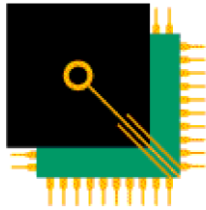


# CANopen File Player User Manual

Manual Revision 1.00



EMBEDDED  
SYSTEMS  
ACADEMY

Information in this document is subject to change without notice and does not represent a commitment on the part of the manufacturer. The software described in this document is furnished under license agreement or nondisclosure agreement and may be used or copied in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the software purchaser's use, without prior written permission.

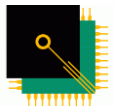
Every effort was made to ensure the accuracy in this manual and to give appropriate credit to persons, companies and trademarks referenced herein.

© Embedded Systems Academy, Inc. 2015, All Rights Reserved

Microsoft® and Windows™ are trademarks or registered trademarks of Microsoft Corporation.

PC® is a registered trademark of International Business Machines Corporation.

CANopen® is a registered trademark of CAN in Automation User's Group.



CANopen File Player was developed by Embedded Systems Academy. Embedded Systems Academy provides training and consulting services, specializing in CAN and CANopen. For more information visit

EMBEDDED  
SYSTEMS  
ACADEMY

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

---

# Contents

Contents .....	3
About This Manual.....	4
Chapter 1 – Introduction .....	5
1.1 About CANopen .....	5
1.2 About CANopen File Player .....	5
Chapter 2 – Installation .....	7
2.1 Installation.....	7
Chapter 3 – Quick Start .....	8
3.1 Introduction .....	8
3.2 Connection.....	8
3.3 Playing a File.....	9
3.4 Errors.....	9
Chapter 4 – Additional Functionality .....	10
4.1 Concise DCF Settings .....	10
Appendix A – Concise DCF CSV Files.....	11
A.1 Overview .....	11
A.1 File Format .....	11
A.2 Data Types .....	12
A.3 Object Dictionary Write Value.....	12
A.4 Object Dictionary Write from Buffer.....	12
A.5 Object Dictionary Read .....	12
A.6 File Information .....	13
A.7 Settings.....	13
A.8 Pausing .....	13
A.9 Waiting.....	14
A.10 Network Management.....	14
A.11 LSS Master.....	14

## About This Manual

This manual follows some set conventions with the aim of making it easier to read. The following conventions are used:

0x	Hexadecimal (base 16) values are prefixed with "0x".
<i>italic text</i>	Replace the text with the item it represents
[ ]	Items inside [ and ] are optional
a   b	a OR b may be used
...	One or more items may go here.
a -> b	Choose item a from a menu followed by item b

This manual frequently uses CANopen terminology as defined by the CANopen standard CiA301 (see [www.can-cia.org](http://www.can-cia.org) for more info). Readers that are not yet familiar with all the CANopen terms may want to consider reading a book like [www.canopenbook.com](http://www.canopenbook.com) or the official standard to update their knowledge on CANopen technology and terminology.

# Chapter 1 – Introduction

## 1.1 About CANopen

CANopen is a higher layer protocol that runs on a CAN network. The CAN specification defines only the physical and data link layers in the ISO/OSI 7-layer Reference Model. This means that only the physical bus and the CAN message format is defined, but not how the CAN messages should be used. CANopen provides an open and standardized but customizable description of how to transfer data of different types between different CAN nodes. This allows off the shelf CANopen compliant nodes to be purchased and plugged into a network with the minimum of effort. It also can be used in place of an in-house proprietary higher layer protocol development.

The development of CANopen is supervised by the CAN in Automation User's Group and is being turned into an international standard. Use of CANopen does not require the payment of any royalties and the specification may be expanded or altered to suit if closed networks are being developed.

Typical applications for CANopen include:

- Commercial Vehicles
- Medical Equipment
- Maritime Electronics
- Building Automation
- Light Rail Systems

## 1.2 About CANopen File Player

CANopen File Player is a utility for real-time playback of CAN messages and for playing of sophisticated configuration files. It requires a PEAK CAN interface. Supported formats include:

- PEAK PCAN-View log files
- Vector CANalyzer log files
- ESAcademy CANopen Magic log files
- Concise Device Configuration Files
- Concise DCF CSV Files

This allows quick and easy reproduction of background traffic for standardizing testing, analysis of previous network operation and debugging of networks. Files can be played at full speed or singles-stepped through to play one message at a time.

