

Manuscript 3.2

i⁽²⁾PEPICO scripting language manual

31 August 2026 – Updated manual, including polygon image ROIs

Script generals

The script contains either one command or a remark in each non-empty line. Remarks are introduced by an exclamation mark as the first character in the line. Labels are specified by a colon as the 1st character in the line. The script is not case sensitive. There are two command types.

Assignment lines have a variable on the left, an assignment operator, `:=`, and an expression on the right. The variable name can be any non-reserved name. Function names and constants cannot be used.

Procedural commands have the function name followed by the function parameters in square brackets. When more than one parameter is given, they are separated by commas. Curly brackets, `{}`, indicate a mathematical expression, and they can contain variables, constants and function calls as well as standard mathematical functions in the usual notation.

String constants, i.e. for printing to the output file, have to be given in parentheses, otherwise they may be processed, e.g. converted to upper case and the spaces removed. Parentheses, however do not protect against all processing, parameters preceded by a backspace (`\`) have the backspace removed and are returned unchanged, in such cases.

Persistent results. The program is also capable of loading several files while keeping the results in memory so that TOF distributions and Image results can be manipulated across files. The results are stored in a slot indexed by the last, optional parameter of Load and can be accessed by passing an extra parameter to **Results**. The **TOFRes**, **Im1Res** and **Im2Res** shorthands can only be used to access the default, index 0 slot. The **Loaded*** functions may also take an extra parameter specifying which result set is to be accessed.

Script reference

- **Add**[TOF distribution 1, TOF distribution 2]
Adds the second TOF distribution to the first one.
- **AddCoincCrit**[CoincCrit] or **AddCoincCrit**[Event1, Event2...]
Adds a coincidence criterion set to the load routine. **CurrCoincCrit** returns the currently set coincidence criteria from the i^2 window, which can be used as a parameter. Alternatively, the individual events can be defined using the **AddEvent** function. At least one **AddCoincCrit** command has to come before any experimental data are loaded. If several criteria sets are defined, the resulting images and time-of-flight distributions can be accessed using the corresponding indexes in the **Results** function.
- **AddEvent**[Channel, TOFMin, TOFMax, ROI, TOF distribution role, Image role]
Returns a reference to a coincidence criterion definition for the given Channel (Ch1...Ch8, Im1, Im2), with the TOF window given by TOFMin and TOFMax (not used for the 1st event in

an event set), the image region of interest given by ROI (only used for Im1 and Im2), what role the event should play in the TOF distribution (TOFStart or TOFStop or Nil), and if an image is defined, which window it should belong to (Im1, Im2). Several **AddEvent** functions can be used as parameters for an **AddCoincCrit** call. If TOFMin and TOFMax are negative, the time-of-flight window is still defined by their absolute values, but relative to the TOFStop event in the coincidence train as opposed to the first event.

- **AddXY**[Series, X, Y]
Adds a new point to the Series specified as the first parameter.
- **ChannelCount**[Channel]
After measurement, **ChannelCount** returns the counts in channels Ch1...Ch8 as well as Im1 and Im2.
- **CleanCoincCrit**
Removes all coincidence criteria added using **AddCoincCrit**.
- **ClockReset**
After loading a file, **ClockReset** returns the number of times an event is preceded by an event with a later time stamp, i.e. the number of times the master clock has been reset.
- **ChannelFreq**[Channel]
After measurement, **ChannelFreq** returns the count rates in channels Ch1...Ch8 as well as Im1 and Im2.
- **Concat**[Param1, Param2...]
Evaluates the parameters and concatenates their value returning one string that can be used *e.g.* as filename.
- **CurrCoincCrit**
Returns a reference to the current coincidence criteria as defined in the i^2 window.
- **CurrCoincSwapROI**[Event index, New ROI]
Returns the currently set coincidence criteria set, but changes the ROI belonging to the specified event to the one given in the New ROI parameter.
- **CurrFullSample**
Returns the full measurement folder.
- **CurrInFile**
Return the current input file name from the main window.
- **CurrInFolder**
Returns the current input folder from the main window.
- **CurrIntTime**
Returns the integration time as set in the main window.
- **CurrOutFile**
Returns the current output file name as set in the main window.
- **CurrRingROIAsCircle**[Index]
Returns the inner circle of a ring ROI in image window specified by Index or an error if the current ROI is not a ring.
- **CurrROI**[Index]
Returns the current ROI from the image window specified by Index (1 or 2).
- **CurrSample**
Returns the current short sample name as set in the main window.

