Content

1 Features

1.1 Supported standards

DiagRA® S supports the following standards:

- ISO 9141
- ISO 11898
- ISO 13400
- ISO 14229
- ISO 14230
- ISO 15765s
- SAE J1939
- SAE J1850 PWM
- SAE J1850 VPW

The following table shows all virtual diagnostic interfaces and the supported protocols.

Standard	PassThru	RP1210 B	DoIP
ISO 9141	✓	✗	✗
ISO 11898	✓	✓	✗
ISO 14229	✓	✗	✓
ISO 14230	✓	✗	✓
ISO 15765	✓	✗	✓
SAE J1939	✗	✓	✗
SAE J1979	✓	✗	✓
SAE J1850 PWM	✓	✗	✗
SAE J1850 VPW	✓	✗	✗

1.2 Supported devices

Note: All timestamps, which are returned by the interfaces, are in milliseconds.

a. PassThru device

a.1 Supported standards

- ISO 9141
- ISO 11898
- ISO 14229
- ISO 14230
- ISO 15765
- SAE J1850 PWM
- SAE J1850 VPW

b. TMC RP1210 B device

b.1 Supported standards

- ISO 11898
- SAE J1939

c. DoIP (ISO 13400 over Ethernet)

DiagRA® S can act as a DoIP ECU / DoIP Gateway. UDP and TCP Sockets according to ISO 13400 will run locally and the diagnostic requests (ISO 142229, ISO 15765, …) will be processed by DiagRA S Core.
DoIP Simulation can be started without UI using DoIPServer.exe –dsproj "<path-to-dsproj-file>" call.

d. RA D-PDU-API (D-PDU Bridge)

DiagRA® S installs and configures a D-PDU Bridge on your computer during the installation process. The D-PDU Bridge offers a D-PDU API as specified in ISO 22900-2 to the user but uses another device standard internally. You can use any of DiagRA® S' devices as D-PDU device by using the D-PDU Bridge.

1.3 Supported log files

a. DiagRA® D log files

- ISO 9141
- ISO 14229
- ISO 14230
- ISO 15765
- SAE J1850 PWM
- SAE J1850 VPW

b. SAE J1699-3 log files

- ISO 15765

c. SAE J1939-84 log files

- SAE J1939

d. IXXAT CAN Traces

- ISO 11898 (CAN Monitoring)
- ISO 14229

e. Vector CAN Traces

- ISO 11898 (CAN Monitoring)
- ISO 14229
- SAE J1939

f. SR Files

- ISO 14229
- ISO 14230
- ISO 1576

g. DiagRA® F log files

- ISO 14229

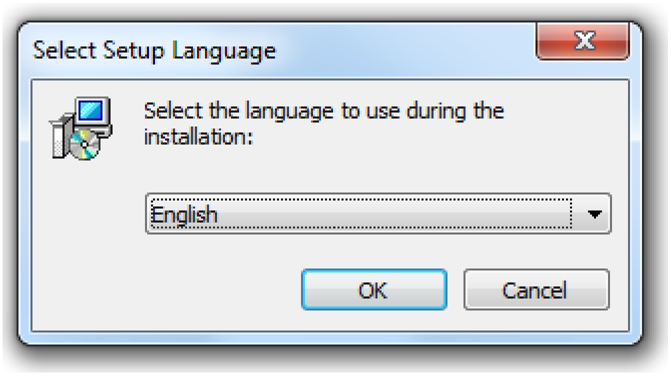
1.4 Hardware Simulation

You can use DiagRA® S for simulating a vehicle on the physical CAN bus. Learn more about it in chapter Hardware Simulation.

2 Setup & Basics

2.1 Setup

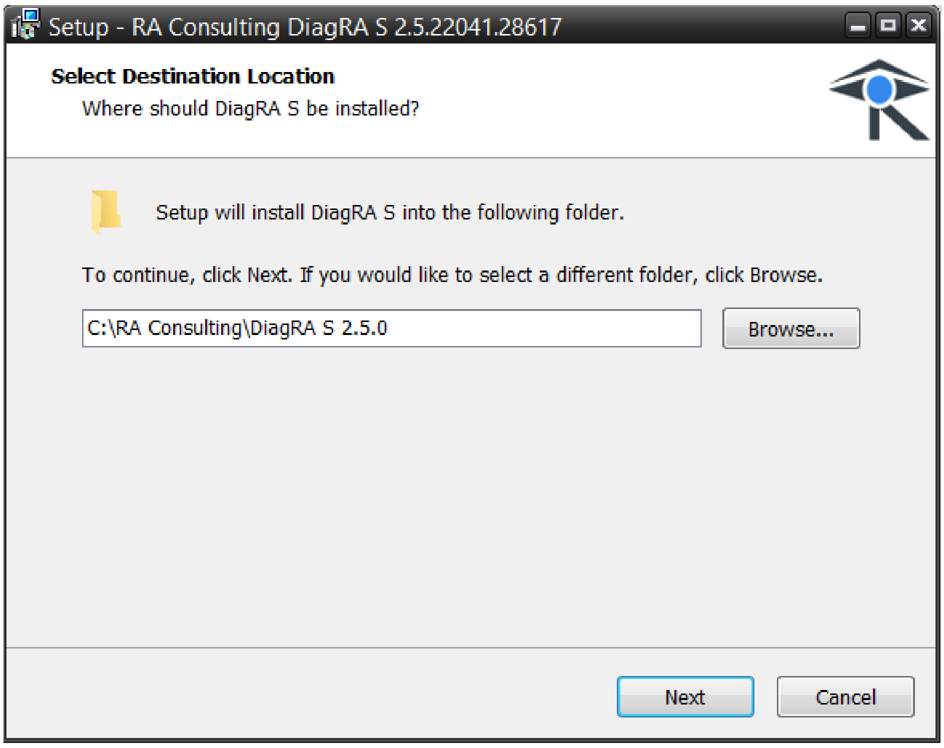
Run the delivered application "DiagRA_S-XXXX-setup.exe" („XXXX" is the version number of your setup, e.g. „1.5.0.12") to install DiagRA® S.


Select Setup Language

Select the language to use during the installation:

English

Select your language in the dialog window and commit your selection by pressing "OK".


Setup - RA Consulting DiagRA S 2.5.22041.28617

Select Destination Location

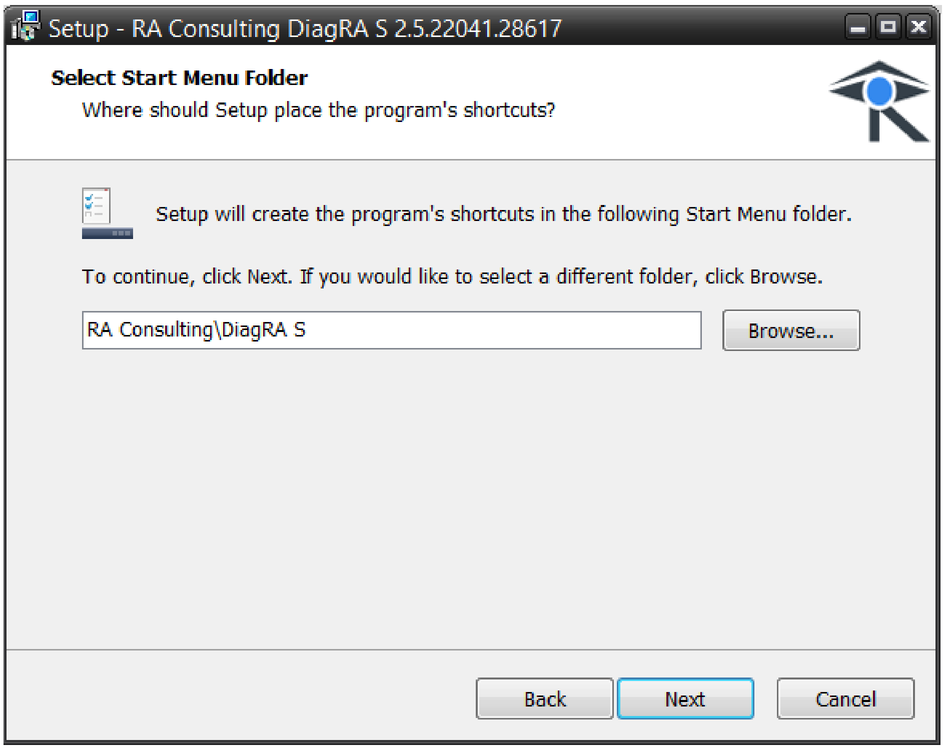
Where should DiagRA S be installed?

Setup will install DiagRA S into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\RA Consulting\DiagRA S 2.5.0

Choose the directory, in which you want to install DiagRA® S. It is highly recommended not to install in the Program Files directory, if you are using Windows Vista or Windows 7. Commit the installation path by clicking on the button “Next >“. Commit the following dialog window by clicking on “Next >“, too.



Setup - RA Consulting DiagRA S 2.5.22041.28617

Select Start Menu Folder

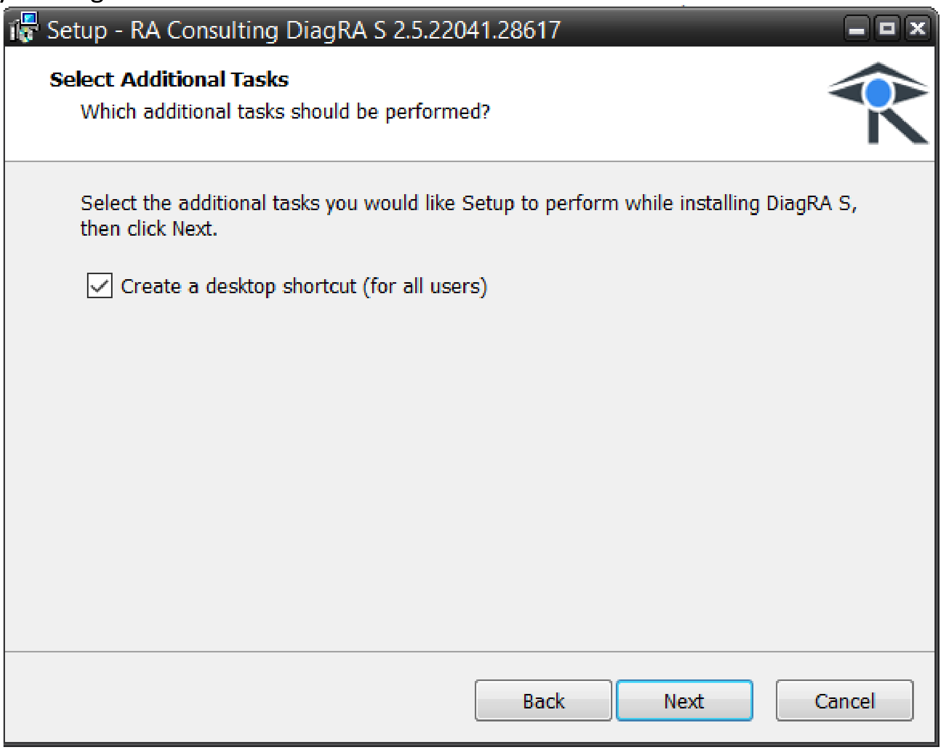
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

RA Consulting\DiagRA S

Select the folder of your start menu in which you want to put the link to DiagRA® S. Commit your selection by clicking on “Next >“.



Setup - RA Consulting DiagRA S 2.5.22041.28617

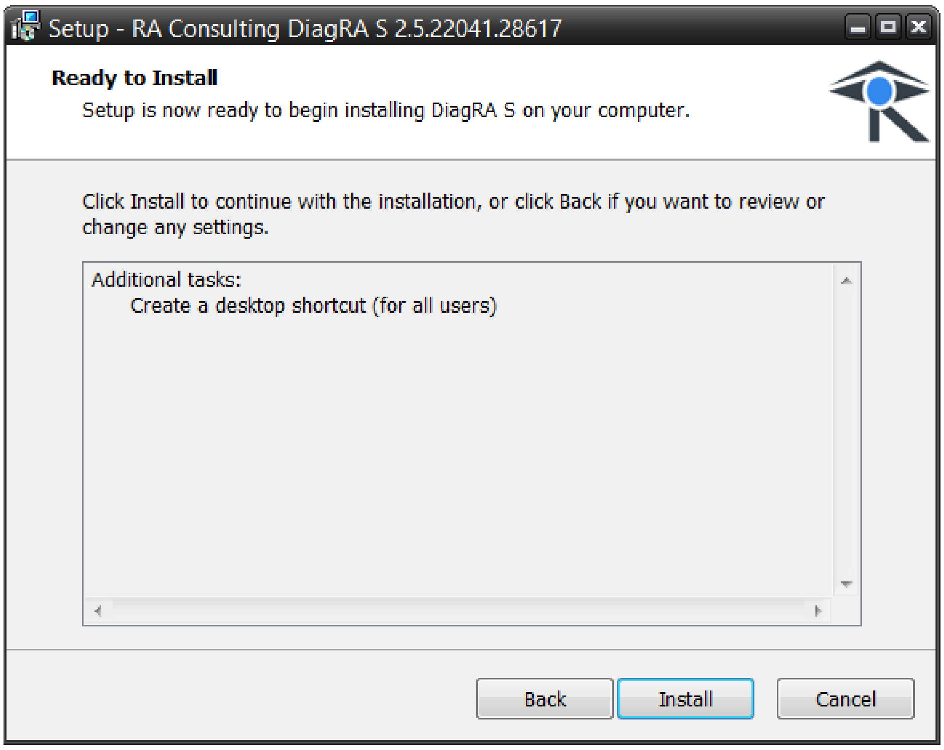
Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing DiagRA S, then click Next.

☑ Create a desktop shortcut (for all users)

Now, you have the possibility to add a link to DiagRA® S to your desktop and to your task bar. After you made your selection, please click on “Next >“ to continue with the installation.


Setup - RA Consulting DiagRA S 2.5.22041.28617

Ready to Install

Setup is now ready to begin installing DiagRA S on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Additional tasks:
Create a desktop shortcut (for all users)

Finally, you will get a summary of your installation. If one or more options are wrong, feel free to use the button "< Back" to go back to one of the selections to edit your settings. Commit your final settings by clicking on "Install". You need to input your delivery number and after that, you will be able to install and use DiagRA® S.

