



Software manual  
ifm CAN Device Tool

**GB**



# Contents

1	Change history . . . . .	3
2	Symbols used . . . . .	4
3	Safety instructions . . . . .	5
4	Introduction . . . . .	6
4.1	Use cases . . . . .	6
4.2	Common system configurations . . . . .	6
5	Installation . . . . .	7
5.1	New installation . . . . .	7
5.2	Update . . . . .	8
6	User interface . . . . .	9
7	"Device Identification" use case . . . . .	10
7.1	"Scan" function . . . . .	10
8	"Safety Configuration" use case . . . . .	12
8.1	Editing parameter data sets . . . . .	12
8.1.1	"Generate" function . . . . .	12
8.1.2	"Copy" function . . . . .	13
8.2	Saving a parameter data set . . . . .	13
8.3	Opening a parameter set . . . . .	14
9	"Download" use case . . . . .	16
9.1	"Download" function . . . . .	16
9.2	"Upload" function . . . . .	17
9.3	"Display" function . . . . .	17
9.4	"Verify" function . . . . .	18
9.5	"Lock" function . . . . .	19
9.6	"Unlock" function . . . . .	20
9.7	"Reset" function . . . . .	20
10	"Maintenance" use case . . . . .	22
10.1	"Read" function . . . . .	22
11	Language selection . . . . .	23
12	Troubleshooting . . . . .	24
12.1	Log files . . . . .	24
12.2	"About" dialogue . . . . .	24

# 1 Change history

Date	Version	Change
14/03/2024	00	First publication
06/01/2025	01	Formatting changed

## 2 Symbols used

- ✓ Requirement
- Instructions
- ▷ Reaction, result
- [...] Designation of keys, buttons or indications
- Cross-reference
-  Important note  
Non-compliance may result in malfunction or interference.
-  Information  
Supplementary note

### **3 Safety instructions**

Please read the software manual before using the software.

Ensure that the software is suitable for your application and the connected sensors without any restrictions.

If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property can occur. That is why installation, set-up and maintenance of the article must only be carried out by qualified personnel authorised by the machine operator.

Non-observance of the instructions, operation which is not in accordance with use as prescribed below, wrong installation or incorrect handling can affect the safety of operators and machinery.

The installation and connection must comply with the applicable national and international standards. Responsibility lies with the person installing the software.

Changes to the source code or to individual components of the software which are not explicitly described in the instructions lead to the loss of the right to benefit from the support provided by ifm electronic gmbh.

## 4 Introduction

The ifm CAN Device TOOL (ifmCDT) is a manufacturer-specific parameter setting and display tool that implements the requirements for an external configuration tool in accordance with EN 50325-5 (CANopen Safety). The ifmCDT is a standalone Windows application that establishes a connection to devices with CANopen interfaces via a CAN interface adapter.

The ifmCDT was developed for the encoders (SR) of the RM9xxS series from ifm electronic gmbh. The ifmCDT was not programmed in a safety-related way. Rather, safety is achieved when configuring the RM9xxS by following the steps for downloading the configuration into the node and the subsequent verification measures ([“Download” use case \(→ 16\)](#)).

### 4.1 Use cases

The ifmCDT fulfils the following tasks:

- Generate the safety signatures via the configuration of the RM9xxS
- Download the safety configuration
- Display the device configuration for verification of the safety configuration
- Display device diagnostics and operating data



The ifmCDT can only be used for devices with a CANopen safety communication profile.

### 4.2 Common system configurations

The ifmCDT program was tested on the following system configurations:

Microsoft Windows 10 Pro, 32bit / 64bit

Microsoft Windows 10 Home, 32bit / 64bit

Microsoft Windows 11 Pro, 32bit / 64bit

Microsoft Windows 11 Home, 32bit / 64bit

The system requires an installed .NET-Framework v4.8 or higher.

An external CAN interface adapter is required to use the communication interface to the device.

The following interface adapters have been integrated so far:

- PEAK PCAN-USB
- Sontheim CANfox
- CANfox (e.g.: EC2112)



Additional interface adapters can be integrated if required.

## 5 Installation

### 5.1 New installation

The installation package of the ifmCDT application is a MSI-Windows-Installerpackage intended to minimise the influence and requirements for installation on a specific target system (Windows platform).

The installation package contains all the necessary files so that the installation can also be carried out locally from a CD-ROM or USB stick. A 32 bit setup version is currently available for installation.

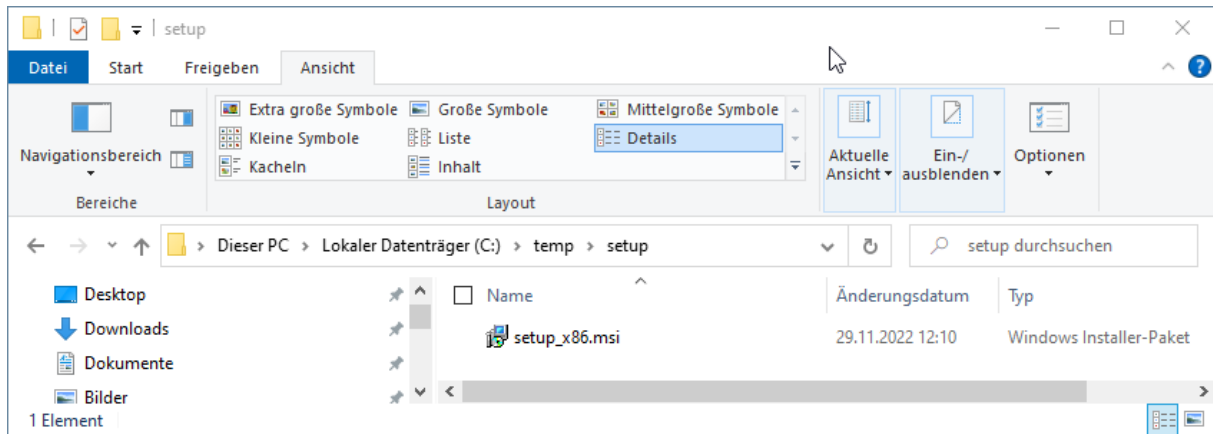


Fig. 1: Installation package of the ifmCDT application

The installation starts after a double-click on the installer package setup\_x86.msi. The installation is carried out automatically and controlled by a setup wizard. The files and programs contained in the installation package were signed by ifm electronic gmbh before publication so that the user can verify the origin and authenticity of the contents.

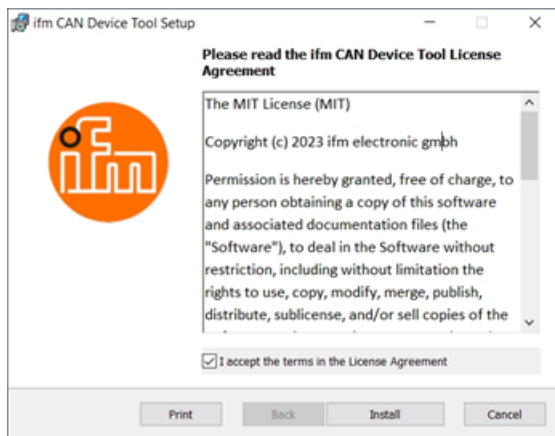


Fig. 2: Setup wizard - licence agreement



The installation computer can only verify ifm electronic as publisher if there is an internet connection.

