PCANopen Magic
User Manual

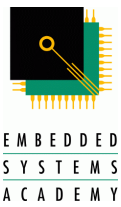
Manual Revision 1.13

PEAK System Technik

For support contact support@esacademy.com

For the latest news on PCANopen Magic visit PEAK-System Technik at

www.peak-system.com



PCANopen Magic was developed for PEAK-System Technik by Embedded Systems Academy. Embedded Systems Academy provides training and consulting services, specializing in CAN, CANopen and Embedded Internetworking. For more information visit

www.esacademy.com

Contents

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".
italictext	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.

This manual frequently uses CANopen terminology as defined by the CANopen standard DS301 (see 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 or the official standard to update their knowledge on CANopen technology and terminology.

Chapter 1 - Introduction

PCANopen Magic is a low cost easy to use but sophisticated utility for accessing, monitoring and controlling nodes on a CANopen Network. It allows read and write accesses to the process data and the configuration variables of CANopen nodes. It works with multiple low cost CAN interfaces. The key features of PCANopen Magic are:

- Network Management of select nodes or all nodes. A single node or all nodes may be placed in Operational, Pre-Operational or Stopped states, or Reset.
- Data Download to a selected node. Manually entered data or data stored in a file may be written to a specific node. Files up to 160kb in size are supported, allowing transfer of Intel Hex Files for reprogramming or calibration/configuration tables.
- Data Upload from a selected node. Data up to 160kb in size may be read from a specific node and either displayed in the PCANopen Magic window or stored in a file.
- Transmission of CAN messages either on a key press or in response to reception of specific messages.
- Indication of the current state of a selected node. PCANopen Magic listens for Heartbeat and Node Guarding messages from a specific node and displays the reported state of the node.
- Indication of the last emergency transmitted by a selected node. PCANopen Magic listens for emergency messages being transmitted from a specific node and displays the last emergency detected.
- Trace window showing messages on the CAN bus. A trace window may be opened showing the current messages on the CAN bus. The trace window understands CANopen and displays detailed information on the meaning of CANopen messages, allowing for CANopen nodes and networks to be debugged. The trace recording may be stopped and started and when stopped provides viewing of up to the last 13000 messages received. The trace window also displays the rate of messages on the CAN bus.
- Exporting of trace window contents to Comma Separated Values files.
- Indication of the bus load, errors and transmit and receive activity. LEDs in the PCANopen Magic window indicate when PCANopen Magic is transmitting and receiving. An LED also indicates the bus load of the CAN bus by lighting for each message detected. Another LED is provided to indicate major CAN bus errors, such as Bus Off.
- Supports multiple CAN interfaces:
    - PEAK PCAN Parallel Port Dongle
    - PEAK PCAN USB

  - PEAK PCAN PCI
  - PEAK PCAN ISA
  - SYSTEC USB-CANmodul

- Does not require Electronic Datasheet Files.

- Command Line Interface. PCANopen Magic may be run from DOS or the Command Line to read data from nodes, write data to nodes and perform Network Management. This allows PCANopen Magic to be controlled from custom applications or batch files, allowing complex sequential accesses to the Object Dictionary to be made.

- Network Overview feature scans for all nodes on the network then shows static and real-time information about each node.

PCANopen Magic is ideally suited for:

- Development of a CANopen slave node

  - Test and debug single object dictionary accesses
  - Test and debug segmented transfers of large data entries
  - Test and debug access sequences using the batch/script mode
  - Constructing an Integrated CANopen Development Environment

- Final test of CANopen nodes

  - Use the batch/script mode to execute complex test sequences

Chapter 2 - Installation

2.1 Install Drivers

Before PCANopen Magic may be used with a CAN interface, the drivers for the interface must be installed. The drivers may be installed before or after PCANopen Magic is installed, and were supplied with the interface. Please follow the directions given with the driver to install it.

2.2 Install PCANopen Magic

Install PCANopen Magic by running the executable file. Once installation is completed PCANopen Magic may be accessed from the Start Menu under Programs.

2.3 Activation

Once installed PCANopen Magic requires activation to turn it into a licensed version. If PCANopen Magic is not activated then it will run unrestricted only 20 times. On the 21st start PCANopen Magic will introduce limitations.

To purchase a license and obtain an activation code, please contact your local distributor or visit PEAK-System Technik at http://www.peak-system.com.

The current number of starts is shown on the splash screen and in the Help About window.

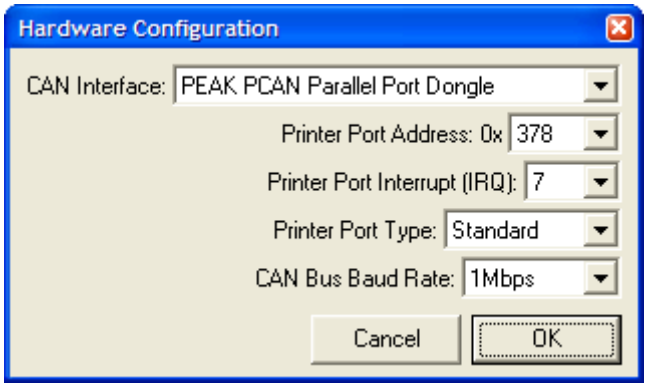
To activate PCANopen Magic complete the following steps

- Start PCANopen Magic and configure your hardware (see the next section)
- Open the Help menu
- Choose Activate…
- Enter your activation code into the box. The code is case insensitive. Do not include spaces.
- Enter your name into the box.
- Optionally enter your company name into the box.
- Click on Activate

The splash screen and Help About window will now show your license number, name and company name.

