E8	04	calibrationIdentifications	1	JMB*36761500	
7E8	04	calibrationIdentifications	2	JMB*4787261111	
7E8	06	calibrationVerificationNumbers	1	0x1791BC82	
7E8	06	calibrationVerificationNumbers	2	0x16E062BE	
7E8	0A	ECUNAME	1	ECM-Engine Control	
7E8	0D	EngineSerialNumber	1	IHGF0123456789ABC	
7E8	0F	ExhaustRegulationOrTypeApprovalNumber	1	ABCDEFGHIJK012345	
7E8	10	ProtocolIdentification	1	ISO 27145-4	
7E8	11	WWH_OBD_GTRNumber	1	GTR_500.000	

Additional Information

ECU	Identifier	Value
7E8	OBD Standard	EOBD and OBD 2

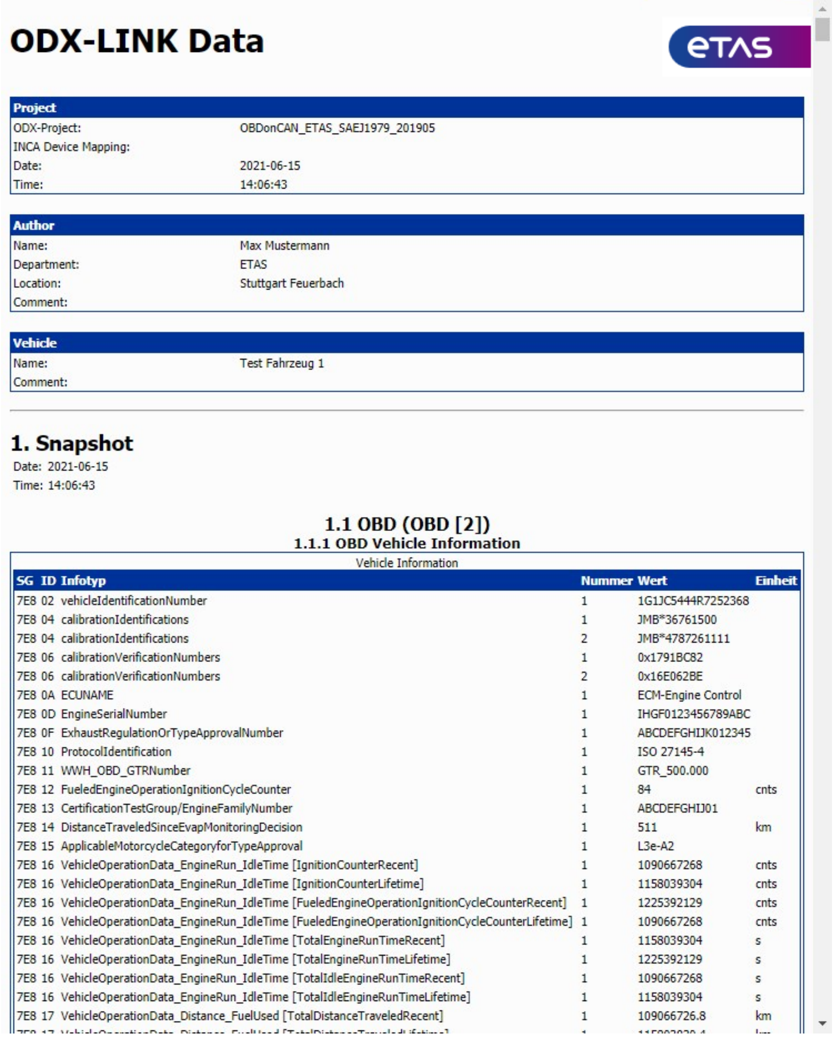
☑ Snapshot relevant ☑ Write Configuration to Snapshot ☑ Read all before Snapshot

5. Select **ODX** > **Snapshot**.

or

   Click the **snapshot** icon.
6. Enter the required data and options in the "DataLogging Configuration" window in the tabs "File" and "Header" (see "Data Logging Configuration " on page 57).
7. Click **OK**.

   The snapshot is taken and opened as a text or HTML file in the relevant editors.



ODX-LINK Data

Project	
ODX-Project:	OBDonCAN_ETAS_SAEJ1979_201905
INCA Device Mapping:	
Date:	2021-06-15
Time:	14:06:43

Author	
Name:	Max Mustermann
Department:	ETAS
Location:	Stuttgart Feuerbach
Comment:	

Vehicle	
Name:	Test Fahrzeug 1
Comment:	

