NEURO CHECK
Industrial Vision Systems

Application Software for Industrial Vision Systems

Code Reader XNC

Plug-In

Version 6.2.4
Neckarstraße 76-1, 71686 Remseck, Germany

Phone: +49 (0) 7146 - 89 56-0
Fax: +49 (0) 7146 - 89 56-29
E-Mail: info@neurocheck.com
Web: www.neurocheck.com

Table of Contents

NeuroCheck Code Reader XNC Plug-In Help - PI_CodeReaderXNC.NET.dll 3
General Information 3
Introduction 3
Installation 6
Licensing 7
Check Functions 8
Read Code 8
Introduction 8
How to Use 9
Parameter Dialog 13
Auto Read Code Settings 14
Target Value Dialog 15
Read Data Matrix 16
Introduction 16
Target Value Dialog 17
Parameter Dialog 18
Generate Code 20
Introduction 20
Parameter Dialog 21
Support Contact 23
About Dialog 23
Support Services 24
Library Licence 25
Apache License 2.0 (Apache) 25

Introduction

About NeuroCheck plug-in DLLs

A plug-in DLL is a .NET assembly that serves to enhance NeuroCheck with user-defined image processing functionality. The NeuroCheck Plug-In Interface offers the opportunity to integrate user-defined check functions for image processing and data handling. A Plug-In can contain an arbitrary number of self-developed check functions.

These check functions have full access to the NeuroCheck runtime data objects such as Images, ROI Lists or Measurement Lists. The Plug-In check function can be added to a check as well as the built-in standard check functions of NeuroCheck.

Please note that for integration of a plug-in check function into your check routine, a Premium license is required. The completed check routine then can be run with any NeuroCheck license (except the Demo Version).

About this Plug-In

The Plug-In `PI_CodeReaderXNC.NET` integrates check functions designed for decoding barcodes. Supported formats include Code 39, Code 128, Data Matrix, Codabar and others.

Preamble

The *Read Code* check function is capable of decoding various barcode formats (1D and 2D), comparing the code string to a target value and storing it into a data register for further use within NeuroCheck. Its core features are:

- Simple parameterization
- Auto rotation of images containing barcodes
- Support for Various code formats
- Target value comparison

See the table below for supported barcode types by *Read Code* (in alphabetical order):

Code type	Format	Sample
Aztec	2D	
Codabar	1D	
Code 39, Code 93	1D	
Code 128	1D	
Data Matrix	2D	
EAN 8, EAN 13	1D	
Code 2/5 Interleaved	1D	
PDF 417	2D	
QR Code	2D	
RSS 14, RSS Expanded	1D	
UPC A, UPC E	1D	

Licensing of PI_CodeReaderXNC

In addition to the standard NeuroCheck license, also a license for the NeuroCheck Plug-In PI_CodeReaderXNC is required. The protection of the Plug-In is stored as special flag in the same security key as for the NeuroCheck license. If the Plug-In cannot detect the special flag, the check functions will not be executed in NeuroChecks automatic mode. For the purpose of evaluation it is possible to execute the check functions in manual mode. In this case a message informs about the missing license every ten executions.

In order to get the license for the PI_CodeReaderXNC, please contact your local NeuroCheck partner. The plug-in license can be added to a standard NeuroCheck license by remote-programming of the security key. The remote-programming works in the same way as a NeuroCheck update.

Installation

Installation

Copy the following files from the zip archive to the plug-in directory within the desired NeuroCheck project (e.g. '`C:\Users\Public\Documents\NeuroCheck\6.2\Default\Software Extensions\PlugIns\`').

- All files inside the `Binaries` directory
- All *.chm files inside the `Documentation` directory

Loading a Plug-In

In order to use a Plug-In the Plug-In assembly must be loaded in NeuroCheck. The management of Plug-Ins takes place within the Software Settings dialog. The Software Settings dialog can be found in the System menu of NeuroCheck.

Please note that it is impossible to load or unload a Plug-In as long as a check routine is opened, that contains the Plug-In check functions. If the currently opened check routine contains Plug-In check functions, close the check routine first.

Within the Software Settings dialog please select the node Plug-Ins and the sub-node Plug-In in the tree to the left. The loaded Plug-In assemblies are shown in the List of Plug-Ins. Press the Add button to open a file selection dialog in order to select a further Plug-In assembly.