2.2 Prerequisites to run DiagRA S

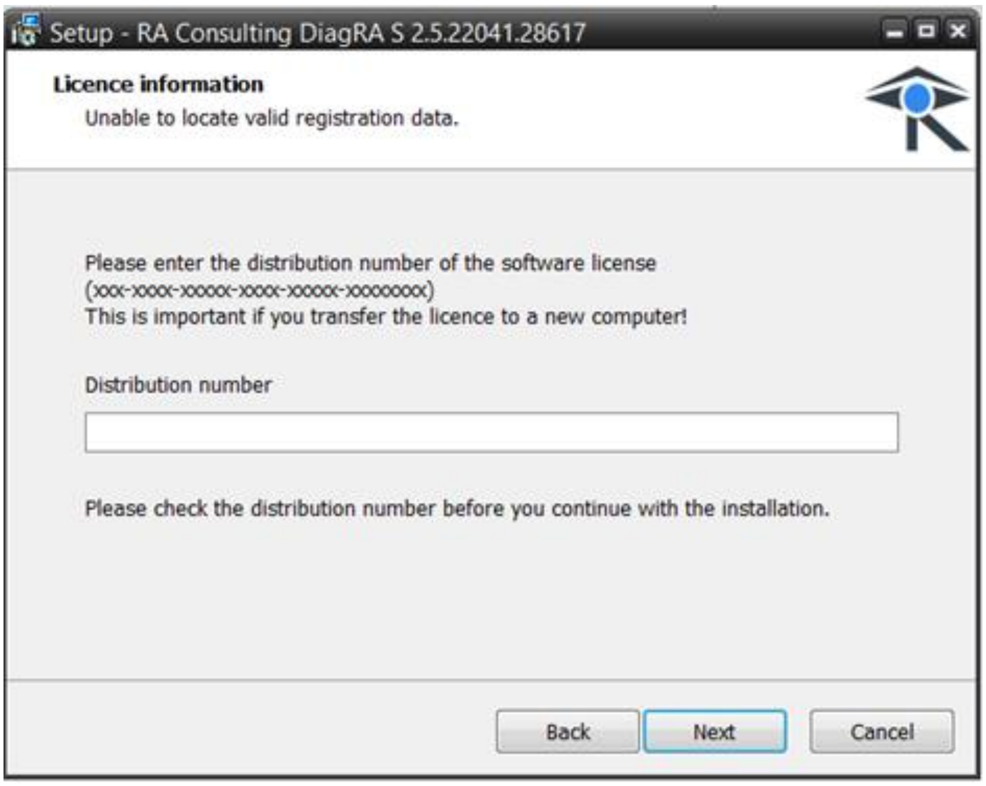
In order to be able to use DiagRA® S, you need rights to read and write to the Registry, the folder in which you have installed the application and to the Windows folder.

Additionally, you must have installed Microsoft's .NET Framework 4.6.1 and Microsoft's Visual Studio 2017 (C++) Runtime. The packages will be installed during the DiagRA® S installation, if they are missing on your computer. An active internet connection is required to install them.

If you don't have rights which are required to install or run DiagRA® S, please contact your local system administrator.

2.3 Registration

When you start DiagRA® S for the first time the following registration dialog will open:



Setup - RA Consulting DiagRA S 2.5.22041.28617

Licence information

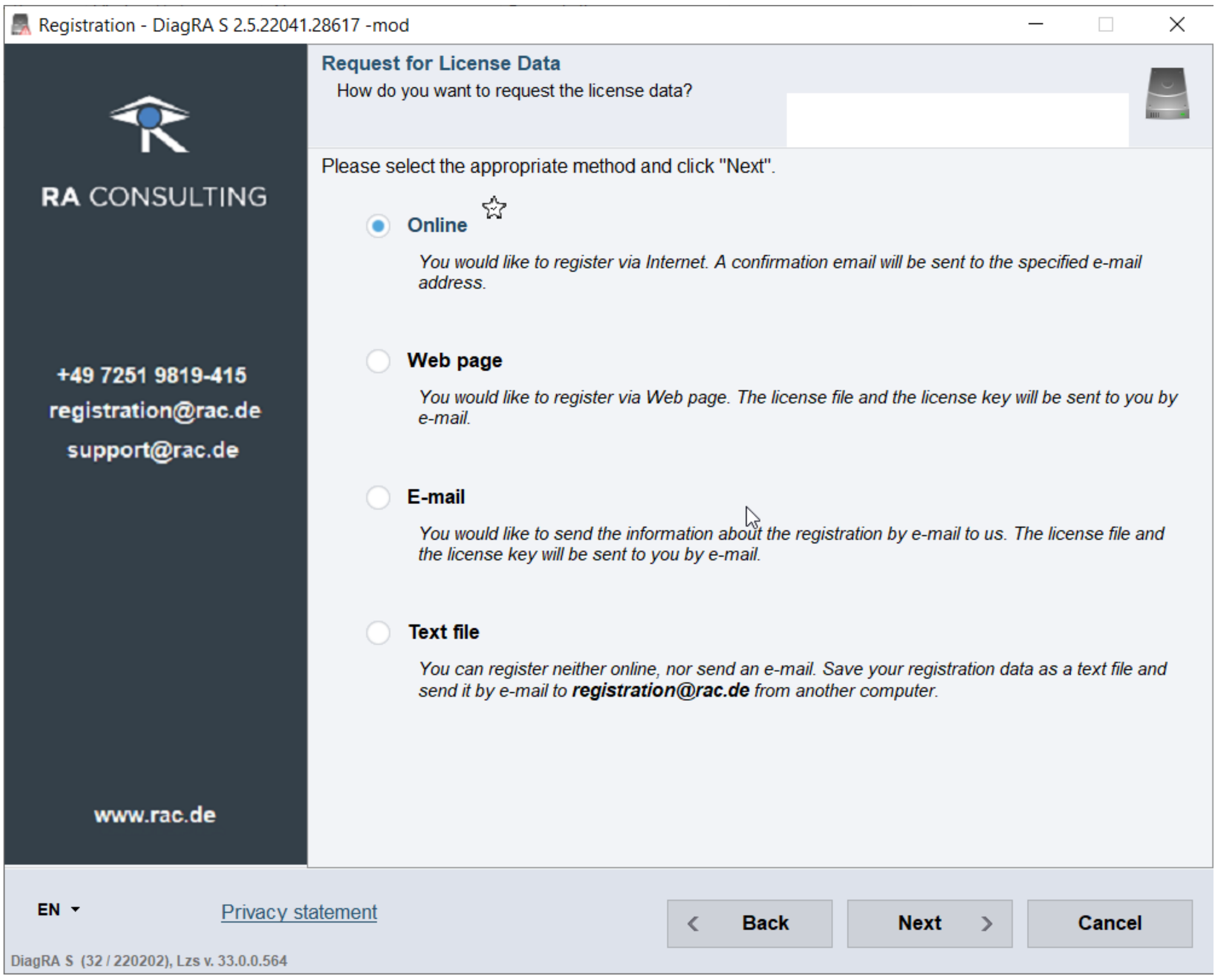
Unable to locate valid registration data.

Please enter the distribution number of the software license
(xxx-xxxx-xxxxx-xxxx-xxxxx-xxxxxxxx)
This is important if you transfer the licence to a new computer!

Distribution number

Please check the distribution number before you continue with the installation.

You have to enter the distribution number you've received from RA Consulting.



Registration - DiagRA S 2.5.22041.28617 -mod

Request for License Data

How do you want to request the license data?

Please select the appropriate method and click "Next".

[✓ **Online**]

You would like to register via Internet. A confirmation email will be sent to the specified e-mail address.

Web page

You would like to register via Web page. The license file and the license key will be sent to you by e-mail.

E-mail

You would like to send the information about the registration by e-mail to us. The license file and the license key will be sent to you by e-mail.

Text file

You can register neither online, nor send an e-mail. Save your registration data as a text file and send it by e-mail to ***registration@rac.de*** *from another computer.*

RA CONSULTING

+49 7251 9819-415
registration@rac.de
support@rac.de

www.rac.de

DiagRA S (32 / 220202), Lzs v. 33.0.0.564

We recommend using the Online Registration, because it's the simplest and fastest method to register your DiagRA® S. After selecting the proper registration method follow the instructions step-by-step to register the application. After the registration has been completed you can communicate with DiagRA® S using its simulated interfaces.

You need to register DiagRA® S only once.

2.4 Basics

DiagRA® S is delivered with multiple tools for executing simulation, creation of simulation files and for further operations on simulation files. This chapter presents the overview of all executables delivered together with DiagRA® S. All paths are given relative to the installation folder of DiagRA® S.

Executables with user interface (UI):

- DiagRA_S.exe: diagnostic communication simulation tool
- DataManager.exe: authoring tool for DiagRA® S simulation files (.dsproj XML files)
- LogViewer.exe: log viewer tool for quick overview of diagnostic log files content
- RA D-PDU API\DiagRA S\{x86,x64}\DeviceSelection.exe: configuration tool for RA D-PDU-API delivered with DiagRA S (version for usage with DiagRA S itself)
- RA D-PDU API\Normal\{x86,x64}\DeviceSelection.exe: configuration tool for delivered RA D-PDU-API (version for usage with any J2535 / RP1210 compatible interface)

Executables with command line interface (CLI):

- DsprojConverter.exe: tool for automated conversion of .dsproj simulation files in any supported version to actual DiagRA S format
- DsprojCreator.exe: tool for creation / updating of .dsproj simulation files based on supported diagnostic communication log files
- DoIPServer.exe: tool for DoIP simulation like the DoIP simulation integrated in DiagRA S (NOTE: tool can't be run simultaneously with DiagRA® S, because both tools are trying to acquire the same TCP/IP port). Useful for automated tests.
- PassThruServer.exe: tool for J2534 (PassThru) simulation like the J2534 simulation "Server" in DiagRA S. Useful for automated tests.
- PassThruRegistryUpdate.exe and x64/PassThruRegistryUpdate.exe: *needed for internal usage only* (for correct deinstallation of DiagRA S from Windows registry)
- RA D-PDU API\DiagRA S\{x86,x64}\FileBuilder.exe: *needed for internal usage only*
- RA D-PDU API\Normal\{x86,x64}\FileBuilder.exe: tool for updating the list of devices available for RA D-PDU-API (you might need to execute these executables in case you've installed new J2534 / RP1210 devices in your system)

2.5 Quick start

The minimum effort to simulate diagnostic communication with DiagRA® S is:

1. Open the application „*DiagRA_S.exe*“.
2. Create a new project over main menu entry "File -> New project" (in case you don't have correct DiagRA S simulation project (see also 4.1a)).
3. Open the Data Manager using menu "Current project -> Data Manager" or corresponding button (see also 4.1a.2).
4. Import one or more log files (see also 4.2a).

5. Select in DiagRA S the project you want to work ("File -> Select project" or "File -> Find project").
6. Start the communication in your application using SAE J2534 / RP1210 / DoIP interface targeting DiagRA® S, e.g. Silver Scan-Tool, DiagRA® D, DiagRA® Embedded, DiagRA® LE, SAE J1699-3 Compliance Test or any other diagnostic tool supporting SAE J2534 / RP1210 / DoIP / D-PDU-API interfaces.

3 Projects

Projects are used to group ECUs, which belong together, e.g. are part of the same vehicle. The chapter Functions of the Data Manager describe how you can create, edit and delete projects. When you are importing a log file, the information saved in this log file are used to create virtual ECUs for this project.

Projects of type "Diagnostics" contain an additional node called "CAN bus". When loading CAN Traces from IXXAT and Vector, you have the option to add those CAN Frames to the CAN bus to enable a rest bus simulation.

Currently, there are three project types supported by DiagRA® S:

- Diagnostics
- SAE J1699-3
- SAE J1939-73/-84

Remarks to SAE J1699-3 projects

To run the SAE J1699-3 compliance test, you need a log file of version 14.06.02 or later. The communication is only supported for ISO 15765.

Please note that DiagRA® S is only able to respond to requests which were saved in the log file. If you are running a SAE J1699-3 test to which there is no information in DiagRA S project file, DiagRA® S won't be able to respond to requests of those tests. To make sure that the SAE J1699-3 will be successful, you need to run the SAE J1699-3 compliance test with an imported log file of the same application version.

You also must note, that aborting test 5.18 is not allowed, because DiagRA® S won't be able to synchronize with the compliance test, after aborting test 5.18.

3.1 Schema

Starting with version of DiagRA® S the XSD schema description is delivered in the installation folder (subfolder "schema"), e.g. "schema\DiagRA_S_Project_2.7.0.xsd" for version 2.7.0. Note that schema files can be applied for validation of project files created with DiagRA® S 2.4.0 or newer, but not with earlier versions of DiagRA® S since format was incompatibly changed in DiagRA® S 2.4.0. However, DiagRA® S is still able to read old project files (backward compatibility) and to convert them to an actual project format. This can be done in two ways:

- automatically on opening of project file in DataManager (if corresponding setting is set to "Show always" (default), see Settings -> Display Options -> "Show project conversion dialog").
- in a batch operation using command line tool delivered with DiagRA S, called DsprojConverter.exe.