1. Snapshot

Date: 2021-06-15
Time: 14:06:43

1.1 OBD (OBD [2])

1.1.1 OBD Vehicle Information

Vehicle Information

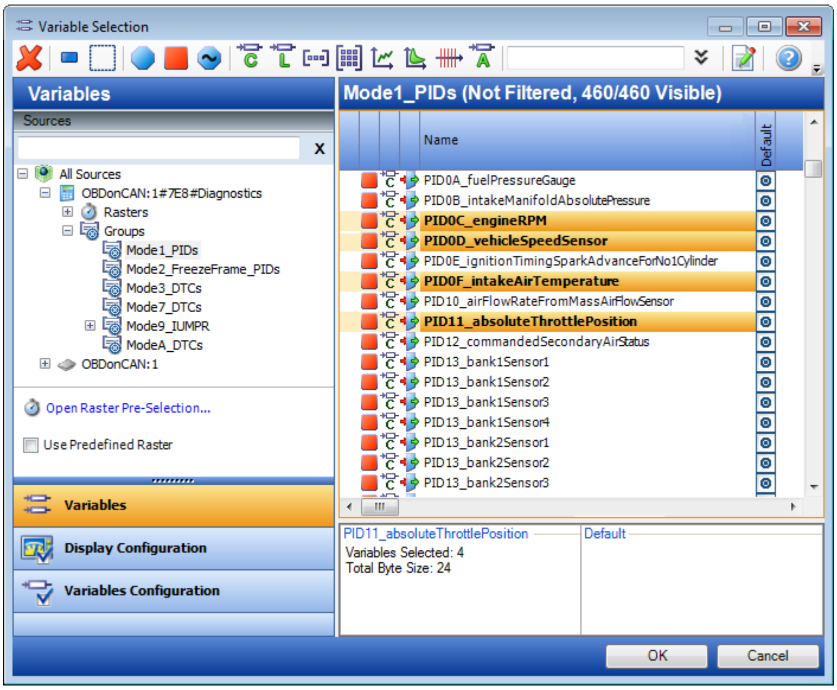
SG	ID	Infotyp	Nummer	Wert	Einheit
7E8	02	vehicleIdentificationNumber	1	1G1JC5444R7252368	
7E8	04	calibrationIdentifications	1	JMB*36761500	
7E8	04	calibrationIdentifications	2	JMB*4787261111	
7E8	06	calibrationVerificationNumbers	1	0x1791BC82	
7E8	06	calibrationVerificationNumbers	2	0x16E062BE	
7E8	0A	ECUNAME	1	ECM-Engine Control	
7E8	0D	EngineSerialNumber	1	IHGF0123456789ABC	
7E8	0F	ExhaustRegulationOrTypeApprovalNumber	1	ABCDEFGHIJK012345	
7E8	10	ProtocolIdentification	1	ISO 27145-4	
7E8	11	WWH_OBD_GTRNumber	1	GTR_500.000	
7E8	12	FueledEngineOperationIgnitionCycleCounter	1	84	cnts
7E8	13	CertificationTestGroup/EngineFamilyNumber	1	ABCDEFGHIJ01	
7E8	14	DistanceTraveledSinceEvapMonitoringDecision	1	511	km
7E8	15	ApplicableMotorcycleCategoryforTypeApproval	1	L3e-A2	
7E8	16	VehicleOperationData_EngineRun_IdleTime [IgnitionCounterRecent]	1	1090667268	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [IgnitionCounterLifetime]	1	1158039304	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [FueledEngineOperationIgnitionCycleCounterRecent]	1	1225392129	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [FueledEngineOperationIgnitionCycleCounterLifetime]	1	1090667268	cnts
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalEngineRunTimeRecent]	1	1158039304	s
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalEngineRunTimeLifetime]	1	1225392129	s
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalIdleEngineRunTimeRecent]	1	1090667268	s
7E8	16	VehicleOperationData_EngineRun_IdleTime [TotalIdleEngineRunTimeLifetime]	1	1158039304	s
7E8	17	VehicleOperationData_Distance_FuelUsed [TotalDistanceTraveledRecent]	1	109066726.8	km

To select measurement signals

1. Select **Variables** > **Variable selection**.

   The Variable Selection dialog box opens.

2. Select the signals to be measured (e.g. PIDs).

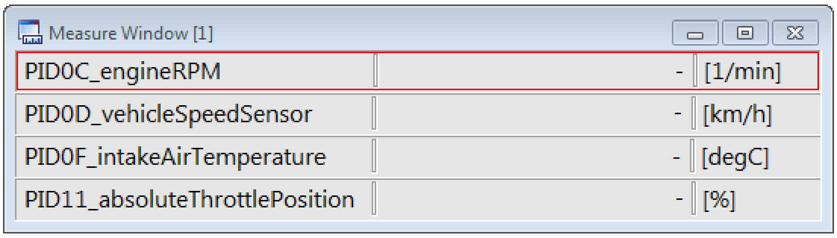


PID11_absoluteThrottlePosition
Variables Selected: 4
Total Byte Size: 24

Default

3. Close the Variable Selection dialog box with **OK**.

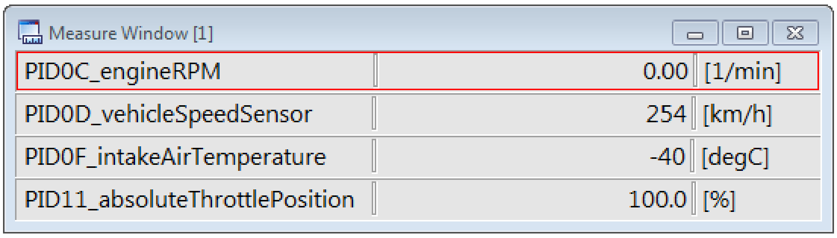
   The selected signals are shown in a measure window.



