PEARL Procedures

Igor procedures for the analysis of PEARL data

Version rev-distro-2.1.1-1-gf419e92-dirty

Tue Jun 16 2020 14:32:36

Contents

Chapter 1

Introduction

1.1 Introduction

PEARL Procedures is a suite of Igor Pro procedures developed for data acquisition and data processing at the PEARL beamline at the Swiss Light Source.

1.2 Installation

PEARL Procedures should be installed according to the regular Igor Pro guidelines. Please read the Igor help `About Igor Pro User Files` for details.

- Make a `pearl-procs` directory in your private or shared `User Procedures` folder, and copy the PEARL Procedures distribution there.
- Create shortcuts of the `pearl-arpes.ipf` and `pearl-menu.ipf` files, and move them to the `Igor Procedures` folder next to your `User Procedures` folder.
- Find the `HDF5.XOP` extension in the `Igor Pro Folder` under `More Extensions/File Loaders`, create a shortcut, and move the shortcut to the `Igor Extensions` folder next to your `User Procedures` folder.
- Find the `HDF5 Help.ihf` next to `HDF5.XOP`, create a shortcut, and move the shortcut to the `Igor Help Files` folder next to your `User Procedures` folder.

1.3 License Information

Author

Matthias Muntwiler, `matthias.muntwiler@psi.ch`

Version

This documentation is compiled from version rev-distro-2.1.1-1-gf419e92-dirty.

Copyright

Chapter 2

Projections

the functions of the anglescan package support the following map projections.

for a description of the different projections, see, for example, https://en.wikipedia.org/wiki/Map_projection

Selector	Projection	Function	Properties
kProjDist = 0	azimuthal equidistant	r = c $*$ theta	radius is proportional to polar angle.
kProjStereo = 1	stereographic	r = c $*$ tan theta/2	circles on sphere map to circles.
kProjArea = 2	azimuthal equal-area	r = c $*$ sin theta/2	preserves area measure.
kProjGnom = 3	gnomonic	r = c $*$ tan theta	great circles map to straight lines.
kProjOrtho = 4	orthographic	r = c $*$ sin theta	k-space mapping in ARPES and LEED.

the projections in this package are defined for 0 $<=$ theta $<$ 90.

Chapter 3

Angle-scan processing

3.1 Introduction

This page describes the data processing steps of angle-scans using the PEARL Procedures. The description relies on using the command line regardless of available GUIs.

3.2 Data reduction

The goal of this step is to import raw data and at the same time eliminate the energy dimension. We want a two-dimensional wave where the first dimension is the angle axis of the detector and the second dimension is the sequence of measurements, scanning one or multiple manipulator angles. The second dimension requires additional one-dimensional waves that describe the polar, tilt and azimuthal angle setting of the manipulator for each dimension index.

The processing steps depend on the complexity of the measured spectrum. The user may have to adopt one of the predefined or a custom procedure accordingly. Here, we describe two procedures that may cover many generic cases or that can serve as a starting point for a refined, customized procedure. However, any procedure that produces the datasets mentioned above is, of course, a valid approach. For instance, you could load the complete three-dimensional ScientaImage dataset, and generate the two-dimensional dataset using your own procedures.

3.2.1 Basic steps

The central import functions are psh5_load_reduced and psh5_load_dataset_reduced. The first form is sufficient if the file contains just one scan and region. Further regions/scans need to be loaded using the second form. The first form is also exposed in the PEARL data explorer window.

The functions require a data reduction function and processing parameters as arguments. Some particular reduction functions are described further below. More can be found in the source code (or obtained from other users). A list of functions that look like reduction functions can be got from adh5_list_reduction_funcs.

The basic call sequence looks as follows. Substitute the arguments in angle brackets as necessary. You may have to analyse a reference spectrum or the complete ScientaImage to figure out the processing parameters beforehand.

First form:

```
setdatafolder root:
string sparam
sparam = "<param1=1.5;param2=test;>"
psh5_load_reduced("<igor-datafolder>", "<igor-filepath>", "<filename>", <
      reduction_function>, sparam)
```

Second form:

```
// open the file
setdatafolder root:   // or other parent folder
variable fid
string sparam
fid = psh5_open_file("<igor-datafolder>", "<igor-filepath>", "<filename>")

// load metadata for scaling
psh5_load_scan_meta(fileID, "<scan 1>")
newdatafolder /s /o attr
psh5_load_scan_attrs(fileID, "<scan 1>")
setdatafolder ::

// load and reduce dataset
sparam = "<param1=1.5;param2=test;>"
psh5_load_dataset_reduced(fid, "<scan 1/region1>", "<ScientaImage>", <
      reduction_function>, sparam)

// close the file
psh5_close_file(fid)
fid = 0
```

3.2.2 Peak integration over linear background

The int_linbg_reduction function converts a two-dimensional Scienta image I(angle, energy) into a one-dimensional angle distribution I(angle). For each angle slice, it calculates a linear background. Then, it integrates the difference between the original data and the background over a specified interval.

The function requires the following, fixed parameters:

Parameter	Description	Typical value
Lcrop	size of the low-energy cropping region	0.11 (fixed mode)
Lsize	size of the low-energy background region	0.2
Hcrop	size of the high-energy cropping region	0.11
Hsize	size of the high-energy background region	0.2
Cpos	position of the peak center	0.5
Csize	size of the center region	0.3

All parameters are relative to the size of the image (length of the energy interval) and must be in the range from 0 to 1.

The cropping region is cut away from the image for the rest of the processing. This is necessary to remove the dark corners in fixed mode but can be neglected in swept mode (cropping size = 0).

The low and high background regions are adjacent to the cropping regions on either side. The function calculates two fix points of the linear background in the center of each background region. The intensity value of each fix point is the average intensity in the background region.

The peak region is integrated over the integral given by the Csize parameter centered at Cpos.

The background-subtracted peak integral is returned in ReducedData1. ReducedData2 receives the error estimate of the peak integral (assuming Poisson statistics).

3.2.3 Peak fitting

The gauss4_reduction function converts a two-dimensional Scienta image I(angle, energy) into a one-dimensional angle distribution I(angle). For each angle slice, it performs a Gaussian curve fit with up to four components on a linear background.

To improve the stability of the fit, the peak positions and widths are kept fixed while the amplitudes of the peaks and the background parameters are variable but constrained to reasonable values (positive amplitude). Furthermore, the function can optionally do a box averaging over three slices.

The function requires the following, fixed parameters:

Parameter	Description
rngl	lower limit of the fit interval
rngh	upper limit of the fit interval
npeaks	number of components
pos1	center energy of peak 1
wid1	width of peak 1
pos2	center energy of peak 2
wid2	width of peak 2
pos3	center energy of peak 3
wid3	width of peak 3
pos4	center energy of peak 3
wid4	width of peak 3
ybox	box size of slice averaging (1 or 3)

The peak parameters should be determined beforehand from fitting a reference spectrum, or the angle-scan integrated over all angles. Peak positions and widths have to be specified only up to the given number of peaks.

The data reduction procedure returns the peak integrals (amplitude times width times square root of 2) in waves named ReducedDataN where N is a numeric index from 1 to npeaks. The waves starting with an index of npeaks+1 contain the corresponding error estimate of the peak integral.

3.2.4 Custom reduction functions

See the documentation and source code of int_linbg_reduction, gauss4_reduction and adh5_default_reduction for help on writing custom reduction functions. To integrate your function with the PEARL data explorer, you have to provide an additional function that prompts for reduction parameters such as prompt_int_linbg_reduction, for example. Since reduction functions cannot be called from the command line, it is redommended to also write an adapter function for testing.

3.3 Normalization

The goal of the data normalization is to get a (still two-dimensional) dataset that ideally contains intensity variations due to diffraction features and statistical fluctuations only. In particular, instrumental variations should be removed. In some cases, it may be necessary to preserve the overall polar dependence of the intensity. Note that this latter case is not properly treated with the methods described here.

Depending on the quality of the measured data, only some of the following processing steps are necessary. Use your own judgement.

There is a GUI for the processing steps in pearl-anglescan-panel.ipf (asp_show_panel function or the PEARL/process menu).

3.3.1 Preparations

Start by creating a new copy of the data and inspecting it:

```
duplicate ReducedData1, NormData1
ad_display_profiles(NormData1)
```

To update the display after changes to NormData1:

```
ad_update_profiles(NormData1)
```

3.3.2 Detector angle range

Crop the detector angle axis to a useful range (usually about -25 to +25 degrees):

```
crop_strip(NormData1, -25, 25)
```

3.3.3 Normalize detector angle

Remove inhomogeneity of the detector in the detector angle axis. This component may also include a contribution from the sample. If your raw data shows a flat distribution, this step is not necessary.

```
normalize_strip_x(NormData1, smooth_method=4, smooth_factor=0.15, check=2)
```

Note that the argument `check=2` causes the function to generate two check waves but not to modify the original data. To inspect the check waves:

```
display check_dist, check_smoo
ModifyGraph rgb(check_dist)=(0,0,0)
```

Vary the `smooth_factor` (between 0.1 and 1.0) until it follows the instrumental curve but does not affect diffraction features. Then set `check=1` to apply the normalization to `NormData1`.

3.3.4 Azimuthal variation (wobble)

Reduce the effect of azimuthal wobble (misaligned surface) on intensity. A misaligned surface may cause a sinusoidal variation of the intensity as a function of azimuthal angle with a 360° period. A strong azimuthal variation may affect the polar normalization in the next step. The azimuthal normalization can be based on a restricted range of polar angles (theta range). You have to find out which value works best for your sample.

```
normalize_strip_phi(NormData1, :attr:ManipulatorTheta, :attr:ManipulatorPhi,
      theta_offset=-8.8, theta_range=10, check=2)
```

Note, however, that his function does not correct for angle shifts induced by the misalignment!

3.3.5 Polar dependence

Remove the polar angle dependence (matrix element and excitation/detection geometry).

```
normalize_strip_theta(NormData1, :attr:ManipulatorTheta, theta_offset=-8.8,
      smooth_method=4, smooth_factor=0.5, check=2)
```

Use the check waves and the `check` argument as described above.

3.4 Binning and plotting

3.4.1 Basic steps

You can bin and plot the data in one step:

```
pizza_service(NormData1, "Nickname1", -8.8, 0.5, 6)
```

or two steps:

```
pizza_service(NormData1, "Nickname2", -8.8, 0.5, 6, nograph=1)
display_hemi_scan("Nickname2")
```

The benefit of the latter is that you have more control over the graph through optional arguments. In particular, you can select the projection or hide the ticks and grids. See display_hemi_scan for details.

The pizza_service function requires the waves with manipulator positions in a specific place, namely `:attr↩ :ManipulatorTheta` (for the polar angle), and the normal emission values as function arguments. If you have moved the waves, or if you have subtracted the offsets yourself, use the alternative pizza_service_2 function.

Additional parameters of the pizza_service function allow for rotational averaging, larger angle steps (default 1 degree), or the creation of metadata including a notebook for xpdPlot.

Note there is currently a bug in the nick name argument of some of the following functions. If the lines shown below do not work, try to switch to the data folder that contains the generated polar plot data, and call the function with an empty nickname `""`.

3.4.2 Refinements

To remove high polar angles above θ = 80 from the plot (and data):

```
trim_hemi_scan("Nickname1", 80)
```

Modify the pseudocolor scale by changing the `polarY0` trace:

```
ModifyGraph zColor(polarY0)={mod_values, *, *, BlueGreenOrange, 0}
ModifyGraph zColor(polarY0)={mod_values, -0.2, 0.2, BlueGreenOrange, 0}
```

To set the contrast to clip specified percentiles of the data points, use the

- set_contrast function:

```
set_contrast(2, 2, graphname="graph_Nickname1", colortable="BlueGreenOrange")
```

3.4.3 Interpolation

Polar plots can also be interpolated to a rectangular matrix, which may in some cases produce nicer images:

```
interpolate_hemi_scan("Nickname1")
display_hemi_scan("Nickname1", graphtype=3, graphname="intp")
matrix = sqrt(x^2 + y^2) <= calc_graph_radius(80) ? matrix : nan
ModifyImage matrix ctab= {*,*,BlueGreenOrange,0}
```

The `matrix =` line optionally removes artefacts at high polar angles. Replace the cut-off angle with your own.

3.4.4 Modulation function

To calculate the modulation function and substitute it in the graph:

```
setdatafolder Nickname1
calc_modulation(values, factor1=pol, factor2=az)
ModifyGraph zColor(polarY0)={mod_values,-0.2,0.2,BlueGreenOrange,0}
```

3.4.5 Projection

The display_hemi_scan and interpolate_hemi_scan functions take an optional argument `projection` which selects one of the following projections. By default, stereographic projection is selected.

Selector	Projection	Function	Properties
kProjDist = 0	azimuthal equidistant	r = c ∗ theta	radius is proportional to polar angle

Selector	Projection	Function	Properties
kProjStereo = 1	stereographic	r = c $*$ tan theta/2	circles on sphere map to circles
kProjArea = 2	azimuthal equal-area	r = c $*$ sin theta/2	preserves area measure
kProjGnom = 3	gnomonic	r = c $*$ tan theta	great circles map to straight lines
kProjOrtho = 4	orthographic	r = c $*$ sin theta	momentum mapping in ARPES and LEED

For a description of the different projections, see Wikipedia, for example. The projections in this package are defined for 0 <= theta < 90.

3.5 Data export

3.5.1 Export picture

The following line is an example of how to export a graph window. Click on the desired graph window, then issue the following command, substituting the file path and file name as appropriate.

```
SavePICT/P=home/E=-5/B=144/O as "some_filename.png"
```

3.5.2 Export processed data

The following line saves the dataset to an Igor text file. The file contains all data necessary to recreate a polar plot without further processing.

```
save_hemi_scan("Nickname1", "home", "some_filename")
```

For structural optimization using the PMSCO software, it is necessary to generate an ETPI file. There is currently no special function for this. Instead, you have to create and set an energy wave,

```
duplicate pol, en
en = 123.4   // kinetic energy of the photoelectron
```

and write the four waves `en, pol, az, values` to a general text file. Be careful about the ordering of the waves! You will also have to rename the file to the `.etpi` extension because Igor always saves with `.txt` extension. If you have a wave with statistical errors, add a fifth column and use the `.etpis` extension.

```
Save /G /M="\n" /O /P=home en, pol, az, values, sig as "Nickname1.etpis.txt"
```

Chapter 4

Todo List

Global adh5_scale_scan (wave data)

incomplete

Global elog_validate_attributes (string logbook, string attributes)

function currently not implemented, always returns 0

Global epics_connect ()

the X03DA channel names are hard-coded.

Global load_hemi_scan (string nickname, string pathname, string filename)

function not implemented

Global mtrx_open_file (string pathName, string fileNameOrPath)

fix possible cache issues, add an option to override the cache.

Chapter 5

Module Index

5.1 Packages

Packages are collections of procedure files to be loaded together. PEARL defines the following packages:

Chapter 6

Namespace Index

6.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 7

Data Structure Index

7.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 8

File Index

8.1 File List

Here is a list of all files with brief descriptions:

Chapter 9

Module Documentation

9.1 ARPES package

data acquisition and analysis package for ARPES at PEARL.

