

NEMESYS

January 2008

Documentation
Windows 32-Bit DLL





cetoni GmbH

Am Wiesenring 6

D- 07554 Korbußen

Tel.: +49 (0) 36602 338-0

Fax: +49 (0) 36602 338-11

E-Mail: info@cetoni.de

Internet: www.cetoni.de

1 Table of contents

1 Table of contents	3
2 Document History	4
3 Introduction	5
4 Including Library Functions.....	6
4.1 General Information	6
4.2 Including	6
4.3 Parameter Storage	7
5 DLL Integration into C++ Builder 2007	8
6 DLL Integration into Delphi 2007.....	9
7 DLL Integration into LabVIEW	10
7.1 Introduction	10
7.2 Integration into LabVIEW.....	10
7.3 Groups of functions	13
7.4 LabVIEW Example.....	13
8 Software Version History	14

2 Document History

Date	Version	Edition	Description
01.08.2006	Ver 1.20	August 2006	<ul style="list-style-type: none">• First Library Version
25.09.2006	Ver 1.25	September 2006	<ul style="list-style-type: none">• DLL integration into National Instruments LabVIEW• A serious memory management bug fixed – borIndmm.dll is now required for proper function of nemesys_dll.dll
10.01.2007	Ver 1.25	Januar 2007	<ul style="list-style-type: none">• DLL integration into Borland Delphi
05.06.2007	Ver 1.30	June 2007	<ul style="list-style-type: none">• Several bug fixes• Only one single DLL required now
15.10.2008	Ver 2.02	Oktober 2008	<ul style="list-style-type: none">• Removed API documentation because the API is documented in the file neMESYS_Windows_32-Bit_DLL_Help.chm
06.01.2010	Ver 2.06	January 2010	<ul style="list-style-type: none">• Updated version history

3 Introduction

This documentation "Windows 32-Bit DLL" provides the instructions for the implemented functions. The library is arranged in groups of functions and helps to simplify the programming of the control software based on Windows. This document describes the interface between a program and the Windows DLL (Dynamic Link Library).

BECAUSE THIS LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THIS LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THIS LIBRARY IS WITH YOU. SHOULD THIS LIBRARY PROVE DEFECTIVE OR INSUFFICIENT, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

The latest edition of these "Windows 32-Bit DLL", additional documentation and software to the neMESYS dosing system may also be found in the internet under <http://www.cetoni.de/service/downloads>.

4 Including Library Functions

4.1 General Information

The library 'nemesys_dll.dll' is an implementation of the neMESYS command set for the communication between neMESYS and a personal computer. Using this library is a simple way to develop your own application.

This library is running on each windows 32-bit operating system. You can include it into different programming environments. All the neMESYS commands are implemented and they can be called directly from your own program. You don't have to care about the protocol details. The only thing you have to ensure is properly connected and configured dosing platform.

The library 'nemesys_dll.dll' offers the whole set of neMESYS commands. Executing neMESYS commands, managing configuration parameters, executing dosing tasks and handling low level communication with the neMESYS control units is the business of this library.

4.2 Including

The following chapter describes how to include all library functions in your own windows program. The way how you do that, depends on the compiler and on the programming language you use. Thus we are going to explain the procedure to include the library based on some examples of the most popular programming languages (Chapter 5). For several high-level programming languages are further documentations and its own examples available (Chapter 5).

In order to have a correctly working communication, you have to include the libraries

nemesys_dll.dll

to your programming environment. You have to copy these files to the working directory of your system.

To open the library 'nemesys_dl.dll' you have to use the function [NCS_OpenDevice\(\)](#). You have to do this before you can execute any neMESYS command. At the end of your

program you have to call [NCS_CloseDevice\(\)](#).

Use the calling convention `__stdcall` for this library. This convention is managing how the parameters are put on the stack and who is responsible to clean the stack after the function execution.

4.3 Parameter Storage

The neMESYS library need to store several device parameters of connected neMESYS dosing units persistent. The library will store these parameters into a `devices.ini` file. The library will create this ini file inside the directory "*C:\Documents and Settings\All Users\Application Data\nemesys*". The neMESYS DLL shares this configuration file with the neMESYS UserInterface application.

5 DLL Integration into C++ Builder 2007

You need the following files to include the library to the programming environment of 'Codegear C++ Builder'.

nemesys_api.h	Constant definitions and declarations of the library functions
nemesys_dll.dll	Dynamic link library
nemesys_dll.lib	Import library (OMF Format)

To include the listed files you have to do the following steps:

First Step The files have to be copied to a working directory of the project.

