



CANscript

Python-Scripting Tool for Processing CAN-Messages

Software Manual

to Product C.1110.01



NOTE

The information in this document has been carefully checked and is believed to be entirely reliable. **esd electronics** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. In particular descriptions and technical data specified in this document may not be constituted to be guaranteed product features in any legal sense.

esd electronics reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

All rights to this documentation are reserved by **esd electronics**. Distribution to third parties, and reproduction of this document in any form, whole or in part, are subject to **esd electronics's** written approval.

© 2021 esd electronics gmbh, Hannover

esd electronics gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone: +49-511-372 98-0
Fax: +49-511-372 98-68
E-Mail: info@esd.eu
Internet: www.esd.eu



This manual contains important information and instructions on safe and efficient handling of CANscript. Carefully read this manual before commencing any work and follow the instructions.

The manual is a product component, please retain it for future use.

Trademark Notices

CANopen® is a registered community trademark of CAN in Automation e.V.
Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.

All other trademarks, product names, company names or company logos used in this manual are reserved by their respective owners.

Document file:	I:\Texte\Doku\MANUALS\PROGRAM\CAN\CAN-Tools\CANscript\Englisch\CANscript_Software_en_14.odt
Date of print:	2021-12-01
Document type number:	DOC0800

Document History

The changes in the document listed below affect changes in the software as well as changes in the description of the facts, only.

Rev.	Chapter	Changes versus previous version	Date
1.2	-	Description updated to match CANscript software version 2.0.3.0. Screenshots taken from Window 7 operating system.	2015-02-13
1.3	1.1.2	Path of example source scripts updated	2019-01-16
	2.	Figures new	
	3.	Step 3. of Quick Start revised (path updated)	
1.4	1.1.2	Revised Python installation description and added some editorial changes.	2021-11-30
	-	Editorial changes	2021-12-01

Technical details are subject to change without further notice.

Table of contents

1. Introduction.....	6
1.1 Installation and Program Call.....	6
1.1.1 CANscript Installation with CAN SDK.....	6
1.1.2 Python Installation and Setup.....	7
2. Functions of the User-Interface.....	8
2.1 Display of the CANscript Program Window.....	8
2.2 Menu Bar.....	9
2.2.1 Opening a Python Script.....	9
2.2.2 Closing a Python Script.....	9
2.2.3 Selecting the Editor and Python Default Paths.....	10
2.2.4 Closing the Program CANscript.....	10
2.3 Display Window.....	11
2.3.1 Editing a Python Script.....	12
2.3.2 Clearing the Display Window.....	12
2.3.3 Closing the Python Script.....	12
2.3.4 Starting / Stopping a Python Script.....	13
3. Quick Start.....	14
4. Order Information.....	15

Typographical Conventions

Throughout this design specification the following typographical conventions are used to distinguish technical terms.

Convention	Example
File and path names	<code>/dev/null</code> or <code><stdio.h></code>
Menu and button names	Start
Programming constants	NULL
Variable names	<i>Count</i>

1. Introduction

This document describes the program **CANscript**. CANscript is a Python scripting tool to handle CAN messages. Via CANscript Python scripts can be executed under a standard user interface.

The program offers:

- execution and management of Python scripts
- allows quick and easy creation of user-defined powerful CAN applications
- supporting the esd NTCAN API in combination with esd CAN hardware interfaces
- helpful run-time status message
- supporting the output of user-defined text messages and data in the status window.

1.1 Installation and Program Call

1.1.1 CANscript Installation with CAN SDK

The tool CANscript is contained in the CAN SDK (Software Development Kit), which is distributed with the **esd-CAN-CD** or can be downloaded from the esd-homepage (www.esd.eu). At the installation of the SDK the program CANscript is automatically installed.

Start the SDK installation file `can_sdk\setup.exe` on the **esd-CAN-CD** and carry out the installation.

If not otherwise defined at the installation, after successful installation the program CANscript can be started under Windows® by selecting the menu items **Start / Program files / esd / CAN SDK / CANscript**.

1.1.2 Python Installation and Setup

In the following text `PYTHONPATH` refers to the installation directory of Python and `CAN_SDK_PATH` to the installation directory of the CAN SDK.