Chapter 3 - Hardware Configuration

When PCANopen Magic is first started a window will open allowing a CAN interface to be configured. A drop down list allows selection of the CAN interface to use. The configuration options displayed will vary and depend on the CAN interface selected.



1. Hardware Configuration

Please refer to the CAN interface manual for details on how to configure the CAN interface.

Select the Baud Rate to use for the CAN Bus. All CANopen Baud Rates are supported.

Note that PCANopen Magic remembers the configuration, so normally the configuration will only have to be made the first time PCANopen Magic is started.

Clicking on Cancel will close PCANopen Magic. Clicking on OK will attempt to configure the CAN interface according to the selections made. If the configuration is successful then the main PCANopen Magic window will open.

Chapter 4 - Main Window Overview

The main PCANopen Magic window is divided into three sections.

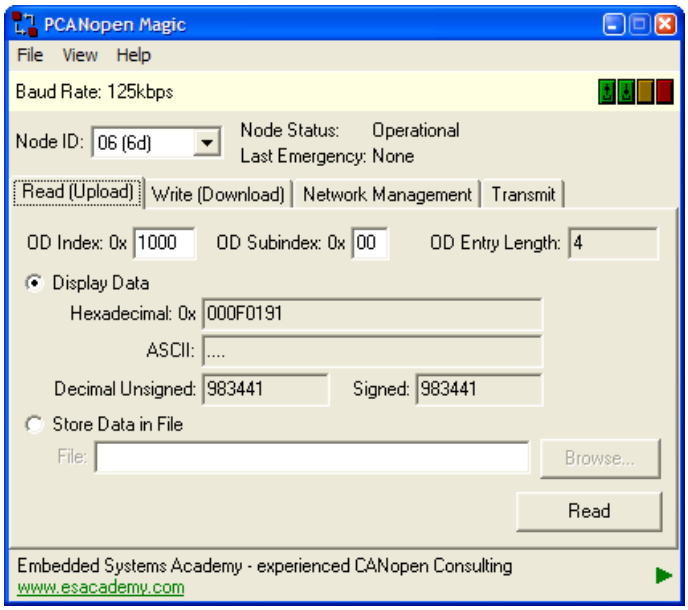
The first section displays information relating to the CAN bus. The current baud rate is shown along with LEDs that indicate when data is being transmitted by PCANopen Magic (green), received by PCANopen Magic (green) or detected on the CAN bus (orange). A final LED (red) indicates if a major bus error has occurred.

The second section provides information on the currently selected CANopen node. The ID of the node to select may be selected using the drop-down list provided. Once selected, the current node status will be displayed if detected, and the last emergency transmitted by that node will be displayed, if any.

The final section contains the controls to read and write from the selected node and to transmit Network Management messages to either the selected node or all nodes. The controls are divided into the following three sections and are accessible by clicking on the corresponding tab:

- Read (Upload). Read data from the selected node.
- Write (Download). Write data to the selected node.
- Network Management. Transmit Network Management messages to either the selected node or all nodes.

Any boxes for entering data that are prefixed with "0x" require hexadecimal values to be entered.



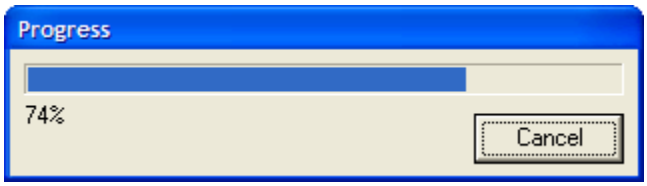
2. Main PCANopen Magic Window

Chapter 5 - Uploading (Reading) from a Node

To upload data from a node:

- Click on the Read (Upload) tab.
- Select the ID of the node to read from the drop-down list in the main window.
- Enter the Index and Subindex of the Object Dictionary entry to upload from into the boxes provided. For example, to read Object Dictionary entry 0x1018 subindex 0x01, enter "1018" into the Index box and "01" into the Subindex box.
- Select Display Data to display the data uploaded in PCANopen Magic.
- or select Store Data in File to store the data uploaded in a file.
    - Enter the path to the file to create or overwrite or click on the Browse button to browse to a file to create or overwrite
- Click on the Read button

During the data transfer the progress window will open and show the progress of the transfer. Click on the Cancel button to cancel the transfer

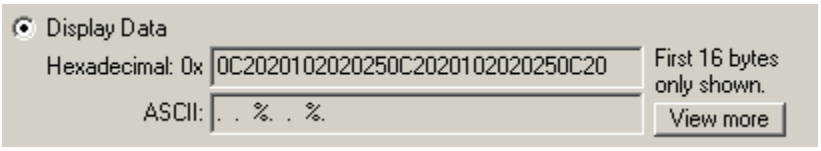


3. Progress Window

If Display Data was selected and the length of the data was 16 bytes or less, then the data will be displayed in Hexadecimal, ASCII and unsigned Decimal formats in the main window. The data will be displayed in little-endian format (least significant byte first) unless the data is four bytes or less, in which case it will be displayed in big-endian format (most significant byte first). If a byte is an unprintable character or not an ASCII character then a "." will be displayed in the ASCII format in place of that byte.

