

Image Tools

Plug-In



Copyright

Copyright © NeuroCheck GmbH
All rights reserved.
Version 6.2.8
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 PI_ImageTools.NET Plug-In Help	3
General Information	3
Introduction	3
Installation	4
Multi-Frame Image Files	5
Check Functions	7
Display Text in Image	7
Introduction	7
Parameter Dialog	8
Load Image from File	10
Introduction	10
Parameter Dialog	11
Load Images from Directory	12
Introduction	12
Parameter Dialog	13
Load Multi-Frame Image	15
Introduction	15
Parameter Dialog	16
Load Multi-Frame Image To Tray	18
Introduction	18
Parameter Dialog	19
Paste Image	20
Introduction	20
Parameter Dialog	21
Save Image to File	22
Introduction	22
Parameter Dialog	23
Save Multi-Frame Image	24
Introduction	24
Parameter Dialog	25
Save Multi-Frame Image From Tray	27
Introduction	27
Parameter Dialog	28
Support Contact	30
About Dialog	30
Support Services	31

Introduction

About NeuroCheck plug-in DLLs in general

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).

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 then 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 one 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.

Multi-Frame Image Files

General

Multi-Frame image files are TIFF-files (*.tif, *.tiff) which can contain multiple images of different dimensions, channel counts and/or bit depths.

Plug-In

This plug-in supports multi-frame images with 1 or 3 channels (monochrome/RGB) as well as 8 or 16 bit color depth.

The check functions 'Load Multi-Frame Image' and 'Load Multi-Frame Image to Tray' execute **synchronous** and therefore might take some time, depending on your frame count and dimensions.

The check functions 'Save Multi-Frame Image' and 'Save Multi-Frame Image from Tray' only start an **asynchronous** process and thus execute quiet fast. But remember: The save operations are executed in the background and your system load is increased until the save operations are completed.

To monitor the current progress of pending save operations you can use the 'Save MFI'- and 'Save MFI from Tray progress viewer'. You can find the progress viewers under Extras -> Save MFI progress viewer Or Extras -> Save MFI from Tray progress viewer respectively.

Configuration

There are some global settings which are accessible in the NeuroCheck plug-in configuration (e.g.

"C:\Users\Public\Documents\NeuroCheck\6.2\Default\Configuration\NC62CFG.PI.CFGX").

Parameter name	Unit	Default Value	Description
SaveMultiFrameImage_MaxNumberOfParallelSaveOperations	piece	5	This value determines how many parallel save operations are handled by the check function 'Save Multi-Frame Image'.
SaveMultiFrameImage_Timeout	ms	10000	This value determines how long the check function 'Save Multi-Frame Image' tries to add the frame to the specified Multi-Frame image file. It should almost never become necessary to increase this value.
SaveMultiFrameImageFromTray_MaxNumberOfParallelSaveOperations	piece	5	This value determines how many parallel save operations are handled by the check function 'Save Multi-Frame Image from Tray'.

Storage process durations for different configurations:

Setup	Compression	JPEG quality [%]	Mean execution time [ms]	File size [MB]
5 small images à 500x400px	NONE		~23	2.42
	LZW* (lossless)		~41	0.76
	JPEG (lossy)	90	~81	0.41
		75	~59	0.25

		50	~59	0.18
360 frames à 1920x800px	NONE		~2,596	540.1
	LZW* (lossless)		~8,115	177.5
	JPEG (lossy)	90	~12,891	43.9
		75	~12,886	29.7
		50	~11,300	10.3

* Lempel-Ziv-Welch-Algorithm

Display Text in Image: Introduction

Function overview

This check function allows writing text into an image. This can for example be used to display obtained information during a check routine. The text can be fix or dynamically loaded from a register cell.

Input data

This check function uses an image as input data object to draw the text on. It uses a fix text or takes it from a register cell of data type string. 16-Bit images have to be converted to 8-Bit beforehand.

Output data


The output of this check function is the modified image with the specified text in it. A new output data object is created.

Result view

The result view shows the modified image with the text in it. If the text exceeds the borders of the image, the rest of the text is truncated. In case of an error, the error message is displayed in the result view.

