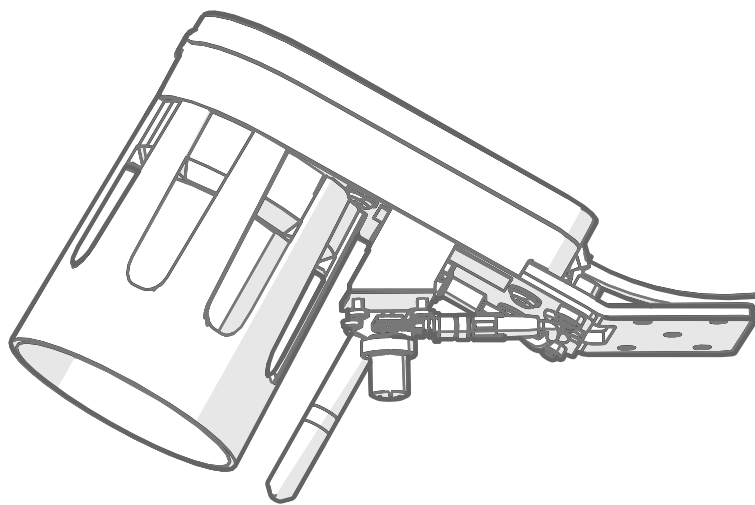


# Technical Reference

Interface Client  
**MD30**



PUBLISHED BY

Vaisala Oyj  
Vanha Nurmijärventie 21, FI-01670 Vantaa, Finland  
P.O. Box 26, FI-00421 Helsinki, Finland  
+358 9 8949 1

Visit our Internet pages at [www.vaisala.com](http://www.vaisala.com).

© Vaisala Oyj 2020

No part of this document may be reproduced, published or publicly displayed in any form or by any means, electronic or mechanical (including photocopying), nor may its contents be modified, translated, adapted, sold or disclosed to a third party without prior written permission of the copyright holder. Translated documents and translated portions of multilingual documents are based on the original English versions. In ambiguous cases, the English versions are applicable, not the translations.

The contents of this document are subject to change without prior notice.

Local rules and regulations may vary and they shall take precedence over the information contained in this document. Vaisala makes no representations on this document's compliance with the local rules and regulations applicable at any given time, and hereby disclaims any and all responsibilities related thereto.

This document does not create any legally binding obligations for Vaisala towards customers or end users. All legally binding obligations and

agreements are included exclusively in the applicable supply contract or the General Conditions of Sale and General Conditions of Service of Vaisala.

This product contains software developed by Vaisala or third parties. Use of the software is governed by license terms and conditions included in the applicable supply contract or, in the absence of separate license terms and conditions, by the General License Conditions of Vaisala Group.

This product may contain open source software (OSS) components. In the event this product contains OSS components, then such OSS is governed by the terms and conditions of the applicable OSS licenses, and you are bound by the terms and conditions of such licenses in connection with your use and distribution of the OSS in this product. Applicable OSS licenses are included in the product itself or provided to you on any other applicable media, depending on each individual product and the product items delivered to you.

## Table of contents

<b>1.</b>	<b>About This Document</b> .....	<b>3</b>
1.1	Version information.....	3
1.2	Related manuals.....	3
1.3	Documentation conventions.....	3
1.4	Trademarks.....	4
<b>2.</b>	<b>Interface Client Overview</b> .....	<b>5</b>
<b>3.</b>	<b>Installing Interface Client</b> .....	<b>6</b>
<b>4.</b>	<b>Starting Interface Client</b> .....	<b>7</b>
<b>5.</b>	<b>Commands</b> .....	<b>11</b>
5.1	Serial Interface Commands.....	11
5.1.1	Get Unit ID Command.....	12
5.1.2	Get Product Info Command.....	12
5.1.3	Get Unit Status Command.....	12
5.1.4	Send Data Command.....	13
5.1.5	Set References Command.....	15
5.1.6	Stop Reference Setting.....	16
5.1.7	Set Road Coefficients Command.....	16
5.1.8	Get Parameter Message.....	17
5.1.9	Set Parameter Command.....	18
5.1.10	Reset Command.....	19
5.2	Other Commands.....	20
5.2.1	Help Command.....	20
5.2.2	Source Command.....	21
5.2.3	Stop Continuous Data Sending Command.....	22
5.2.4	Wait Command.....	23
5.2.5	Bye command.....	23
5.2.6	End of File (eof) Command.....	23
5.2.7	Set Message Number Command.....	23
5.2.8	Set Sender ID Command.....	24
5.2.9	Set Receiver ID Command.....	24
5.2.10	Set Delay Command.....	25
5.2.11	Echo Mode Command.....	26
5.2.12	License Command.....	26
<b>6.</b>	<b>Scripts</b> .....	<b>28</b>
	<b>Technical support</b> .....	<b>31</b>
	<b>Recycling</b> .....	<b>31</b>



# 1. About This Document

## 1.1 Version information

This document describes the installation and use of the MD30 interface client.