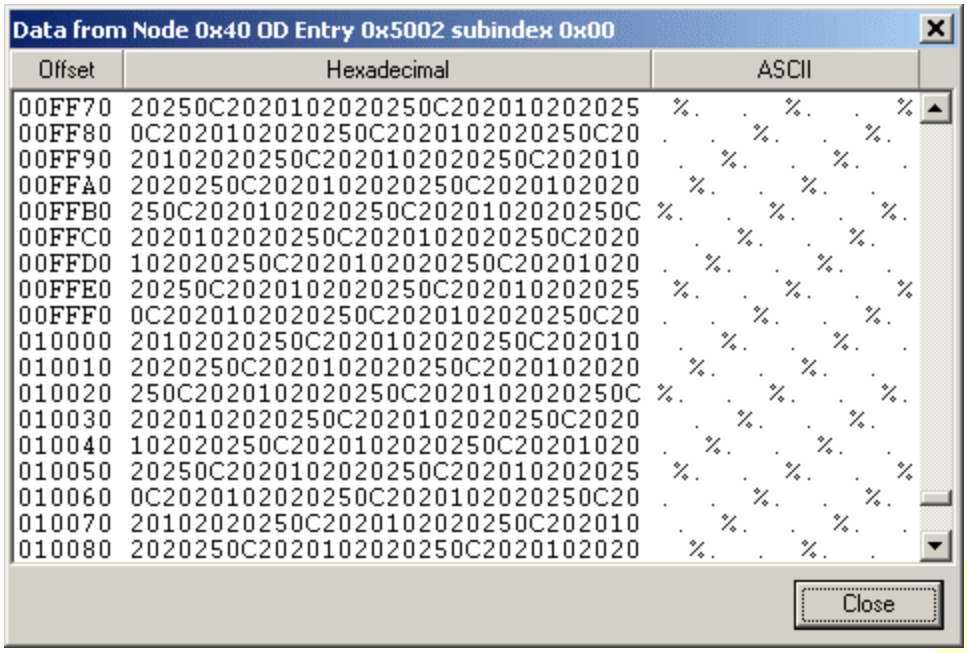
4. Displaying Less Than 17 Bytes

If display data was selected and the length of the data was more than 16 bytes, then the first 16 bytes will be displayed in Hexadecimal and ASCII formats in the main window in little-endian format. If a byte is an unprintable character or not an ASCII character then a "." will be displayed in the ASCII format in place of that byte.



5. Displaying More Than 16 Bytes

A button called View More will be displayed that when clicked on will open a window and display the entire data transferred, including the first 16 bytes displayed in the main window.



6. Displaying Data

The title bar of the window will show the node ID and Object Dictionary entry the data was read from.
Each line displays 16 bytes of data using three columns. The offset column shows the offset in Hexadecimal of the start of the line from the start of the data. The Hexadecimal column shows the data in Hexadecimal and the ASCII column shows the data in ASCII. If a byte is an unprintable character or not an ASCII character then a "." will be displayed in the ASCII format in place of that byte.

The size of the Object Dictionary entry is displayed after the read is completed.

Note that if a node responses with an Initiate SDO Upload Response where the data size is not specified for a segmented transfer, then PCANopen Magic will assume that the

maximum size it can support will be transferred, which is 160000 bytes. Exceeding this limit will cause PCANopen Magic to abort the transfer. Early termination of the transfer by the node using the C flag will not generate an error, however PCANopen Magic will still assume the maximum amount of data has been transferred.

Chapter 6 - Downloading (Writing) to a Node

To download data to a node:

- Click on the Write (Download) tab.
- Select the ID of the node to write to from the drop-down list in the main window.
- Enter the Index and Subindex of the Object Dictionary entry to download to into the boxes provided. For example, to write to Object Dictionary entry 0x1018 subindex 0x01, enter "1018" into the Index box and "01" into the Subindex box.
- Select Enter Data to manually enter the data to write.
    - Select from the drop-down list the base to use for the data format (Hexadecimal, Decimal or ASCII)
    - Enter the data into the box in the chosen format. If entering hexadecimal data four bytes or less in size then enter the data in big-endian format (most siginificant byte first). If entering more than four bytes of data then enter the data in little-endian format (least significant byte first).
      For example to transfer the four byte value 0x12345678, enter "12345678" into the box. To transfer the five bytes 0x12 0x34 0x56 0x78 0x90 with 0x12 transmitted first, enter "1234567890" into the box. To transmit the five byte value 0x1234567890, enter "9078563412" into the box.

      If entering decimal data, then positive or negative values may be entered. All positive values will be assumed to be Unsigned. All negative values will be assumed to be Signed.