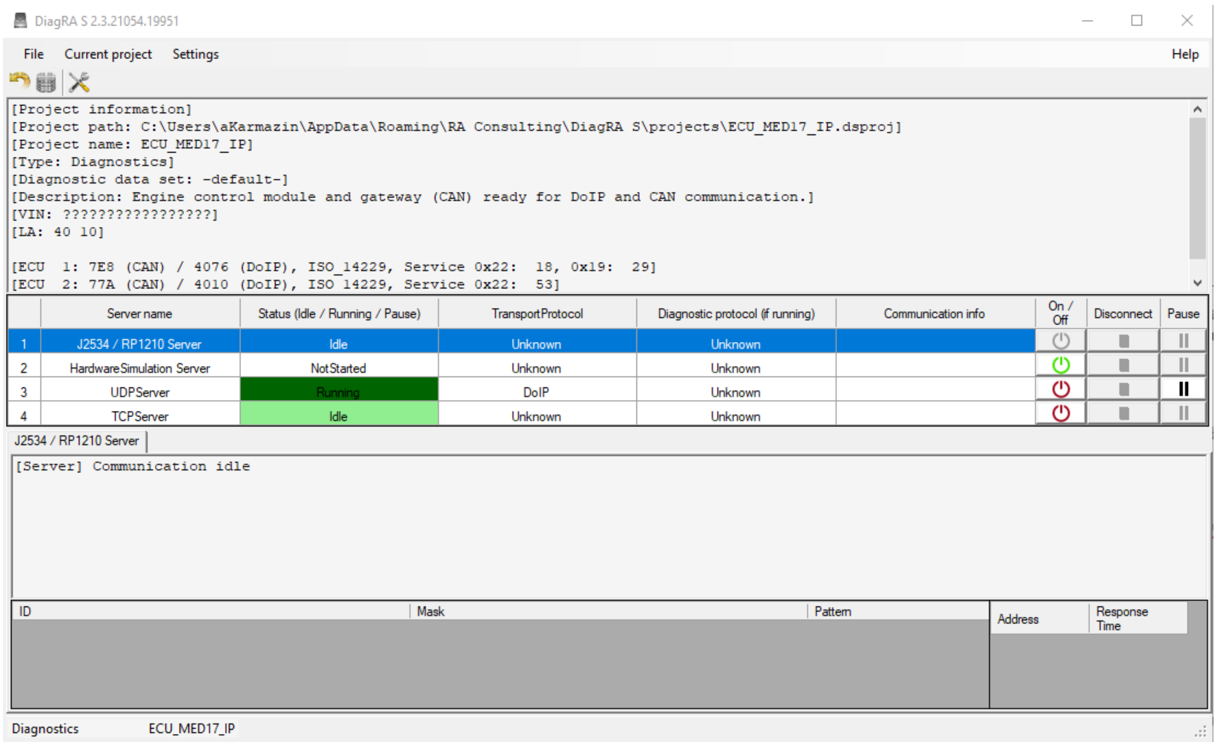
DsprojConverter.exe tool supports two arguments:
--fileOrFolder "<fullPathToFileOrFolder>"
--backup <true/false>

If folder path is provided as an argument, all project files in folder will be converted to the actual DiagRA S format. Backup setting controls if backups for project files in old format needs to be created (i.e. as MyProjectName.dsproj__{OLD_VERSION}.backup)

4 Overview

4.1 DiagRA® S

You should see the following window after you've started DiagRA® S.



```
[Project information]
[Project path: C:\Users\aKarmazin\AppData\Roaming\RA Consulting\DiagRA S\projects\ECU_MED17_IP.dsproj]
[Project name: ECU_MED17_IP]
[Type: Diagnostics]
[Diagnostic data set: -default-]
[Description: Engine control module and gateway (CAN) ready for DoIP and CAN communication.]
[VIN: ?????????????????]
[LA: 40 10]

[ECU  1: 7E8 (CAN) / 4076 (DoIP), ISO_14229, Service 0x22:  18, 0x19:  29]
[ECU  2: 77A (CAN) / 4010 (DoIP), ISO 14229, Service 0x22:  53]
```

	Server name	Status (Idle / Running / Pause)	TransportProtocol	Diagnostic protocol (if running)	Communication info	On / Off	Disconnect	Pause
1	J2534 / RP1210 Server	Idle	Unknown	Unknown				
2	HardwareSimulation Server	NotStarted	Unknown	Unknown				
3	UDPServer	Running	DoIP	Unknown				
4	TCPServer	Idle	Unknown	Unknown				

```
[Server] Communication idle
```

ID	Mask	Pattern	Address	Response Time

The window is divided in two areas. You'll find the main menu at the top of the window. The most important functions are also available in the tool bar below the main menu.

The central and lower part of the window consists of three parts (from above to bottom):
- project detail text area (contains textual overview about currently selected .dsrpoj file)
- table with "simulation servers" and their status
- tab control with communication outputs for selected "simulation server"

All simulation servers can run in parallel. You can switch between servers while selecting corresponding line in servers' table. Depending on the project type and project data definitions, some servers might be unavailable for certain projects:

- HardwareSimulation requires project of type "Diagnostics" with valid diagnostic modules (ISO 14229, ISO 15765)
- UDPServer and TCPServer are available only if correct DoIP identification data is present in project file

You can control servers using buttons in columns “On/Off”, “Disconnect” and “Pause”.

- “On/Off”: start / stop server. If server is not started, no active connection to this simulation server is possible.
    - J2534/RP1210 Server, UDPServer and TCPServer are started automatically.
    - J2534/RP1210 Server currently doesn’t support manual start and stop. This feature will be added in one of future releases.
    - HardwareSimulation needs to be actively started by the user since it requires physical connection to the CAN bus, which needs to be configured prior to the connection. See documentation for Hardware Simulation feature.
- “Pause”: requests are accepted, but no request processing takes place. No responses are sent back.
- “Disconnect”: disconnect from current client application.
    - J2534/RP1210 Server: disconnect from the tester application and wait for next connection. Tester will have to actively reconnect again. Available only if any diagnostic tester is currently connected.
    - TCPServer: disconnect current TCP socket and wait for next connection. Available only if any TCP connection to tester is active.
    - UDPServer, HardwareSimulation: not available.

Below the server table you can find one or more tabs. After starting the application, there is at least one tab called “XXXServer” present, e.g. J2534/RP1210 Server. It shows the communication details for the currently selected simulation server. In this tab, you will see a log of the communication between DiagRA® S and its underlying interfaces and you can see all communicating ECUs and the message filters during an active communication. After establishing a communication with one of the simulated interfaces, more tabs will be added, which are protocol dependent. For example: You’ll see the tabs Mode 01 to Mode 0A for a communication over SAE J1979 and each of them is showing the communication of its Mode

The buttons in the toolbar are (from left to right):

- Reset Project
- Data Manager
- Settings

a. Functions of main window

a.1 Reset project

Project will be loaded again. Internal communication state will be reverted.

a.2 Data Manager

Using this button, you can open the Data Manager, which will be described later in this document. In the Data Manager, you can import log files to create virtual ECUs and you can add, modify and delete values of the ECU. You can also create completely new ECUs, display values of an ECU and much more.

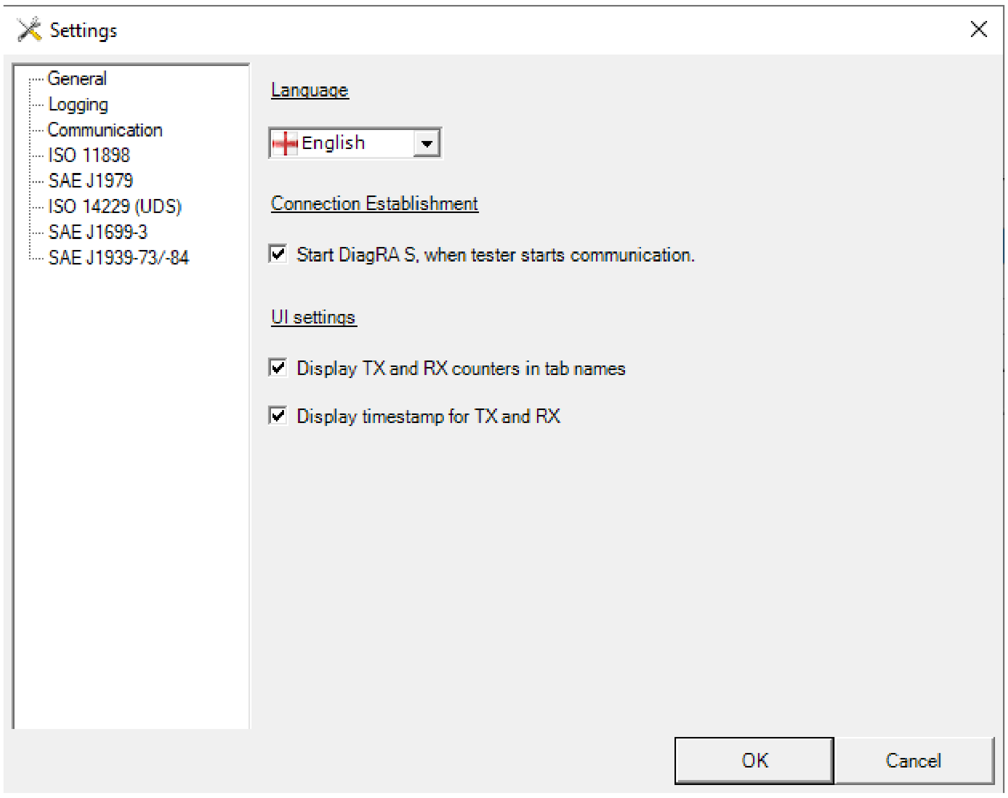
a.3 Settings

Using this button, you can open corresponding dialog for DiagRA® S settings.

b. General Settings

On the first page, you'll find the language selection. Currently, there are three languages supported by DiagRA® S: English, German and Chinese. After selecting a new language, the Settings dialog will automatically be updated. After confirming the settings, the rest of the application will update, too.

If the second option is selected, a tester is able to start DiagRA® S remote-controlled. When the tester starts the communication with one of the virtual interfaces and DiagRA® S is not running, it will be started.



Language

English

Connection Establishment

☑ Start DiagRA S, when tester starts communication.

UI settings

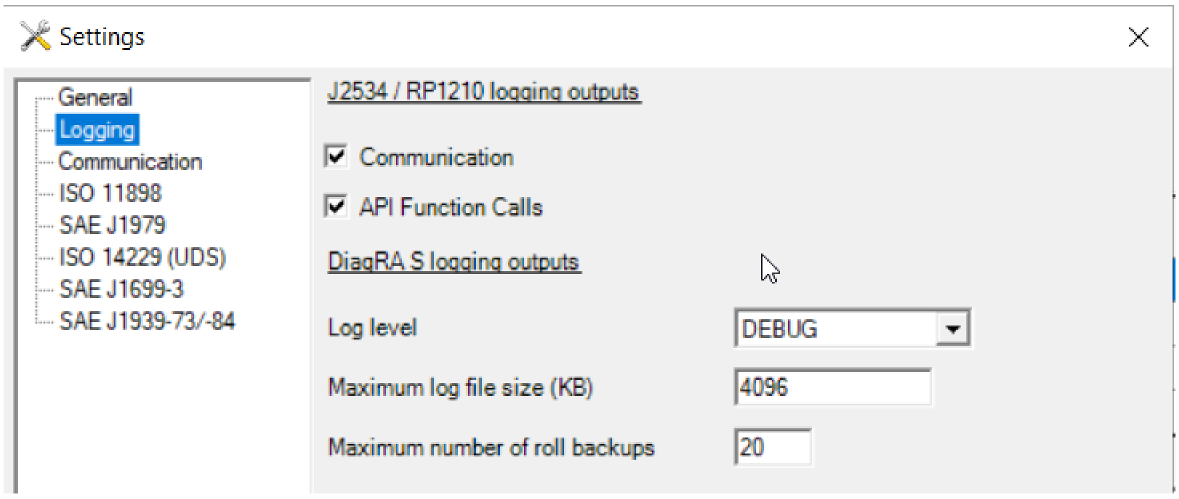
☑ Display TX and RX counters in tab names

☑ Display timestamp for TX and RX

b.1 Logging

"Communication" and "API Function Calls" controls the logging outputs of SAE J2534 / RP1210 interfaces.

Logging settings of DiagRA S itself can be configured in subsection "DiagRA S logging outputs controls": log level, maximum log file size in KB and maximum number of roll backups.



J2534 / RP1210 logging outputs

☑ Communication

☑ API Function Calls

DiagRA S logging outputs

Log level	DEBUG
Maximum log file size (KB)	4096
Maximum number of roll backups	20

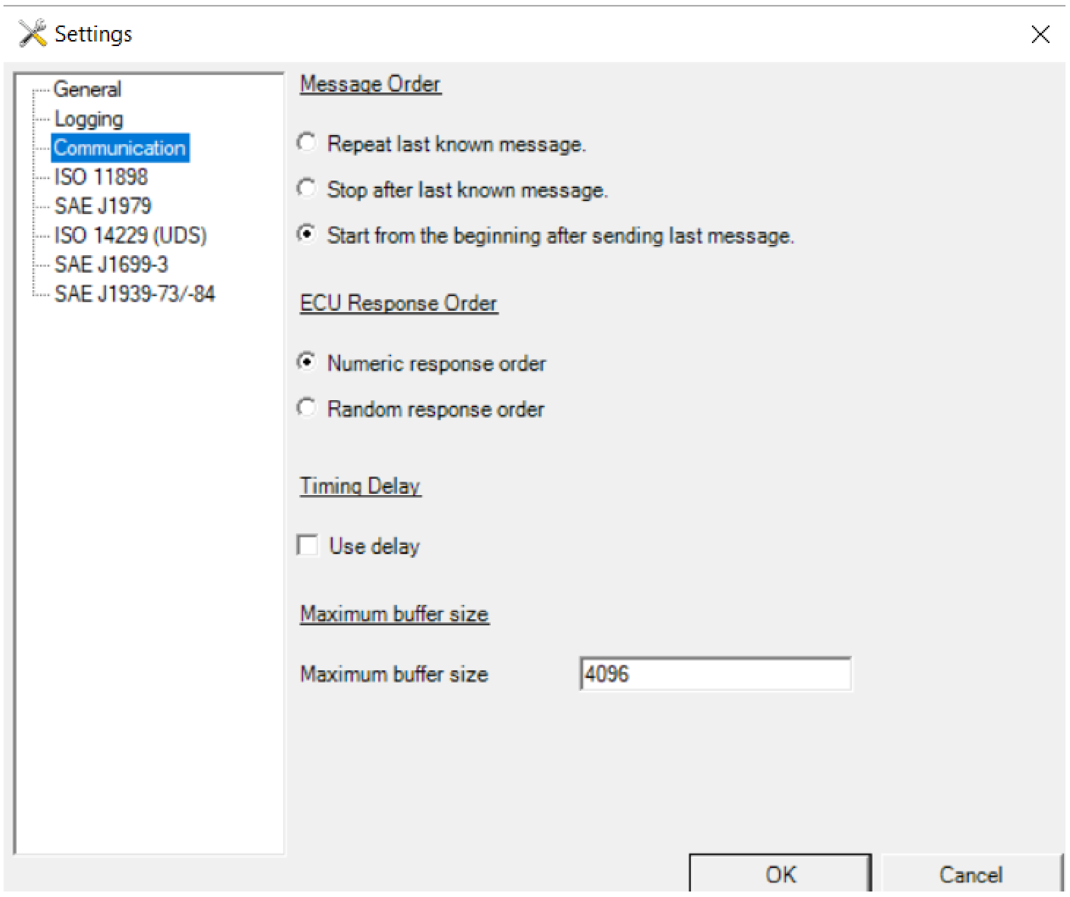
b.2 Communication Settings

When importing one or more log files, the saved communication is used to create virtual ECUs or a virtual CAN bus. The number of stored messages may be large, but it’s finite. That means, if the communication with DiagRA® S runs long enough, the last known message will be reached at any time. The message order option sets the behavior of DiagRA® S when the last known message is sent. The three options are (from top to bottom):

1. The last known message is repeated.
2. The communication stops after the last known message has been sent.
3. After the last known message has been sent, the communication restarts at the first known message.

The second option on this page is the ECU response order. The default option, *Numeric response order*, states that all ECUs respond in a numerical order to a request. For example, if you have two ECUs with CAN identifiers 7E8 and 7E9 and you are using the OBD protocol, the ECU 7E8 will always respond before 7E9. The *Random response order* randomizes the responses, i.e. you can’t say which ECU responds first and which responds last. This option can be used only in case response delay option is turned off (see below).

The next option handles the activation of the response delay. This feature, which is explained in detail in chapter 6, enables the user to change the response timings of ECUs and single responses during runtime. The “Maximum buffer size” sets the maximal number of Bytes DiagRA® S accepts in an incoming message. Any messages that are longer will result in a Overflow-FlowControl message.