- **CurrScanIntTime**
Returns the scan integration time from the scan window.
- **CurrScanOut**
Returns the current output file name in the scan window.
- **CurrScanStart**
Returns the current start energy in the scan window.
- **CurrScanStep**
Returns the step size set in the scan window.
- **CurrScanStop**
Returns the maximum scan energy set in the scan window.
- **CurrSetE**
Returns the value from the monochromator control field in the main window.
- **CycFile**
In a **CycleFiles** cycle, **CycFile** returns the current file name together with the path.
- **CycleFiles[Folder, Pattern]**
Sets up a cycle in the folder given by the 1st parameter using the 2nd parameter, e.g. *.i2d, as pattern. In the cycle, the **CycFile** function returns the current file name, and the **Next** function can be used to return to the beginning of the cycle until there are no more files to process.
- **End**
Stops the script.
- **EPICS[ID]**
Returns the current formatted EPICS value for ID (see constants).
- **EPICSFloat[ID]**
As **EPICS**, but always returns an unformatted number, i.e. even for e.g. absorber position.
- **Equals[Param1, Param2] and Smaller[Param1, Param2] and Larger[Param1, Param2] and And[Param1, Param2] and Or[Param1, Param2] and XOR[Param1, Param2] and Not[Param1]**
Logical functions.
- **Exists[What]**
Checks if the What (e.g. an image or a TOF distribution) parameter exists.
- **FillLoadedParams**
Fills the information window with the loaded parameters.
- **For[Variable, Initial value, Final value]**
Sets variable to Initial value, runs till the corresponding **Next** command, increases the cycle variable by 1, and repeats until the cycle variable reaches Final value. If the cycle variable is FileCounter, it can be used in file names to index files using exclamation marks.
- **Gosub[Label]**
Continues the execution of the script at Label. Labels have a colon as the 1st character in a line. The execution returns
- **Goto[Label]**
Continues the execution of the script at Label. Labels have a colon as the 1st character in a line.
- **If[Condition, True branch, False branch]**
Conditional operation. The first parameter is evaluated, and if it returns a **True** value, i.e. a non-zero number, the True branch is executed, if **False**, i.e. zero, is returned, the False

branch takes control. As well as having an arithmetic formula to generate the Boolean value, see also **Equals** for the available logical functions.

- **ImageCenter**[Image, ROI]
Returns the center coordinates of Image, returned by **Results**, over the given region of interest. The results can be used e.g. to evaluate the image radius using **ImageRadius**.
- **ImageInt**[Image, ROI]
Returns the sum of counts in the Image over the region of interest given by the ROI parameter. The Image parameter is returned by **Results**, there are several functions that return image regions of interest, which can be used as the second parameter.
- **ImageIntP2**[Image, ROI, EVector]
Integrates the image by multiplying each pixel with the 2nd-degree Legendre polynomial of the cosine of the angle between the axis defined by the pixel and the ROI center and the E vector corresponding to the polarization of the light. The image ROI must be a circle or a ring. The result can be used in angular anisotropy calculations. The Image parameter is returned by **Results**, there are several functions that return image regions of interest, which can be used as the second parameter.
- **ImageRadius**[Image, ROI, Center]
Calculates the average radius for Image, as returned by **Results**, around Center, of the format (0.1;-0.1), taking the region of interest into account.
- **Load**[Folder, Name, TOFMin, TOFMax, TOFResolution, Result index]
Loads an experimental data set with the predefined coincidence criteria sets. The folder is an optional parameter specifying the file location. In **CycleFiles** cycles, the Name parameter already contains the folder, thus, the Folder parameter can be **Nil**. The Name parameter can also be **Memory**, indicating that the previous measurement to memory should be processed. TOFMin, TOFMax and TOFResolution stand for the desired TOF distribution parameters in μs . The optional Result index selects the persistent result slot and defaults to 0. Loading another dataset into the same slot replaces that slot's previous results.
- **LoadedChannelCount**[Channel]
As **ChannelCount**, but it applied to the loaded data set.
- **LoadedIdentifier**[Result index]
Returns the identifier stored in the loaded measurement. The optional result index defaults to 0.
- **LoadedMeasTime**
Returns the measurement time of the loaded data set.
- **LoadedMeasStart**
Returns the date and time at the start of the measurement of the loaded data set.
- **LoadedParam**[When, ID]
Similarly to **EPICS**, it returns the beamline parameters for a loaded measurement either at the **Start** or at the end (**Final**) of the measurement.
- **LoadedParamFloat**[When, ID]
As **LoadedParam**, but always returns a number.
- **Measure**[Sample, Output name, Integration time, Min. beam current, Refresh]
If the short sample name (not the full path) and a file name is given, **Measure** starts a measurement and saves the data to file with overwrite protection. And empty Sample and Memory as Output requests a measurement to memory, which can then be loaded and

processed using **Load** and discarded. If the optional parameter Min. beam current is given, the measurement does not start while the actual beam current is smaller the minimum. If the Refresh Boolean parameter is also given, it tells the program whether to refresh images and TOF distributions while the measurement is running (default) or not.