Files

- file fermi-edge-analysis.ipf

  tools for analysing the Fermi edge measured by the Scienta EW4000 analyser.
- file pearl-anglescan-panel.ipf

  interactive processing of angle scanned XPD data.
- file pearl-anglescan-process.ipf

  processing and holographic mapping of angle scanned XPD data.
- file pearl-area-display.ipf

  visualization tools for 2D and 3D data.
- file pearl-area-import.ipf

  HDF5 file import from EPICS area detectors.
- file pearl-area-profiles.ipf

  profile extraction for multi-dimensional datasets acquired from area detectors.
- file pearl-compat.ipf

  compatibility procedures for igor 8
- file pearl-data-explorer.ipf

  preview and import panel for PEARL data
- file pearl-elog.ipf

  interface for writing ELOG entries with Igor graphs as attachment.
- file pearl-fitfuncs.ipf

  various fit functions for photoelectron spectroscopy.
- file pearl-pmsco-import.ipf

  data import/export procedures for multiple scattering calculations.
- file pearl-pshell-import.ipf

  import data from PShell
- file pearl-scienta-countrate.ipf

  count rate functions for Scienta detector images.
- file pearl-scienta-preprocess.ipf

  preprocessing functions for Scienta detector images.
- file pearl-vector-operations.ipf

  basic vector geometry operations.

9.1.1 Detailed Description

data acquisition and analysis package for ARPES at PEARL.

The purpose of a package is to load a bunch of dependent procedure files. The ARPES package loads the following files which are helpful when working with ARPES data (any data from the Scienta analyser, that is) from PEARL.

- pearl-area-import.ipf
- pearl-area-display.ipf
- pearl-area-profiles.ipf
- pearl-data-explorer.ipf
- pearl-scienta-preprocess.ipf
- pearl-anglescan-process.ipf
- pearl-anglescan-panel.ipf
- pearl-anglescan-tracker.ipf
- pearl-elog.ipf

Most of these files require the HDF5.XOP. The following files are loaded if the EPICS.XOP is present:

- pearl-area-live.ipf
- pearl-epics.ipf
- pearl-arpes-scans.ipf
- pearl-sample-tracker.ipf

Chapter 10

Namespace Documentation

10.1 PearlAnglescanPanel Namespace Reference

interactive processing of angle scanned XPD data.

10.1.1 Detailed Description

interactive processing of angle scanned XPD data.

PearlAnglescanPanel is declared in pearl-anglescan-panel.ipf.

10.2 PearlAnglescanProcess Namespace Reference

processing and holographic mapping of angle scanned XPD data.

10.2.1 Detailed Description

processing and holographic mapping of angle scanned XPD data.

PearlAnglescanProcess is declared in pearl-anglescan-process.ipf.

10.3 PearlAreaDisplay Namespace Reference

instant visualization of angle scan and manipulator position.

10.3.1 Detailed Description

instant visualization of angle scan and manipulator position.

- 3D graphics in a Gizmo window
- 2D graphics with profiles
- Slicer panel

data dictionary of global variables in view_xxxx folder:

- gizmo_graphname = name of the gizmo window

- slice_graphname = name of the slice/profiles graph window
- slice_panelname = name of the slicer panel

10.3.2 Miscellaneous

```
scienta HDF5 import, display, and slicer how-to

/// ad_display_brick(root:x03da_scienta_20130821_01560:data)
/// ad_scale_scienta(root:x03da_scienta_20130821_01560:data)
/// ad_scale_extra(root:x03da_scienta_20130821_01560:data,root:x03da_scienta_20130821_01560:MonoEnergy,root:x0
/// ad_display_brick(root:x03da_scienta_20130821_01560:data)
/// ad_brick_slicer(root:x03da_scienta_20130821_01560:data, "gizmo_x03da_scienta_20130821_01")
///

@author matthias muntwiler, matthias.muntwiler@psi.ch

@copyright 2013-15 Paul Scherrer Institut @n
Licensed under the Apache License, Version 2.0 (the "License"); @n
  you may not use this file except in compliance with the License. @n
  You may obtain a copy of the License at
  http://www.apache.org/licenses/LICENSE-2.0
```

PearlAreaDisplay is declared in pearl-area-display.ipf.

10.4 PearlAreaImport Namespace Reference

HDF5 file import from EPICS area detectors.

10.4.1 Detailed Description

HDF5 file import from EPICS area detectors.

PearlAreaImport is declared in pearl-area-import.ipf.

10.5 PearlAreaProfiles Namespace Reference

profile extraction for multi-dimensional datasets acquired from area detectors.

10.5.1 Detailed Description

profile extraction for multi-dimensional datasets acquired from area detectors.

PearlAnglescanTracker is declared in pearl-area-profiles.ipf.

10.6 PearlArpes Namespace Reference

data acquisition and analysis package for ARPES at PEARL.

10.6.1 Detailed Description

data acquisition and analysis package for ARPES at PEARL.

PearlArpes is declared in pearl-arpes.ipf.

10.7 PearlCompat Namespace Reference

compatibility procedures for igor 8

10.7.1 Detailed Description

compatibility procedures for igor 8

PearlCompat is declared in pearl-compat.ipf.

10.8 PearlDataExplorer Namespace Reference

preview and import panel for PEARL data

10.8.1 Detailed Description

preview and import panel for PEARL data

PearlDataExplorer is declared in pearl-data-explorer.ipf.

10.9 PearlElog Namespace Reference

interface for writing ELOG entries with Igor graphs as attachment.

10.9.1 Detailed Description

interface for writing ELOG entries with Igor graphs as attachment.

PearlElog is declared in pearl-elog.ipf.

10.10 PearlFitFuncs Namespace Reference

various fit functions for photoelectron spectroscopy.

10.10.1 Detailed Description

various fit functions for photoelectron spectroscopy.

PearlFitFuncs is declared in pearl-fitfuncs.ipf.

10.11 PearlMatrixImport Namespace Reference

data file import for omicron matrix (STM) files

10.11.1 Detailed Description

data file import for omicron matrix (STM) files

PearlMatrixImport is declared in pearl-matrix-import.ipf.

10.12 PearlPmscoImport Namespace Reference

data import/export procedures for multiple scattering calculations.

10.12.1 Detailed Description

data import/export procedures for multiple scattering calculations.

PearlPmscoImport is declared in pearl-pmsco-import.ipf.

10.13 PearlPShellImport Namespace Reference

import data from PShell

10.13.1 Detailed Description

import data from PShell

PearlPShellImport is declared in pearl-pshell-import.ipf.

10.14 PearlScientaCountrate Namespace Reference

count rate functions for Scienta detector images.

10.14.1 Detailed Description

count rate functions for Scienta detector images.

PearlScientaCountrate is declared in pearl-scienta-countrate.ipf.

10.15 PearlScientaPreprocess Namespace Reference

preprocessing functions for Scienta detector images.

10.15.1 Detailed Description

preprocessing functions for Scienta detector images.

PearlScientaPreprocess is declared in pearl-scienta-preprocess.ipf.

10.16 PearlVectorOperations Namespace Reference

basic vector geometry operations.

10.16.1 Detailed Description

basic vector geometry operations.

PearlVectorOperations is declared in pearl-vector-operations.ipf.

Chapter 11

Data Structure Documentation

11.1 DoniachSunjicStruct Struct Reference

Data Fields

- wave pw
- wave yw
- wave xw
- variable precision
- variable oversampling
- wave xdw
- wave model
- wave broadening
- wave convolution

11.1.1 Detailed Description

Definition at line 370 of file pearl-fitfuncs.ipf.

11.1.2 Field Documentation

11.1.2.1 broadening

```
wave DoniachSunjicStruct::broadening
```

Definition at line 386 of file pearl-fitfuncs.ipf.

11.1.2.2 convolution

```
wave DoniachSunjicStruct::convolution
```

Definition at line 387 of file pearl-fitfuncs.ipf.

11.1.2.3 model

`wave DoniachSunjicStruct::model`

Definition at line 385 of file pearl-fitfuncs.ipf.

11.1.2.4 oversampling

`variable DoniachSunjicStruct::oversampling`

Definition at line 380 of file pearl-fitfuncs.ipf.

11.1.2.5 precision

`variable DoniachSunjicStruct::precision`

Definition at line 379 of file pearl-fitfuncs.ipf.

11.1.2.6 pw

`wave DoniachSunjicStruct::pw`

Definition at line 370 of file pearl-fitfuncs.ipf.

11.1.2.7 xdw

`wave DoniachSunjicStruct::xdw`

Definition at line 384 of file pearl-fitfuncs.ipf.

11.1.2.8 xw

`wave DoniachSunjicStruct::xw`

Definition at line 376 of file pearl-fitfuncs.ipf.

11.1.2.9 yw

`wave DoniachSunjicStruct::yw`

Definition at line 375 of file pearl-fitfuncs.ipf.

The documentation for this struct was generated from the following file:

- pearl-fitfuncs.ipf

11.2 errorCode Struct Reference

from matrixfilereader help

Data Fields

- int32 SUCCESS
- int32 UNKNOWN_ERROR
- int32 ALREADY_FILE_OPEN
- int32 EMPTY_RESULTFILE
- int32 FILE_NOT_READABLE
- int32 NO_NEW_BRICKLETS
- int32 WRONG_PARAMETER
- int32 INTERNAL_ERROR_CONVERTING_DATA
- int32 NO_FILE_OPEN
- int32 INVALID_RANGE
- int32 WAVE_EXIST

11.2.1 Detailed Description

from matrixfilereader help

Definition at line 197 of file pearl-matrix-import.ipf.

11.2.2 Field Documentation

11.2.2.1 ALREADY_FILE_OPEN

```
int32 errorCode::ALREADY_FILE_OPEN
```

Definition at line 200 of file pearl-matrix-import.ipf.

11.2.2.2 EMPTY_RESULTFILE

```
int32 errorCode::EMPTY_RESULTFILE
```

Definition at line 201 of file pearl-matrix-import.ipf.

11.2.2.3 FILE_NOT_READABLE

```
int32 errorCode::FILE_NOT_READABLE
```

Definition at line 202 of file pearl-matrix-import.ipf.

11.2.2.4 INTERNAL_ERROR_CONVERTING_DATA

```
int32 errorCode::INTERNAL_ERROR_CONVERTING_DATA
```

Definition at line 205 of file pearl-matrix-import.ipf.

11.2.2.5 INVALID_RANGE

```
int32 errorCode::INVALID_RANGE
```

Definition at line 207 of file pearl-matrix-import.ipf.

11.2.2.6 NO_FILE_OPEN

```
int32 errorCode::NO_FILE_OPEN
```

Definition at line 206 of file pearl-matrix-import.ipf.

11.2.2.7 NO_NEW_BRICKLETS

```
int32 errorCode::NO_NEW_BRICKLETS
```

Definition at line 203 of file pearl-matrix-import.ipf.

11.2.2.8 SUCCESS

```
int32 errorCode::SUCCESS
```

Definition at line 197 of file pearl-matrix-import.ipf.

11.2.2.9 UNKNOWN_ERROR

```
int32 errorCode::UNKNOWN_ERROR
```

Definition at line 199 of file pearl-matrix-import.ipf.

11.2.2.10 WAVE_EXIST

```
int32 errorCode::WAVE_EXIST
```

Definition at line 208 of file pearl-matrix-import.ipf.

11.2.2.11 WRONG_PARAMETER

```
int32 errorCode::WRONG_PARAMETER
```

Definition at line 204 of file pearl-matrix-import.ipf.

The documentation for this struct was generated from the following file:

- pearl-matrix-import.ipf

Chapter 12

File Documentation

12.1 anglescan-processing.dox File Reference

12.2 fermi-edge-analysis.ipf File Reference

tools for analysing the Fermi edge measured by the Scienta EW4000 analyser.

```
#include "pearl-area-profiles"
```

Functions

- variable analyse_curved_edge (wave data)
- variable record_results (variable index)
- variable integrate_curved_edge (wave data, wave data_sig)
- variable slit_correction (wave data, wave data_out, variable epass)
- threadsafe variable FermiFuncLinDOS2D_corr (variable w, threadsafe x, wave y)
- threadsafe variable slit_shift (variable ypix, variable epass)
- variable show_shift (wave data)
- variable analyser_energy_resolution (variable epass, variable slit)

  calculate the energy resolution of the analyser

Variables

- static const variable mcp_radius_pix = 555

  MCP radius seen by the camera in pixels.
- static const variable mcp_radius_mm = 20

  physical size (radius) of the MCP in mm
- static const variable hemi_radius_mm = 200

  physical size (radius) of the hemisphere in mm
- static const variable mcp_radius_epass = 0.04

  energy range imaged on MCP relative to the pass energy

12.2.1 Detailed Description

tools for analysing the Fermi edge measured by the Scienta EW4000 analyser.

proposed procedure

- angular normalization
- fit curved fermi function
- calculate corrected energy coordinates and map to single independent variable
- fit normal fermi function

Author

matthias muntwiler, matthias.muntwiler@psi.ch
thomas dienel

Copyright

12.2.2 Function Documentation

12.2.2.1 analyse_curved_edge()

```
variable analyse_curved_edge (
            wave data )
```

Definition at line 27 of file fermi-edge-analysis.ipf.

12.2.2.2 analyser_energy_resolution()

```
variable analyser_energy_resolution (
            variable epass,
            variable slit )
```

calculate the energy resolution of the analyser

Parameters

epass	pass energy in eV
slit	analyser entrance slit in mm

Returns

energy resolution (FWHM)

Definition at line 306 of file fermi-edge-analysis.ipf.

12.2.2.3 FermiFuncLinDOS2D_corr()

```
threadsafe variable FermiFuncLinDOS2D_corr (
```

```
            variable w,
            threadsafe x,
            wave y )
```

Definition at line 216 of file fermi-edge-analysis.ipf.

12.2.2.4 integrate_curved_edge()

```
variable integrate_curved_edge (
            wave data,
            wave data_sig )
```

Definition at line 116 of file fermi-edge-analysis.ipf.

12.2.2.5 record_results()

```
variable record_results (
            variable index )
```

Definition at line 98 of file fermi-edge-analysis.ipf.

12.2.2.6 show_shift()

```
variable show_shift (
            wave data )
```

Definition at line 284 of file fermi-edge-analysis.ipf.

12.2.2.7 slit_correction()

```
variable slit_correction (
            wave data,
            wave data_out,
            variable epass )
```

Definition at line 173 of file fermi-edge-analysis.ipf.

12.2.2.8 slit_shift()

```
threadsafe variable slit_shift (
            variable ypix,
            variable epass )
```

Definition at line 267 of file fermi-edge-analysis.ipf.

12.2.3 Variable Documentation

12.2.3.1 hemi_radius_mm

```
const variable hemi_radius_mm = 200  [static]
```

physical size (radius) of the hemisphere in mm

Definition at line 263 of file fermi-edge-analysis.ipf.

12.2.3.2 mcp_radius_epass

```
const variable mcp_radius_epass = 0.04  [static]
```

energy range imaged on MCP relative to the pass energy

Definition at line 265 of file fermi-edge-analysis.ipf.

12.2.3.3 mcp_radius_mm

```
const variable mcp_radius_mm = 20  [static]
```

physical size (radius) of the MCP in mm

Definition at line 261 of file fermi-edge-analysis.ipf.

12.2.3.4 mcp_radius_pix

```
const variable mcp_radius_pix = 555  [static]
```

MCP radius seen by the camera in pixels.