During installation, the setup requires administrator rights to change settings. All application files should then be installed in the program directory of the Windows computer.

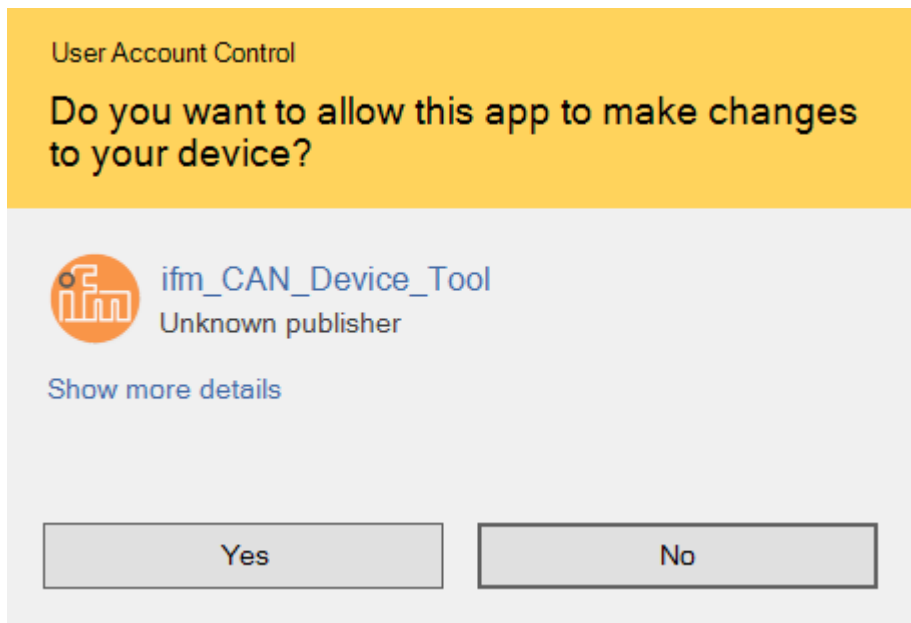


Fig. 3: User account control

When the installation is finished, the Setup-Wizard displays the end of the process. During the installation process, a shortcut icon is installed on the desktop of the target system.

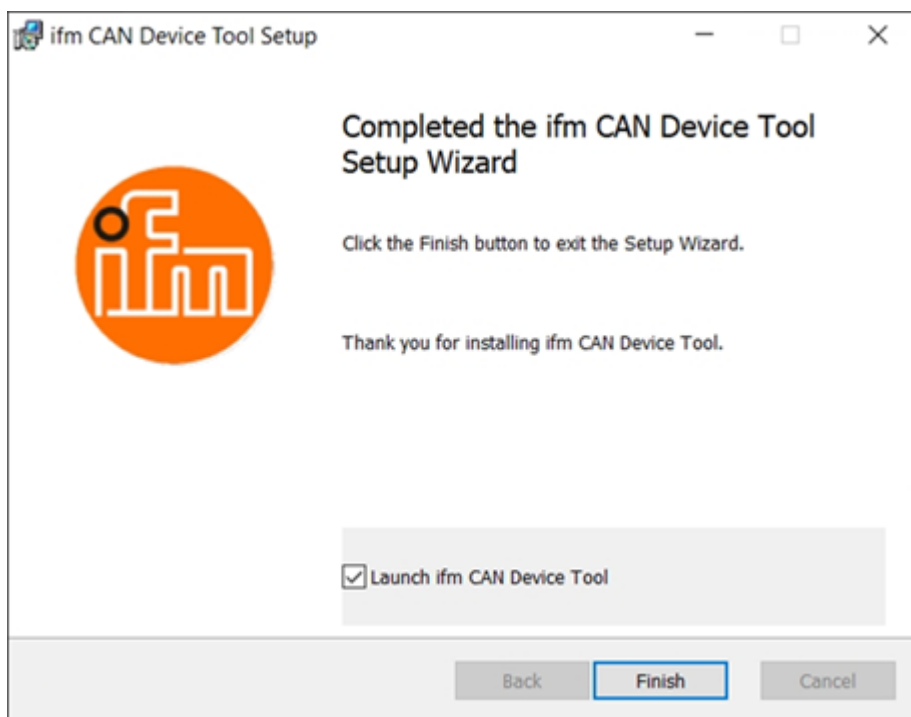


Fig. 4: Setup-Wizard – final screen.

## 5.2 Update

The application can be updated simply by running an updated setup package again on the target system. The setup installer updates the application automatically.



## 6 User interface

The user interface of the ifmCDT application is a dialogue-based Windowsinterface whose sub-functions have been divided into tabs depending on the available use cases (see Figure 5). Depending on the operating mode and the capabilities of the configured device series, more or fewer tabs are available.

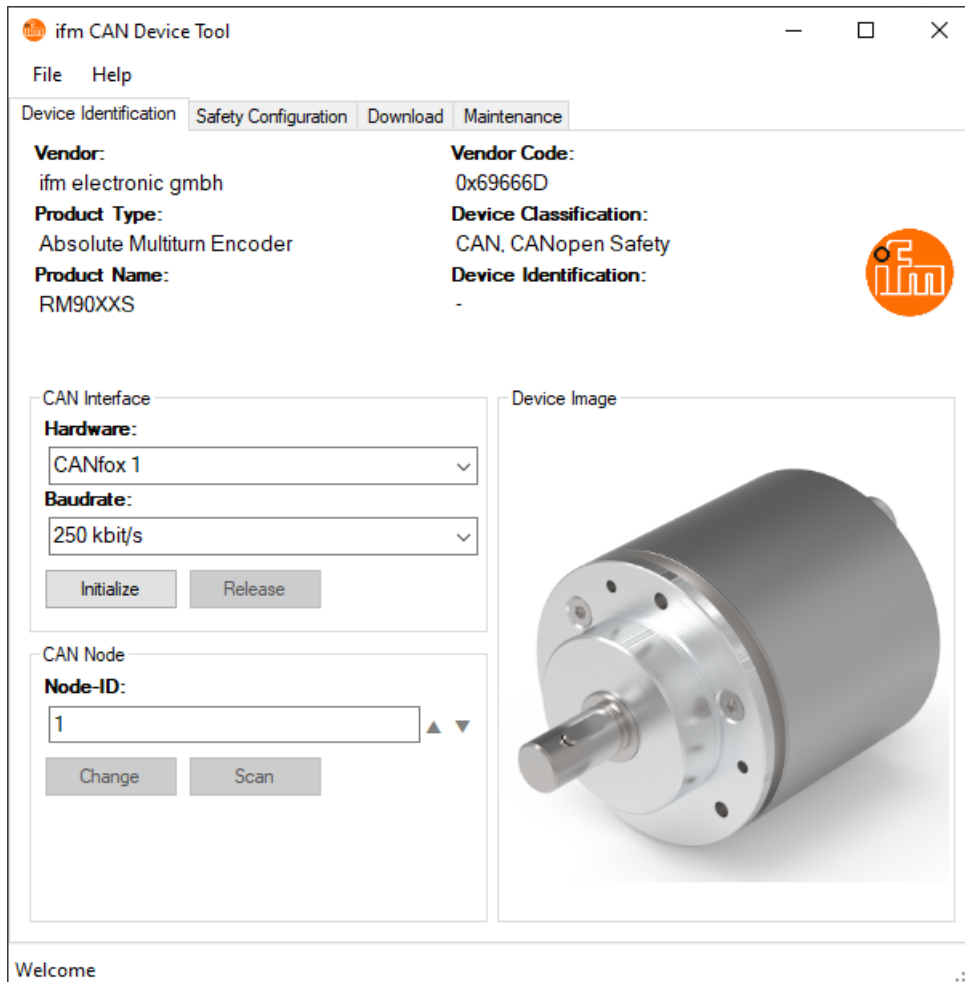


Fig. 5: User interface of the ifmCDT application

At the top of the dialogue, a menu bar provides basic functions for application control depending on the selected tab (use case).