1. Executing the scripts requires an installed Python 3.x version. If this prerequisite is already met you can skip this step. The installer for a Python 3.x version can be found on the **esd-CAN-CD** in the directory `/Redist/Python/`. Alternatively you can download the Python installer from www.python.org. In either case run the installer and follow the instructions.
2. Using the NTCAN-API with Python requires the installation of additional components. The installation process for these components is described in detail in the file `CAN_SDK_PATH\lib\python\readme.txt`.

Note: The required components might support only a specific Python version. If the components are incompatible to the Python version you want to use please contact **esd**.

3. In order to use the NTCAN-API write the line `import ntcn` in your Python modules.

Some example scripts can be found under:

`\Users\\Documents\ESD\CANscript\scripts`

You can also open the sub-directory `\ESD\` with a click on *Documents* (quick access) in your dialogue window.

`canopen.py` - contains CANopen® functions
`canopen2.py` - contains extended CANopen® functions for SDO support
`util.py` - contains miscellaneous auxiliary functions
`main.py` - contains examples for the use of the `canopen.py`-library

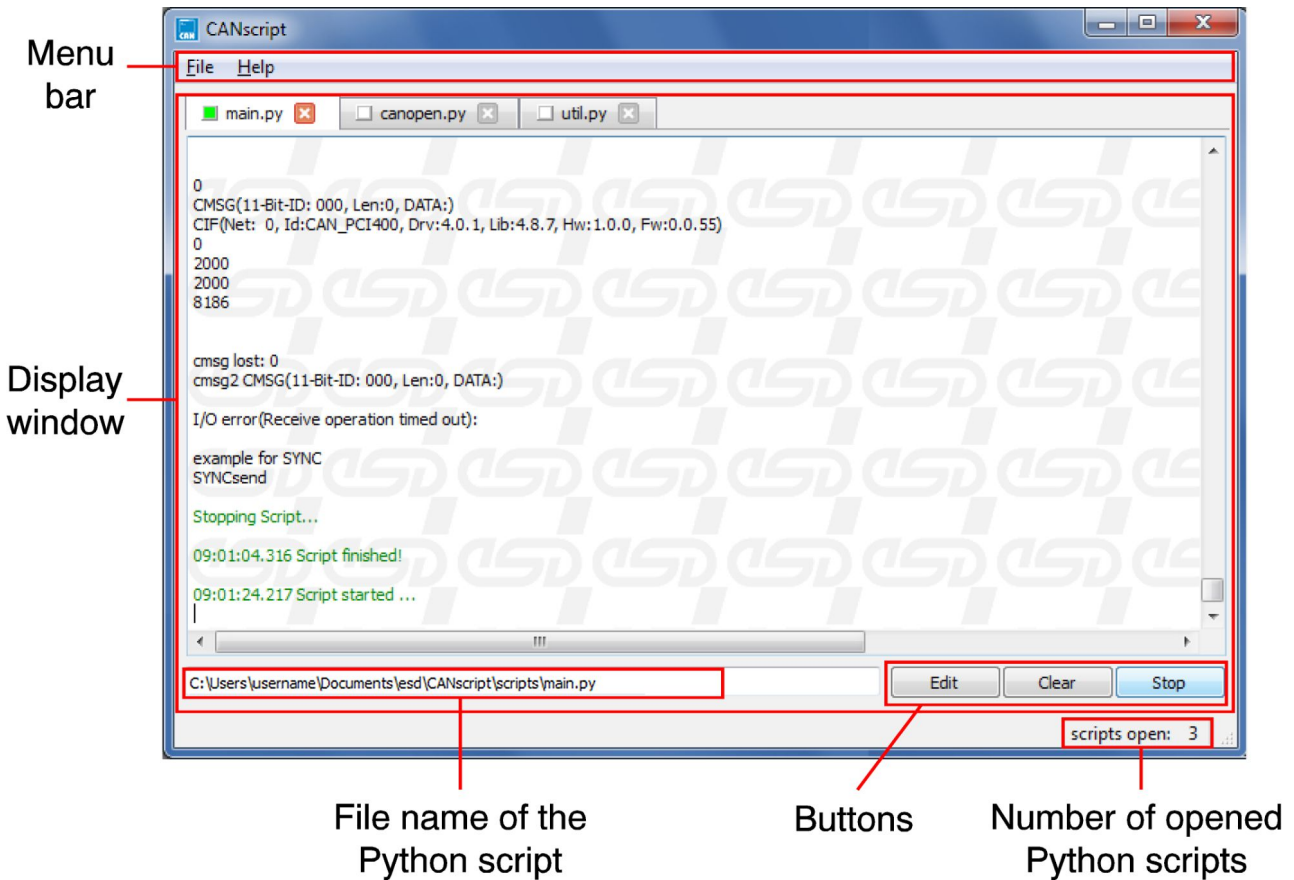
To edit the scripts we recommend an editor with Python syntax support, such as:

- Visual Studio Code <https://code.visualstudio.com/>
- Notepad++ <http://notepad-plus-plus.org>
- Programmers Notepad 2 <http://www.pnotepad.org>
- SciTE <http://www.scintilla.org>

These editors can be downloaded from the homepages under the URLs mentioned above and installed on your computer.

2. Functions of the User-Interface

2.1 Display of the CANscript Program Window

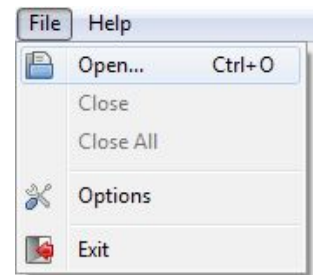


The program window is divided into the following fields:

- Menu bar** Contains the main menu. It includes commands for opening and closing Python scripts and options for managing and monitoring them.
- Display window** The display window is divided into registers. Each register shows the respective output of a Python script.
- Buttons** Via the buttons individual functions for the display and about the operation of the script can be called.
- File name of the Python script** Name and path of the current Python script, displayed in the upper register.
- Number of opened Python scripts** Number of Python scripts opened and represented as register.

2.2 Menu Bar

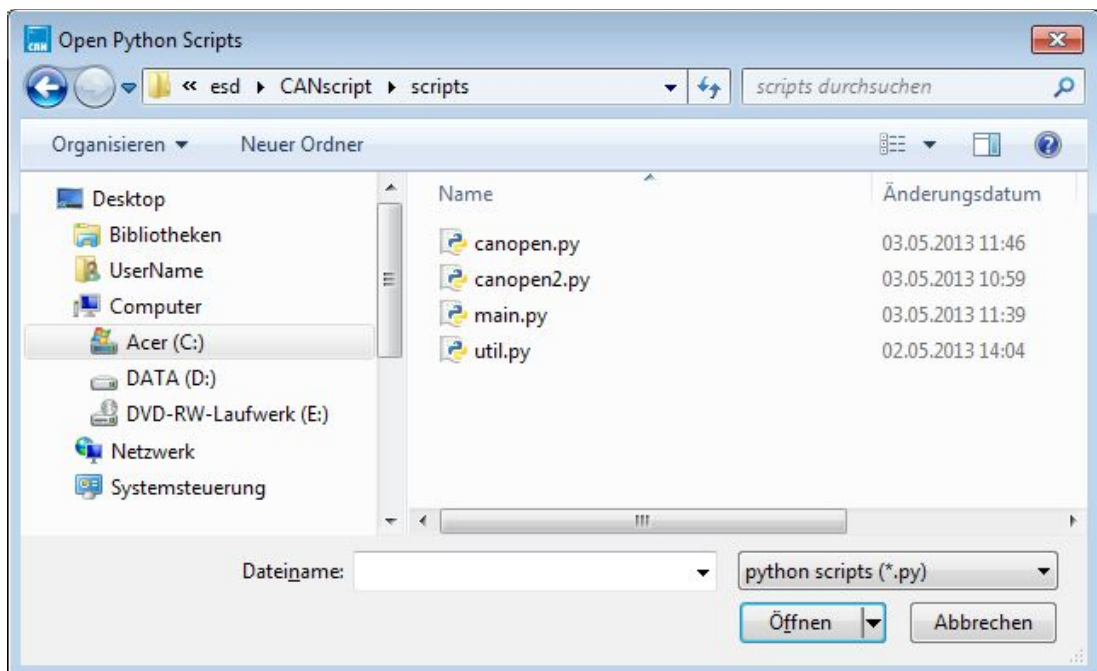
The menu bar includes the menu items **File** and **Help**.