Definition at line 259 of file fermi-edge-analysis.ipf.

12.3 mainpage.dox File Reference

12.4 pearl-anglescan-panel.ipf File Reference

interactive processing of angle scanned XPD data.

```
#include "pearl-anglescan-process"
#include "pearl-pmsco-import"
```

Namespaces

- PearlAnglescanPanel

  interactive processing of angle scanned XPD data.

Functions

- static variable AfterCompiledHook ()

  initialize package data once when the procedure is first loaded

- static variable init_package ()
- static variable save_prefs ()

  save persistent package data to the preferences file.
- static variable load_prefs ()

  load persistent package data from the preferences file.
- variable asp_import_raw (wave raw_data)

  import raw data
- variable asp_display_previews ()

  display a graph window of the processed data.
- variable asp_display_dist_check (variable xdist, variable ydist)

  display a graph window of the distribution checks.
- static variable do_init_process (variable check)

  initialize the process data with a copy of the raw data.
- static variable do_crop (variable check, variable force=defaultValue)

  crop the process data.
- static variable delete_rows (string rows, wave data, wave theta, wave tilt, wave phi)

  delete individual rows from the data strip
- static variable do_norm_alpha (variable check, variable force=defaultValue)

  alpha-normalize the process data.
- static variable do_norm_phi (variable check, variable force=defaultValue)

  phi-normalize the process data.
- static variable do_norm_theta (variable check, variable force=defaultValue)

  theta-normalize the process data.
- static variable do_norm_thetaphi (variable check, variable force=defaultValue)

  theta,phi-normalize the process data.
- variable asp_calculate_output ()

  calculate the output using all enabled processing filters.
- string asp_display_output (dfref data_df=defaultValue, string data_name=defaultValue)

  display the output diffractogram
- variable asp_update_graph ()

  update graphs with new color table or contrast
- variable asp_close_graphs ()

  close all graphs created by the angle scan panel
- string asp_duplicate_output (string dest_name, variable do_graph=defaultValue)

  copy the output data to a new folder
- variable asp_save_output_itx ()

  save the output diffractogram to an igor text file
- variable asp_save_output_etpi (variable ekin)

  save the output diffractogram to a PMSCO ETPI file
- static variable check_norm_alpha ()
- static variable check_norm_phi ()
- static variable check_norm_theta ()
- static variable check_norm_thetaphi ()
- static variable preview_crop ()
- static variable preview_norm_alpha ()
- static variable preview_norm_phi ()
- static variable preview_norm_theta ()
- static variable preview_norm_thetaphi ()
- variable asp_show_panel ()

  create the angle scan processing panel
- static variable arrange_controls ()

- static variable update_menus ()

  update the popup menus to reflect the values of the global variables
- static variable bp_load_prefs (WMButtonAction *ba)
- static variable bp_save_prefs (WMButtonAction *ba)
- static variable bp_source_select (WMButtonAction *ba)
- static variable bp_source_update (WMButtonAction *ba)
- static variable bp_norm_alpha_check (WMButtonAction *ba)
- static variable bp_norm_theta_check (WMButtonAction *ba)
- static variable bp_norm_phi_check (WMButtonAction *ba)
- static variable bp_norm_thetaphi_check (WMButtonAction *ba)
- static variable bp_crop_preview (WMButtonAction *ba)
- static variable bp_norm_alpha_preview (WMButtonAction *ba)
- static variable bp_norm_phi_preview (WMButtonAction *ba)
- static variable bp_norm_theta_preview (WMButtonAction *ba)
- static variable bp_norm_thetaphi_preview (WMButtonAction *ba)
- static variable bp_output_calc (WMButtonAction *ba)
- static variable bp_output_duplicate (WMButtonAction *ba)
- static variable bp_output_etpi (WMButtonAction *ba)
- static variable bp_output_itx (WMButtonAction *ba)
- static variable bp_graph_update (WMButtonAction *ba)
- static variable bp_graph_png (WMButtonAction *ba)
- static variable pmp_norm_alpha_mode (WMPopupAction *pa)
- static variable pmp_norm_theta_domain (WMPopupAction *pa)
- static variable pmp_norm_theta_mode (WMPopupAction *pa)
- static variable pmp_norm_thetaphi_mode (WMPopupAction *pa)
- static variable pmp_graph_mode (WMPopupAction *pa)
- static variable pmp_graph_projection (WMPopupAction *pa)
- static variable pmp_graph_colortable (WMPopupAction *pa)

Variables

- static const string package_name = "pearl_anglescan_panel"

  package name is used as data folder name
- static const string package_path = "root:packages:pearl_anglescan_panel:"

  data folder path

12.4.1 Detailed Description

interactive processing of angle scanned XPD data.

steps to process a hemispherical scan into a diffractogram:

1. load the scan file in data reduction mode.
2. open the angle scan panel by calling asp_show_panel().
3. select the dataset.
4. set the processing parameters.
5. start the processing and display the result.
6. refine parameters and repeat steps 4-5.
7. export the result in the desired form.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.4.2 Function Documentation

12.4.2.1 AfterCompiledHook()

```
static variable AfterCompiledHook ( )  [static]
```

initialize package data once when the procedure is first loaded

Definition at line 50 of file pearl-anglescan-panel.ipf.

12.4.2.2 arrange_controls()

```
static variable arrange_controls ( )  [static]
```

Definition at line 1052 of file pearl-anglescan-panel.ipf.

12.4.2.3 asp_calculate_output()

```
variable asp_calculate_output ( )
```

calculate the output using all enabled processing filters.

the diffractogram is calculated, to display the graph, call asp_display_output.

Definition at line 620 of file pearl-anglescan-panel.ipf.

12.4.2.4 asp_close_graphs()

```
variable asp_close_graphs ( )
```

close all graphs created by the angle scan panel

Definition at line 728 of file pearl-anglescan-panel.ipf.

12.4.2.5 asp_display_dist_check()

```
variable asp_display_dist_check (
            variable xdist,
            variable ydist )
```

display a graph window of the distribution checks.

if the window exists, it is updated and brought to the front of the window stack.

Parameters

xdist	1: display the x-distribution; 0: don't.
ydist	1: display the y-distribution; 0: don't.

Definition at line 273 of file pearl-anglescan-panel.ipf.

12.4.2.6 asp_display_output()

```
string asp_display_output (
             dfref data_df = defaultValue,
             string data_name = defaultValue )
```

display the output diffractogram

this function either displays the current working data from the angle scan panel data folder, or hologram data from a specified other data folder. in either case, the graph is displayed using the current projection, mode, color table and contrast settings of the angle scan panel.

Parameters

data_df	data folder where the diffractogram data is located. default: the folder of the angle scan panel package.
data_name	name of the diffractogram data. this is normally the name of a sub-folder that contains the data. default: the name specified in the output_name variable in the package data folder.

Definition at line 666 of file pearl-anglescan-panel.ipf.

12.4.2.7 asp_display_previews()

```
variable asp_display_previews ( )
```

display a graph window of the processed data.

if the window exists, it is updated and brought to the front of the window stack.

Definition at line 249 of file pearl-anglescan-panel.ipf.

12.4.2.8 asp_duplicate_output()

```
string asp_duplicate_output (
             string dest_name,
             variable do_graph = defaultValue )
```

copy the output data to a new folder

Parameters

dest_name	name of destination folder. can contain a relative (starting with a colon) or absolute (starting with "root:") path. folders in the path must exist except for the last one. the destination folder does not need to exist. existing data in the destination folder is overwritten.
do_graph	switch to duplicate the graph window as well (1). default: 0 (do not create a graph)

Returns

name of the graph window

Definition at line 768 of file pearl-anglescan-panel.ipf.

12.4.2.9 asp_import_raw()

```
variable asp_import_raw (
             wave raw_data )
```

import raw data

the raw data are copied into the package data folder, and the process waves are initialized.

Parameters

raw_data	2D intensity distribution. the x dimension is the detection angle. the y dimension is an arbitrary sequence of manipulator scan positions. ManipulatorTheta, ManipulatorPilt and ManipulatorPhi waves that indicate the manipulator angles for each y position must be present in the :attr sub-folder next to the raw data wave.

Definition at line 204 of file pearl-anglescan-panel.ipf.

12.4.2.10 asp_save_output_etpi()

```
variable asp_save_output_etpi (
             variable ekin )
```

save the output diffractogram to a PMSCO ETPI file

Parameters

ekin	kinetic energy in eV

Definition at line 816 of file pearl-anglescan-panel.ipf.

12.4.2.11 asp_save_output_itx()

```
variable asp_save_output_itx ( )
```

save the output diffractogram to an igor text file

Definition at line 801 of file pearl-anglescan-panel.ipf.

12.4.2.12 asp_show_panel()

```
variable asp_show_panel ( )
```

create the angle scan processing panel

Definition at line 900 of file pearl-anglescan-panel.ipf.

12.4.2.13 asp_update_graph()

```
variable asp_update_graph ( )
```

update graphs with new color table or contrast

applies to the preview graph and the diffractogram.

Definition at line 710 of file pearl-anglescan-panel.ipf.

12.4.2.14 bp_crop_preview()

```
static variable bp_crop_preview (
             WMButtonAction * ba )  [static]
```

Definition at line 1370 of file pearl-anglescan-panel.ipf.

12.4.2.15 bp_graph_png()

```
static variable bp_graph_png (
             WMButtonAction * ba )  [static]
```

Definition at line 1526 of file pearl-anglescan-panel.ipf.

12.4.2.16 bp_graph_update()

```
static variable bp_graph_update (
             WMButtonAction * ba )  [static]
```

Definition at line 1512 of file pearl-anglescan-panel.ipf.

12.4.2.17 bp_load_prefs()

```
static variable bp_load_prefs (
             WMButtonAction * ba )  [static]
```

Definition at line 1239 of file pearl-anglescan-panel.ipf.

12.4.2.18 bp_norm_alpha_check()

```
static variable bp_norm_alpha_check (
            WMButtonAction * ba )  [static]
```

Definition at line 1314 of file pearl-anglescan-panel.ipf.

12.4.2.19 bp_norm_alpha_preview()

```
static variable bp_norm_alpha_preview (
            WMButtonAction * ba )  [static]
```

Definition at line 1384 of file pearl-anglescan-panel.ipf.

12.4.2.20 bp_norm_phi_check()

```
static variable bp_norm_phi_check (
            WMButtonAction * ba )  [static]
```

Definition at line 1342 of file pearl-anglescan-panel.ipf.

12.4.2.21 bp_norm_phi_preview()

```
static variable bp_norm_phi_preview (
            WMButtonAction * ba )  [static]
```

Definition at line 1398 of file pearl-anglescan-panel.ipf.

12.4.2.22 bp_norm_theta_check()

```
static variable bp_norm_theta_check (
            WMButtonAction * ba )  [static]
```

Definition at line 1328 of file pearl-anglescan-panel.ipf.

12.4.2.23 bp_norm_theta_preview()

```
static variable bp_norm_theta_preview (
            WMButtonAction * ba )  [static]
```

Definition at line 1412 of file pearl-anglescan-panel.ipf.

12.4.2.24 bp_norm_thetaphi_check()

```
static variable bp_norm_thetaphi_check (
            WMButtonAction * ba )  [static]
```

Definition at line 1356 of file pearl-anglescan-panel.ipf.

12.4.2.25 bp_norm_thetaphi_preview()

```
static variable bp_norm_thetaphi_preview (
            WMButtonAction * ba )  [static]
```

Definition at line 1426 of file pearl-anglescan-panel.ipf.

12.4.2.26 bp_output_calc()

```
static variable bp_output_calc (
            WMButtonAction * ba )  [static]
```

Definition at line 1440 of file pearl-anglescan-panel.ipf.

12.4.2.27 bp_output_duplicate()

```
static variable bp_output_duplicate (
            WMButtonAction * ba )  [static]
```

Definition at line 1455 of file pearl-anglescan-panel.ipf.

12.4.2.28 bp_output_etpi()

```
static variable bp_output_etpi (
            WMButtonAction * ba )  [static]
```

Definition at line 1476 of file pearl-anglescan-panel.ipf.

12.4.2.29 bp_output_itx()

```
static variable bp_output_itx (
            WMButtonAction * ba )  [static]
```

Definition at line 1498 of file pearl-anglescan-panel.ipf.

12.4.2.30 bp_save_prefs()

```
static variable bp_save_prefs (
            WMButtonAction * ba )  [static]
```

Definition at line 1253 of file pearl-anglescan-panel.ipf.

12.4.2.31 bp_source_select()

```
static variable bp_source_select (
             WMButtonAction * ba )  [static]
```

Definition at line 1267 of file pearl-anglescan-panel.ipf.

12.4.2.32 bp_source_update()

```
static variable bp_source_update (
             WMButtonAction * ba )  [static]
```

Definition at line 1293 of file pearl-anglescan-panel.ipf.

12.4.2.33 check_norm_alpha()

```
static variable check_norm_alpha ( )  [static]
```

Definition at line 836 of file pearl-anglescan-panel.ipf.

12.4.2.34 check_norm_phi()

```
static variable check_norm_phi ( )  [static]
```

Definition at line 843 of file pearl-anglescan-panel.ipf.

12.4.2.35 check_norm_theta()

```
static variable check_norm_theta ( )  [static]
```

Definition at line 850 of file pearl-anglescan-panel.ipf.

12.4.2.36 check_norm_thetaphi()

```
static variable check_norm_thetaphi ( )  [static]
```

Definition at line 857 of file pearl-anglescan-panel.ipf.

12.4.2.37 delete_rows()

```
static variable delete_rows (
             string rows,
             wave data,
             wave theta,
             wave tilt,
             wave phi )  [static]
```

delete individual rows from the data strip

Parameters

in	*rows*	comma-separated list of row indices or ranges. ranges are specified by joining the start and end index with a hyphen, e.g. 24-46. the list does not need to be ordered.
in,out	*data*	2D data the original wave is modified.
in,out	*theta*	theta positions along Y dimension of data. the original wave is modified.
in,out	*tilt*	tilt positions along Y dimension of data the original wave is modified.
in,out	*phi*	phi positions along Y dimension of data the original wave is modified.

Definition at line 406 of file pearl-anglescan-panel.ipf.

12.4.2.38 do_crop()

```
static variable do_crop (
            variable check,
            variable force = defaultValue )  [static]
```

crop the process data.

Parameters

in	*check*	select which output to generate (currently not used). • 0 (default) process data, don't calculate check waves. • 1 process data and calculate check waves. • 2 calculate check waves only, do not calculate process data.
in	*force*	override the global enable flag. • 0 (default) apply this filter only if its global enable flag is set. • 1 apply this filter even if it is not enabled.

Definition at line 358 of file pearl-anglescan-panel.ipf.

12.4.2.39 do_init_process()

```
static variable do_init_process (
            variable check )  [static]
```

initialize the process data with a copy of the raw data.

applies the angle offsets.

Parameters

in	*check*	select which output to generate (currently not used). • 0 (default) process data, don't calculate check waves. • 1 process data and calculate check waves. • 2 calculate check waves only, do not calculate process data.

Definition at line 318 of file pearl-anglescan-panel.ipf.

12.4.2.40 do_norm_alpha()

```
static variable do_norm_alpha (
             variable check,
             variable force = defaultValue )  [static]
```

alpha-normalize the process data.

