CANStream

User guide

2014

Administrator

Cobalt Solutions

11/7/2014

CANStream user guide

Contents

[Introduction 6](#_Toc405388180)

[Getting started 7](#_Toc405388181)

[Computer requirements 7](#_Toc405388182)

[License agreement 8](#_Toc405388183)

[General overview 11](#_Toc405388184)

[Installation and registration 12](#_Toc405388185)

[Installation 12](#_Toc405388186)

[Registration 12](#_Toc405388187)

[PEAK USB CAN adapter driver 12](#_Toc405388188)

[Main interface 13](#_Toc405388189)

[CAN Controller panel 14](#_Toc405388190)

[CAN controller modes 15](#_Toc405388191)

[CAN controller panel tool bar 16](#_Toc405388192)

[Menu strip 17](#_Toc405388193)

[CAN Configuration 17](#_Toc405388194)

[Cycle 17](#_Toc405388195)

[Recording 18](#_Toc405388196)

[Tools 18](#_Toc405388197)

[View 19](#_Toc405388198)

[Help 19](#_Toc405388199)

[Status strip 20](#_Toc405388200)

[CAN Frames configuration 21](#_Toc405388201)

[Single bus configuration 21](#_Toc405388202)

[Bus properties 21](#_Toc405388203)

[CAN message properties 21](#_Toc405388204)

[CAN signal properties 21](#_Toc405388205)

[Virtual CAN signal 21](#_Toc405388206)

[Vector CAN data base (dbc) import 21](#_Toc405388207)

[Multiple buses configuration 21](#_Toc405388208)

[CAN Bus information 21](#_Toc405388209)

[Configuration import & export 21](#_Toc405388210)

[Manual mode 22](#_Toc405388211)

[Manual control configuration 22](#_Toc405388212)

[Data Tx 23](#_Toc405388213)

[Engineering format data Tx 24](#_Toc405388214)

[Raw format data Tx 25](#_Toc405388215)

[Data Rx 27](#_Toc405388216)

[Engineering format data Rx 27](#_Toc405388217)

[Raw format data Rx 27](#_Toc405388218)

[Rx data graphic 28](#_Toc405388219)

[Cycle mode 30](#_Toc405388220)

[Cycle playing 30](#_Toc405388221)

[Cycle RX data panel 32](#_Toc405388222)

[Cycle starting and ending points setting 33](#_Toc405388223)

[Cycle loop number setting 35](#_Toc405388224)

[Virtual CAN signals in cycle 35](#_Toc405388225)

[Cycle creation 37](#_Toc405388226)

[Easy cycle creation 38](#_Toc405388227)

[Cycle data association 41](#_Toc405388228)

[Advanced cycle creation 45](#_Toc405388229)

[Record replay 51](#_Toc405388230)

[CAN controller appearance options 53](#_Toc405388231)

[Control layout 53](#_Toc405388232)

[Manual control layout 53](#_Toc405388233)

[Cycle control layout 54](#_Toc405388234)

[Grids contents control 55](#_Toc405388235)

[Grid rows 55](#_Toc405388236)

[Grid columns 55](#_Toc405388237)

[Data logging 61](#_Toc405388238)

[Data logging start & stop 61](#_Toc405388239)

[Automatic recording 62](#_Toc405388240)

[Data file formats 62](#_Toc405388241)

[Peak PCAN trace file 63](#_Toc405388242)

[Record data file 64](#_Toc405388243)

[Logging file conversion 65](#_Toc405388244)

[Manual conversion 65](#_Toc405388245)

[Automatic conversion 66](#_Toc405388246)

[Conversion process control 67](#_Toc405388247)

[Data file storage structure 67](#_Toc405388248)

[Recording event & Recording session 68](#_Toc405388249)

[Recording event creation & edition 69](#_Toc405388250)

[Recording event selection 71](#_Toc405388251)

[Recording session creation & edition 72](#_Toc405388252)

[Recording session selection 73](#_Toc405388253)

[User information 75](#_Toc405388254)

[Data analysis 77](#_Toc405388255)

[Analysis window 78](#_Toc405388256)

[Data viewer page creation 79](#_Toc405388257)

[Data viewer Toolbar 81](#_Toc405388258)

[Data set information 82](#_Toc405388259)

[Data browser 86](#_Toc405388260)

[Root folder tree 87](#_Toc405388261)

[Events & sessions list 87](#_Toc405388262)

[Logging data list 89](#_Toc405388263)

[Toolbar 91](#_Toc405388264)

[Graphic window 92](#_Toc405388265)

[Presentation 92](#_Toc405388266)

[Channels list 92](#_Toc405388267)

[Main graphic cursor 92](#_Toc405388268)

[Reference cursor 92](#_Toc405388269)

[Zoom 92](#_Toc405388270)

[Legend 92](#_Toc405388271)

[Statistics 92](#_Toc405388272)

[Graphic snapshot 92](#_Toc405388273)

[Graphic printing 92](#_Toc405388274)

[Graphic window configuration 92](#_Toc405388275)

[Graphic series layout 92](#_Toc405388276)

[Main graphic series properties 92](#_Toc405388277)

[General graphic window properties 92](#_Toc405388278)

[Graphic grids properties 92](#_Toc405388279)

[Graphic abscise properties 92](#_Toc405388280)

[Cursors graphical properties 92](#_Toc405388281)

[Graphic series detailed properties 92](#_Toc405388282)

[Series general properties 92](#_Toc405388283)

[Series value formats 92](#_Toc405388284)

[Series Y axis 92](#_Toc405388285)

[Series custom grid 92](#_Toc405388286)

[Series reference lines 92](#_Toc405388287)

[Virtual channels 93](#_Toc405388288)

[Virtual channels edition 93](#_Toc405388289)

[Virtual channels library 93](#_Toc405388290)

[Virtual channel properties 93](#_Toc405388291)

[Virtual channel testing 93](#_Toc405388292)

[Virtual channel elements objects 93](#_Toc405388293)

[Virtual channel built-in functions 93](#_Toc405388294)

[Virtual channels use 96](#_Toc405388295)

[Use of virtual channels in the manual data Rx 96](#_Toc405388296)

[Use of virtual channels for cycle creation 96](#_Toc405388297)

[Virtual CAN signals 96](#_Toc405388298)

[Presentation 96](#_Toc405388299)

[Use of virtual CAN signals in the manual data Tx 97](#_Toc405388300)

[Use of virtual CAN signals in cycle player 97](#_Toc405388301)

[Built-in signals 98](#_Toc405388302)

[Built-in signals edition 98](#_Toc405388303)

[Built-in signals library 100](#_Toc405388304)

[Built-in signals properties 101](#_Toc405388305)

[Built-in signals types 102](#_Toc405388306)

[Built-in signals use 108](#_Toc405388307)

# Introduction

First of all, thanks for using CANStream!

This document is dedicated to the countless lost hours of both people having to search for simple answers, and people having to answer questions that shouldn't have to be asked in the first place. Documentation is always hard to get, good documentation nearly impossible.

This markup is based in the default appearance of Microsoft Windows. This can be different on your own system, but the names (text) should always the same. This is especially important on the screenshots. They are only for reference; your appearance of CANStream can be quite different.

Also, when talking about the left and right mouse buttons, the logical left and right is meant. These are the same as the physical left and right mouse buttons in the case of a right-handed setup. However, if you have a left-handed setup you will probably have the buttons swapped so act accordingly. This means in general you do what you normally do for most actions; CANStream mostly follows the standard Windows application behavior.

Likewise, the layout is discussed as it is displayed in a left-to-right order.

# Getting started

## Computer requirements

|  |  |
| --- | --- |
| Hardware | |
| CPU | Pentium 4 1.2 GHz |
| RAM | 1 GB |
| Hard drive | 25 MB free space |
| USB port | 1 minimum |
| Operating system | |
| Any of those Windows version | 2000, XP (32/64 bits), Vista (32/64 bits), Seven (32/64 bits), 8 (32/64 bits) |
| .NET Framework | 4.0 |

## License agreement

**DEFINITIONS**

“Software” means CANStream and its components. "User" means the Software customer or end user.

**OWNERSHIP**

Cobalt Solutions is the owner of the Software and of its copyright.

**LICENSE**

The Software ownership is not transferred to the User. Cobalt Solutions grant the User a simple, not-exclusive license to use the Software only on limited numbers of workstations as defined in the appendix to the present agreement, for a period of time depending of type of license. During this period the User is entitled to receive services as described in appendix to the present agreement. When the license will expire, licensed features will be automatically disabled. If the license is not renewed the User may continue to use the Software under terms and conditions set forth in this "LICENSE AGREEMENT", but is not entitled to receive any services of any kind. If the license is not renewed within four (4) months from the expiry date, then the renewal option will be lost and the User needs to order a brand new license.

**COPY RESTRICTIONS**

The User is allowed to make one (1) backup or archive copy only for reserve. All copyright notices and any other proprietary legends on the original Software shall be reproduced on such copy. Both the original and the reserve copy of the Software are subject to the restrictions set forth in this license agreement and they shall be destroyed in case of license termination.

**RESTRICTIONS**

The User shall not rent or loan, distribute or sublicense the original or the backup copy of the Software and the manuals. User may not modify, translate, edit, decompile, reverse engineer, disassemble the Software or create derivative utilities based upon it. User may not edit the structure of the Software, nor ask to any other person to do so in order to modify its structure, dimensions, etc. THE USER SHALL NOT COPY THE MANUALS OR THE RELATED DOCUMENTATION, unless Cobalt Solutions gives written authorization. The license will immediately terminate without notice if the User fails to comply with any provision set forth herein.

**DISCLAIMER OF RESPONSABILITY**

The Software is provided "AS IS" and without warranty of any kind.

**LIMITED WARRANTY**

Cobalt Solutions reserves the right to grant the User the license for the Software, and he will replace the media on which the Software is furnished (disks and/or CD and/or password keys) which prove to be defective in material or workmanship under normal use. This warranty is good during a period of thirty (30) days starting from the date of purchase, and it does not apply to damages or problems resulting from misuse, abuse, accident or improper installation.

DISCLAIMER OF WARRANTY AND LIMITATION OF LIABILITY. EXCEPT AS SET FORTH ABOVE, NO OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OR CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE APPLICABLE TO THIS PRODUCT. COBALT SOLUTIONS SHALL NOT BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, EXPRESSLY OR NOT SET FORTH HEREIN.

**THIRD PARTY RESPONSIBILITY**

The Software copyright is ownership of Cobalt Solutions. The User may be considered directly liable by Cobalt Solutions for any action concerning the use of the Software, made directly or indirectly, that does not comply with this "LICENSE AGREEMENT".

**USER DUTIES AND MISUSE**

The User agrees to make any effort in order to assure that his employees, consultants or other people that can access to the Software comply with the terms set forth in this "LICENSE AGREEMENT", including the prohibition to authorize anyone to use parts of the Software in order to identify its source code. The User shall immediately make any effort to stop any misuse that comes to his knowledge.

**LICENCE TERMINATION**

Cobalt Solutions reserves the right to terminate the license at any time if the User fails to comply with any provision of this "LICENSE AGREEMENT". In this case, the User must immediately destroy the Software and return the media and all the copies to Cobalt Solutions.

**CONTROLLING LAWS**

This agreement is governed by the laws of France, also for what concerns patent and copyright. In case of judicial acts or proceedings consequent to a demonstrated failure to comply with the conditions set forth in this "LICENSE AGREEMENT"., all the legal costs will be at losing party expense. This agreement regulates any Software update and/or upgrade, and constitutes the entire agreement between the parties with respect to the use of the Software and the related documentation. Any dispute falls under the competence of the Law Court of Paris (France).

***APPENDIX TO END USER LICENCE AGREEMENT***

**NUMBER OF INSTALLATIONS**

Number of workstations on which the Software may be installed on CANStream: a finite number of installations defined by User at order placement.

**IMPORTANT NOTE**

This covers only installations of Cobalt Solutions proprietary software. Any license fees for third party software are not included in the Cobalt Solutions license fee.

**SERVICES**

For one year from the order of a new license or renewal of an existing one the User is entitled to receive the following services

1. supply of latest CANStream software release at the time of new license purchase or renewal;
2. User is entitled to request, under payment, customization of CANStream software;
3. Supply of upgraded features (at the sole discretion of MMM) and debugged Wintax4 software releases, if any, during a valid license period;

# General overview

CANStream is a .NET application intended to be used for CAN (Controller Area Network) bus communication development, testing and validation. CANStream allows to extensively test any system using CAN communication by sending and receiving CAN frames. Thanks to its powerful built-in mathematical expression evaluator, CANStream can also behave as a real control system feeding back the test device or commanding a third party device with context sensible data. Extended data logging and data analysis features of CANStream provide a comprehensive solution for testing, testing results storage and analysis.

CANStream extensively uses the PCAN-Basic API developed by PEAK System for the PCAN-USB adapter

For more information about PEAK System: <http://www.peak-system.com>

You will therefore need to have PCAN-USB adapter and at least one free USB port available to make a complete use of CANStream.

CANStream can manage up to eight PCAN-USB adapters at the same time, thus you will need eight USB ports available.

As any .NET application, CANStream needs a Microsoft .NET framework to operate. You might need to install or upgrade your .NET framework. If so, please visit <http://www.microsoft.com/net> to download the latest version of the Microsoft .NET framework.

# Installation and registration

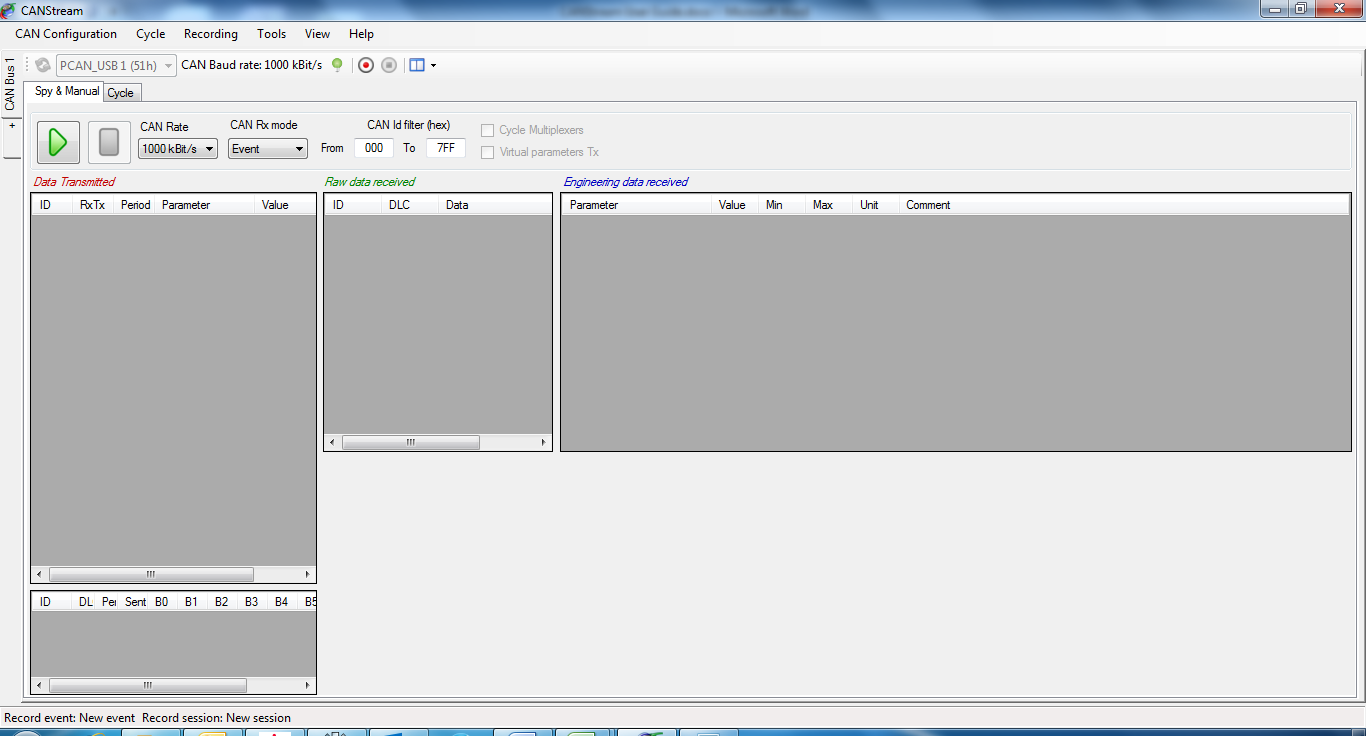
## Installation

## Registration

## PEAK USB CAN adapter driver

# Main interface

At startup CANStream comes as the picture bellow.



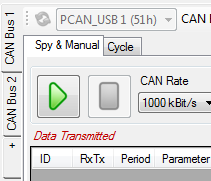
The main CANStream interface is composed of different elements

* CAN Controller panels using the most of the space
* The menu strip, on the top, regrouping all commands and tools accesses
* A status bar on the bottom showing different information such like the current record event and session, the current CAN configuration name and error flags

## CAN Controller panel

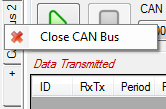
The CAN controller panel is where the CANStream magic happens!

There is one panel for each Peak PCAN-USB adapter you want to use. By default only one panel is opened, simply click the ‘+’ tab to open a new panel and use an additional PCAN-USB adapter.