## 7 "Device Identification" use case

The device identification use case is used to display technical information about the device and to configure the CAN interface.

Manufacturer information, e.g. the manufacturer ID, is displayed in the upper half of the dialogue.

The connection settings for the CAN interface, in particular the node address, are set and programmed in the lower half of the dialogue.

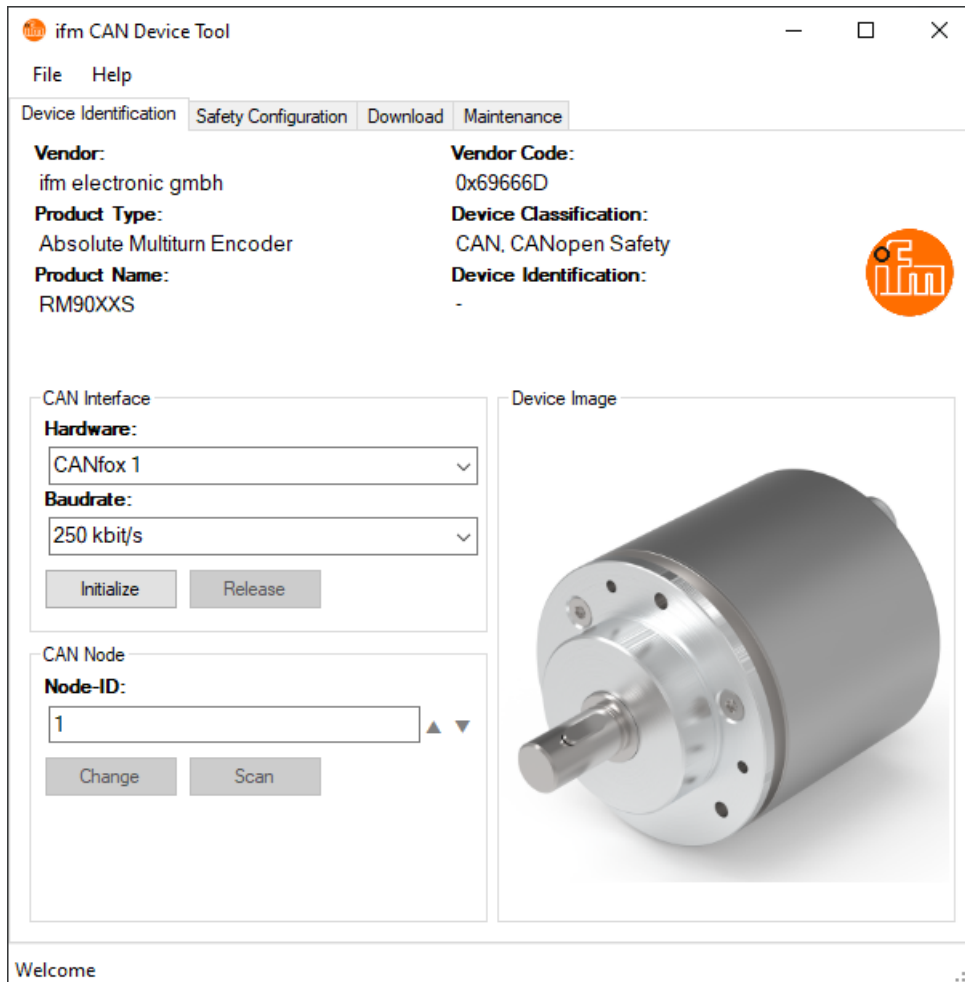


Fig. 6: Application: Device ID

The user selects the CAN interface adapter that has been connected to the Windows computer and recognised. This connection can later be used to communicate with the device.



The Windows computer requires a CAN interface adapter to be connected to the CAN connection of the RM9xxS so that data can be exchanged with the device.

The user sets the baud rate and the node address of the RM9xxS with which data is to be exchanged.



The node address of the device can be set via LSS; cf. the pin connection of the device.

The user initialises or enables the CAN interface with which the RM9xxS is to be operated.

### 7.1 "Scan" function

The scan function reads the identity objects from the set device. The device indicates receipt of the enquiry with a corresponding flashing pattern. This allows the user to verify that the correct device has been selected.

At the same time, the ifmCDT reads several device identifications, for example, the serial number. After the readout process, the display fields for the manufacturer information in the upper part of the dialogue contain these identifications that can be compared with the type plate of the device.

## 8 “Safety Configuration” use case

The safety configuration use case enables the user to calculate the safety signatures via the SRDO mappings and the application-specific parameter data sets of the device. The calculations are carried out in accordance with the EN 50325-5 calculation rule.

### 8.1 Editing parameter data sets

When the ifmCDT is started, the application loads the default parameter data sets of the device. All parameters can be edited in the table depending on the application. The help for the features shows the permitted value ranges; settings may be restricted by selection options.

There are settings for the system functions of the device, in particular for position data and speed detection.

There are tabs for setting the SRDO communication and for setting the application.

ifm CAN Device Tool

File Help

Device Identification Safety Configuration Download Maintenance

SRDO1-13FF/01 SRDO2-13FF/02 Safety Position-61FF/01 Safety Speed-61FF/02

Feature	Value
<b>SRDO communication parameter</b>	Object 0x1301
Information Direction	tx, SRDO producer
Refresh Time / SCT	25
SRVT	20
COB-ID 1	0x101
COB-ID 2	0x102
<b>SRDO mapping parameter</b>	Object 0x1381
Highest sub-index supported	Sub-indices from 01h to 08h valid (enabled)
Sub-index	1
SR ADO 1 (plain data)	0x61200108
Sub-index	2
SR ADO 1 (bitwise inverted data)	0x61210108
Sub-index	2

**Tool Information:**  
Setting concerning the SRDO's transmit/receive (tx/rx) direction

**Parameter Set Description:**  
SRDO1 Mapping Safety Configuration Signature (Index 0x13FF/Subindex 01)

**Konfiguration:**  
ifm\_cia

**Signatur der Safety-Konfiguration:**

HEX 0x250D Generate

Fig. 7: Application: safety configuration

#### 8.1.1 “Generate” function

The [Generate] function generates the signatures using the selected parameter settings in the selected format. The results are displayed in the corresponding display field.



When you change the tab, the signature display changes accordingly.