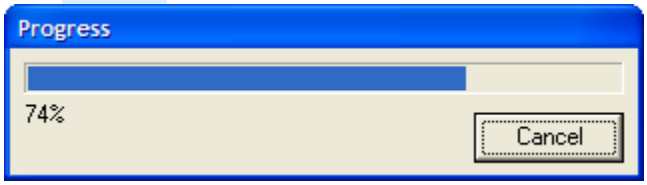
      Up to 100 bytes of data may be entered.
    - Enter the length of the data to write in bytes. This can be more than the data entered, in which case the data is padded to the desired size. Hexadecimal data is padded with 0x00. Unsigned Decimal data is padded with 0x00. Signed Decimal data is padded with 0xFF. ASCII data is padded with spaces (0x20).

      The size is especially important when entering signed decimal values, as it indicates how many bytes to use to store the value – something that cannot be determined from the value itself.
- or select Write Data from File to read the data downloaded from a file.

- Enter the path to the file to read or click on the Browse button to browse to a file to read.

- Click on the Write button

During the data transfer the progress window will open and show the progress of the transfer. Click on the Cancel button to cancel the transfer



7. Progress Window

Chapter 7 - Network Management

To perform Network Management functions:

- Click on the Network Management tab.
- If you wish to manage a specific node select Single Node.
    - Select the ID of the node to manager from the drop-down list in the main window.
- or if you wish to manage all nodes, select All Nodes.
- Click on one of the five buttons to transmit a command for the node or nodes to switch to that state. For example click on Operational to place the selected node or all nodes in the Operational state. A single command will be transmitted for each click of one of the buttons.

Chapter 8 - Transmit

To configure CAN messages for transmission:

- Click on the Transmit tab
- Select a transmit message to configure from the drop-down list
- Check the Enable box to enable the message
- Enter a descriptive name for the message. This name will appear in the drop-down list, in the Transmit List window and in the Trace window.
- If you want to transmit a message with a specific ID then select the relevant radio button and enter the ID into the box. If you want to transmit a PDO using the default connection set, then select the relevant radio button, enter the Node ID that defines the PDO and select the PDO from the drop-down list.
- To transmit the message on a key press, check the relevant checkbox and choose the key from the drop-down list
- To transmit the message on reception of a message with a specific identifier, check the relevant checkbox and enter the message identifier into the box.
- Select the length of data in the message from the drop-down list
- Enter the data for the message into the boxes

To transmit CAN messages:

- Click on the Transmit Now button to immediately transmit a single copy of the message
- If the message has been configuring to transmit on a key press, then press the key while the PCANopen Magic window has the input focus to transmit a single copy of the message
- If the message has been configured to transmit on reception of a message with a specific identifier, then the message will be automatically transmitted whenever the relevant message is received.

If more than one message is configured to trigger on the same key press or on reception of the same message identifier, then the messages will be transmitted in order starting with message 0. This allows burst transfers to be configured on a single key press or on reception of a single message.

By configuring messages to transmit on reception of message ID 0x80, PDOs transmitted in response to a SYNC message can be simulated.

Messages transmitted by PCANopen Magic, either using the Transmit section or by using one of the other sections (Upload, Download or Network Management) cannot be used to trigger transmission of a message by identifier.

All settings are automatically saved when PCANopen Magic is closed.

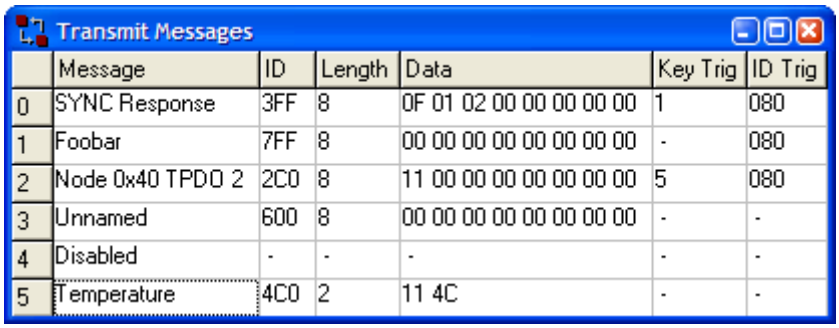
A summary of all the transmit messages configured may be viewed by opening the Transmit List window. See the following section for details.

Chapter 9 - Transmit List Window

The transmit list window may be opened by the following method:

- Open the View menu
- Select Transmit List…

The transmit list window may be left open while PCANopen Magic is used. For example the transmit list window may be opened then the configuation of the transmit messages may be changed. The trace window will then update to show the new configuration. The transmit list window may also be left open while the trace window is being displayed.



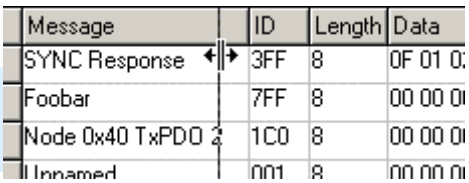
	Message	ID	Length	Data	Key Trig	ID Trig
0	SYNC Response	3FF	8	0F 01 02 00 00 00 00 00	1	080
1	Foobar	7FF	8	00 00 00 00 00 00 00 00	-	080
2	Node 0x40 TPDO 2	2C0	8	11 00 00 00 00 00 00 00	5	080
3	Unnamed	600	8	00 00 00 00 00 00 00 00	-	-
4	Disabled	-	-	-	-	-
5	Temperature	4C0	2	11 4C	-	-

8. Transmit List Window

The transmit list window is divided into the following columns:

- Message. The Message column shows the name of the message.
- ID. The ID column shows the ID in hexadecimal that will be used when the message is transmitted. If the option to transmit a PDO using the Default Connection Set ID was selected, then this column will show the actual COB (Communications Object ID).
- Length. The Length column shows the length of data in the message.
- Data. The Data column shows the data to be transmitted in hexadecimal and byte order, starting with byte 0.
- Key Trig. The Key Trig column shows, which key press, will trigger transmission of the message. If the message does not use a key press trigger then this column will show a dash (“-“).
- ID Trig. The ID Trig column shows the ID of the message in hexadecimal which when received will trigger transmission of the message. If the message does not use the reception of ID trigger then this column will show a dash (“-“).