PID0C_engineRPM	-	[1/min]
PID0D_vehicleSpeedSensor	-	[km/h]
PID0F_intakeAirTemperature	-	[degC]
PID11_absoluteThrottlePosition	-	[%]

To start measuring

1. Select **Measurement** > **Start visualization**.
2. The measure values of the selected diagnostic signals are displayed.



PID0C_engineRPM	0.00	[1/min]
PID0D_vehicleSpeedSensor	254	[km/h]
PID0F_intakeAirTemperature	-40	[degC]
PID11_absoluteThrottlePosition	100.0	[%]

3. Stop measuring with **Measurement** > **Stop Measuring**.

To generate additional ECU-specific measurement signals for the OBD Mode 6 test results

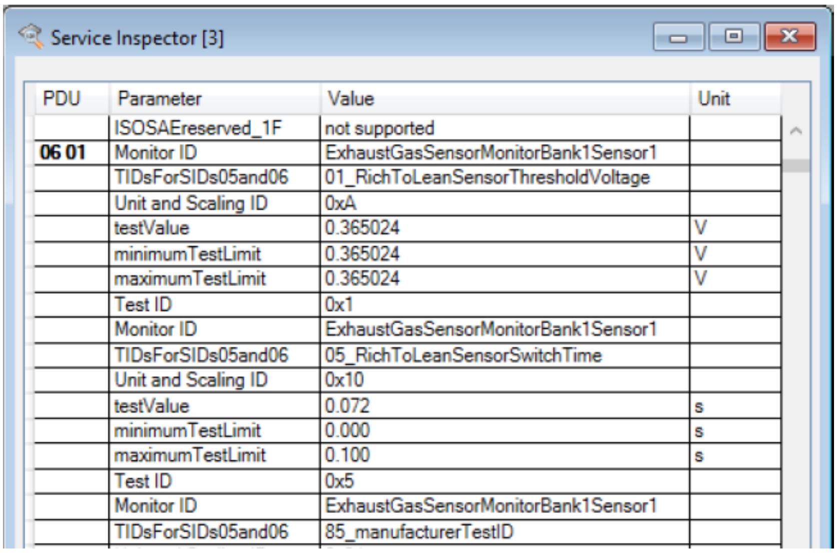
1. Select **ODX** > **User views** > **Service Inspector**.

   The "Service Inspector" user view opens.
2. Click **Configure**.

   The configuration window opens.

3. Select "General" and activate the option "Create Measurement Signals from Response Parameters".
4. Select "Request Service".
5. Activate the service "[0x6] .... 1MID".
6. Click **OK**.
7. Click **Read**.

   All OBD Mode 6 MIDs are queried and the results shown.

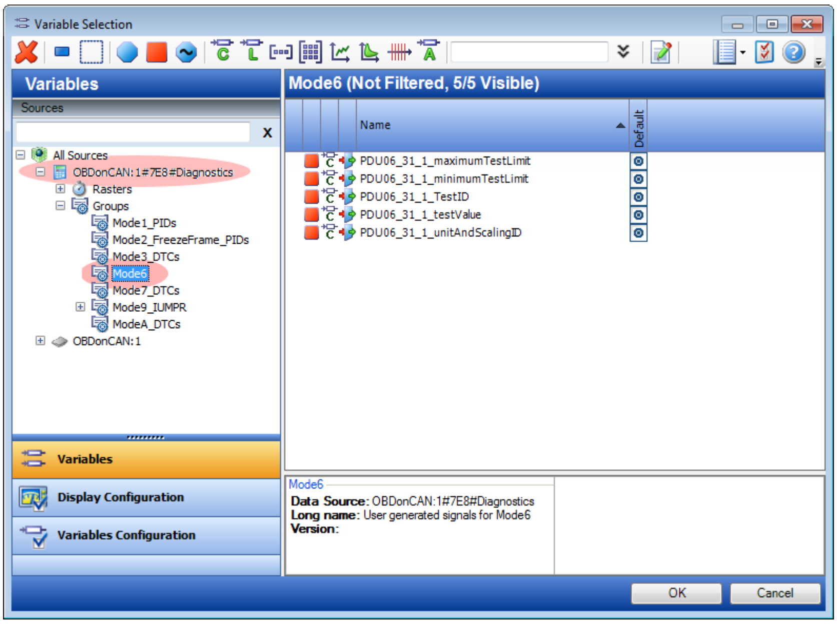
   

PDU	Parameter	Value	Unit
	ISOSAEreserved_1F	not supported	
06 01	Monitor ID	ExhaustGasSensorMonitorBank1Sensor1	
	TIDsForSIDs05and06	01_RichToLeanSensorThresholdVoltage	
	Unit and Scaling ID	0xA	
	testValue	0.365024	V
	minimumTestLimit	0.365024	V
	maximumTestLimit	0.365024	V
	Test ID	0x1	
	Monitor ID	ExhaustGasSensorMonitorBank1Sensor1	
	TIDsForSIDs05and06	05_RichToLeanSensorSwitchTime	
	Unit and Scaling ID	0x10	
	testValue	0.072	s
	minimumTestLimit	0.000	s
	maximumTestLimit	0.100	s
	Test ID	0x5	
	Monitor ID	ExhaustGasSensorMonitorBank1Sensor1	
	TIDsForSIDs05and06	85_manufacturerTestID	

   In the background, a diagnostic signal is created for each of the parameters displayed.
8. Close the experiment.