The Concise DCF CSV file option allows sophisticated test sequences to be constructed, including sequences for interacting with bootloaders, with support for:

- NMT

- SDO read/write
- SDO read to a buffer and write from a buffer
- LSS configuration
- Wait for bootup/heartbeat
- Pausing
- Setting SDO and back-to-back timeouts, number of retries

# Chapter 2 – Installation

## 2.1 Installation

To install the software run the installer and it will guide you through the steps. In addition you will need to install the drivers supplied with your CAN interface and the Microsoft .NET Framework 4.0 or later. If you don't have the .NET framework the installer will automatically download and install it for you.

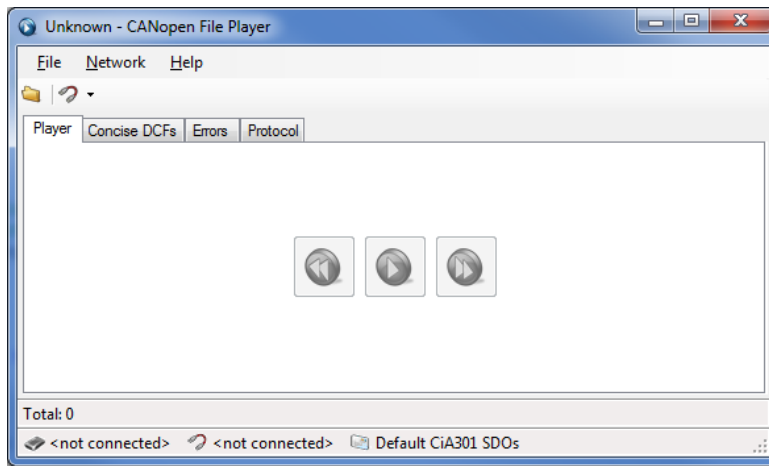
# Chapter 3 – Quick Start

## 3.1 Introduction

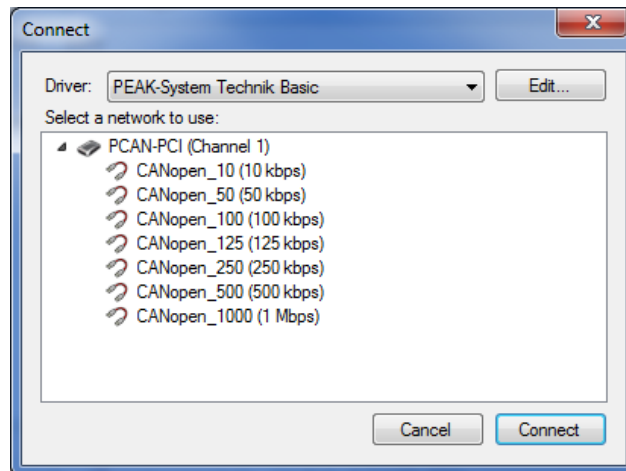
This quick start guide introduces key features of software in a step by step manner. Further details of the features mention are given later in this manual.

## 3.2 Connection

When the application is started the main window is displayed.



Connect to your CAN interface by clicking on the Connect toolbar button.



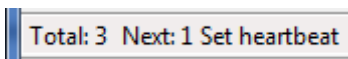
Choose your CAN interface vendor, then interface, then a bus speed. Click on Connect. The current connection settings are always shown at the bottom of the main window.



## 3.3 Playing a File

Choose a file to play by clicking on the Open toolbar button. If the file contains messages grouped into “networks” then you will be prompted to choose a “network”.

The status bar at the bottom of the window shows how many messages (or actions) are in the file and which message (or action) will be performed next.



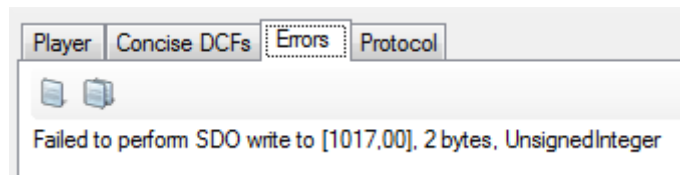
Under the Player tab click on the play button to play the file. If it is a log file then the messages will be played in real-time onto the CAN bus. If it is a CDCF or CSV CDCF file then the actions will be performed one by one.



To pause playback click on the stop button. To single-step first pause playback then click on the next button repeatedly. To go back to the beginning click on the rewind button. If playback reaches the end of the file it will automatically go back to the beginning and stop.