Table 1 Document versions (English)

Document code	Date	Description
M212384EN-C	June 2020	<ul style="list-style-type: none"> <li>Updated description of ZIP package contents (document <i>Vaisala Mobile Detector MD30 Interface Description</i> removed)</li> </ul>
M212384EN-B	March 2020	<ul style="list-style-type: none"> <li>For version 2.0 of the MD30 interface client.</li> <li>For version C of the interface.</li> </ul>
M212384EN-A	November 2019	First version. <ul style="list-style-type: none"> <li>For version 1.0 of the MD30 interface client.</li> <li>For version C of the interface.</li> </ul>

## 1.2 Related manuals

Table 2 MD30 manuals

Document code	Name
M212339EN	<i>Vaisala Mobile Detector MD30 Product and Package Description Reference Guide</i>
M212169EN	<i>Vaisala Mobile Detector MD30 Setup Guide</i>
M212309EN	<i>Vaisala Mobile Detector MD30 Maintenance and Troubleshooting Reference Guide</i>
M212432EN	<i>Mounting Temperature Sensors Separately Installation Guide</i>
M212201EN	<i>Vaisala Mobile Detector MD30 Interface Description</i>
M212384EN	<i>Vaisala Mobile Detector MD30 Interface Client Technical Reference</i>
M212382EN	<i>Vaisala Mobile Detector MD30 Interface Client License Agreement</i>
M212310EN	<i>Vaisala Mobile Detector MD30 OSS Licenses</i>

## 1.3 Documentation conventions



**WARNING!** Warning alerts you to a serious hazard. If you do not read and follow instructions carefully at this point, there is a risk of injury or even death.



**CAUTION! Caution** warns you of a potential hazard. If you do not read and follow instructions carefully at this point, the product could be damaged or important data could be lost.



**Note** highlights important information on using the product.



**Tip** gives information for using the product more efficiently.



Lists tools needed to perform the task.



Indicates that you need to take some notes during the task.

## 1.4 Trademarks

Vaisala® is a registered trademark of Vaisala Oyj.

Python® is a trademark or registered trademark of the Python Software Foundation.

All other product or company names that may be mentioned in this publication are trade names, trademarks, or registered trademarks of their respective owners.

## 2. Interface Client Overview

The MD30 interface client is a Python script that demonstrates the functionality of the mobile road sensor interface.

The interface client can be used to verify the implementation of the data reporting interface of Vaisala Mobile Detector MD30.

You can use the interface client to:

- View the available commands
- Test the commands

The functionality of the mobile road sensor interface is covered in *Vaisala Mobile Detector MD30 Interface Description*.



Use this document and *Vaisala Mobile Detector MD30 Interface Description* together, and make sure that they are for the same version of the interface.

You can implement the mobile road sensor interface with Python, C++, or other programming language of your choice.

The interface client comes in a zip package.

Table 3 ZIP Package Contents

Name	Description
<i>md30InterfaceClient.py</i>	Interface client software
<i>Communication</i>	Folder for serial port communication
<i>ICDlibrary</i>	Folder for MD30 command handlers
<i>Interpreter.py</i>	Command line interface
<i>MessageReceiver.py</i>	Data receiver
<i>exampleScript.txt</i>	Command script example
<i>Licenses.txt</i>	Open source licenses used by interface client
<i>Documentation</i>	Folder that contains: <ul style="list-style-type: none"> <li>• <i>Vaisala Mobile Detector MD30 Interface Client Technical Reference</i> (this document)</li> <li>• <i>Vaisala Mobile Detector MD30 Interface Client License Agreement</i></li> </ul>

## 3. Installing Interface Client

The interface client has been developed and tested in Windows 10 with Python 3.7.4 and PySerial 3.4.

To use the interface client, you need the following (not delivered by Vaisala):

1. Python 3
  2. Module PySerial
- ▶ 1. Install Python 3, if you do not have it already:
- a. Download Python 3 installer from <https://www.python.org/>.
  - b. Install the software. Follow the instructions in <https://docs.python.org/3/using/index.html>.  
In the instructions given in this document, it is assumed that python installation path is added to the PATH environment variable.
2. Install Module PySerial, if you do not have it already:
- a. See the instruction about Python module installation: <https://docs.python.org/3/using/index.html>.
  - b. To install, type the following command:

```
pip install pyserial
```

3. Download the MD30 interface client file zip from <http://www.vaisala.com/MD30>.
4. Extract the zip file contents to a desired location.