CANStream can handle up to eight PCAN-USB adapters, so it is possible to open a maximum of eight panels.

To close a panel and release its PCAN-USB adapter, stop the current panel operation (Manual control or Cycle player). Then right-click on the panel you want to close, and click the ‘Close CAN Bus’ command.



The CAN controller panel is composed of a tool bar and a multi-tab panel.

### CAN controller modes

A CAN controller has two operation modes, ‘Manual & Spy’ and ‘Cycle’. Each mode has its own tab in the panel.

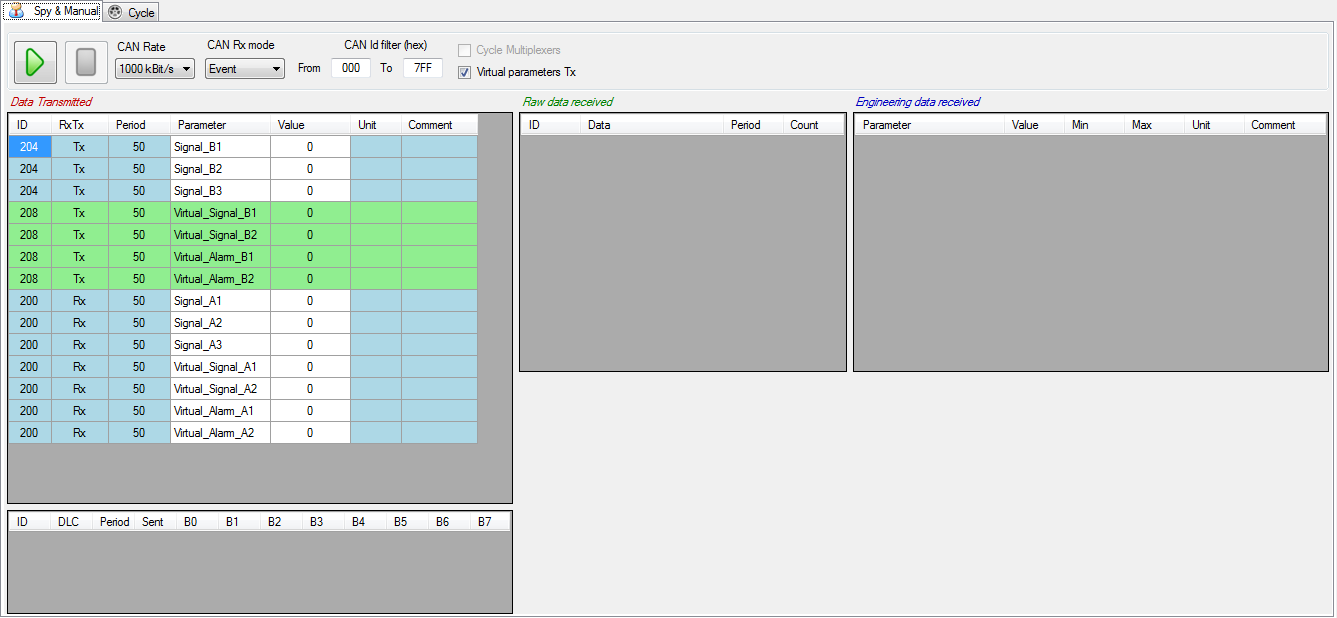


#### Spy & Manual mode

The ‘Spy & Manual’ mode is intended to be used to spy the content of frames (or messages) circulating on the bus.

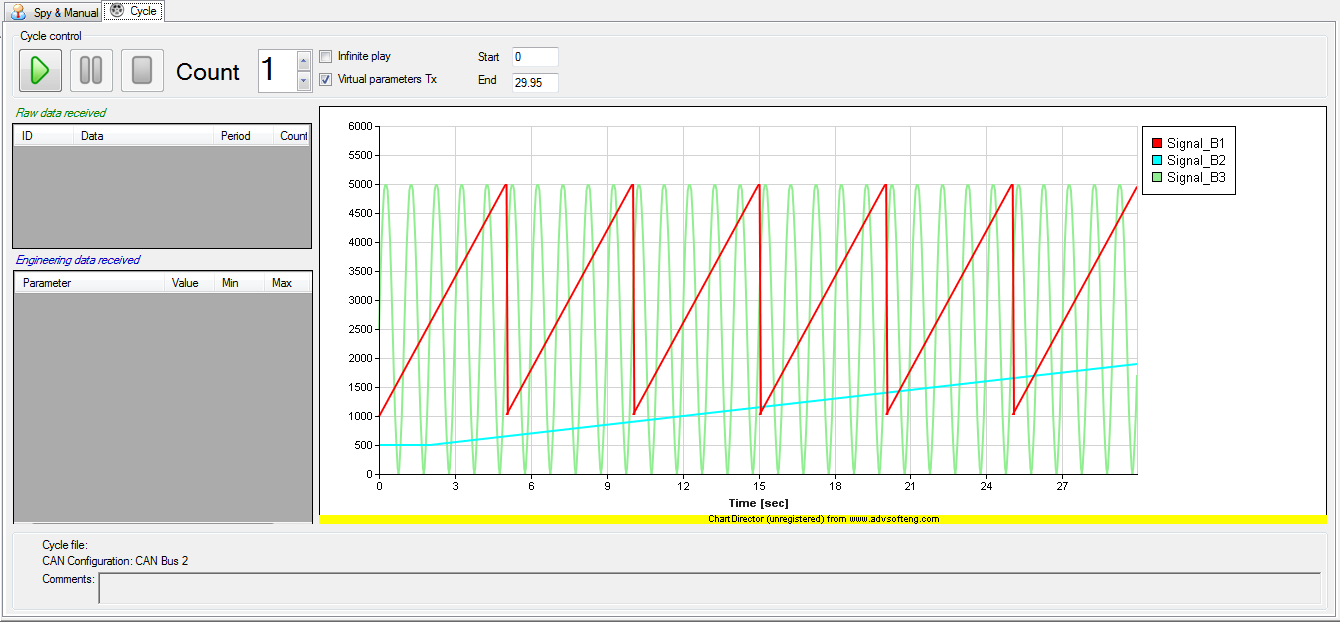
Using this mode, it also possible to send messages containing values set by the user.

Both reception and transmission handle ‘Engineering’ and ‘Raw’ formats of data. ‘Engineering’ means that user can read and write actual physical values of CAN signals while the ‘Raw’ format means raw byte values of the CAN messages.



#### Cycle mode

In ‘Cycle’ mode CANStream is sending a pre-defined sequence of message. This mode allows simulation of real usage conditions in which values have a very high dynamic that couldn’t be replicated by a human.



### CAN controller panel tool bar

The CAN controller panel tool bar contains common commands for both ‘Spy & Manual’ and ‘Cycle’ modes



At the CANStream start up or on new controller opening, if a PCAN-USB adapter is connected and free (not used by another application) CANStream takes the control of this adapter and tool bar is as shown above.

By default the CAN Baud rate is set to 1000 kBit/s, however Baud rate can be changed in the CAN Configuration editor.

The ‘Stop connection’ button  permits to stop the connection and to release the PCAN-USB adapter in order to make it available for another controller panel or another application. Click the ‘Start connection’ button  to restart the connection using the same PCAN-USB adapter.

When the connection stops, ‘Refresh PCAN USB Devices list’ command  and the PCAN USB device list are enabled.



Click the ‘Refresh PCAN USB Devices list’ button  to refresh the ‘PCAN USB Devices list’. Click in the list to select the PCAN-USB device that you want use.



The ‘Start stream recording’ button  starts the data recorder. When it is clicked, the ‘Stop stream recording’ button  becomes enabled allowing the stop of data recording.

Command ‘View’  contains several sub-commands to customize the appearance of the ‘Spy & Manual’ panel. See the ‘Manual control layout’ section for more details.

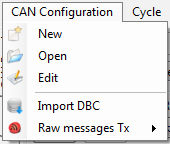
## Menu strip

The menu strip gives access to all CANStream functions and tools.



### CAN Configuration

This menu contains all function related to the CAN configuration of the current controller.



New: Create a new CAN configuration.

Open: Open a CAN configuration file.

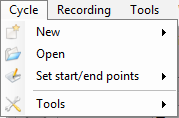
Edit: Edit the current CAN Configuration.

Import DBC: Import a DBC file and use it as current CAN configuration.

Raw messages Tx: Management of transmitted raw messages list.

### Cycle

This menu contains all functions related to the cycle player.



New: Create a new cycle. Check the ‘Cycle creation’ section for more details.

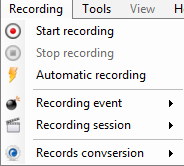
Open: Load a cycle into the cycle player of the current CAN controller.

Set start/end points: Access to cycle starting and ending points control functions. Check the ‘Cycle playing’ section for more details.

Tools: Access to cycle creation tools. Check the ‘Cycle creation’ section for more details.

### Recording

This menu contains all functions related to the data recorder.



Start recording: Start the data recorder.

Stop recording: Stop the data recorder.

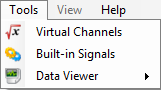
Automatic recording: Enable or disable the automatic recording. Check the ‘Data recording’ section for more details.

Recording event: Access to recording event control functions. Check the ‘Recording event’ for more details.

Recording session: Access to recording session control functions. Check the ‘Recording session’ for more details.

### Tools

This menu contains all CANStream tools access

.

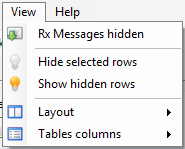
Virtual Channles: Open the virtual channels editor

Built-in Signals: Open the built-in signals editor

Data Viewer: Open the data viewer

### View

This menu contains layout configuration functions for the manual mode of the current CAN controller.



Rx Messages hidden: Hide Rx messages in the transmission panel

Hide selected rows: Hide the selected rows of the active grid

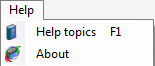
Show selected rows: Show hidden rows of the active grid

Layout: Access to the control layout configuration functions.

Tables columns: Access to the grid columns configuration functions

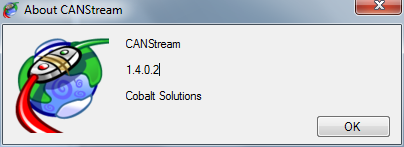
### Help

This menu contains access to the help file and the ‘About’ form



Help topics: Open the CANStream user guide. This function is also accessible by pressing the ‘F1’ key.

About: Open the CANStream ‘About’ form containing different information including the release name and version.



## Status strip

The status bar of CANStream is located at the bottom of the main window. This bar contains different information.



The name of the CAN configuration file currently in use



Names of current data recording event and session



In case of CAN error, name and description of the error detected will also appear in the status bar.



# CAN Frames configuration

## Single bus configuration

### Bus properties

### CAN message properties

#### Message map

### CAN signal properties

### Virtual CAN signal

### Vector CAN data base (dbc) import

## Multiple buses configuration

### CAN Bus information

### Configuration import & export

# Manual mode

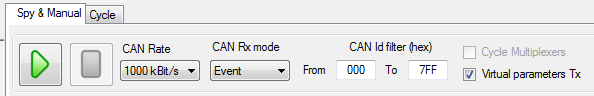
The ‘Spy & Manual’ mode is intended to be used to spy the content of frames (or messages) circulating on the bus.

Using this mode, it also possible to send messages containing values set by the user.

Both reception and transmission handle ‘Engineering’ and ‘Raw’ formats of data. ‘Engineering’ means that user can read and write actual physical values of CAN signals while the ‘Raw’ format means raw byte values of the CAN messages.

## Manual control configuration

The ‘Spy & Manual’ mode is controlled through the tool bar of the ‘Spy & Manual’ panel.



This tool bar contains the following elements

Start button: Start data transmission and reception

Stop button: Stop data transmission and reception

 CAN rate selection: Allows definition of the CAN communication speed.

 Rx mode selection: Allows selection of the message reception mode (event, manual, periodic)

 Rx IDs filter: Allows the definition of the range of CAN message IDs that have to be considered in the reception panel.

 Virtual CAN signal Transmission enabled flag: Enable and disable the transmission of virtual CAN signals

 Multiplexer auto-cycling flag: If the CAN configuration in use contains some multiplexed message, CANStream will automatically cycle the multiplexer value for data transmission.

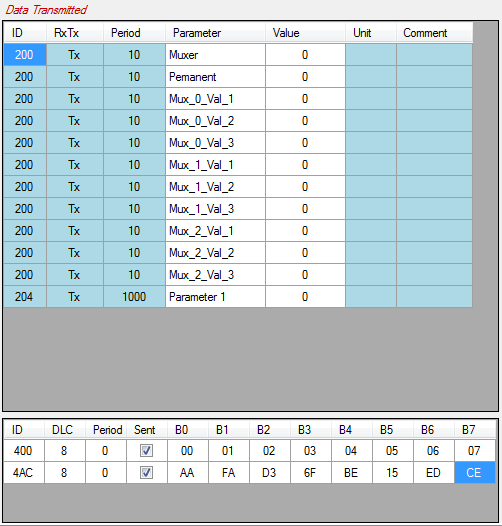
## Data Tx

The ‘Start’ button  of the tool bar will be enabled only if an operational PCAN-USB device has been selected.

Click that button to start the data transmission. While the manual mode is operating, ‘Start’ button  is disabled. Click the ‘Stop’ button  to stop the manual mode operation and consequently the data transmission.

Those commands control both transmission and reception parts of the ‘Spy & Manual’ mode.

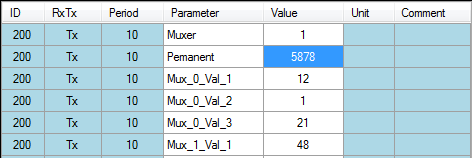
Transmission panel of ‘Spy & Manual’ control is divided in two parts: ‘Engineering’ on the top and ‘Raw’ data transmission at the bottom.



### Engineering format data Tx

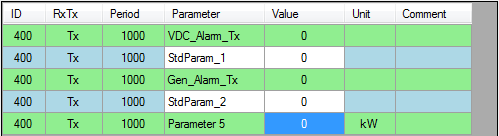
In order to send engineering data over the CAN bus a CAN configuration file has to be loaded. Once loaded, this CAN configuration is displayed in the ‘Engineering data transmission’ grid.

Simply change the content of the ‘Value’ cell to send a new value.



#### Virtual data Tx

Virtual CAN signal appear with a green color into the grid.

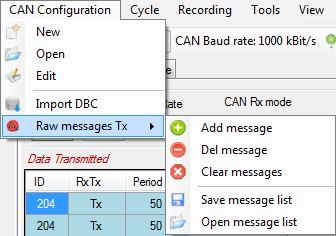


Since the value of a virtual CAN signal is calculated, its value cannot be set by the user.

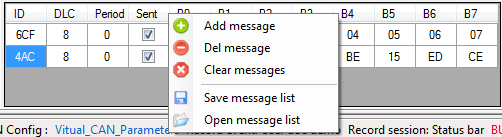
Virtual CAN signal transmission can be turned on and off at any time, checking or unchecking the box 

### Raw format data Tx

To send ‘Raw’ CAN messages over the bus, you first need to define those messages. Click the ‘CAN Configuration\ Raw messages Tx’ menu to get the raw messages management functions.



Alternatively, right click in the raw message grid to open the creation menu.



This contextual menu contains a bunch of commands to manage the raw message list.

Add message: Add a message in the list



Del message: Remove a message from the list



Clear messages: Remove all messages of the list

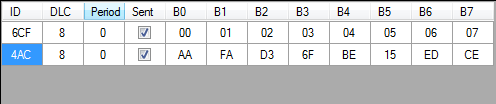


Save: Save the message list in a file



 Open: Open a message list file.

Once messages have been loaded or created it is possible to set their properties in the different cells of the grid



ID: Message identifier

DLC: Message byte length

Period: Message transmission period in millisecond

Sent: Define whether or not the message is sent

B0… B7: Message bytes value

If the property ‘Period’ is set to zero, message is not sent over.

## Data Rx

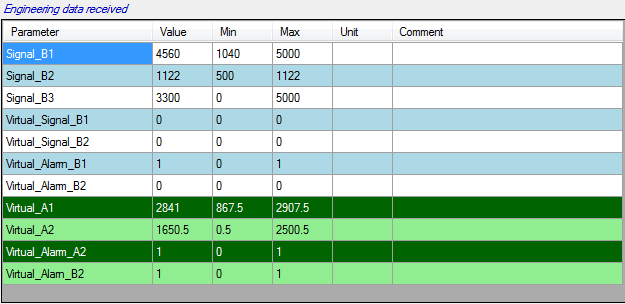
The ‘Start’ button  of the tool bar will be enabled only if an operational PCAN-USB device has been selected.

Click that button to start the data reception. While the manual mode is operating, ‘Start’ button  is disabled. Click the ‘Stop’ button  to stop the manual mode operation and consequently the data reception.

Those commands control both transmission and reception parts of the ‘Spy & Manual’ mode.

### Engineering format data Rx

In order to decode data received over the CAN bus, a CAN configuration file has to be loaded. Then when CAN frames are received they are decoded and signals are shown in the ‘Engineering data received’ grid.

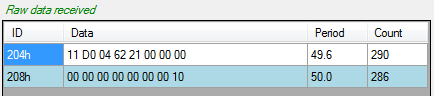


#### Virtual data Rx

If one or more virtual channels libraries are loaded and data received are containing signal used in those virtual channels, virtual channels are automatically computed and their values are shown into the grid with green background color.