2.2.1 Opening a Python Script

By selecting the menu item **File / Open...** a Python script can be opened.

Open... Opens a Python script
 For this the following dialogue box **Open Python Scripts** is opened.
 An existing Python script can also be opened via the key combination **Ctrl+O**.



Here you can select an existing Python script. The scripts have got the extension ***.py**.

With **Open** you select the script. Opened scripts are shown in a register in the display window (see page 11). Opening several Python scripts will open correspondingly many registers in the display window.

Via **Cancel** you can leave the selection window without opening a script.

2.2.2 Closing a Python Script

By selecting the menu items **File / Close** and **Close all** Python scripts can be closed.

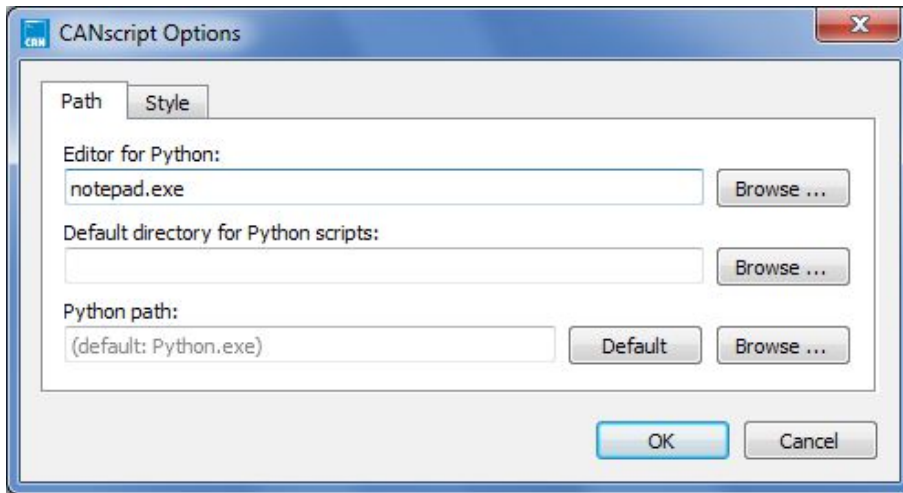
Close Closes the script currently represented in the display window. If the selected script is still operating, you will be asked for confirmation before it is closed.

Close All Closes all opened Python scripts.

2.2.3 Selecting the Editor and Python Default Paths

By selecting the menu item **File / Options** an editor and Python pathes can be set.

Options In this selection box the editor and the path of the Python program as well as the path of the Python scripts can be set.



Editor for Python:

In order to set the editor enter the editor's file and path name. You can also choose the file directly via the browser.

Default directory for Python Scripts:

Here the default directory path of the Python scripts can be entered. The directory path can either be entered directly or selected via the browser.

Python path:

Here the directory path of the Python program can be entered. A click to the **Default** button switches the directory path to the value defined in the Windows registry.

Confirm the entries with **OK**, or choose **Cancel** to close the selection window without accepting the changes.