Go to the location where you extracted the zip file and open command prompt. See [Starting Interface Client \(page 7\)](#).



## 4. Starting Interface Client

You can use some of the interface client commands without a connection to MD30.

To test the MD30 interface commands, you must have an RS-232 connection to MD30 and MD30 must be powered from a suitable power supply.

- For RS-232 settings, see *Vaisala Mobile Detector MD30 Interface Description*
- For MD30 cable pinout, see *Vaisala Mobile Detector MD30 Setup Guide*

To start the interface client, you need to know the port to which MD30 is connected and the MD30 port speed.

1. Go to the folder where the interface client is installed and open Command Prompt.
2. The default port speed of the interface client is 115200 and the default port COM0.
  - If the MD30 port and port speed match the default settings, type the following:

```
python md30InterfaceClient.py
```

- If you need to change the port and/or port speed of the interface client to match the MD30 port speed, type the following:

```
python md30InterfaceClient.py -P <port> -B <port speed>
```

For example:

```
python md30InterfaceClient.py -P COM4 -B 9600
```

In both cases, the interface client starts and gives the following response:

```
Welcome to MD30 interface client. MD30 interface client demonstrates
the features of the MD30 serial interface.

MD30 interface client version 1.0
Interface version C
Copyright (c) Vaisala Oyj. All rights reserved.
/>
```

- If you need to change the port speed of MD30 to match the interface client port speed, continue to [step 3](#).

3. To change the port speed of MD30 to match the interface client:
  - a. Use parameter 0x10 to define the port speed of MD30:

```
setparam 0x10 <port speed>
```

For example:

```
setparam 0x10 4
```

Property	Value
Port speed	<ul style="list-style-type: none"> <li>• 0 = 9600</li> <li>• 1 = 19200</li> <li>• 2 = 38400</li> <li>• 3 = 57600</li> <li>• 4 = 115200</li> </ul>

- b. Reset MD30:

```
reset
```

- c. Close the interface client:

```
bye
```

- d. Restart the interface client using the new port speed:

```
python md30InterfaceClient.py -P <port> -B <port speed>
```

For example:

```
python md30InterfaceClient.py -P COM0 -B 115200
```

Example:

Port speed is changed from 115200 bps to 9600 bps and back. Parameter 0x10 contains the port speed, which is taken into use after the reset.

```
C:\>python md30InterfaceClient.py -P COM4 -B 115200

Welcome to MD30 interface client. MD30 interface client demonstrates the
features of the MD30 serial interface.

MD30 interface client version 1.0
Interface version C
Copyright (c) Vaisala Oyj. All rights reserved.
/> setparam 0x10 0
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 0, Len: 2, icd_version: C,
Err: 0

/> reset
/> Unit ID: 1, Client ID: 0, ID: 0x50, Nb: 1, Len: 2, icd_version: C,
Err: 0
RESET ACK

/> bye

C:\>python md30InterfaceClient.py -P COM4 -B 9600

Welcome to MD30 interface client. MD30 interface client demonstrates the
features of the MD30 serial interface.

MD30 interface client version 1.0
Interface version C
Copyright (c) Vaisala Oyj. All rights reserved.
/> getunitid
/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 0, Len: 10, icd_version: C,
Err: 0
Serial number: R2730001

/> setparam 0x10 4
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 1, Len: 2, icd_version: C,
Err: 0

/> getparam 0x10
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 2, Len: 5, icd_version: C,
Err: 0
ID: 0x10 Value: 4 (Baud rate)

/> reset
/> Unit ID: 1, Client ID: 0, ID: 0x50, Nb: 3, Len: 2, icd_version: C,
Err: 0
RESET ACK
```

```
/> bye
```

```
C:\>python md30InterfaceClient.py -P COM4 -B 115200
```

```
Welcome to MD30 interface client. MD30 interface client demonstrates the  
features of the MD30 serial interface.
```

```
MD30 interface client version 1.0
```

```
Interface version C
```

```
Copyright (c) Vaisala Oyj. All rights reserved.
```

```
/> getunitid
```

```
/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 0, Len: 10, icd_version: C,
```

```
Err: 0
```

```
Serial number: R2730001
```

# 5. Commands

The interface client operates as a command line interpreter.

You can use the following types of commands in the interface client:

- Serial interface commands for MD30 interaction
- Other commands for interface client operation

The commands are case insensitive.



To browse the command history, use the up and down arrows of your computer keyboard.

## More information

- [Serial Interface Commands \(page 11\)](#)
- [Other Commands \(page 20\)](#)

## 5.1 Serial Interface Commands

Every response to a serial interface command contains a header.

