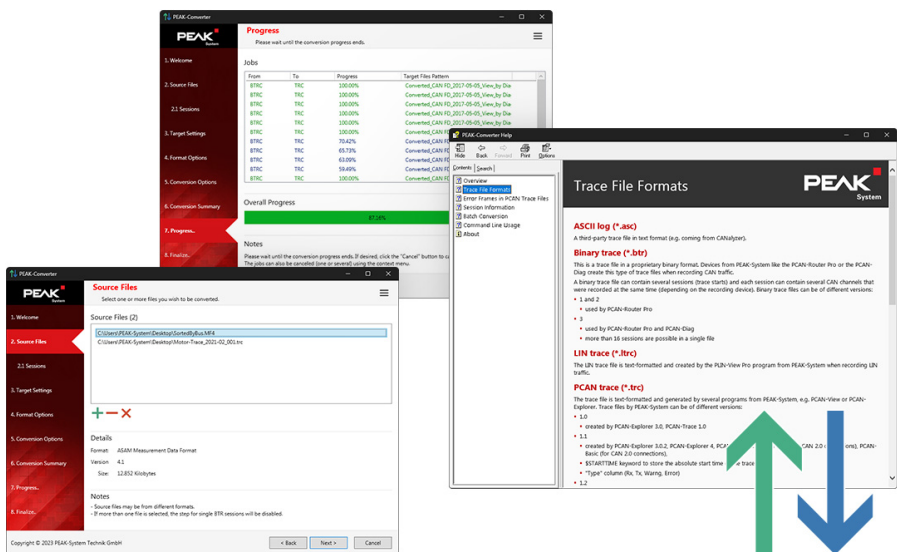


# PEAK-Converter 3



# Imprint

PCAN and PLIN are registered trademarks of PEAK-System Technik GmbH.

All other product names in this document may be the trademarks or registered trademarks of their respective companies. They are not explicitly marked by ™ or ®.

© 2024 PEAK-System Technik GmbH

Duplication (copying, printing, or other forms) and the electronic distribution of this document is only allowed with explicit permission of PEAK-System Technik GmbH. PEAK-System Technik GmbH reserves the right to change technical data without prior announcement. The general business conditions and the regulations of the license agreement apply. All rights are reserved.

PEAK-System Technik GmbH  
64293 Darmstadt  
Germany

Phone: +49 6151 8173-20

Fax: +49 6151 8173-29

[www.peak-system.com](http://www.peak-system.com)  
[info@peak-system.com](mailto:info@peak-system.com)

Document version 1.0.1 (2024-12-11)

# Contents

<b>Imprint</b>	<b>2</b>
<b>Contents</b>	<b>3</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 Overview</b>	<b>5</b>
<b>3 Trace Files Formats</b>	<b>7</b>
3.1 ASCII Log (*.asc)	7
3.2 Binary trace (*.btr)	7
3.3 LIN Trace (*.ltrc)	8
3.4 PCAN trace (*.trc)	8
3.5 PEAK binary trace (*.btrc)	10
3.6 Measurement Data Format 4 (*.mf4)	10
<b>4 Error Frames in PCAN Trace Files</b>	<b>11</b>
<b>5 Session Information</b>	<b>13</b>
5.1 Kind of Session	13
5.2 Sectors Count, First/Last Sector	13
<b>6 Batch Conversion</b>	<b>14</b>
6.1 Drag and Drop	14
6.2 Command Line	14
<b>7 Command Line Usage</b>	<b>15</b>
7.1 Command Line Format	15
7.2 Source File	15
7.3 Parameters	16
7.4 Return Values (Errorlevel)	19
7.5 Example	19

# 1 Introduction

Products like PCAN-Explorer, PCAN-Diag FD, or PCAN-Router Pro FD are capable of recording the data traffic of the connected buses. The software PEAK-Converter for Windows converts trace files to various output formats for further processing or analysis. Besides the formats of PEAK-System (\*.trc, \*.btr, \*.btrc), third-party formats like \*.csv, \*.asc, or \*.mf4 are also provided.

# 2 Overview

With the PEAK-Converter, you can convert trace files coming from different sources (devices, programs) to target files that can be used for further analyzing.

The following table shows the conversion possibilities of the PEAK-Converter.