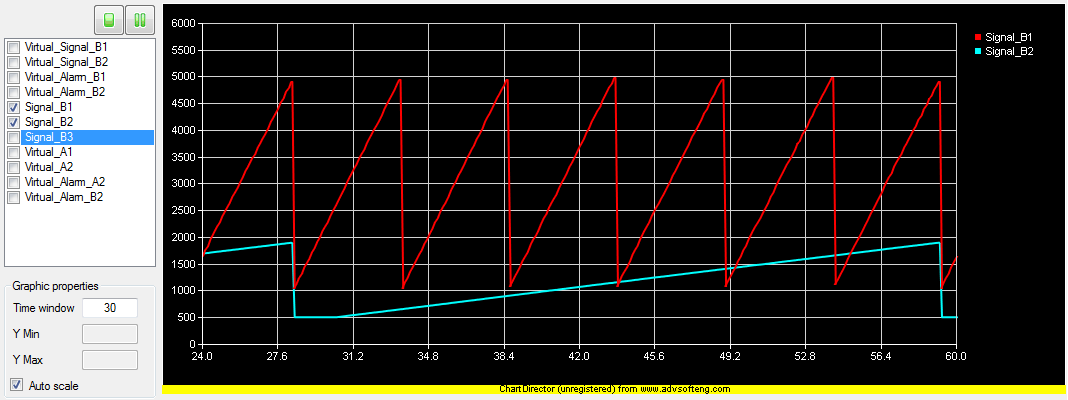
### Raw format data Rx

Received frames raw values (before decoding) are shown into the ‘Raw data received’ grid.

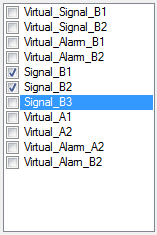


### Rx data graphic

When CAN frames received contains data that can be decoded using the loaded CAN configuration, values of CAN signals received can be traced in real time into a graphic.



The graphic shows CAN parameters values as a function of the time. By default, the graphic is empty, simply select data to trace by checking channels in the channels list.



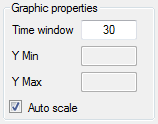
Using the small toolbar on the top of the channel list, it is possible to stop or freeze the graphic.



Button stops the real time graphic plotting. Once clicked this button becomes ‘Start plot’ button . Those two commands actually reset the graphic window so all data plotted will be lost after a ‘Stop/Start’ commands.

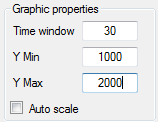
Button  freezes the real time graphic plotting. Once clicked this button becomes ‘Resume’ button . Unlike the ‘Stop’ button, ‘Freeze’ command doesn’t reset the graphic. On graphic resuming, graph will appear as if it never stopped before.

Graphic time window and Y axis scale can be changed through the ‘Graphic properties’ panel.



Field ‘Time window’ define the time span between the beginning and the end of graphic.

If check box ‘Auto scale’ is ticked, all values of all plotted signal are shown in the graphic. Uncheck that box to define the scale of the Y axis. After having typed a value into the ‘Y min’ or ‘Y max’ fields; press ‘Enter’ to update the graphic.



# Cycle mode

The ‘Cycle’ mode of CANStream sends a pre-defined sequence of message. This mode allows simulation of real usage conditions in which values have a very high dynamic that couldn’t be replicated by a human.

The ‘Cycle’ mode is controlled through the tool bar of the ‘Cycle’ panel.



This tool bar contains the following elements

Start button: Starts cycle play

Pause button: Makes a pause in the cycle.

Stop button: Stop cycle play

 Cycle loops: Number of cycle repetition

Infinite repetition: If that box is checked, cycle is in definitively repeated until the ‘Stop’ button is clicked.

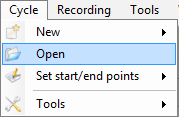
 Virtual CAN signal sent: If that box is checked, virtual CAN signal values are sent along cycle data.

 Cycle starting point: Defines the cycle play starting time for partial cycle playing.

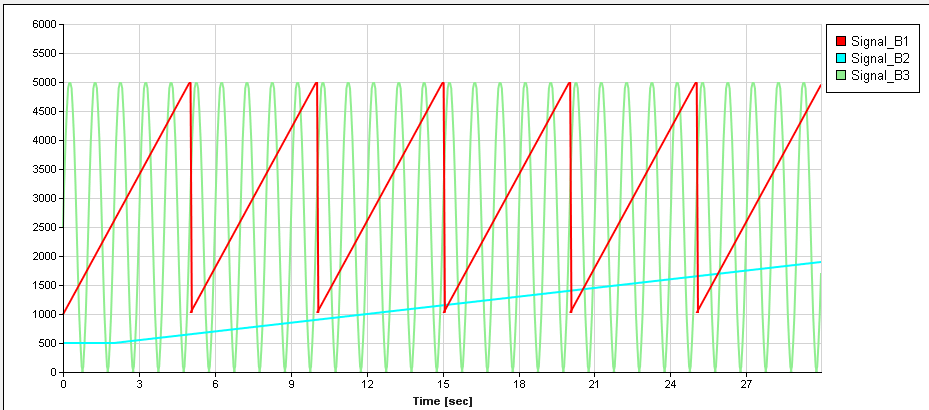
 Cycle ending point: Defines the cycle play ending time for partial cycle playing.

## Cycle playing

Open a cycle using the ‘Cycle\Open’ command of the menu strip.



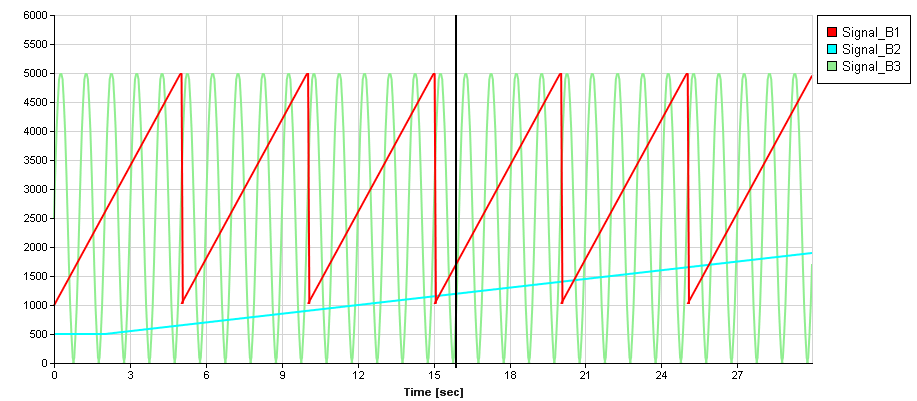
Once loaded, cycle data are showed into the cycle graphic.



Prior to start the cycle; set the number of cycle repetitions in the ‘Count’ field. Then, click the ‘Play’ button  to start the cycle.

The cycle can be stopped at any time using the ‘stop’ button or just held at the current position clicking the ‘Pause’ button.

During the cycle playing, a cursor appears in the graphic showing the current position within the cycle.



In addition of this cursor, some more information shows up in the cycle control panel while the cycle is played.

 Current repetition number (starting at one)

Cycle progression bars.

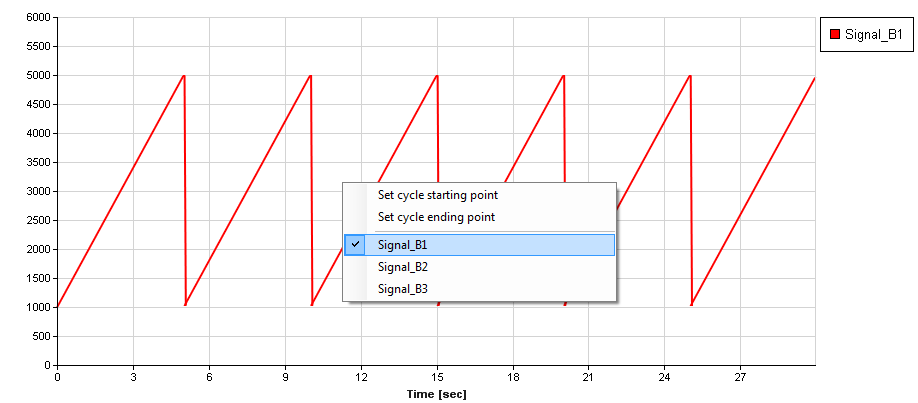


Those progress bars show the progression of the cycle.

Current cycle: Progression of the current cycle

Total: Progression of the whole sequence considering the number of repetitions. For instance, if repetition is set to two, the ‘Total’ bar will be filled up two times slower than the ‘Current cycle’ bar.

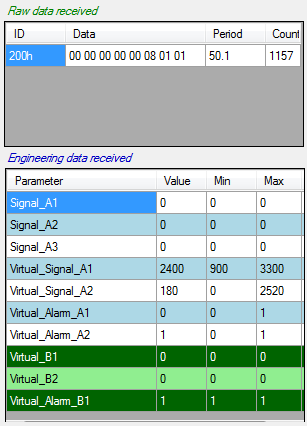
By default, all cycle data are plotted into the graphic. To hide one or more signal, right click in the graphic and uncheck signals that you don’t want to see.



### Cycle RX data panel

During cycle playing, data received are shown and decoded in the ‘Received data’ panel. As per the manual mode, this panel is split in two parts: raw data and engineering data.

In the panel ‘Raw data received’ received CAN frames and values of their bytes are shown. While the ‘Engineering data received’ shows decoded signal values of received frames.



As per the manual mode, columns and rows of those two grids can be customized using their contextual menus or the ‘View’ menu of the main menu strip.

### Cycle starting and ending points setting

By default, a cycle is played from the beginning to the last sample value. This is not mandatory since user can define the part of the cycle that will be played. Then cycle will no longer be played from the beginning to the end but from the starting and/or ending points defined by the user.

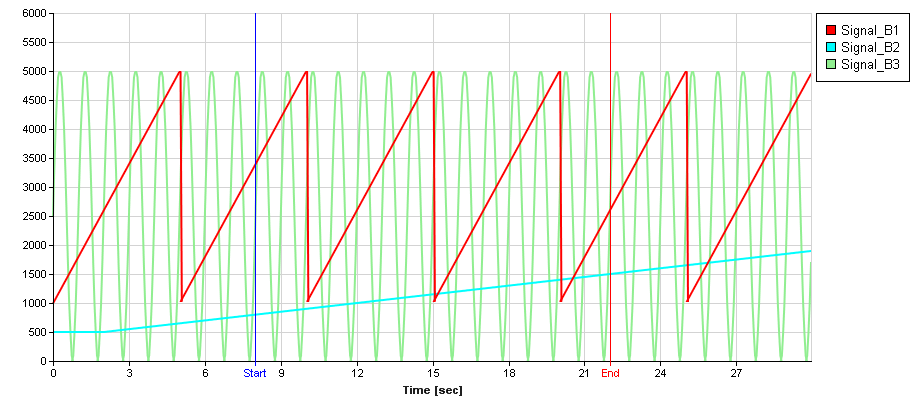
Even better, user doesn’t necessarily need to define both starting and ending points. If only the starting point is defined, then cycle will be played from this point to the end. Alternatively, if only the ending point is set, cycle will be played from the beginning to this ending point.

There are two methods to define cycle starting and ending points

The numeric method: simply change the value of ‘Start’ and ‘End’ fields. Value of those fields must be time in second.

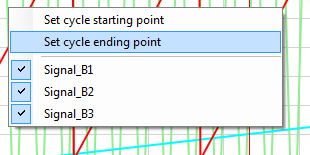


If values set are different than actual cycle beginning and ending points cursors appear in the graphic showing current cycle starting and ending points.



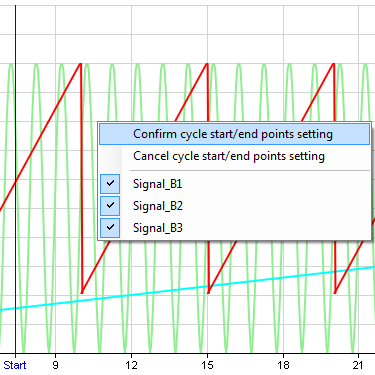
The second method consists in positioning those cursors directly into the graphic.

To do so, right click in the graphic and click ‘Set cycle starting point’ to set the starting point or ‘Set cycle ending point’ for the ending point.



Then place the cursor at the position you want the cycle to start or end by clicking this position into the graphic.

Once that position defined, right click again into the graphic to get the contextual menu and click ‘Confirm cycle start/end points settings’ to confirm starting or ending position.



If you click ‘Cancel cycle start/end points settings’, the cursor goes back to the previous position defined if any. Otherwise it simply disappears and starting or ending are either the beginning or the end of the cycle.

### Cycle loop number setting

A cycle can be played multiple times. To define the number of cycle repetitions, simply change the value ‘Count’ prior to launch the cycle.

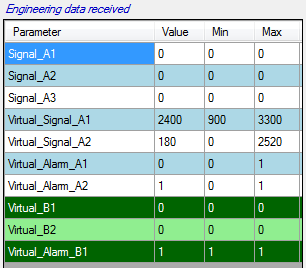


Alternatively, a cycle can be played in definitively by checking the box ‘Infinite play’ . In that case, cycle will be played and repeated until the ‘Stop’ button  is clicked.

### Virtual CAN signals in cycle

As per manual mode, virtual channels can be computed based on values received on the CAN bus while a cycle is played. This can be particularly useful if virtual channels are alarms sent to a device to stop unit operation or ignore commands.

When a virtual channel is computed, it appears in the ‘Engineering data received’ with green background.



# Cycle creation

Prior to be loaded and played, a cycle has to be somehow created…

CANStream proposes three methods to create cycle files:

* Easy cycle creation

With this method the cycle is created based on a data file. This mode has very few options, that’s why it is called ‘easy’.

* Advanced cycle creation

Unlike the ‘easy’ mode, the ‘advanced’ mode has extended options in order to tweak the cycle.

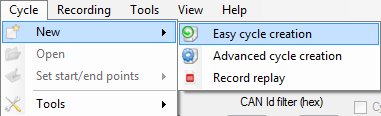
* Record replay

This is actually the easiest cycle creation mode! Cycle is built using a record which is converted into a cycle file.

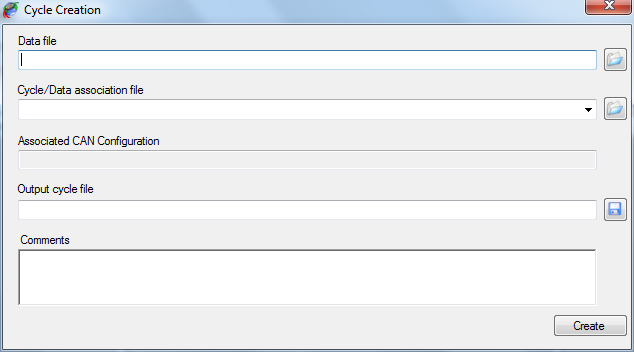
## Easy cycle creation

The ‘Easy cycle creation’ method creates a cycle based on a data file. This mode has very few options, that’s why it is called ‘easy’.

Click the ‘Cycle\New\Easy cycle creation’ menu of the main menu strip to access the ‘Easy cycle creation’ mode.



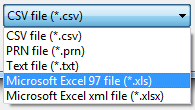
The ‘Easy cycle creation’ form shows up.



To create a cycle using this method, simply follow steps described below:

* Select a cycle source data file by clicking the ‘Open’  button on the right of the ‘Data file’ field.

Several formats of data are supported for cycle creation. Change the extension by changing the extension filter of the open file dialog.



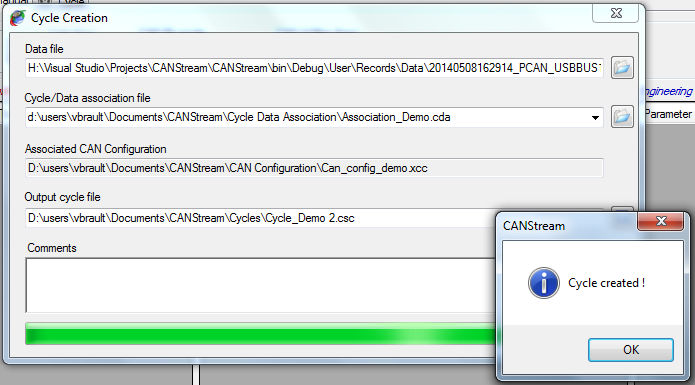
* Select a ‘Cycle/Data association’ file by clicking the ‘Open’  button on the right of ‘Cycle/Data association file’ filed. If your association file is located in the ‘Cycle Data Association’ folder of the ‘CANStream’ folder of ‘My documents’, it will present in the list. So you can drop down the list and pick up the association file that you want to use.

Once the association file selected, the CAN configuration for which the association has been made for is displayed (for information) in the ‘Associated CAN configuration’ text box. This text box cannot be modified, so select another association file (or make a new one) if you don’t want use the CAN configuration showed.

* Set the name and path of the output cycle file clicking the ‘Save’  button.
* Optionally write a description of your cycle in the ‘Comments’ area.
* Then click the ‘Create’ button  to launch the cycle creation.