### 8.1.2 “Copy” function

In the context menu of the display element for the signature of the safety configuration, the user can copy the calculated signature for the safety configuration to the clipboard and paste it into an engineering tool, e.g. CODESYS.

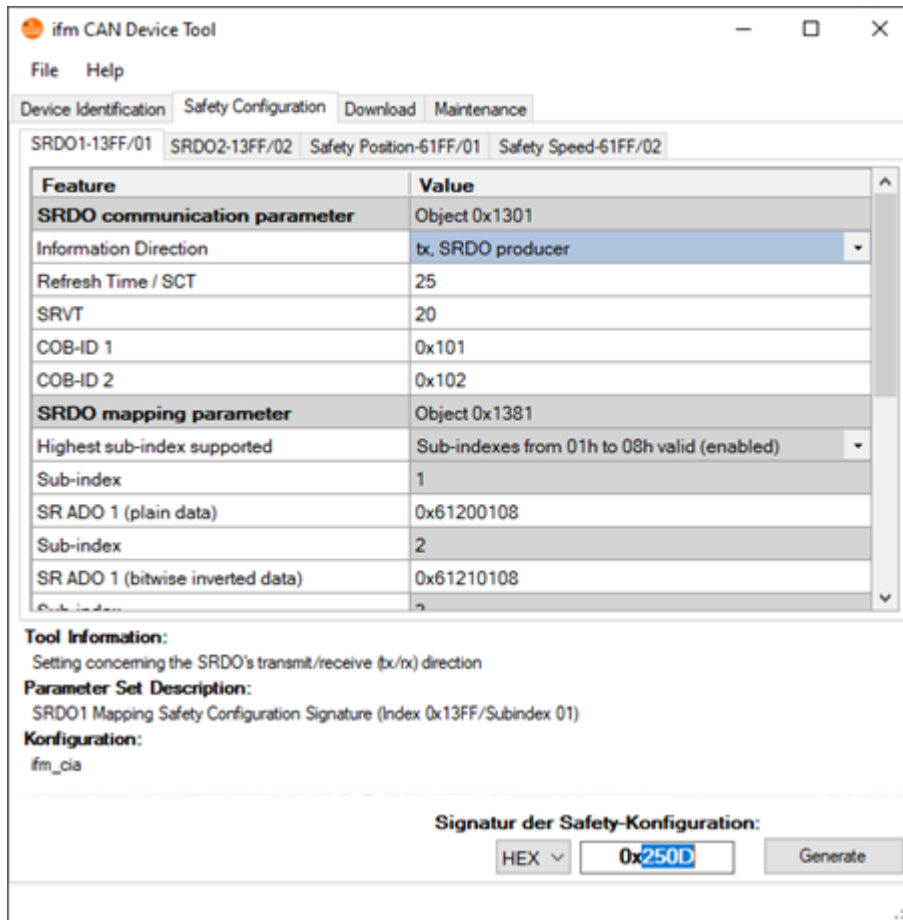


Fig. 8: “Copy” context menu

## 8.2 Saving a parameter data set

The user can save a parameter set with user-specific settings for the given structure in a PRSX file in order to open it again at a later time if necessary ([Opening a parameter set \(→ 14\)](#)). To do this, the user calls up the menu item [File] > [Save]. The ifmCDT application saves the current setting of all parameter tables in a file after the user has specified the name and storage location of the file.

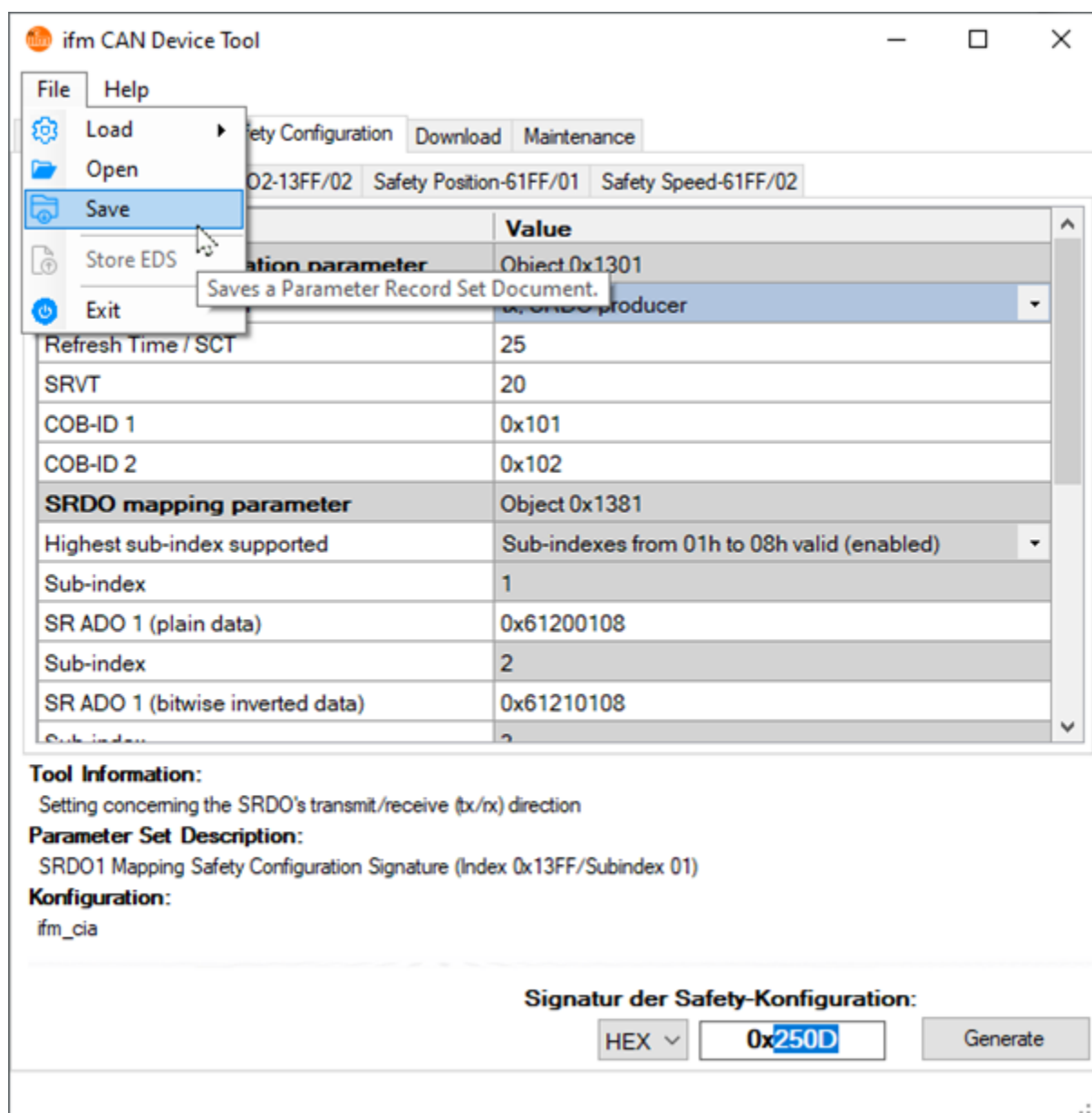


Fig. 9: Dialogue window with loaded parameter set and file-IO dialogue



The save function is only available if a parameter data set profile is available.