Properties

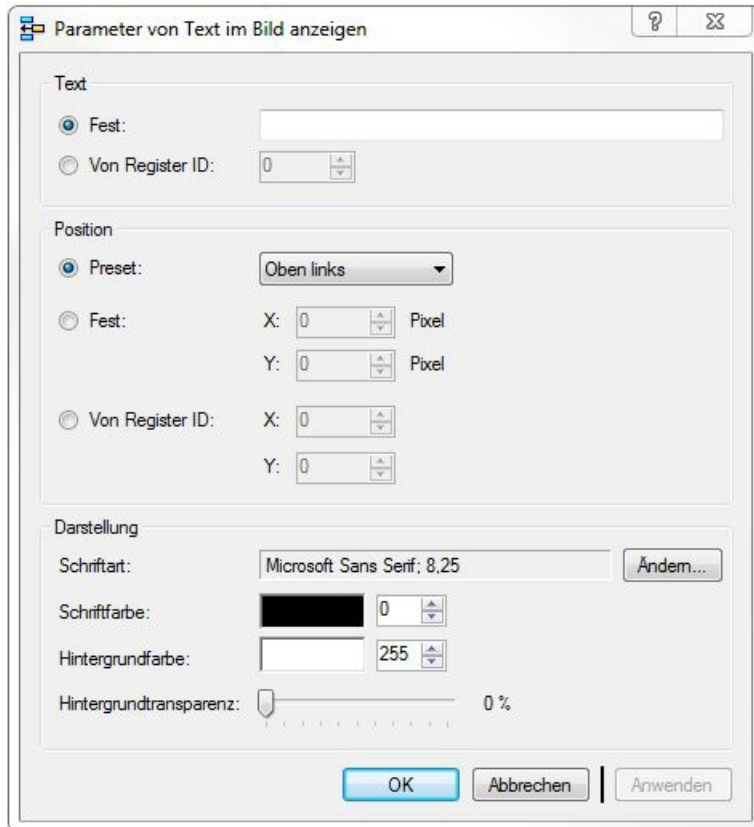
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Display Text in Image: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

☑ [Screenshot of Parameter Dialog](#)



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

Element	Description
Text	Choose the source of the text from the following options: <ul style="list-style-type: none"> • Fix text: Sets a fix text to be displayed. • From register ID: Gets the text from a register cell. The register cell has to have the data type string.
Alignment	The position of the text in the image. The following options are available: <ul style="list-style-type: none"> • Top left • Top center • Top right • Bottom left • Bottom center • Bottom right
Font	Specifies the used font for the text and additional options, for example font size, font style and effects.
Colors	Depending on the pixel format of the image, you are able to specify only the gray value for the text and the background color or its RGB values plus background color. If the image is an RGB image, the two buttons labeled Color... allow selecting the color for the text and background from a color palette.

Element	Description
Background color transparency	Defines the transparency of the background.
Preview	Displays the text with the selected parameters.

Load Image from File: Introduction

Function overview

This check function loads an image from hard disk (or network drive etc.) and transfers it to the NeuroCheck data pool. The file name can be fix or dynamically loaded from a register cell. The image channels have to be 8-bit per channel. Other sizes are not allowed and will result in an error message.

Use case

The use of this check function is that it loads images without triggering a warning message during automatic mode.

Input data

This check function requires no input data object. It uses a fix file name or takes it from a register cell and loads the image.

Output data

The check function returns an image as output data object.


Result view

The result view shows the loaded image.

In case of an error, the error message is displayed in the result view.

Properties

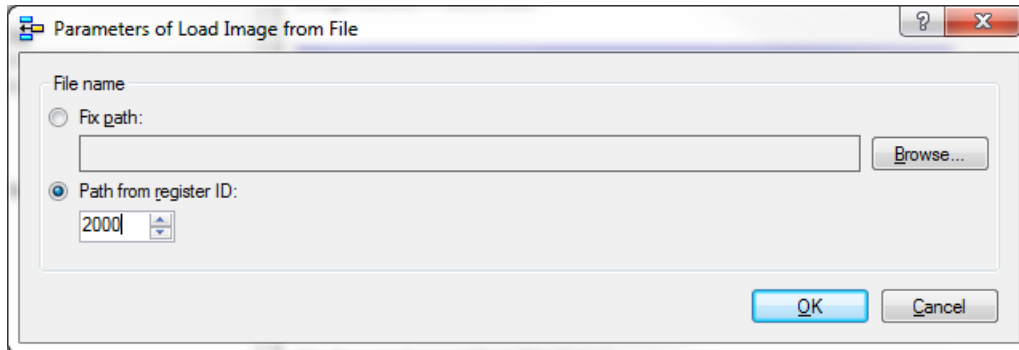
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Load Image from File: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

☑ [Screenshot of Parameter Dialog](#)



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

Element	Description
File name	<p>Choose the source of the file name from the following options:</p> <ul style="list-style-type: none"> • Fix path: Loads the image directly from a *.BMP, *.JPG, *.JPEG, *.PNG, *.TIF, *.TIFF file. • Path from register ID: Gets the file name from a register cell and then loads the image file. The register cell has to have the data type String.