   A note is displayed explaining that the ODX project has changed as new signals have been generated with a query about whether you want to save the changes.
9. Confirm with **OK** or **Yes**.
10. Open the experiment again.
11. Open the Variable Selection dialog box via **Variables** > **Variable Selection**.
12. From the "Variables" field, select "ODX test device: 1#7E8#Diagnostics".

    The new signals are in the "Mode6" group and can be selected for measurement there.



Mode6 (Not Filtered, 5/5 Visible)

Name	Default
PDU06_31_1_maximumTestLimit	⊙
PDU06_31_1_minimumTestLimit	⊙
PDU06_31_1_TestID	⊙
PDU06_31_1_testValue	⊙
PDU06_31_1_unitAndScalingID	⊙

Mode6

Data Source: OBDonCAN:1#7E8#Diagnostics
Long name: User generated signals for Mode6
Version:

6 ODX-LINK Troubleshooting

6.1 Errors When Adding an ODX Project to the Database

Error Message	Error Description	Remedy
Unable to create ODX project. The MVCI Server is in use.	The applications "Tool X" and ODX-LINK, and the adding of an ODX project to the database cannot take place simultaneously.	End "Tool X" or ODX-LINK respectively.
ODX file import failed due to one of the following reasons: some ODX files or files referenced by the selected ODX files (e.g. java code) are missing.	An error was reported by the MVCI Server during the conversion or verification of the ODX project.	Check the reference files of the project to be imported.
ODX file import failed due to one of the following reasons: the ODX files are not ODX V2.0.1 or V2.2 compliant.		Create the ODX project and the relevant ODX files in accordance with the ODX V2.0.1 or V2.2 specification.
ODX file import failed due to one of the following reasons: the ODX files are inconsistent and violate ASAM ODX checker rules.		Check whether the ODX files conform to the ASAM-ODX rules.
Reinstall ODX Add-on Installation.	Necessary directories or files of the ODX Add-on installation are missing.	Install ODX Add-on again.

Error Message	Error Description	Remedy
Can't save ODX project to file '*'.	There is too little storage space in the `ETAS temp` folder.	Delete the contents of `ETAS temp` and reboot INCA.
	You do not have the necessary user rights.	Contact your system administrator.
The ODX project is inconsistent. It contains no project file (*.prj)	There is too little storage space in the `ETAS temp` folder.	Delete the contents of `ETAS temp` and reboot INCA.
	You do not have the necessary user rights.	Contact your system administrator.
	The database object for the ODX project is not valid.	Generate a new database object using the ODX files.
Decompressing failed	There is too little storage space in the `ETAS temp` folder.	Delete the contents of `ETAS temp` and reboot INCA.
	You do not have the necessary user rights.	Contact your system administrator.
Can't create temporary directory '*'	There is too little storage space in the `ETAS temp` folder.	Delete the contents of `ETAS temp` and reboot INCA.
	You do not have the necessary user rights.	Contact your system administrator.
Can't remove temporary directory	INCA cannot remove a file from the `ETAS temp` temporary directory.	
	INCA is still accessing this file.	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA.
	Another process is still accessing this file.	End all other applications which could be accessing the `ETAS temp` directory. It may be necessary to reboot your PC.
	You do not have the necessary user rights.	Contact your system administrator.

Error Message	Error Description	Remedy
Can't copy file '*' to '*'	There is too little storage space in the `ETAS temp` folder.	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA. It may be necessary to reboot your PC.
	You do not have the necessary user rights.	Contact your system administrator.
Close ODX-LINK	The applications "Tool X" and ODX-LINK, and the adding of an ODX project to the database cannot take place simultaneously.	End "Tool X" or ODX-LINK respectively.
Error accessing MVCI Server	An error occurred when trying to access the MVCI Server.	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA.
		End all other applications which could be accessing the `ETAS temp` directory. It may be necessary to reboot your PC.
Can't remove the file '*'	INCA cannot remove a file from the `ETAS temp` temporary directory.	
	INCA is still accessing this file.	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA.
	Another process is still accessing this file.	End all other applications which could be accessing the `ETAS temp` directory. It may be necessary to reboot your PC.
	You do not have the necessary user rights.	Contact your system administrator.