Message Order

( ) Repeat last known message.
( ) Stop after last known message.
(●) Start from the beginning after sending last message.

ECU Response Order

(●) Numeric response order
( ) Random response order

Timing Delay

☐ Use delay

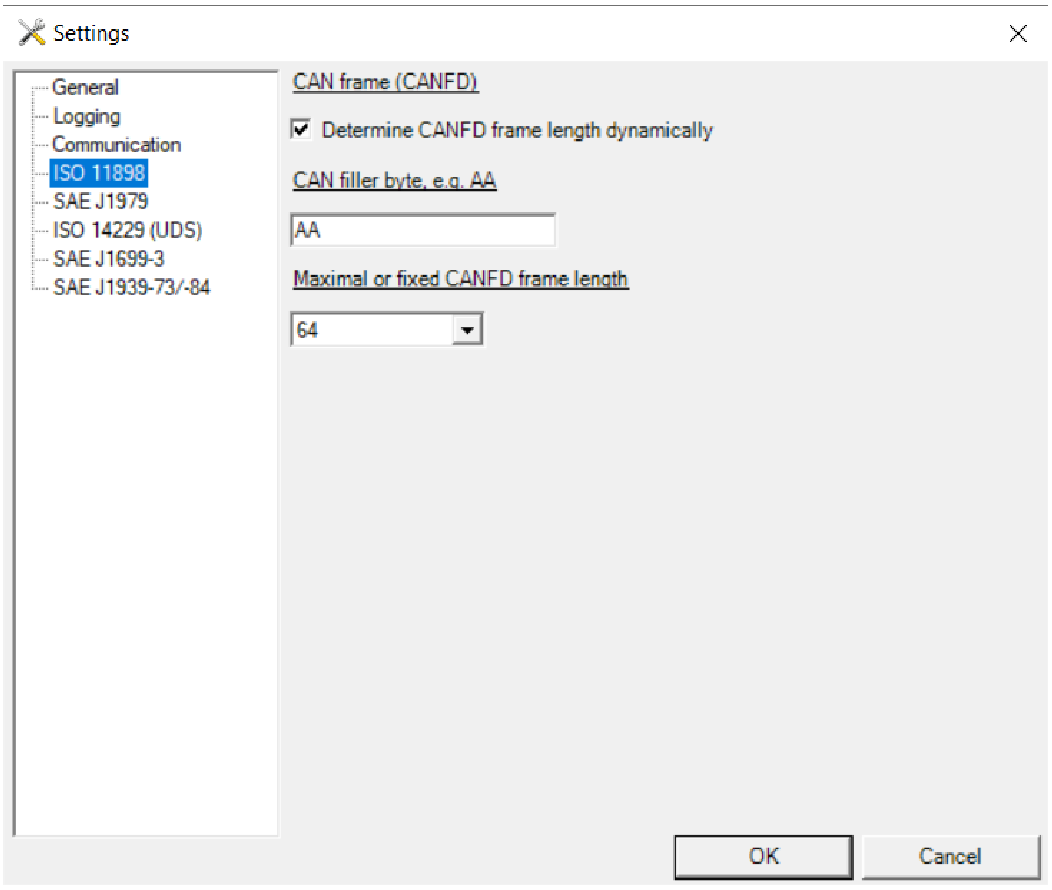
Maximum buffer size

Maximum buffer size: 4096

b.3 ISO 11898

For ISO 11898 (and ISO-TP, i.e. ISO 15765-2) following settings are available:

1) Determine CANFD frame length dynamically (or use fixed CANFD frame length)
2) Value of CAN filler byte
3) Maximal (or fixed) CANFD frame length



CAN frame (CANFD)

☑ Determine CANFD frame length dynamically

CAN filler byte, e.g. AA

AA

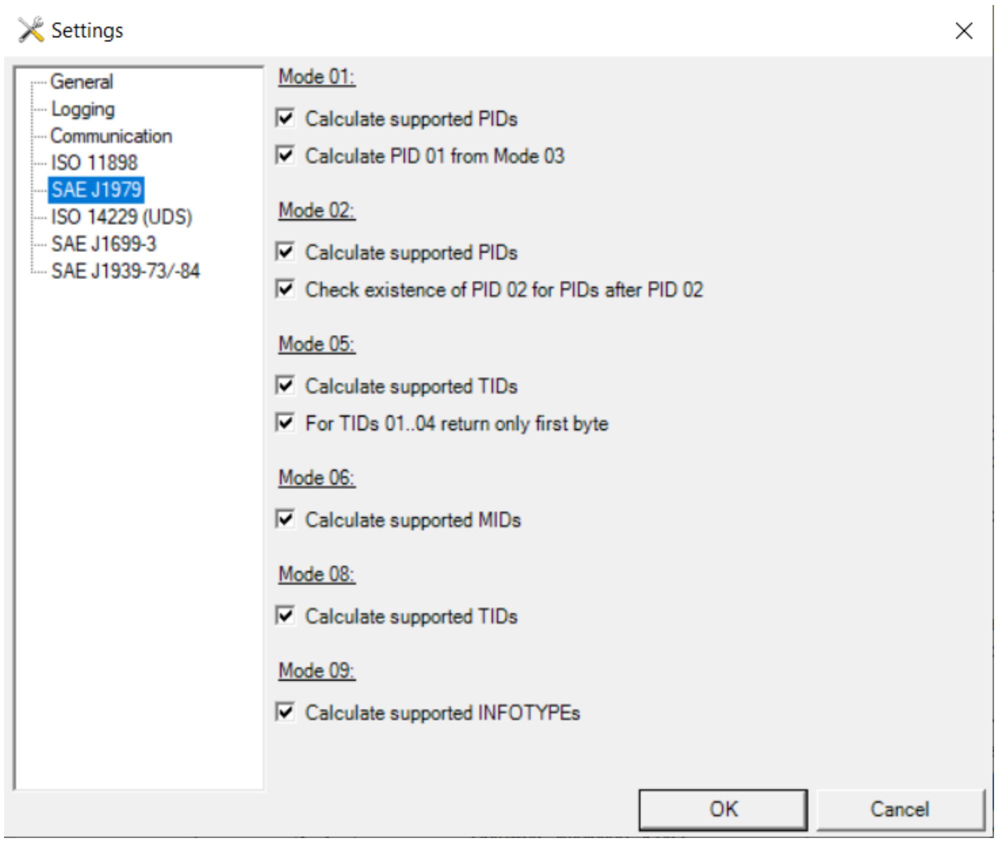
Maximal or fixed CANFD frame length

64

b.4 SAE J1979

For OBD on CAN (J1979) following settings are available:

1) PIDs, TIDs, MIDs, INFOTYPEs calculation. If enabled, values are calculated by DiagRA® S itself (default).
2) Mode 2: require existence of PID 02 for PIDs after PID 02. If PID 02 is not defined in project file, all other PIDs after PID 02 (e.g. PID 03) will be not answered.
3) Mode 5: for TIDs 01..04 the answer can be limited to first byte (as standard defines it).


Settings

- General
- Logging
- Communication
- ISO 11898
- SAE J1979
- ISO 14229 (UDS)
- SAE J1699-3
- SAE J1939-73/-84

Mode 01:

- [✓] Calculate supported PIDs
- [✓] Calculate PID 01 from Mode 03

Mode 02:

- [✓] Calculate supported PIDs
- [✓] Check existence of PID 02 for PIDs after PID 02

Mode 05:

- [✓] Calculate supported TIDs
- [✓] For TIDs 01..04 return only first byte

Mode 06:

- [✓] Calculate supported MIDs

Mode 08:

- [✓] Calculate supported TIDs

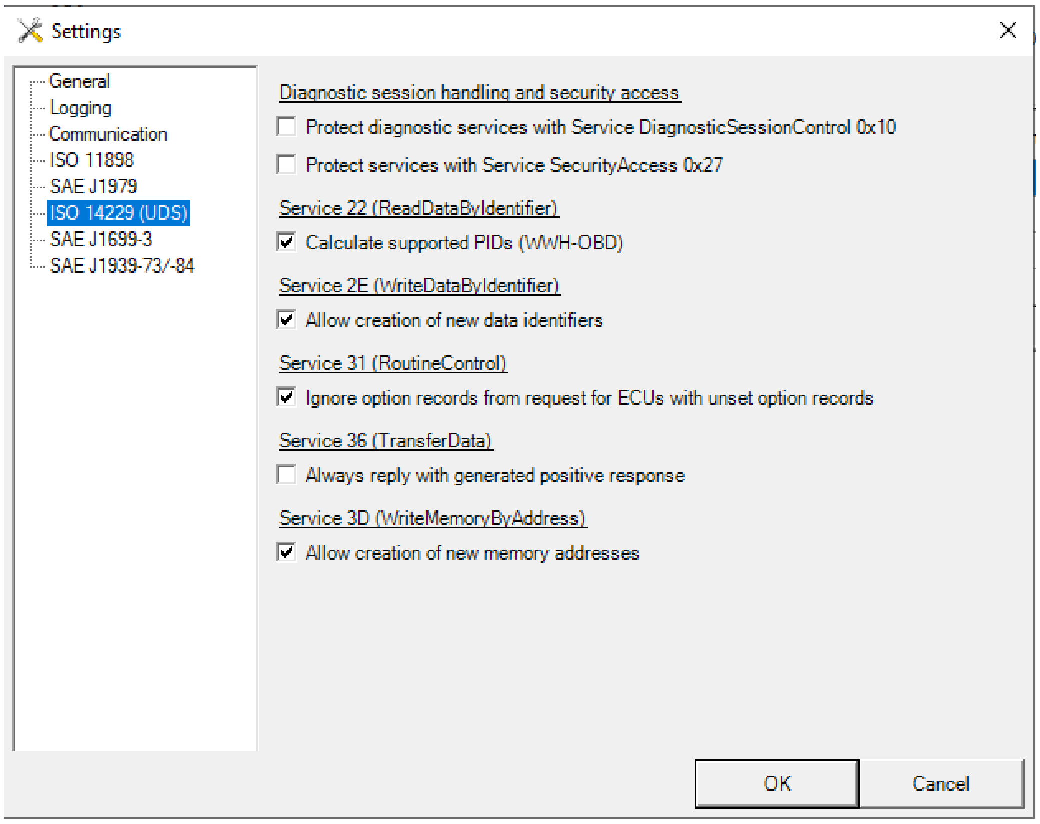
Mode 09:

- [✓] Calculate supported INFOTYPEs

b.5 ISO 14229 (UDS)

For UDS (ISO 14229) there are following settings available:

1) Protecting diagnostic services using DiagnosticSessionControl (according to UDS standard). By default, this option is deactivated.
2) Protecting diagnostic services using SecurityAccess service (according to UDS standard). By default, this option is deactivated.
3) PIDs, TIDs, MIDs, INFOTYPEs calculation for WWH-OBD (ISO 27145) and OBD on UDS (SAE J1979-2). If enabled, values are calculated by DiagRA® S itself (default).
4) Two options for controlling behavior of DiagRA® S in case new data identifier (Service 2E – WriteDataByIdentifier) or new memory address (Service 3D – WriteMemoryByAddress) is written to DiagRA® S: if option is activated (default), DiagRA® S will create new data identifier (memory address) and allow reading it. If option is deactivated, DiagRA® S will answer such service execution with negative response. If data identifier (memory address) is already defined in dsproj file, DiagRA® S will add corresponding value to the list of answers and it will be returned on next read service execution for this data identifier (memory address).
5) One option for controlling behavior of handling optionRecords for Service31 RoutineControl: “Ignore option records from request for ECUs with unset option records”. If this option is activated (by default it is deactivated) and for ECU no optionRecord values are defined at all, it is assumed that all optionRecords are equally valid and the response is calculated without matching of optionRecords (behavior of DiagRA S before version 2.3.0).
6) One option for controlling behavior of handling requests for Service 36 TransferData (which are typically very long): “Always reply with generated positive response”. If this option is activated (by default it is deactivated), all requests for these service will be responded with generated positive response, independent of the fact that corresponding request is stored in the project file or not.



Diagnostic session handling and security access

- [ ] Protect diagnostic services with Service DiagnosticSessionControl 0x10
- [ ] Protect services with Service SecurityAccess 0x27

Service 22 (ReadDataByIdentifier)

- [✓] Calculate supported PIDs (WWH-OBD)

Service 2E (WriteDataByIdentifier)

- [✓] Allow creation of new data identifiers

Service 31 (RoutineControl)

- [✓] Ignore option records from request for ECUs with unset option records

Service 36 (TransferData)

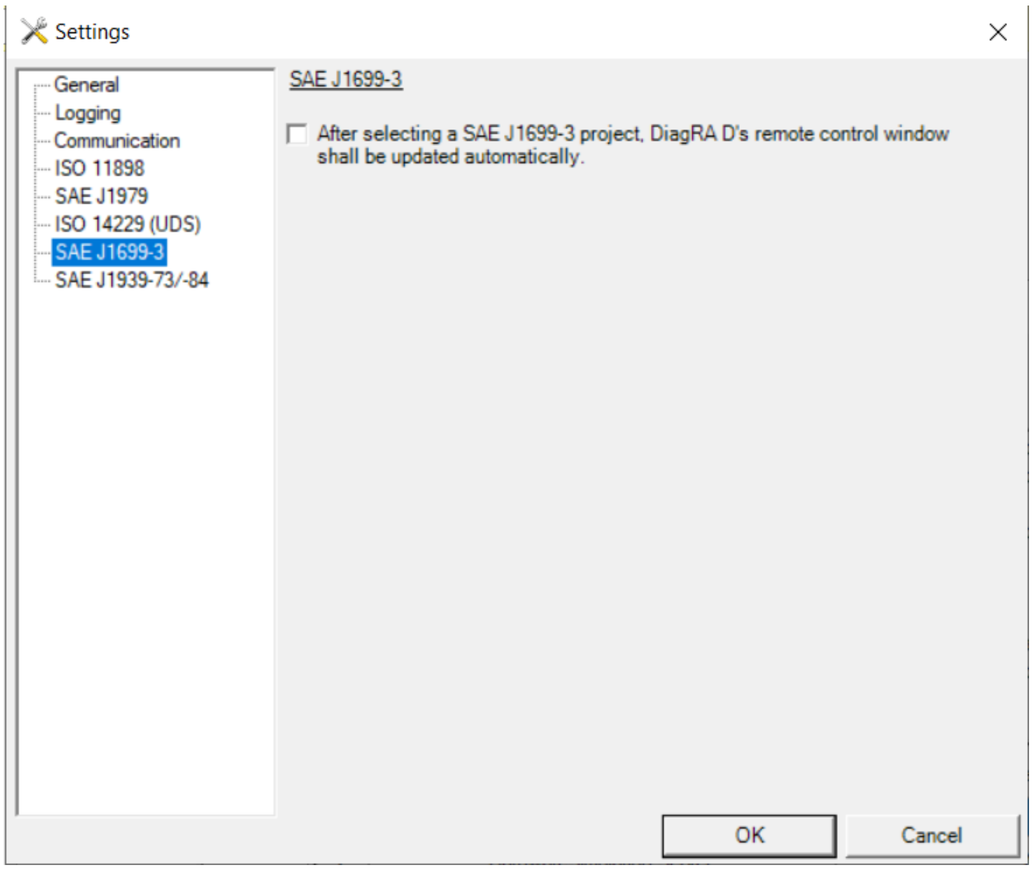
- [ ] Always reply with generated positive response

Service 3D (WriteMemoryByAddress)

- [✓] Allow creation of new memory addresses

b.6 SAE J1699-3

DiagRA® D offers a comfortable GUI to run the SAE J1699-3 Compliance Test and DiagRA® S offers an easy way to configure it. If this option is selected, DiagRA® S will change the settings of DiagRA® D's remote control of SAE J1699-3, so that you don't need to change any settings in DiagRA® D manually.