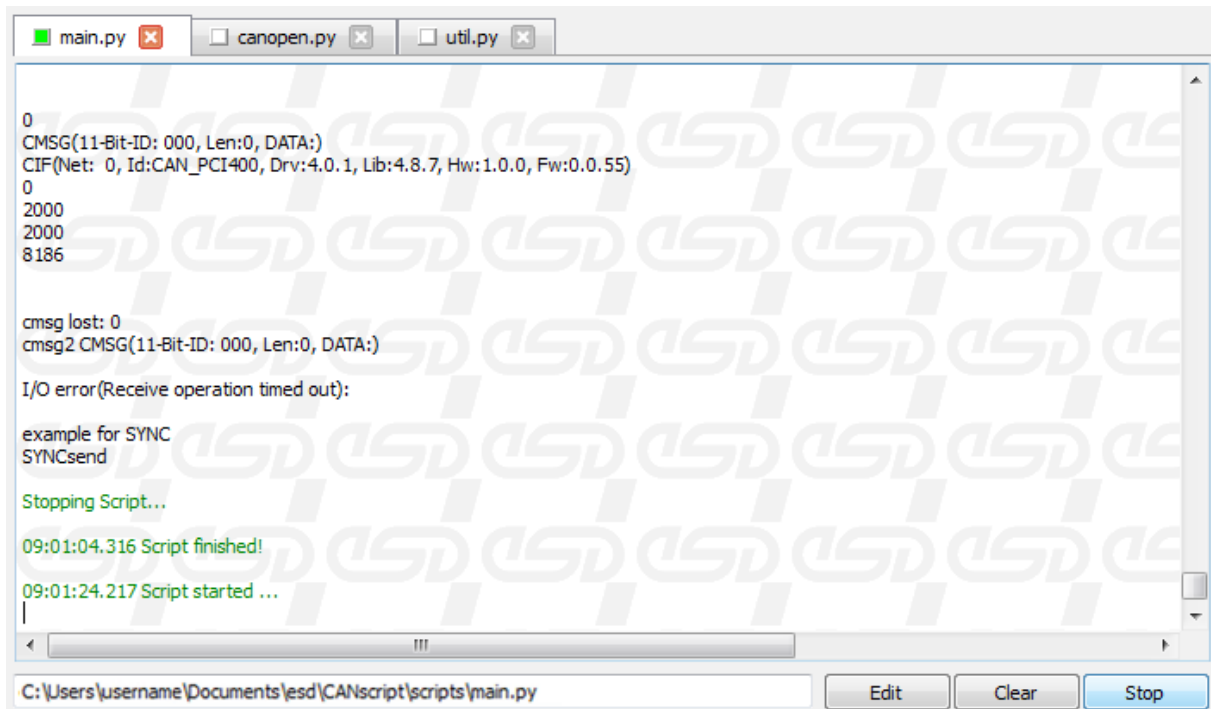
2.2.4 Closing the Program CANscript

By selecting the menu item **File / Exit** you can close the program CANscript.

Exit Via **Exit** you close the program CANscript.

2.3 Display Window

The display window is divided into registers. Each register contains the outputs of a Python script. After opening a script via the menu items **File / Open** the register of the script appears in the display window.



In addition to the file name of the script the tab contains a status box. This represents the status of the Python script. Python scripts which have not yet been started are represented by a white status box and the window does not contain further entries, yet.

Colour	Meaning of the colour of the status box of the tab
white	script opened (not yet executed)
green	script operating
red	script was stopped according to errors
yellow	script was stopped manually or has run through to the end
green/red	script is running, but warnings or errors occurred during execution
yellow/red	script was stopped manually or has run through to the end and warnings or errors occurred

If several Python scripts are open, the registers are overlaid. You can select the Python scripts by means of the tabs.

In the lower left corner of the display window the complete path and file name of the according Python script is shown.