Source	Target					
	TRC	ASC	BTR	CSV	BTRC	MDF4
TRC	■	■	■	■	■	■
ASC	■	-	-	■	-	-
BTR	■	■	-	■	-	-
LTRC	-	■	-	■	-	-
BTRC	■	■	-	■	-	-
MDF4	■	-	-	-	-	-

File format	Description	Supported format versions
TRC	Text-based trace file format from PEAK-System for recording CAN data; created by PCAN-Basic, PCAN-View, PCAN-Trace, and PCAN-Explorer	1.0, 1.1, 1.2, 1.3, 2.0, 2.1
ASC	ASCII log; Text-based log file format; created by CANalyzer	Input: up to 15 Output: 7, 8.1
BTR	Binary trace file format from PEAK-System for recording CAN data; created by PCAN-Router Pro, PCAN-Diag 2, and PCAN-MiniDisplay; can contain multiple trace sessions	1.0, 2.0, 3.0
LTRC	Text-based trace file format from PEAK-System for recording LIN data; created by PLIN-View Pro	1.0, 1.1, 1.2

File format	Description	Supported format versions
CSV	Comma-Separated Values: Common text-based file format for storing tabular data; used for import into spreadsheets	-
BTRC	Binary trace file format from PEAK-System or recording different data types such as CAN or LIN; created by PCAN-Diag FD and PCAN-Router Pro FD	
MDF4	Measurement Data Format 4: Binary trace file format from the Association for Standardization of Automation and Measuring Systems (ASAM)	4.1, 4.11 (4.1.1), 4.2

For more information about the file formats, see chapter 3 *Trace Files Formats*.

# 3 Trace Files Formats

## 3.1 ASCII Log (\*asc)

A third-party trace file in text format (e.g. coming from CANalyzer).



Starting with ASCII log format version 8, a new representation for messages generated by CAN FD capable devices was introduced. Any data message received, even a Classic CAN (CAN 2.0) one, is labeled with "CANFD" denoting CAN FD capability of the device. This is not an error in the conversion process, but rather the application of the new file format specification. If this format is not desired and CAN FD is not used, please convert using the ASCII log format version 7 as destination format.

## 3.2 Binary trace (\*btr)

This is a trace file in a proprietary binary format. Devices from PEAK-System like the PCAN-Router Pro or the PCAN-Diag create this type of trace files when recording CAN traffic.

A binary trace file can contain several sessions (trace starts) and each session can contain several CAN channels that were recorded at the same time, depending on the recording device. Binary trace files can be of different versions:

### Versions 1 and 2

- used by PCAN-Router Pro

### Version 3

- used by PCN-ROuter Pro and PCAN-Diag
- more than 16 sessions are possible

## 3.3 LIN Trace (\*ltrc)

The LIN trace file is text-formatted and created by the PLIN-View Pro program from PEAK-System when recording LIN traffic.

### Version 1.0

- created by PLIN-View Pro 1.0

### Version 1.1

- created by PLIN-View Pro 1.1.4 to 2.x
- \$STARTTIME keyword to store the absolute start time of the trace file
- Comment “Start time” now stored in ISO 8601:2004 format.
- Wrong error code label “Ck” in recorded frames replaced with “CK”

### Version 1.2

- created by PLIN-View Pro 3.0
- Column “Data bytes (hex)” renamed to “Data bytes (hex) or Bus event”

## 3.4 PCAN trace (\*trc)

The trace file is text-formatted and generated by several programs from PEAK-System, e.g. PCAN-View or PCAN-Explorer. Trace files by PEAK-System can be of different versions:



## **Version 1.0**

- created by PCAN-Explorer 3.0, PCAN-Trace 1.0

## **Version 1.1**

- created by PCAN-Explorer 3.0.2, PCAN-Explorer 4, PCAN-Trace 1.5, PCAN-View (for CAN 2.0 connections), PCAN-Basic (for CAN 2.0 connections)
- \$STARTTIME keyword to store the absolute start time of the trace file
- "Type" column (Rx, Tx, Warng, Error)

## **Version 1.2**

- created by PCAN-Explorer 5 Beta
- time offset has resolution of 1 microsecond
- "Bus" column

## **Version 1.3**

- created by PCAN-Explorer 5, PCAN-Explorer 6 (for CAN 2.0 connections)
- "Reserved" column
- J1939 support
- header section contains a table of connections/busses

## **Version 2.0**

- created by PCAN-View 4 and 5 (for CAN FD connections), PCAN-Basic (for CAN FD connections)
- supports CAN FD messages
- \$COLUMNS keyword
- separate Type and Direction (Rx/Tx) columns

## **Version 2.1**

- created by PCAN-Explorer 6 (for CAN FD connections)
- optional Bus column "B"

- optional Reserved column "R"
- J1939 support (compared to version 2.0)
- EV (Event) message type
- Data Length/Data Length Code columns are not empty for message types ST (Hardware Status change), EC (Error Counter change), ER (Error Frame)

## 3.5 PEAK binary trace (\*.btrc)

This is a trace file in a proprietary binary format. In contrast to the older binary trace format (\*.btr) it is not intended to contain CAN data only, but to store various data types like LIN, other protocols, and/or information. The PCAN-Diag FD is currently the only device using this format for recording CAN traffic.