The columns may be resized by moving the pointer over the edge of a column heading and dragging. PCANopen Magic remembers the widths of the columns and the size of the window.



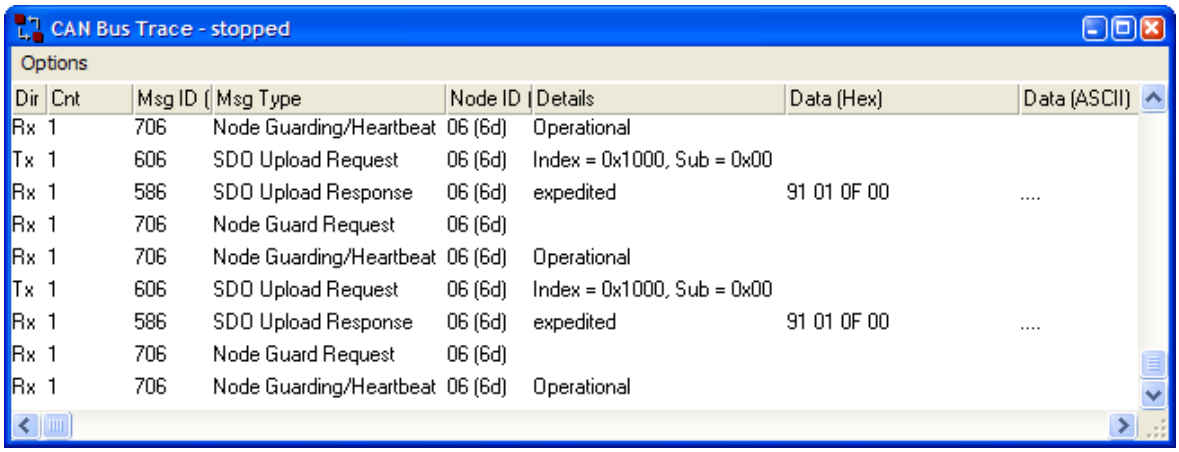
9. Column Resizing

Chapter 10 - Trace Window

The trace window may be opened by the following method:

- Open the View menu
- Select Trace…

The trace window may be left open while PCANopen Magic is used. For example the trace window may be opened then an Upload operation may be started using PCANopen Magic. The trace window will then show the SDO messages being transmitted between PCANopen Magic and the node.



10. Trace Window

The trace window shows the current messages on the bus with the most recent message at the bottom of the window. The title bar of the window shows how many messages are on the bus per second.

The trace window is divided into the following columns:

- Dir. The Dir column shows the direction of the message with respect to PCANopen Magic. Tx means PCANopen Magic transmitted the message. Rx means PCANopen Magic received the message.

- Cnt. The Cnt column shows the number of messages represented by that line. For the Continuous trace, this is always one as one message is shown per line. For the Static trace it shows the number of messages received for the message identifer in the Msg ID column.

- Msg ID (Hex). The Msg ID (Hex) column shows the identifier of the message.

- Msg Type. The Msg Type column shows the type of CANopen message, for example SDO Upload Request, SYNC, Heartbeat/Node Guarding, etc.

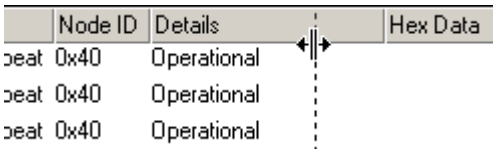
- Node ID (Hex). The Node ID (Hex) column shows the ID of the node that the message was sent to or sent from.

- Details. The Details column provides any additional CANopen related information about the message, for example the state of the toggle bit, or the textual description of an emergency message.

- Data (Hex). The Data (Hex) column shows the data transferred in the message if any in Hexadecimal format.

- Data (ASCII). The Data (ASCII) column shows the data transferred in the message if any in ASCII format. If a byte is an unprintable character or not an ASCII character then a "." will be displayed in the ASCII format in place of that byte.

The columns may be resized by moving the pointer over the edge of a column heading and dragging. PCANopen Magic remembers the widths of the columns and the size of the window.



11. Column Resizing

The trace window recording may be started and stopped by the following method:

- Open the Options Menu.

- Choose Stop Trace to stop the trace.

- Choose Start Trace to start the trace.

When the trace is stopped the previous messages received up to the last 13000 messages may be viewed by use of the scrollbar that will appear. For performance reasons the scrollbar is not displayed while the trace is running.