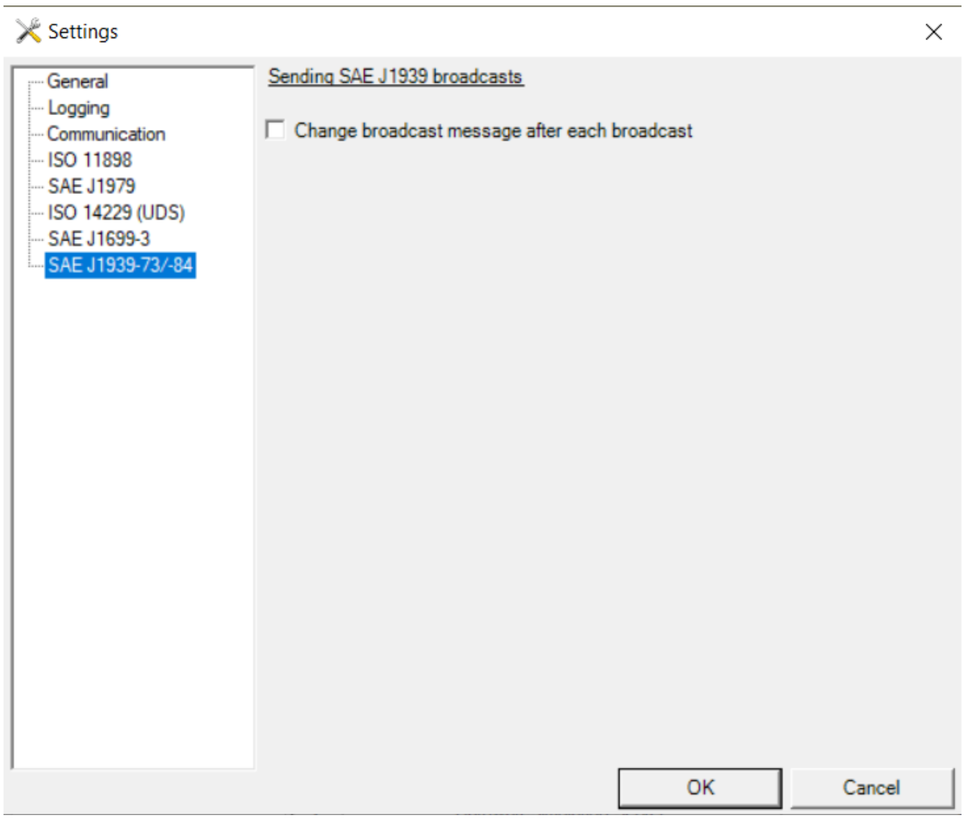


SAE J1699-3

☐ After selecting a SAE J1699-3 project, DiagRA D's remote control window shall be updated automatically.

b.7 SAE J1939-73/-84

Many Parameter Groups will be sent using a broadcast message. The message will change each time, if this option is selected. If it's not selected, the message will only change after a request was received.



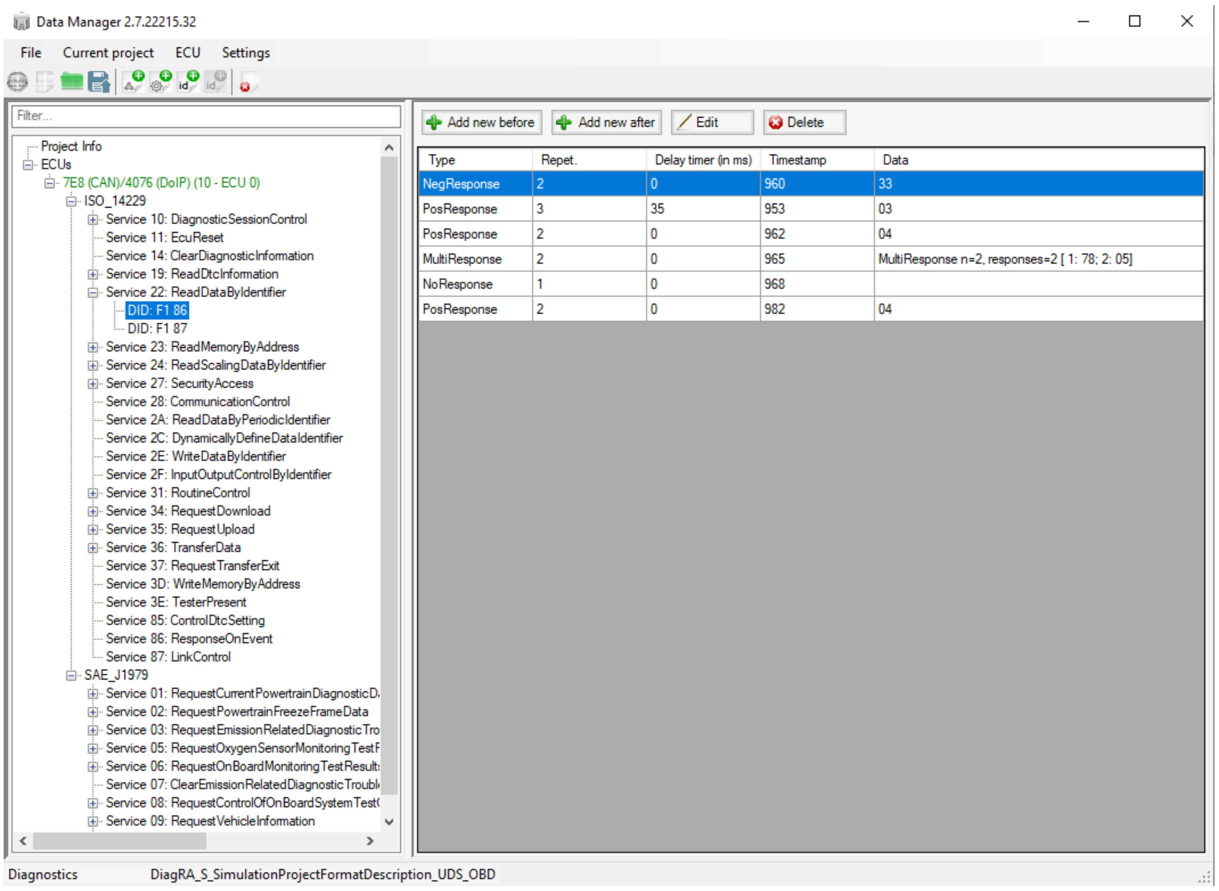
Settings

- General
- Logging
- Communication
- ISO 11898
- SAE J1979
- ISO 14229 (UDS)
- SAE J1699-3
- SAE J1939-73/-84

Sending SAE J1939 broadcasts

☐ Change broadcast message after each broadcast

4.2 Data Manager



```
Project Info
ECUs
  7E8 (CAN)/4076 (DoIP) (10 - ECU 0)
    ISO_14229
      Service 10: DiagnosticSessionControl
      Service 11: EcuReset
      Service 14: ClearDiagnosticInformation
      Service 19: ReadDtcInformation
      Service 22: ReadDataByIdentifier
        DID: F1 86
        DID: F1 87
      Service 23: ReadMemoryByAddress
      Service 24: ReadScalingDataByIdentifier
      Service 27: SecurityAccess
      Service 28: CommunicationControl
      Service 2A: ReadDataByPeriodicIdentifier
      Service 2C: DynamicallyDefineDataIdentifier
      Service 2E: WriteDataByIdentifier
      Service 2F: InputOutputControlByIdentifier
      Service 31: RoutineControl
      Service 34: RequestDownload
      Service 35: RequestUpload
      Service 36: TransferData
      Service 37: RequestTransferExit
      Service 3D: WriteMemoryByAddress
      Service 3E: TesterPresent
      Service 85: ControlDtcSetting
      Service 86: ResponseOnEvent
      Service 87: LinkControl
    SAE_J1979
      Service 01: RequestCurrentPowertrainDiagnosticD
      Service 02: RequestPowertrainFreezeFrameData
      Service 03: RequestEmissionRelatedDiagnosticTro
      Service 05: RequestOxygenSensorMonitoringTestF
      Service 06: RequestOnBoardMonitoringTestResult
      Service 07: ClearEmissionRelatedDiagnosticTroubl
      Service 08: RequestControlOfOnBoardSystemTest(
      Service 09: RequestVehicleInformation
```

Type	Repet.	Delay timer (in ms)	Timestamp	Data
NegResponse	2	0	960	33
PosResponse	3	35	953	03
PosResponse	2	0	962	04
MultiResponse	2	0	965	MultiResponse n=2, responses=2 [1: 78; 2: 05]
NoResponse	1	0	968	
PosResponse	2	0	982	04

Diagnostics DiagRA_S_SimulationProjectFormatDescription_UDS_OBD

The image above shows the opened Data Manager for a project of type “Diagnostics”. For all project types, the left tree view will contain following nodes:

- Project Info: contains brief overview over project content
- CAN Bus: contains CAN messages to be sent during rest bus simulation
- ECUs: parent node for all ECU instances

Each ECU node contains identification data for the ECU (CAN ID, DoIP Logical Ecu Address). If setting “Show ECU name after each identifier” is activated and for project file the correct diagnostic data set is chosen, the ECU node will also contain some text corresponding to ECU.

Remark. Diagnostic data set is the definition of diagnostic data according to DiagRA D format (adw.xml and .cod files). It is only possible if DiagRA® D is installed and corresponding files are defined in folder structure at “C:\Users\Public\Documents\RA Consulting\DiagRA D\codini”

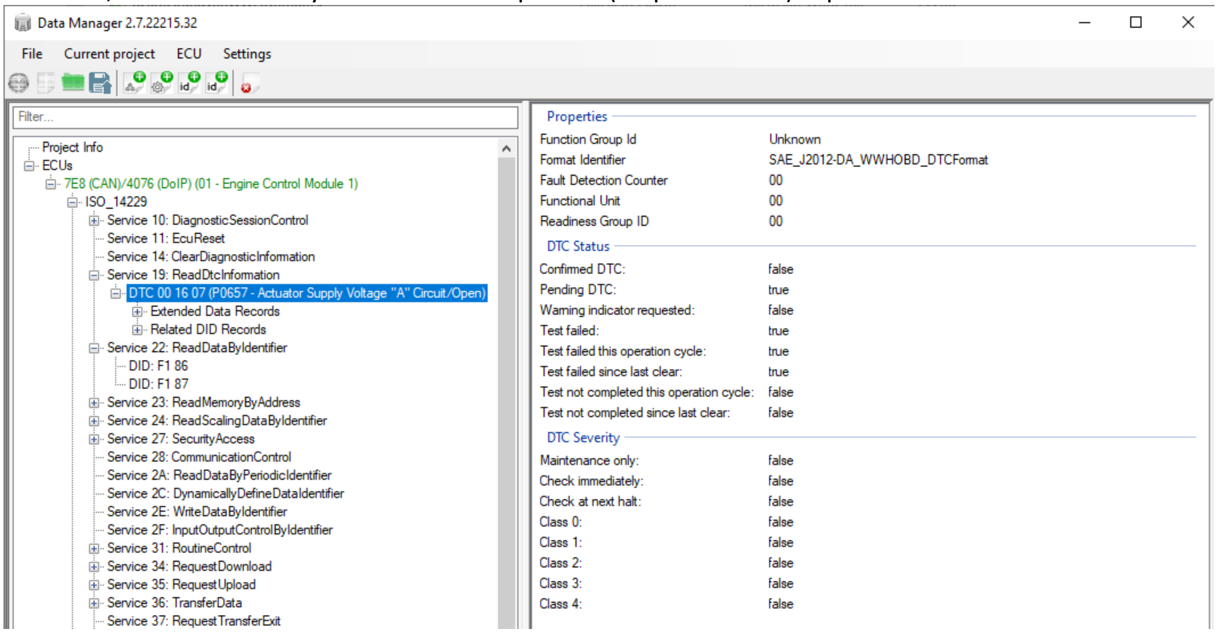
If you fold the ECU node in the tree view, you can see all content for the selected ECU. The content in a tree view depends on the type of the selected project. For diagnostic projects, all services will be displayed as nodes and all DTCs and values (e.g. DID) will be child nodes of the proper service nodes.

For SAE J1939 projects, all Parameter Groups of the selected ECU are listed in the tree view. If there are broadcast or additional SPN information about a PGN, they will be displayed as child nodes of the corresponding PGN.

For SAE J1699-3 projects, all tests are listed, which were made with the selected ECU. If a test failed, its text will be displayed in red, otherwise it will be displayed in green.

PGNs of SAE J1939, DTCs and values of diagnostic projects have a check box next to their name. They are used to display values and DTCs in the diagram. So far, the diagram in the center of the window was empty. If you select one or more values or DTCs they are displayed in the diagram.