Second Step Write the instruction **#include "nemesys_api.h"** to your program to include the constant definitions and the declarations of the library functions.

Third Step The file 'nemesys_dll.lib' has to be added to the project. For that purpose you have to open the menu point 'Add to project...' of the menu 'Project'. Add the file 'nemesys_dll.lib'. The figure below shows the German version of this dialog.

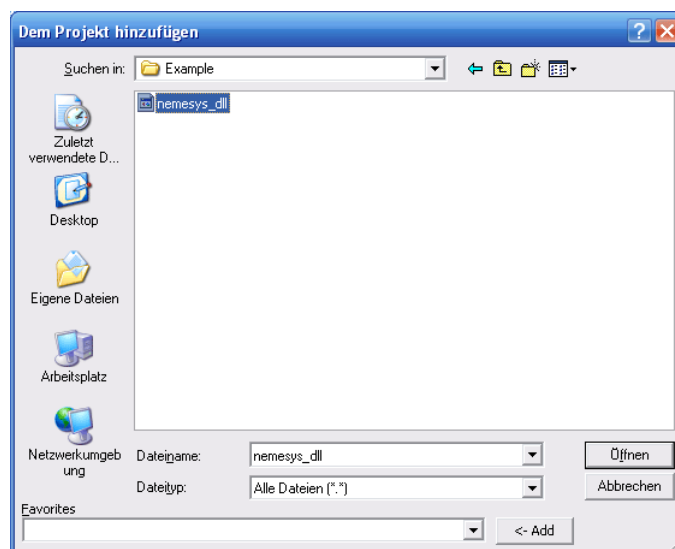


Figure 1 - Adding Library in Borland C++ Builder

After these three steps you can execute directly all library functions in your own code.

6 DLL Integration into Delphi 2007

The following files are necessary to include the library functions to the programming environment of 'Borland Delphi'.

nemesys_api.pas Constant definitions and declarations of the library functions
nemesys_dll.dll Dynamic link library

To include the listed files you have to do the following steps:

First Step You have to copy all files to the working directory of the project.

Second Step Write the instruction '**uses nemesys_api**' to your program to include the constant definitions and the declarations of the library functions.

After these two steps you can execute all the neMESYS commands directly in your own program.

7 DLL Integration into LabVIEW

7.1 Introduction

This documentation “DLL Integration into LabVIEW” explains how the ‘Windows 32-Bit DLL’ is integrated into a program. This document should simplify the programming of the control software based on Windows.

It is a short overview, which important functions are used and like a program must be developed. A complete overview of all existing functions is available in the document “Windows 32-Bit DLL”.

cetoni GmbH certifies that the content of this library is correct according his best knowledge.

BECAUSE THIS LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THIS LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THIS LIBRARY IS WITH YOU. SHOULD THIS LIBRARY PROVE DEFECTIVE OR INSUFFICIENT, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

7.2 Integration into LabVIEW

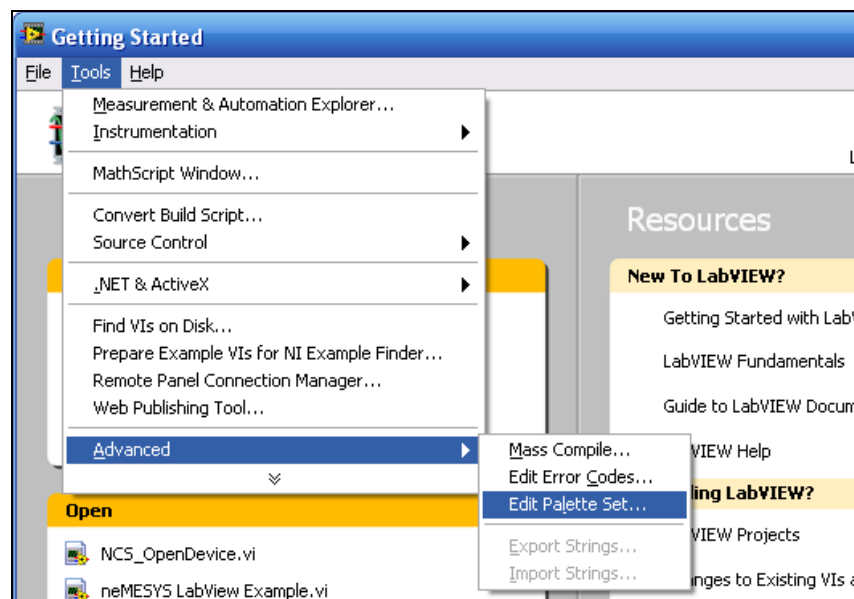
For an easy start with LabVIEW programming, all of the ‘Windows 32-Bit DLL’ function blocks are already configured for you (with LabVIEW 8.0). For each neMESYS command we have a VI block. In order to use the different VIs you have to perform the following steps:

First Step Copy the directory ‘errors’ from your installation directory into the directory ‘...*National Instruments*\LabVIEW X.x\user.lib’

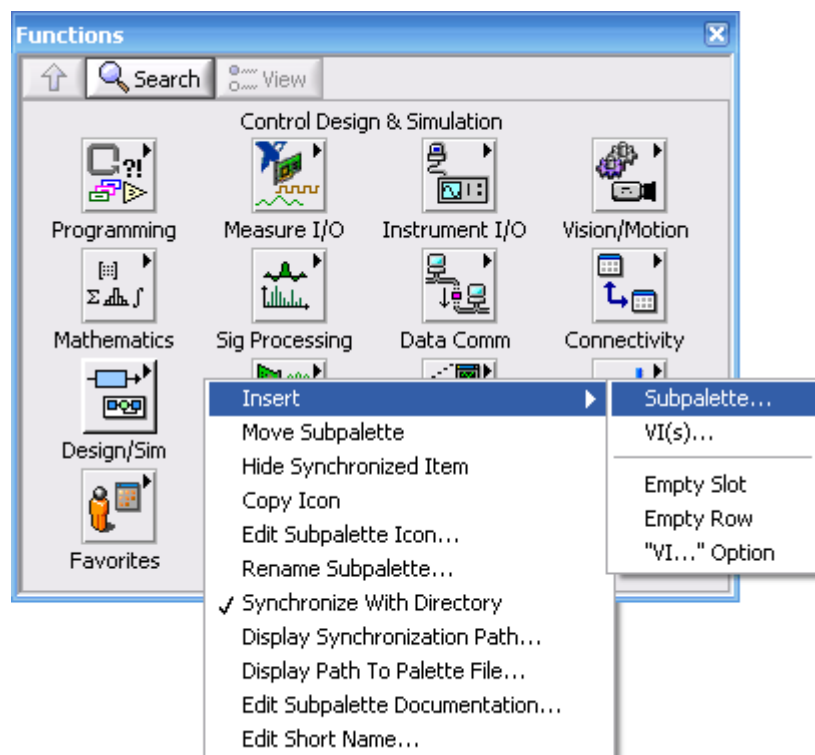
Second Step Start LabVIEW

Third Step The neMESYS VI’s have to be added to the LabVIEW Palette Set. Select

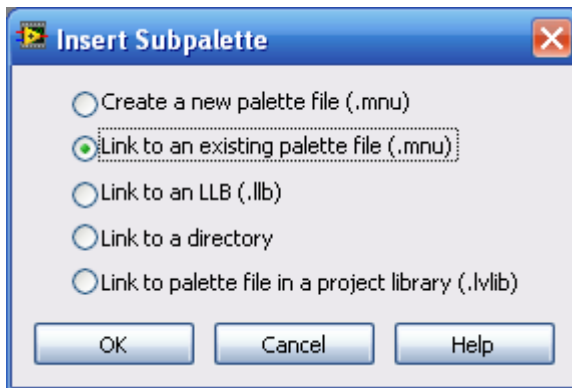
Advanced -> Edit Palette Set from the Tools menu