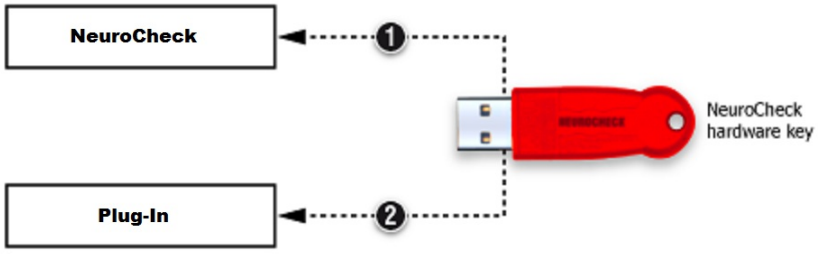
Inserting a Plug-In check function to a check routine

A Plug-In check function is inserted using the Check Function Select dialog. All check functions of loaded Plug-Ins are listed in the Plug-In category of the Check Function Select dialog. Within the Plug-In category the check functions are ordered in sub-categories, where each sub-category represents the check functions of a Plug-In.

Besides the category the user will hardly notice any difference between the usage of Plug-In check functions and built-in check functions.

Licensing

This section describes the licensing mechanism for this NeuroCheck Plug-In.



1. Protection of NeuroCheck

NeuroCheck requires a valid license which is provided as hardware security key (dongle). USB and LPT dongles are available. Please note that a Premium license is required in order to integrate a plug-in check function into your check routine. If the check routine is completed once (including the plug-in functions) it can be run with any NeuroCheck license (except demo).

You obtain the standard NeuroCheck license when you purchase the software from your local NeuroCheck partner.

2. Protection of Plug-In

In addition to the standard NeuroCheck license, also a license for the NeuroCheck Plug-In is required. The protection of the plug-in is stored as a special flag in the same dongle as for the NeuroCheck license. If the plug-in cannot detect the special flag, the execution of the plug-in check functions in the automatic mode will always return NOK and the check functions in manual mode will periodically return NOK with a license error.

In order to get the license for the plug-in, please contact your local NeuroCheck partner. The license can be added to a standard NeuroCheck license by remote-programming of the dongle. The remote-programming works in the same way as a NeuroCheck update.

Read Code: Introduction

Function

This check function provides a powerful code reader for various different 1D and 2D barcode formats. It furthermore supports auto rotation of the barcode image and output of the code string to register.

See "How to Use" for usage information.

Input data

This check function requires an image and a list of ROIs as input objects.

Output data

None.

Visualization

Result image, result code as table, input image and rotated image.

Properties

 Check function group Plug-In

 The check function has a Parameter dialog.

 The check function has a Target Value dialog.

Read Code: How to Use

This check function provides the functionality to decode a barcode in an image.

The following steps are a tutorial on how to use the check function.

Step 1: Input image

The check function supports grayscale and color images (8bit). 16bit images are currently **NOT** supported.

Step 2: Define ROI(s) containing the barcode

Define rectangular ROI(s) enclosing the area where the barcode is. The check function internally always uses a ROI's surrounding rectangle, even if you have defined a different type (e.g. circular or polyline ROIs).

Note: Many code types (e.g. Code 128) require a so called quiet zone, i.e. some empty space before and after the actual code. See below for example ROIs.

Not OK: Almost no quiet zone around the code. Decoding might fail.

OK: Some empty space is included before and after the barcode.

Step 3: Parameterize Read Code

Read Code only requires a very basic set of parameters. The available options are described in the following.

a) Code type

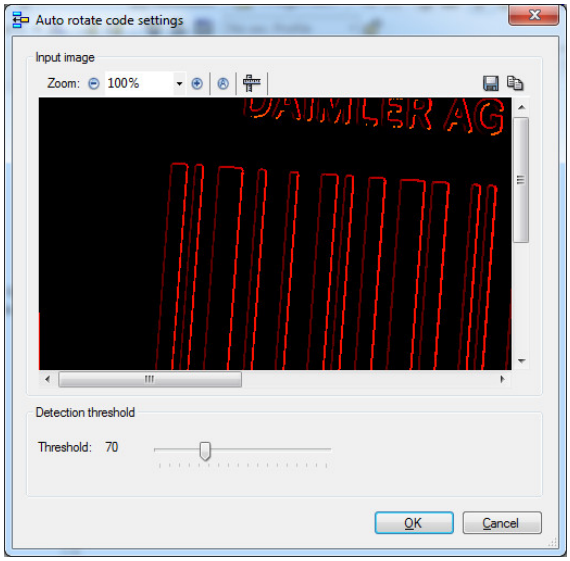
Define the code type expected here. For debugging purposes, you may also select the option "ALL CODE TYPES". In this option, *Read Code* will try to decode the image provided with every code type supported.