For some of selected modes the details can be displayed on the right side, e.g. for diagnostic modules, DTCs and for any container of responses (response data) represented as a table.



```
Project Info
ECUs
  7E8 (CAN)/4076 (DoIP) (01 - Engine Control Module 1)
    ISO_14229
      Service 10: DiagnosticSessionControl
      Service 11: EcuReset
      Service 14: ClearDiagnosticInformation
      Service 19: ReadDtcInformation
        DTC 00 16 07 (P0657 - Actuator Supply Voltage "A" Circuit/Open)
          Extended Data Records
          Related DID Records
      Service 22: ReadDataByIdentifier
        DID: F1 86
        DID: F1 87
      Service 23: ReadMemoryByAddress
      Service 24: ReadScalingDataByIdentifier
      Service 27: SecurityAccess
      Service 28: CommunicationControl
      Service 2A: ReadDataByPeriodicIdentifier
      Service 2C: DynamicallyDefineDataIdentifier
      Service 2E: WriteDataByIdentifier
      Service 2F: InputOutputControlByIdentifier
      Service 31: RoutineControl
      Service 34: RequestDownload
      Service 35: RequestUpload
      Service 36: TransferData
      Service 37: RequestTransferExit
```

Properties

Function Group Id	Unknown
Format Identifier	SAE_J2012-DA_WWHOBD_DTCFormat
Fault Detection Counter	00
Functional Unit	00
Readiness Group ID	00

DTC Status

Confirmed DTC:	false
Pending DTC:	true
Warning indicator requested:	false
Test failed:	true
Test failed this operation cycle:	true
Test failed since last clear:	true
Test not completed this operation cycle:	false
Test not completed since last clear:	false

DTC Severity

Maintenance only:	false
Check immediately:	false
Check at next halt:	false
Class 0:	false
Class 1:	false
Class 2:	false
Class 3:	false
Class 4:	false

a. Main menu

At the top of the window, you can see the main menu of the Data Manager. The most important functions are also available in the tool bar right below the main menu:

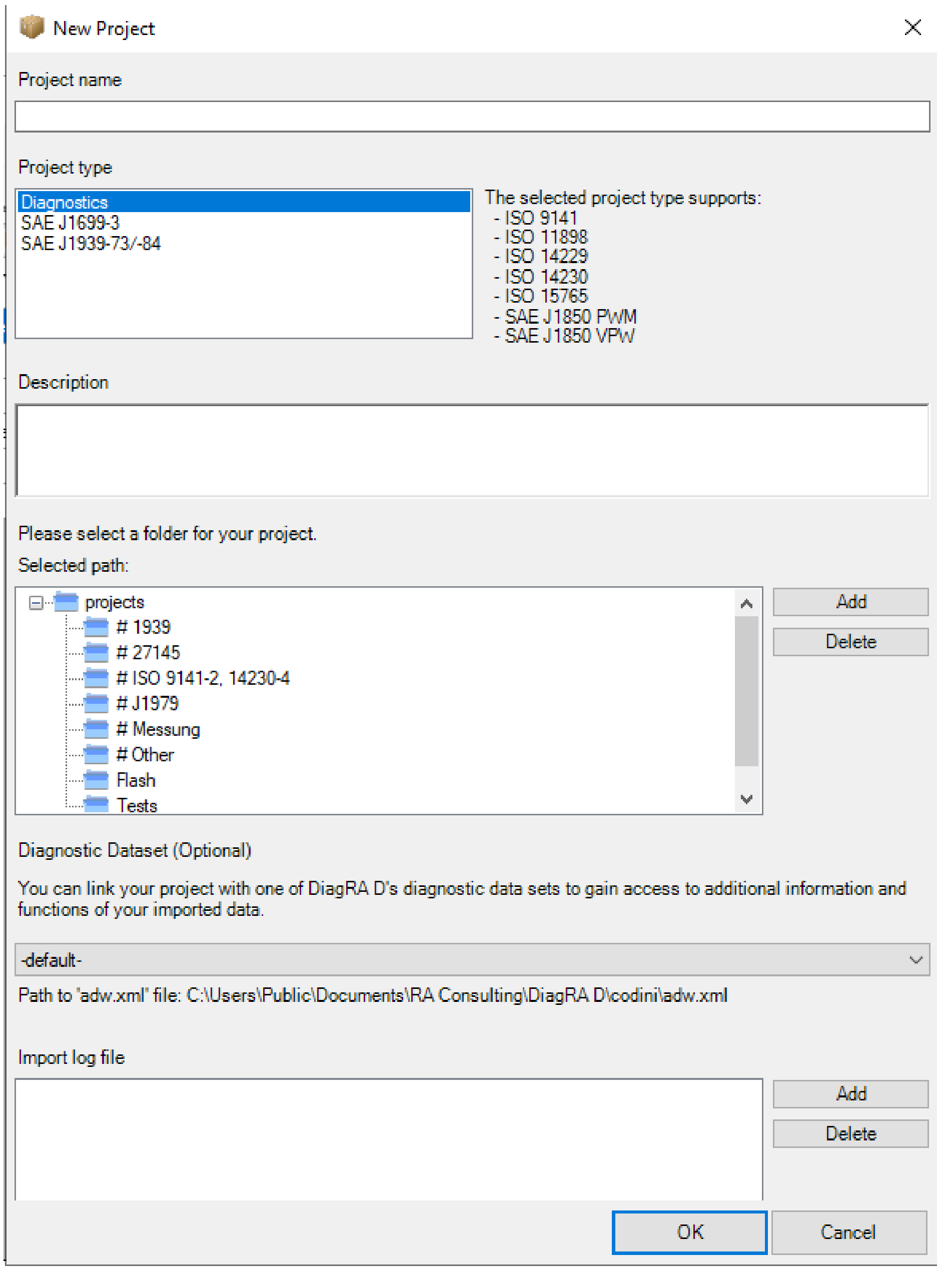
1. File
   - New project
     - Opens dialog for creation of new project
   - Select project
     - Provides submenu for selection of the project file
   - Recently opened projects
     - Contains 10 last open projects, choosing menu items selects this project again
   - Find project
     - Opens dialog for searching projects by their properties
   - Refresh project list
     - Updates the list of projects for “Select project” menu
   - Import log file
     - Opens „Import of log files” dialog
   - Recently used log files
     - Opens selected files in LogViewer.exe
2. Current project
   - Edit project
     - Opens dialog for editing main properties of project (incl. renaming it). The type of the project and its path can’t be changed.
   - Reload project
     - Reloads project from the disk
   - Synchronize with Diagnostic DataSet

        - Opens dialog for synchronization of CAN ID / DoIP Ecu Logical Address based on selection of Diagnostic DataSet
    - Open project file in default editor
        - Opens current project file in default system editor
    - Delete project
        - Deletes current project
3. ECU
    - Create new ECU
        - Opens dialog for creation of new ECU. This function is not available for SAE J1699-3 projects
    - Timing delay
        - Opens dialog for setting up the response delays (in ms) for all ECUs
    - Logfile Information
        - Shows log file informations for currently selected ECU
    - Enable / Disable ECU
        - Enables / Disable selected ECU for communication
    - Set CAN Identifier
        - Opens dialog for changing CAN Identifier for selected ECU
    - Remove ECU
        - Removes current ECU from project
4. Settings
    - Settings
        - Opens settings dialog

b. Functions of the Data Manager

b.1 New Project

Click on this button to create a new project. The following dialog will show up:


New Project

Project name

Project type

Diagnostics
SAE J1699-3
SAE J1939-73/-84

The selected project type supports:
- ISO 9141
- ISO 11898
- ISO 14229
- ISO 14230
- ISO 15765
- SAE J1850 PWM
- SAE J1850 VPW

Description

Please select a folder for your project.
Selected path:

projects
1939
27145
ISO 9141-2, 14230-4
J1979
Messung
Other
Flash
Tests

Diagnostic Dataset (Optional)

You can link your project with one of DiagRA D's diagnostic data sets to gain access to additional information and functions of your imported data.

-default-

Path to 'adw.xml' file: C:\Users\Public\Documents\RA Consulting\DiagRA D\codini\adw.xml

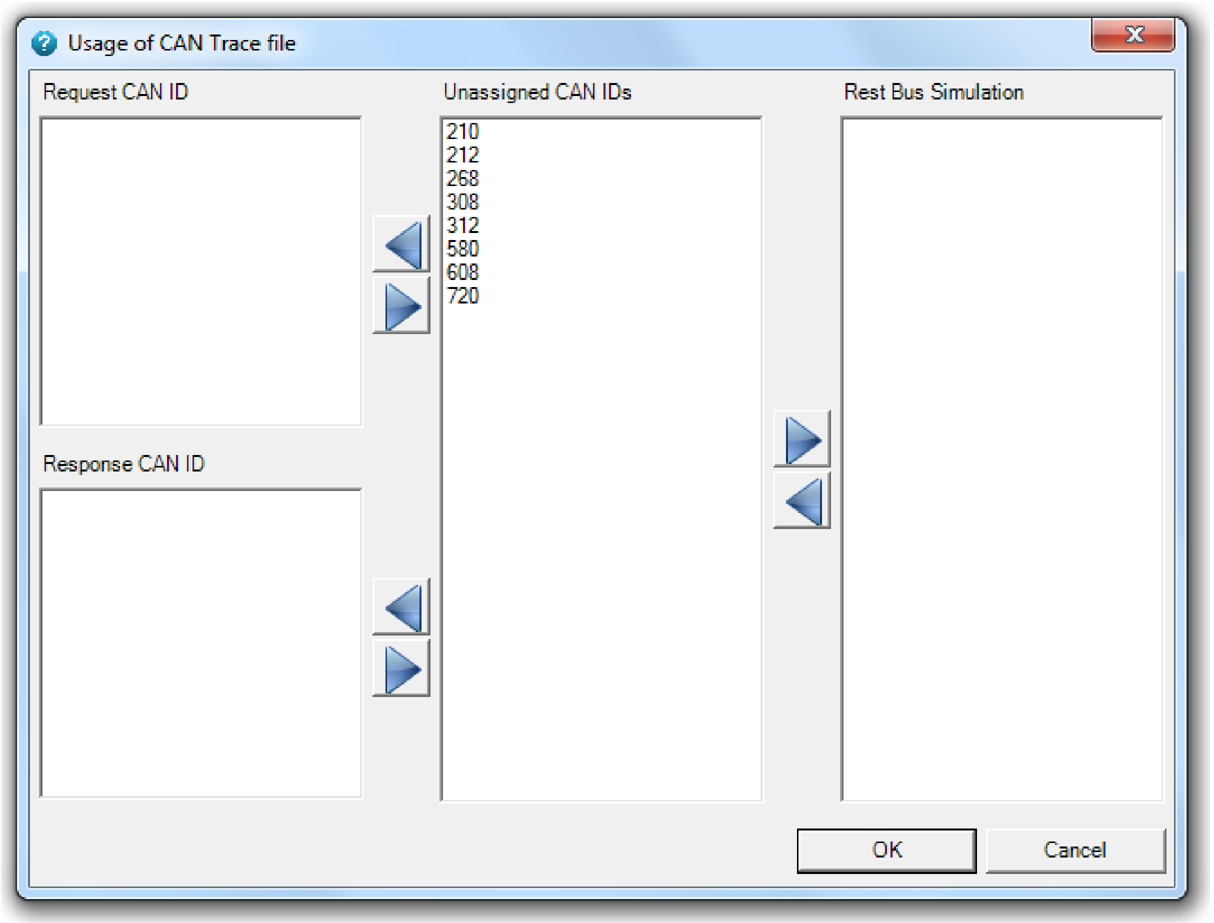
Import log file

1. You need to you need to enter a name for your project. This name can be any non-empty text.
2. Yon need to select the type of your project. You must take one of the following: SAE J1699-3 Compliance Test, SAE J1939 Heavy-Duty OBD or “Diagnostics”, which contains several diagnostic standards like OBD and UDS. The supported protocols for each project type are listed in the right part of the window.
3. You can optionally add some description to your project
4. You can optionally select a folder in which your project will be saved. You can create new if needed. The purpose of this step is to organize your projects.
5. You can optionally set up diagnostic dataset (DiagRA D® installation is required). It might be useful for some advanced visualization features, like displaying the names of ECUs in your projects and displaying textual representation for DTCs.
6. You can optionally add some log files or CAN Trace files during the project creation. These files will be imported automatically after the project is created.

b.2 Import and save log files

In order to simulate ECUs, you can import one or more log files. When importing a log file or a CAN Trace, a new ECU will be created for address found in the imported file. Each ECU gets its raw data from the recorded communication messages in the log file. That means, during a simulation each simulated ECU will respond with the same messages as they did during the real communication. Some values are an exception to this, because they have to be calculated at runtime. To save the ECUs, simply click on the “Save” button in the tool bar.

When importing a CAN Trace file, the following dialog window will show up:



210
212
268
308
312
580
608
720

All un-assigned CAN Identifier are listed in the central column. To use CAN Identifiers for the rest bus simulation, select them in the central column and click on the button with the right arrow. As you can see, the selected CAN Identifiers are moved to the right column and will be used for rest bus simulation after committing your selection by clicking on “OK”.

You can also reconstruct a diagnostic protocol like ISO 14229 from the stored CAN frames, by assigning the un-assigned CAN Identifiers from the central column to the lists of Request CAN Identifiers and Response CAN Identifiers. The CAN Identifier must come in pairs, i.e. there must be a Response CAN Identifier for each Request CAN Identifier.

CAN Identifiers which are still un-assigned, when you close the window are discarded and won’t be used for neither rest bus simulation nor diagnostic communication.

NOTE. If correct diagnostic dataset is selected, DataManager will use corresponding adw.xml file for assignment of CAN IDs to corresponding request and response CAN ID columns.

b.3 Project editing

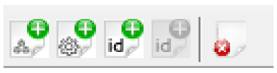
Additionally to management of ECUs, for projects of type “Diagnostics” it is possible to create / remove diagnostic modules and corresponding data definitions inside of these modules.

What is a diagnostic module?
Diagnostic module contains data definitions used for simulation of diagnostic protocol.

Extended Raw diagnostic module is a most primitive one and can be used for simulation of any diagnostic protocol. It defines full request-response pairs for diagnostic communication. It is quite simple to define, but it requires that all pairs of request-response (or request pattern with response) are defined in the diagnostic module. Every request not matching definitions of **Extended Raw** diagnostic module will be not responded by this module. It is important to know that this diagnostic module has a priority over other diagnostic modules since it is requested first on incoming diagnostic request.

Other diagnostic modules provide more complex data definitions based on corresponding diagnostic standards. I.e. **ISO 14229** diagnostic module contains data definitions for UDS communication, **SAE J1979** diagnostic module contains data definitions for emission-related diagnostics (OBD communication) and **KWP 2000** diagnostic module contains data definitions for KWP 2000 communication. The data definitions are structured by diagnostic services and their subfunctions / data identifiers. Definition of diagnostic data in these diagnostic modules allows to reduce data amount in a project file and allows to simulate some protocol specific business logic but DiagRA S itself without requiring defining all request-response pairs in **Extended Raw** module. For example, for UDS following business logic is supported: diagnostic session handling (some services can be executed only after diagnostic session is set to something else as “default”), security access handling (successful SecurityAccess (0x27) execution is pre-requirement for execution of services 0x34, 0x35, 0x36), multi-DID request handling (for ReadDataByIdentifier (0x22) or support for all ReadDtcInformation (0x19) subfunctions based on the unified model for DTC.

Creation of diagnostic modules and corresponding data definitions is currently available only over tool bar (see buttons with “+” symbol).