Depending of the type of the source file (Excel or ASCII) and its size, it may take several minutes to create the cycle. This is particularly long with large Excel files since the Excel automation interface for .NET application is a kind of slow…

However, during the creation a progress bar appears indicating progression of the cycle creation process. At the end of the cycle creation, a message box pops up to let the user know that creation process has ended.



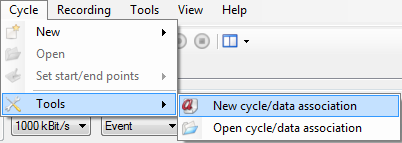
### Cycle data association

A ‘Cycle data association’ file is a file containing association between CAN parameters and data source channels.

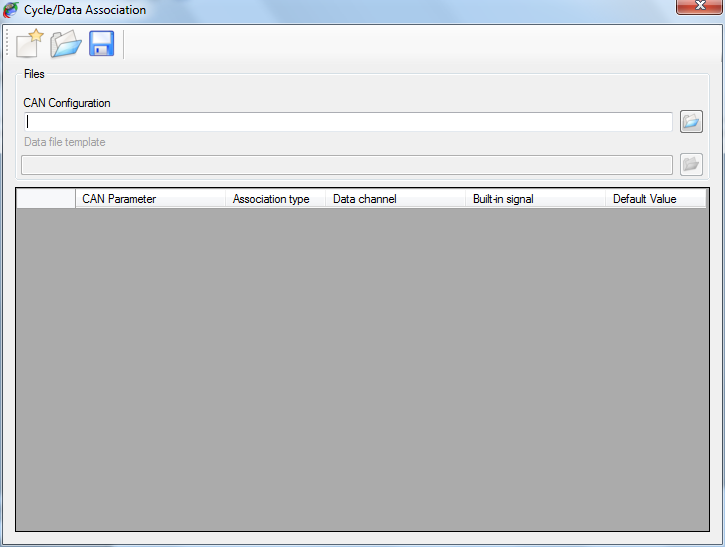
For example, let’s say that you have a CAN configuration containing a CAN parameter called ‘Speed’ and that you want create a cycle using this configuration. You have a source data file to create your cycle but in this data file, the ‘Speed’ channel is called ‘Velocity’.

A ‘Cycle data association’ file will fix that issue since it allow to associate CAN parameters with a data channel. So in our case we will simply states that the CAN parameter ‘Speed’ is using values of the data channel ‘Velocity’ as source of data.

To open the ‘Cycle data association’ edition form, click the ‘Cycle\Tools\New Cycle\Data association’ menu.



The ‘Cycle data association’ edition form appears



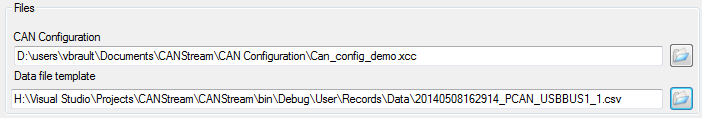
This form contains a tool bar on the top to access the ‘Cycle data association’ file management commands (new, open, save).

A control panel in which CAN configuration and data file are intended to be specified

A grid, containing actual CAN parameter/data channel associations.

Click the ‘Open’  button on the right of the ‘CAN configuration’ field.

Then select the data file that you want to use as template to set associations by clicking the ‘Open’  button on the right of the ‘Data file template’ field

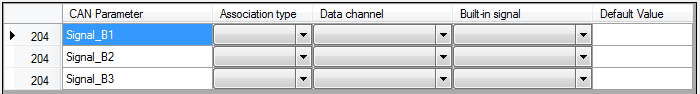


As mentioned by its label, that file is a template. So it doesn’t mean that association works only with this particular data file. It means association will work with any data file having the same structure as the template.

This is important here to underline that by ‘structure’ we consider both data channel name and data channel location. If channel ‘Velocity’ is placed on the third column of the data file, association will work with all data file having a channel ‘Velocity’ on the third column. If ‘Velocity’ moves to the fifth column for some reasons, association won’t any longer work and association file will have to be modified accordingly.

Another important point is that CANStream consider data of the first column as time vector. If time isn’t on the first column it may results to unpredictable cycle.

Once the CAN configuration loaded the association grid is filled, each row of the grid being a CAN parameter of the CAN configuration.



The association grid has six columns:

* Message ID: Identifier of the message containing the CAN parameter (for information only)
* CAN Parameter: Name of the CAN parameter in the CAN configuration
* Association type: Type of the association
* Data channel: Name of the data channel associated to the CAN parameter
* Built-in signal: Name of the built-in signal associated to the CAN parameter
* Default value: Fix value associated to the CAN parameter

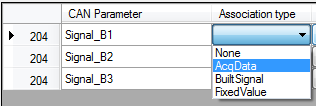
#### Association type

Main purpose of an association file is to associate CAN parameters with data channel. Actually, a CAN parameter can be associated with something else than a data channel.

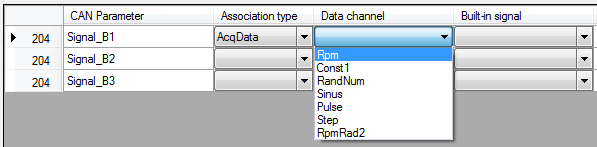
There are four types of association:

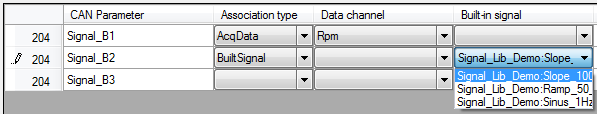
* None: CAN parameter isn’t associated, its value will zero in the cycle
* AcqData: CAN parameter is associated to a data channel. CAN parameter values in the cycle will be values of the data channel.
* Built-in signal: CAN parameter is associated to a built-in signal. CAN parameter values in the cycle will be values of the built-in signal
* Fixed value: CAN parameter is associated to a fix value. CAN parameter values in the cycle will be the value defined all the time.

Click on ‘Association type’ cell of a CAN parameter and pick up the association that you want to use.

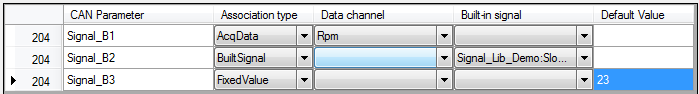


Once the association type defined, select the data channel or built-in signal in the corresponding cell.





For a ‘Fixed value’ association simply type a value into the ‘Default value’ cell.



Several CAN parameters can be associated with the same data channel or built-in signal.

Once all associations are set click the ‘Save’  button of the tool bar to save the association file.

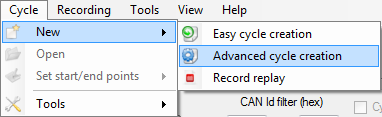
## Advanced cycle creation

The ‘Advanced cycle creation’ method has extended options for cycle creation allowing tweaking the cycle in order to fit with particular needs.

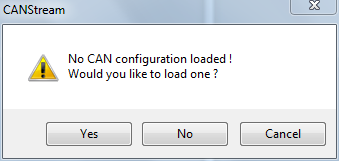
With the ‘Advanced cycle creation’ method, cycle creation configuration can be saved into a cycle creation configuration file (\*.x3c) in order to have configuration re-useable. Such file contains all information needed to create the cycle (CAN configuration, built-in signals, virtual channels) so it can be easily distributed to others without having to give a whole bunch of files.

The biggest advantage of using the ‘Advanced cycle creation’ is that it can use virtual channels as source of data for the cycle. Thus, since virtual channels can be computed using multiple built-in signals, variables and functions, it allows the generation of complex profiles for the cycle.

Click the ‘Cycle\New\Advanced cycle creation’ menu of the main menu strip to access the ‘Advanced cycle creation’ mode.



If there is no CAN configuration currently loaded, a message box pops up proposing to load one.

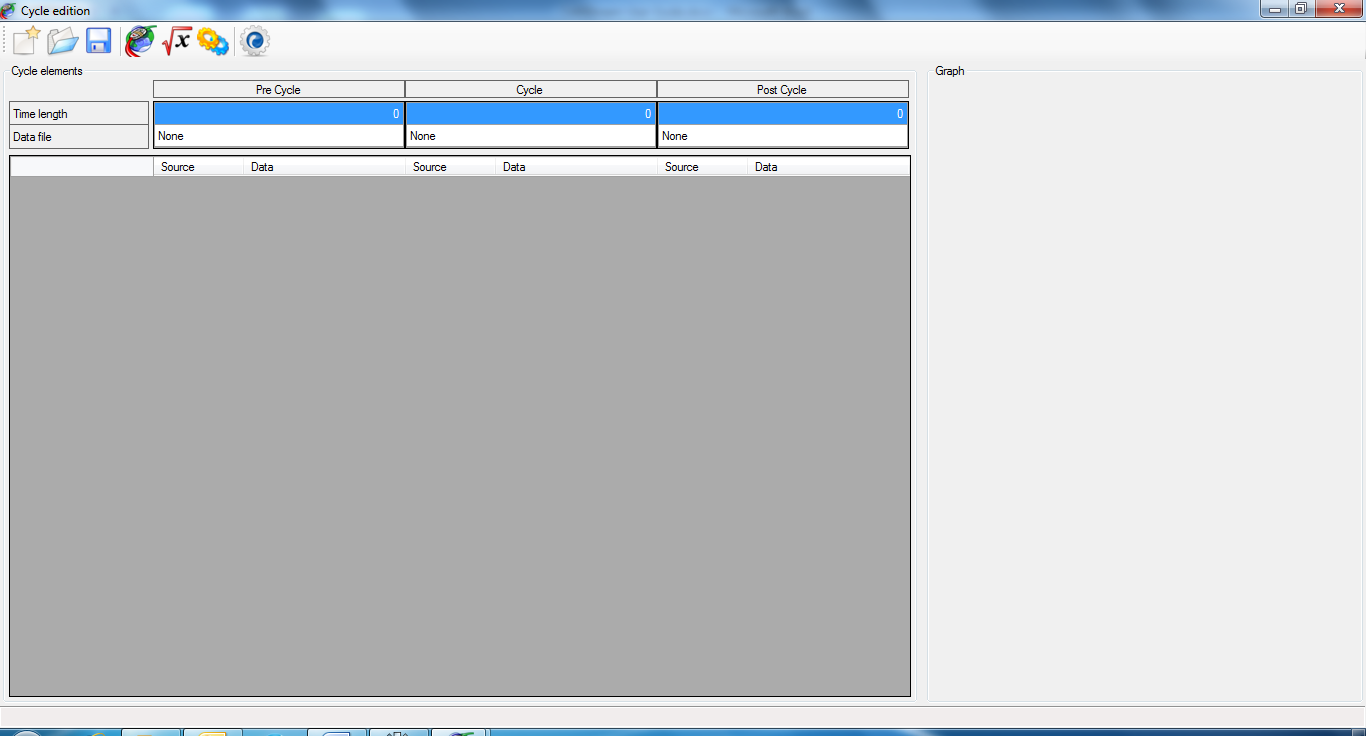


If you answer ‘Yes’ a ‘File open’ dialog will show up in which you can select the CAN configuration that has to be used to create the cycle.

If you answer ‘No’, the ‘Advanced cycle creation’ form will open but you still will have to open a CAN configuration file in order to create a cycle.

If you answer ‘Cancel’, it cancels the cycle creation and returns to the main form of CANStream.

Well, let’s say that we have answered ‘No’ to this question… Form appears as below.

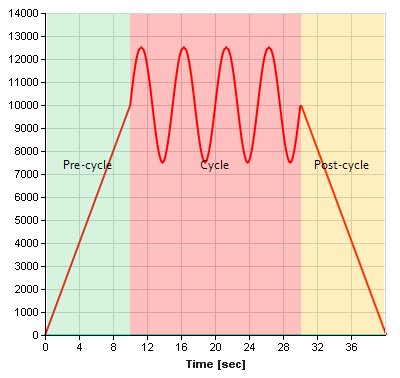


In the ‘Advanced cycle creation’ environment, a cycle is split in three parts: ‘Pre-cycle’, ‘In cycle’ and ‘Post-cycle’.

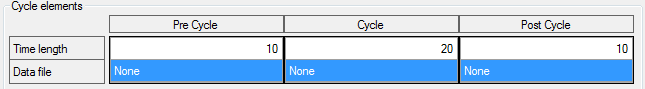
Those parts can be considered as sub-cycles, final whole cycle being the concatenation of the three parts. If we consider the ‘In cycle’ part as the actual cycle, the ‘Pre-cycle’ part is the sequence played before the cycle like an introduction. The ‘Post-cycle’ part is therefore the sequence played after the cycle like a conclusion.

‘Pre’ and ‘Post’ cycle concept is very useful when the system to test has to be gently started and stopped.

For instance, something looking like the profile bellow



The grid on the top regroups cycle parts properties.

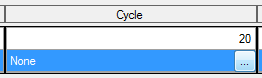


There is one column for each cycle part.

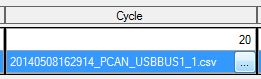
Row ‘Time length’ specifies the time length (in second) of each part, final cycle time length being the sum of cycle parts length. So in the example above, final cycle will be 40 seconds (10 + 20 + 10 = 40).

Row ‘Data file’ contains the file used as data source for each cycle part. All cycle parts can use the same or different files.

To set a data file for a cycle part, click in the ‘Data file’ cell of the part to make the ‘Open file’ button visible.



Click this button to open the open file dialog and select a file. Once the data file selected, its name is shown in the cell.



If the time length of the data file exceeds the time length of the cycle part, CANStream will propose you to change the cycle part length in order to match the length of the data file but this is not mandatory.

If the time length of a part exceeds the data file time length, the last sample value of the data file is used until the end of the cycle part.

The tool bar at the top of the form contains some commands:

New: Create a new cycle creation configuration file

Open: Open a cycle creation configuration file

Save: Save the current cycle creation configuration file

Load CAN Configuration: Load a CAN configuration file

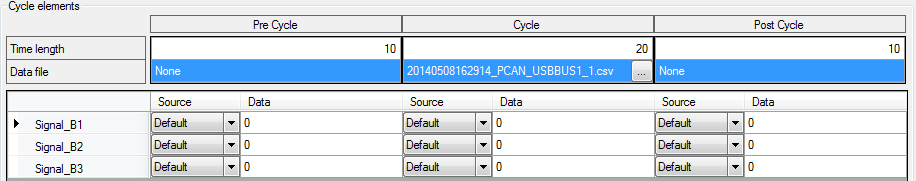
Virtual channel: Reload the current virtual channels library collection (in case of virtual channel modification during the cycle edition).

 Built-in Signals: Reload the current built-in signals library collection (in case of built-in signals modification during the cycle edition).

 Build cycle: Starts the cycle creation process.

If there was not CAN configuration loaded at the form start up, click the ‘Load CAN Configuration’ button  of the tool bar to load one.

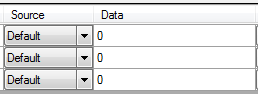
Once loaded, all CAN parameters contained in CAN messages set with ‘Transmission’ (Tx) flag will be display in the grid.



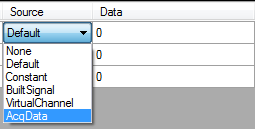
Each row of the grid represents a CAN parameter and has three groups of two columns.

Those three groups are respectively, ‘Pre-cycle’, ‘Cycle’ and ‘Post-cycle’ parts of the cycle.

The two columns of each group are configuration of the CAN parameter for a given cycle part.



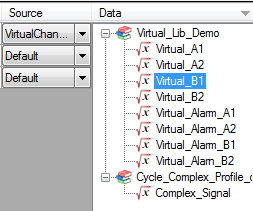
Column ‘Source’ defines the data source for the CAN parameter. Click the ‘Source’ cell to drop the source list down and pick up the desired source.



There are six possible sources:

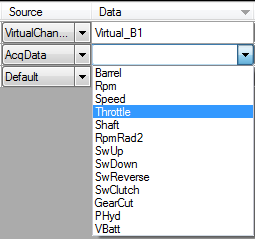
* None: CAN parameter value is set to zero
* Default: CAN parameter value is set with the default value which is zero
* Constant: CAN parameter value is set with the constant value set in the ‘Data’ cell.
* BuiltSignal: CAN parameter value is set with value of the built-in signal defined in the ‘Data’ cell
* VirtualChannel: CAN parameter value is set with value of the virtual channel defined in the ‘Data’ cell
* AcqData: CAN parameter value is set with value of the acquisition data channel defined in the ‘Data’ cell

If the ‘Source’ property is set to ‘BuiltSignal’ or ‘VirtualChannel’, click the ‘Data’ cell to get the list of items available.



Available item list takes a form of a tree where libraries are roots of the tree and items (virtual channels or built-in signals are branches. Just double-click an item to select it as data source of the CAN parameter.

If the ‘Source’ property is set to ‘AcqData’, click the ‘Data’ cell to get the list of data channels available. Simply double-click a channel to select it as data source of the CAN parameter.



You don’t need here to select a source for each CAN parameter and for all cycle parts. If a cycle part length is null, it won’t generated so it can be ignored while selecting data sources of CAN parameters.

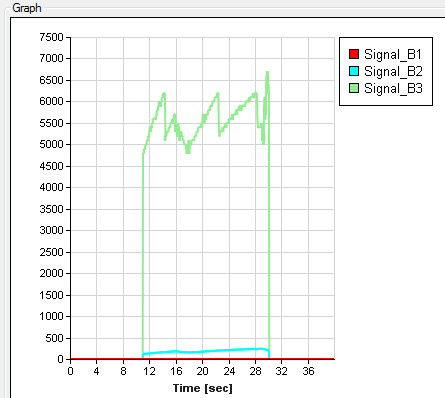
Once all CAN parameter of your interest are set, click the ‘Build cycle’ button  to launch the cycle creation.