Error Message	Error Description	Remedy
Can't load ODX project '*'	An error occurred when adding an ODX project to the INCA database	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA. It may be necessary to reboot your PC.
The file * is missing.	A necessary file of the ODX Add-on installation is missing.	Install ODX Add-on again.
The ODX project is inconsistent. It contains no file '*'	There is too little storage space in the `ETAS temp` folder.	Delete the contents of `ETAS temp` and reboot INCA.
	You do not have the necessary user rights.	Contact your system administrator.
	The database object for the ODX project is not valid.	Generate a new database object using the ODX files.
The directory * is missing..	A necessary directory of the ODX Add-on installation is missing.	Install ODX Add-on again.
The ODX project is inconsistent. File '*' is not readable	There is too little storage space in the `ETAS temp` folder.	Delete the contents of `ETAS temp` and reboot INCA.
	You do not have the necessary user rights.	Contact your system administrator.
	The database object for the ODX project is not valid.	Generate a new database object using the ODX files.

Error Message	Error Description	Remedy
Can't delete temporary ODX project in file '*'	INCA cannot remove a file from the `ETAS temp` temporary directory.	
	INCA is still accessing this file.	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA. It may be necessary to reboot your PC.
	Another process is still accessing this file.	End all other applications which could be accessing the `ETAS temp` directory. It may be necessary to reboot your PC.
	You do not have the necessary user rights.	Contact your system administrator.
Can't delete temporary file '*'	INCA cannot remove a file from the `ETAS temp` temporary directory	
	INCA is still accessing this file.	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA. It may be necessary to reboot your PC.
	Another process is still accessing this file.	End all other applications which could be accessing the `ETAS temp` directory. It may be necessary to reboot your PC.
	You do not have the necessary user rights.	Contact your system administrator.

Error Message	Error Description	Remedy
Can't delete temporary file '*'	INCA cannot remove a file from the `ETAS temp` temporary directory	
	INCA is still accessing this file.	Shut down INCA, delete the contents of `ETAS temp` and reboot INCA. It may be necessary to reboot your PC.
	Another process is still accessing this file.	End all other applications which could be accessing the `ETAS temp` directory. It may be necessary to reboot your PC.
	You do not have the necessary user rights.	Contact your system administrator.
Error in TP_BLOP generation: Sending a Tester Present Message is required for *	The ODX-COM PARAMs define a "tester present" message without parameters. KWPOnCAN does not support this.	Set the value of the "CP_TesterPresentHandling" parameter to 1 or set the byte size of the "CP_TesterPresentMessage" parameter to a value > 0.

6.2 Errors When Opening the ODX Project

Error Message	Error Description	Remedy
Diagnostic signal description file error: {Error, Line, Position}	A syntax error is contained in the description file (DSL file).	Exchange the DSL file or have the programmer correct the DSL file.
Failed to open DiagSignalListSchema.xsd.	The file DiagSignalListSchema.xsd was not found.	Check whether the file lies in the correct folder *(ETAS\INCA 7.5\ODX)* and is properly named. If the file was not revised, instal ODX anew.
Failed to open VirtualDeviceTemplate.a2l.	The file VirtualDeviceTemplate.a2l was not found.	Check whether the file lies in the correct folder *(ETAS\INCA 7.5\ODX)* and is properly named. If the file was not revised, instal ODX anew.
Logical Link '{logical link}' not found inside the diagnostic signal description.	The logical link which you have assigned to a hardware is not described in the DSL file. This means that no diagnostic signal is available to you with this logical link.	Assign the hardware another logical link, if necessary.
No valid request found in DSL file for signal: '{signal name}'	An appropriate request (inquiry) is missing or the referenced request is faulty for a diagnostic signal in the DSL file. A faulty Request is localized by one of the following error messages (see below).	Exchange the DSL file or have the programmer correct the DSL file.

Error Message	Error Description	Remedy
Request: '{request name}' doesn't contain a parameter with name: '{parameter name}'	The Request contains no parameters with the appropriate name in the DSL file.	Exchange the DSL file or have the programmer correct the DSL file.
Request: '{request name}' incorrect: value missing	In the DSL file, the applicable value is missing by a parameter with this Request.	Exchange the DSL file or have the programmer correct the DSL file.
Request: '{request name}' incorrect: short name missing.	In the DSL file, the appropriate short name is missing by a parameter with this Request. .	Exchange the DSL file or have the programmer correct the DSL file.
Request: '{request name}' incorrect: type not 'const'.	In the DSL file, the Request does not have the correct type ('const') for the above signal.	Exchange the DSL file or have the programmer correct the DSL file.
Request: '{request name}' incorrect: type for parameter '{parameter name}' isn't 'variable' or 'field'.	In the DSL file, the Request does not have the correct type ('variable'or 'field') for the above signal .	Exchange the DSL file or have the programmer correct the DSL file.
Request: '{request name}' doesn't contain a valid PDU: '{incorrect PDU}'	The data input in the DSL file for the PDU is missing or is faulty.	Exchange the DSL file or have the programmer correct the DSL file.