Parameters

in	*check*	select which output to generate. • 0 (default) process data, don't calculate check waves. • 1 process data and calculate check waves. • 2 calculate check waves only, do not calculate process data.
in	*force*	override the global enable flag. • 0 (default) apply this filter only if its global enable flag is set. • 1 apply this filter even if it is not enabled.

Definition at line 453 of file pearl-anglescan-panel.ipf.

12.4.2.41 do_norm_phi()

```
static variable do_norm_phi (
             variable check,
             variable force = defaultValue )  [static]
```

phi-normalize the process data.

Parameters

in	*check*	select which output to generate. • 0 (default) process data, don't calculate check waves. • 1 process data and calculate check waves. • 2 calculate check waves only, do not calculate process data.
in	*force*	override the global enable flag. • 0 (default) apply this filter only if its global enable flag is set. • 1 apply this filter even if it is not enabled.

Definition at line 494 of file pearl-anglescan-panel.ipf.

12.4.2.42 do_norm_theta()

```
static variable do_norm_theta (
            variable check,
            variable force = defaultValue )  [static]
```

theta-normalize the process data.

Parameters

in	*check*	select which output to generate. • 0 (default) process data, don't calculate check waves. • 1 process data and calculate check waves. • 2 calculate check waves only, do not calculate process data.
in	*force*	override the global enable flag. • 0 (default) apply this filter only if its global enable flag is set. • 1 apply this filter even if it is not enabled.

Definition at line 537 of file pearl-anglescan-panel.ipf.

12.4.2.43 do_norm_thetaphi()

```
static variable do_norm_thetaphi (
            variable check,
            variable force = defaultValue )  [static]
```

theta,phi-normalize the process data.

Parameters

in	*check*	select which output to generate. • 0 (default) process data, don't calculate check waves. • 1 process data and calculate check waves. • 2 calculate check waves only, do not calculate process data.
in	*force*	override the global enable flag. • 0 (default) apply this filter only if its global enable flag is set. • 1 apply this filter even if it is not enabled.

Definition at line 584 of file pearl-anglescan-panel.ipf.

12.4.2.44 init_package()

```
static variable init_package ( )  [static]
```

Definition at line 75 of file pearl-anglescan-panel.ipf.

12.4.2.45 load_prefs()

```
static variable load_prefs ( )  [static]
```

load persistent package data from the preferences file.

the preferences file is an Igor packed experiment file in a special preferences folder.

this function is called automatically when the procedure is first compiled, or whenever the user clicks the corresponding button.

Definition at line 168 of file pearl-anglescan-panel.ipf.

12.4.2.46 pmp_graph_colortable()

```
static variable pmp_graph_colortable (
             WMPopupAction * pa )  [static]
```

Definition at line 1641 of file pearl-anglescan-panel.ipf.

12.4.2.47 pmp_graph_mode()

```
static variable pmp_graph_mode (
             WMPopupAction * pa )  [static]
```

Definition at line 1609 of file pearl-anglescan-panel.ipf.

12.4.2.48 pmp_graph_projection()

```
static variable pmp_graph_projection (
             WMPopupAction * pa )  [static]
```

Definition at line 1625 of file pearl-anglescan-panel.ipf.

12.4.2.49 pmp_norm_alpha_mode()

```
static variable pmp_norm_alpha_mode (
             WMPopupAction * pa )  [static]
```

Definition at line 1545 of file pearl-anglescan-panel.ipf.

12.4.2.50 pmp_norm_theta_domain()

```
static variable pmp_norm_theta_domain (
             WMPopupAction * pa )  [static]
```

Definition at line 1561 of file pearl-anglescan-panel.ipf.

12.4.2.51 pmp_norm_theta_mode()

```
static variable pmp_norm_theta_mode (
            WMPopupAction * pa )  [static]
```

Definition at line 1577 of file pearl-anglescan-panel.ipf.

12.4.2.52 pmp_norm_thetaphi_mode()

```
static variable pmp_norm_thetaphi_mode (
            WMPopupAction * pa )  [static]
```

Definition at line 1593 of file pearl-anglescan-panel.ipf.

12.4.2.53 preview_crop()

```
static variable preview_crop ( )  [static]
```

Definition at line 864 of file pearl-anglescan-panel.ipf.

12.4.2.54 preview_norm_alpha()

```
static variable preview_norm_alpha ( )  [static]
```

Definition at line 870 of file pearl-anglescan-panel.ipf.

12.4.2.55 preview_norm_phi()

```
static variable preview_norm_phi ( )  [static]
```

Definition at line 877 of file pearl-anglescan-panel.ipf.

12.4.2.56 preview_norm_theta()

```
static variable preview_norm_theta ( )  [static]
```

Definition at line 884 of file pearl-anglescan-panel.ipf.

12.4.2.57 preview_norm_thetaphi()

```
static variable preview_norm_thetaphi ( )  [static]
```

Definition at line 891 of file pearl-anglescan-panel.ipf.

12.4.2.58 save_prefs()

```
static variable save_prefs ( )  [static]
```

save persistent package data to the preferences file.

this function is called when the user clicks the corresponding button.

Definition at line 142 of file pearl-anglescan-panel.ipf.

12.4.2.59 update_menus()

```
static variable update_menus ( )  [static]
```

update the popup menus to reflect the values of the global variables

Definition at line 1210 of file pearl-anglescan-panel.ipf.

12.4.3 Variable Documentation

12.4.3.1 package_name

```
const string package_name = "pearl_anglescan_panel"  [static]
```

package name is used as data folder name

Definition at line 45 of file pearl-anglescan-panel.ipf.

12.4.3.2 package_path

```
const string package_path = "root:packages:pearl_anglescan_panel:"  [static]
```

data folder path

Definition at line 47 of file pearl-anglescan-panel.ipf.

12.5 pearl-anglescan-process.ipf File Reference

processing and holographic mapping of angle scanned XPD data.

```
#include "pearl-vector-operations"
#include "pearl-polar-coordinates"
#include <New Polar Graphs>
```

Namespaces

- PearlAnglescanProcess

  processing and holographic mapping of angle scanned XPD data.

Functions

- variable strip_delete_frames (wave strip, variable qlo, variable qhi, wave theta, wave tilt, wave phi)

  delete a contiguous range of frames from a strip.
- variable normalize_strip_x (wave strip, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue)

  divide the strip by the average X distribution.
- variable normalize_strip_phi (wave strip, wave theta, wave phi, variable theta_offset=defaultValue, variable theta_range=defaultValue, variable check=defaultValue)

  divide the strip by a sine function in phi (wobble correction).
- variable normalize_strip_theta (wave strip, wave theta, variable theta_offset=defaultValue, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue)

  divide the strip by the average polar distribution.
- variable normalize_strip_thetaphi (wave strip, wave theta, wave phi, variable theta_offset=defaultValue, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue)

  divide the strip by a smooth polar-azimuthal distribution.
- variable normalize_strip_theta_scans (wave strip, wave theta, variable theta_offset=defaultValue, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue)

  divide the strip piecewise by a smooth polar distribution.
- variable normalize_strip_2d (wave strip, wave theta, variable theta_offset=defaultValue, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue)

  divide the strip by a two-dimensional normalization function.
- variable crop_strip (wave strip, variable xlo, variable xhi)

  crop a strip at the sides.
- variable crop_strip_theta (wave strip, variable theta_lo, variable theta_hi, wave theta, wave tilt, wave phi)

  crop a strip in theta.
- variable pizza_service (wave data, string nickname, variable theta_offset, variable tilt_offset, variable phi_offset, variable npolar=defaultValue, variable nograph=defaultValue, variable folding=defaultValue, variable xpdplot=defaultValue)

  create a pizza plot from a measured (energy-integrated) data strip
- variable pizza_service_2 (wave data, string nickname, wave m_theta, wave m_tilt, wave m_phi, variable npolar=defaultValue, variable nograph=defaultValue, variable folding=defaultValue, variable xpdplot=defaultValue)

  create a pizza plot from a measured (energy-integrated) data strip
- variable show_analyser_line (variable theta, variable tilt, variable phi, variable theta_offset, variable tilt_offset, variable phi_offset, variable npolar=defaultValue, variable nograph=defaultValue, variable xpdplot=defaultValue)

  calculate and display the line seen by the analyser for a specific emission angle
- variable convert_angles_ttpd2polar (wave theta, wave tilt, wave phi, wave data, wave polar, wave azi)

  convert angles from TTPA (theta-tilt-phi-analyser) scheme to polar coordinates.
- variable convert_angles_ttpa2polar (wave theta, wave tilt, wave phi, wave analyser, wave polar, wave azi)

  convert angles from TTPA (theta-tilt-phi-analyser) scheme to polar coordinates.
- static variable line_average (wave source, wave dest)
- static variable calc_nth (variable Theta_st, variable Theta_in, variable th, variable Phi_ran, variable Phi_ref, string Holomode)

  calculate the number of phis for a given theta
- static variable calc_phi_step (variable Theta_in, variable th, variable Theta_st, variable Phi_ran, variable Phi_ref, string Holomode)

  calculate delta-phi for a given theta
- static variable Calc_The_step (variable th, variable Theta_st, string Holomode)

  calculate delta-theta for a given theta
- static variable CalcN_Theta (string HoloMode, variable Theta_in, variable Theta_ran, variable Theta_st)

  calculate the number of thetas for a pattern

- variable make_hemi_grid (variable npol, string nickname, variable xpdplot=defaultValue)

  create a hemispherical, constant solid angle grid
- string get_hemi_nickname (wave w)

  finds the nick name given any hemi wave
- string get_hemi_prefix (wave w)

  finds the prefix given any hemi wave
- dfr find_hemi_data (string nickname, string *prefix, string *intwave)

  finds the folder, prefix and name of holo waves given their nick name
- variable clear_hemi_grid (string nickname)

  clear a hemispherical scan grid
- variable duplicate_hemi_scan (string source_nickname, dfref dest_folder, string dest_nickname, variable xpdplot=defaultValue)

  duplicate a hemispherical scan dataset.
- variable rotate_hemi_scan (string nickname, variable angle)

  azimuthally rotate a hemispherical scan dataset.
- string prepare_hemi_scan_display (string nickname, variable projection=defaultValue)

  create waves for plotting a hemispherical angle scan.
- string display_hemi_scan (string nickname, variable projection=defaultValue, variable graphtype=default↩ Value, variable do_ticks=defaultValue, variable do_grids=defaultValue, string graphname=defaultValue)

  display a plot of a hemispherical angle scan.
- static string display_polar_graph (string graphname, variable angle_offset=defaultValue, variable do_↩ ticks=defaultValue)

  displays an empty polar graph
- static string draw_hemi_axes (string graphname, variable do_grids=defaultValue)

  draw polar and azimuthal grids in an existing polar graph.
- variable draw_diffraction_cone (string graphname, string groupname, variable theta_axis, variable theta_↩ inner, variable phi)

  draw the circle of a diffraction cone in a stereographic polar graph.
- string display_scanlines (string nickname, variable alpha_lo, variable alpha_hi, wave m_theta, wave m_tilt, wave m_phi, variable folding=defaultValue, variable projection=defaultValue)

  display a polar graph with lines indicating the angles covered by an angle scan.
- threadsafe variable calc_graph_radius (variable polar, variable projection=defaultValue)

  calculate the projected polar angle
- threadsafe variable calc_graph_polar (variable x, variable y, variable projection=defaultValue)

  calculate polar angle from Cartesian coordinate
- threadsafe variable calc_graph_azi (variable x, variable y, variable projection=defaultValue, variable zero↩ Angle=defaultValue)

  calculate azimuthal angle from Cartesian coordinate
- static variable update_polar_info (string graphname)

  update the angles info based on cursors A and B of a given polar graph window
- static variable polar_graph_hook (WMWinHookStruct *s)

  polar graph window hook
- variable set_polar_graph_cursor (string nickname, string cursorname, variable polar_angle, variable azim_↩ angle, string graphname=defaultValue)
- variable hemi_add_anglescan (string nickname, wave values, wave polar, wave azi, wave weights=default↩ Value)

  add an arbitrary angle scan to a hemispherical scan grid.
- variable hemi_add_aziscan (string nickname, wave values, variable polar, wave azi, wave weights=default↩ Value)

  add an azimuthal scan to a hemispherical scan grid.
- variable interpolate_hemi_scan (string nickname, variable projection=defaultValue)

  interpolate a hemispherical scan onto a rectangular grid

- variable quick_pizza_image (wave data, string nickname, variable theta_offset, variable tilt_offset, variable phi_offset, variable npolar=defaultValue, variable nograph=defaultValue, variable folding=defaultValue)

  map angle scan data onto a rectangular grid in stereographic projection
- variable save_hemi_scan (string nickname, string pathname, string filename)

  save a hemispherical scan to an Igor text file
- variable load_hemi_scan (string nickname, string pathname, string filename)

  load a hemispherical scan from an Igor text file
- variable import_tpi_scan (string nickname, wave theta, wave phi, wave intensity, variable folding=defaultValue, variable npolar=defaultValue, variable nograph=defaultValue, variable xpdplot=defaultValue)

  import a hemispherical scan from theta-phi-intensity waves and display it
- variable trim_hemi_scan (string nickname, variable theta_max)

  trim a hemispherical scan at grazing angle
- wave hemi_polar_cut (string nickname, variable azim)

  extract a polar cut from a hemispherical scan.
- wave hemi_azi_cut (string nickname, variable pol)

  extract an azimuthal cut from a hemispherical scan
- static variable check_contrast (wave values, variable pcmin, variable pcmax, variable *vmin, variable *vmax)
- variable set_contrast (variable pcmin, variable pcmax, string graphname=defaultValue, string colortable=default↩ Value)

  set the pseudocolor contrast by percentile.
- variable AngleToK (wave inwave)

  k-space mapping of 2D angle-energy distribution (scienta image)

Variables

- const variable kProjDist = 0
- const variable kProjStereo = 1
- const variable kProjArea = 2
- const variable kProjGnom = 3
- const variable kProjOrtho = 4
- static const variable kProjScaleDist = 2
- static const variable kProjScaleStereo = 2
- static const variable kProjScaleArea = 2
- static const variable kProjScaleGnom = 0.06744519021
- static const variable kProjScaleOrtho = 2

12.5.1 Detailed Description

processing and holographic mapping of angle scanned XPD data.

the functions in this file map angle scanned data measured at PEARL onto a hemispherical angle grid which is compatible with XPDplot. the resulting data are in a canonical polar coordinate system (normal emission <=> polar angle = 0, azimuthal axis right-handed) which is anchored in the sample surface. the orientation of polar graphs (phi = 0 at 3 o'clock) created by this procedure corresponds to a top view of the sample surface at normal emission, and the handle of the sample plate pointing to the left (phi = 180). this is the canonical orientation of a spherical coordinate system where phi = 0 corresponds to the positive part of the x axis.