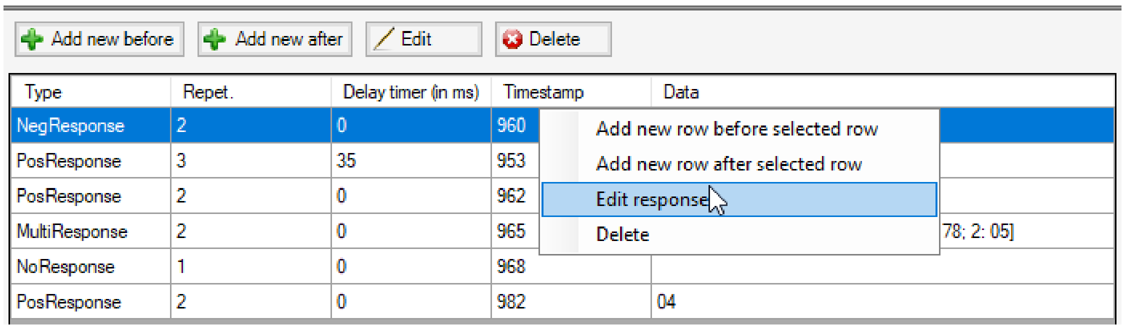


Tool bar buttons are dependent on currently selected node in a tree on the left side. Buttons are allowing (from left to right):

- add diagnostic module
- add diagnostic service / request / request pattern
- add identifier (UDS data identifier, UDS DTC etc.)
- add “sub”-identifier (option record for UDS routine identifier, extended record number for UDS DTC etc.)
- delete selected node (ECU, diagnostic module, diagnostic service, identifier, “sub”-identifier)

NOTE. Currently it is possible to create all types of diagnostic modules (**ISO 14229**, **SAE J1979**, **KWP 2000**, **Extended Raw**), but editing of data definitions is available only for **ISO 14229** and **Extended Raw** diagnostic modules. The support for editing OBD (**SAE J1979**) data definitions in Data Manager will be added in one of future releases. **KWP 2000** editing support in Data Manager will be implemented only in case it is explicitly requested by customers. To fill the data in such diagnostic modules, use log file import of manual editing of dsproj files (based on corresponding XSD schema definition and examples delivered together with DiagRA® S)

For **ISO 14229** and **Extended Raw** diagnostic modules it is also possible to edit response data tables using buttons above the response data table and context menu for table entries.


Add new before
Add new after
Edit
Delete
Add new row before selected row
Add new row after selected row
Edit response
Delete

Type	Repet.	Delay timer (in ms)	Timestamp	Data
NegResponse	2	0	960	
PosResponse	3	35	953	
PosResponse	2	0	962	
MultiResponse	2	0	965	78; 2: 05]
NoResponse	1	0	968	
PosResponse	2	0	982	04

It is important to note that only the “data record” part needs to be specified in response data table for **ISO 14229** diagnostic module excluding Service-ID, data identifier etc. (the same is valid for **SAE J1979** and **KWP 2000** as well). On the other side, the response data table for **Extended Raw** diagnostic module contains full response definitions (including service-ID, data identifier etc.).

Following four types of responses are supported:

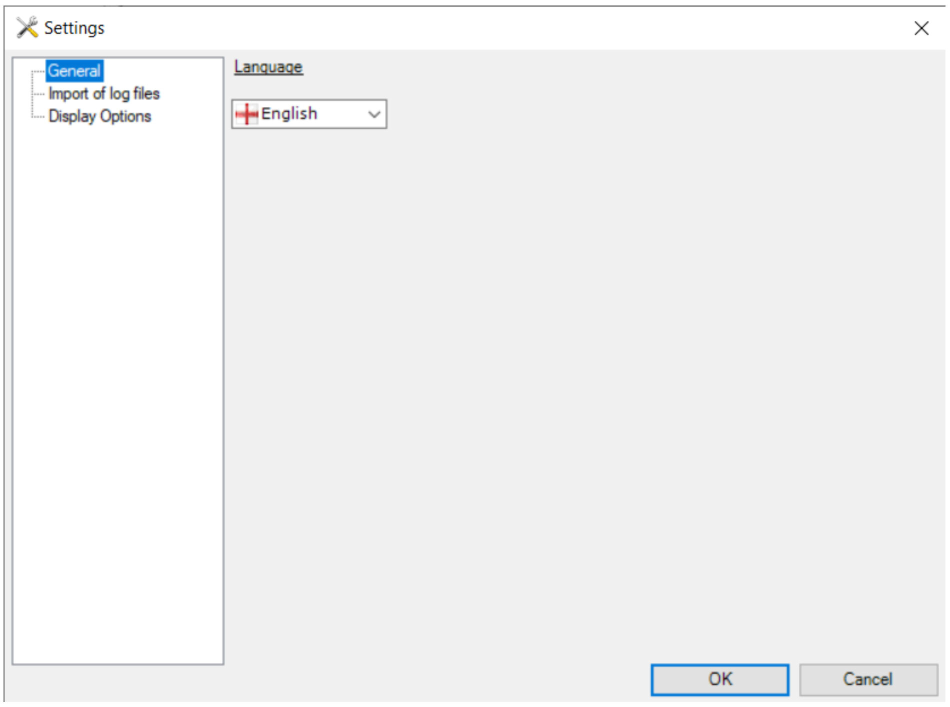
- PosResponse (positive response)
- NegResponse (negative response)
- NoResponse (response is suppressed)
- MultiResponse (multi-part response consisting of multiple response messages sent for one request)

NOTE: UDS Service 0x19 ReadDtcInformation supports only positive responses.

c. Settings

c.1 General Settings

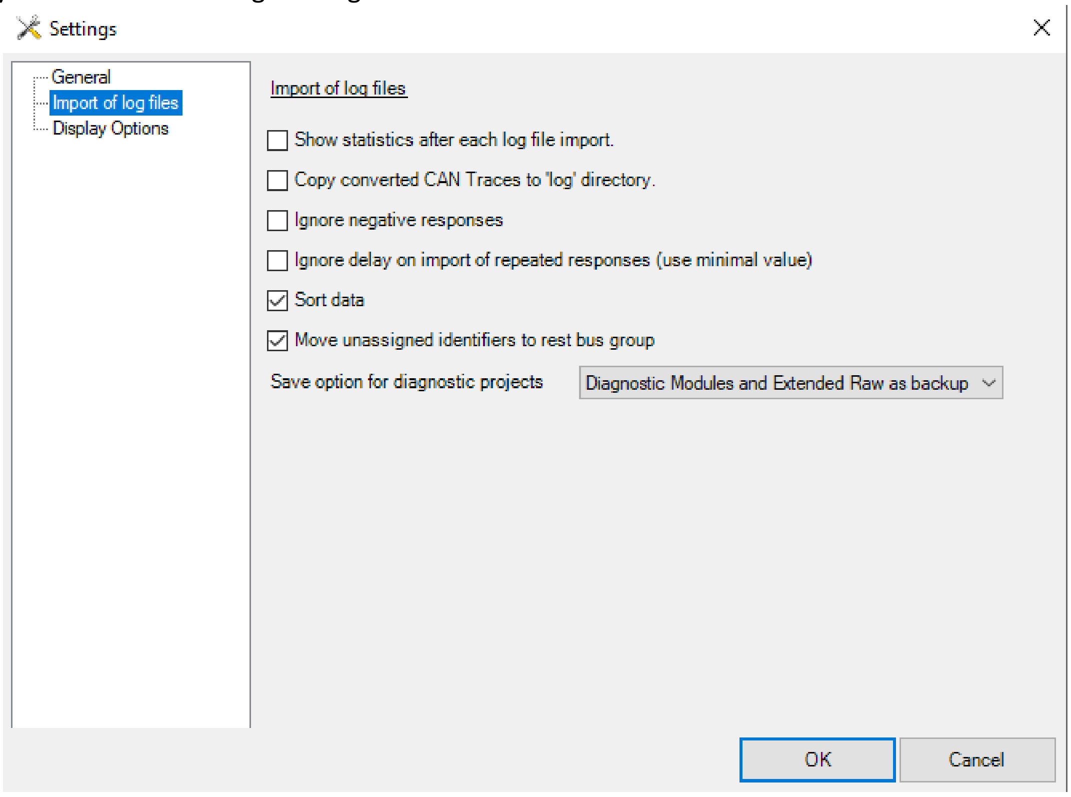
On the first page, you‘ll find the language selection. Currently, there are three languages supported by Data Manager of DiagRA® S: English, German and Chinese. After selecting a new language, the Settings dialog will automatically be updated. After confirming the settings, the rest of the application will update, too.



Language: English

c.2 Import of log files

Here you can set the settings for log files.



Import of log files

☐ Show statistics after each log file import.

☐ Copy converted CAN Traces to 'log' directory.

☐ Ignore negative responses

☐ Ignore delay on import of repeated responses (use minimal value)

☑ Sort data

☑ Move unassigned identifiers to rest bus group

Save option for diagnostic projects: Diagnostic Modules and Extended Raw as backup

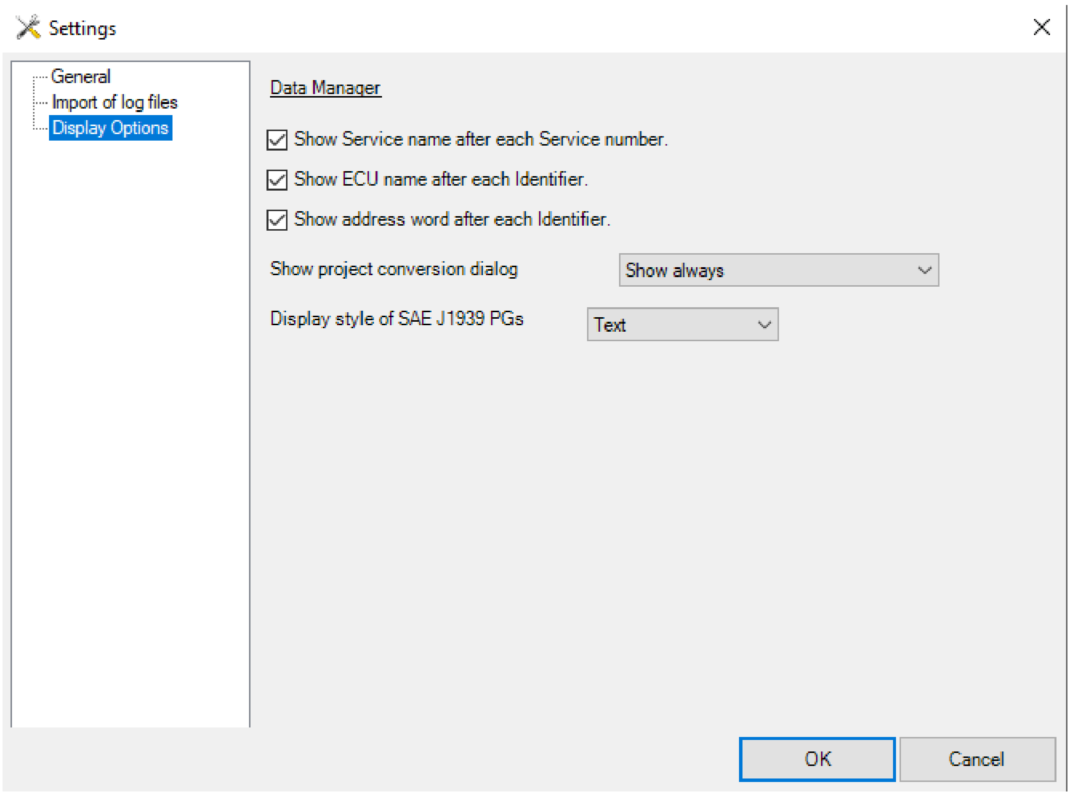
1) *Show statistics after each log file import*. When selecting the first option, a statistic dialog is shown after each log file import displaying a summary of the import, like the number of requests and responses stored in the imported file. *Default*: disabled.
2) *Copy converted CAN traces to project directory.* The temporary log file, which is created when you import a CAN Trace, is copied as a permanent file into the “log” directory, if the second option is enabled. *Default*: disabled.
3) *Ignore negative responses.* During log file import all negative responses will be ignored. *Default*: disabled.
4) *Ignore delay on import of repeated responses (use minimal value)*. During log file import the consequent responses with different value of delay will be handled as same responses. Delay value of imported response will be taken as minimal value of all delays for the same response data. *Default*: disabled.
5) *Sort data*. If enabled, data / request identifiers (for certain protocols) are sorted before storing in project files. *Default*: enabled.
6) *Move unassigned identifiers to rest bus group*. Affects behavior for import of diagnostic trace files (TRA / ASC / SR). In case this option is activated, all CAN identifiers not determined as request / response CAN ID will be assigned to rest bus group.
7) *Save option for diagnostic projects*. Defines how project files of type ‘Diagnostics’ are created: as diagnostic modules (UDS etc.), as ExtendedRaw or both at the same time (ExtendedRaw is stored in backup file). *Default*: Diagnostic modules and extended raw as backup.

DsprojCreator.exe

Starting from v2.7.0.42 there is a possibility to create project files using command line interface of DsprojCreator.exe. For information about its parameters run this executable without arguments.

c.3 Display Options

You can enable or disable additional display options of the Data Manager on this settings page.



