ManoScript manual

i2PEPICO scripting language | Version 3.3 | 15 October 2026

Use ManoScript to acquire measurements, process closed data files, calculate coincidence results, and export analysis. This manual describes the script syntax and native commands, followed by runnable Export2D measurement and recovery examples. Check experimental settings before using the examples with hardware.

Script syntax

Each nonempty line contains one command or one remark. A remark begins with an exclamation mark as the first character; a label begins with a colon as the first character. Command names, variable names, labels, and import names are case insensitive.

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.

Put literal text in double quotation marks to retain spaces. The interpreter removes the quotation marks before evaluating the command. A leading backslash on a parameter escapes evaluation: it is removed, and the remaining parameter is returned literally. This is why Output examples use \2D_TPES_index.txt.

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.

Save interchange scripts with the .scr extension and CRLF line endings. script.dat is the application's script-tree storage; it is not an individual interchange script.

Labels and subroutines

A label is a line whose first character is a colon. An indented line is not a label. The first word after the colon is the label name. Any further words are the names of the subroutine's parameters, separated by spaces, not commas: :Normalize X Y declares the label Normalize with parameters X and Y. A parameter name starts with a letter or underscore and contains only letters, digits and underscores. It must not be a function name, a constant or FileCounter, and it must not appear twice on one label. Do not put a remark on a label line; put it on the line above. A label name may contain backslashes, for example :Imports\Helpers\Normalize. This is a convenient way to keep names unique, but the program does not interpret the backslashes. Each label name may occur only once in the whole run, including all imported scripts. All labels are checked before the first line runs. The errors are "Duplicate label", "Invalid parameter name", "Duplicate parameter", "Reserved name or constant used as parameter", "Empty label name" and "Comma in label". A subroutine ends with Return. Without Return, execution continues with the following lines, including the next label and the code after it.

Subroutine parameters

Gosub[Label, Value1, Value2…] passes one value per parameter, in order. All values are evaluated in the calling code first, and only then copied into the parameter variables. Parameters are ordinary variables, and all variables are global. A parameter overwrites any existing variable with the same name, keeps its value after Return, and can be used to return a result. Changing a parameter inside the subroutine does not change the variable that was passed in.

Arguments may calculate values or call ordinary functions, but must not perform control flow. Gosub, Goto, For, CycleFiles, Return, Next, End, and While in a Gosub argument raise "Control flow in Gosub argument". Curly-bracket calculations in a nested Gosub are evaluated before the containing If chooses its branch; calculate them into variables first when needed.

Imports

Subroutines can be kept in library scripts. In the script tree, create a folder named Imports directly under the root. This folder may contain only scripts, no subfolders, and the script names in it must be unique. A script uses libraries by naming them on its very first line, for example Imports[Helpers, Math]. The names are script names from the Imports folder, not paths. They must not contain \, /, spaces, commas or quotation marks. The Imports line must be the first line: a blank line or a remark above it switches imports off, and an Imports line anywhere else is an error. Libraries may import other libraries in the same way, and they can also be run on their own.

When the script starts, the program builds a run copy. The first line becomes the remark ! Imports[…] and is followed by the rest of the script. Each library then follows, with the libraries it imports placed before it. Each library is preceded by an End line and a remark ! Import: Name. A library is included only once, even if it is named several times. The End lines stop the script when execution runs past the end of one script, so it does not continue into the next. Libraries share all variables, loaded results and coincidence criteria with the calling script.

The run copy is fixed at start, so editing any script during the run has no effect. Changes to a library must be saved before the next start. While the script runs, line numbers match the editor for the main script; inside a library they continue past the end of the main script. If an error is found in a library while the run copy is built, the line number refers to that library. The errors at start are "Import not found", "Ambiguous import", "Imports folder contains a subfolder", "Invalid import name" and "Import is only valid as the first line". A malformed Imports line further down the script reports its syntax error instead (for example "Missing]" or "Empty argument list").

Command reference

Signatures describe required and optional arguments. Square brackets belong to the script syntax. An ellipsis means additional arguments of the same kind. Integration times and Sleep use seconds; TOF coordinates use microseconds. Result indices start at 0, while coincidence indices start at 1.

`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 i2 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 indices 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 first 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.

`AlarmMessaging[Enabled]`

Sets the alarm-messaging checkbox in the notification window. A nonzero numeric value enables it; zero or nonnumeric text disables it. Delivery uses the configured notification settings.