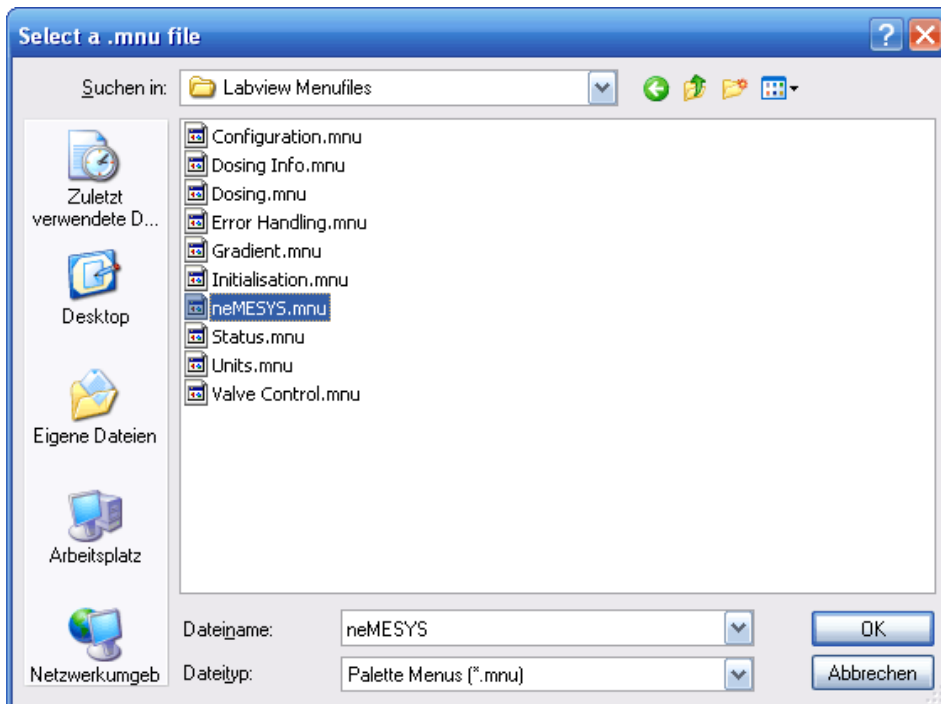
Then click with the right mouse button into the palette and select *Insert -> Sub palette* from the popup menu



Select *Link to an existing palette file (.mnu)* from the *Insert Sub palette* window and then click *OK*.



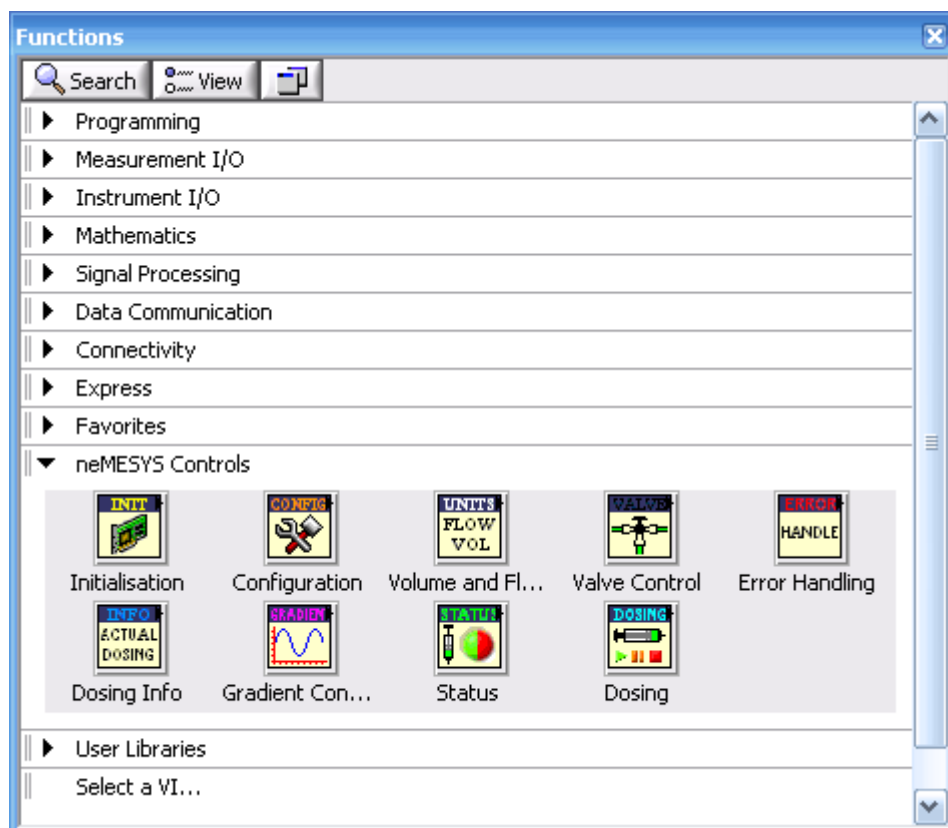
Select the file *neMESYS.mnu* from the *...neMESYS Library\Labview Menufiles* directory within your installation directory.