Load Images from Directory: Introduction

Function overview

This check function loads a list of images (one at a time) from the hard disk (or network drive etc.) and transfers it to the NeuroCheck data pool. The path name containing the images can be fixed or dynamically loaded from a register cell. The image channels have to be either 8 or 16-bit per channel. Other sizes are not allowed and will result in an error message.

Use case

Use this check function to load all images from a source directory one by one into a check routine.

Input data

This check function requires no input data object. It uses a fixed directory path or takes it from a register cell and loads the list of images.

Output data

The check function returns the current image in the list as output data object.


Result view

The result view shows the loaded image.

In case of an error, the error message is displayed in the result view.

Properties

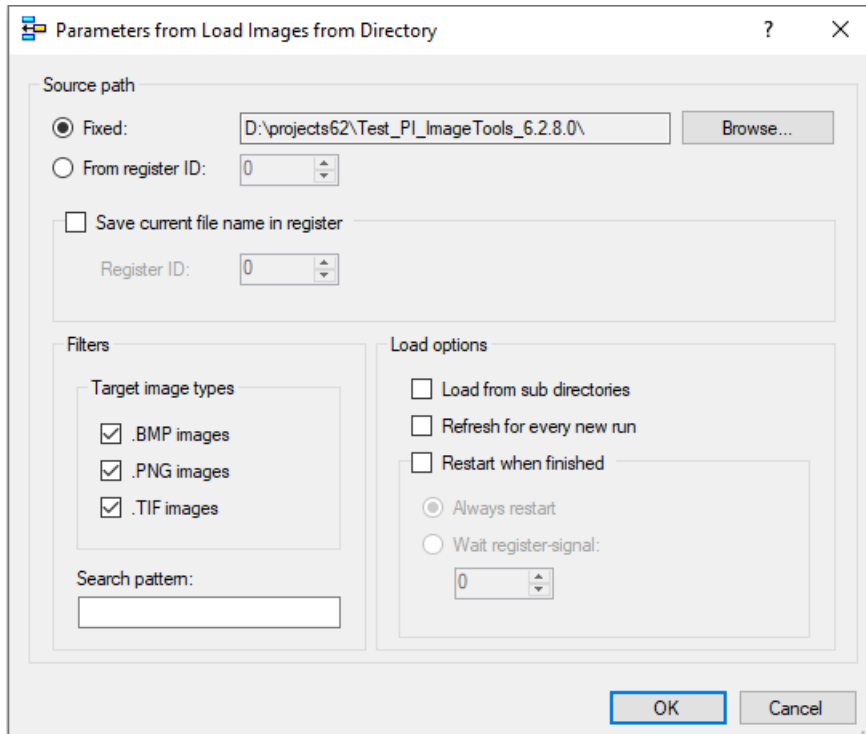
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Load Images from Directory: Parameter Dialog

This plug-in check function has a **Parameter** dialog.


☑ Screenshot of Parameter Dialog



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


Element	Description
Root path	<p>Choose the source of the file name from the following options:</p> <ul style="list-style-type: none"> • Fixed: Loads the image list from a fixed directory. • From register ID: Gets the path name from a register cell and then loads the image list. The register cell has to have the data type String.
Save current path in register	<p>Saves the current image in the list to a specified register cell:</p> <ul style="list-style-type: none"> • Register ID: The ID of the register cell, where the current image path is saved. The register cell has to have the data type String
Filters	<p>Apply a set of filters when listing a directory:</p> <ul style="list-style-type: none"> • Target image types: The image formats to be listed. Supported formats are .BMP, .PNG and .TIF. • Search pattern: The search string to match against the names of files in path. This parameter can contain a combination of valid literal path and wildcard (* and ?) characters, but it doesn't support regular expressions. This applies only to the files listed and not the path name itself. <ul style="list-style-type: none"> ◦ More info at: https://docs.microsoft.com/en-us/dotnet/api/system.io.directory.enumeratefiles?view=net-5.0

Element	Description
Load options	<p>Indicates how the images should be listed:</p> <ul style="list-style-type: none"> • Load from sub directories: Includes sub directories in the listing process. • Refresh for every new run: Reloads the image list every time the check function is executed. • Restart when finished: Restart from the beginning after all images in the directory have been listed: <ul style="list-style-type: none"> ◦ Always restart: Always restart when list-end is reached. ◦ Wait register-signal: Restart only if the specified register cell contains a Boolean value of "True". After restarting, the register cell is set back to "False".