And[Param1, Param2]

Returns 1 when both arguments evaluate to nonzero numeric values; otherwise returns 0.

ChannelCount[Channel]

After measurement, ChannelCount returns the counts in channels Ch1…Ch8 as well as Im1 and Im2.

ChannelFreq[Channel]

After measurement, ChannelFreq returns the count rates in channels Ch1…Ch8 as well as Im1 and Im2.

CleanCoincCrit

Removes all coincidence criteria added using AddCoincCrit.

ClockReset or ClockReset[Result index]

Returns the number of master-clock resets recorded in the loaded result set. The optional result index defaults to 0.

Concat[Param1, Param2...]

Evaluates the parameters and concatenates their value returning one string that can be used for example as filename.

CurrCoincCrit

Returns a reference to the current coincidence criteria as defined in the i2 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

Measurement build: returns the absolute folder of the sample currently typed or selected in the main window, with a trailing backslash, creating it if necessary. Selecting a listed item is not required. A short sample name is resolved under the configured day folder; an absolute value is used as given and need not lie under the current day. Spaces, control characters, UNC and device paths, root-relative (\Sample) and drive-relative (C:Sample) forms, and a colon inside a folder name raise with the sample, folder and script line; no folder is created for a rejected value. Use this value for the Export2D measurement caller. The analysis build returns the configured sample folder as before.

CurrInFile

Returns 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 sample field from the main window. This may be a short sample name or an absolute folder path. Use CurrFullSample when a normalized complete folder path is required.

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 over matching files in Folder using Pattern, for example *.i2d. CycFile returns the current filename including its path. Next repeats the cycle until all matches have been processed. Matches are sorted. If no files match, CycleFiles raises immediately with the folder, mask, and script line, before entering the body.

DayRelativeSample[Folder]

Converts a raw folder under the configured day folder (Root\YYYY-MM\YYYY-MM-DD\) into the day-relative sample accepted by Measure, for example acetaldehyde\Run1. The day folder itself returns an empty sample. Folder is a folder path, not a sample name: a relative argument is expanded from the process working directory, like EnsureDirectory. Forward slashes, repeated separators, trailing separators and . or .. segments are normalized, and the day prefix is compared ordinally without case sensitivity. Before returning, the sample is checked to reconstruct exactly the same folder when Measure prefixes the day folder. A folder outside the current day, a sibling of the day folder, spaces in the folder or data root, UNC and device paths, root-relative and drive-relative forms, and a colon inside a folder name raise with the folder, day folder, reconstructed folder and script line. Never returns a parent traversal or an absolute fallback. No directory is created. The Export2DMeas wrapper uses it so that Measure receives the relative sample and the Meas field keeps showing it.

End

Stops the script.

EnsureDirectory[Path]

Creates missing parent directories synchronously and returns the normalized absolute directory path with a trailing backslash. Exactly one nonempty path is required. Relative paths use the process working directory. Existing directories and their contents are preserved. Spaces and Unicode are supported without a shell. A file blocking the requested path or a parent, or another creation failure, raises with the path and script line. Export2DOpen uses this command before selecting its index or writing spectra.

EPICS[ID]

Returns the current formatted EPICS value for ID (see constants).

EPICSFloat[ID]

As EPICS, but always returns an unformatted number, that is, even for for example absorber position.

Equals[Param1, Param2]

Returns 1 when both numeric values are equal (1 equals 1.0), or when the text values match without case sensitivity. Otherwise returns 0. Windows folder comparisons ignore case, but do not normalize trailing separators or relative paths.

Exists[What]

Checks if the What (for example an image or a TOF distribution) parameter exists.

FileExists[FullPath]

Returns numeric text 1 if the evaluated path is an existing file, otherwise 0. Directories, missing files and an empty path return 0. Exactly one argument is required. The native Unicode filesystem API is used without opening a persistent handle or invoking a shell. Relative paths use the process working directory. This command does not replace Exists, which still checks analysis-result objects. Export2DResume first flushes pending output, then uses FileExists to decide whether to print an initial header; matching scientific settings when resuming are the caller's responsibility.

FillLoadedParams or FillLoadedParams[Result index]

Fills the information window with parameters from the loaded result set. The optional result index defaults to 0.

