

# Unified Diagnostic Services

**Unified Diagnostic Services (UDS)** is a diagnostic [communication protocol](#) used in [electronic control units](#) (ECUs) within automotive electronics, which is specified in the ISO 14229-1.<sup>[1]</sup> It is derived from ISO 14230-3 ([KWP2000](#)) and the now obsolete [ISO 15765-3](#) (Diagnostic Communication over [Controller Area Network](#) (DoCAN)<sup>[2]</sup>). 'Unified' in this context means that it is an international and not a company-specific standard. By now this communication protocol is used in all new ECUs made by Tier 1 suppliers of [Original Equipment Manufacturer](#) (OEM), and is incorporated into other standards, such as [AUTOSAR](#). The ECUs in modern vehicles control nearly all functions, including [electronic fuel injection](#) (EFI), [engine control](#), the transmission, anti-lock braking system, door locks, braking, window operation, and more.

Diagnostic tools are able to contact all ECUs installed in a vehicle which has UDS services enabled. In contrast to the [CAN bus](#) protocol, which only uses the first and second layers of the [OSI model](#), UDS utilizes the fifth and seventh layers of the OSI model. The Service ID (SID) and the parameters associated with the services are contained in the payload of a message frame.

Modern vehicles have a diagnostic interface for off-board diagnostics, which makes it possible to connect a computer (client) or diagnostics tool, which is referred to as tester, to the communication system of the vehicle. Thus, UDS requests can be sent to the controllers which must provide a response (this may be positive or negative). This makes it possible to [interrogate](#) the fault memory of the individual control units, to update them with new firmware, have low-level interaction with their hardware (e.g. to turn a specific output on or off), or to make use of special functions (referred to as routines) to attempt to understand the environment and operating conditions of an ECU to be able to [diagnose](#) faulty or otherwise undesirable behavior.

## Services

---

SID (Service Identifier)

Function group	Request SID	Response SID	Service	Description
Diagnostic and Communications Management	0x10	0x50	Diagnostic Session Control	<p>UDS offers access levels called "sessions". Different sessions usually offer different levels of access to services and/or sub-functions. During normal use (on start), the session should be <b>0x01 Default Session</b>.</p> <p>This "Diagnostic Session Control" service allows the user to switch between available sessions specific to the ECU. Some sessions may not have been implemented. Others may have access control measures, requiring authorization through services like <b>0x27 Security Access</b> or <b>0x29 Authentication</b>. Additionally, sessions may require certain operating conditions to be met, such as manufacturer specific "pre-production" or "boot modes", which are separate from UDS.</p> <p>The following sessions are standard, but vehicle manufacturer and supplier-specific sessions are often implemented in conjunction:</p> <ul style="list-style-type: none"> <li>• <b>0x01 Default Session</b> which typically has the lowest level of access to services and/or sub-functions. This will usually allow <b>0x14 Clear Diagnostic Information</b>, certain <b>0x22 Read Data By Identifier</b> "Data Identifiers", and more.</li> <li>• <b>0x02 Programming Session</b> which gives access to "Upload / Download" and "Data Transmission" services, and sometimes advanced access to other services and/or sub-functions like .</li> <li>• <b>0x03 Extended Diagnostic Session</b> which typically offers further "Diagnostic and Communications Management", "Data Transmission", "Input / Output Control", and "Remote Activation of Routine" services.</li> <li>• <b>0x04 Safety System Diagnostic Session</b> offers largely the same as 0x03 Extended Diagnostic</li> </ul>