A PEAK binary trace file is arranged in blocks (normally 4 kByte per block) in which several channels can store different data types. PEAK binary trace files can be of different versions. At this time, a single version exists:

### Version 1.0

- first official release version of the PEAK binary trace format
- used by the PCAN-Diag FD

## 3.6 Measurement Data Format 4 (\*.mf4)

A third-party binary trace format maintained by the Association for Standardization of Automation and Measuring Systems (ASAM). It is a standard intended as a general purpose file format for all kinds of measured data. Coming from an automotive background, this standard is suited for all phases of the automotive development life cycle. Using the standard to store measured data provides access to a large range of tools that are capable of accessing the data stored in MDF4 files.

# 4 Error Frames in PCAN Trace Files

In a PCAN trace file, all relevant information of Error Frames is contained in either the CAN ID and 4 data bytes (format version 1.x) or in 5 data bytes (since format version 2).

Position in trace format version		Contained information	Possible values and their meaning
1.x	2.x		
CAN ID	Data byte 0	Type of error frame	1 = Bit error 2 = Form error 4 = Stuff error 8 = Other error
Data byte 0	Data byte 1	Direction	0 = Error occurred while transmitting 1 = Error occurred while receiving
Data byte 1	Data byte 2	Current position in bit stream	2 = ID.28 to ID.21 3 = Start of frame 4 = Bit SRTR 5 = Bit IDE 6 = ID.20 to ID.18 7 = ID.17 to ID.13 8 = CRC sequence 9 = Reserved Bit 0 10 = Data field 11 = Data Length Code 12 = Bit RTR 13 = Reserved Bit 1 14 = ID.4 to ID.0 15 = ID.12 to ID.5 17 = Active Error Flag 18 = Intermission 19 = Tolerate dominant bits 23 = Error delimiter 24 = CRC delimiter 25 = Acknowledge slot 26 = End of frame 27 = Acknowledge limiter 28 = Overload Flag

Position in trace format version		Contained information	Possible values and their meaning
1.x	2.x		
Data byte 2	Data byte 3	Receive Error Counter	Current value of the Receive Error Counter
Data byte 3	Data byte 4	Transmit Error Counter	Current value of the Transmit Error Counter

# 5 Session Information

A binary trace file (\*.btr) can contain several traces. These are called sessions. The following information is shown for sessions that can be selected for conversion.

## 5.1 Kind of Session

During recording of CAN traffic, the trace file was handled in one of the following ways:

- Linear: The recording stops when the trace file is full.
- Circular: When the trace file is full, the recording continues at the beginning of the trace file.

## 5.2 Sectors Count, First/Last Sector

A binary trace file's structure consists of sectors. Each sector contains 512 bytes and can hold up to 8 CAN messages. Sectors are numbered starting with 0. The first sector (0) of a binary trace file contains management information, thus, the sectors that hold CAN data start at sector number 1.

# 6 Batch Conversion

You have different possibilities to convert several files at once without user interaction.

## 6.1 Drag and Drop

### Preparation

1. Run the PEAK-Converter and open the *General Options* dialog box (via the three-lines icon in the upper right area of the window).
2. Customize the default options and leave the program.
3. On the Windows Desktop, create a shortcut icon to PEAK-Converter.exe.

### Action

If you drag and drop one or more files onto the shortcut icon for the PEAK-Converter, those files are directly converted. This process uses the options as set before.



You may also drag a folder onto the shortcut icon. PEAK-Converter tries to convert each file contained in that folder.

## 6.2 Command Line

If you use scripting or just are comfortable using the command line, PEAK-Converter.exe can be run with several file names as parameters. Those files are then directly converted.

Furthermore, parameters are available to customize the behavior per run.

For more information, see chapter 7 *Command Line Usage*.

# 7 Command Line Usage

For batch files or scripting in general, the PEAK-Converter can be called in a command line with parameters. Hence, the user interface isn't shown and files are converted right away without user interaction.

## 7.1 Command Line Format

```
PEAK-Converter.exe "[Path\]SourceFileName" /Param1=Value]
/Param2 [=Value] ...
```

An overview of available command line parameters is shown with the following parameter: /?