FlushOutput or FlushOutput[]

Publishes everything printed so far, including an unfinished Print line after trimming its trailing tab. This also happens automatically before Output switches files and at script completion, including Stop, Terminate, and a guarded attempt on error. The whole output is written to the real filename plus .tmp, then moved onto the named file after closing the write handle. Every failed move is ignored: the named file stays unchanged and the complete .tmp remains for retry or manual recovery. A failed temporary write raises and attempts partial-file cleanup without changing the named file. With no active output, this command does nothing. Return does not flush. Use an explicit flush after PrintLn when an external program needs to monitor slowly written output.

For[Variable, Initial value, Final value]

Assigns the initial integer value, executes the body, and increments the variable by one at each Next. The final value is included. The body executes at least once, even if the initial value exceeds the final value. Use FileCounter to substitute exclamation marks in measurement filenames.

Gosub[Label, Value1, Value2...]

Calls the subroutine at Label. When the subroutine executes Return, execution continues on the line after the Gosub. If the label declares parameters (see Labels and subroutines), give one value for each, in order. A label without parameters takes no values. The values are evaluated before any parameter is set, so Gosub[Normalize, A, X] into :Normalize X Y sets X to the value of A and Y to the old value of X. Label may also be a variable or expression that gives the label name. A subroutine may call other subroutines on its own lines. "Wrong parameter count in line N" is raised if the number of values does not match. No value is evaluated and no jump happens. Gosub has no return value. When Gosub is used inside another function, such as a branch of If or the statement of While, expressions in curly brackets in its values are evaluated before the call, even when that branch is not taken. You may calculate such values into a variable first: T := {B + 1}, then If[C, Gosub[Normalize, A, T], Nil].

Goto[Label]

Continues execution at a label without parameters. A missing label raises "Label not found"; a label declaring parameters raises "Goto target has parameters". Use Gosub and Return for subroutines. Goto does not create a return frame.

If[Condition, True branch, False branch]

Conditional operation. The first parameter is evaluated, and if it returns a True value, that is, a non-zero number, the True branch is executed, if False, that is, 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.

Im1Res[Coincidence index]

Shorthand for Results[Coincidence index, Im1, 0]. This accesses the default result slot only; use Results to select another persistent result slot.

Im2Res[Coincidence index]

Shorthand for Results[Coincidence index, Im2, 0]. This accesses the default result slot only; use Results to select another persistent result slot.

ImageCenter[Image, ROI]

Returns the center coordinates of Image, returned by Results, over the given region of interest. The results can be used for example 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 second-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.

Larger[Param1, Param2]

Returns 1 only when both values are numeric and Param1 is greater than Param2; otherwise returns 0.

LastMeasuredFile or LastMeasuredFile[]

Returns the absolute resolved path of the last successfully completed file-backed Measure in this script run, including substituted counter digits and any overwrite-protected rename. The value is assigned only after the destination has been destroyed and its final buffered raw-data write succeeded. It is empty before a successful measurement, at the start of each run, after a failed new attempt, and after memory/no-destination measurements. Load does not change it. The getter is a reserved zero-argument command value, not an assignable script variable. In NOMEAS/analysis builds it remains empty.

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] or LoadedChannelCount[Channel, Result index]

Returns the channel count from a loaded result set. The optional result index defaults to 0.

LoadedIdentifier[Result index]

Returns the identifier stored in the loaded measurement. The optional result index defaults to 0.

LoadedMeasStart or LoadedMeasStart[Result index]

Returns the date and time at the start of the loaded measurement. The optional result index defaults to 0.

LoadedMeasTime or LoadedMeasTime[Result index]

Returns the integration time of the loaded measurement in seconds. The optional result index defaults to 0.

LoadedParam[When, ID] or LoadedParam[When, ID, Result index]

Returns the formatted beamline parameter for ID from the loaded measurement at Start or Final. The optional result index defaults to 0.

LoadedParamFloat[When, ID] or LoadedParamFloat[When, ID, Result index]

Returns the numeric beamline parameter for ID from the loaded measurement at Start or Final. The optional result index defaults to 0.

Measure[Raw folder, Output name, Integration time, Min. beam current, Refresh]

Acquires with the existing native three-, four- or five-argument forms. An absolute raw folder, or a short sample name under the configured day folder, and a filename select a file-backed measurement with counter substitution and overwrite protection. An empty output selects no destination; Memory selects a memory measurement. Minimum beam current is optional and waits before acquisition until the beam is sufficient. The optional fifth Refresh argument controls image/TOF refresh and defaults to True. LastMeasuredFile is cleared before each attempt and becomes the absolute resolved path only after successful file-backed acquisition and its final buffered write. The imported Meas wrapper is called as Gosub[Meas, RawFolder, Pattern, Duration, MinBeam]; it deliberately uses four actual arguments and processes the completed path outside native Measure. Stop/Terminate retain their existing semantics; interrupted acquisition does not guarantee a scientifically complete measurement.