At the first cycle building, a ‘file save’ dialog pops up. Set the name and path of the output cycle file. The cycle can be built several times but the ‘file save’ dialog will show up only once.

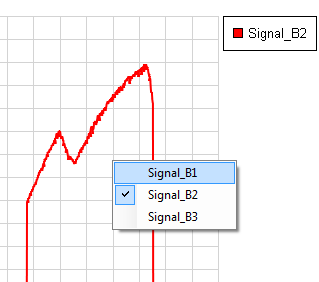
During cycle building, the ‘Abort’ button  will appear. Click this button at any time while CANStream is building the cycle to cancel the process.

At the end of the cycle creation process, a message box will pop up indicating that cycle has been successfully created.

In addition, a graphic preview of the cycle will be drawn on right side of the form.



As per the cycle player graph, right click in the graph to select traces that you want to see in the graphic.



## Record replay

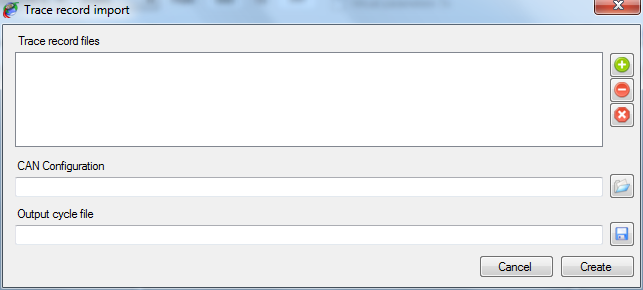
The ‘Record replay’ cycle creation method simply transform one or more PCAN Trace files (\*.trc) into a cycle file. This method is particularly useful to quickly replicate a behavior that has been previously recorded.

With this method you don’t need to associate anything, frames contained in the PCAN Trace file are simply reformatted to fit with the CANStream cycle file format.

Click the ‘Cycle\New\Record replay’ menu of the main menu strip to access the ‘Record replay creation’ mode.



The ‘Record replay’ form shows up.



To create a cycle using this method, simply follow steps described below:

* Add one or more PCAN Trace file into the ‘Trace record file’. Basically all trace files are concatenated to each other in the order they appear in the list in order to make the final cycle.

Trace files list control commands are located on the left of the list. There are three commands:

Add: Add a PCAN Trace file into the list



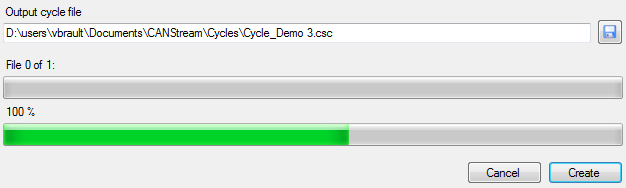
Del: Remove a PCAN Trace file from the list



 Clear: Clear all PCAN Trace file from the list

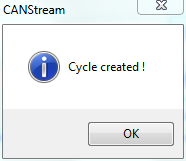
* Select a CAN configuration file by clicking the ‘Open’ button  on the right of the ‘CAN Configuration’ field. This CAN configuration is only use to get the CAN bus speed (1000 kBit/s, 500 kBit/s) nothing else.
* Define name and file path of the output cycle by clicking the ‘Save’ button  on the right of the ‘Output cycle file’ field.
* Click the ‘Create’ button  to launch the cycle creation.

During the cycle creation process two progress bars appears.



The first bar, on the top, indicates the process progression among PCAN Trace file to convert. The second bar, at the bottom, indicates the progression of the conversion for the current file.

At the end of the cycle creation, when all PCAN Trace file have been converted, a message box pops up indicating that process has ended.



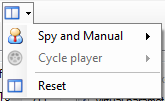
At any time in the process it is possible to cancel the cycle creation by clicking the ‘Cancel’ button .

# CAN controller appearance options

The appearance of a CAN controller can be customized in order to best fit with user needs.

## Control layout

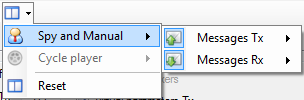
The ‘View’  command of the tool bar contains functions to customize the appearance of the current CAN controller.



Whether the active panel is ‘Spy and Manual’ or ‘Cycle’ the corresponding commands group is enabled. ‘Reset’ button restores all default layout options for both ‘Spy and Manual’ and ‘Cycle’ panels.

### Manual control layout

Click ‘Spy and Manual’ to get layout commands of this mode.



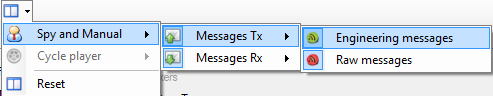
The menu is divided in two parts:

* ‘Messages Tx’ for the transmission part of the manual control
* ‘Messages Rx’ for the reception part of the manual control

The ‘Message Tx’ menu is divided in two parts:

Engineering messages: Showing or hiding the ‘Engineering’ section of the data transmission panel.

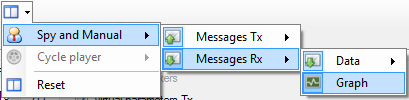
Raw messages: Showing or hiding the ‘Raw’ data section of the data transmission panel.



The ‘Message Rx’ menu is also divided in two parts:

Data: Showing or hiding the ‘Data’ section of the data reception panel.

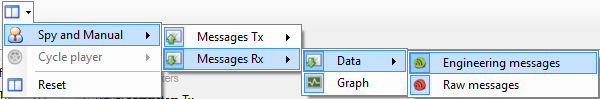
Graph: Showing or hiding the ‘Graphic trace’ section of the data reception panel.



Finally, ‘Data’ menu is split in two parts

Engineering messages: Showing or hiding the ‘Engineering’ section of the data reception panel.

Raw messages: Showing or hiding the ‘Raw data section of the data reception panel.



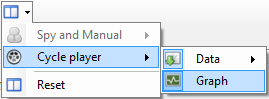
### Cycle control layout

Click ‘Cycle player’ menu to get layout commands of this mode.

The ‘Cycle player menu is divided in two parts:

Data: Showing or hiding the ‘Data’ section of the cycle panel.

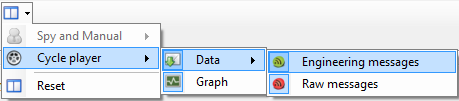
Graph: Showing or hiding the ‘Graphic trace’ section cycle.



Then, ‘Data’ sub-menu is split in two parts

Engineering messages: Showing or hiding the ‘Engineering’ section of the data reception panel.

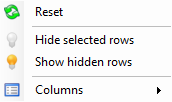
Raw messages: Showing or hiding the ‘Raw data section of the data reception panel.



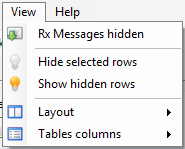
## Grids contents control

The contents of all grids (columns and rows) of a CAN controller can be individually customized.

To customize the content of a grid, right click on it to get its contextual menu.



Grid customization commands are also available through the ‘View’ menu of the main menu strip.



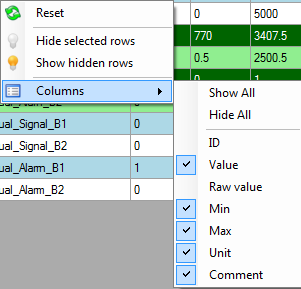
### Grid rows

Each row of a grid can be hidden. Select rows that you want hide and click ‘Hide selected rows’ . Then click on ‘Show hidden rows’ to make hidden rows visible.

### Grid columns

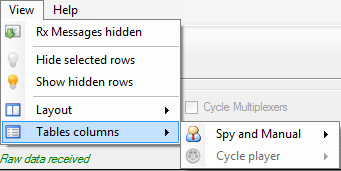
The width of the different columns of a grid cannot be changed but it is possible to selected columns that have to be shown and columns that have to be hidden.

Menu ‘Columns’ of the contextual menu of a grid contains columns title for the grid owning the contextual menu. Simply check or uncheck column titles to make columns visible or invisible.



Commands ‘Show all’ and ‘Hide all’ show and hide all columns at once. Then every single column can be individually set checking or unchecking its title.

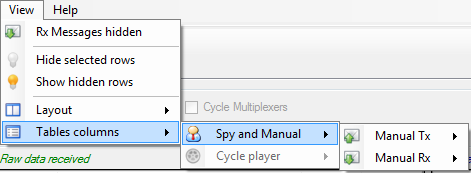
The ‘Table columns’ menu of the ‘View’ menu contains columns control commands of all grid of a CAN controller.



Whether the active panel is ‘Spy and Manual’ or ‘Cycle’ the corresponding commands group is enabled.

#### Manual control grid columns

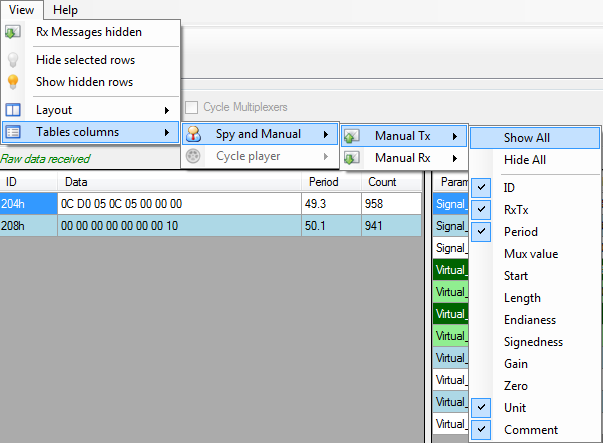
Click ‘Spy and Manual’ to get grid columns commands of this mode.



The menu is divided in two parts:

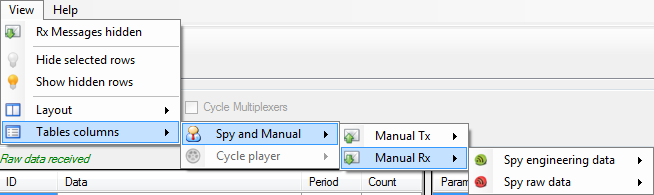
* ‘Messages Tx’ for the transmission part of the manual control
* ‘Messages Rx’ for the reception part of the manual control

The ‘Manual Tx’ contains columns control commands of the engineering data transmission grid.

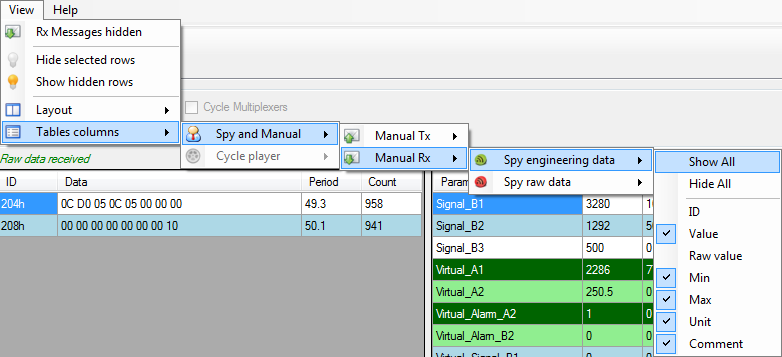


Columns of the raw data transmission grid cannot be customized since all columns are need.

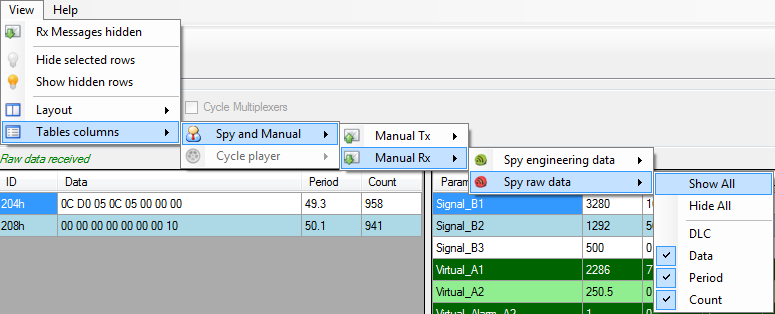
The ‘Manual Rx’ menu is divided in two parts:



Engineering messages: containing columns control commands of the engineering data reception grid.

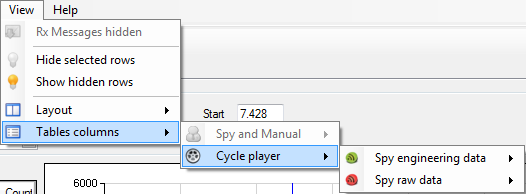


Raw messages: containing columns control commands of the raw data reception grid.



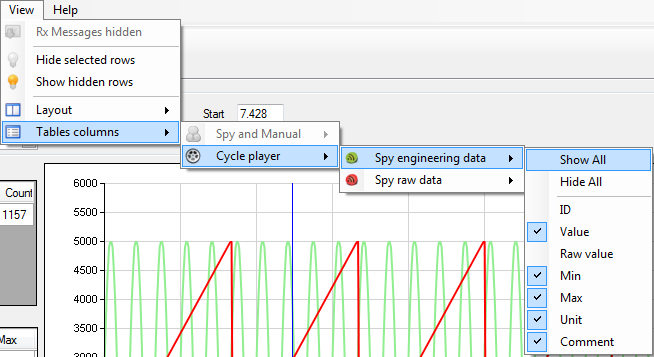
#### Manual control grid columns

Click ‘Cycle player’ to get grid columns commands of this mode.

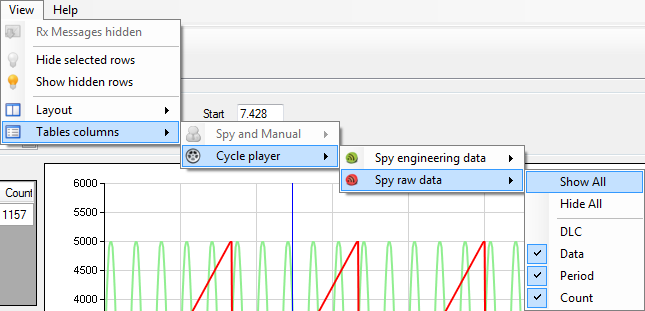


The ‘Cycle player’ menu is divided in two parts:

Engineering messages: containing columns control commands of the engineering data reception grid for the cycle panel.



Raw messages: containing columns control commands of the raw data reception grid for the cycle panel.



.

# Data logging

While manual or cycle player modes are running it is possible to record all CAN frames circulating in the CAN bus. For the time that data logger is turned on, the content of all frames as well as their time stamps will be recorded and stored into a file.

Such logging files are actually very difficult to read for a human. Indeed, it is a very large file containing hexadecimal values. Fortunately CANStream offers the ability to decode those files and convert it to a readable format. On the top of that, CANStream can also show logging data in a graphic window to make easy and accurate the record analysis.

Data recording doesn’t require any special configuration; just turn it on and off when you need. However, logging file naming convention being a kind of meaningless, it may be very difficult to identify which particular logging file among twenty or thirty files (or even more) is corresponding to a particular test. To assist the user in this task, CANStream has also some functions to enhance the logging files storage.

## Data logging start & stop

Data logger can be turned on when a PCAN-USB device is connected and used by CANStream. When this is the case, the ‘Start stream logging’ button  of the tool bar becomes active.

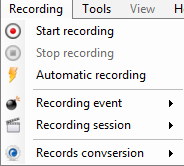


Just click this button to start logging the data. When data logging starts, the ‘Start stream logging’ button is disabled and the ‘Stop steam logging’ button is active. Simply click this button to stop the data logging.



You don’t necessarily need to start the ‘Spy & Manual’ or the ‘Cycle’ mode to record data. If CAN frames are circulating in the bus they will be recorded whether or not CANStream is operating.

Start and stop logging commands are also available in the ‘Recording’ menu of the main menu strip.



### Automatic recording

If you don’t want manually start and stop the logger each time you are testing, you can turn on the ‘Automatic recording’ feature.

To enable this function, click the ‘Automatic recording’ item in the ‘Recording’ menu of the main menu strip.

When active, the ‘Automatic recording’ is automatically turning the logger on and off when you start and stop CANStream ‘Spy & Manual’ or ‘Cycle’ mode. Thus your entire work session can be recorded without the need to care about the logger itself.

While automatic recording is active the thunderbolt icon of its menu item is highlighted.



## Data file formats

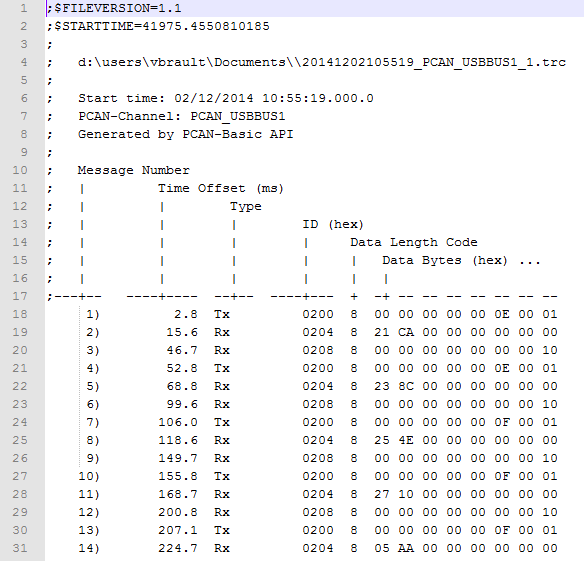
Data logging function of CANStream is dealing with two different file extensions: ‘\*.trc’ and ‘\*.csv’.