Table 4 Header Information in Response Messages

Field	Description
Unit ID	Message receiver ID
Client ID	Message sender ID
ID	Message ID
Nb	Message number
Len	Data length
icd_version	Interface version
Err	Error code

## More information

- [Get Unit ID Command \(page 12\)](#)
- [Get Product Info Command \(page 12\)](#)
- [Get Unit Status Command \(page 12\)](#)
- [Send Data Command \(page 13\)](#)
- [Set References Command \(page 15\)](#)
- [Stop Reference Setting \(page 16\)](#)
- [Set Road Coefficients Command \(page 16\)](#)
- [Get Parameter Message \(page 17\)](#)
- [Set Parameter Command \(page 18\)](#)
- [Reset Command \(page 19\)](#)

### 5.1.1 Get Unit ID Command

The **getunitid** command sends the GET UNIT ID message to MD30 and parses the response.

#### Example

```
> getunitid
> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 0, Len: 10, icd_version: C, Err: 0
Serial number: R2730001
```

### 5.1.2 Get Product Info Command

The **getproductinfo** command sends the GET FULL PRODUCT INFO message to MD30 and parses the response.

The command does not take parameters.

#### Example

```
> getproductinfo
> Unit ID: 1, Client ID: 0, ID: 0x11, Nb: 2, Len: 113, icd_version: C, Err: 0
Product Name: MD30, Serial Number: R2730001, SW Version: 1.0.0, MT10 ID:
541226C864A4B1C2, HMP Serial Number: R1120189
```

### 5.1.3 Get Unit Status Command

The **getunitstatus** command sends the GET UNIT STATUS message to MD30 and parses the response.

## Example

```

/> getunitstatus
/> Unit ID: 1, Client ID: 0, ID: 0x12, Nb: 1, Len: 10, icd_version: C, Err: 0
Status info: 0x00000000, Error bits: 0x00000000

```

## 5.1.4 Send Data Command

The **senddata** command sends the SEND DATA message to MD30 and parses the response.

The data interval is given as a parameter in milliseconds. If the parameter value is:

- 0 = only one data set is sent as a response
- 25 ... 5000 = message sending starts with the defined data interval

Only numeric values are counted as parameters.

### Example 1

Data request with data interval 0.

Only one data set is sent as a response.

```

/> senddata 0
/> Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 3, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.48, RH: 38.24, DP:
8.44, FP:
8.44, TS 22.11, C: 14468, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000

```

W	Water layer thickness
I	Ice layer thickness
S	Snow layer thickness
G	Grip
St	Road state
EN	EN15518 state
TA	Air temperature
RH	Relative humidity of the air
DP	Dew point temperature
FP	Frost point temperature
TS	Surface temperature
C	Analyze count
DW	Data status warning
DE	Data status error
USt	Unit status info
UE	Unit error info

### Example 2

Data request with data interval 1000.

The unit starts to send data with the defined data interval. You can stop the data sending with command **s**.

```

/> senddata 1000
/> Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 6, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.49, RH: 38.23, DP:
8.45, FP:
8.45, TS 22.47, C: 17129, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 7, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.49, RH: 38.23, DP:
8.45, FP:
8.45, TS 22.35, C: 17168, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 8, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.49, RH: 38.23, DP:
8.45, FP:
8.45, TS 22.25, C: 17208, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 9, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.49, RH: 38.23, DP:
8.45, FP:
8.45, TS 22.35, C: 17247, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 10, Len: 54, icd_version: C, Err: 0
W: 0.01, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.49, RH: 38.23, DP:
8.45, FP:
8.45, TS 22.13, C: 17287, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 11, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.49, RH: 38.23, DP:
8.45, FP:
8.45, TS 22.31, C: 17326, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 12, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.49, RH: 38.23, DP:
8.45, FP:
8.45, TS 22.29, C: 17366, DW: 0x0000, DE: 0x0000, USt: 0x00000000, UE:
0x00000000

```

### Example 3

Data request with data interval **10**.

The data interval is not valid, which causes an invalid data error (error code = 4).

```

/> senddata 10
/> Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 8, Len: 2, icd_version: C, Err: 4

```





For an explanation of the error codes, see *Vaisala Mobile Detector MD30 Interface Description*.

### More information

- [Stop Continuous Data Sending Command \(page 22\)](#)

## 5.1.5 Set References Command

The **setrefs** command sends the SET REFERENCES message to the unit and parses the response.

The set references command initiates the reference setting either with parameter PLATE = 0 or ROAD = 1.

### Example 1

Set references with parameter PLATE (0). The status info bits indicate that the input voltage is low. However, this condition does not prevent the reference setting.