The "Search pattern" gives the opportunity to filter items based on indexes, dates or other partial text patterns. Examples:

1. The search pattern **2021*** will make the check function list only files that start with the prefix **2021**.
2. ***123.bmp** will list all Bitmap files that end with the suffix **123**.
3. ***abc*cde.png** will list all .PNG files with **abc** in the middle and that end with **cde**.



Use the parameter "Refresh for every new run" when new files might be created or copied to the specified directory after the list is initialized. However, it can happen that the new files are placed before the current item in the list, since the check function lists files by name (alphabetical order). In these cases, the new files will be skipped but will be included in the next run if the "Restart when finished" is active.

Load Multi-Frame Image: Introduction

Function overview

This function can be used to open Multi-Frame image file(s) and load specified frames into the NeuroCheck check routine.

Input data

This check function requires no input data object.

Output data

The check function returns an image as output data object.

Result view

The result view shows the loaded image.

In case of an error, the error message is displayed in the result view.

Properties



Check function group Plug-In.

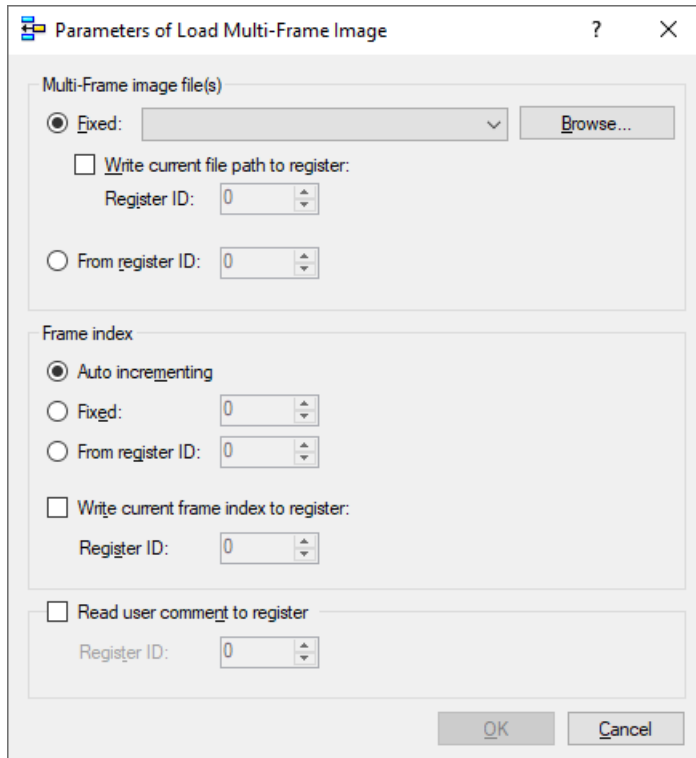


The check function has a [Parameter Dialog](#).

Load Multi-Frame Image: Parameter Dialog

This plug-in check function has a **Parameter** dialog.


☑ [Screenshot of Parameter Dialog](#)



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

Element	Description
Multi-Frame image file(s)	<p>Files which should be loaded.</p> <ul style="list-style-type: none"> • Fixed: Choose one or more Multi-Frame image files. When multiple files are selected, they are used alternately for each execution (except with auto incrementing index). <ul style="list-style-type: none"> ◦ (optional) Write current file path to register: Specify a register where you want to write the file path of the currently loaded Multi-Frame image file to. The register cell has to have the data type String. • From register ID: The check function determines the file path (dynamically) from the Data Register Manager using the specified register ID.

Element	Description
Frame index	<p>Index of the frame that should be loaded.</p> <ul style="list-style-type: none"> • Auto incrementing: The check function loads the first frame (index 0) at the first execution and will iterate with each following execution over all available frames. Changing the parameters or switching between Manual- and Automatic-Mode will reset the index and the automatic incrementation will restart at index 0. When all frames have been loaded, the procedure is performed for all other multi-frame files as well or restarted from scratch if only one Multi-frame file is selected. • Fixed: The check function will always return the image at the specified index. • From register ID: The check function determines the index (dynamically) from the Data Register Manager at runtime using the specified ID. The register cell has to have the data type Integer. • (optional) Write current frame index to register: Specify a register where you want to write the frame index of the currently loaded Multi-Frame image file to. The register cell has to have the data type Integer.
Read user comment to register	<p>Reads the UTF-8 encoded text from the ImageDescription (0x010E) tag of the currently loaded frame and saves it to the provided register cell of type String.</p>



The **automatic iteration** across all frames is reset if:

- the parameter configuration has changed,
- the operating mode was changed (manual/automatic) or
- the file path in the register has changed.