Note

the orientation of the sample coordinate system has changed in version 1.6. the change was necessary for compatibility with other data analysis software and calculation programs.
data imported with version 1.5 and earlier, must be offset by 180 deg in phi to be compatible with the new version. data imported and displayed by the same code version will give the same picture but with different azimuthal axis. the new graph functions shows a warning if they are applied to code imported with earlier versions.

the measurement geometry is hard-coded but may be parametrized in the future. the theta rotation axis is perpendicular to the scattering plane. the angle dispersive axis of the analyser is parallel to the theta rotation axis. the tilt rotation axis is in the scattering plane. it rotates with theta. at normal emission it is perpendicular to the axis of the lens stack of the analyser. the phi rotation axis corresponds to the surface normal of the sample. it rotates with theta and with tilt. at normal emission it is parallel to the axis of the lens stack of the analyser.

coordinate transformations: (to be revised - v1.6)

```
*       theta_sample = theta_manipulator - theta_offset
*       phi_sample = phi_manipulator - phi_offset
*
```

valid for theta_manipulator = normal emission only: (to be revised - v1.6)

```
*       theta_sample = | -(tilt_manipulator - tilt_offset) |
*       phi_sample = 270 if tilt_manipulator - tilt_offset > 0
*       phi_sample = 90 if tilt_manipulator - tilt_offset < 0
*
```

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

Version

1.8 canonical orientation of spherical coordinate system.

12.5.2 Function Documentation

12.5.2.1 AngleToK()

```
variable AngleToK (
             wave inwave )
```

k-space mapping of 2D angle-energy distribution (scienta image)

courtesy of F. Matsui

Parameters

inwave	2D wave, x = kinetic energy (eV), y = polar angle (deg) note: the kinetic energy is with reference to the vacuum level at the sample. if the work functions of the analyser and the sample differ: Ekin,sample = Ekin,analyser + WFanalyser - WFsample where WFanalyser = Ephot - EFermi

Returns

the output wave has the name of the input wave with the suffix "_k".

Definition at line 3332 of file pearl-anglescan-process.ipf.

12.5.2.2 calc_graph_azi()

```
threadsafe variable calc_graph_azi (
             variable x,
             variable y,
             variable projection = defaultValue,
             variable zeroAngle = defaultValue )
```

calculate azimuthal angle from Cartesian coordinate

Parameters

x,y	projected Cartesian coordinate
projection	mapping function from polar to cartesian coordinates. all supported projections are azimuthal, they have no effect on the azimuthal coordinate. see Projections for details. • kProjDist = 0 azimuthal equidistant • kProjStereo = 1 stereographic (default) • kProjArea = 2 azimuthal equal-area • kProjGnom = 3 gnomonic (0 <= polar < 90) • kProjOrtho = 4 orthographic
zeroAngle	zeroAngleWhere parameter of polar graphs • 0 (default) zero is at the 3 o'clock position • 180 zero is at the 9 o'clock position • other values not tested

Returns

polar angle in degrees

Definition at line 2393 of file pearl-anglescan-process.ipf.

12.5.2.3 calc_graph_polar()

```
threadsafe variable calc_graph_polar (
```

```
            variable x,
            variable y,
            variable projection = defaultValue )
```

calculate polar angle from Cartesian coordinate

this is the reverse mapping to calc_graph_radius()

Parameters

x,y	projected Cartesian coordinate
projection	mapping function from polar to cartesian coordinates. see Projections for details. • kProjDist = 0 azimuthal equidistant • kProjStereo = 1 stereographic (default) • kProjArea = 2 azimuthal equal-area • kProjGnom = 3 gnomonic (0 <= polar < 90) • kProjOrtho = 4 orthographic

Returns

polar angle in degrees

Definition at line 2340 of file pearl-anglescan-process.ipf.

12.5.2.4 calc_graph_radius()

```
threadsafe variable calc_graph_radius (
            variable polar,
            variable projection = defaultValue )
```

calculate the projected polar angle

Parameters

polar	polar angle in degrees
projection	mapping function from polar to cartesian coordinates. see Projections for details. • kProjDist = 0 azimuthal equidistant • kProjStereo = 1 stereographic (default) • kProjArea = 2 azimuthal equal-area • kProjGnom = 3 gnomonic (0 <= polar < 90) • kProjOrtho = 4 orthographic

Returns

projected radius. the radius is scaled such that grazing emission maps to 2.

Definition at line 2295 of file pearl-anglescan-process.ipf.

12.5.2.5 calc_nth()

```
static variable calc_nth (
             variable Theta_st,
             variable Theta_in,
             variable th,
             variable Phi_ran,
             variable Phi_ref,
             string Holomode )  [static]
```

calculate the number of phis for a given theta

adapted from XPDplot 8.03

Definition at line 1149 of file pearl-anglescan-process.ipf.

12.5.2.6 calc_phi_step()

```
static variable calc_phi_step (
             variable Theta_in,
             variable th,
             variable Theta_st,
             variable Phi_ran,
             variable Phi_ref,
             string Holomode )  [static]
```

calculate delta-phi for a given theta

adapted from XPDplot 8.03

Definition at line 1178 of file pearl-anglescan-process.ipf.

12.5.2.7 Calc_The_step()

```
static variable Calc_The_step (
             variable th,
             variable Theta_st,
             string Holomode )  [static]
```

calculate delta-theta for a given theta

adapted from XPDplot 8.03

Definition at line 1222 of file pearl-anglescan-process.ipf.

12.5.2.8 CalcN_Theta()

```
static variable CalcN_Theta (
             string HoloMode,
             variable Theta_in,
             variable Theta_ran,
             variable Theta_st )  [static]
```

calculate the number of thetas for a pattern

adapted from XPDplot 8.03

Definition at line 1253 of file pearl-anglescan-process.ipf.

12.5.2.9 check_contrast()

```
static variable check_contrast (
             wave values,
             variable pcmin,
             variable pcmax,
             variable * vmin,
             variable * vmax )  [static]
```

Definition at line 3212 of file pearl-anglescan-process.ipf.

12.5.2.10 clear_hemi_grid()

```
variable clear_hemi_grid (
             string nickname )
```

clear a hemispherical scan grid

values and weights waves are set to zero. the intensity wave is set to NaN.

Parameters

nickname	folder name or name prefix of holo waves. may be empty.

Definition at line 1532 of file pearl-anglescan-process.ipf.

12.5.2.11 convert_angles_ttpa2polar()

```
variable convert_angles_ttpa2polar (
             wave theta,
             wave tilt,
             wave phi,
             wave analyser,
             wave polar,
             wave azi )
```

convert angles from TTPA (theta-tilt-phi-analyser) scheme to polar coordinates.

the angles are in the manipulator coordinate system.

Parameters

in	*theta*	offset-corrected theta angle, normal emission = 0, grazing emission = 90. one dimensional wave.
in	*tilt*	offset-corrected tilt angle, normal emission = 0 same dimension size and scale as theta
in	*phi*	phi angle, range -360 $<$ phi $<$ +360 offset correction is optional as long as the angles lie in the accepted range. same dimension size and scale as theta
in	*analyser*	analyser angle scale corresponding to the slices scale of Scienta. one dimensional wave. this values are constant regardless of manipulator angle.
out	*polar*	wave to receive the polar coordinates.
out	*azi*	wave to receive the azimuthal coordinates.

for the output parameters polar and azi, you need to pass in existing numeric waves. dimension size does not matter, the waves are redimensioned by the function so that they have the same dimensions as the intensity data set. X dimension = analyser scale, Y dimension = manipulator scan.

Definition at line 1088 of file pearl-anglescan-process.ipf.

12.5.2.12 convert_angles_ttpd2polar()

```
variable convert_angles_ttpd2polar (
             wave theta,
             wave tilt,
             wave phi,
             wave data,
             wave polar,
             wave azi )
```

convert angles from TTPA (theta-tilt-phi-analyser) scheme to polar coordinates.

similar to convert_angles_ttpa2polar() but reads the analyser angles from the X scale of data

Definition at line 1049 of file pearl-anglescan-process.ipf.

12.5.2.13 crop_strip()

```
variable crop_strip (
             wave strip,
             variable xlo,
             variable xhi )
```

crop a strip at the sides.

the strip is cropped in place, data outside the region of interest is lost.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan
in	*xlo*	lowest analyser angle to keep (will be rounded to nearest existing point)
in	*xhi*	highest analyser angle to keep (will be rounded to nearest existing point)

Remarks

cropping should be done after smoothing and normalization operations to reduce artefacts.

Definition at line 673 of file pearl-anglescan-process.ipf.

12.5.2.14 crop_strip_theta()

```
variable crop_strip_theta (
             wave strip,
             variable theta_lo,
             variable theta_hi,
             wave theta,
             wave tilt,
             wave phi )
```

crop a strip in theta.

the strip is cropped in place, data outside the region of interest is lost.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan
in	*ylo*	lowest polar angle to keep (will be rounded to nearest existing point)
in	*yhi*	highest polar angle to keep (will be rounded to nearest existing point)
in,out	*theta*	polar angle along the Y dimension of strip. this wave is modified: cropped rows are deleted.
in,out	*tilt*	tilt angle along the Y dimension of strip. this wave is modified: cropped rows are deleted.
in,out	*phi*	azimuthal angle along the Y dimension of strip. this wave is modified: cropped rows are deleted.

Definition at line 705 of file pearl-anglescan-process.ipf.

12.5.2.15 display_hemi_scan()

```
string display_hemi_scan (
             string nickname,
             variable projection = defaultValue,
             variable graphtype = defaultValue,
             variable do_ticks = defaultValue,
             variable do_grids = defaultValue,
             string graphname = defaultValue )
```

display a plot of a hemispherical angle scan.

the scan data must exist in the current data folder. azimuth = 0 should be at 9 o'clock. then the orientation is the same as the sample at normal emission and phi = 0, the handle of the sample plate pointing to the left.

Parameters

nickname	name prefix of holo waves. may be empty.
projection	mapping function from polar to cartesian coordinates. see Projections for details. • kProjDist = 0 azimuthal equidistant • kProjStereo = 1 stereographic (default) • kProjArea = 2 azimuthal equal-area • kProjGnom = 3 gnomonic (0 <= polar < 90) • kProjOrtho = 4 orthographic
graphtype	type of graph • 1 (pol, az) trace in Igor "New Polar" (default). • 2 XPDplot (reserved, not implemented). • 3 matrix in Igor "New Polar". the matrix wave is a 2D wave with X and Y scaling corresponding to the selected projection. matrix waves can be created by interpolate_hemi_scan(). note: the pol and az waves are required as well.

Parameters

do_ticks	select which ticks to draw. value must be the arithmetic OR of all selected items. default: 3 • 0 none • 1 major azimuthal • 2 minor azimuthal S
do_grids	select which grids to draw. value must be the arithmetic OR of all selected items. default: 3 • 0 none • 1 radius at 0 and 90 degree azimuth • 2 circle at 30 and 60 degree polar
graphname	name of graph window. default: nickname if empty, a default name is assigned. if a window with this name is existing, the function brings it to the front, and does nothing else.

Returns

the name of the graph window

Definition at line 1784 of file pearl-anglescan-process.ipf.

12.5.2.16 display_polar_graph()

```
static string display_polar_graph (
             string graphname,
             variable angle_offset = defaultValue,
             variable do_ticks = defaultValue )  [static]
```

displays an empty polar graph

the graph is drawn using Wavemetrics "New Polar Graphs.ipf".

initially the graph is empty. hemispherical scans are displayed by adding a trace that coveres the whole plot area, and setting the trace color to a function of the intensity. traces are added by calling WMPolarAppendTrace.

the following items of the graph > packages menu might be useful:

- modify polar graph
- color table control
- show polar cursors
- polar graph legend

parameters can be changed programmatically as shown in the code of this function. after programmatic parameter changes, call WMPolarAxesRedrawGraphNow(graphname).

Parameters

graphname	requested name of new graph window. if empty, a default name is assigned. if a window with this name is existing, the function brings it to the front, and does nothing else.
angle_offset	azimuth (on screen) where angle 0 is plotted (zeroAngleWhere parameter of polar graphs). starting with version 1.6, the default is 0. for hemi grids created with earlier versions, it should be set to 180 for correct orientation.

Parameters

do_ticks	select which ticks to draw. value must be the arithmetic OR of all selected items. default: 3 • 0 none • 1 major azimuthal • 2 minor azimuthal

Returns

the name of the graph window.

Version

1.7 interface change: the trace drawing code is moved to display_hemi_scan, so that this function can be reused by other graph types, e.g. display_scanlines.

Definition at line 1929 of file pearl-anglescan-process.ipf.

12.5.2.17 display_scanlines()

```
string display_scanlines (
             string nickname,
             variable alpha_lo,
             variable alpha_hi,
             wave m_theta,
             wave m_tilt,
             wave m_phi,
             variable folding = defaultValue,
             variable projection = defaultValue )
```

display a polar graph with lines indicating the angles covered by an angle scan.

Parameters

nickname	nick name for output data. this will become the name of a child folder containing the output.
alpha_lo	low limit of the analyser angle.
alpha_hi	high limit of the analyser angle.
m_theta	manipulator theta angles, 0 = normal emission. size = dimsize(data, 1)
m_tilt	manipulator tilt angles, 0 = normal emission. size = dimsize(data, 1)
m_phi	manipulator phi angles, 0 = azimuthal origin. size = dimsize(data, 1)
folding	rotational averaging, default = 1
projection	mapping function from polar to cartesian coordinates. see calc_graph_radius().

Remarks

this function is extremely slow.

Definition at line 2180 of file pearl-anglescan-process.ipf.

12.5.2.18 draw_diffraction_cone()

```
variable draw_diffraction_cone (
             string graphname,
             string groupname,
             variable theta_axis,
             variable theta_inner,
             variable phi )
```

draw the circle of a diffraction cone in a stereographic polar graph.

the diffraction cone consists of a circle marking the diffraction ring, and a dot marking the axis. the cone is drawn as a group of draw objects on the UserFront layer. the objects can be edited interactively.

Parameters

graphname	name of graph window (not implemented yet).
groupname	name of a drawing group. if the group exists (from a previous cone) it is replaced. if the group doesn't exist, a new one is created.
theta_axis	polar angle of the cone axis in degrees.
theta_inner	polar angle of the innermost point of the circle in degrees.
phi	azimuthal angle of the cone axis in degrees.

Warning

EXPERIMENTAL! this function is under development. the interface and behaviour of this function may change significantly in future versions.

Definition at line 2116 of file pearl-anglescan-process.ipf.

12.5.2.19 draw_hemi_axes()

```
static string draw_hemi_axes (
             string graphname,
             variable do_grids = defaultValue )  [static]
```

draw polar and azimuthal grids in an existing polar graph.

the function adds the following draw objects to a polar graph:

- concentric circles at polar angles 0, 30, and 60 degrees with labels.
- radial axes at 0 and 90 degree azimuth.

the objects are added to the ProgFront drawing layer and will appear in front of the data trace. in interactive drawing mode, you can select the active drawing layer by clicking the tree icon while holding the Alt key.

the graph must have been created by display_polar_graph(). the function reads the projection mode from the window user data "projection".

Parameters

graphname	name of graph window.

Parameters

do_grids	select which optional grids to draw. value must be the arithmetic OR of all selected items. default: 3 • 0 none • 1 radius at 0 and 90 degree azimuth • 2 circle at 30 and 60 degree polar

Warning

EXPERIMENTAL! this function is under development. the interface and behaviour of this function may change significantly in future versions.

Definition at line 2044 of file pearl-anglescan-process.ipf.