### 8.3 Opening a parameter set

The user can open parameter sets saved in the form of PRSX files if existing settings are to be reused. To do this, call up the menu item [File] > [Open]. The ifmCDT application opens the settings of a PRSX file after the user has selected the corresponding file in a file selection dialogue.

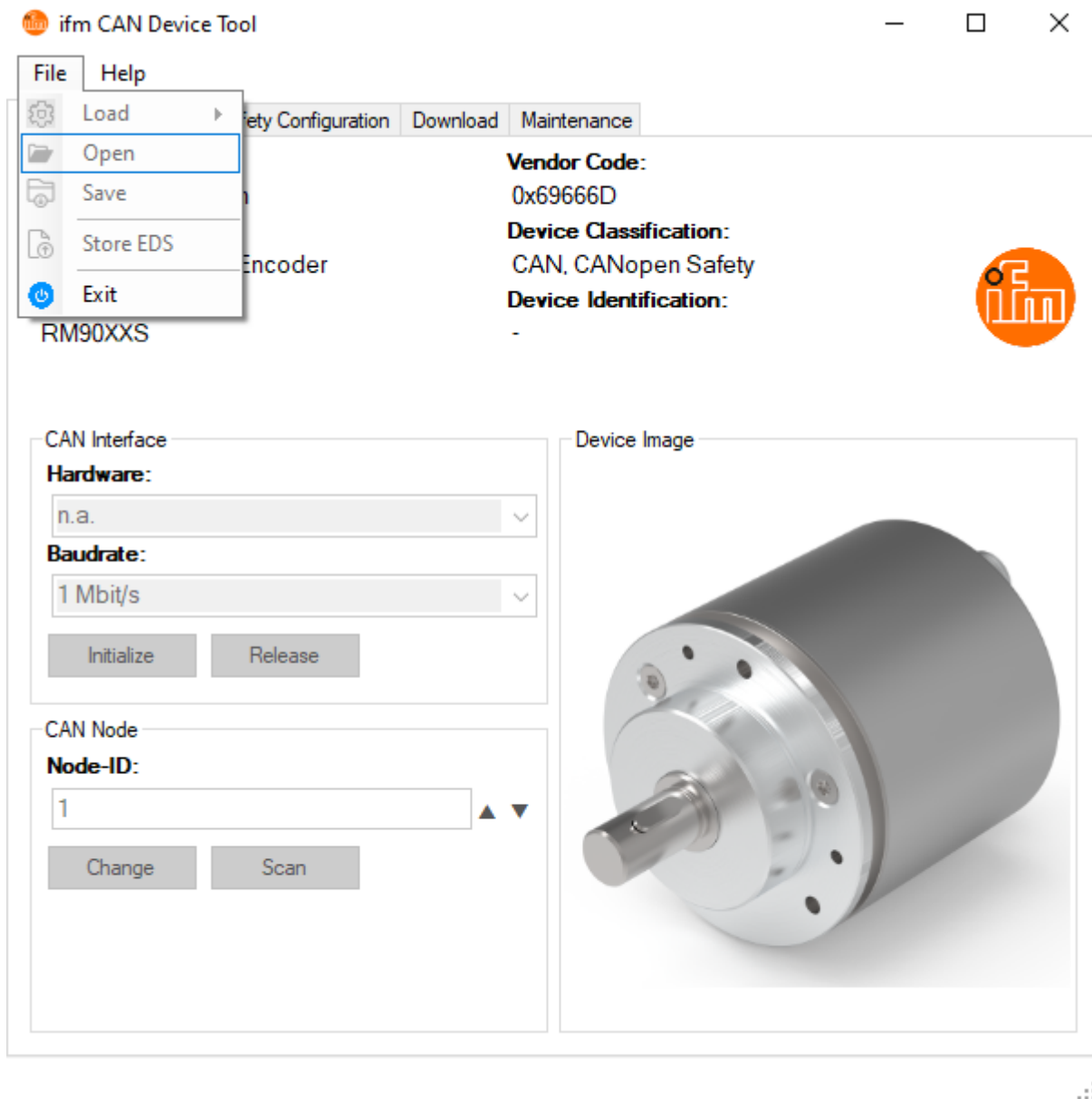


Fig. 10: Dialogue window before opening saved settings

It is possible to open a parameter set from the PRSX file at any time. The previously opened parameter tables are reloaded and overwritten. Instead, the settings from the PRSX file are displayed.



The settings loaded from a PRSX file can be transferred directly to a selected device.

## 9 “Download” use case

The [Download] use case is used to set tool parameters, which means that the ifmCDT configures the encoder (SR) independently of another engineering tool. This is done on the basis of the previously set and calculated safety configuration (“[Safety Configuration](#)” use case (→ [12](#))).

The ifmCDT also enables the device configuration to be read out to verify the device configuration already installed.

Finally, the user can lock the device configuration against changes by means of [Lock] requests.

Fig. 11: Dialogue window for setting tool parameters.

### 9.1 “Download” function

The [Download] function sends the safety configuration set in the tool to the selected device via SDO access.

The settings change the object directory immediately and are saved in a non-volatile way.



The previously made settings are transferred (“[Safety Configuration](#)” use case (→ [12](#))). Download is only possible with a calculated signature and initialised CAN interface.

The device only accepts a download if the device configuration has not been completed.



## 9.2 “Upload” function

The [Upload] function reads the device configuration from the selected device. The ifmCDT compares the information read with the tool settings. The result of the evaluation is displayed in a message dialogue.

The function saves the data record for the device configuration and the comparison with the current setting locally in a file on the Windows computer.



If the read signatures differ from the calculated values of the ifmCDT, the display elements of the deviceSCID are highlighted in yellow.



After the update, the ifmCDT also displays the status of the object "[Configuration valid]" (0x13FE). Without a valid "[Configuration valid]", the node cannot establish SRDO communication.

## 9.3 “Display” function

The [Display] function reads the device configuration saved from the hard disk after a successful upload and displays it in an independent text editor (ifm Configuration Display). This display is to be used to compare the device configuration with the tool configuration. For the safety configuration, all objects compared must be identical. In addition, the ConfigValid object must be valid and have the value 0xA5.