Load Multi-Frame Image To Tray: Introduction

Function overview

This function can be used to transfer all images within the specified Multi-Frame image file(s) into the data tray.

Input data

This check function requires no input data object. It uses a fixed file path or determines it dynamically from the register and loads the images into the data store.

Output data

The check function returns no output data object.


Result view

The result view signals a successful transfer of the frames to the data tray.

In case of an error, the error message is displayed in the result view.

Properties

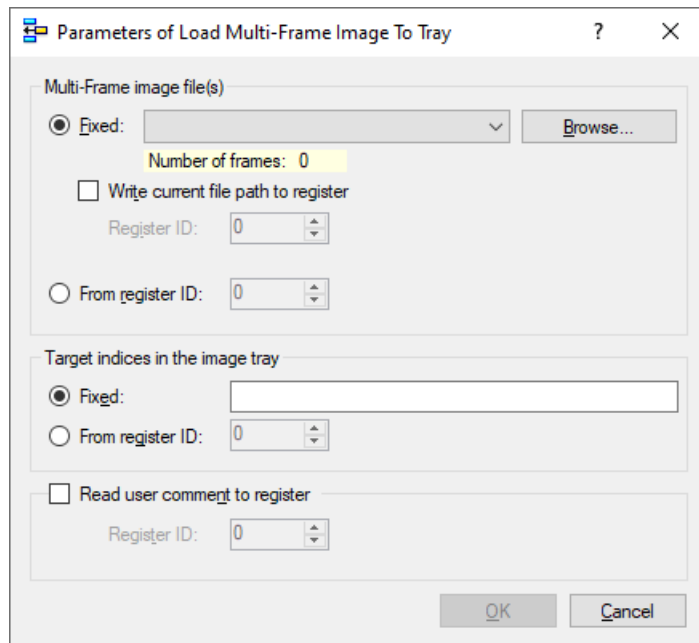
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Load Multi-Frame Image To Tray: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

[Screenshot of Parameter Dialog](#)



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

Element	Description
Multi-Frame image file(s)	<p>Hint: Each file must have the same number of frames.</p> <ul style="list-style-type: none"> • Fixed: Chose one or more Multi-Frame image files. When multiple files are selected, they are used alternately for each execution (except with auto incrementing index). <ul style="list-style-type: none"> ◦ (optional) Write current file path to register: Specify an register where you want to write the file path of the currently loaded Multi-Frame image file. The register cell has to have the data type String. • From register ID: The check function determines the file path (dynamically) from the Data Register Manager using the specified register ID. The register cell must be of type string.
Target indices in the image tray	<p>Specify the indices where you want to insert the imported frames. You can specify single indices by separating them with a comma (","), define index ranges by the following scheme ("[first index] - [last index]") or combine both ways.</p> <p>Example: If the Multi-Frame image file contains of 6 frames you could enter the following: "0,5,10-13".</p> <p>Hint: The frame count inside the Multi-Frame image file(s) must match the specified indices.</p> <ul style="list-style-type: none"> • Fixed: The check function will always use the specified indices. • From register ID: The check function determines the indices (dynamically) from the Data Register Manager using the specified register ID. The register cell must be of type string.
Read user comment to register	<p>Reads the UTF-8 encoded text from the ImageDescription (0x010E) tag of the first frame in the file and saves it to the provided register cell of type String.</p>

Paste Image: Introduction

Function overview

This check function pastes content from an image into another image. Therefore two input images are needed. The check function always replaces the content of the first input image by the content of the second input image. So the first input image will be overwritten. The offset from the parameter is in relation to the first image. Use the offset to replace a specific region of the first input image by the second image. The offset can be fixed or dynamically loaded from two register cells. The first image will always keep its dimensions. Only the region where the second image overlaps the first image will be pasted. The area of the second image which is outside of the dimensions will be ignored.

The input images need to have the same data type and the same number of channels. Different types of input images are not allowed and will result in an error message.

Use case

Use this check function to combine regions of two images very quickly. But always keep in mind that the original data of the first input image is overwritten.

Input data

This check function requires two images as input data objects. The input images need to have the same data type and color type.

Output data

The check function does not return any output data. It overwrites the original content of the first input image. The check function does not create a data object in the NeuroCheck data pool.


Result view

The result view shows the two input images. The original second input image, and the input image with the replaced content from the second input image.

In case of an error, the error message is displayed in the result view.