12.5.2.20 duplicate_hemi_scan()

```
variable duplicate_hemi_scan (
             string source_nickname,
             dfref dest_folder,
             string dest_nickname,
             variable xpdplot = defaultValue )
```

duplicate a hemispherical scan dataset.

this function works only for hemi scans created by make_hemi_grid() (or compatible functions). the angle grid is recreated rather than copied point-by-point. the new dataset is independent from the original one.

if the version of the source dataset is pre 1.6, it is converted to version 1.6.

Parameters

source_nickname	name prefix for waves. source data must be in current data folder.
dest_folder	destination folder. folder must exist.
dest_nickname	name prefix for destination waves. must be unique in the current data folder. otherwise existing waves get overwritten. may be empty.
xpdplot	XPDplot compatibility • 0 (default) create the data structures required by this module • 1 create additional waves and notebook required by XPDplot

Definition at line 1579 of file pearl-anglescan-process.ipf.

12.5.2.21 find_hemi_data()

```
dfr find_hemi_data (
             string nickname,
             string * prefix,
             string * intwave )
```

finds the folder, prefix and name of holo waves given their nick name

the function looks for holo waves in the following order:

1. if nickname is empty, check for prefix-less waves in current folder.
2. if nickname is the name of a child folder in the current data folder, clear the (prefix-less) waves in the child folder.
3. nickname is prefix of waves in current folder.
4. nickname is prefix of waves in root folder.

Parameters

in	*nickname*	folder name or name prefix of holo waves. may be empty.
out	*prefix*	name prefix of waves. may be empty.
out	*intwave*	name of intensity/values wave

Returns

reference of the data folder which contains the waves

Definition at line 1498 of file pearl-anglescan-process.ipf.

12.5.2.22 get_hemi_nickname()

```
string get_hemi_nickname (
            wave w )
```

finds the nick name given any hemi wave

the nick name is either the name of a child folder in the current data folder (PEARL specification), or a prefix of the hemi wave names (XPDplot specification).

Returns

the nick name

Definition at line 1442 of file pearl-anglescan-process.ipf.

12.5.2.23 get_hemi_prefix()

```
string get_hemi_prefix (
            wave w )
```

finds the prefix given any hemi wave

the prefix is the part of the wave name before the first underscore. the prefix is used by XPDplot where it is identical to the nick name. the prefix is empty in the PEARL specification.

Returns

the prefix

Definition at line 1467 of file pearl-anglescan-process.ipf.

12.5.2.24 hemi_add_anglescan()

```
variable hemi_add_anglescan (
             string nickname,
             wave values,
             wave polar,
             wave azi,
             wave weights = defaultValue )
```

add an arbitrary angle scan to a hemispherical scan grid.

the hemi grid must have been created in the current data folder by the make_hemi_grid function. the function determines the bin size at the given polar angle, and adds all data points which fall into a bin. a point which lies exactly on the upper boundary falls into the next bin. this function does not clear previous values before adding new data. values are added to the _tot wave, weights to the _wt wave. the intensity (_i) wave is calculated as _tot / _wt.

Definition at line 2538 of file pearl-anglescan-process.ipf.

12.5.2.25 hemi_add_aziscan()

```
variable hemi_add_aziscan (
             string nickname,
             wave values,
             variable polar,
             wave azi,
             wave weights = defaultValue )
```

add an azimuthal scan to a hemispherical scan grid.

the hemi grid must have been created in the current data folder by the make_hemi_grid function. the function determines the bin size at the given polar angle, and calculates the mean values of the data points which fall into a bin. a point which lies exactly on the upper boundary falls into the next bin.

Definition at line 2608 of file pearl-anglescan-process.ipf.

12.5.2.26 hemi_azi_cut()

```
wave hemi_azi_cut (
             string nickname,
             variable pol )
```

extract an azimuthal cut from a hemispherical scan

the function extracts all azimuthal angles that are present for the given polar angle.

the hemi grid must have been created in the current data folder by the make_hemi_grid function. correct ordering is required.

Parameters

nickname	name of the scan dataset. can be empty if no prefix is used. the dataset must be in the current datafolder.
pol	polar angle in degrees

Returns

reference of the created wave. the wave has the same name as the intensity wave of the dataset with the suffix "_azi" and the azimuthal angle rounded to integer. it is created in the same datafolder as the original data.

Definition at line 3159 of file pearl-anglescan-process.ipf.

12.5.2.27 hemi_polar_cut()

```
wave hemi_polar_cut (
            string nickname,
            variable azim )
```

extract a polar cut from a hemispherical scan.

for each polar angle, the function first extracts all azimuthal angles. the intensity is then interpolated between the nearest neighbours of the given azimuth.

the hemi grid must have been created in the current data folder by the make_hemi_grid function. correct ordering is required.

Parameters

nickname	name of the scan dataset. can be empty if no prefix is used. the dataset must be in the current datafolder.
azim	azimuthal angle in degrees

Returns

reference of the created wave. the wave has the same name as the intensity wave of the dataset with the suffix "_azi" and the azimuthal angle rounded to integer. it is created in the same datafolder as the original data.

Definition at line 3073 of file pearl-anglescan-process.ipf.

12.5.2.28 import_tpi_scan()

```
variable import_tpi_scan (
            string nickname,
            wave theta,
            wave phi,
            wave intensity,
            variable folding = defaultValue,
            variable npolar = defaultValue,
            variable nograph = defaultValue,
            variable xpdplot = defaultValue )
```

import a hemispherical scan from theta-phi-intensity waves and display it

in the tpi format, the hemi scan data is represented by a triple of flat one-dimensional waves corresponding to the polar angle (theta), azimuthal angle (phi) and intensity. no specific sort order is required.

Parameters

nickname	nick name for output data • in default mode, this will become the name of a child folder containing the output. • in XPDplot mode, this will become a prefix of the generated data in the root folder.
theta	theta angles, 0 = normal emission.
phi	phi angles, 0 = azimuthal origin. size = dimsize(data, 1)
intensity	intensity wave, see requirements above.
npolar	number of polar angles, determines polar and azimuthal step size. default = 91 (1 degree steps)
folding	rotational averaging. example: 3 = average to 3-fold symmetry. default = 1.
nograph	display a new graph window? • 0 (default) display a new polar graph • 1 don't display a new graph
xpdplot	XPDplot compatibility • 0 (default) create waves in child folder $nickname • 1 create waves in root folder (compatible with XPDplot)

Definition at line 2985 of file pearl-anglescan-process.ipf.

12.5.2.29 interpolate_hemi_scan()

```
variable interpolate_hemi_scan (
             string nickname,
             variable projection = defaultValue )
```

interpolate a hemispherical scan onto a rectangular grid

the scan data must exist in the current data folder or in the sub-folder given by the nickname parameter.

the interpolated data is written to a new two-dimensional wave "matrix". the wave has a fixed size of 181 x 181 points optimized for 1-degree polar steps.

missing values (nan) are interpolated. this works well only if the missing values are reasonable sparse. the function also applies a gaussian filter to smooth the image. empty rings at high polar angles map are preserved.

to display the result call display_hemi_scan() with graphtype=3.

Parameters

nickname	name prefix of holo waves. may be empty.
projection	mapping function from polar to cartesian coordinates. see Projections for details. • kProjDist = 0 azimuthal equidistant • kProjStereo = 1 stereographic (default) • kProjArea = 2 azimuthal equal-area • kProjGnom = 3 gnomonic (0 <= polar < 90) • kProjOrtho = 4 orthographic

Definition at line 2726 of file pearl-anglescan-process.ipf.

12.5.2.30 line_average()

```
static variable line_average (
             wave source,
             wave dest )  [static]
```

Definition at line 1131 of file pearl-anglescan-process.ipf.

12.5.2.31 load_hemi_scan()

```
variable load_hemi_scan (
             string nickname,
             string pathname,
             string filename )
```

load a hemispherical scan from an Igor text file

Todo function not implemented

Definition at line 2936 of file pearl-anglescan-process.ipf.

12.5.2.32 make_hemi_grid()

```
variable make_hemi_grid (
             variable npol,
             string nickname,
             variable xpdplot = defaultValue )
```

create a hemispherical, constant solid angle grid

all necessary waves are created in the current data folder with step size 90 / (npol - 1)

adapted from XPDplot 8.03

Parameters

npol	number of polar angles, determines polar and azimuthal step size. recommended 91 for 1-degree steps.
nickname	name prefix for waves. nick name must be unique in the current data folder. otherwise existing waves get overwritten. may be empty.
xpdplot	XPDplot compatibility • 0 (default) create the data structures required by this module • 1 create additional waves and notebook required by XPDplot

Definition at line 1291 of file pearl-anglescan-process.ipf.

12.5.2.33 normalize_strip_2d()

```
variable normalize_strip_2d (
             wave strip,
             wave theta,
             variable theta_offset = defaultValue,
             variable smooth_method = defaultValue,
             variable smooth_factor = defaultValue,
             variable check = defaultValue )
```

divide the strip by a two-dimensional normalization function.

Warning

experimental. this function is under development.

Parameters

check	enable output of intermediate results • 0 (default) don't create additional waves • 1 create check waves in the current folder • 2 calculate check waves only, do not modify strip

Returns

if check waves are enabled, the following waves are created (overwritten if existing):

- check_dist average theta distribution
- check_smoo smoothed distribution used to normalize the strip

Definition at line 613 of file pearl-anglescan-process.ipf.

12.5.2.34 normalize_strip_phi()

```
variable normalize_strip_phi (
             wave strip,
             wave theta,
             wave phi,
             variable theta_offset = defaultValue,
             variable theta_range = defaultValue,
             variable check = defaultValue )
```

divide the strip by a sine function in phi (wobble correction).

the sine function is a curve fit to the intensity integrated over detector angle with a period of 360°.

this normalization may be useful if the intensity varies with a 360° periodicity in the azimuthal angle, e.g. due to misalignment of the surface normal and the azimuthal rotation axis of the manipulator (wobble). note, however, that this function does not correct other effects of wobble such as angle shifts.

the strip is normalized in place, previous data is overwritten.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan

Parameters

in	*theta*	polar manipulator angle.
in	*phi*	azimuthal manipulator angle, arbitrary offset.
in	*theta_offset*	theta value corresponding to normal emission (default 0).
in	*theta_range*	maximum (offset corrected) theta to consider in the sine fit (default 10).
	check	enable output of intermediate results • 0 (default) don't create additional waves • 1 create check waves in the current folder • 2 calculate check waves only, do not modify strip

Returns

if check waves are enabled, the following waves are created (overwritten if existing):

- check_dist average theta distribution
- check_smoo smoothed distribution used to normalize the strip

Definition at line 283 of file pearl-anglescan-process.ipf.

12.5.2.35 normalize_strip_theta()

```
variable normalize_strip_theta (
            wave strip,
            wave theta,
            variable theta_offset = defaultValue,
            variable smooth_method = defaultValue,
            variable smooth_factor = defaultValue,
            variable check = defaultValue )
```

divide the strip by the average polar distribution.

this is a simple way to remove the polar angle dependence. the strip is normalized in place, previous data is overwritten.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan
in	*theta*	polar manipulator angle, 0 = normal emission, 90 = grazing emission
in	*theta_offset*	
in	*smooth_method*	smoothing method • 0 none • 1 binomial (requires monotonic theta), see Igor's Smooth operation • 2 boxcar (requires monotonic theta), see Igor's Smooth operation • 3 polynomial fit per slice • 4 (default) Loess, see Igor's Loess operation caution: binomial and boxcar smoothing are not aware of theta. this may give unpredictable results if theta is non-monotonic.

Parameters

in	*smooth_factor*	smoothing parameter, depends on smooth_method • binomial/boxcar: see Igor's Smooth operation • loess: see Igor's Loess operation, 0 <= smooth_factor <= 1, default 0.5 • polynomial fit: polynomial degree, 1 = linear (default), 2 = quadratic
	check	enable output of intermediate results • 0 (default) don't create additional waves • 1 create check waves in the current folder • 2 calculate check waves only, do not modify strip

Returns

if check waves are enabled, the following waves are created (overwritten if existing):

- check_dist average theta distribution
- check_smoo smoothed distribution used to normalize the strip

Definition at line 366 of file pearl-anglescan-process.ipf.

12.5.2.36 normalize_strip_theta_scans()

```
variable normalize_strip_theta_scans (
             wave strip,
             wave theta,
             variable theta_offset = defaultValue,
             variable smooth_method = defaultValue,
             variable smooth_factor = defaultValue,
             variable check = defaultValue )
```

divide the strip piecewise by a smooth polar distribution.

Warning

experimental. this function is under development.

Definition at line 523 of file pearl-anglescan-process.ipf.

12.5.2.37 normalize_strip_thetaphi()

```
variable normalize_strip_thetaphi (
             wave strip,
             wave theta,
             wave phi,
             variable theta_offset = defaultValue,
             variable smooth_method = defaultValue,
             variable smooth_factor = defaultValue,
             variable check = defaultValue )
```

divide the strip by a smooth polar-azimuthal distribution.

this is a simple way to remove the polar angle dependence. in contrast to normalize_strip_theta this function also removes a smooth variation over azimuthal angles.

the strip is normalized in place, previous data is overwritten.

Warning

experimental. this function is under development.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan
in	*theta*	polar manipulator angle, 0 = normal emission, 90 = grazing emission
in	*phi*	azimuthal manipulator angle.
in	*theta_offset*	
in	*smooth_method*	smoothing method • 0 none • 4 (default) Loess, see Igor's Loess operation
in	*smooth_factor*	smoothing parameter, depends on smooth_method • loess: see Igor's Loess operation, 0 <= smooth_factor <= 1, default 0.5
	check	enable output of intermediate results • 0 (default) don't create additional waves • 1 create check waves in the current folder • 2 calculate check waves only, do not modify strip

Returns

if check waves are enabled, the following waves are created (overwritten if existing):

- check_dist average theta distribution
- check_smoo smoothed distribution used to normalize the strip

Definition at line 464 of file pearl-anglescan-process.ipf.

12.5.2.38 normalize_strip_x()

```
variable normalize_strip_x (
            wave strip,
            variable smooth_method = defaultValue,
            variable smooth_factor = defaultValue,
            variable check = defaultValue )
```

divide the strip by the average X distribution.

this is a simple way to remove the effect of the angle-dependence of the analyser transmission function. the strip is normalized in place, previous data is overwritten.

the function can handle sparse NaNs.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan
in	*smooth_method*	smoothing method • 0 none • 1 binomial, see Igor's Smooth operation • 2 boxcar, see Igor's Smooth operation • 3 scienta_ang_transm() function fit • 4 (default) LOESS smoothing, see Igor's Loess operation
in	*smooth_factor*	num parameter of Igor's Smooth operation. the default value depends on smooth_method. it is 0.5 for LOESS smoothing, 2 otherwise.
in	*check*	enable output of intermediate results • 0 (default) don't create additional waves • 1 create check waves in the current folder • 2 calculate check waves only, do not modify strip

Returns

if check waves are enabled, the following waves are created (overwritten if existing):

- check_dist average X distribution
- check_smoo smoothed distribution used to normalize the strip

Definition at line 187 of file pearl-anglescan-process.ipf.

12.5.2.39 pizza_service()

```
variable pizza_service (
            wave data,
            string nickname,
            variable theta_offset,
            variable tilt_offset,
            variable phi_offset,
            variable npolar = defaultValue,
            variable nograph = defaultValue,
            variable folding = defaultValue,
            variable xpdplot = defaultValue )
```

create a pizza plot from a measured (energy-integrated) data strip

accepts angle-scan data as returned by adh5_load_reduced(), maps them onto a hemispherical scan grid, and displays a polar graph.