Function group	Request SID	Response SID	Service	Description
				Session, but does not always offer sensitive or unrelated services and/or sub-functions. It may offer further routines and data transmission sub-functions.
Diagnostic and Communications Management	0x11	0x51	ECU Reset	<p>The service "ECU reset" is used to restart the control unit (ECU). Depending on the control unit hardware and implementation, different forms of reset can be used:</p> <ul style="list-style-type: none"> <li>• <b>0x11 01 Hard Reset</b> simulates a shutdown of the power supply.</li> <li>• <b>0x11 02 key Off-On Reset</b> simulates the drain and turn on the ignition with the key.</li> <li>• <b>0x11 03 Soft Reset</b> allows the initialization of certain program units and their storage structures.</li> </ul> <p>Again, there are reserved values that can be defined for vehicle manufacturers and vehicle suppliers specific use.</p>
Diagnostic and Communications Management	0x14	0x54	Clear Diagnostic Information	<p>Clears diagnostic trouble codes (DTC's) from ECU memory.</p> <ul style="list-style-type: none"> <li>• <b>0x14 FF FF 33</b> clears emissions-related DTC's</li> <li>• <b>0x14 FF FF FF</b> clears all codes DTC's</li> </ul> <p>Like other sub-functions in this list, there are manufacturer-specific reserved values that offer unique and/or specific functionality.</p>
Diagnostic and Communications Management	0x27	0x67	Security Access	<p>Security check is available to enable the most security-critical services. For this purpose a "Seed" is generated and sent to the client by the control unit. From this "Seed" the client has to compute a "Key" and send it back to the control unit to unlock the security-critical services.</p>
Diagnostic and Communications Management	0x28	0x68	Communication Control	<p>With this service, both the sending and receiving of messages can be turned off in the control unit.</p>
Diagnostic and Communications Management	0x29	0x69	Authentication	<p>An update (2020) of the standard added this service to provide a standardized approach to more modern methods of authentication than are permitted by the</p>

Function group	Request SID	Response SID	Service	Description
				Security Access (0x27) service, including bidirectional authentication with PKI-based Certificate Exchange.
Diagnostic and Communications Management	0x3E	0x7E	Tester Present	If no communication is exchanged with the client for a long time, the control unit automatically exits the current session and returns to the "Default Session" back, and might go to sleep mode. Therefore, there is an extra service which purpose is to signal to the device that the client is still present.
Diagnostic and Communications Management	0x83	0xC3	Access Timing Parameters	In the communication between the controllers and the client, certain times must be observed. If these are exceeded, without a message being sent, it must be assumed that the connection was interrupted. These times can be called up and changed.
Diagnostic and Communications Management	0x84	0xC4	Secured Data Transmission	
Diagnostic and Communications Management	0x85	0xC5	Control DTC Settings	<p>Enables, disables, or otherwise controls the transmission of DTC's. This is helpful when maintenance or modification work is being performed on a vehicle to avoid anomalous or annoying behavior, like beeping on an instrument cluster or safety systems activating in response to an external DTC. Like most UDS settings, this does not typically persist after a reboot.</p> <ul style="list-style-type: none"> <li>• <b>0x01 On</b> enables the transmission of DTC's.</li> <li>• <b>0x02 Off</b> disabled the transmission of DTC's.</li> </ul>
Diagnostic and Communications Management	0x86	0xC6	Response On Event	
Diagnostic and Communications Management	0x87	0xC7	Link Control	The Service Link Control is used to set the baud rate of the diagnostic access. It is usually implemented only at the central gateway.
Data Transmission	0x22	0x62	Read Data By Identifier	With this service, it is possible to retrieve one or more values of a control unit. This can be information of all kinds and of different lengths such as Partnumber or the software version. Dynamic values such as the current state of the sensor can be queried. Each value is associated to a Data Identifier (DID) between