Note: The setting "ALL CODE TYPES" is only meant for testing purposes, as the execution time for *Read Code* will increase significantly with this option selected. As soon as you know which code type will be present in the input image, you should adjust this setting accordingly.

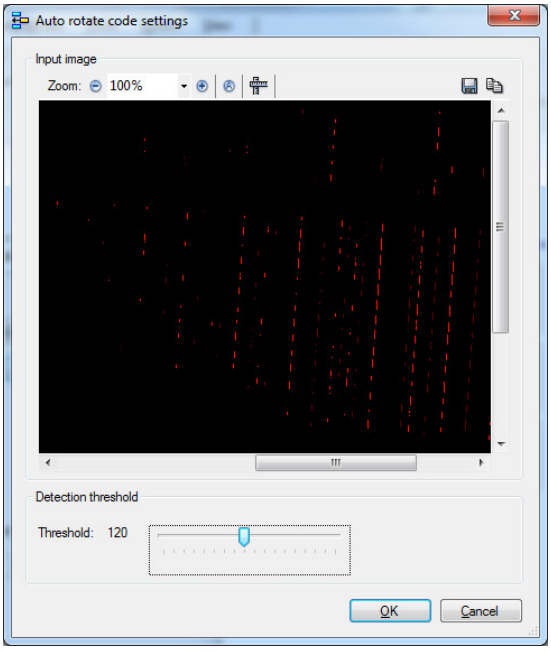
b) Auto rotate code

Read Code is only capable of reading barcodes if they are provided horizontally or vertically. With this option selected, the check function tries to determine the code's rotation and compensate the angular offset. The compensation is carried out by extracting the code's edges and examining its prominent angles. In the Settings dialog (see below), you may adjust the threshold value of the edge extraction to ensure stable results.

Note: Auto rotation can be time consuming when working with large images. In many cases, manual rotation compensation carried out by NeuroCheck® standard check functions may lead to decreased execution times.



Good threshold value for auto rotation.



Threshold value is too high for auto rotation.

c) ROI handling

Two different options for ROI Handling are available:

- All ROIs first match
  Every input ROI is examined for barcodes. As soon as a barcode is recognized and decoded successfully, the subsequent ROIs are skipped and not examined.
- First ROI only
  Only the very first input ROI (index 0) is examined for barcodes.

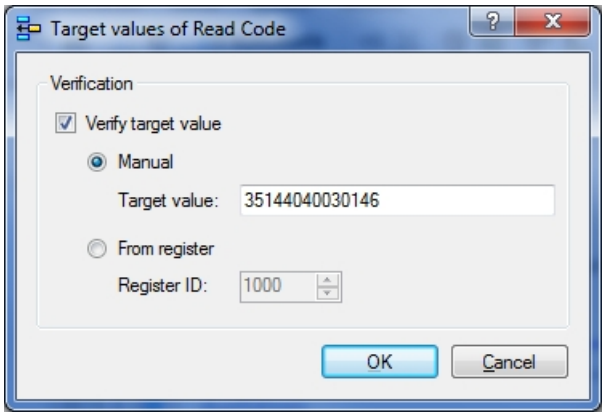
d) Write result to register

With this option selected, *Read Code* will write the code string found to the register with the ID specified.

Note: The target register has to be of type String, even if the barcode format decoded only supports numbers.

Step 4: Target value definition

You may define a target value for the check function's result code string in the Target Value dialog (see below).



Read Code Target Value dialog.

a) Verify target value

This option globally enables (checked) or disables (unchecked) target value verification of the check function.

b) Manual

If you want to enter the target value manually, select this option and enter the target code string into the text box.

c) From register

If you need a dynamic code string for target value verification, select this option and enter the register ID the target code string is stored in.

Note: The register has to be of type **String**, even if the code type only supports numbers.