- In the response to **setrefs**, the success is 1, which means that reference setting was started successfully.  
The status request after sending the **setrefs** command confirms that the reference setting is ongoing.  
The last status request, which is sent after about 30 seconds, confirms that the reference setting is ready.

```

/> getunitstatus
/> Unit ID: 1, Client ID: 0, ID: 0x12, Nb: 12, Len: 10, icd_version: C, Err: 0
Status info: 0x00000020, Error bits: 0x00000000
/> setrefs 0

/> Unit ID: 1, Client ID: 0, ID: 0x30, Nb: 13, Len: 11, icd_version: C, Err: 0
Success: 1, Status info: 0x00000020, Error bits: 0x00000000

/> getunitstatus
/> Unit ID: 1, Client ID: 0, ID: 0x12, Nb: 14, Len: 10, icd_version: C, Err: 0
Status info: 0x00000022, Error bits: 0x00000000

/> getunitstatus
/> Unit ID: 1, Client ID: 0, ID: 0x12, Nb: 17, Len: 10, icd_version: C, Err: 0
Status info: 0x00000020, Error bits: 0x00000000

```

### Example 2

Set references fails if a reference setting is already ongoing. An attempt to restart the reference setting does not interfere with the ongoing reference setting.

```

/> setrefs 1
/> Unit ID: 1, Client ID: 0, ID: 0x30, Nb: 22, Len: 11, icd_version: C, Err: 0
Success: 1, Status info: 0x00000000, Error bits: 0x00000000
/> setrefs 1
/> Unit ID: 1, Client ID: 0, ID: 0x30, Nb: 23, Len: 11, icd_version: C, Err: 0
Success: 0, Status info: 0x00000002, Error bits: 0x00000000
/> getunitstatus
/> Unit ID: 1, Client ID: 0, ID: 0x12, Nb: 24, Len: 10, icd_version: C, Err: 0
Status info: 0x00000002, Error bits: 0x00000000

```

## 5.1.6 Stop Reference Setting

The **stoprefs** command sends the STOP REFERENCE SETTING message to the unit.

### Example

The reference setting is started and it is checked that it is actually ongoing. After the **stoprefs** command the reference setting is no longer ongoing and it is indicated that the reference setting was interrupted by the client.

```

/> setrefs 1
/> Unit ID: 1, Client ID: 0, ID: 0x30, Nb: 25, Len: 11, icd_version: C, Err: 0
Success: 1, Status info: 0x00000000, Error bits: 0x00000000

/> getunitstatus
/> Unit ID: 1, Client ID: 0, ID: 0x12, Nb: 26, Len: 10, icd_version: C, Err: 0
Status info: 0x00000002, Error bits: 0x00000000

/> stoprefs
/> Unit ID: 1, Client ID: 0, ID: 0x32, Nb: 27, Len: 2, icd_version: C, Err: 0
Stop reference setting ack

/> getunitstatus
/> Unit ID: 1, Client ID: 0, ID: 0x12, Nb: 28, Len: 10, icd_version: C, Err: 0
Status info: 0x00002000, Error bits: 0x00000000

```

## 5.1.7 Set Road Coefficients Command

The **setroadcoeffs** command sends the SET ROAD COEFFICIENTS message to the unit and parses the response.

The command updates parameters 0x53, 0x54, and 0x55. All 3 values must be defined.

## Example

```

/> setroadcoeffs 1.5 2.25 3.0
/> Unit ID: 1, Client ID: 0, ID: 0x31, Nb: 6, Len: 3, icd_version: C, Err: 0
Success: 1

/> getparam 0x53
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 7, Len: 8, icd_version: C, Err: 0
ID: 0x53 Value: 1.50000 (Reference setting value of laser 1)

/> getparam 0x54
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 8, Len: 8, icd_version: C, Err: 0
ID: 0x54 Value: 2.25000 (Reference setting value of laser 2)

/> getparam 0x55
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 9, Len: 8, icd_version: C, Err: 0
ID: 0x55 Value: 3.00000 (Reference setting value of laser 3)

```

## 5.1.8 Get Parameter Message

The **getparam** command sends the GET PARAMETER message to MD30 and parses the response.

The command takes the parameter data ID as a parameter. Only numeric values are counted as parameters.

### Example 1

```

/> getparam 0x20
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 9, Len: 6, icd_version: C, Err: 0
ID: 0x20 Value: 0 (Autosend data interval)

```

### Example 2

```

/> getparam 0x51
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 4, Len: 8, icd_version: C, Err: 0
ID: 0x51 Value: 11.02406 (Reference value of laser 2)

```

### Example 3

The given ID 0x15 is undefined, which causes an invalid data error (error code = 4).

```

/> getparam 0x15
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 11, Len: 2, icd_version: C, Err: 4
Error reading data

```



For an explanation of the error codes, see *Vaisala Mobile Detector MD30 Interface Description*.