Function group	Request SID	Response SID	Service	Description
				0 and 65535; for example, the <a href="#">VIN</a> DID is 61840d (0xF190). Normal CAN signals are meant for information that some ECU uses in its functionality. DID data is sent on request only, and is for information that no ECU uses, but a service tool or a software tester can benefit from.
Data Transmission	0x23	0x63	Read Memory By Address	Read data from the physical memory at the provided address. This function can be used by a testing tool, in order to read the internal behaviour of the software.
Data Transmission	0x24	0x64	Read Scaling Data By Identifier	
Data Transmission	0x2A	0x6A	Read Data By Identifier Periodic	With this service, values are sent periodically by a control unit. The values to be sent must be defined to only using the "Dynamically Define Data Identifier".
Data Transmission	0x2C	0x6C	Dynamically Define Data Identifier	<p>This service offers the possibility of a fix for a device specified Data Identifier (DID) pool to configure another Data Identifier. This is usually a combination of parts of different DIDs or simply a concatenation of complete DIDs.</p> <p>The requested data may be configured or grouped in the following manner:</p> <ul style="list-style-type: none"> <li>• Source DID, position, length (in bytes), Sub-Function Byte: defineByIdentifier</li> <li>• Memory address length (in bytes), Sub-Function Byte: defineByMemoryAddress</li> <li>• Combinations of the two above methods through multiple requests.</li> </ul>
Data Transmission	0x2E	0x6E	Write Data By Identifier	With the same Data Identifier (DID), values can also be changed. In addition to the identifier, the new value is sent along.
Data Transmission	0x3D	0x7D	Write Memory By Address	The "Write Memory By Address" service allows the external diagnostic tool to write information into the ECU at one or more contiguous memory locations.
Stored Data Transmission	0x14	0x54	Clear Diagnostic Information	Delete all stored DTC

Function group	Request SID	Response SID	Service	Description
Stored Data Transmission	0x19	0x59	Read DTC Information	DTC stands for " <a href="#">Diagnostic Trouble Codes</a> ". Each DTC handled by the control unit fault is stored with its own code in the error memory and can be read at any time. In addition to the error, additional information will be stored, which can also be read.
Input / Output Control	0x2F	0x6F	Input Output Control By Identifier	<p>This service allows an external system intervention on internal / external signals via the diagnostic interface.</p> <p>By specifying a so-called option bytes additional conditions for a request can be specified, the following values are specified:</p> <p><i>ReturnControlToECU</i>: The device must get back controls of the mentioned signals.</p> <p><i>ResetToDefault</i>: The tester prompts to reset signals to the system wide default value.</p> <p><i>Freeze Current State</i>: The device shall freeze the current signal value.</p> <p><i>ShortTermAdjustment</i>: The device shall use the provided value for the signal</p>
Remote Activation of Routine	0x31	0x71	Routine Control	<p>Control routine services of all kinds can be performed. There are three different message types:</p> <ul style="list-style-type: none"> <li>• With the start-message, a service can be initiated. It can be defined to confirm the beginning of the execution or to notify when the service is completed.</li> <li>• With the Stop message, a running service can be interrupted at any time.</li> <li>• The third option is a message to query the results of the service.</li> </ul> <p>The start and stop message parameters can be specified. This makes it possible to implement every possible project-specific service.</p>

Function group	Request SID	Response SID	Service	Description
Upload / Download	0x34	0x74	Request Download	Downloading new software or other data into the control unit is introduced using the "Request Download". Here, the location and size of the data is specified. In turn, the tester specifies how large the data packets can be.
Upload / Download	0x35	0x75	Request Upload	The service "request upload" is almost identical to the service "Request Download". With this service, the software from the control unit is transferred to the tester. The location and size must be specified. Again, the size of the data blocks are specified by the tester.
Upload / Download	0x36	0x76	Transfer Data	For the actual transmission of data, the service "Transfer Data" is used. This service is used for both uploading and downloading data. The transfer direction is notified in advance by the service "Request Download" or "Upload Request". This service should try to send packets at maximum length, as specified in previous services. If the data set is larger than the maximum, the "Transfer Data" service must be used several times in succession until all data has arrived.
Upload / Download	0x37	0x77	Request Transfer Exit	A data transmission can be 'completed' when using the "Transfer Exit" service. This service is used for comparison between the control unit and the tester. When it is running, a control unit can answer negatively on this request to stop a data transfer request. This will be used when the amount of data (set in "Request Download" or "Upload Request") has not been transferred.
Upload / Download	0x38	0x78	Request File Transfer	This service is used to initiate a file download from the client to the server or upload from the server to the client. Additionally information about the file system are available by this service.
		0x7F	Negative Response	This response is given when a service request could not be performed, for example having a not supported Data Identifier. A Negative Response Code will be included.