## 7.2 Source File

### Path

- Any path available in Windows can be used (with backslashes as directory separators).
- There must not be a backslash at the end of the path.
- If the path is omitted, the program looks for the given file name in the current working directory.



If you enter a path without a source file name, PEAK-Converter tries to convert each file contained in that folder (batch conversion).

## Source File name

- One or several files can be indicated.
- Wildcards (like `tr*. *`) are not possible.

## 7.3 Parameters

- A parameter can be indicated as complete word or abbreviated.
- It is not distinguished between upper and lower case letters.
- A value is only needed for specific parameters (see below).
- If a parameter is omitted in the command line, the corresponding preset from the *General Options* dialog box in the user interface is used.

`/?`

An overview of available command line parameters is shown. No further action is done.

**`/CsvSeparator (/CS)`**

The character that is used to separate items within the CSV file if the latter is used as target format (see `TargetFormat` parameter).

Example: `/CS,`

**`/TargetFormat (/TF)`**

The target format for the converted file(s). Possible formats are:

- PCAN: A text-based trace format by PEAK-System.
- ASC: A text-based trace format used by third-party tools.
- CSV: A common format to exchange tabular data. CSV means Comma-Separated Values. The separator character can be altered with the `CsvSeparator` parameter.
- BTR: A binary trace format from PEAK-System to be used for recording and playback of CAN data.



- BTRC: A binary trace format from PEAK-System to be used for recording and playback of different data types like CAN, LIN, other protocols, and information.
- MF4: A third-party binary trace format maintained by the Association for Standardization of Automation and Measuring Systems (ASAM).

For more information, see chapter 3 *Trace Files Formats*.

Example: /TF=PCAN

### **/TargetVersion (/TV)**

Specific version to be used for the target format.

Currently the following formats have version support:

- PCAN: 1.0, 1.1, 1.2, 1.3, 2.0, and 2.1
- BTR: 1, 2, and 3 (latter appropriate for most cases)
- BTRC: 1
- ASC: 7, and 8.1
- MF4: 4.1, 4.11, and 4.2

For more information, see chapter 3 *Trace Files Formats*.

Example: /TV=1.2

### **/TargetDirectory (/TD)**

The directory where the target file(s) are written. This directory must already exist. There must not be a backslash at the end of the path. Use quotation marks around the path if it has spaces.

Example: /TD="C:\Users\MySelf\Desktop\Converter Test"

### **/TargetFileName (/TN)**

The base name of the target file(s). This name must not contain path information nor a file name extension. If this parameter is omitted, the name of the source file is used.

Example: /TN="MyConvertedData"

### **/Sessions (/SN)**

If the input file is of the BTR format (binary trace), it may contain several sessions. By default, all sessions are converted. If only specific sessions are to be converted, they can be indicated with this parameter, each separated with a comma (,).

Example: /SN=1,3,4

### **/DateTimeInFileName (/DT)**

Add the date and time info to the name of the target file.

Example: /DT

### **/MDF4Compression (/MC)**

Enable the compression of data while converting to MDF4, using MDF4 DZ (data zipped) blocks.

Example: /MC

### **/OverrideExisting (/OE)**

If the target file already exists, it is overwritten.

Example: /OE

### **/WindowsShutdown (/WS)**

Windows is shut down after the conversion process. This can be helpful when very large files are converted.



Make sure that any open document or other data in Windows are saved before using this parameter.

Example: /WS

# 7.4 Return Values (Errorlevel)

The success of a command line call can be checked by using the Errorlevel code that is returned by the PEAK-Converter. Following return values make up the Errorlevel:

Designation	Value	Description
Success	0	One or more valid trace files have been found and all were converted successfully.
NoJobs	1	Neither a file nor a directory with files have been found for conversion.
ArgumentsError	2	An error has been found in the command line arguments, e.g. a trailing backslash in a path.
SomesConverted	4	Several traces were converted, but at least one conversion failed or was canceled.
NoneConverted	8	The conversion of all files failed or was canceled.
ShutdownFailed	16	The /WindowsShutdown parameter was used but could not be enabled by the program. This return value is added to any other possible value.

# 7.5 Example

```
PEAK-Converter.exe "C:\Users\Admin\Desktop\Test_123.btr"  
/SN=1,3 /TF="CSV" /CS=";" /TD="C:\Users\Admin\Desktop\TEST  
Converter"
```

(in a single line)

This command line uses sessions 1 and 3 from the binary trace file `Test_123.btr` and converts them to a CSV file each (semicolon as item separator within the file). The two resulting output files are written to the Test Converter directory on the Windows Desktop using `Test_123` as file name.