The trace window may filter out messages that are not relevant to the currently selected node. The currently selected node is the node whose ID is selected in the PCANopen Magic main window.
Messages that will be shown are SDOs transmitted to and from the node, SYNC messages, emergency messages and Network Management messages. The trace window may be filtered using the following method:

- Open the Options Menu.

- Choose Monitor Current Node to filter the messages.

- Choose Monitor All Nodes to stop filtering the messages.

The contents of the trace buffer may be saved in a Comma Separated Value file for processing in other applications, such as Excel. To export the trace buffer:

- Open the Options Menu
- Choose Export Ascending Trace… or Export Descending Trace…

Ascending format lists the most recent messages first. Descending format lists the oldest messages first.

The trace window may be switched between static and continuous trace. When in the continuous trace mode, messages are displayed in the trace window in the order they were received, with the most recent message at the bottom. When in the static trace mode, messages are displayed according to message identifier, with the lowest message identifier at the bottom. When a message is received with the same identifier as a message currently displayed in the trace window, the message is overwritten in the trace window with the new data, direction, data length, etc. To switch to static mode:

- Open the Options Menu
- Choose Static Trace

To switch back to the continuous mode:

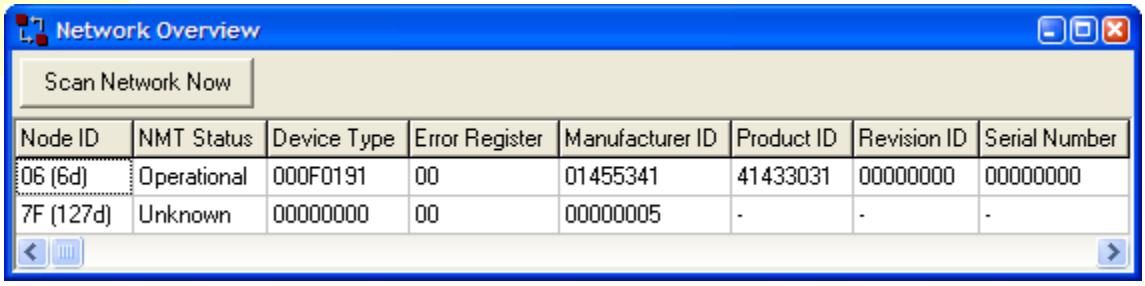
- Open the Options Menu
- Choose Continuous Trace

Chapter 11 - Network Overview Window

The Network Overview window may be opened by the following method:

- Open the View menu
- Select Network Overview…

The Network Overview window may be left open while PCANopen Magic is used. For example the Network Overview window may be opened then an Upload operation may be started using PCANopen Magic.



12. Network Overview Window

Initially the Network Window will be empty. Clicking on the Scan Network Now button will result in PCANopen Magic scanning for and detecting all the CANopen Nodes on the network. For a CANopen node to be detected it must provide error free access to Object Dictionary entry 0x1000.

During the network scan the progress window will be displayed. To cancel the scan click on the Cancel button.

Once the network scan has completed each detected node will be listed in the table. The table is divided up into the following columns. Some of the columns show static data that is only updated on a network scan. Other columns show real-time data that is continuously updated. All data is displayed in hexadecimal format.

- Node ID. The Node ID column shows the ID of the node. Static data.

- NMT Status. The NMT Status column shows the current NMT status of the node. This column shows real-time data that is updated when Node Guarding or Heartbeat messages are detected on the bus. If Node Guarding or Heartbeat messages are not detected for a specific node, then this column will display “Unknown”.

- Device Type. The Device Type column shows the value stored in Object Dictionary entry 0x1000. Static data displayed in big-endian format.

- Error Register. The Error Register column shows the value stored in Object Dictionary entry 0x1001. Static data displayed in big-endian format.

- Manufacturer ID. The Manufacturer ID column shows the value stored in Object Dictionary entry 0x1018 subindex 0x01. Static data displayed in big-endian format.

- Product ID. The Product ID column shows the value stored in Object Dictionary entry 0x1018 subindex 0x02. Static data displayed in big-endian format.

- Revision ID. The Revision ID column shows the value stored in Object Dictionary entry 0x1018 subindex 0x03. Static data displayed in big-endian format.

- Serial Number. The Serial Number column shows the value stored in Object Dictionary entry 0x1018 subindex 0x04. Static data displayed in big-endian format.

Any Object Dictionary entries that are not implemented or generated an error when an attempt was made to read them will be displayed as "-".

The columns may be resized by moving the pointer over the edge of a column heading and dragging. PCANopen Magic remembers the widths of the columns and the size of the window.

Chapter 12 - Command Line

12.1 Configuration

The Command Line executable version of PCANopen Magic is called PCOMAGIC.EXE. It is recommended that the path to the folder where PCANopen Magic was installed is added to the PATH system variable, so that PCOMAGIC.EXE is available from any folder. On Windows 95, 98 and ME this will be done automatically during installation. However on Windows 2000 and XP it must be manually added by completing the following steps:

- Choose Start, Settings and Control Panel to open the Control Panel.
- Double-click on System.
- Click on the Advanced tab.
- Click on Environment Variables…
- Select Path in the top box to change the PATH variable for the current user. Select Path in the lower box to change the PATH variable for all users (requires administrator logon).
- Click on Edit…
- In the Variable Value box add the path to the folder where PCANopen Magic was installed. Paths are separated with semicolons. Do not include the final '\'.
- Click on OK
- Click on OK

The change will take effect only after the machine has been restarted when using Windows 95, 98 or ME. The change will take effect only for new Command windows opened on Windows 2000 or XP.