☑ Show Service name after each Service number.
☑ Show ECU name after each Identifier.
☑ Show address word after each Identifier.
Show project conversion dialog: Show always
Display style of SAE J1939 PGs: Text

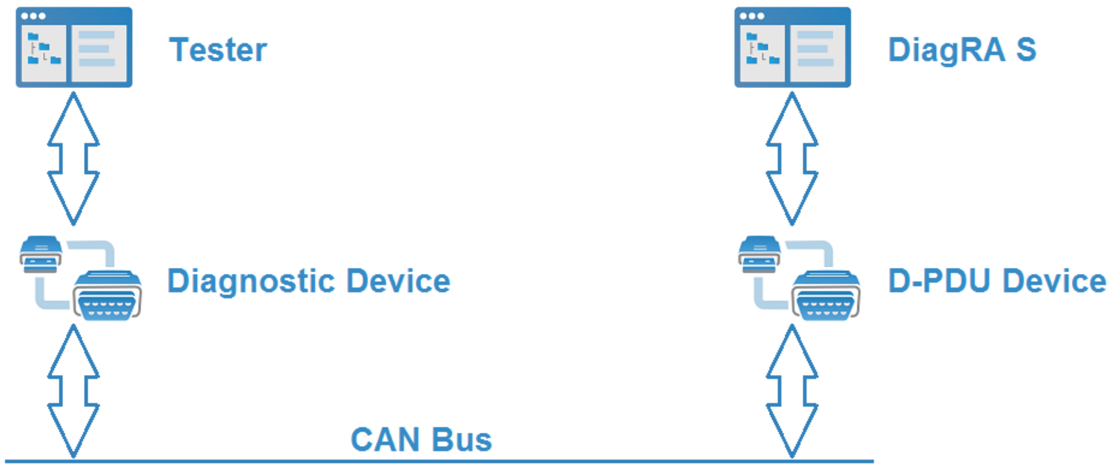
d. Supported features

You can't use all features of the Data Manager with all project types. The following tables show all features and their availabilities:

Function / Project	Diagnostics	SAE J1939	SAE J1699
Importing log files			
DiagRA® D Log files	✓	✓	✗
SAE J1699-3 Log files	✗	✗	✓
SAE J1939-84 Log files	✗	✓	✗
IXXAT CAN Trace	✓	✗	✗
Vector CAN Trace	✓	✓	✗
DiagRA® F log files	✓	✗	✗
DST-i simiulator files	✓	✗	✗
SimFrom simulator files	✓	✗	✗
ECU functions			
Log file informations	✓	✓	✓
Enable / Disable ECU	✓	✓	✓
Set CAN Identifier	✓	✓	✓
Remove ECU	✓	✓	✓
Other functions			
Create new ECU	✓	✓	✗

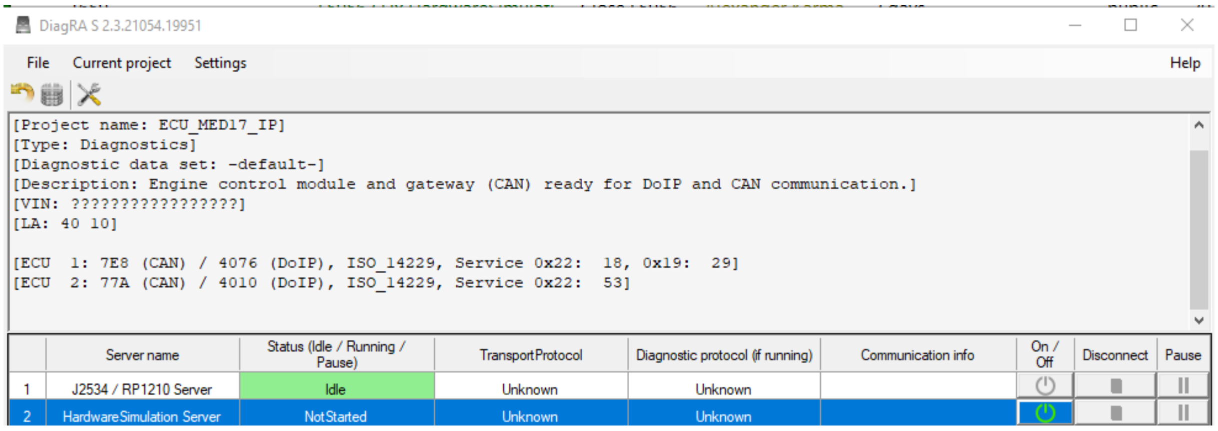
5 Hardware Simulation

You can use the software simulation as described in the past sections or you can use DiagRA® S for hardware simulation using a physical CAN bus.



DiagRA® S will monitor the physical CAN bus using a D-PDU device and all recognized Requests are processed sending their resulting Responses back to the CAN bus. By loading proper log files, DiagRA® S enables you to simulate a complete vehicle network.

To start the hardware simulation you need to click on the “On/Off” button of HardwareSimulation server:

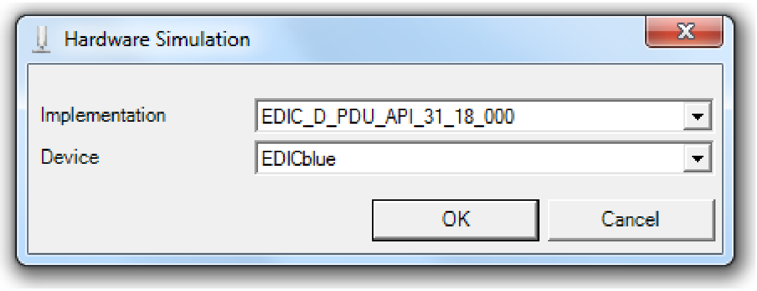


```
[Project name: ECU_MED17_IP]
[Type: Diagnostics]
[Diagnostic data set: -default-]
[Description: Engine control module and gateway (CAN) ready for DoIP and CAN communication.]
[VIN: ?????????????????]
[LA: 40 10]

[ECU  1: 7E8 (CAN) / 4076 (DoIP), ISO_14229, Service 0x22:  18, 0x19:  29]
[ECU  2: 77A (CAN) / 4010 (DoIP), ISO_14229, Service 0x22:  53]
```

	Server name	Status (Idle / Running / Pause)	TransportProtocol	Diagnostic protocol (if running)	Communication info	On / Off	Disconnect	Pause
1	J2534 / RP1210 Server	Idle	Unknown	Unknown		⏻	■	⏸
2	HardwareSimulation Server	NotStarted	Unknown	Unknown		⏻	■	⏸

The following window will open:



Implementation	EDIC_D_PDU_API_31_18_000
Device	EDICblue

Please select the device you want to use for DiagRA® S and commit your selection by clicking on “OK”. You’ll receive a message as soon as the initialization is completed. You can use your Tester application from this point to communicate with DiagRA® S using the physical CAN bus.

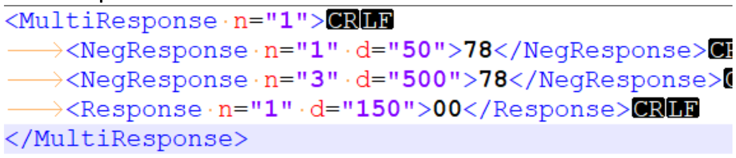
By clicking the “On/Off” button again, the communication will be stopped.

6 Response delay

In a default communication mode without response delay activated, DiagRA S communication core will answer as soon as possible in a synchronous way, i.e. no further request / command from diagnostic interface can be processed at the same time.

After activating the response delay, the communication core of DiagRA S will process requests in an asynchronous way over internal response queue, which considers the values of delay. Delay can be defined at two levels: at ECU level (as XML node “Delay” in DSPROJ files) and at response level (attribute “d” in DSPROJ files). If both values are present, the delay is calculated as a sum of both. The delay in the model describes the minimal delay for sending response after latest corresponding matching request. The only exception is MultiResponse: here the delay for first response is the delay regarding request, the delay values for other responses is relative to the previous response.

Consider following MultiResponse for UDS DataIdentifier “01 02”:



```
<MultiResponse n="1">CRLF
    <NegResponse n="1" d="50">78</NegResponse>C
    <NegResponse n="3" d="500">78</NegResponse>C
    <Response n="1" d="150">00</Response>CRLF
</MultiResponse>
```

The request “22 01 02” will result in several delayed responses.

1. 7F 22 78 after >= 50 ms
2. 3 responses 7F 22 78 after >= 500 ms each (i.e. relative to request: at 550 ms; at 1050 ms and at 1550 ms relative to request).
3. And as last the positive response 62 01 02 00 in about 150 ms after latest negative response, i.e. >=1700 ms after request.

Response delay can be edited at ECU level using slider. On the left side, you’ll find the ECU address (or CAN Identifier) and the current offset for each ECU on the right side. The minimum value is 0 which means that each ECU will respond as fast as possible. This is the default value for all projects and ECUs. The maximum value of each slider can be set in the program settings of Data Manager.

7 Running simulations from command line

Besides running simulation over DiagRA® S user interface it is possible to run simulation from command line.

7.1 Running DoIP simulator DoIPServer.exe

In the installation folder of DiagRA® S you can find DoIPServer.exe, which is a console application for running DoIP diagnostic simulation. As soon as DoIPServer.exe is started, it tries to listen for UDP port 13400 and accepts incoming TCP connections over the same port. However, since the port can be bound only to one application at the same time, you can't run DoIPServer.exe on the same machine in parallel with DiagRA S or other instances of DoIPServer.exe.

Starting of DoIPServer.exe requires only one argument to be passed:
--dsproj "<path-to-dsproj-file>"

There are no further command line arguments supported by DoIPServer.exe yet. This executable doesn't require an installation and can run on any computer. The only pre-requirement is the valid DiagRA S license installed on this computer and that dependencies of DoIPServer.exe can be resolved (all needed files are located in DiagRA S installation folder).
NOTE: DoIPServer.exe is currently compiled to be run in .NET Framework 4.6.1 and newer. However, there is a .NET Core 2.0 version of this executable obtainable, which can run also under Linux. In case of interest, please request it separately.

7.2 Running PassThru simulator PassThruServer.exe

In the installation folder of DiagRA® S you can find PassThruServer.exe, which is a console application for running simulation server part in case your diagnostic application under test ("diagnostic tester") uses PassThru.dll (J2534 / PassThru driver, either x86 or x64) for diagnostic communication. As soon as PassThruServer.exe is started, it waits for the connection from PassThru.dll. The connection will be established in case client application ("diagnostic tester") calls PassThruOpen function. It is usually happened automatically, since PassThru.dll iterates over all processes and searches for either PassThruServer.exe (first choice) or DiagRA_S.exe (second choice). In case multiple PassThruServer.exe instances are running simultaneously, the choice of PassThruServer.exe (simulator process) will be not deterministic. To overcome this limitation, it is possible to configure PassThru.dll to connect to the concrete process given by process id. In this case the file "PassThruConfig.ini" needs to be in the same folder as PassThru.dll. This INI file can have following content:

```
[Connection]
ServerProcessId=<process-id>
[Logging]
LogFilesFolder=<path-to-log-files-folder>
```

Starting of PassThruServer.exe requires only one argument to be passed:
--dsproj "<path-to-dsproj-file>"

Optionally, following arguments might be additionally provided:
--randomizeEcuOrder <true/false>
--bLogPassThruApiCallsAndLogPassThruComm <0,1,2,3>
"bLogPassThruApiCallsAndLogPassThruComm" defines logging level inside PassThru.dll: 3 – highest level (most logging output), 0 – lowest level (few logging output)

NOTE 1. PassThruServer.exe acts as a server part for PassThru.dll. Both make sense only for Windows. If you are interested to run DiagRA® S dsproj-file based ECU Simulation over SocketCAN on Linux machine, the DiagRA® S CanPlayer is obtainable. In case of interest, please request it separately.

8 Appendix: Diagnostic Devices

8.1 SAE J2534 (PassThru)

a. Supported Functions

PassThru Functions	Status
PassThruOpen	✓
PassThruClose	✓
PassThruConnect	✓
PassThruDisconnect	✓
PassThruReadMsgs	✓
PassThruWriteMsgs	✓
PassThruStartPeriodicMsg	✓
PassThruStopPeriodicMsg	✓
PassThruStartMsgFilter	✓
PassThruStopMsgFilter	✓
PassThruSetProgrammingVoltage	✓
PassThruReadVersion	✓
PassThruGetLastError	✓
PassThruIoctl	✓

b. Supported Configuration

PassThru IOCTLs	Status
GET_CONFIG	✓
SET_CONFIG	✓
READ_VBATT	✓
FIVE_BAUD_INIT	✓
FAST_INIT	✓
CLEAR_TX_BUFFER	✓
CLEAR_RX_BUFFER	✓
CLEAR_PERIODIC_MSGS	✗
CLEAR_MSG_FILTERS	✓
CLEAR_FUNCT_MSG_LOOKUP_TABLE	✗
ADD_TO_FUNCT_MSG_LOOKUP_TABLE	✗
DELETE_FROM_FUNCT_MSG_LOOKUP_TABLE	✗
READ_PROG_VOLTAGE	✗
GET_CONFIG	✓

PassThru SET_CONFIG / GET_CONFIG Parameters	Status
DATA_RATE	✓
LOOPBACK	✓
NODE_ADDRESS	✓
NETWORK_LINE	✗
P1_MIN	✓
P1_MAX	✗
P2_MIN	✓
P2_MAX	✓

P3_MIN	✗
P3_MAX	✓
P4_MIN	✗
P4_MAX	✓
W0	✗
W1	✗
W2	✗
W3	✗
W4	✗
W5	✗
TIDLE	✗
TINIL	✗
TWUP	✗
PARITY	✗
BIT_SAMPLE_POINT	✗
SYNC_JUMP_WIDTH	✗
T1_MAX	✗
T2_MAX	✗
T3_MAX	✗
T4_MAX	✗
T5_MAX	✗
ISO15765_BS	✗
ISO15765_STMIN	✗
BS_TX	✗
STMIN_TX	✗
DATA_BITS	✗
FIVE_BAUD_MOD	✗
ISO15765_WFT_MAX	✗

8.2 TMC RP1210 B

a. Supported Functions

RP1210 B Functions	Status
RP1210_ClientConnect	✓
RP1210_ClientDisconnect	✓
RP1210_SendMessage	✓
RP1210_ReadMessage	✓
RP1210_ReadVersion	✓
RP1210_ReadDetailedVersion	✓
RP1210_GetErrorMsg	✓
RP1210_GetLastErrorMsg	✓
RP1210_GetHardwareStatus	✗
RP1210_SendCommand	✓

b. Supported Configuration

RP1210 B Parameters	Status

RP1210_Reset_Device	✗
RP1210_Set_All_Filters_States_to_Pass	✗
RP1210_Set_Message_Filtering_For_J1939	✓
RP1210_Set_Message_Filtering_For_CAN	✓
RP1210_Set_Message_Filtering_For_J1708	✗
RP1210_Set_Message_Filtering_For_J1850	✗
RP1210_Set_Message_Filtering_For_ISO15765	✗
RP1210_Generic_Driver_Command	✗
RP1210_Set_J1708_Mode	✗
RP1210_Echo_Transmitted_Messages	✓
RP1210_Set_All_Filters_States_to_Discard	✓
RP1210_Set_Message_Receive	✗
RP1210_Protect_J1939_Address	✗
RP1210_Set_Broadcast_For_J1708	✗
RP1210_Set_Broadcast_For_CAN	✗
RP1210_Set_Broadcast_For_J1939	✗
RP1210_Set_Broadcast_For_J1850	✗
RP1210_Set_J1708_Filter_Type	✗
RP1210_Set_J1939_Filter_Type	✗
RP1210_Set_CAN_Filter_Type	✗
RP1210_Set_J1939_Interpacket_Time	✗
RP1210_SetMaxErrorMsgSize	✗
RP1210_Disallow_Further_Connections	✗
RP1210_Set_J1850_Filter_Type	✗
RP1210_Release_J1939_Address	✗
RP1210_Set_ISO15765_Filter_Type	✗
RP1210_Set_Broadcast_For_ISO15765	✗
RP1210_Set_ISO15765_Flow_Control	✗
RP1210_Clear_ISO15765_Flow_Control	✗
RP1210_Set_ISO15765_Link_Type	✗
RP1210_Set_J1939_Baud	✗
RP1210_Set_BlockTimeout	✗
RP1210_Set_J1708_Baud	✗