Finally save the changes you just made to the palette. Now the neMESYS VI's are accessible through the functions palette.

7.3 Groups of functions

The VI's of *neMESYS LabVIEW Integration* are grouped according to the [function groups](#) of '*neMESYS Windows 32-Bit DLL*'.



For detailed information about the function groups and the single VI's please refer to the [functions](#) description of the '*neMESYS Windows 32-Bit DLL*'.

7.4 LabVIEW Example

The example '*neMESYS Labview Example.vi*' in LabVIEW is a dialog based application. The application shows you, how the communication with the neMESYS devices has to be initialised and how several dosing tasks can be realized.

First opens a dialog and scans for connected neMESYS devices. Then the VI reads several configuration parameters from first connected device and sets up the user interface accordingly. Then the VI executes a calibration move and a complete refill and empty sequence for the first connected device. If this procedure is finished, a number of buttons are displayed for manual control of first dosing unit.

8 Software Version History

Changes from Version 2.05 to version 2.06

- Added functions for reading digital and analog input and for writing digital output
- Fixed bug of inverted valve switching
- Added function to disable voltage of a device `NCS_SetDisabled()`

Changes from Version 2.04 to version 2.05

- Fixed documentation of dosing functions
- Support for mid pressure modules improved
- New internal module type configuration

Changes from Version 2.03 to version 2.04

- Fixed `NCS_GetSyringeLevelMax()` and `NCS_GetSyringeLevelMin()` to return
- proper values for all dosing units

Changes from Version 2.027 to version 2.03

- Fixed `NCS_CloseDevice()` exception
- Fixed return value of `NCS_SwitchValve()`
- Adjusted `NCS_IsValveSwitchedToOutput()` to return the right state
- according to configuration of inverted valve switching
- `NCS_DoseVolume()`, `NCS_GenerateFlow()` and `NCS_SetSyringeLevel()` now
- return `-ERR_PERM` if the target position is outside of the device
- and syringe limits
- Fixed `NCS_DoseVolume()` and `NCS_SetSyringeLevel()` not to ignore
- software limits
- Adjusted SI volume and flow units to neMESYS GUI
- Added functions `NCS_SetSyringeLimits()` and
- `NCS_GetSyringeLimits()` to read and write these limits
- Fixed `NCS_GetSyringeLevelMax()` and `NCS_GetSyringeLevelMaxEx()`
- to return proper values
- Added function `NCS_GetSyringeLevelMinEx()` and `NCS_GetSyringeLevelMin()`
- to query minimum syringe level if syringe limits are set
- Documentation updated with new functions

Changes from Version 2.02 to version 2.027

- Fixed return value of `NCS_IsOperational()`

-
- Some error codes renamed for consistency

Changes from Version 1.34 to version 2.02

- Support for high pressure syringe pumps
- Support for inverted valve switching
- Syringe configuration changed to from volume and length to inner diameter and piston stroke
- support for new internal device configuration
- some small changes in API
- support for reading pressure sensor - NCS_GetPressureIs()

Changes from Version 1.33 to version 1.34

- Support for current sensing - NCS_GetCurrentIs()

Changes from Version 1.30 to version 1.33

- Fixed function NCS_SetValveAutomatic() - it is now possible to disable valve automatic
- DLL now shared device configuration file devices.ini with neMESYS UserInterface
- devices.ini is now stored into application data directory in all users profile
- Gradient functions removed
- Flow profile functions added
- support for enhanced flow profiles (duration and valve action)

Changes from Version 1.29 to version 1.30

- Fixed function StopAllUnits() - Function failed in gradient mode
- Fixed function StartGradient() - Time grid value of 0 caused division by zero exception
- Improved device scan and device detection
- Only one single DLL required now

Changes from Version 1.28 to version 1.29

- Fixed a typo in nemesys_api.h
- Fixed a serious bug in message receive procedure (This bug was introduced by version 1.28)

Changes from Version 1.25 to version 1.28

- Fixed some small bugs in internal neMESYS control
- Additional dosing functions with configurable flow rate unit and volume unit
- Enumerations for flow rate units and volume units

Changes from Version 1.20 to version 1.25

- Serious memory management bug fixed
- Minor DLL tweaks for LabVIEW integration

Release Version 1.20

- Initial release of library