Parameters

data	2D intensity wave, see requirements above • X-axis analyser angle • Y-axis manipulator scan. no specific ordering required. manipulator angle waves (ManipulatorTheta, ManipulatorTilt, ManipulatorPhi) must be in the subfolder *attr* below the data wave.
nickname	nick name for output data • in default mode, this will become the name of a child folder containing the output. • in XPDplot mode, this will become a prefix of the generated data in the root folder.
theta_offset	manipulator theta angle corresponding to normal emission. the offset is subtracted from the ManipulatorTheta wave before processing.
tilt_offset	manipulator tilt angle corresponding to normal emission the offset is subtracted from the ManipulatorTilt wave before processing.
phi_offset	manipulator phi angle corresponding to phi_result = 0 the offset is subtracted from the ManipulatorPhi wave before processing.
npolar	number of polar angles, determines polar and azimuthal step size. default = 91 (1 degree steps)
folding	rotational averaging, default = 1
nograph	display a new graph window? • 0 (default) display a new polar graph • 1 don't display a new graph
xpdplot	XPDplot compatibility • 0 (default) create waves in child folder $nickname • 1 create waves in root folder (compatible with XPDplot)

Attention

if you modify the structure of the data wave, e.g. delete some angles, this function cannot be used because the manipulator settings do not correspond to the original manipulator waves! instead, create your own manipulator waves and use pizza_service_2().

Definition at line 769 of file pearl-anglescan-process.ipf.

12.5.2.40 pizza_service_2()

```
variable pizza_service_2 (
             wave data,
             string nickname,
             wave m_theta,
             wave m_tilt,
             wave m_phi,
             variable npolar = defaultValue,
             variable nograph = defaultValue,
             variable folding = defaultValue,
             variable xpdplot = defaultValue )
```

create a pizza plot from a measured (energy-integrated) data strip

accepts angle-scan data as returned by adh5_load_reduced(), maps them onto a hemispherical scan grid, and displays a polar graph.

the behaviour of this function is the same as pizza_service() except that the manipulator waves are specified explicitly.

Parameters

data	2D intensity wave, see requirements above • X-axis analyser angle • Y-axis manipulator scan. no specific ordering required. manipulator angle waves (ManipulatorTheta, ManipulatorTilt, ManipulatorPhi) must be in the subfolder *attr* below the data wave.
nickname	nick name for output data • in default mode, this will become the name of a child folder containing the output. • in XPDplot mode, this will become a prefix of the generated data in the root folder.
m_theta	manipulator theta angles, 0 = normal emission. size = dimsize(data, 1)
m_tilt	manipulator tilt angles, 0 = normal emission. size = dimsize(data, 1)
m_phi	manipulator phi angles, 0 = azimuthal origin. size = dimsize(data, 1)
npolar	number of polar angles, determines polar and azimuthal step size. default = 91 (1 degree steps)
folding	rotational averaging, default = 1
nograph	display a new graph window? • 0 (default) display a new polar graph • 1 don't display a new graph
xpdplot	XPDplot compatibility • 0 (default) create waves in child folder $nickname • 1 create waves in root folder (compatible with XPDplot)

Definition at line 862 of file pearl-anglescan-process.ipf.

12.5.2.41 polar_graph_hook()

```
static variable polar_graph_hook (
            WMWinHookStruct * s )  [static]
```

polar graph window hook

this hook converts the cursor positions to polar coordinates and displays them in a text box on the graph. the text box is visible while the cursor info box is visible.

Definition at line 2473 of file pearl-anglescan-process.ipf.

12.5.2.42 prepare_hemi_scan_display()

```
string prepare_hemi_scan_display (
            string nickname,
```

```
            variable projection = defaultValue )
```

create waves for plotting a hemispherical angle scan.

the scan data must exist in the current data folder.

Parameters

nickname	name prefix of holo waves. may be empty.
projection	mapping function from polar to cartesian coordinates. see Projections for details. • kProjDist = 0 azimuthal equidistant • kProjStereo = 1 stereographic (default) • kProjArea = 2 azimuthal equal-area • kProjGnom = 3 gnomonic (0 <= polar < 90) • kProjOrtho = 4 orthographic

Definition at line 1702 of file pearl-anglescan-process.ipf.

12.5.2.43 quick_pizza_image()

```
variable quick_pizza_image (
            wave data,
            string nickname,
            variable theta_offset,
            variable tilt_offset,
            variable phi_offset,
            variable npolar = defaultValue,
            variable nograph = defaultValue,
            variable folding = defaultValue )
```

map angle scan data onto a rectangular grid in stereographic projection

accepts angle-scan data as returned by adh5_load_reduced, maps them onto a rectangular grid in stereographic projection

Parameters

data	2D data wave, X-axis = analyser angle, Y-axis = manipulator scan (no specific ordering required)

Precondition

manipulator angles as attributes in attr folder next to the data wave

Warning

EXPERIMENTAL

Definition at line 2792 of file pearl-anglescan-process.ipf.

12.5.2.44 rotate_hemi_scan()

```
variable rotate_hemi_scan (
             string nickname,
             variable angle )
```

azimuthally rotate a hemispherical scan dataset.

this function works only for hemi scans created by make_hemi_grid() (or compatible functions).

Parameters

nickname	name prefix for waves. source data must be in current data folder.
angle	azimuthal rotation angle in degrees.

Definition at line 1654 of file pearl-anglescan-process.ipf.

12.5.2.45 save_hemi_scan()

```
variable save_hemi_scan (
             string nickname,
             string pathname,
             string filename )
```

save a hemispherical scan to an Igor text file

Definition at line 2903 of file pearl-anglescan-process.ipf.

12.5.2.46 set_contrast()

```
variable set_contrast (
             variable pcmin,
             variable pcmax,
             string graphname = defaultValue,
             string colortable = defaultValue )
```

set the pseudocolor contrast by percentile.

set the minimum and maximum values of the pseudocolor scale such that a specified percentile of the distribution lies outside the limits.

the new contrast is applied to traces and images of the selected graph that have pseudocolor tables.

the function is not specific to angle scans. it can be used for any pseudocolor trace or image plots except contour plots.

Parameters

pcmin	percentile below the minimum color (0-100).
pcmax	percentile above the maximum color (0-100).
graphname	name of graph. default: top graph.
colortable	name of new colortable. default: keep current table.

Definition at line 3247 of file pearl-anglescan-process.ipf.

12.5.2.47 set_polar_graph_cursor()

```
variable set_polar_graph_cursor (
             string nickname,
             string cursorname,
             variable polar_angle,
             variable azim_angle,
             string graphname = defaultValue )
```

Definition at line 2493 of file pearl-anglescan-process.ipf.

12.5.2.48 show_analyser_line()

```
variable show_analyser_line (
             variable theta,
             variable tilt,
             variable phi,
             variable theta_offset,
             variable tilt_offset,
             variable phi_offset,
             variable npolar = defaultValue,
             variable nograph = defaultValue,
             variable xpdplot = defaultValue )
```

calculate and display the line seen by the analyser for a specific emission angle

this can be used to compare to an hemispherical plot and check the manipulator angle.

Parameters

theta	manipulator theta angle
tilt	manipulator tilt angle
phi	manipulator phi angle
theta_offset	manipulator theta angle corresponding to normal emission
tilt_offset	manipulator tilt angle corresponding to normal emission
phi_offset	manipulator phi angle corresponding to phi_result = 0
npolar	number of polar angles, determines polar and azimuthal step size. default = 91 (1 degree steps)
nograph	display a new graph window? • 0 (default) display a new polar graph • 1 don't display a new graph
xpdplot	XPDplot compatibility • 0 (default) create waves in child folder $nickname • 1 create waves in root folder (compatible with XPDplot)

Remarks

the function creates angle scan data under the nickname *analyser*.

Definition at line 979 of file pearl-anglescan-process.ipf.

12.5.2.49 strip_delete_frames()

```
variable strip_delete_frames (
            wave strip,
            variable qlo,
            variable qhi,
            wave theta,
            wave tilt,
            wave phi )
```

delete a contiguous range of frames from a strip.

this can be used to remove a region of bad frames due to, e.g., measurement problems. the function operates on 2D intensity data and manipulator coordinates at the same time.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan. the result is written to the original wave.
in,out	*theta*	1D data, manipulator scan. the result is written to the original wave.
in,out	*tilt*	1D data, manipulator scan. the result is written to the original wave.
in,out	*phi*	1D data, manipulator scan. the result is written to the original wave.
in	*qlo*	point index of first frame to delete.
in	*qhi*	point index of last frame to delete. qhi must be greater or equal than qlo.

Definition at line 105 of file pearl-anglescan-process.ipf.

12.5.2.50 trim_hemi_scan()

```
variable trim_hemi_scan (
            string nickname,
            variable theta_max )
```

trim a hemispherical scan at grazing angle

the function recalaculates the values wave from totals and weights but sets elements above a given polar angle to nan.

Parameters

nickname	name of the scan dataset. can be empty if no prefix is used. the dataset must be in the current datafolder.
theta_max	highest polar angle to keep (0...90 degrees).

Definition at line 3034 of file pearl-anglescan-process.ipf.

12.5.2.51 update_polar_info()

```
static variable update_polar_info (
            string graphname )  [static]
```

update the angles info based on cursors A and B of a given polar graph window

the function reads the projection mode from the user data of the graph window and the zeroAngleWhere variable

from the associated WMPolarGraph data folder.

the calculated angles are written to the csrA_theta, csrA_phi, csrB_theta, and csrB_phi global variables in the polar graph data folder. the angles text box of the graph updates from to these variables dynamically.

Parameters

graphname	name of polar graph window

Definition at line 2438 of file pearl-anglescan-process.ipf.

12.5.3 Variable Documentation

12.5.3.1 kProjArea

```
const variable kProjArea = 2
```

Definition at line 2270 of file pearl-anglescan-process.ipf.

12.5.3.2 kProjDist

```
const variable kProjDist = 0
```

Definition at line 2268 of file pearl-anglescan-process.ipf.

12.5.3.3 kProjGnom

```
const variable kProjGnom = 3
```

Definition at line 2271 of file pearl-anglescan-process.ipf.

12.5.3.4 kProjOrtho

```
const variable kProjOrtho = 4
```

Definition at line 2272 of file pearl-anglescan-process.ipf.

12.5.3.5 kProjScaleArea

```
const variable kProjScaleArea = 2  [static]
```

Definition at line 2276 of file pearl-anglescan-process.ipf.

12.5.3.6 kProjScaleDist

```
const variable kProjScaleDist = 2  [static]
```

Definition at line 2274 of file pearl-anglescan-process.ipf.

12.5.3.7 kProjScaleGnom

```
const variable kProjScaleGnom = 0.06744519021  [static]
```

Definition at line 2278 of file pearl-anglescan-process.ipf.

12.5.3.8 kProjScaleOrtho

```
const variable kProjScaleOrtho = 2  [static]
```

Definition at line 2279 of file pearl-anglescan-process.ipf.

12.5.3.9 kProjScaleStereo

```
const variable kProjScaleStereo = 2  [static]
```

Definition at line 2275 of file pearl-anglescan-process.ipf.

12.5.3.10 kProjStereo

```
const variable kProjStereo = 1
```

Definition at line 2269 of file pearl-anglescan-process.ipf.

12.6 pearl-anglescan-tracker.ipf File Reference

```
#include "pearl-area-profiles"
#include "pearl-area-import"
#include "pearl-scienta-preprocess"
#include "pearl-anglescan-process"
#include <New Polar Graphs>
```

Functions

- static variable AfterCompiledHook ()

  initialize package data once when the procedure is first loaded
- static variable init_package ()
- static variable save_prefs ()

  save persistent package data to the preferences file.
- static variable load_prefs ()

  load persistent package data from the preferences file.
- static variable IgorQuitHook (string app)

  disconnect EPICS channels before Igor quits.
- variable ast_setup ()

set up data structures, display graph, and try to connect to analyser.

- variable ast_prepare (variable theta_offset=defaultValue, variable tilt_offset=defaultValue, variable phi_↩ offset=defaultValue)

  prepare for new measurement and clear the data buffer.
- variable ast_set_processing (string reduction_func, string reduction_params)

  set the data processing parameters
- variable ast_add_image (wave image, variable theta, variable tilt, variable phi)

  process and add a detector image to the tracker scan.
- variable ast_export (dfref folder, string nickname, variable xpdplot=defaultValue)

  export tracker data to a separate, independent data set.
- variable ast_import (string nickname)

  import tracker data from an existing angle scan dataset.
- variable ast_update_detector (variable theta, variable tilt, variable phi, variable range)

  update the current position indicator.
- variable ast_close ()

  stop tracker, close graph, release data structures.
- static variable setup_data ()
- static variable extend_data (variable num_slices)

  extend the data buffer for the next polar scan
- static variable setup_detector ()
- static variable add_image_data (wave image, variable theta, variable tilt, variable phi)

  reduce a detector image and add the result to the data buffer.
- static variable process_image_data ()

  process the data buffer to generate the tracker dataset.
- static variable update_detector (variable theta, variable tilt, variable phi, variable range)

  update the current position indicator.
- static variable setup_graph ()

  create the graph window.
- static variable update_data_graph ()
- static variable update_detector_graph ()
- static variable epics_connect ()

  connect the angle scan tracker to EPICS
- static variable epics_disconnect_chid (string chid_var_name)
- static variable epics_disconnect ()
- static variable ast_window_hook (WMWinHookStruct ∗s)

  window hook
- variable ast_callback_data (variable chan)

  callback function for new analyser data from EPICS.
- variable ast_callback_detector (variable chan)

  callback function for new detector state from EPICS.
- variable ast_callback_manip (variable chan)

  callback function for new manipulator position from EPICS.
- static variable bp_capture (WMButtonAction ∗ba)
- static variable toggle_capture ()
- static variable update_capture ()
- static variable pmp_data (WMPopupAction ∗pa)
- static variable pmp_data_mouseup (WMPopupAction ∗pa)
- static variable export_tracker_data ()

  export tracker data (with prompt)
- static variable import_tracker_data ()

  import tracker data (with prompt)
- static variable save_tracker_data ()

save tracker data to file (with prompt)

- static variable load_tracker_data ()

  import tracker data from file (with prompt)
- static variable pmp_parameters (WMPopupAction *pa)
- static variable pmp_parameters_mouseup (WMPopupAction *pa)
- static variable edit_reduction_params ()
- static variable edit_offsets ()

Variables

- version
- static const string package_path = "root:packages:pearl_anglescan_tracker:"

  data folder path
- static const string prefs_objects = "projection;theta_offset;tilt_offset;phi_offset;reduction_func;reduction_↩ params"

  semicolon-separated list of persistent variable, string, and wave names

12.6.1 Function Documentation

12.6.1.1 add_image_data()

```
static variable add_image_data (
            wave image,
            variable theta,
            variable tilt,
            variable phi )  [static]
```

reduce a detector image and add the result to the data buffer.

Parameters

image	detector image with correct X (energy) and Y (angle) scaling
theta	polar angle of manipulator
tilt	tilt angle of manipulator
phi	azimuthal angle of manipulator

the manipulator angles are corrected by the preset offsets internally.