Properties

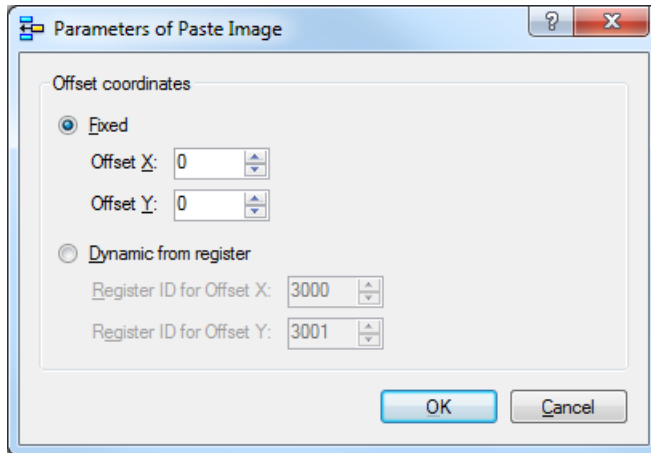
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Paste Image: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

☑ Screenshot of Parameter Dialog



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

Element	Description
Offset coordinates - Fixed	Defines a fix offset in X and Y dimension. The offset is the point where the second image is pasted into the first image.
Offset coordinates - Dynamic from register	Defines the register IDs, for the offset. The offset is the point where the second image is pasted into the first image.

Save Image to File: Introduction

Function overview

This check function saves an image to the hard disk (or network drive etc.) from the NeuroCheck data pool. The file name can be fixed or dynamically loaded from a register cell. The image channels have to be 8-bit per channel. Other sizes are not allowed and will result in an error message.

Input data

This check function requires an image as input data object.

Output data

The check function returns has no output data object.


Result view

The result view shows the saved image.

In case of an error, the error message is displayed in the result view.

Properties

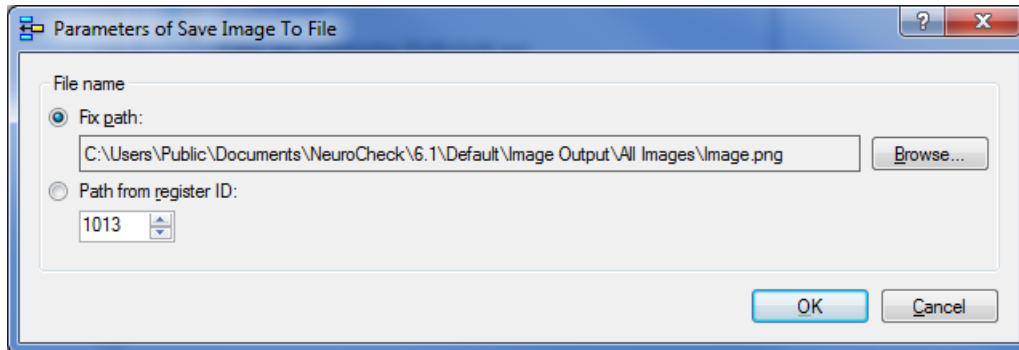
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Save Image to File: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

☑ Screenshot of Parameter Dialog



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

Element	Description
File name	<p>Choose the file name from the following options:</p> <ul style="list-style-type: none"> • Fix path: Saves the image directly as a *.BMP, *.JPG, *.JPEG, *.PNG, *.TIF, *.TIFF file. • Path from register ID: Gets the file name from a register cell and then saves the image file. The register cell has to have the data type String.

Save Multi-Frame Image: Introduction

Function overview

This function can be used to save the input image to a new multi-frame image file.

Input data

This check function requires an image as input data object.

Output data

The check function returns no output data object.


Result view

The result view shows the saved image.

In case of an error, the error message is displayed in the result view.

Properties

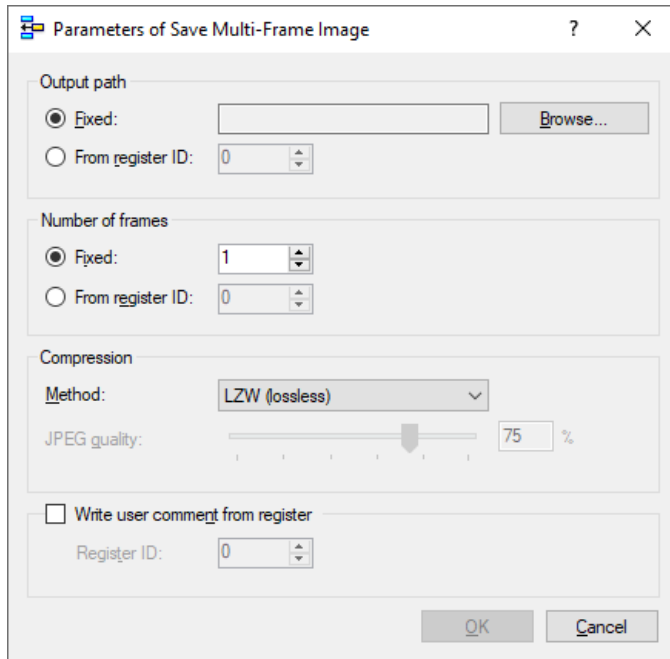
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Save Multi-Frame Image: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