- **Multiply**[TOF distribution or image data, Factor]
Multiplies the loaded TOF distribution or image data with the given factor.
- **NewSeries**[Name]
Returns a reference to a new series, and changes the TOF window to 2D plot mode. The result of the **NewSeries** function can then be passed on to **AddXY** to add points to the plot. The new series will appear under the optional Name in the graph window.
- **Next**
Returns the execution to the beginning of the cycle. **Return** is a synonym for **Next**.
- **Nil**
- **Output**[Folder, Name]
Specifies the output file location and name. The file is created anew each time with overwrite protection. **Print** and **PrintLn** can be used to write to the output file.
- **PlotImage**[Image, ImageWindow]
Plots the image, given by the first parameter, in the image window specified by the second parameter. Images can be accessed after **Load** using the **Results** function, the ImageWindow can either be 1 or 2, depending on which window the image is to be plotted in.
- **PlotTOF**[TOF distribution]
Plots the specified TOF distribution, returned by the **Results** function, in the TOF window.
- **Point**[X, Y]
Can be used in ROI definitions for the center point, alternatively to the immutable and constant (X;Y) specification. It can also represent a polygon vertex.
- **Print**[Param1, Param2...]
Evaluates the parameters and prints their value (or the text) to the output file without starting a new line afterwards.
- **PrintEPICSFixed**[What]
Prints the parameter (should be a number) into the zoom line of the Beamline parameters (EPICS) window. Result is also shown in main window in the analysis program.
- **PrintEPICSFixedStr**[What]
Prints the parameter as string into the zoom line of the Beamline parameters (EPICS) window. Result is also shown in main window in the analysis program.
- **PrintLn**[Param1, Param2...]
The same as **Print** but does start a new line after the write operation.
- **ReadArray**[File name, Array variable]
The file is read and each line is assigned as a new item to array variable. The index of the first element is zero, i.e. if **ReadArray**[x.txt, ab] is executed and x.txt has 10 lines, ab[0] to ab[9] will be set. See also **WriteArray**.
- **Results**[Coincidence index, What, Result index]
Returns an image or TOF distribution from a loaded result set. What can be **Im1**, **Im2** or **TOF**. Coincidence index 1 refers to the first coincidence-criteria set added using **AddCoincCrit**, 2 to the second, and so on. The optional Result index selects the persistent result slot and

defaults to 0; it corresponds to the final optional parameter of **Load**. **Im1Res[Index]**, **Im2Res[Index]** and **TOFRes[Index]** are shorthand for the default result slot only.

- **ROIAll**
Returns an image region of interest for the whole image.
- **ROIArea[ROI]**
Returns the area of the given image region of interest in pixel numbers.
- **ROICircle[Center, Radius, Anisotropy]**
Returns a circular image region of interest, of which the center is given as, e.g., (0.1;-0.1).
- **ROICross[Center, Width]**
Returns a cross image region of interest, of which the center is given as, e.g., (0.1;-0.1), the width is a scalar.
- **ROIPolygon[Point1, Point2, Point3, ...]**
Returns a polygon image region of interest whose ordered vertices are given as (X;Y), for example: `ROIPolygon[(0.12;-0.05),(-0.08;-0.05),(0;0.20)]`. `Point[X,Y]` may be used instead of (X;Y). At least three non-collinear vertices are required. Coordinates must be between -10 and 10 and are rounded to three decimal places.
- **ROIring[Center, Radius1, Radius2, Anisotropy]**
Returns a ring image region of interest, of which the center is given as, e.g., (0.1;-0.1).
- **ROIToText[ROI]**
Converts the given image region of interest into a readable canonical text format to be printed. Whenever current ROIs are used in evaluation scripts, the analysis can only be repeated if the ROIs are printed in the output file. Note that this is not the Manuscript command to create a ROI, but the ROI signature.
- **SaveIm[Image, ROI, SaveROI, FileName, FileExt,Precision]**
Saves the loaded Image under the given ROI (can be ROIAll if the total image is to be saved) to the file named FileName, of which the extension is changed to “.img” with FileExt included before the extension (e.g. to allow for different ms-VMIs belonging to the same file). SaveROI can be True or False and tells the program whether to save the ROI in the file, as well. If the ROI is not saved, the output only consists of 400 × 400 numbers. The last optional parameter, Precision, specifies the number of digits to be printed after the decimal point (default 2).
- **SaveMyself[Folder]**
The source of the script is saved under the script’s name with overwrite protection in the specified folder.
- **SaveTOF[TOF distribution, TOFMin, TOFMax, FileName, FileExt, SaveTimes, Precision]**
Saves the loaded TOF distribution between times TOFMin and TOFMax to the file named FileName, of which the extension is changed to “.tof” with FileExt included before the extension (e.g. to allow for saving center and ring TOF distributions belonging to the same file). SaveTimes can be True or False and tells the program whether to include times of flight or only the ion counts per channel. The last parameter, Precision, is optional, and specifies the number of digits to be printed after the decimal point. The default, 0, is only reasonable for non-normalized and non-subtracted TOF distributions with solely integer counts in each TOF bin.
- **ScriptTime**
Returns the number of seconds the script has been running.