Step 5: Result visualization

Read Code provides four different result views (see below):

1. **Result image:** Shows the input image and examined ROI(s). If a code was successfully decoded and checked against a target value, it is displayed green. If the target value differs from the decoded string, the ROI will be highlighted yellow. If no code was found within the ROI(s) provided, they will be drawn with red borders.
2. **Result code:** Displays a table view with the code format and code string detected. If a target value is specified, it is appended at the table's last column.
3. **Input image:** Shows the original input image without modifications.
4. **Rotated image:** Displays the image after auto rotation has been applied.
   Note: This view will always be available, even if auto rotation has been disabled. This visualization is calculated on the fly, i.e. only if the user requests it by selecting the corresponding tab. The generation of this view will not affect *Read Code*'s execution time.

Result image view.

Code type	Code value	Target code
CODE 128	35144040030146	35144040030146

Result code view.

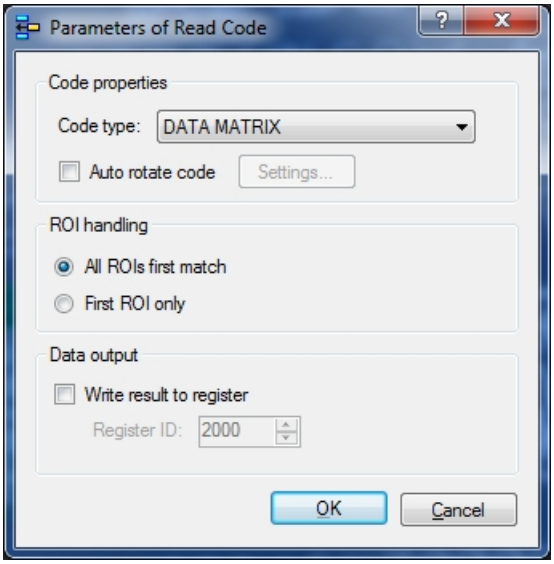
Rotated image view.

Rotated image view.

Read Code: Parameter Dialog

This check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Code type	Code format expected in the input image.
Auto rotate code	Check for auto rotation compensation of the barcode present in the input image.
ROI handling	Specify here, from which ROI the barcode should be read.
All ROIS first match	All input ROIs are iterated until a barcode is detected.
First ROI only	Only the first input ROI (index 0) is examined on possible barcodes.
Write result to register	If checked, the code string detected is transferred to the register with the provided ID.

Read Code: Auto Read Code Settings

In this dialog, you see the input image after an edge detection algorithm was applied to it. It is used to detect the longest edge in the image which is used as a reference to rotate the image.

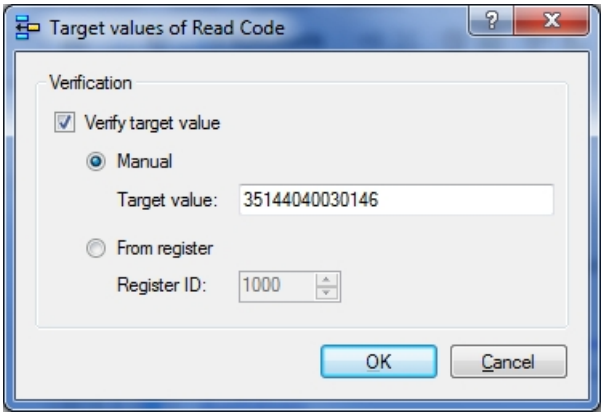
This **Parameter** dialog contains the following elements:

Element	Description
Threshold	Specify here, at which point edges with low contrast in the displayed image are removed.

Read Code: Target Value Dialog

This check function has a **Parameter** dialog.

Screenshot of Target Value Dialog



The **Target Value** dialog contains the following elements:

Element	Description
Verify target value	Global setting for verifying a target value.
Manual	Target value is provided manually by entering the target code string into the text box.
From register	Target value is read from register with the ID provided.

Read Data Matrix: Introduction

Function

This check function reads Data Matrix code inside a single region of interest.

Input data

This check function requires an image and a list of ROIs as input objects.

Output data

None.

Visualization

Result image, result code, detection image, input ROIs and input image.

Properties

 Check function group Plug-In

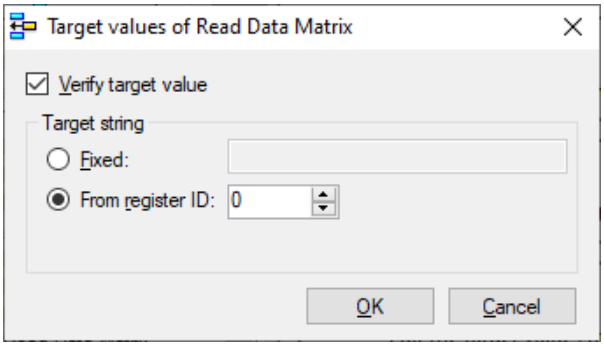
 The check function has a Parameter dialog.