6.3 Errors When Starting a Measurement

Error Message	Error Description	Remedy
Failed to create diagnostic service for signal: '{signal}'	In the DSL file, a diagnostic service is performed which does not exist in the ODX Project.	Correct the ODX Project. Exchange the DSL file or have the programmer correct the DSL file.
{signal name} getting TrgtSvr-Sink: '{sink name}'	The signal does not receive a Target Server Object.	Start the measurement anew. Load the experiment anew.
Unavailable parameter '{parameter name}' for Request with ID '{request ID}'. Failed to create diagnostic service for signal: '{signal name}'	In the DSL file, a request is performed whose parameter does not exist in the ODX Project.	Correct the ODX Project. Exchange the DSL file or have the programmer correct the DSL file.

6.4 Errors During the Measurement

Error Message	Error Description	Remedy
Service for the following diagnostic signals reported execution state: '{list of signals}' Execution state: '{text}' Description of the error: ' {text}' Vendor description of the error: '{text}'	The response of a diagnostic service has not been answered "positively". All signals which could not be measured are listed in the error message. If no signals could be measured, the link to the hardware may not work correctly.	Remove the signals which are not supported by the control device. Check the connection with the hardware and initialize the hardware anew.
No response received for signal: '{signal name}'	In the response of the diagnostic service, no reply is present for the signal. Possibly, the control device does not support all diagnostic signals, which are included in the DSL file.	Remove the signals which are not supported by the control device. Check the connection with the hardware and initialize the hardware anew.
Parameter for the following diagnostic signal has an error: '{signal}' Description of the error: ' {text}' Vendor description of the error: '{text}'	In the response, a parameter for the signal is faulty and could not be evaluated. This means that no physical value could be calculated. This could have been caused by a faulty ODX data input or a faulty response from the control device.	Correct the ODX Project or exchange it.

Error Message	Error Description	Remedy
Service for the following diagnostic signals has an error: '{list of signals}' Description of the error: '{text}' Vendor description of the error: '{text}'	In the answer, either a response or a parameter of the response is faulty. This could have been caused by a faulty ODX data input or a faulty response from the control equipment.	Correct the ODX Project or exchange it.
Exception by updating value for signal '{signal name}': '{exception}'	The conversion of a data type of the D-server in a data type, which INCA understands, has failed. This could have been caused by a faulty ODX data input or a faulty answer from the control device.	Correct the ODX Project or exchange it.
{signal name}writing value:'{exception}'	A value of the signal could not be written into the target server.	Start the measurement anew.

7 ODX Communication Parameters

The communication parameters are used to initialize the devices used in diagnosis ia ODX and are stored in the `ComparamSpec` section of the ODX project (odx-c).

Based on the existing ODX parameters, an ASAP2 TP_BLOP is generated which is required to generate the ASAP1b driver.

The following sections explain the ODX parameters supported.

Note

ODX-LINK can work with the parameters from the ASAP2 file.
The following rule applies: If an ASAP2 project was assigned to a diagnostic device during hardware configuration, initialization takes place via the parameters from ASAP2 (as was the case so far) – if, however, no ASAP2 file was assigned, initialization takes place via the parameters of the ODX project.

7.1 A2L Structure: TP_BLOP

This structure is required for the following protocols:

- KWPOnCAN
- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) → A2L(y))	Default Value	Comment
blob version	-	-	-	KWPOnCAN: 0x201; UDSonCAN: 0x301	The default value is always used.
Protocol version	-	-	-	KWPOnCAN: VDA_1996; UDSonCAN: ISO14229_1_2003	The default value is always used.
byte order	-	-	-	KWPOnCAN: MSB_FIRST; UDSOnCAN: BYTEORDER_MSB_FIRST	This parameter is only required for measuring/calibration jobs. The default value is used for diagnostic jobs.

7.2 A2L Structure: CAN

This structure is required for the following protocols:

- KWPOnCAN
- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range/Unit	Conversion (ODX(x) >→ A2L(y))	Default Value	Comment
Baudrate	CP_Baudrate	[0x0; 0xFFFFFFFF] Unit: baud		-	
sample point	CP_BitSamplePoint	[0; 100]	y = x	-	
samplecount per bit	CP_SamplesPerBit	[0; 1]	0=>1;1=>3	-	

A2L Parameter	ODX Parameter	Value Range/Unit	Conversion (ODX(x) >→ A2L(y))	Default Value	Comment
BTL_CYCLES	-	-	Is generated using a "Best Match" algorithm from "CP_BitSamplePoint" and "CP_SyncJumpWidth".	-	
SJW length	CP_SyncJumpWidth	[0; 100]	y = round(CP_SyncJumpWidth * BTL_CYCLES) / 100)	-	
SYNC_EDGE	CP_INCA_SYNC_EDGE	[0; 2]	y = x1)	0	The default value is always used.
1) This is an optional INCA-specific parameter. If it is not defined, the value defined in the "Default Value" column is used.					