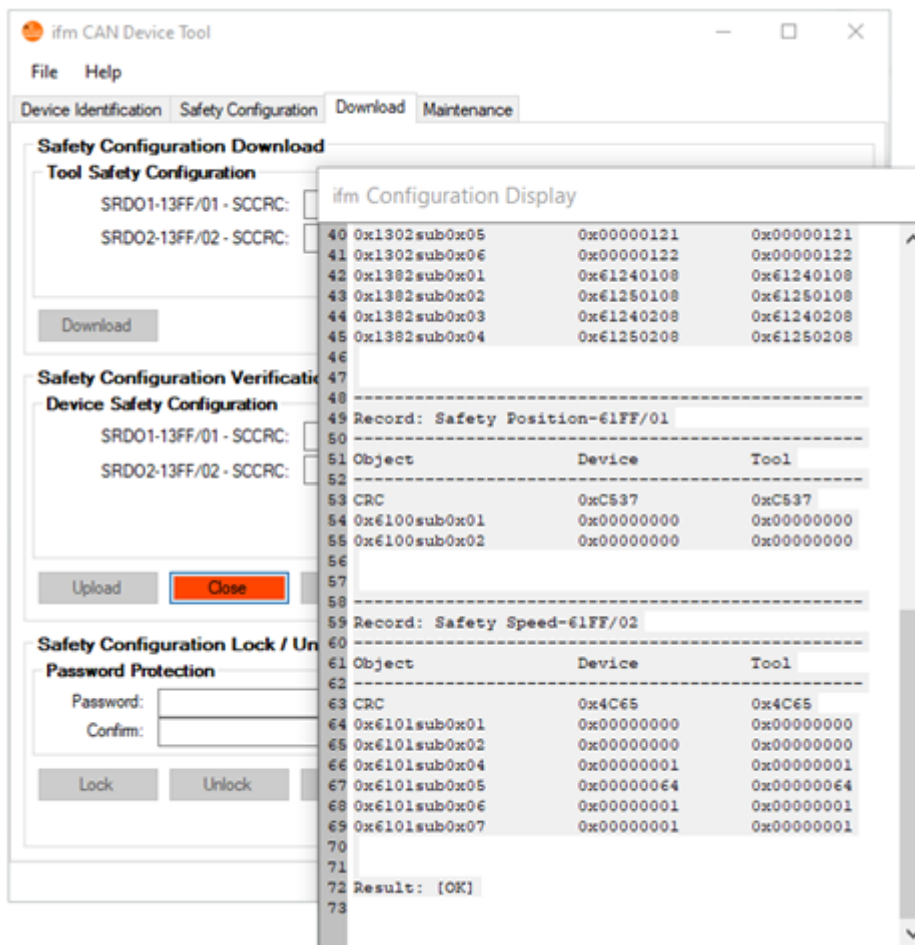


Fig. 12: Dialogue window with external display of the device configuration.

To close the display again, use the [Close] function.



The overall assessment of the comparison is at the end at the text file. The overall rating must be "[OK]".

ifm Configuration Display			
40	0x1302sub0x05	0x00000121	0x00000121
41	0x1302sub0x06	0x00000122	0x00000122
42	0x1382sub0x01	0x61240108	0x61240108
43	0x1382sub0x02	0x61250108	0x61250108
44	0x1382sub0x03	0x61240208	0x61240208
45	0x1382sub0x04	0x61250208	0x61250208
46			
47			
48	-----		
49	Record: Safety Position-61FF/01		
50	-----		
51	Object	Device	Tool
52	-----		
53	CRC	0xC537	0xC537
54	0x6100sub0x01	0x00000000	0x00000000
55	0x6100sub0x02	0x00000000	0x00000000
56			
57			
58	-----		
59	Record: Safety Speed-61FF/02		
60	-----		
61	Object	Device	Tool
62	-----		
63	CRC	0x4C65	0x4C65
64	0x6101sub0x01	0x00000000	0x00000000
65	0x6101sub0x02	0x00000000	0x00000000
66	0x6101sub0x04	0x00000001	0x00000001
67	0x6101sub0x05	0x00000064	0x00000064
68	0x6101sub0x06	0x00000001	0x00000001
69	0x6101sub0x07	0x00000001	0x00000001
70			
71			
72	Result: [OK]		
73			

Fig. 13: Dialogue window with external display of the device configuration.

## 9.4 “Verify” function

The user must verify the transferred safety configuration in the current application. Once the correct function has been validated and the device configuration has been verified, the user confirms the settings with the [Verify] function.



Only a verified configuration can be finalised.

The screenshot shows the 'ifm CAN Device Tool' window with the 'Safety Configuration' tab selected. The interface is divided into three main sections:

- Safety Configuration Download:** Contains 'Tool Safety Configuration' with four input fields: SRD01-13FF/01 - SCCRC (0x250D), SRD02-13FF/02 - SCCRC (0x2083), Safety Position-61FF/01 - SCCRC (0xC537), and Safety Speed-61FF/02 - SCCRC (0x4C65). A 'Download' button is at the bottom.
- Safety Configuration Verification:** Contains 'Device Safety Configuration' with the same four input fields as above. A 'Configuration valid' checkbox is checked. Below are 'Upload', 'Display', and 'Verify' buttons. A status message 'Device Safety Configuration verified' is shown with a checked checkbox.
- Safety Configuration Lock / Unlock:** Contains 'Password Protection' with 'Password:' and 'Confirm:' input fields. Below are 'Lock', 'Unlock', and 'Reset' buttons. A status message 'Device Safety Configuration locked' is shown with an unchecked checkbox.

At the bottom left, it says 'Done.' and there is a green progress bar on the right.

Fig. 14: Dialogue window with display of the verified device configuration.

## 9.5 “Lock” function

The [Lock] function finalises the safety configuration of the device. This configuration can then only be changed after it has been unlocked or reset.



The user can assign a password to finalise the safety configuration. This password is saved in the device and must be reused when unlocking the configuration. If no explicit entry is made, the default password will be used automatically.

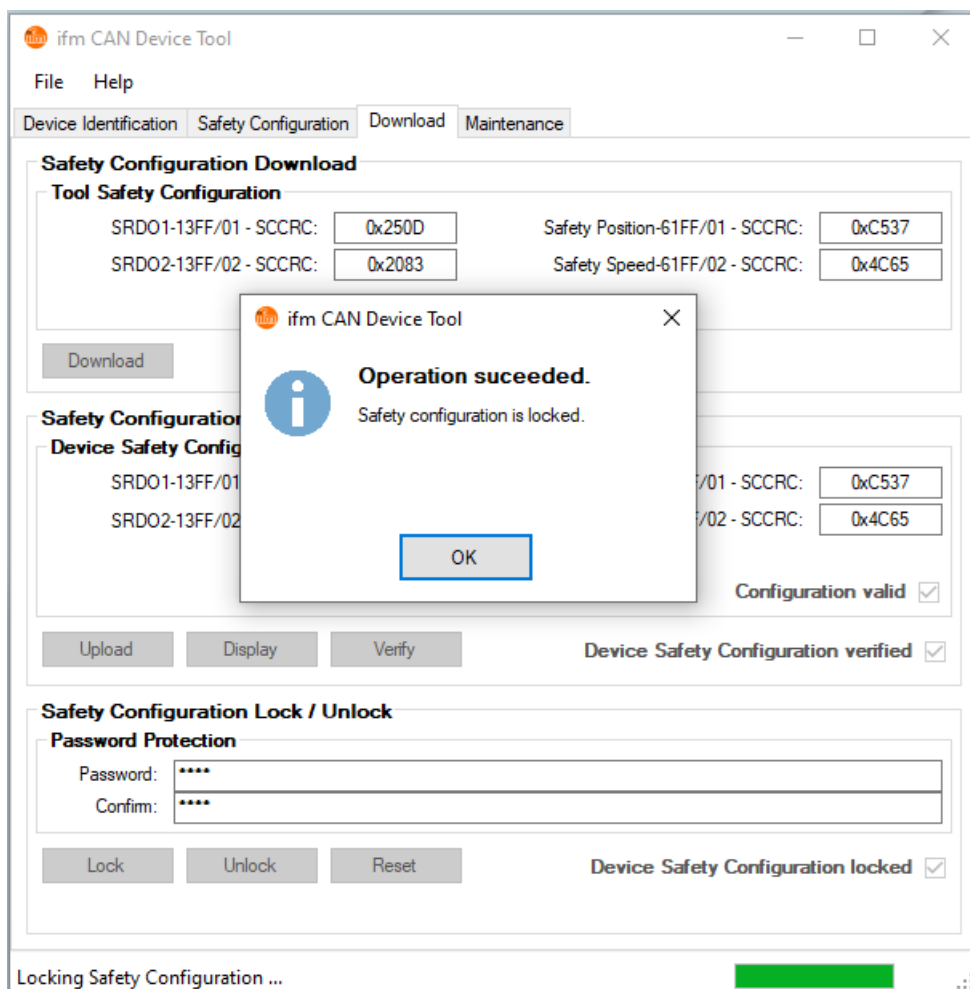


Fig. 15: Dialogue window displaying the completed device configuration.