## 5.1.9 Set Parameter Command

The **setparam** command sends the SET PARAMETER message to the unit and parses the response.

The command takes 2 parameters: the parameter ID and the new parameter value. Only numeric values are counted as parameters.

### Example 1

Setting an integer value. Only integer values are accepted.

```

/> setparam 0x11 1
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 2, Len: 2, icd_version: C, Err: 0
/> getparam 0x11

/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 3, Len: 5, icd_version: C, Err: 0
ID: 0x11 Value: 1 (Send CRC ack)
/> setparam 0x11 0.5
Value given in wrong format

/> getparam 0x11
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 4, Len: 5, icd_version: C, Err: 0
ID: 0x11 Value: 1 (Send CRC ack)

```

### Example 2

Setting a floating point value.

```

/> setparam 0x40 0.95
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 5, Len: 2, icd_version: C, Err: 0

```

### Example 3

Assigning an invalid ID.

An invalid ID causes an invalid data error (error code = 4).

```

/> setparam 0x16 1
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 7, Len: 2, icd_version: C, Err: 4

```

### Example 4

Using an invalid value.

An invalid value causes an invalid data error (error code = 4).

```
/> setparam 0x31 2
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 8, Len: 2, icd_version: C, Err: 4
```

### Example 5

Setting a value to a read-only parameter.

An attempt to change the value of a read-only parameter causes an invalid data error (error code = 4).

```
/> setparam 0x12 1
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 9, Len: 2, icd_version: C, Err: 4
```



For an explanation of the error codes, see *Vaisala Mobile Detector MD30 Interface Description*.

## 5.1.10 Reset Command

The **reset** command sends the `RESTART UNIT` message to MD30 and parses the response.

### Example

After sending the acknowledgement the MD30 restarts itself.

```
/> reset
/> Unit ID: 1, Client ID: 0, ID: 0x50, Nb: 33, Len: 2, icd_version: C, Err: 0
RESET ACK
```

## 5.2 Other Commands

### More information

- [Help Command \(page 20\)](#)
- [Source Command \(page 21\)](#)
- [Stop Continuous Data Sending Command \(page 22\)](#)
- [Wait Command \(page 23\)](#)
- [Bye command \(page 23\)](#)
- [End of File \(eof\) Command \(page 23\)](#)
- [Set Message Number Command \(page 23\)](#)
- [Set Sender ID Command \(page 24\)](#)
- [Set Receiver ID Command \(page 24\)](#)
- [Set Delay Command \(page 25\)](#)
- [Echo Mode Command \(page 26\)](#)
- [License Command \(page 26\)](#)

### 5.2.1 Help Command

The **he`l`p** and **?**  commands print the version information of the interface client and the list of commands.

The **he`l`p <command>** prints a short description of the requested command.

#### Example 1

```

/> help

MD30 interface client version 1.0
Interface version C
Copyright (c) Vaisala Oyj. All rights reserved.

Documented commands (type help <topic>):
=====
bye          getproductinfo  license  setdelay      setrefs      stoprefs
echo        getunitid       reset    setmessagenumber setroadcoeffs wait
eof         getunitstatus   s        setparam      setsenderid
getparam    help            senddata setreceiverid  source

```

#### Example 2

```

/> help getparam
"Get a parameter value device: getparam 0x20

```

## 5.2.2 Source Command

The **source** command is used for executing a set of commands from a file. The interface client shows the command output.

The file may also contain comment lines, which start with #.

### Example 1

The file *exampleScript.txt* contains the commands: **getunitid**, **getproductinfo**, and **getunitid**.

```
getunitid
getproductinfo
getunitid
```

Executing *exampleScript.txt*.

```
/> source exampleScript.txt
Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 2, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

Unit ID: 1, Client ID: 0, ID: 0x11, Nb: 3, Len: 113, icd_version: C, Err: 0
Product Name: MD30, Serial Number: R2730001, SW Version: 1.0.0, MT10 ID:
541226C864A4B1C2, HMP
Serial Number: R1120189

/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 4, Len: 10, icd_version: C, Err: 0
Serial number: R2730001
```

### Example 2

The commented line, product info request, is removed from the response.

```
getunitid
#getproductinfo
getunitid
```

Executing the modified *exampleScript.txt*.

```

/> source exampleScript.txt
Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 10, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 11, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

/>

```

### More information

- [Scripts \(page 28\)](#)

## 5.2.3 Stop Continuous Data Sending Command

The **s** command can be used to stop continuous data sending or to request a single set of measurement data.

The command **s** is the same as **s enddata 0**.