‘\*.trc’ files  are raw data file generated by the Peak PCAN-USB driver. Those files contain CAN frames in hexadecimal format and their timestamps.

‘\*.csv’ files  are decoded and converted logging file generated by CANStream. Those files are intended to be used by the user either in the data analysis window of CANStream or in an external application such as Microsoft Excel or equivalent.

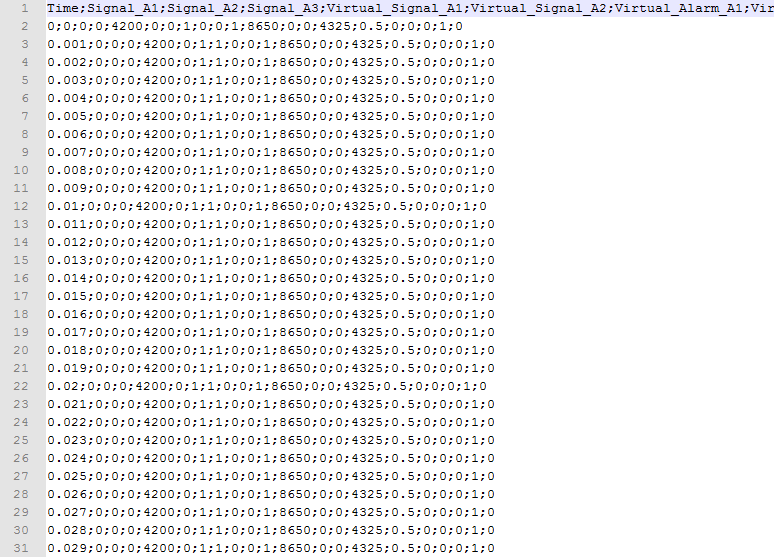
### Peak PCAN trace file

Example of a Peak PCAN trace file (\*.trc) 



### Record data file

Example of a CANStream record data file (\*.csv) 



## Logging file conversion

Logging files conversion from the Peak ‘PCAN trace’ format (\*.trc) to the CANStream ‘Record data file’ (\*.csv) is made using a CAN configuration file in order to identify CAN messages and get CAN signals values out of raw byte values of recorded frames.

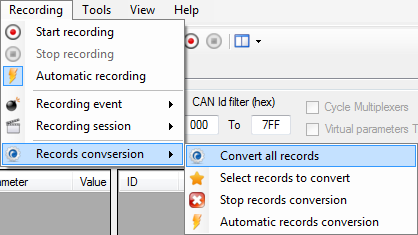
Very basically, CANStream identifies CAN messages thanks to their identifiers and decodes frame byte values using CAN parameter properties set into the CAN configuration file.

There are two methods to convert data logging file; the manual conversion and the automatic conversion. With manual conversion, user has to manually launch the conversion process while in automatic mode logging files are automatically converted as soon as the data logger stops and the logging file is closed.

### Manual conversion

The ‘Record conversion’ sub-menu of the ‘Recording’ menu of the main menu strip contains all data file conversion commands.

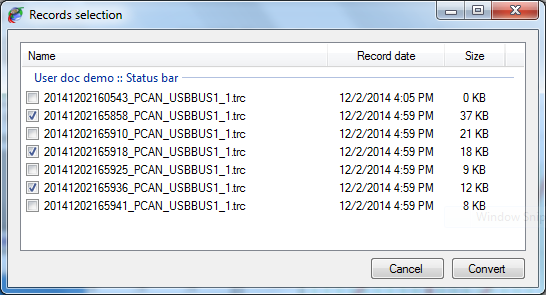
As far as the manual conversion, two options are available. It is possible to either convert all logging files at once or to manually select in a list files that have to be converted.



Click the ‘Convert all records’  menu to convert all data files by a single click.

To make your own selection of files to convert, click the ‘Select record to convert’  menu.

The file selection form shows up.

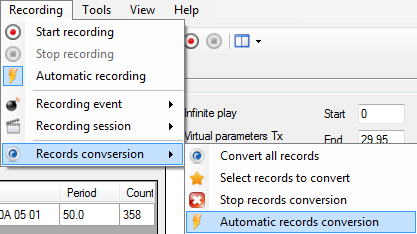


Check the box of all files that you want to convert and then click the ‘Convert’ button once you are ready to launch the conversion process.

### Automatic conversion

As per the automatic recording, automatic conversion only needs to be activated.

Click the ‘Automatic record conversion’ item in the ‘Recording conversion’ menu of the main menu strip.



When active, the ‘Automatic conversion’ is automatically launching the file conversion process when the data logger stops and the logging file is closed. Coupled with the automatic recording, the automatic conversion allows you to have your entire work session recorded and converted without the need to care about neither the logger nor conversion.

While automatic conversion is active, the thunderbolt icon of the menu item is highlighted.



### Conversion process control

A progress bar appears in the main form status bar during the conversion process.



This bar shows the progression of the complete conversion process. When the bar is full all logging files have been converted.

You can stop the conversion at any time during the process by click the ‘Stop conversion’ button  of the status bar.

### Data file storage structure

The data file storage structure is described in great details in the next section. However, basic structure is the following.

All data files (\*.trc and \*.csv) are stored in the ‘Records’ folder in ‘CANStream’ folder of the user ‘Documents’ directory (…\users\xxx\Documents\CANStream\Records)

The ‘Records’ folder contains three sub-folders: ‘Data’, ‘Raw’ and ‘Stack’.

* The ‘Data’ folder contains converted record data files (\*.csv) 
* The ‘Raw’ folder contains PCAN Trace data files (\*.trc)  that have been converted already.
* The ‘Stack’ folder contains PCAN Trace data files (\*.trc)  that are waiting to be converted.

You may find also a couple of XML files in those three folders. Check the next section for more details about those XML files.

## Recording event & Recording session

Until you have only three, four or five data files it is fairly easy to remember and identify what those file are. Imagine now that you have fifty of those data files! It becomes a kind of tricky to figure out what file is the file corresponding to the test you are looking for, especially if you are looking for an old record.

To assist you with file archiving and make data files research a way easier, CANStream uses the ‘Event\Session’ structure for data storage.

With such structure, data files are not stored in a unique folder. A folder is created for each event and sub-folders are created for every session. Then data files are placed into their respective session folder.

Term ‘event’ should be considered as a major group. Let’s say ‘CANStream documentation’. Inside this event we have multiple ‘sessions’ each of them corresponding to a specific topic. Let’s say that we have three sessions ‘Manual mode’, ‘Cycle mode’ and ‘Virtual channel’.

Thanks to this structure, we know that all data of the ‘CANStream documentation’ folder are data that have been generated for the documentation. Then, session folders indicate for what exact part of the documentation, data files have been recorded.

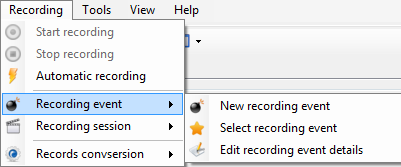
On the top of that, for every session of every event, CANStream creates an XML file containing information about the session or event. An event/session XML file contains the name of the event or session, its date and description and optionally a bunch of user information. This list of user information can be anything the final user has found useful to identify and track data files. In our example (CANStream documentation) we can set the CANStream release name and the name of the CAN configuration that have been used. Number and content of user information is actually unlimited so you can have how many you want and write whatever you want.

Name of both current recording event and current session is shown in the main form status bar.



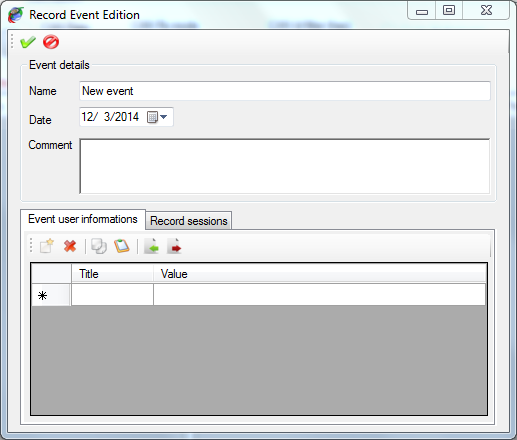
### Recording event creation & edition

Record event control commands are available in the ‘Record event’ item  of the ‘Record’ menu of the main menu strip.



Click the ‘New recording event’ menu  to create a new recording event or click the ‘Edit recording event details’ menu  to edit the current recording event.

The recording event edition form shows up.



Tool bar on the top contains only two commands

Apply: Create the record event or apply changes to the current record event



Cancel: Cancel the record event creation or change made to the current record event



When creating a new record event, this event becomes the current record event. So all records performed afterward will stored under this new event.

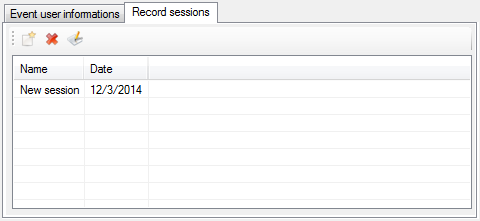
Frame ‘Event details’ contains main information about the recording event

* Name: Name of the recording event
* Date: Starting date of the recording event
* Comment: Recording event description.

Panel at the bottom of the form has two tabs

* Event user information: Management of user information attached to the record event. Check the ‘User information’ section for more details
* Record sessions: Management of the record sessions of the current event

The ‘Record sessions’ panel contains session control commands for the current record event. When creating a new record event, a session ‘New session’ is created by default.



All sessions of the record event are listed in the record session list. Tool bar on the top of the list contains session control commands.

New: Create a new session inside the current record event



Delete: Delete a session of the current record event.



Edit: Edit details of the selected session. Check ‘Recording session creation & edition’ for more details

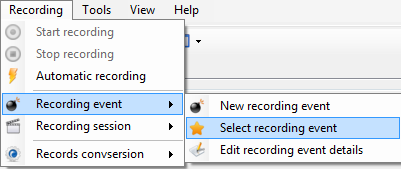


When deleting a record session, session is only removed from the list. Data that may have been recorded under the session to delete are not deleted.

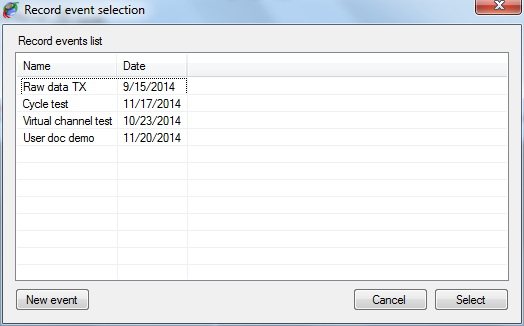
### Recording event selection

At any time, you can switch to a different recording event. If the recording event that you want switch to has already been created, it is possible to pick it up in a list.

Click the ‘Select recording event’ menu  to switch to another recording event



The available record events list pops up.

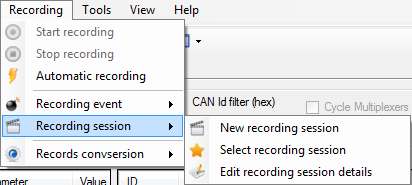


Just double click the record event you want to use or click the ‘Select’ button after having selected it into the list.

Alternatively, you can also create a new record event by clicking the ‘New event’ button.

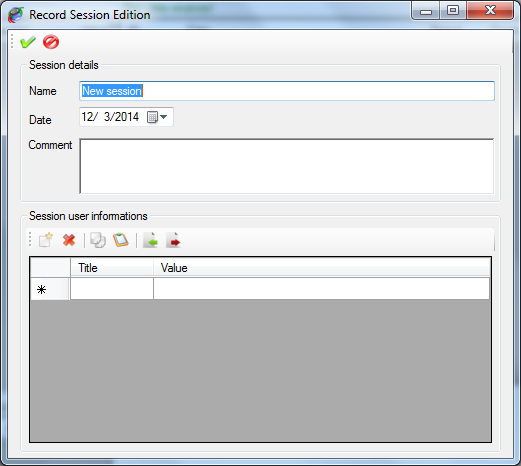
### Recording session creation & edition

Record session control commands are available in the ‘Record session’  item of the ‘Record’ menu of the main menu strip.



Click the ‘New recording session’ menu  to create a new recording session or click the ‘Edit recording session details’ menu  to edit the current recording event.

The recording session edition form shows up.



Tool bar on the top contains only two commands

 Apply: Create the record session or apply changes to the current record session

 Cancel: Cancel the record session creation or change made to the current record session

When creating a new record session, this session becomes the current record session. So all records performed afterward will stored under this new session.

Frame ‘Session details’ contains main information about the recording session

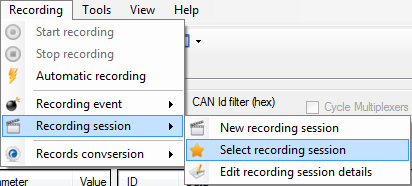
* Name: Name of the recording event
* Date: Starting date of the recording event
* Comment: Recording event description.

Panel ‘Session user information’ contains control commands for user information attached to the record session. Check the ‘User information’ section for more details

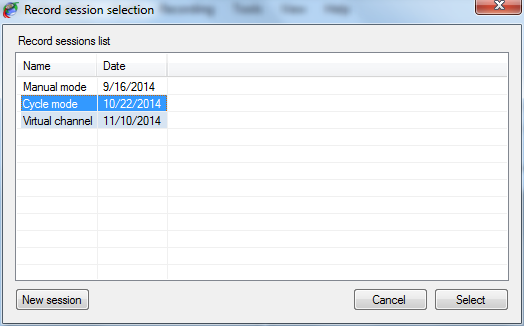
### Recording session selection

At any time, you can switch to a different recording session. It is possible to pick up in a list the recording session that you want switch to if this session has already been created.

Click the ‘Select recording session’ menu  to switch to another recording event



The available record session list pops up.



Just double click the record session you want to use or click the ‘Select’ button after having selected it into the list.

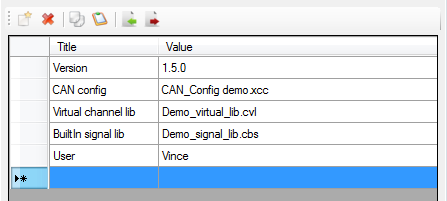
Alternatively, you can also create a new record session by clicking the ‘New session button.

### User information

User information for record event or record session is an optional collection of information defined by the user and which are attached to an event or to a session.

Number and content of this information is unlimited so you can have as many as you want of user information and they can contain whatever you want attached to an event or to a session. User information is not mandatory; it is possible to have event or session without any user information.

User information control panel is common for both recording event and recording session objects.



User information control panel is composed of a grid containing all user information, and a tool bar in which user info control commands are accessible.

User information is made of two fields:

* Title: The title of the user information
* Value: Value of the user information

To be considered as valid (and consequently to be stored) a user information must have a least a title. Value can remain empty but all grid rows having their ‘Title’ cell empty will be ignored.

To create a new ‘user information’, simply write its title in the empty row at the bottom of the list. Then write its value in the ‘Value’ cell.

The tool bar contains the following commands:

Add: Create a new user information



Del: Remove a user information from the list



Copy: Copy a user information on the clipboard



Past: Past a user information from the clipboard



Import: Import a user information collection file



Export: Export the current collection of user information into a file

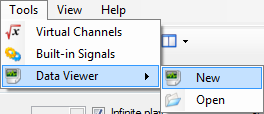


In order to avoid the need of having to recreate all the time your collection of user information, it is possible to export a collection into a file and re-use this file whenever you need it. The same user information can be used either for a record event or a record session without any restriction.

# Data analysis

In addition of logging and decoding data CANStream offers the possibility to analyze those data into a graphic window. The data viewer of CANStream allows loading multiple data files at once even if they are not part of the same record event or session. Data files are concatenated in order to create a data set that appear as a continuous record in the data viewer.

The menu ‘Tools’ of the main menu strip gives access to this data analysis window

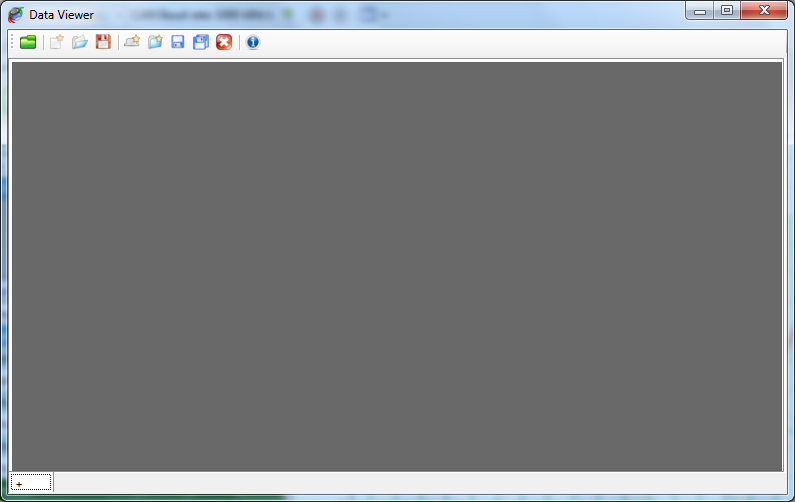


Item ‘Data Viewer’ of the ‘Tools’ menu contains two sub-menu: ‘New’  and ‘Open’ 

Click the ‘New’  menu to open an empty data analysis window and click the ‘Open’  menu to open an existing data analysis window.