12.2 Syntax

The command line has the following syntax:

PCOMAGIC [*directives*]

Where:

directives space separated list of directives or
@*commandfile*
commandfile An ASCII file containing a space separated or newline separated list of directives

There are two types of directives:

Configuration	configure the hardware and CAN/CANopen functionality
Operation	an operation to be performed on the CANopen network

Directives may appear in any order in the list, however the operation directives are processed in the order listed. For example the following command line:

```
PCOMAGIC NMT(0x14, START) SDODOWNLOAD(23, 0x5FFF, 0x01, ..\foobar.bin)
```

Transmits a network management message before performing an SDO Download. However the following command line:

```
PCOMAGIC SDODOWNLOAD(23, 0x5FFF, 0x01, ..\foobar.bin) NMT(0x14, START)
```

Performs the SDO Download and then transmits a network management message.

Operation directives may appear more than once on a command line, allowing complex operations to be performed. For example:

```
PCOMAGIC NMT(0x14, PREOPERATIONAL) SDODOWNLOAD(0x14, 0x5FFF, 0x01,
..\foobar.bin)
NMT(0x14, START) SDOUPLOAD(0x18, 0x2000, 0x00, c:\config.txt)
```

Places node 0x14 in Preoperational mode, downloads a binary file to Object Dictionary entry 0x5FFF subindex 0x01, starts the node, then uploads ASCII data from Object Dictionary entry 0x2000 subindex 0x00 of node 0x18.

Directive names are case insensitive.
Whitespace is ignored except when it is between directives, where it is required.
All directives are optional. When a directive is omitted the default setting for that directive is used.

Paths may contain spaces. Paths may also contain newlines, carriage returns and tabs, however these characters are converted to spaces before the path is used, allowing word wrapping at the spaces in paths.

All values passed to a directive may be in decimal or Hexadecimal except where noted. Hexadecimal values must be prefixed with "0x or suffixed with "H" or "h".

12.3 Directives

NMT

Description: Transmits a network management message

More than one allowed: Yes

Type: Operation

Syntax: NMT (*nodeid*, *command*)

Where:

nodeid	The ID of the Node to receive the message or zero to transmit the message to all nodes.
command	The command to send to the node(s). One of: START STOP PREOPERATIONAL RESET RESETCOMM

Output: NMT message sent to *nodeid* [*directive*]
Or: NMT message send failed: *reason* (*nodeid*) [*directive*]

nodeid	The ID of the node receiving the message in hexadecimal with no prefix or suffix
reason	The reason for the failure
directive	The directive being executed

Examples: NMT(0x14, START)
NMT(2AH, STOP)

SDODOWNLOAD

Description: Writes data to a node. Either a file to write data from or hex data directly may be specified. If the Hex data is four bytes or less it must be entered into the directive in big-endian format for human readability. If the Hex data is more than four bytes then it must be entered in little-endian format.

More than one allowed: Yes

Type: Operation

Syntax: SDODOWNLOAD(*nodeid*, *index*, *subindex*, *path* | *data*)

Where:

nodeid	The ID of the node to write to
index	The Object Dictionary index to write to
subindex	The Object Dictionary entry subindex to write to
path	Path to the file to take the data to write from
data	Hexadecimal data to write

Output: SDO Download complete [*directive*]
Or: SDO Download failed: *reason* [*directive*]

Where:

reason	The reason for the failure
directive	The directive being executed

Examples:
SDODOWNLOAD(0x14, 0x2000, 0x1A, C:\shared\test.hex)
SDODOWNLOAD(23, 0x5FFF, 0x01, ..\foobar.bin)
SDODOWNLOAD(23, 0x5FFF, 0x01, 0x11223344)
SDODOWNLOAD(0x14, 0x2000, 0x1A, 33H)
SDODOWNLOAS(0x07, 0x20AA, 0x00, 0xA24D2A5A2B5C3D8A)

SDOUPLOAD

Description: Reads data from a node. If four bytes or less are read then the data is stored in big-endian format for human readability. If more than four bytes are read then the data is stored in little-endian format.

More than one allowed: Yes

Type: Operation

Syntax: SDOUPLOAD(*nodeid*, *index*, *subindex*, *path*)

Where:

nodeid	The ID of the node to read from
index	The Object Dictionary index to read from
subindex	The Object Dictionary entry subindex to read from
path	Path to the file to store the read data

If the file name in the path ends with a ".txt" extension then the read data will be stored in an ASCII file in both hexadecimal and ASCII formats, with up to 16 bytes per line. Each line will start with a hexadecimal offset into the block of data read for the first byte represented on that line.
If the file name in the path ends with any other extension then the read data will be stored in a binary file.
If the path is "screen" then the read data will be output to the screen in both hexadecimal and ASCII formats, with up to 16 bytes per line. Each line will start with a hexadecimal offset into the block of data for the first byte represented on that line.