## 9.6 “Unlock” function

The [Unlock] function unlocks the safety configuration of the device if it was previously locked.



It is necessary to enter the previously used password in the input fields provided in order to unlock the configuration.

If the password is unknown, the device can be reset to the factory settings using the [Reset] function (“Reset” function → [20](#))).

## 9.7 “Reset” function

The [Reset] function resets the safety configuration of the device to the factory settings.



If the security configuration is closed, the user is prompted to confirm the reset or cancel the process in a dialogue box before the reset.

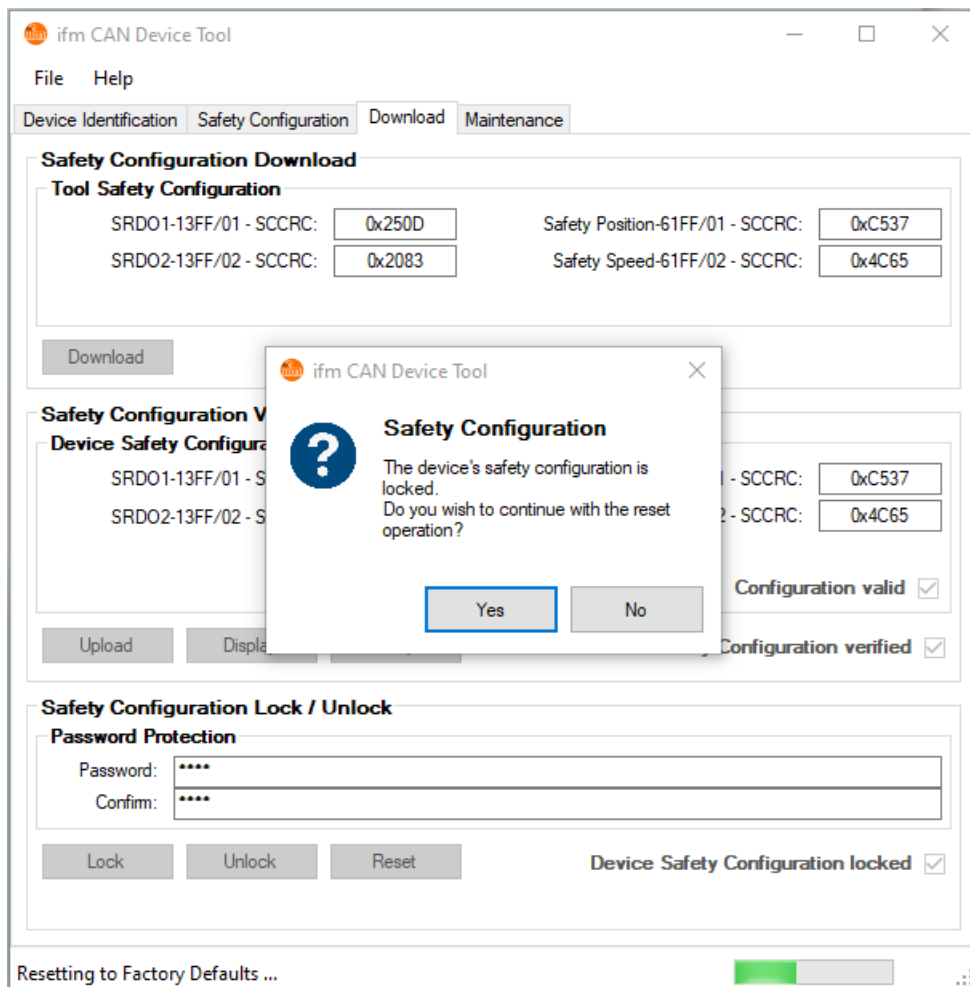


Fig. 16: Dialogue window with user query when device configuration is complete.

## 10 “Maintenance” use case

The maintenance use case is used to display the operating data and the existing diagnostic entries in the device’s diagnostic buffer.

### 10.1 “Read” function

The [Read] function reads out both the operating data and the diagnostic buffer of the device and displays the information on the user interface.

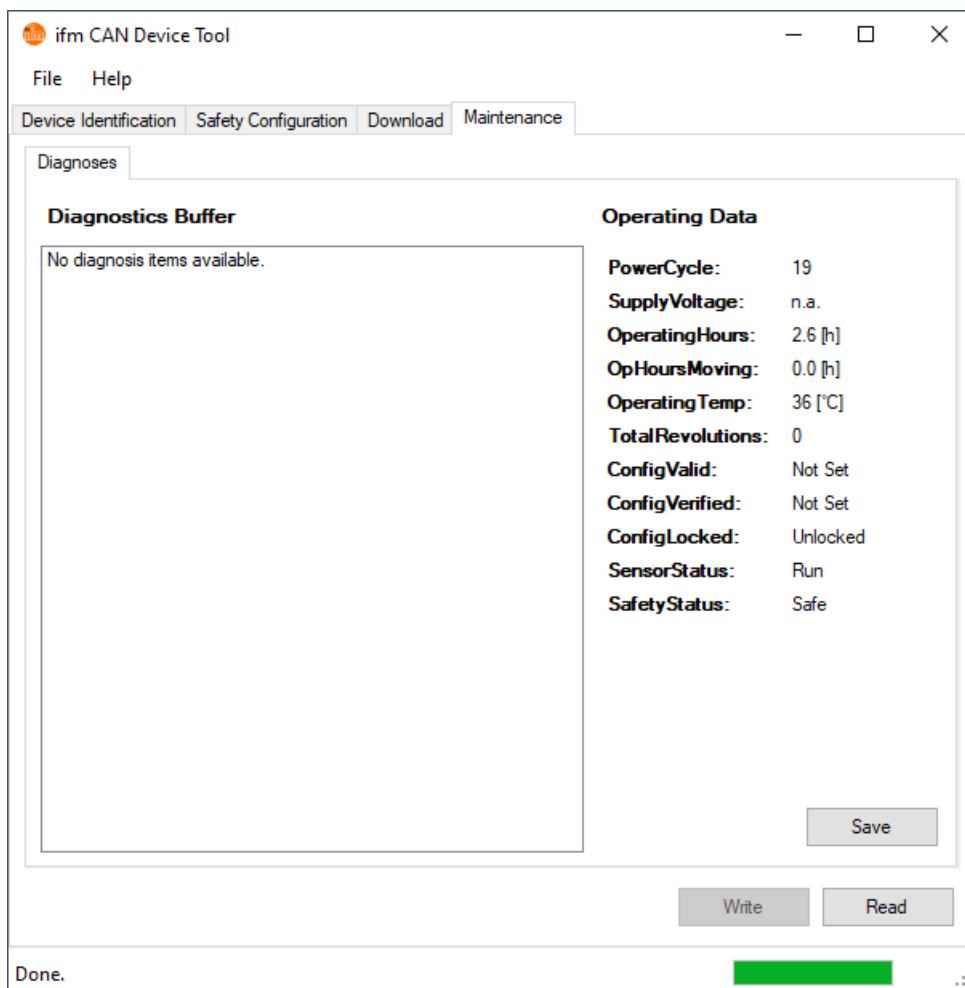


Fig. 17: Dialogue window with display of operating data.