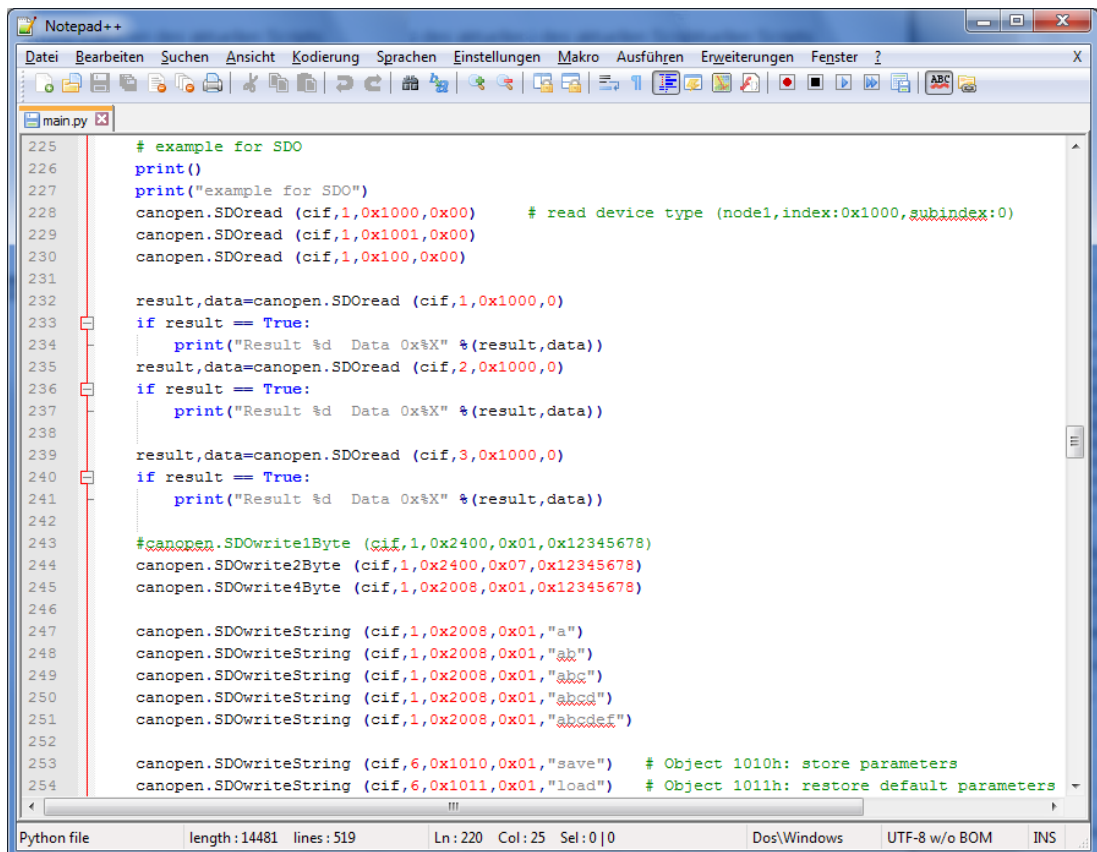
The buttons for starting/stopping and editing the current script are located in the right corner.

2.3.1 Editing a Python Script

Edit

By clicking the **Edit** button the current Python script is opened in the window of the editor selected under **Options**. The design of the editor window depends on the used editor.

The following figure shows the Python script `main.py` opened with the editor *Notepad++* as an example.



```
225 # example for SDO
226 print()
227 print("example for SDO")
228 canopen.SDOread (cif,1,0x1000,0x00) # read device type (node1,index:0x1000,subindex:0)
229 canopen.SDOread (cif,1,0x1001,0x00)
230 canopen.SDOread (cif,1,0x100,0x00)
231
232 result,data=canopen.SDOread (cif,1,0x1000,0)
233 if result == True:
234     print("Result %d Data 0x%X" %(result,data))
235 result,data=canopen.SDOread (cif,2,0x1000,0)
236 if result == True:
237     print("Result %d Data 0x%X" %(result,data))
238
239 result,data=canopen.SDOread (cif,3,0x1000,0)
240 if result == True:
241     print("Result %d Data 0x%X" %(result,data))
242
243 #canopen.SDOwrite1Byte (cif,1,0x2400,0x01,0x12345678)
244 canopen.SDOwrite2Byte (cif,1,0x2400,0x07,0x12345678)
245 canopen.SDOwrite4Byte (cif,1,0x2008,0x01,0x12345678)
246
247 canopen.SDOwriteString (cif,1,0x2008,0x01,"a")
248 canopen.SDOwriteString (cif,1,0x2008,0x01,"ab")
249 canopen.SDOwriteString (cif,1,0x2008,0x01,"abc")
250 canopen.SDOwriteString (cif,1,0x2008,0x01,"abcd")
251 canopen.SDOwriteString (cif,1,0x2008,0x01,"abcde")
252
253 canopen.SDOwriteString (cif,6,0x1010,0x01,"save") # Object 1010h: store parameters
254 canopen.SDOwriteString (cif,6,0x1011,0x01,"load") # Object 1011h: restore default parameters
```

2.3.2 Clearing the Display Window

Clear

By clicking the **Clear** button the current output of the selected Python script is deleted and the according window is cleared.