## 3.4 Errors

Any errors during playback will be shown under the Errors tab.

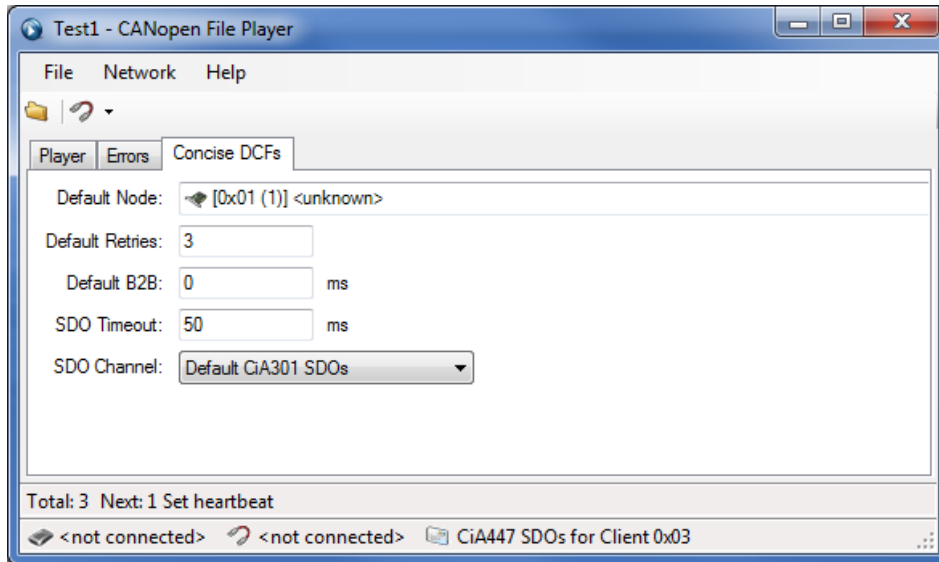


Click on the Clear toolbar button to clear this window. Click on the Copy button to copy the errors to the clipboard.

# Chapter 4 – Additional Functionality

## 4.1 Concise DCF Settings

The Concise CDF files (CDCF and CSV) have settings available under the “Concise DCFs” tab. Changes to these settings take effect when playing of a CDCF file is started from the beginning of the file.



If the CDCF file does not specify a node id to use then the setting will be taken from here.

The “default B2B” time stops transmission of messages that are back-to-back too quickly. A delay is inserted between back-to-back messages.

All SDOs transmitted by a CDCF use the SDO channel selection.

# Appendix A – Concise DCF CSV Files

## A.1 Overview

Concise DCF files are binary files that describe a set of writes to object dictionary entries. Embedded Systems Academy has expanded this file format to allow additional functionality. In addition the CANopen File Player can use CSV files as a user-friendly text based version of Concise CDF files. Playback of binary CDCF and text-based CSV CDCF files is identical in functionality.

## A.1 File Format

The CSV files can be created in any spreadsheet application or even a text editor. Here is an example.

	A	B	C	D	E	F
1	Comment	Command	Index	Subindex	Type	Data
2	Name	FILE_INFO			STRING	Clear heartbeat and error list
3	Version	FILE_VERSION			STRING	V1.00 of 20-JUN-15
4		SETTING_NODE_ID			UNSIGNED8	0x40
5		SETTING_SDO_TIMEOUT			UNSIGNED16	500
6		SETTING_RETRIES			UNSIGNED8	8
7	Set heartbeat		0x1017	0x00	UNSIGNED16	0x0000
8		SETTING_NODE_ID_ADD			INTEGER8	1
9	Set heartbeat		0x1017	0x00	UNSIGNED16	0x0000
10		SETTING_NODE_ID_ADD			INTEGER8	-1
11	Set heartbeat		0x1017	0x00	UNSIGNED16	0x0000

There are six columns:

Name	Description
Comment	A user-friendly description for the line. This is displayed in the status bar of the player
Command	An optional command. Blank if index and subindex are specified
Index	An optional object dictionary index to access. Blank if a command is used
Subindex	An optional object dictionary subindex to access. Blank if a command is used
Type	Data type
Data	Data value. The meaning depends on the command or index/subindex

Each line must have either a command or an index and subindex pair. It cannot have both.