### Example

```

/> senddata 1000
/> Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 12, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.94, RH: 38.35, DP:
8.89, FP:
8.89, TS 23.48, C: 15296, DW: 0x07e0, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 13, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.94, RH: 38.35, DP:
8.89, FP:
8.89, TS 23.64, C: 15335, DW: 0x07e0, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 14, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.94, RH: 38.35, DP:
8.89, FP:
8.89, TS 23.46, C: 15375, DW: 0x07e0, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 15, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.94, RH: 38.35, DP:
8.89, FP:
8.89, TS 23.42, C: 15414, DW: 0x07e0, DE: 0x0000, USt: 0x00000000, UE:
0x00000000
s
/> Unit ID: 1, Client ID: 0, ID: 0x20, Nb: 13, Len: 54, icd_version: C, Err: 0
W: 0.00, I: 0.00, S: 0.00, G: 0.82, St: 1, EN: 1, TA: 23.94, RH: 38.35, DP:
8.89, FP:
8.89, TS 23.58, C: 15444, DW: 0x07e0, DE: 0x0000, USt: 0x00000000, UE:
0x00000000

```



**More information**

- ▶ [Send Data Command \(page 13\)](#)

## 5.2.4 Wait Command

The **wait** command causes the interface client to wait a defined time period. The time is given in milliseconds. The command can be used in scripts to control the time between commands.

```
wait 5
```

## 5.2.5 Bye command

The **bye** command closes the interface client.

```
bye
```

## 5.2.6 End of File (eof) Command

The **eof** command closes the interface client.

```
eof
```

## 5.2.7 Set Message Number Command

The **setmessagenumber** command defines the message number that is used in the command. The response message has the same number (**Nb**). The value can be from 0 to 255.

## Example

```
> setmessagenumber 10
> getunitid
> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 10, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

> setmessagenumber 100
> getunitid
> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 100, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

> setmessagenumber 0
> getunitid
> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 0, Len: 10, icd_version: C, Err: 0
Serial number: R2730001
```

## 5.2.8 Set Sender ID Command

The **setsenderid** command defines the sender ID that is used in the commands sent by the interface client.

The value can be from 0 to 255. In the interface client, the sender ID is shown as **Client ID**.

### Example

```
> setsenderid 10
> getunitid
> Unit ID: 1, Client ID: 10, ID: 0x10, Nb: 1, Len: 10, icd_version: C, Err: 0
Serial number: R2730001
```

## 5.2.9 Set Receiver ID Command

The **setreceiverid** command defines the receiver ID that is used in the commands sent by the interface client.

The value can be from 0 to 255. In the interface client, the receiver ID is shown as **Unit id**. When the receiver ID is changed, the change must be done also to MD30 to be able to communicate with the device and the change requires reset before to take effect.

### Example 1

The receiver ID of MD30 is changed to 10. The communication works with the old receiver ID until a reset command is given. After the reset, the interface client must use the correct receiver ID to be able to communicate.

```

/> getunitid
/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 0, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

/> setparam 0x13 10
/> Unit ID: 1, Client ID: 0, ID: 0x41, Nb: 1, Len: 2, icd_version: C, Err: 0

/> getparam 0x13
/> Unit ID: 1, Client ID: 0, ID: 0x40, Nb: 2, Len: 5, icd_version: C, Err: 0
ID: 0x13 Value: 10 (Sensor ID)

/> reset
/> Unit ID: 1, Client ID: 0, ID: 0x50, Nb: 3, Len: 2, icd_version: C, Err: 0
RESET ACK

/> getparam 0x13
/> setreceiverid 10
/> getparam 0x13
/> Unit ID: 10, Client ID: 0, ID: 0x40, Nb: 5, Len: 5, icd_version: C, Err: 0
ID: 0x13 Value: 10 (Sensor ID)

```

## Example 2

The receiver ID of MD30 is not known. Use receiver ID 255 to check what the ID is. The value can be checked from the response header (**Unit ID**)

```

/> getunitid
/> setreceiverid 255
/> getunitid
/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 9, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

/> setreceiverid 10
/> getunitid
/>
/> setreceiverid 1
/> getunitid
/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 11, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

```

## 5.2.10 Set Delay Command

The **setdelay** command sets the delay between consecutive instructions when executing a script.

The delay is given in milliseconds.

Use the command when executing commands from a script from interface client. Give the command before the script is executed.

```
setdelay 500
```

## 5.2.11 Echo Mode Command

The **echo** command enables the hexadecimal presentation of the command byte data that is sent and received by the interface client. Parameters **on** and **1** enable printing the byte data, and **off** and **0** disable it.

The sent data is printed between **==>** and **<==**.