Output: SDO Upload complete [*directive*]
Or: SDO Upload failed: *reason* [*directive*]

Where:

reason	The reason for the failure
directive	The directive being executed

Examples:
```
SDOUPLOAD(0x14, 0x2000, 0x1A, C:\shared\test.hex)
SDOUPLOAD(23, 0x5FFF, 0x01, ..\foobar.bin);
```

HARDWARE

Description: Selects and configures the CAN hardware interface

More than one allowed: No

Type: Configuration

Syntax: HARDWARE(*type*, *options*)

Where:

type Hardware interface type. One of:

PEAKPARALLEL
PEAKUSB
PEAKPCI
PEAKISA
SYSTECUSB

options Options for hardware configuration. Dependant on the hardware type:

PEAKPARALLEL:

portaddr, *irq*, *porttype*, *baudrate*

portaddr The address of the printer port
irq The IRQ number used for the printer port
porttype One of:

STANDARD
EPP

baudrate The CAN bus baud rate in kbps. One of:

10
20
50
125
250
500
800
1000

PEAKUSB:

baudrate

baudrate The CAN bus baud rate in kbps. One of:

10
20
50
125
250
500
800
1000

PEAKPCI:

baudrate

baudrate The CAN bus baud rate in kbps. One of:

10
20
50
125
250
500
800
1000

PEAKISA:

ioaddr, *irq*, *baudrate*

ioaddr The I/O address for the ISA card
irq The IRQ number used for the ISA card
baudrate The CAN bus baud rate in kbps. One of:

10
20
50
125
250
500
800
1000

SYSTECUSB:

device, *baudrate*

	device	The number of the USB-CANmodul to use or "ANY" for the first one found by windows. device may be in the range 0 – 254.
	baudrate	The CAN bus baud rate in kbps. One of: 10 20 50 125 250 500 800 1000

Output: Hardware initialized
Or: Hardware initialization failed: *reason*

Where:

reason	The reason hardware initialization failed

Default: PEAK Parallel port dongle, with printer port address of 378H, using IRQ 7, standard printer port and 1Mbps baud rate
HARDWARE(PEAKPARALLEL, 0x378, 7, STANDARD, 1000)

Examples: HARDWARE(PEAKPARALLEL, 278H, 6, EPP, 125)
HARDWARE(PEAKUSB, 800)
HARDWARE(SYSTECUSB, ANY, 250)

QUIET

Description:	Outputs information about the processing of each directive to an ASCII output file rather than the standard output, where a program can process it. When this directive is used all output except directive syntax errors are redirected to the specified output file.
More than one allowed:	No
Type:	Configuration
Syntax:	QUIET(*path*) Where: *path* Path of the output file to generate.
Output:	None. Or: Output file creation failed (*path*) Where: *path* The path of the output file to generate as passed to the QUIET directive.
Default:	Output is sent to the standard output.
Examples:	QUIET(test.txt) QUIET(..\test.txt) QUIET(C:\work\test.txt)

ODWRITE

Description: Writes data to a node. The same as the SDODOWNLOAD directive. Either a file to write data from or hex data directly may be specified. If the Hex data is four bytes or less it must be entered into the directive in big-endian format for human readability. If the Hex data is more than four bytes then it must be entered in little-endian format.

More than one allowed: Yes

Type: Operation

Syntax: ODWRITE(*nodeid*, *index*, *subindex*, *path* | *data*)

Where:

nodeid	The ID of the node to write to
index	The Object Dictionary index to write to
subindex	The Object Dictionary entry subindex to write to
path	Path to the file to take the data to write from
data	Hex data to write

Output: OD Write complete [*directive*]
Or: OD Write failed: *reason* [*directive*]

Where:

reason	The reason for the failure
directive	The directive being executed

Examples:
ODWRITE (0x14, 0x2000, 0x1A, C:\shared\test.hex)
ODWRITE (23, 0x5FFF, 0x01, ..\foobar.bin)
ODWRITE (23, 0x5FFF, 0x01, 0x11223344)
ODWRITE (0x14, 0x2000, 0x1A, 0x33)
ODWRITE (0x07, 0x20AA, 0x00, 0xA24D2A5A2B5C3D8A)

ODREAD

Description: Reads data from a node. The same as the SDOUPLOAD directive.

More than one allowed: Yes

Type: Operation

Syntax: ODREAD(*nodeid*, *index*, *subindex*, *path*)

Where:

nodeid	The ID of the node to read from
index	The Object Dictionary index to read from
subindex	The Object Dictionary entry subindex to read from
path	Path to the file to store the read data

If the file name in the path ends with a ".txt" extension then the read data will be stored in an ASCII file in both hexadecimal and ASCII formats, with up to 16 bytes per line. Each line will start with a hexadecimal offset into the block of data read for the first byte represented on that line.
If the file name in the path ends with any other extension then the read data will be stored in a binary file.
If the path is "screen" then the read data will be output to the screen in both hexadecimal and ASCII formats, with up to 16 bytes per line. Each line will start with a hexadecimal offset into the block of data for the first byte represented on that line.

Output: OD Read complete [*directive*]
Or: OD Read failed: *reason* [*directive*]

Where:

reason	The reason for the failure
directive	The directive being executed

Examples: ODREAD (0x14, 0x2000, 0x1A, C:\shared\test.hex)
ODREAD (23, 0x5FFF, 0x01, ..\foobar.bin);