- **SetAxisLength[NewLength]**
Changes the axis length of the images to NewLength. The measurement and the data analysis programs have to be restarted every time the image size is changed. Starting September 2016, the image size is saved to file, and is changed automatically upon loading a file with a different image size. For earlier experimental data, the image size has to be adjusted manually prior to loading the file.
- **Set [Variable, Value]**
The Set function is equivalent to the assignment line Variable := Value, but can be invoked from within an If function, for instance.
- **SetE[E] or SetE[E, NoBlock]**
Sets the light energy to E and waits for the monochromator to reach the given energy. If NoBlock is specified as 2nd parameter, the execution continues immediately, and the script does not wait for the monochromator.
- **SetEPICS[ID, NewValue]**
Can be used to write a new value to an EPICS channel, i.e. close and open valves, gas filter set voltages directly etc. Channel IDs are not documented may change from version to version and need to be confirmed.
- **SetNoiselter[n]**
When the false coincidence background is subtracted in e.g. **TOFInt**, the noise subtraction is done n times (default 1 unless noise offset is set as 0). The first interval starts at TOFMin + NoiseOffset, the second one at TOFMin + NoiseOffset + 0.96 μ s and so on.
- **ShellExec[command]**
Executes the command (path, application name and optional parameters) and returns the output.
- **Sleep[Time]**
Pauses the execution for Time number of seconds.
- **StripPath[Name]**
Returns the filename part of a full file specification. Useful on **CycFile** when printing the path several times is not desirable.
- **SwitchEPICSLabels[OldLabel, NewLabel]**
Switch count rate labels in the EPICS window, to be used to include low-res HPTDC channels if required, e.g. by **SwitchEPICSLabels[Ch3, Ch9]**.
- **TOFCoG[TOF distribution, TOFMin, TOFMax, NoiseOffset, MomentIndex, MomentCenter]**
Returns the n^{th} moment of the given TOF distribution around MomentCenter. If MomentIndex is 1 and MomentCenter 0, this is equivalent to the peak center of gravity.
- **TOFInt[TOF distribution, TOFMin, TOFMax, NoiseOffset]**
Returns the integral of the TOF distribution between TOFMin and TOFMax in μ s. NoiseOffset (in μ s, a multiple of 0.96 μ s for high count rates) can be used to subtract the false coincidence background, or can be 0 when no subtraction is desired.
- **WaitTil[Time]**
Suspends the execution of the script until the given time passes. As Windows time and date formats can be almost anything, careful testing is required before the first use on any computer.

- **While**[Condition, Statement]
Evaluates the Condition, and executes Statement over and over again until Condition becomes False.
- **WriteArray**[File name, Array variable, Low limit, High limit]
The array variable is written (*without overwrite protection!*) to file as specified by the index range in the last two parameters. See also **ReadArray**.

Constants and special variables

Name	Meaning
Ch1, Ch2, Ch3, Ch4, Ch5, Ch6, Ch7 and Ch8	TDC channel references
Im1 and Im2	Image 1 and 2 references
MeasStart	Use as first coincidence criterion to define first recorded event as trigger
TOF	TOF distribution reference
Start and Final	Loaded measurement parameters at the beginning or at the end of the measurement
E	Monochromator energy channel
EF	Experimental chamber full range
SF	Source chamber full range
EV	Electron detector voltage (8A)
IV	Ion detector voltage (8B)
I_Ring	Beam current
SP	Source pressure
Memory	Measure to or load from memory
True and False	Logical values
TOFStart and TOFStop	Event role in TOF distribution
NoBlock	SetE parameter; not to wait for mono
FileCounter	Exclamation marks in filenames are replaced by it
ShowBusy	You can choose to show the “I’m busy”-window, which has been disabled by default.

Further EPICS channels can be referenced based on their ID, see channels.dat file.

For examples see the sample script.dat file.