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

Does nothing and returns an empty value, for example as an unused branch of If or an omitted Folder in Load.

Not[Param1]

Returns 1 only when the argument is numeric and zero. Returns 0 for nonzero numeric values and for nonnumeric text.

Or[Param1, Param2]

Returns 0 only when both arguments are numeric and zero. Otherwise returns 1, including when either argument is nonnumeric.

Output[Folder, Name] or Output[Folder, Name, Mode]

Selects the output location and filename. Two arguments or Mode 1 protect an existing file by choosing a numbered alternative. Mode 0 replaces the requested canonical name. Mode 2 resumes that name: existing text is loaded once, with its detected encoding and line boundaries, then subsequent Print/PrintLn text is published through the same whole-buffer temporary-file mechanism. A missing file starts with the usual filename line. An existing empty file remains empty until text is printed. An unreadable existing file raises before the active output is flushed or changed. Reopening the same active append destination retains pending text and unpublished buffered rows instead of rereading stale disk content. Mode must evaluate to 0, 1 or 2; count and mode are checked before switching. Both .out and .txt are preserved, with .txt used for other names. Selection does not create or truncate the file. Switching destinations flushes the old output, including a pending line. All failed temporary-file moves remain ignored. Use EnsureDirectory first for missing parents; see FlushOutput for publication and error behavior.

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 appends their values to the current output line. A tab follows each value. Text accumulates until PrintLn, FlushOutput, an output switch, or script completion finishes the line.

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, ...] or PrintLn[]

Appends values as Print does, removes the final tab, and finishes the pending line. PrintLn[] finishes an existing pending line; it does nothing when no text is pending. It does not by itself force publication to disk.

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, that is, if ReadArray[x.txt, ab] is executed and x.txt has 10 lines, ab[0] to ab[9] will be set. See also WriteArray.

RequestSubtract[Image]

Returns 1 when the selected image window exposes an enabled and checked subtraction control; otherwise returns 0. Pass Im1 or Im2.

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, for example, (0.1;-0.1).

ROICross[Center, Width]

Returns a cross image region of interest, of which the center is given as, for example, (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, for example, (0.1;-0.1).

ROIRingAsCircle[ROI]

Returns a circle ROI with the supplied ring's center, inner radius and anisotropy. The original ring is unchanged. The argument must be a ring ROI, including a stored handle or CurrROI for a selected ring. Use ROIRingAsCircle[Ring] as the Center argument to avoid separately specifying matching circle geometry.

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 Manoscript 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 (for example 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]

Saves the script as it is being run, under the script's name with the extension .scr and overwrite protection, in the specified folder. For a script with Imports, this is the run copy: the first line is a remark and all imported libraries are included, so the saved file runs without the Imports folder. The same text is recorded in the logbook when the script starts.

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 (for example 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.

SendMessage[Text]

Passes a script message to the notification window. Delivery depends on the configured notification settings.

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.

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.

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 second 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, that is, close and open valves, gas filter set voltages directly etc. Channel IDs are not documented, may change, and need to be confirmed.

SetNoiseIter[n]

When the false coincidence background is subtracted in for example 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 µs and so on.

SetSlit[TRXR, X, TRYB, TRYT]

Moves the front-end slits to the given positions in mm, for example SetSlit[-25, X, -20, 20]. Allowed values (inclusive): TRXR from -25 to 0, TRYB from -20 to 0, TRYT from 0 to 20. Values are never clamped. The second argument (TRXW) is ignored because that motor is disabled; use a placeholder such as X. It is evaluated like any other argument (a variable is read, a bracket call runs), but its value is discarded. All three active values are checked before any motor is written; a wrong argument count or an invalid value stops the script with an error naming the line.

After writing the three positions, SetSlit waits 3 s, then checks once per second until all three motors report done (DMOV = 1), and finally waits a further 1 s. A request to the current position completes normally. If any motor is still not done 90 s after the positions were written (the initial 3 s included), the script stops with an error listing the three DMOV values. Stop takes effect only after SetSlit has finished; Terminate interrupts the wait but does not stop the motors. The slit set and done channels are not logged; the existing slit readbacks are logged as before. In the analysis program, SetSlit does nothing.