7.3 A2L Structure: CAN Address

This structure is required for the following protocols:

- KWPOnCAN
- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) → A2L(y))	Default Value	Comment
CAN_ID ECU	CP_CanRespUSDTId CP_CanRespUSDTFormat	[0x0; 0x1FFFFFFF] [0x4; 0x0F]	4)	-	
CAN_ID tester	CP_CanPhysReqId CP_CanPhysReqFormat	[0x0; 0x1FFFFFFF] [0x0; 0x3F]	5)	-	
TGT_ECU	CP_CanPhysReqExtAddr CP_CanPhysReqFormat	0x0; 0xFF] [0x0; 0x3F]	6)	-	
TGT_tester	CP_CanRespUSDTExtAddr CP_CanRespUSDTFormat	[0x0; 0xFF] [0x4; 0x0F][	7)	-	

4) y = CP_CanRespUSDTId | ((CP_CanRespUSDTFormat & 0x02) << 30)

5) y = CP_CanPhysReqId | ((CP_CanPhysReqFormat & 0x02) << 30)

6) if ((CP_CanPhysReqFormat & 0x08) == 0x08): y = CP_CanPhysReqExtAddr otherwise no value is written

7) if ((CP_CanRespUSDTFormat & 0x08) == 0x08): y = CP_CanPhysReqExtAddr otherwise no value is written

7.4 A2L Structure: CAN TesterPresentOptions

This structure is required for the following protocols:

- KWPOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) => A2L(y))	Default Value	Comment
tester present	CP_TesterPresentHandling, CP_TesterPresentReqRsp, CP_TesterPresentMessage	[0; 1] [0; 1] Each byte: [0x0; 0xFF], Byte length: 1 .. 12	See conversion parameters		

Conversion parameters:

if (CP_TesterPresentHandling = 0) or (bytesize of CP_TesterPresentMessage = 0): 3 [1]

if (bytesize of CP_TesterPresentMessage = 1): TesterPresent_WithoutParameter

if (bytesize of CP_TesterPresentMessage > 1 and CP_TesterPresentReqResp = 0):
TesterPresent_WithParameter_NoResponseRequired

if (bytesize of CP_TesterPresentMessage > 1 and CP_TesterPresentReqResp = 1):
TesterPresent_WithParameter_ResponseRequired

[1] This does not conform to KWP2000 AML but is supported by the ETAS ASAP1b-driver

7.5 A2L Structure: SESSION TesterPresentOptions

This structure is required for the following protocols:

- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) => A2L (y))	Default Value	Comment
tester present	CP_TesterPresentHandling, CP_TesterPresentReqRsp, CP_TesterPresentMessage	[0; 1] [0; 1] Each byte: [0x0;0xFF], Byte length: 1 .. 12	See conversion parameters		

Conversion parameters:

if (bytesize of CP_TesterPresentMessage = 1): Error: "Sending a Tester Present Message without parameters is not allowed"

if (CP_TesterPresentHandling = 0 or bytesize of CP_TesterPresentMessage = 0): NoTesterPresent

if (CP_TesterPresentHandling != 0 and bytesize of CP_TesterPresentMessage != 0 and CP_TesterPresentReqResp = 0): TesterPresent_WithParameter_NoResponseRequired

if (CP_TesterPresentHandling != 0 and bytesize of CP_TesterPresentMessage != 0 and CP_TesterPresentReqResp = 1): TesterPresent_WithParameter_ResponseRequired

7.6 A2L Structure: CAN NETWORK_LIMITS

This structure is required for the following protocols:

- KWPOnCAN
- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) => A2L (y))	Default Value	Comment
WFT_MAX	CP_CanMaxNumWaitFrames	[0; 1027]	y = x	-	
XDL_MAX	CP_INCA_XDL_MAX	[0x0; 0xFFFFFFFF]	y = x[1)]	500	The default value is always used.
1) This is an optional, INCA-specific parameter. If it is not defined, the value defined in the "Default Value" column is used.					

7.7 A2L Structure: DIAG_BAUD

This structure is required for the following protocol:

- KWPOnCAN

A2L Parameter	ODX Parameter	Value Range/Unit	Conversion (ODX(x) => A2L(y))	Default Value	Comment
Baudrate	CP_INCA_DIAG_BAUD_Baudrate	[0x0; 0xFFFFFFFF] Unit: baud	y=x [1], [2]	10400 (default baud rate)	The default value is always used.
diagnostic mode	CP_INCA_DIAG_BAUD_DiagnocsticMode	[0;0xFF]	y=x[1]	0x86 (= default session)	The default value is always used.
BD_PARA	CP_INCA_DIAG_BAUD_BD_PARA	Each byte: [0x0;0xFF], Byte length:1 .. 12	y=x [1]	0x06 0x00 0x28 0xA0	The default value is always used.
1) This is an optional, INCA-specific parameter. If it is not defined, the value defined in the "Default Value" column is used. 2) With this parameter, it is assumed that "baud" is the unit used. No deviating unit can be defined in ODX.					