 The check function has a Target Value dialog.

Read Data Matrix: Target Value Dialog

This check function has a **Parameter** dialog.

Screenshot of Target Value Dialog



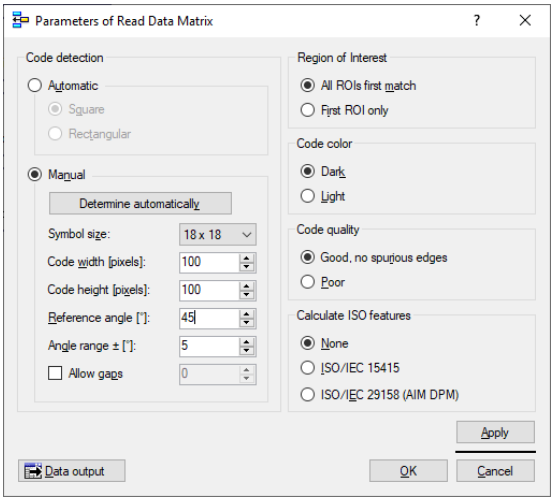
The **Target Value** dialog contains the following elements:

Element	Description
Verify target value	Global setting for verifying a target value.
Fixed	Target value is provided manually by entering the target code string into the text box.
From register ID	Target value is read from register with the ID provided.

Read Data Matrix: Parameter Dialog

This check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Code detection	**Automatic:** The code will be automatically be detected. Can be useful for tasks when size, rotation etc. of the code varies. • **Square:** Code format is expected to be square (e.g. 10x10, 18x18, etc.) • **Rectangular:** Code format is expected to be rectangular (e.g. 8x18, 12x36, etc.) **Manual:** The code will be detected with a priori knowledge from the user. This approach is more robust than the automatic approach. • **Determine automatically:** An attempt is made to automatically determine the values of the manual approach with the help of the automatic approach. • In some cases, if the attempt to automatically fill out the values is successful, however the manual approach can't read the code, try to increase the **Angle range.** • If the attempt failed to automatically fill out the values, it does **NOT** mean that the code can't be read with the manual approach. In this case, you have to try to find a suitable parameterization yourself.

Element	Description
	• **Symbol size:** Number of data points "data cells" in Column and Rows. • **Code width [pixels]**: Width of the code in pixels. • **Code height [pixels]**: Height of the code in pixels. • **Reference Angle**: Code orientation angle. • **Angle range**: Range in which the "Reference Angle" can vary. • **Allow gaps:** Is the Code created with empty space between the "data cells" (this is often the case if the data matrix code is created width dots instead of squares). Here, you can also enter the maximal Value (in vertical and horizontal direction) of the "empty space" in pixels between the "data cells".
Region of Interest	**All ROIS first match:** All input ROIs are iterated until a Data Matrix code is detected. **First ROI only:** Only the first input ROI (index 0) is examined on possible Data Matrix code.
Code color	**Dark:** If this option is selected, the check function searches for a code with a dark frame. **Light:** If this option is selected, the check function searches for a code with a light frame
Code quality	**Good, no spurious edges:** If this option is selected, code detection is accelerated, because the check function can skip processing steps for the distinction of code and noise edges. **Poor:** If this option is selected, damaged codes and codes on problematic surfaces can be found more reliably.
Calculate ISO features	**None:** No evaluation **ISO/IEC 15415:** Standard for the evaluation of printed Data Matrix codes on labels. **ISO/IEC 29158 (AIM DPM):** Standard for the evaluation of printed Data Matrix codes using Direct Part Marking (DPM) technologies.

Generate Code: Introduction

Function

This check function generates a code image based on an input string from register. This code image can be used e.g. in the Process View for transferring information to other systems via handheld barcode scanners.

Input data

This check function requires no input data.

Output data

Image showing the code.

Visualization

Image showing the code.