Values can be in decimal or hexadecimal. Hexadecimal values must be prefixed with "0x" or "0X" or suffixed with "H" or "h".

## A.2 Data Types

The following data types are supported.

Type	Description
UNSIGNED8	Unsigned eight-bit value
INTEGER8	Signed eight-bit value
UNSIGNED16	Unsigned 16-bit value
UNSIGNED24	Unsigned 24-bit value
UNSIGNED32	Unsigned 32-bit value
UNSIGNED40	Unsigned 40-bit value
UNSIGNED48	Unsigned 48-bit value
UNSIGNED56	Unsigned 56-bit value
UNSIGNED64	Unsigned 64-bit value
STRING	String
DOMAIN	The path and name of a binary file. If the path is relative then it is assumed relative to the location of the CSV CDCF file Or the file filter and description of a file to obtain from the user separated by a vertical bar ' ', e.g. "*.bin Firmware files"
LSSMASTERRECORD	Four unsigned 32-bit values in hexadecimal followed by an unsigned eight-bit value in hexadecimal, separated by spaces

## A.3 Object Dictionary Write Value

To write to the object dictionary add a line that specifies the index, subindex, data type and value to write.

4	Set heartbeat	-	0x1017	0x00	UNSIGNED16	200
---	---------------	---	--------	------	------------	-----

## A.4 Object Dictionary Write from Buffer

To write to the object dictionary from the internal buffer two lines are needed. The first line contains a command that tells the player the write is from the internal buffer rather than a fixed value.

7		CONTROL_SDO_BUFFER			UNSIGNED8	0
8	Write heartbeat time from buffer		0x1017	0x00	UNSIGNED16	0

For both lines the Data column is not used.

## A.5 Object Dictionary Read

To read an object dictionary entry two lines are needed. The first line contains a command that tells the player the second line is a read not a write.

5		CONTROL_SDO_READ			UNSIGNED8	1
6	Read heartbeat time		0x1017	0x00	UNSIGNED16	200

The value for the CONTROL\_SDO\_READ command affect how the read is performed. It always has the UNSIGNED8 type.

Value Bit	Description
0	When set the value read must match the value on the following line
1	When set if the value read doesn't match the value on the following line then the player will wait and try again
7	When set the value read will be copied to an internal buffer

## A.6 File Information

A set of commands are not executed, but instead provide information about the file. Their values are always STRING.

Command	Type	Description
FILE_INFO	STRING	A description of what the file does.
FILE_VERSION	STRING	Version number and date of the file

## A.7 Settings

Settings are commands that change the configuration of the file player. These override the settings made in the user interface. They may appear anywhere in a file and when they appear in the middle of a file they immediately change the setting for the rest of the file, or until the setting is changed again.

Command	Type	Description
SETTING_B2B_TIMEOUT	UNSIGNED16	Back-to-back minimum time in milliseconds
SETTING_NODE_ID	UNSIGNED8	The ID of the node to access. Zero to use the setting in the user interface.
SETTING_NODE_ID_ADD	INTEGER8	A positive or negative value to add to the current node ID
SETTING_RETRIES	UNSIGNED8	Maximum number of retries
SETTING_SDO_TIMEOUT	UNSIGNED16	Maximum SDO timeout in milliseconds

## A.8 Pausing

The command CONTROL\_PAUSE has the type UNSIGNED16 and pauses playing of the file for the specified number of milliseconds.

## A.9 Waiting

The command CONTROL\_WAIT\_FOR has the type UNSIGNED8 and waits for a specific bootup or heartbeat message to appear on the bus.

Command Value	Description
0x00	Waits for a bootup message
0x05	Waits for an operational heartbeat message
0x7F	Waits for a pre-operational heartbeat message
0xFF	Waits for any type of heartbeat message

## A.10 Network Management

The command CONTROL\_NMT has the type UNSIGNED16 and sends a network management message.

The upper byte of the Data value for the command specifies the node ID to send the NMT command to. A value of zero means "all nodes".

The lower byte of the Data value for the command specifies the type of NMT command. The values are taken directly from CiA301, e.g. 0x81 = reset.

## A.11 LSS Master

The command CONTROL\_LSSM has the type LSSMASTERRECORD and configures a single LSS slave on the network. It sends the configuration and start commands to the unconfigured slave that has the matching identity value as defined in the Data column for this command. The LSS slave is assigned the node id specified in the Data column for this command.