7.8 A2L Structure: TIME_DEF USDTP_TIMING

This structure is required for the following protocols:

- KWPOnCAN

A2L Parameter	ODX Parameter	Value Range/Unit	Conversion (ODX(x) => A2L(y))	Default Value	Comment
As	CP_As	[0; 20000000] Unit: µs	y = round (x/1000) 3)	-	
Bs	CP_Bs	[0; 20000000] Unit: µs		-	
Cr	CP_Cr	[0; 20000000] Unit: µs		-	
p2Min	CP_P2Min	[0; 250000] Unit: µs			
p2Max	CP_P2Max	[0; 125000000] Unit: µs			
p3Min	CP_P3Min	[0; 250000] Unit: µs			
p3Max	CP_P3Max_Ecu	[0; 100000000] Unit: µs			

3) With this parameter, it is assumed that "µs" is the unit used. No deviating unit can be defined in ODX.

7.9 A2L Structure: USDTP_TIMING_DEFAULTS

This structure is required for the following protocols:

- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range/Unit	Conversion (ODX(x) => A2L(y))	Default Value	Comment
As	CP_As	[0; 20000000] Unit: µs	y = round (x/1000)$^{3)}$	-	
Bs	CP_Bs	[0; 20000000] Unit: µs		-	
Cr	CP_Cr	[0; 20000000] Unit: µs		-	
p2Min	CP_P2Min	[0; 250000] Unit: µs		-	
p2Max	CP_P2Max	[0; 125000000] Unit: µs		-	
p3Min	CP_P3Phys	[0; 125000000] Unit: µs		-	
p3Max	CP_TesterPresentTime	[0; 30000000] Unit: µs		-	

3) With this parameter, it is assumed that "µs" is the unit used. No deviating unit can be defined in ODX.

7.10 A2L Structure: SESSION

This structure is required for the following protocols:

- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) => A2L (y))	Default Value	Comment
session identifier	CP_INCA_ UDSSessionIdentifier	char length: 1 .. 99	1)	"UDSSession"	The default value is always used.
diagnostic mode	CP_INCA_ USDSDiagnosticMode	[0;0xFF]	y=x 1)	0x01 (= default session)	The default value is always used.
1) This is an optional, INCA-specific parameter. If it is not defined, the value defined in the "Default Value" column is used.					

7.11 A2L Structure: ADDRESS_AND_LENGTH_FORMAT_IDENTIFIER

This structure is required for the following protocols:

- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) => A2L(y))	Default Value	Comment
AALFI general setting	-	-	-	0x13	These parameters exist for reasons of compatibility to TP_BLOP and should not be used by the ASAP1b driver for diagnostic jobs.
AALFI_FOR_CHECKSUM_CALCULATION	-	-	-	0x13	
AALFI_FOR_ERASE_MEMORY	-	-	-	0x13	
AALFI_FOR_WRITE_MEMORY_BY_ADDR	-	-	-	0x13	
AALFI_FOR_READ_MEMORY_BY_ADDR	-	-	-	0x13	
AALFI_FOR_DYNAMICALLY_DEFINE_DATA_ID	-	-	-	0x13	
AALFI_FOR_REQUEST_DOWNLOAD	-	-	-	0x13	

7.12 A2L Structure: SESSION SessionOpeningOrder

This structure is required for the following protocols:

- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) => A2L(y))	Default Value	Comment
session opening order	CP_INCA_ UDSSessionOpeningOrder	[0; 2]	1)	2	The default value is always used.
1) This is an optional, INCA-specific parameter. If it is not defined, the value defined in the "Default Value" column is used.					

7.13 A2L Structure: CAN Transport Protocol Version

This structure is required for the following protocols:

- UDSOnCAN

A2L Parameter	ODX Parameter	Value Range	Conversion (ODX(x) => A2L(y))	Default Value	Comment
transport protocol version	-	-	-	ISO15765_ 2_2003	The default value is always used.

8 Contact Information

Technical Support

For details of your local sales office as well as your local technical support team and product hotlines, take a look at the ETAS website:

www.etas.com/hotlines

ETAS offers trainings for its products:

www.etas.com/academy

ETAS Headquarters

ETAS GmbH

Borsigstraße 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	Internet:	www.etas.com

Glossary

D

Diagnostic database

The diagnostic database contains all services, their parameters and the possible ECU responses in hexadecimal and plain text notation. The diagnostic database is stored in a file with the ".pdx" or ".odx" extension.

E

ETAS temp folder

The ETAS temp folder contains temporary ETAS files. During the initial installation of an ETAS software product, you have defined the location where the ETAS temp folder is stored.

O

ODX project

An ODX project combines all components required for communicating and interpreting data. This includes, for example, the diagnostic database and interface definition.

ODX project file

The ODX project file contains the following information: - A pointer to the database with the services that have been defined for your ECU. This database contains all the services, their parameters and the possible ECU responses in hexadecimal and plain text notation. - The hardware interface configuration - The definitions for the links between logical and physical interfaces The ODX project file has the ".prj" file extension.

P

PDU

Protocol Data Unit. The data transmitted to the ECU after the service ID. The PDUs specify the parameters and the associated values for each service.

Figures

Index