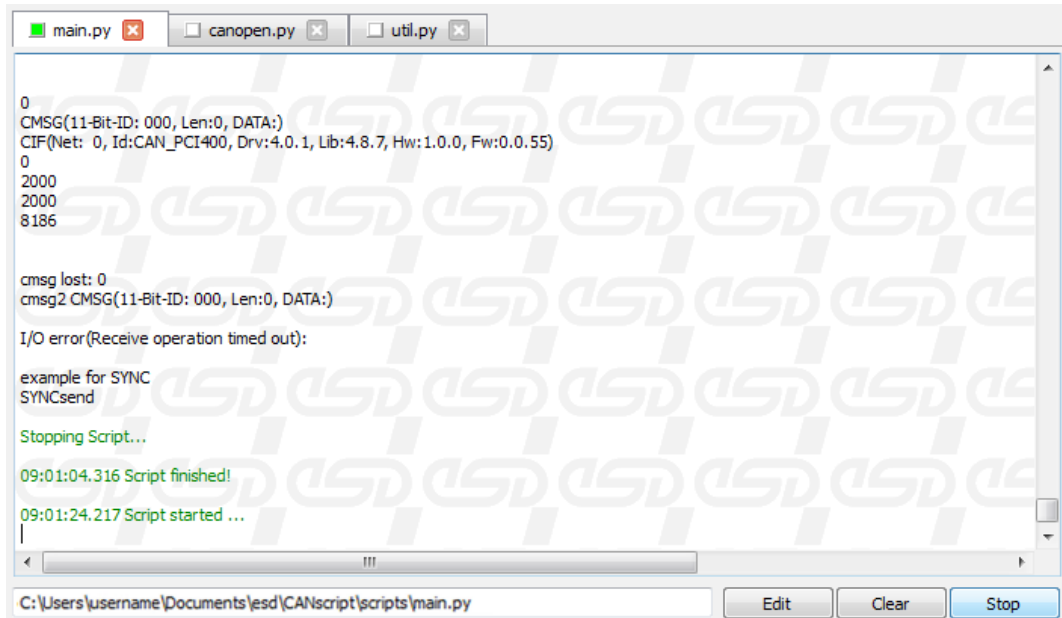
2.3.3 Closing the Python Script

Close

Clicking the **Close** button closes the selected Python script.

2.3.4 Starting / Stopping a Python Script

Start/Stop Via the **Start** button the Python script displayed in the current register is started. After the program has been started it can be stopped via the same button with **Stop**.



The start time and the outputs of the Python script are displayed. If the Python script is stopped, the script's end time is displayed. Times are displayed in green letters.

If an error occurs during the operation of the Python script, the protocol displays an error message in red letters.

3. Quick Start

Please follow the steps below for an easy program application:

1. Install the CAN SDK on your computer. CANscript is installed with the CAN SDK as one of the CAN Tools and follow the additional steps to install Python and the NTCAN-API Python bindings as described in chapter 1.1.2.
2. If you have chosen the default installation directories after successful installation the program CANscript can be started under Windows by selecting the menu items **Start / Program files / esd / CAN SDK / CANscript**.
3. To open a Python script select the menu item **Open** in the main menu under **File**. With the dialogue box **Open Python Scripts** you can now select a Python script in the following directory:

\Users\\Documents\ESD\CANscript\scripts

You can also open the sub-directory **\ESD** with a click on *Documents* (quick access) in your dialogue window.

The selected script will be displayed in the current tab of the display window.

4. Start running the Python script via the **Start** button.
5. The outputs of the Python script will be represented in the display window.
6. Stop the playback of data via the **Stop** button.

4. Order Information

Type	Properties	Order No.
CANscript	Python scripting tool CANscript is contained in the scope of delivery of the CAN-SDK and can be downloaded from our website www.esd.eu .	C.1110.01

Table 1: Order information

PDF Manuals

Manuals are available in English and usually in German as well. For availability of manuals see table below.

Please download the manuals as PDF documents from our esd website www.esd.eu for free.

Manuals		Order No.
CANscript-ME	Hardware manual in English (this manual)	C.1110.21
CANscript-MD	Hardware manual in German	C.1110.20

Table 2: Available manuals

Printed Manuals

If you need a printout of the manual additionally, please contact our sales team: sales@esd.eu for a quotation. Printed manuals may be ordered for a fee.