SetVoltageSet[Name]

This registered legacy command is unsupported. Every call raises "SetVoltageSet support ended". It does not apply a voltage set.

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.

Smaller[Param1, Param2]

Returns 1 only when both values are numeric and Param1 is less than Param2; otherwise returns 0.

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, for example by SwitchEPICSLabels[Ch3, Ch9].

TOFCoG[TOF distribution, TOFMin, TOFMax, NoiseOffset, MomentIndex, MomentCenter]

Returns the nth 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 µs. NoiseOffset (in µs, a multiple of 0.96 µs for high count rates) can be used to subtract the false coincidence background, or can be 0 when no subtraction is desired.

TOFRes[Coincidence index]

Shorthand for Results[Coincidence index, TOF, 0]. This accesses the default result slot only; use Results to select another persistent result slot.

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.

`XOr[Param1, Param2]`

Returns 1 when exactly one argument evaluates to a nonzero numeric value; otherwise returns 0.

Constants and special variables

The following names are reserved. Further EPICS channels are identified by their numeric ID in i2PEPICO's channels.dat; verify the installed beamline configuration before using an ID.

Name	Meaning
Ch1 through Ch16	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
On	1
Off	0
ShowBusy	Enables the busy window; disabled by default.

Completed measurement export

The script.dat of the analysis package already contains Export2D, Export2DSLS1 and Export2DSLS2 in its top-level Imports folder; ScriptImport.exe adds or updates them in any other script.dat. Measurement scripts also need Export2DMeas in the same folder. The following examples are main scripts. Paste the measurement example into a Measure script and the rebuild example into a Process script, then use the ordinary start control. Imports must be the literal first line. The Measure example uses SLS2. For rebuilding historical SLS1 files, set UseSLS1 := 1 immediately after Imports in the Process example.

Check the ring, filenames, energy ranges, integration time and minimum beam current before running. ROIRingAsCircle derives the matching center from the ring. CurrFullSample gives the full configured raw sample folder; CurrInFolder selects the rebuild input. Each raw folder has a fixed PIMP child. Native filename protection and counter substitution still apply.

Measure two energy ranges

```
Imports[Export2DMeas]
Ring := ROIRing[(-0.109;-0.134),0.03,0.06,1]
Gosub[Export2DMeasureOpen, CurrFullSample, ROIRingAsCircle[Ring], Ring]
For[FileCounter,0,9]
SetE[{10.525+FileCounter*0.010}]
Gosub[Meas, CurrFullSample, acetone_d6.DD2!!!.i2d, 3000, 380]
Next
For[FileCounter,0,12]
SetE[{10.5475+FileCounter*0.005}]
Gosub[Meas, CurrFullSample, acetone_d6.DD3!!!.i2d, 3000, 380]
Next
```

Meas takes exactly four arguments: full raw folder, filename pattern, integration time and minimum beam current. It acquires with the native Refresh default, processes LastMeasuredFile after its final raw write, saves Center/Ring/All TOFs and publishes one index row. Native Measure contains no automatic Export2D hook. The raw folder must lie under the current day folder without spaces: Export2DMeasureOpen and Meas reject other folders with DayRelativeSample before creating raw or PIMP folders, and Meas passes the checked day-relative sample to native Measure, so the sample field stays relative.

Rebuild closed measurements

```
Imports[Export2D]
Ring := ROIRing[(-0.109;-0.134),0.03,0.06,1]
Gosub[Export2DRebuild, CurrInFolder, *.i2d, ROIRingAsCircle[Ring], Ring]
```

Press Stop in PIMP and run this while acquisition/export is idle. Rebuild replaces the canonical index and selected derived TOFs, keeping raw measurements and unrelated files. A mask with no matches raises before replacement output opens. Press Play in PIMP after completion to reread all spectra. Resume flushes pending rebuild output before checking whether its header exists.

No caller entry label, tail End, FlushOutput, Close or EnsureDirectory is required. The interpreter inserts End boundaries before imported libraries and flushes at completion. Use Export2DClose only when switching sessions within one run. Repeated MeasureOpen for the same active folder keeps its ROIs and settings; Close/Open is required to change them. Consistent scientific settings on resume remain the caller's responsibility. An interrupted run may publish partial output; failed temporary-file moves retain their existing behavior.

For PIMP controls and analysis workflows, use the PIMP user guide. This manual is the reference for ManoScript itself. Other sample scripts are stored in the application's script.dat.