# Negative response codes

---

Negative response from ECU contains SID 0x7F and two payload bytes: request's SID and error code. These codes can be found in freely available software (for example, [BusMaster \(https://github.com/rbei-etas/busmaster/blob/de29ec1514558b3b7e0504e07d38ce0d65616d0f/Sources/BUSMASTER/UDS\\_Protocol/UDS\\_NegRespMng.cpp#L25\)](https://github.com/rbei-etas/busmaster/blob/de29ec1514558b3b7e0504e07d38ce0d65616d0f/Sources/BUSMASTER/UDS_Protocol/UDS_NegRespMng.cpp#L25) ) as well as in the ISO itself.



<b>NRC</b>	<b>Description</b>
0x10	General reject
0x11	Service not supported
0x12	Subfunction not supported
0x13	Incorrect message length or invalid format
0x14	Response too long
0x21	Busy, repeat request
0x22	Conditions not correct
0x24	Request sequence error
0x25	No response from subnet component
0x26	Failure prevents execution of requested action
0x31	Request out of range
0x33	Security access denied
0x34	Authentication failed
0x35	Invalid key
0x36	Exceeded number of attempts
0x37	Required time delay not expired
0x38	Secure data transmission required
0x39	Secure data transmission not allowed
0x3A	Secure data verification failed
0x50	Certificate validation failed, invalid time period
0x51	Certificate validation failed, invalid signature
0x52	Certificate validation failed, invalid chain of trust
0x53	Certificate validation failed, invalid type
0x54	Certificate validation failed, invalid format
0x55	Certificate validation failed, invalid content
0x56	Certificate validation failed, invalid scope
0x57	Certificate validation failed, invalid certificate
0x58	Ownership verification failed
0x59	Challenge calculation failed
0x5A	Setting access right failed
0x5B	Session key creation/derivation failed
0x5C	Configuration data usage failed
0x5D	Deauthentication failed

0x70	Upload download not accepted
0x71	Transfer data suspended
0x72	General programming failure
0x73	Wrong block sequence number
0x78	Request correctly received, response pending
0x7E	Subfunction not supported in active session
0x7F	Service not supported in active session
0x81	RPM too high
0x82	RPM too low
0x83	Engine is running
0x84	Engine is not running
0x85	Engine run time too low
0x86	Temperature too high
0x87	Temperature too low
0x88	Vehicle speed too high
0x89	Vehicle speed too low
0x8A	Throttle/pedal too high
0x8B	Throttle/pedal too low
0x8C	Transmission range not in neutral
0x8D	Transmission range not in gear
0x8F	Brake switch not closed
0x90	Shifter lever not in park
0x91	Torque converter clutch locked
0x92	Voltage too high
0x93	Voltage too low
0x94	Resource temporary unavailable

## See also

---

- [On-board diagnostics](#), general article about diagnostic services in vehicles
- [OBD-II PIDs](#), about the US standard

# References

---

1. "Iso 14229-1:2020 Unified diagnostic services (UDS) Part 1: Application layer" (<https://www.iso.org/standard/72439.html>) .
2. "Iso 15765-3:2004 Diagnostics on Controller Area Networks (CAN) Part 3: Implementation of unified diagnostic services (UDS on CAN)" (<https://www.iso.org/standard/33618.html>) .

# External links

---

- Unified Diagnostic Services - ISO 14229 ([https://automotive.softing.com/fileadmin/sof-files/pdf/de/ae/poster/UDS\\_Faltposter\\_softing2016.pdf](https://automotive.softing.com/fileadmin/sof-files/pdf/de/ae/poster/UDS_Faltposter_softing2016.pdf)) (poster by softing.com)
- PCAN-UDS 2.x API description ([https://www.peak-system.com/produktcd/Pdf/English/PCAN-UDS-API\\_UserMan\\_eng.pdf](https://www.peak-system.com/produktcd/Pdf/English/PCAN-UDS-API_UserMan_eng.pdf))