## Analysis window

When creating a new analysis window, the window comes empty.



The data viewer window is composed of two elements

* The tool bar on the top, containing all data viewer control commands
* The multi-tab panel containing all pages of the data viewer.

Since the data viewer window can have multiple pages it can be considered as book. Each page of this book is a graphic window with a different graphic configuration.

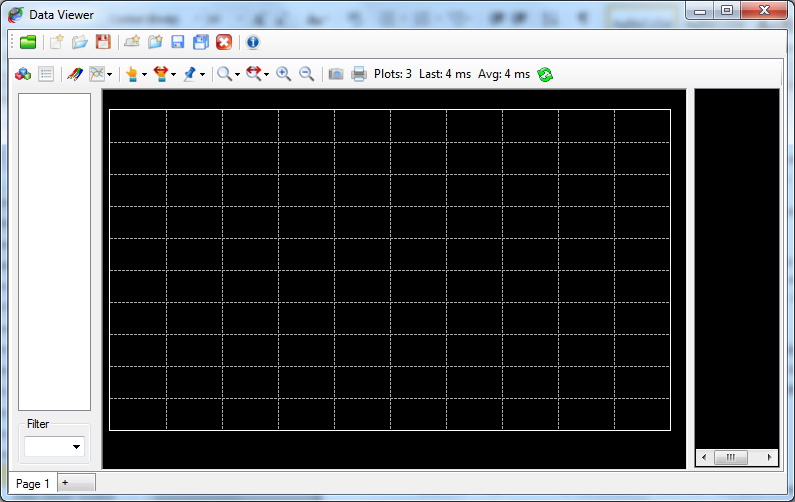
Data to be displayed are loaded through the data browser. Check the ‘Data browser’ section for more details. However the set of data loaded is unique over the entire book. It means that all pages are actually showing the same set of data. It is consequently impossible to see the data file ‘A’ in the first page and the data file ‘B’ in the second page.

This behavior might sound as a weakness or a missing functionality, but actually this is not a weakness. From experienced users know that while working with multiple pages it is possible to get lost and not remember what exact data set is loaded on each page. So to make things easy all pages are showing the same data set.

### Data viewer page creation

Click the ‘+’ tab  at the bottom or click the ‘New data viewer page’ button  of the tool bar to create a new page in the data viewer.

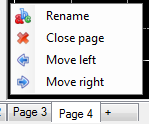
The new data viewer page is created.



A data viewer book can contain as many pages as you wish. Each single page appears as a tab of the multi-tab panel.



Right click on a page tab to get the viewer page contextual menu.



This contextual menu has four commands.

Rename: Rename the active page of the data viewer



Close: Close the active page of the data viewer



Move left: Move the active page one step on the left



Move right: Move the active page one step on the right

Click the ‘Rename’ menu  to rename a page. A text box pops up.



Write the new name for the page in this text box.



Then press the ‘Enter’ key.



Data viewer pages can be individually saved and loaded through commands available in the tool bar. The entire data viewer book can be saved as well.

An entire section of the documentation is dedicated to the usage and configuration of the graphic window. Please check this section for more details.

### Data viewer Toolbar

Toolbar of the ‘Data viewer’ window contains the following commands

 Data browser: Open the data browser window

New: Create new data viewer book

Open: Open a data viewer book



Save: Save the current data viewer book (all pages)



New page: Create a new data viewer page



Open page: Open an existing data viewer page (only one)



Save page: Save the current data viewer page



Save all pages: Save all pages of the data viewer book



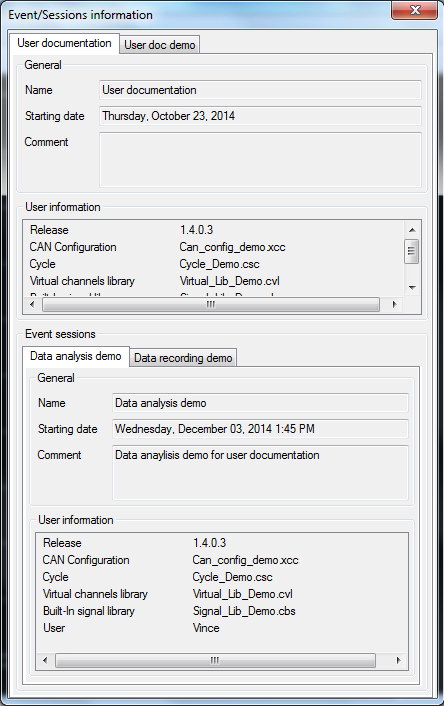
Close page: Remove the current page from the data viewer book



Info: Shows event and session details of the loaded data set.

### Data set information

When data are displayed into the data viewer it is possible to see their record event and record session information. Click the ‘Show event/session information’ button  of the tool bar to get the information window.



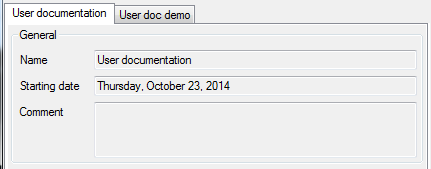
Event and session information cannot be changed from this window. They are only displayed for information purpose.

Information window contains only a multi-tab panel.

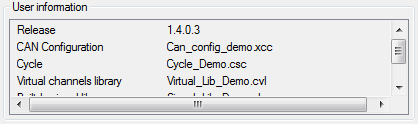
Each tab of the panel corresponds to one the record event of a data file of the current data set.



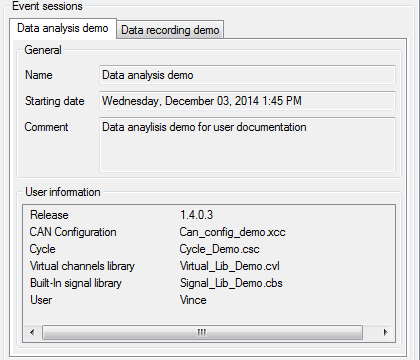
The ‘General’ section of the panel shows general information about the recording event.



The ‘User information’ section shows all user information attached to the record event.



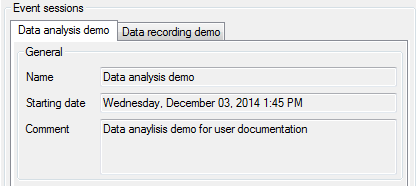
Section ‘Record sessions’ shows details of each record session being part of the current data set.



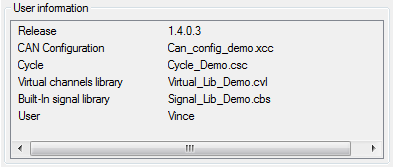
Each record session has its own tab in the ‘Event session’ section multi-tab panel.



As per record event, ‘General’ section of a record session tab shows general information about the record session.



The ‘User information’ section shows all user information attached to the record session.

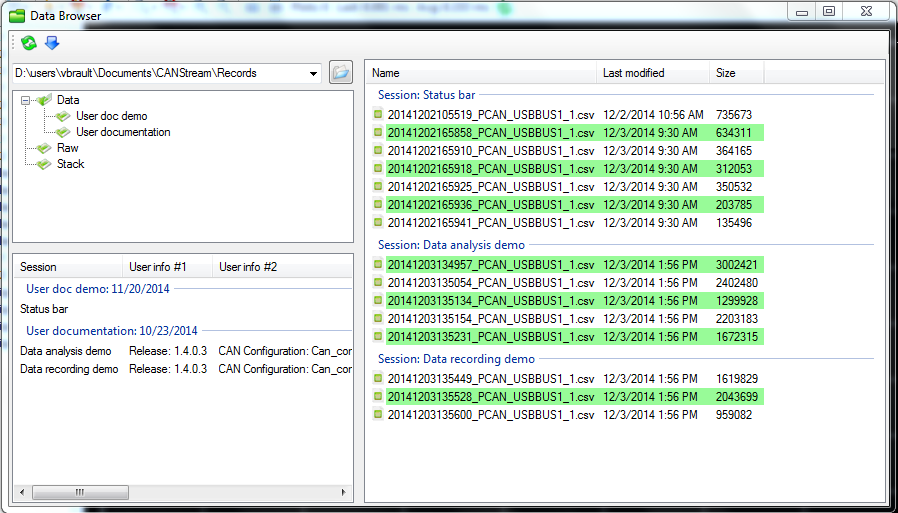


## Data browser

The CANStream data browser in a data file explorer. Data that have to be displayed in the data viewer has to be loaded from this explorer.

Click the ‘Data browser’  button of the data viewer tool bar to open the data browser.

The data browser shows up.

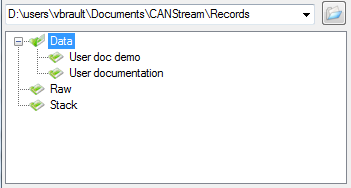


The data browser is made of four parts:

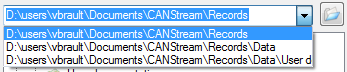
* The tool bar: Toolbar contains only two commands, ‘Refresh’ and ‘Load’
* The root folder tree: It shows the folder arborescence from current root folder
* The event list: It shows all record events contained in the selected folder
* The data file list: It shows all data files grouped by record session present in the selected folder and all its sub-folders.

### Root folder tree

The root folder tree shows the folder arborescence from the current root folder.



Click the ‘Open’  button to change the root folder. Root folder text box is actually a list that can be dropped in order to get the history of all root folder previously used and quickly switch among those root folders.

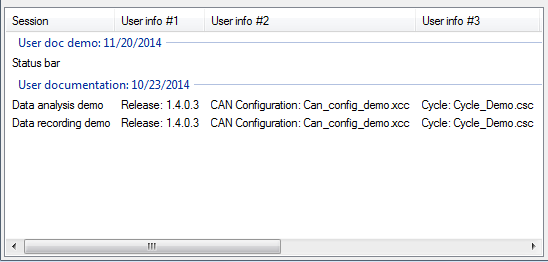


The last ten folders used as root folder are saved in this list.

Click a folder to explore its subfolders and all their child folders.

### Events & sessions list

The events and sessions list is located at the bottom of the root folder tree.

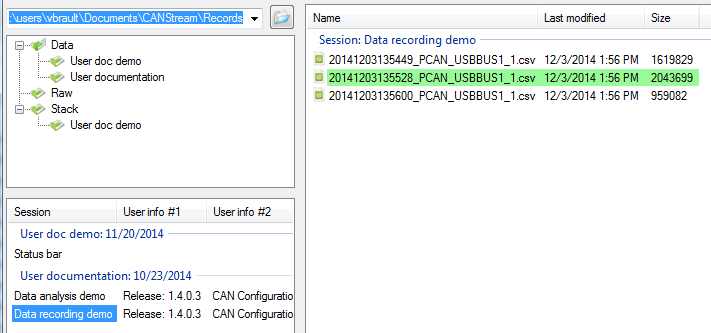


Each group of the list represents a record event.



Items of each group are a record session.

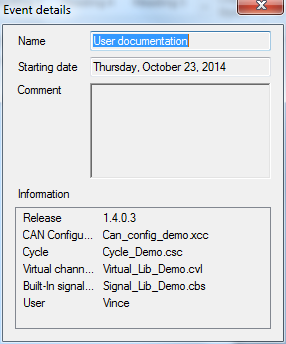
Select a session item in order to apply a filter to the logging data list and show only data file part of the selected record session.



To get information about a particular record event of the list, right click one of its sessions.

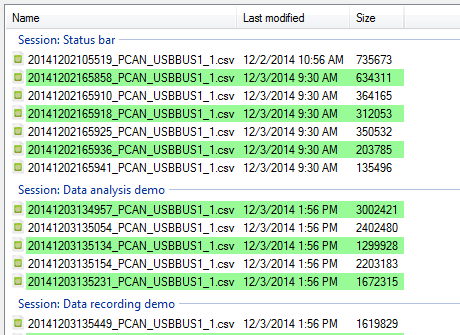


Click the ‘Event details’  menu to get the information window.



### Logging data list

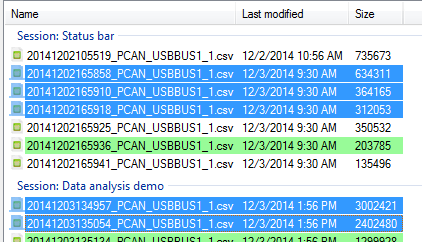
The logging data list contains actual data files that can be loaded into the data viewer window.



Each group of the list represents a record session.

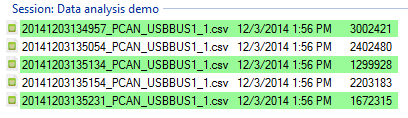


Select all files that you want to load and press the ‘Enter’ key or click the ‘Load selected files’ button  of the toolbar.

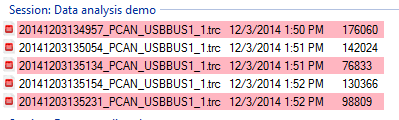


The data browser shows the two logging data formats. ‘\*.csv’ files  for converted data file and ‘\*.trc’ files  for raw PCAN-Trace data file.

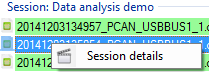
‘\*.csv’ files  only can be loaded in the data viewer. Files with such extension appear with a green background.



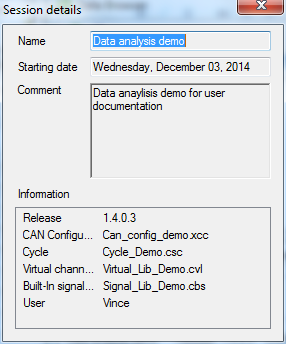
‘\*.trc’ files  appear with a red background but they cannot be loaded directly into the data viewer. You need to convert those file (if it has not been done already) prior to see those data.



To get information about a particular record session of the list, right click one of its data files.



Click the ‘Session details’  menu to get the information window.



### Toolbar

The data browser toolbar contains only two commands

Refresh: Refresh the folder tree, event and session list as well as the data file list