☑ Screenshot of Parameter Dialog



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

Element	Description
Output path	<p>Choose the file path from the following options:</p> <ul style="list-style-type: none"> • Fixed: Saves the image with a static output path which won't change during execution. • From register ID: Gets the output path from a register cell and then saves the image to that file. The register cell has to have the data type String.
Number of frames	<p>Set the number of frames after which the file will be locked for adding more frames and will be saved to disk.</p> <ul style="list-style-type: none"> • Fixed: The image will be closed after a fixed number of frames. • From register ID: Gets the number of frames from a register cell. The register cell has to have the data type Integer.

Element	Description								
Compression	<p>This value determines whether and how the Multi-Frame image frames are being compressed.</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>No compression</td> <td>Fast exports with the tradeoff of larger files.</td> </tr> <tr> <td>LZW (lossless)</td> <td>A compromise of fast export times and small files.</td> </tr> <tr> <td>JPEG (lossy)</td> <td>Small file but longer duration storage process. When using JPEG compression you can set the desired JPEG quality.</td> </tr> </tbody> </table> <p>Here you can find some examples regarding the export duration and file size.</p>	Option	Usage	No compression	Fast exports with the tradeoff of larger files.	LZW (lossless)	A compromise of fast export times and small files.	JPEG (lossy)	Small file but longer duration storage process. When using JPEG compression you can set the desired JPEG quality.
Option	Usage								
No compression	Fast exports with the tradeoff of larger files.								
LZW (lossless)	A compromise of fast export times and small files.								
JPEG (lossy)	Small file but longer duration storage process. When using JPEG compression you can set the desired JPEG quality.								
Write user comment from register	<p>Adds text from the provided register cell of type String as a comment to the TIFF frame. The comment is saved to the ImageDescription (0x010E) tag of the current frame with UTF-8 encoding.</p>								



You can use several instances of this check function to write to the same file. To achieve this, the output path in the parameter configurations must be identical and the file may not already contain the desired number of frames.



You can modify the desired number of frames of a multi-frame file if the new value is higher than the number of frames added previously.



If a file is incomplete or an error occurs due to faulty parameters, the current contents of the file are written and a log message is generated. In addition, the file name is changed so that it can be seen how many of the desired frames are contained in the file. Example: "_[File name] (1 of 4 frames).tiff"



Exported user comment text length should not exceed 200 000 characters per TIFF frame. Exact limit may depend on the total TIFF file size.

Save Multi-Frame Image From Tray: Introduction

Function overview

This check function takes an user defined amount of images from the NeuroCheck data tray and saves them into a single Multi-Frame image file.

Input data

This check function requires no input data object. It uses a fixed output path or determines it dynamically from the register and saves the resulting image to the hard disk.

Output data

The check function returns no output data object.


Result view

The result view signals a successful start of the save process which is executed as a background task.

In case of an error, the error message is displayed in the result view.

Properties

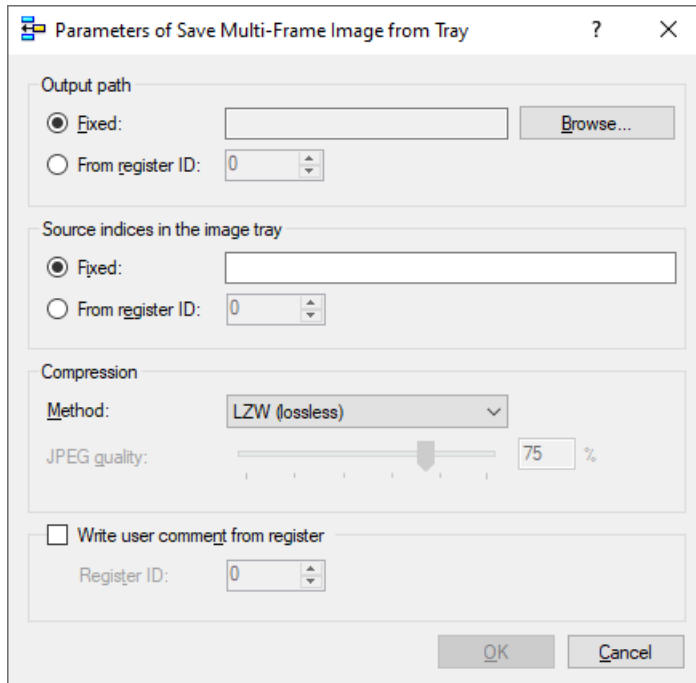
 Check function group Plug-In.