## 11 Language selection

The entire user interface (GUI) of the ifmCDT application is multilingual, which means that the application starts with the corresponding localisations – if available – depending on the settings of the operating system. In addition, the user can change the language setting at runtime via the [Help] menu.

The following languages are currently available:

1. English (default)
2. German
3. French

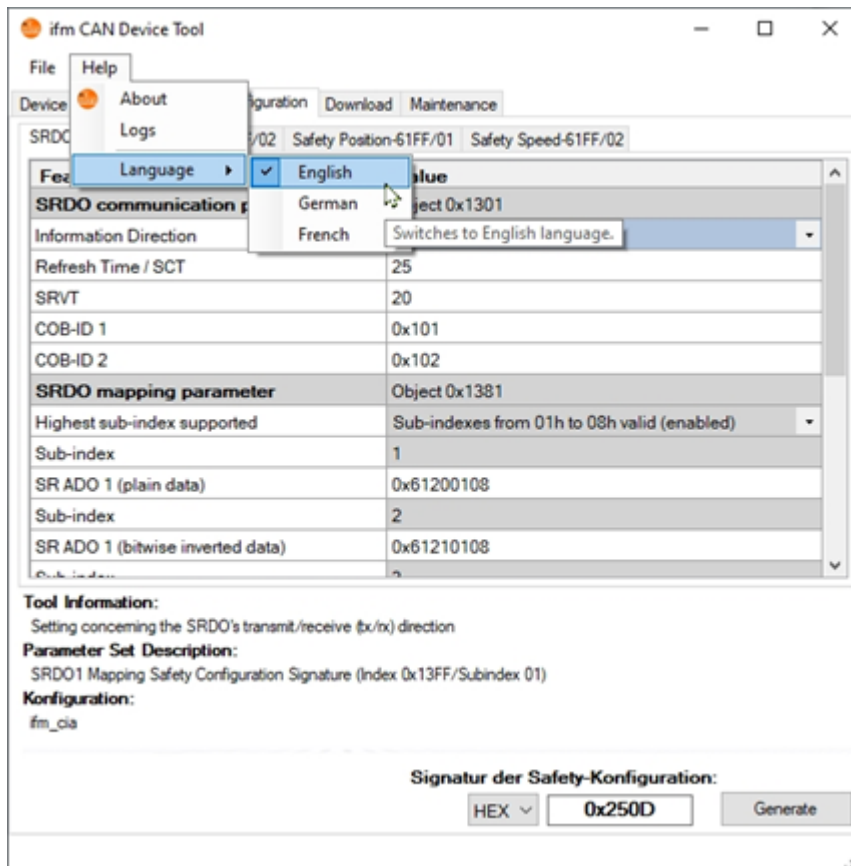


Fig. 18: Dialogue window with the selection of the language setting.



When operating on systems with localisations not listed, the default setting is automatically used.

## 12 Troubleshooting

### 12.1 Log files

The ifmCDT application writes log files in ASCII format each time the program is started. The log files are located below the installation folder in the folder [Logs]. The log files provide information about use cases and error handling.

The user can open the log directory at runtime via the [Help] menu.

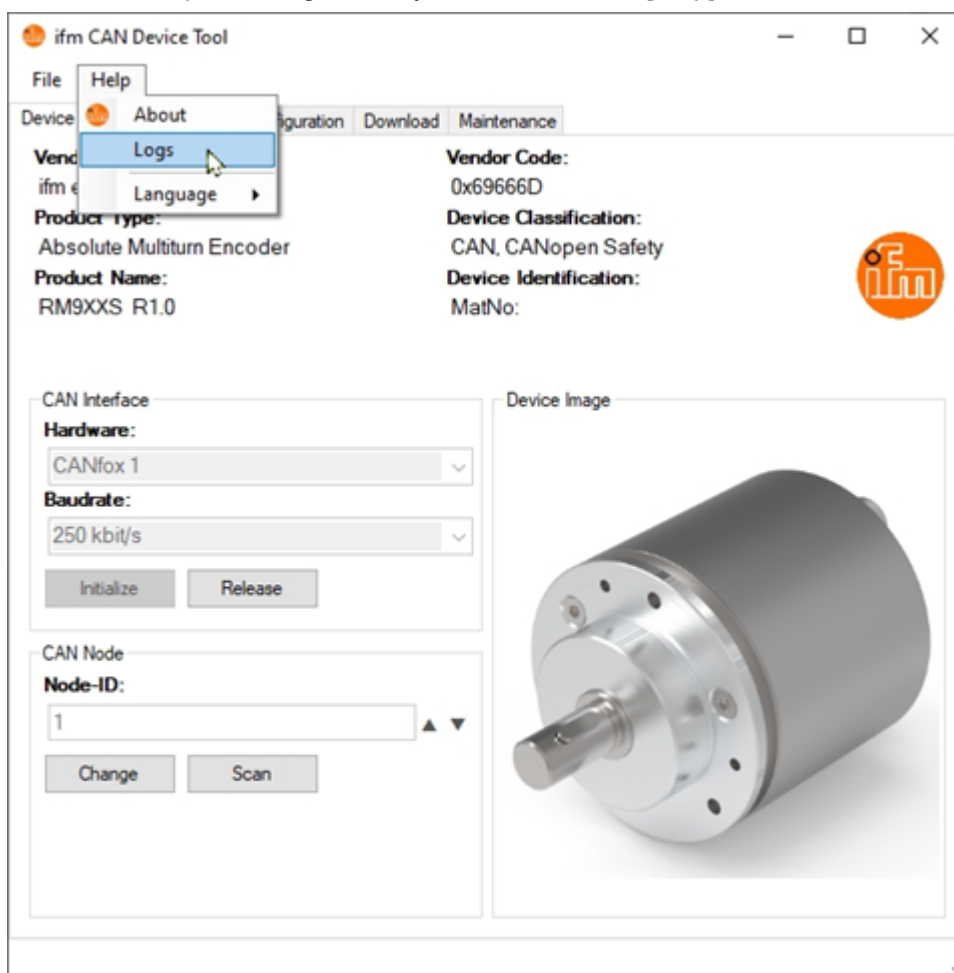


Fig. 19: Dialogue window with menu item for displaying the log directory

### 12.2 “About” dialogue

Information can be found on the application and in particular the version status via the [Help] > [About] menu item.