 Load: Load data files selected into the data viewer

## Graphic window

### Presentation

### Channels list

### Main graphic cursor

### Reference cursor

### Zoom

### Legend

### Statistics

### Graphic snapshot

### Graphic printing

## Graphic window configuration

### Graphic series layout

### Main graphic series properties

### General graphic window properties

### Graphic grids properties

### Graphic abscise properties

### Cursors graphical properties

## Graphic series detailed properties

### Series general properties

### Series value formats

### Series Y axis

### Series custom grid

### Series reference lines

# Virtual channels

## Virtual channels edition

### Virtual channels library

### Virtual channel properties

### Virtual channel testing

### Virtual channel elements objects

### Virtual channel built-in functions

|  |  |
| --- | --- |
| *Function* | *Description* |
| Abs (A) | Returns the absolute value of the A number. |
| Acos (A) | Returns the angle whose cosine is the A number. |
| Asin (A) | Returns the angle whose sine is the A number. |
| Atan (A) | Returns the angle whose tangent is the A number. |
| Atan2 (A, B) | Returns the angle whose tangent is the quotient of specified A and B numbers. |
| Atm2Bar (A) | Return pressure A converted from atmosphere to bar |
| Atm2kMmHg (A) | Return pressure A converted from atmosphere to mercury millimeter |
| Atm2kPa (A) | Return pressure A converted from atmosphere to kilo Pascal |
| Atm2Pa (A) | Return pressure A converted from atmosphere to Pascal |
| Atm2Psi (A) | Return pressure A converted from atmosphere to psi |
| Bar2Atm (A) | Return pressure A converted from bar to atmosphere |
| Bar2kPa (A) | Return pressure A converted from bar to kilo Pascal |
| Bar2MmHg (A) | Return pressure A converted from bar to mercury millimeter |
| Bar2Pa (A) | Return pressure A converted from bar to Pascal |
| Bar2Psi (A) | Return pressure A converted from bar to psi |
| BigMul (A, B) | Produces the full product of two 32-bit A and B numbers. |
| Cal2Joule (A) | Return energy A converted from calorie to Joule |
| Cal2kCal (A) | Return energy A converted from calorie to kilo calorie |
| Cal2kJ (A) | Return energy A converted from calorie to kilo Joule |
| Ceiling (A) | Returns the smallest integral value that is greater than or equals to the specified A number. |
| Celcius2Fahrenheit (A) | Return temperature A converted from Celsius degree to Fahrenheit degree |
| Celcius2Kelvin (A) | Return temperature A converted from Celsius degree to Kelvin degree |
| Cos (A) | Returns the cosine of the specified angle. |
| Cosh (A) | Returns the hyperbolic cosine of the specified angle. |
| Deg2Grad (A) | Return value of angle 'A" converted from degree to gradient |
| Deg2Rad (A) | Return value of angle 'A" converted from degree to radian |
| DivRem (A, B, out C) | Calculates the quotient of A and B numbers and also returns the remainder in an output parameter C. |
| Exp (A) | Returns e raised to the A power. |
| Fahrenheit2Celcius (A) | Return temperature A converted from Fahrenheit degree to Celsius degree |
| Fahrenheit2Kelvin (A) | Return temperature A converted from Fahrenheit degree to Kelvin degree |
| Feet2Km (A) | Return length A converted from feet to kilometer |
| Feet2Meter (A) | Return length A converted from feet to meter |
| Feet2Mile (A) | Return length A converted from feet to mile |
| Feet2Yard (A) | Return length A converted from feet to yard |
| Floor(A) | Returns the largest integer less than or equal to the specified A number. |
| Floor(A) | Returns the largest integer less than or equal to the specified A number. |
| Grad2Deg (A) | Return value of angle 'A" converted from gradient to degree |
| Grad2Rad (A) | Return value of angle 'A" converted from gradient to radian |
| Gram2Kg (A) | Return weight A converted from gram to kilogram |
| Gram2Ounce (A) | Return weight A converted from gram to ounce |
| Gram2Pound (A) | Return weight A converted from gram to pound |
| Gram2Tonne (A) | Return weight A converted from gram to tone |
| IEEERemainder (A, B) | Returns the remainder resulting from the division of the A number by the B number. |
| IFEQ (A, B) | Return 1 if value of argument 'A' is equal to value of argument 'B' |
| IFGE (A, B) | Return 1 if value of argument 'A' is greater or equal to value of argument 'B' |
| IFGT (A, B) | Return 1 if value of argument 'A' is greater than value of argument 'B' |
| IFLE (A, B) | Return 1 if value of argument 'A' is smaller or equal to value of argument 'B' |
| IFLT (A, B) | Return 1 if value of argument 'A' is smaller than value of argument 'B' |
| IFNE (A, B) | Return 1 if value of argument 'A' is not equal to value of argument 'B' |
| INRG (A, Min, Max) | Return 1 if value of argument 'A' is contained within the 'Min/Max' range |
| Joule2Cal (A) | Return energy A converted from Joule to calorie |
| Joule2kCal (A) | Return energy A converted from Joule to kilo calorie |
| Joule2kJ (A) | Return energy A converted from Joule to kilo Joule |
| KCal2Cal (A) | Return energy A converted from kilo calorie to calorie |
| KCal2Joule (A) | Return energy A converted from kilo calorie to Joule |
| KCal2kJ (A) | Return energy A converted from kilo calorie to kilo Joule |
| Kelvin2Celcius (A) | Return temperature A converted from Kelvin degree to Celsius degree |
| Kelvin2Fahrenheit (A) | Return temperature A converted from Kelvin degree to Fahrenheit degree |
| Kg2Gram (A) | Return weight A converted from kilogram to gram |
| Kg2Ounce (A) | Return weight A converted from kilogram to ounce |
| Kg2Pound (A) | Return weight A converted from kilogram to pound |
| Kg2Tonne (A) | Return weight A converted from kilogram to tone |
| Kj2Cal (A) | Return energy A converted from kilo Joule to calorie |
| Kj2Joule (A) | Return energy A converted from kilo Joule to Joule |
| Kj2kCal (A) | Return energy A converted from kilo Joule to kilo calorie |
| Km2Feet (A) | Return length A converted from kilometer to feet |
| Km2Meter (A) | Return length A converted from kilometer to meter |
| Km2Mile (A) | Return length A converted from kilometer to mile |
| Km2Yard (A) | Return length A converted from kilometer to yard |
| kPa2Atm (A) | Return pressure A converted from kilo Pascal to atmosphere |
| kPa2Bar (A) | Return pressure A converted from kilo Pascal to bar |
| kPa2MmHg (A) | Return pressure A converted from kilo Pascal to mercury millimeter |
| kPa2Pa (A) | Return pressure A converted from kilo Pascal to Pascal |
| kPa2Psi (A) | Return pressure A converted from kilo Pascal to psi |
| kph2mph (A) | Return speed A converted from kilometer per hour to mile per hour |
| kph2ms (A) | Return speed A converted from kilometer per hour to meter/second |
| Log (A) | Returns the natural (base e) logarithm of the A number. |
| Log (A, B) | Returns the logarithm of the A number in the specified B base. |
| Log10 (A) | Returns the base 10 logarithm of the A number. |
| Max (A, B) | Returns the larger of A and B numbers |
| Meter2Feet (A) | Return length A converted from meter to feet |
| Meter2Km (A) | Return length A converted from meter to kilometer |
| Meter2Mile (A) | Return length A converted from meter to mile |
| Meter2Yard (A) | Return length A converted from meter to yard |
| Mile2Feet (A) | Return length A converted from mile to feet |
| Mile2Km (A) | Return length A converted from mile to kilometer |
| Mile2Meter (A) | Return length A converted from mile to meter |
| Mile2Yard (A) | Return length A converted from mile to yard |
| Min (A, B) | Returns the smaller of A and B numbers |
| MmHg2Atm (A) | Return pressure A converted from mercury millimeter 2 kilo atmosphere |
| MmHg2Bar (A) | Return pressure A converted from mercury millimeter 2 bar |
| MmHg2kPa (A) | Return pressure A converted from mercury millimeter 2 kilo Pascal |
| MmHg2Pa (A) | Return pressure A converted from mercury millimeter 2 Pascal |
| MmHg2Psi (A) | Return pressure A converted from mercury millimeter 2 psi |
| mph2kph (A) | Return speed A converted from mile per hour to kilometer per hour |
| mph2ms (A) | Return speed A converted from mile per hour to meter/second |
| ms2kph (A) | Return speed A converted from meter/second to kilometer per hour |
| ms2mph (A) | Return speed A converted from meter/second to mile per hour |
| OOR (A, Min, Max) | Return 1 if value of argument 'A' is not contained within the 'Min/Max' range |
| Ounce2Gram (A) | Return weight A converted from ounce gram |
| Ounce2Kg (A) | Return weight A converted from ounce kilogram |
| Ounce2Pound (A) | Return weight A converted from ounce pound |
| Ounce2Tonne (A) | Return weight A converted from ounce tone |
| Pa2Atm (A) | Return pressure A converted from Pascal to atmosphere |
| Pa2Bar (A) | Return pressure A converted from Pascal to bar |
| Pa2kPa (A) | Return pressure A converted from Pascal to kilo Pascal |
| Pa2MmHg (A) | Return pressure A converted from Pascal to mercury millimeter |
| Pa2Psi (A) | Return pressure A converted from Pascal to psi |
| Pound2Gram (A) | Return weight A converted from pound to gram |
| Pound2Kg (A) | Return weight A converted from pound to kilogram |
| Pound2Ounce (A) | Return weight A converted from pound to ounce |
| Pound2Tonne (A) | Return weight A converted from pound to tone |
| Pow (A, B) | Returns the A number raised to the B power. |
| Psi2Atm (A) | Return pressure A converted from psi to atmosphere |
| Psi2Bar (A) | Return pressure A converted from psi to bar |
| Psi2kPa (A) | Return pressure A converted from psi to kilo Pascal |
| Psi2MmHg (A) | Return pressure A converted from psi to mercury millimeter |
| Psi2Pa (A) | Return pressure A converted from psi to Pascal |
| Rad2Deg (A) | Return value of angle 'A" converted from radian to degree |
| Rad2Grad (A) | Return value of angle 'A" converted from radian to gradient |
| Round (A) | Rounds the value A to the nearest integral value. |
| Sign (A) | Returns the value indicating the sign of the A number. |
| Sin (A) | Returns the sine of the A angle. |
| Sinh (A) | Returns the hyperbolic sine of the A angle. |
| Sqrt (A) | Returns the square root of the A number. |
| Tan (A) | Returns the tangent of the A angle. |
| Tanh (A) | Returns the hyperbolic tangent of the A angle. |
| Tonne2Gram (A) | Return weight A converted from tone to gram |
| Tonne2Kg (A) | Return weight A converted from tone to kilogram |
| Tonne2Ounce (A) | Return weight A converted from tone to ounce |
| Tonne2Pound (A) | Return weight A converted from tone to pound |
| Truncate(Decimal) | Calculates the integral part of a specified A number. |
| Yard2Feet (A) | Return length A converted from yard to feet |
| Yard2Km (A) | Return length A converted from yard to kilometer |
| Yard2Meter (A) | Return length A converted from yard to meter |
| Yard2Mile (A) | Return length A converted from yard to mile |

## Virtual channels use

### Use of virtual channels in the manual data Rx

### Use of virtual channels for cycle creation

## Virtual CAN signals

### Presentation

### Use of virtual CAN signals in the manual data Tx

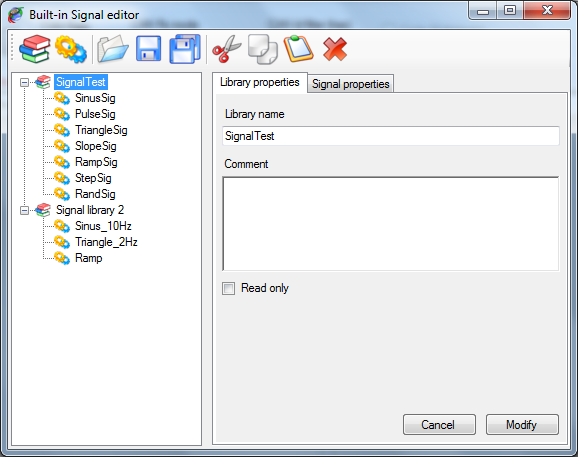
### Use of virtual CAN signals in cycle player

# Built-in signals

Built-in signals are a tool used for cycle creation. Instead of making a cycle based on real data contained into a text file or an Excel file, CANStream can generate data of a periodic signal that will be used as a source of data in the cycle.

## Built-in signals edition

As per virtual channels, built-in signals are organized by libraries. A library can contain several signals and multiple libraries can loaded together at the same time.



The built-in signal editor is composed of:

* A tool bar, on the top, containing signals and libraries control commands.
* A tree view, on the left, in which appears all signal libraries loaded.
* A multi-tab panel, on the right, for signal and library properties edition.

In the tree view, each built-in signal library is represented by the  node. All built-in signals contained in a library are child node (or branch) of the library and are represented by the  node.

The tool bar contains the following commands:

 New signals library: Create a new built-in signals library

 New signal: Create a new built-in signal into the current library

 Open library: Load an existing built-in signals library

 Save library: Save the current built-in signals library

 Save all libraries: Save all loaded built-in signals libraries

 Cut: Cut an item an place it onto the clipboard

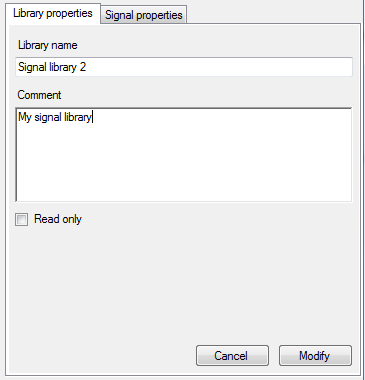
 Copy: Copy an item an place it onto the clipboard

 Past: Paste an item from the clipboard

Delete: Delete the selected item

### Built-in signals library

Click on a library item  of the tree view to edit its properties or click the ‘Create new signals library’  button to create a new signal.



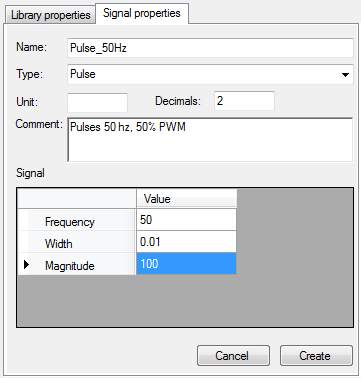
A built-in signals library has very few properties:

* Library name: Name of library
* Comment: Description of the library
* Read only: Check that box if you want prevent unwanted changes in the library

Make all changes you want and then click the ‘Modify’ button to apply your changes. In the case of the creation of a new library, ‘Modify’ button will become ‘Create’, just click it to create the library.

### Built-in signals properties

Click a signal node  to edit its properties or click the ‘Create new signal’  button to create a new signal.



A built-in signal has several properties

* Name: Name of the signal
* Type: Kind of periodic signal to be generated. Check built-in signals type section for more details.
* Unit: Unit of the signal
* Decimal: Number of signal decimal digits
* Comment: Signal description
* Signal math properties grid: Content of this grid depends of the type of signal selected. Check built-in signals type section for more details.

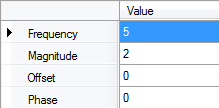
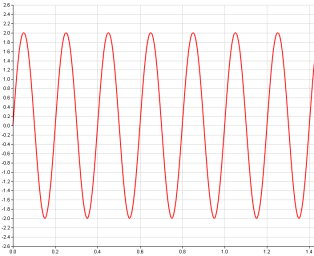
Set, at least, the signal name and type as well as its math properties and then click the ‘Create’ button to create the signal. In case of signal edition, this button will become ‘Modify’, just click it to apply all changes you may have done.

### Built-in signals types

There are seven signal types available; each of them has its own math properties.

#### Sinus

Generate a sinusoidal waveform.

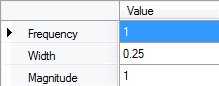
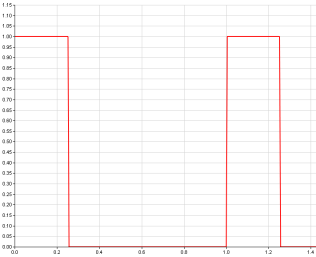


Uses the following math properties

* Frequency: Sinusoid frequency in Hertz
* Magnitude: Peak to peak signal magnitude
* Offset: Signal offset with respect to zero
* Phase: Sinusoid phase offset

#### Pulses

Generate multiple pulses waveform.

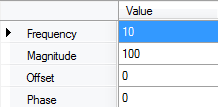
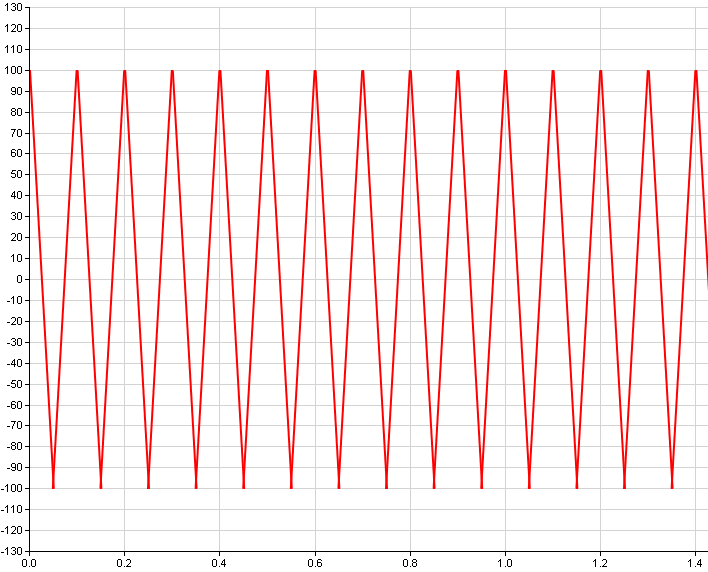


Uses the following math properties

* Frequency: Pulses frequency in Hertz
* Width: Pulse width with respect to the signal period in second
* Magnitude: Peak to peak signal magnitude

#### Triangle

Generate a triangle waveform.

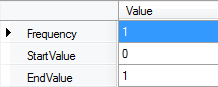
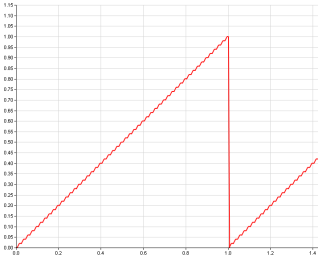


Uses the following math properties

* Frequency: Triangle frequency in Hertz
* Magnitude: Peak to peak signal magnitude
* Offset: Signal offset with respect to zero
* Phase: Triangle phase offset

#### Slopes

Generate multiple slopes waveform.

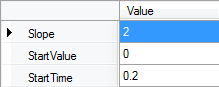
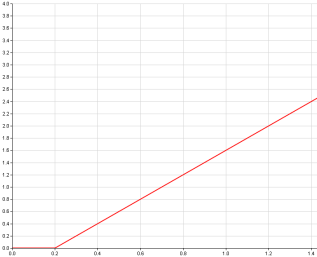


Uses the following math properties

* Frequency: Slopes frequency in Hertz
* Start value: Slope starting value
* End value: Slope ending value

#### Ramp

Generate a ramp waveform.

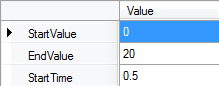
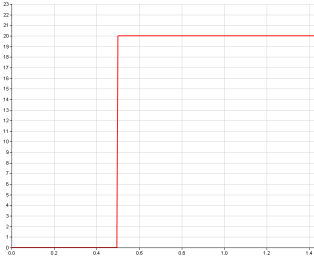


Uses the following math properties

* Slope: Ramp gradient
* Start value: Ramp starting value
* End value: Ramp starting time in second

#### Step

Generate a single step waveform.

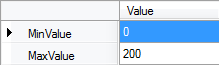
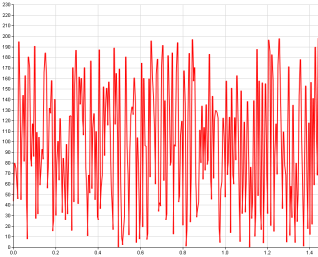


Uses the following math properties

* Start value: Step starting value
* End value: Step ending value
* Start time: Stepping time in second

#### Random

Generate a random values waveform.



Uses the following math properties

* Min value: Signal minimum value
* Max value: Signal maximum value

## Built-in signals use