Properties

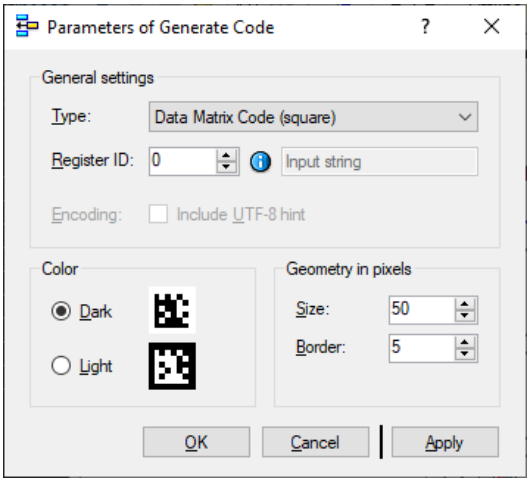
Check function group Plug-In

The check function has a Parameter dialog.

Generate Code: Parameter Dialog

This check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



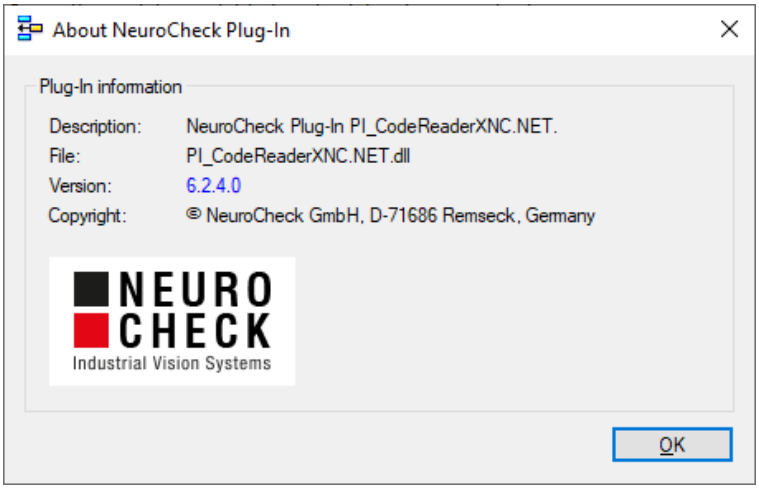
The **Parameter** dialog contains the following elements:

Element	Description
General settings	Properties of the code, like type and content. **Type:** Output code type. Currently, three formats are supported: Data Matrix Code (square), Data Matrix Code (rectangular) and QR Code. **Register ID:** Defines the register ID for the code content. Only String registers are supported. Please make sure that the register used for the code content does exist and has the required data type. Otherwise the check function will fail. **Encoding:** When checked, an encoding hint for UTF-8 ("\000026") is included in the generated code. Some handheld readers need this hint to determine the correct encoding and read special characters. Some readers, however, include the coding hint in the output code, which might cause problems in further processing of the code content. This option is only available for QR codes.
Color	• **Dark:** The code color will be black and the background color will be white. • **Light:** The code color will be white and the background color will be black.
Geometry in pixels	**Size:** Determines the code size (width and height) in pixels. The optimal code size depends on the content length. You have to figure out the size by trying to read the code with your reading device.  If the size is smaller than the minimal code size required by the content, an incomplete/incorrect code could be displayed.  If '**Data Matrix Code (rectangular)**' is selected in **Type**, then the size parameter only determines the height in pixels. The width depends on the height and the content length. **Border:** Size of the border, which is added to all sides to the code. For some algorithms to read the code successfully, it is required to have an empty space (quiet zone) around the code.

About Dialog

This dialog displays version information about the NeuroCheck Plug-In **PI_CodeReaderXNC.NET.dll**.

Screenshot of About Dialog



Support Services

For technical support, please contact your local NeuroCheck partner or NeuroCheck GmbH:

Phone: +49 (0) 7146 - 89 56-40
E-Mail: support@neurocheck.com
Web: www.neurocheck.com

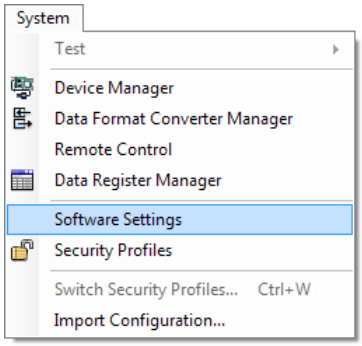
Before contacting us, please provide some important information about your system:

Information about your NeuroCheck installation and your PC setup:

- Use the NeuroCheck Diagnostics tool to check your installation and computer configuration.
- The NeuroCheck Diagnostics is installed in the "Tools" folder within your NeuroCheck installation.

Log file information:

- Logging for NeuroCheck can be activated in **System** > **Software Settings** > **Diagnosis** > **Logging**.



Apache License 2.0 (Apache)

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and

3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.