### Example

```
>/> echo on
>/> getunitid
==>
0xab 0x00 0x01 0x10 0x0c 0x00 0x00 0x87 0xca
<==
>/> 0xab 0x01 0x00 0x10 0x0c 0x0a 0x00 0x43 0x00 0x52 0x32 0x37 0x33 0x30 0x30
0x30 0x31 0x4f 0x66
Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 12, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

>/> echo off
>/> getunitid
>/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 13, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

>/> echo 1
>/> getunitid
==>
0xab 0x00 0x01 0x10 0x0e 0x00 0x00 0xe7 0xa4
<==
>/> 0xab 0x01 0x00 0x10 0x0e 0x0a 0x00 0x43 0x00 0x52 0x32 0x37 0x33 0x30 0x30
0x30 0x31 0x89
0xec0xab 0x01 0x00 0x10 0x0e 0x0a 0x00 0x43 0x00 0x52 0x32 0x37 0x33 0x30 0x30
0x30 0x31 0x89 0xec

Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 14, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

>/> echo 0
>/> getunitid
>/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 15, Len: 10, icd_version: C, Err: 0
Serial number: R2730001
```

## 5.2.12 License Command

The **license** command prints the open source licenses from *license.txt*, 25 lines at a time.

The number of lines to print at a time can be given as a parameter.

- To print the next set of lines, press **ENTER**.
- To stop the printing, type **q** and press **ENTER**.

### Example

The information is printed 10 lines at a time. After the first set of lines the printing stops.

```
/> license 10
This document contains the licenses and notices for open source software used
in this product.
With respect to the open source software listed in this document, if you have
any questions, please contact us at opensourcesw@vaisala.com.

You may request a copy of certain open source code as required by certain
applicable OSS licenses by sending an email to opensourcesw@vaisala.com with
the subject line
"OSS source code request"

PYTHON 3.7.4 PSF LICENCE AGREEMENT FOR PYTHON 3.7.4
=====
q
/>
```

## 6. Scripts

The interface client can execute commands from a text file. Each command is on a line of its own and the commands are read one command at a time. Comment lines start with the character #.

There are 2 ways to execute the scripts:

- With the parameter `-f`
- With the source command

### Example 1

The file `exampleScript.txt` contains a comment line and 3 commands. Use the `-f` parameter to execute the commands from `exampleScript.txt`.

```
# This is a comment line
getunitid
getproductinfo
getunitid
```

The commands are executed and the interface client closes.

```
c:\client> python.exe md30InterfaceClient.py -P COM4 -f exampleScript.txt

Welcome to MD30 interface client. MD30 interface client demonstrates the
features of the MD30 serial interface.

MD30 interface client version 1.0
Interface version C
Copyright (c) Vaisala Oyj. All rights reserved.
Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 0, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

Unit ID: 1, Client ID: 0, ID: 0x11, Nb: 1, Len: 113, icd_version: C, Err: 0
Product Name: MD30, Serial Number: R2730001, SW Version: 1.0.0, MT10 ID:
541226C864A4B1C2, HMP SerialNumber: R1120189
Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 2, Len: 10, icd_version: C, Err: 0
Serial number: R2730001
```

### Example 2

The command `source exampleScript.txt` executes a script containing the commands **getunitid**, **getproductinfo**, and **getunitid**.

```
/> source exampleScript.txt
Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 0, Len: 10, icd_version: C, Err: 0
Serial number: R2730001

Unit ID: 1, Client ID: 0, ID: 0x11, Nb: 1, Len: 113, icd_version: C, Err: 0
Product Name: MD30, Serial Number: R2730001, SW Version: 1.0.0, MT10 ID:
541226C864A4B1C2, HMP
Serial Number: R1120189
/> Unit ID: 1, Client ID: 0, ID: 0x10, Nb: 2, Len: 10, icd_version: C, Err: 0
Serial number: R2730001
```

You can adjust the delay in the interface client in 2 ways when executing a script.

1. The **setdelay** command is intended to be used when scripts are executed from the interface client with the **source** command. The delay must be set before starting script execution. With **setdelay**, the scripts can be kept more compact.
2. The **wait** command can be used in the script and must be given each time a delay is needed.

#### More information

- [Source Command \(page 21\)](#)





# Technical support



Contact Vaisala technical support at [helpdesk@vaisala.com](mailto:helpdesk@vaisala.com). Provide at least the following supporting information as applicable:

- Product name, model, and serial number
- Software/Firmware version
- Name and location of the installation site
- Name and contact information of a technical person who can provide further information on the problem

For more information, see [www.vaisala.com/support](http://www.vaisala.com/support).

# Recycling



Recycle all applicable material.



Follow the statutory regulations for disposing of the product and packaging.





**VAISALA**

[www.vaisala.com](http://www.vaisala.com)