 The check function has a [Parameter Dialog](#).

Save Multi-Frame Image From Tray: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

☑ Screenshot of Parameter Dialog



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

Element	Description
Output path	<p>Choose the file path from the following options:</p> <ul style="list-style-type: none"> • Fixed: Saves the image with a static output path which wont change during execution. • From register ID: Gets the output path from a register cell and then saves the image to that file. The register cell has to have the data type String.
Source indices in the image tray	<p>Specify the data tray indices of the images you want to save. You can specify single indices by separating them with a comma (","), define index ranges by the following scheme ("[first index] - [last index]") or combine both ways.</p> <p>Example: If you want to save 6 data tray images you could enter "0,5,10-13".</p> <ul style="list-style-type: none"> • Fixed: The check function will always use the specified indices. • From register ID: The check function determines the indices (dynamically) from the Data Register Manager using the specified register ID. The register cell must be of type string.

Element	Description								
Compression	<p>This value determines whether and how the Multi-Frame image frames are being compressed.</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>No compression</td> <td>Fast exports with the tradeoff of larger files.</td> </tr> <tr> <td>LZW (lossless)</td> <td>A compromise of fast export times and small files.</td> </tr> <tr> <td>JPEG (lossy)</td> <td>Small file but longer duration storage process. When using JPEG compression you can set the desired JPEG quality.</td> </tr> </tbody> </table> <p>Here you can find some examples regarding the export duration and file size.</p>	Option	Usage	No compression	Fast exports with the tradeoff of larger files.	LZW (lossless)	A compromise of fast export times and small files.	JPEG (lossy)	Small file but longer duration storage process. When using JPEG compression you can set the desired JPEG quality.
Option	Usage								
No compression	Fast exports with the tradeoff of larger files.								
LZW (lossless)	A compromise of fast export times and small files.								
JPEG (lossy)	Small file but longer duration storage process. When using JPEG compression you can set the desired JPEG quality.								
Write user comment from register	Adds text from the provided register cell of type String as a comment to the TIFF frame. The comment is saved to the ImageDescription (0x010E) tag of all frames with UTF-8 encoding.								



Because the save operation is performed asynchronously in the background, an overflow of pending Multi-Frame image files must be prevented. This limit is hard set to 5 pending images. After the limit is reached, each following check function execution will synchronously wait for a previous save operation to complete.



If a file is incomplete or an error occurs due to faulty parameters, the current contents of the file are written and a log message is generated. In addition, the file name is changed so that it can be seen how many of the desired frames are contained in the file. E.g.: "_[File name] (1 of 4 frames).tiff"

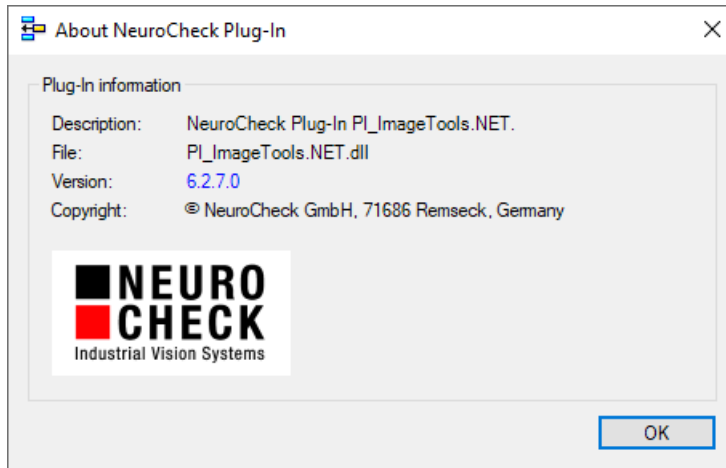


Exported user comment text length should not exceed 200 000 characters per TIFF frame. Exact limit may depend on the total TIFF file size.

About Dialog

This dialog displays version information about the NeuroCheck Plug-In **PI_ImageTools.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**.