Definition at line 453 of file pearl-anglescan-tracker.ipf.

12.6.1.2 AfterCompiledHook()

```
static variable AfterCompiledHook ( )  [static]
```

initialize package data once when the procedure is first loaded

Definition at line 73 of file pearl-anglescan-tracker.ipf.

12.6.1.3 ast_add_image()

```
variable ast_add_image (
             wave image,
             variable theta,
             variable tilt,
             variable phi )
```

process and add a detector image to the tracker scan.

Parameters

image	detector image with correct X (energy) and Y (angle) scaling
theta	polar angle of manipulator
tilt	tilt angle of manipulator
phi	azimuthal angle of manipulator

the manipulator angles are corrected by the preset offsets internally.

Definition at line 285 of file pearl-anglescan-tracker.ipf.

12.6.1.4 ast_callback_data()

```
variable ast_callback_data (
             variable chan )
```

callback function for new analyser data from EPICS.

Definition at line 856 of file pearl-anglescan-tracker.ipf.

12.6.1.5 ast_callback_detector()

```
variable ast_callback_detector (
             variable chan )
```

callback function for new detector state from EPICS.

save the manipulator position at the beginning of image acquisition. it is used by ast_callback_data().

Definition at line 961 of file pearl-anglescan-tracker.ipf.

12.6.1.6 ast_callback_manip()

```
variable ast_callback_manip (
             variable chan )
```

callback function for new manipulator position from EPICS.

Definition at line 990 of file pearl-anglescan-tracker.ipf.

12.6.1.7 ast_close()

```
variable ast_close ( )
```

stop tracker, close graph, release data structures.

Definition at line 360 of file pearl-anglescan-tracker.ipf.

12.6.1.8 ast_export()

```
variable ast_export (
             dfref folder,
             string nickname,
             variable xpdplot = defaultValue )
```

export tracker data to a separate, independent data set.

the exported data can then be used for further processing. the data is exported to the current data folder, or root if XPDplot compatibility is requested.

Parameters

folder	destination folder path
nickname	name prefix for waves
xpdplot	xpdplot compatibility, see make_hemi_grid() for details • 0 (default) • 1 create additional waves and notebook required by XPDplot

Definition at line 307 of file pearl-anglescan-tracker.ipf.

12.6.1.9 ast_import()

```
variable ast_import (
             string nickname )
```

import tracker data from an existing angle scan dataset.

Parameters

nickname	name prefix for waves. data must be in current data folder.

Definition at line 329 of file pearl-anglescan-tracker.ipf.

12.6.1.10 ast_prepare()

```
variable ast_prepare (
             variable theta_offset = defaultValue,
             variable tilt_offset = defaultValue,
             variable phi_offset = defaultValue )
```

prepare for new measurement and clear the data buffer.

optionally, set new manipulator offsets. the offsets are the manipulator readback coordinates where the sample surface is oriented in normal emission, and the handle of the sample plate points horizontally to the left (9 o'clock).

Parameters

theta_offset	set new theta offset. default: no change.
tilt_offset	set new tilt offset. default: no change.
phi_offset	set new phi offset. default: no change.

Definition at line 211 of file pearl-anglescan-tracker.ipf.

12.6.1.11 ast_set_processing()

```
variable ast_set_processing (
             string reduction_func,
             string reduction_params )
```

set the data processing parameters

the parameters will be effective for subsequent measurements only. previously acquired data is not affected. the processing parameters are saved with the preferences.

Parameters

reduction_func	name of custom reduction function, e.g. "int_linbg_reduction". any user-defined function with the same signature as adh5_default_reduction() is allowed.
reduction_params	parameter string for the reduction function. the format depends on the actual function. for int_linbg_reduction, e.g., "Lcrop=0.1;Hcrop=0.1;Lsize=0.2;Hsize=0.2;Cpos=0.5;Csize=0.4".

Definition at line 261 of file pearl-anglescan-tracker.ipf.

12.6.1.12 ast_setup()

```
variable ast_setup ( )
```

set up data structures, display graph, and try to connect to analyser.

Definition at line 194 of file pearl-anglescan-tracker.ipf.

12.6.1.13 ast_update_detector()

```
variable ast_update_detector (
             variable theta,
             variable tilt,
             variable phi,
             variable range )
```

update the current position indicator.

Parameters

theta	polar angle of manipulator
tilt	tilt angle of manipulator
phi	azimuthal angle of manipulator

Parameters

range	angle range (60 or 45)

the manipulator angles are corrected by the preset offsets internally.

Definition at line 349 of file pearl-anglescan-tracker.ipf.

12.6.1.14 ast_window_hook()

```
static variable ast_window_hook (
             WMWinHookStruct * s )  [static]
```

window hook

disconnects from EPICS when the window is closed.

Definition at line 841 of file pearl-anglescan-tracker.ipf.

12.6.1.15 bp_capture()

```
static variable bp_capture (
             WMButtonAction * ba )  [static]
```

Definition at line 1023 of file pearl-anglescan-tracker.ipf.

12.6.1.16 edit_offsets()

```
static variable edit_offsets ( )  [static]
```

Definition at line 1244 of file pearl-anglescan-tracker.ipf.

12.6.1.17 edit_reduction_params()

```
static variable edit_reduction_params ( )  [static]
```

Definition at line 1228 of file pearl-anglescan-tracker.ipf.

12.6.1.18 epics_connect()

```
static variable epics_connect ( )  [static]
```

connect the angle scan tracker to EPICS

the tracker uses channels of the analyser and the manipulator.

if the EPICS XOP is not loaded, the function does nothing. if channels are not available, the function exits with an error code after a timeout of 5 seconds. the Igor run-time error status is reset to suppress the error dialog.

Returns

zero if successful, non-zero if an error occurred

Todo the X03DA channel names are hard-coded.

Definition at line 680 of file pearl-anglescan-tracker.ipf.

12.6.1.19 epics_disconnect()

```
static variable epics_disconnect ( )  [static]
```

Definition at line 806 of file pearl-anglescan-tracker.ipf.

12.6.1.20 epics_disconnect_chid()

```
static variable epics_disconnect_chid (
             string chid_var_name )  [static]
```

Definition at line 792 of file pearl-anglescan-tracker.ipf.

12.6.1.21 export_tracker_data()

```
static variable export_tracker_data ( )  [static]
```

export tracker data (with prompt)

Definition at line 1105 of file pearl-anglescan-tracker.ipf.

12.6.1.22 extend_data()

```
static variable extend_data (
             variable num_slices )  [static]
```

extend the data buffer for the next polar scan

call this function if the buffer is full.

Parameters

num_slices	number of slices that the measurement contains

Definition at line 402 of file pearl-anglescan-tracker.ipf.

12.6.1.23 IgorQuitHook()

```
static variable IgorQuitHook (
             string app )  [static]
```

disconnect EPICS channels before Igor quits.

Definition at line 188 of file pearl-anglescan-tracker.ipf.

12.6.1.24 import_tracker_data()

```
static variable import_tracker_data ( )  [static]
```

import tracker data (with prompt)

Definition at line 1135 of file pearl-anglescan-tracker.ipf.

12.6.1.25 init_package()

```
static variable init_package ( )  [static]
```

Definition at line 98 of file pearl-anglescan-tracker.ipf.

12.6.1.26 load_prefs()

```
static variable load_prefs ( )  [static]
```

load persistent package data from the preferences file.

the preferences file is an Igor packed experiment file in a special preferences folder.

this function is called automatically when the procedure is first compiled, or whenever the user clicks the corresponding button.

Definition at line 163 of file pearl-anglescan-tracker.ipf.

12.6.1.27 load_tracker_data()

```
static variable load_tracker_data ( )  [static]
```

import tracker data from file (with prompt)

Definition at line 1175 of file pearl-anglescan-tracker.ipf.

12.6.1.28 pmp_data()

```
static variable pmp_data (
            WMPopupAction * pa )  [static]
```

Definition at line 1071 of file pearl-anglescan-tracker.ipf.

12.6.1.29 pmp_data_mouseup()

```
static variable pmp_data_mouseup (
            WMPopupAction * pa )  [static]
```

Definition at line 1085 of file pearl-anglescan-tracker.ipf.

12.6.1.30 pmp_parameters()

```
static variable pmp_parameters (
            WMPopupAction * pa )  [static]
```

Definition at line 1194 of file pearl-anglescan-tracker.ipf.

12.6.1.31 pmp_parameters_mouseup()

```
static variable pmp_parameters_mouseup (
            WMPopupAction * pa )  [static]
```

Definition at line 1208 of file pearl-anglescan-tracker.ipf.

12.6.1.32 process_image_data()

```
static variable process_image_data ( )  [static]
```

process the data buffer to generate the tracker dataset.

Definition at line 507 of file pearl-anglescan-tracker.ipf.

12.6.1.33 save_prefs()

```
static variable save_prefs ( )  [static]
```

save persistent package data to the preferences file.

this function is called when the user clicks the corresponding button.

Definition at line 139 of file pearl-anglescan-tracker.ipf.

12.6.1.34 save_tracker_data()

```
static variable save_tracker_data ( )  [static]
```

save tracker data to file (with prompt)

Definition at line 1164 of file pearl-anglescan-tracker.ipf.

12.6.1.35 setup_data()

```
static variable setup_data ( )  [static]
```

Definition at line 372 of file pearl-anglescan-tracker.ipf.

12.6.1.36 setup_detector()

```
static variable setup_detector ( )  [static]
```

Definition at line 434 of file pearl-anglescan-tracker.ipf.

12.6.1.37 setup_graph()

```
static variable setup_graph ( )  [static]
```

create the graph window.

Definition at line 594 of file pearl-anglescan-tracker.ipf.

12.6.1.38 toggle_capture()

```
static variable toggle_capture ( )  [static]
```

Definition at line 1037 of file pearl-anglescan-tracker.ipf.

12.6.1.39 update_capture()

```
static variable update_capture ( )  [static]
```

Definition at line 1055 of file pearl-anglescan-tracker.ipf.

12.6.1.40 update_data_graph()

```
static variable update_data_graph ( )  [static]
```

Definition at line 645 of file pearl-anglescan-tracker.ipf.

12.6.1.41 update_detector()

```
static variable update_detector (
            variable theta,
            variable tilt,
            variable phi,
            variable range )  [static]
```

update the current position indicator.

Parameters

theta	polar angle of manipulator
tilt	tilt angle of manipulator
phi	azimuthal angle of manipulator
range	angle range (60 or 45)

the manipulator angles are corrected by the preset offsets internally.

Definition at line 559 of file pearl-anglescan-tracker.ipf.

12.6.1.42 update_detector_graph()

```
static variable update_detector_graph ( )  [static]
```

Definition at line 657 of file pearl-anglescan-tracker.ipf.

12.6.2 Variable Documentation

12.6.2.1 package_path

```
const string package_path = "root:packages:pearl_anglescan_tracker:"  [static]
```

data folder path

Definition at line 68 of file pearl-anglescan-tracker.ipf.

12.6.2.2 prefs_objects

```
const string prefs_objects = "projection;theta_offset;tilt_offset;phi_offset;reduction_↩
func;reduction_params"  [static]
```

semicolon-separated list of persistent variable, string, and wave names

Definition at line 70 of file pearl-anglescan-tracker.ipf.

12.6.2.3 version

```
version
```

Initial value:

```
= 1.6
```

```
static const string package_name = "pearl_anglescan_tracker"
```

Definition at line 5 of file pearl-anglescan-tracker.ipf.

12.7 pearl-area-display.ipf File Reference

visualization tools for 2D and 3D data.

```
#include "pearl-compat"
```

Namespaces

- PearlAreaDisplay

  instant visualization of angle scan and manipulator position.

Functions

- static string graphname_from_dfref (dfref df, string prefix)

  compose a valid and unique graph name from a data folder reference
- string ad_display (wave image)

  open a new graph window with a 2D image.
- string ad_display_histogram (wave image)

  display the histogram of a 2D image.
- string ad_display_profiles (wave image, string filter=defaultValue)

  open a new profiles graph window.
- wave ad_add_overlay (wave image, string rgba=defaultValue)

  add an overlay on top of the displayed image
- variable ad_update_profiles (wave image)

  update a profiles graph with new data.
- variable ad_profiles_cursor_mode (wave image, variable mode)

  switch cursors on a profiles graph
- variable ad_profiles_set_cursor (wave image, string cursorname, variable xa, variable ya, variable pscale=defaultValue)

  move a cursor to the specified position in a profiles graph.
- variable ad_profiles_crosshairs (wave image, variable clear=defaultValue)

  draw permanent crosshairs in a profiles graph.
- static wave get_source_image (wave view)

  find the source image wave corresponding to the given view.
- static dfr make_view_folder (wave source)

  create a view data folder.
- static dfr get_view_folder (wave source)

  find the view data folder corresponding to the given source.
- static wave get_view_image (wave source)

  find the view image wave corresponding to the given source.
- static variable bp_reset_cursors (WMButtonAction *ba)
- static variable svp_smoothing (WMSetVariableAction *sva)
- static variable pmp_export (WMPopupAction *pa)
- variable ad_profiles_hook (WMWinHookStruct *s)

  hook function for user events in the profiles window.
- variable ad_calc_cursor_profiles (wave image)

  calculate profiles, statistics, and histogram of a cross-hair delimited region of interest.
- variable ad_calc_profiles (wave image, variable pa, variable qa, variable pb, variable qb)

  calculate profiles, statistics, and histogram of a rectangular region of interest.
- variable ad_export_profile (wave view_image, variable dim, variable trace=defaultValue, variable show=defaultValue, variable overwrite=defaultValue)

export a profile from a profiles graph to the source data folder.

- static variable set_trace_colors (string graphname)
- variable ad_calc_histogram (wave image)

  calculate the histogram.
- variable ad_default_image_filter (wave image, string options)

  abstract filter function for image display.
- variable ad_box_filter (wave image, string options)

  boxcar smoothing filter.
- variable ad_transpose_filter (wave image, string options)

  transpose image filter.
- string ad_display_brick (wave data)

  open a new "gizmo" window with three-dimensional data.
- variable ad_brick_slicer (wave data)

  open a slicer panel for 3D data.
- string ad_display_slice (wave data)

  display three-dimensional data by 2D slice.
- static variable update_slice_info ()

  update controls with data scale limits.
- variable ad_gizmo_set_plane (wave brick, variable dim, variable value)

  set the position of a slicing plane of a 3D brick in a Gizmo window.
- variable ad_profiles_set_slice (wave brick, variable dim, variable value)

  set the position of the slicing plane of a 3D brick in a profiles window.
- static variable slp_slice_position (WMSliderAction *sa)

  set slice coordinate (slider procedure).
- static variable svp_slice_position (WMSetVariableAction *sva)

  set slice coordinate (button procedure).
- static variable bp_move_slice (WMButtonAction *ba)

  move slice (button procedure).
- static variable bp_extract_slice (WMButtonAction *ba)

  export a slice (button procedure).
- static variable bp_move_slice_center (wave brick, variable dim, string posvariable)

  move the slice to the center of the dimension (button procedure).
- static variable ad_slicer_move_bg (WMBackgroundStruct *s)

  move a slice by one step (background task).
- variable ad_slicer_init_bg ()

  initialize the slice animation background task.
- variable ad_slicer_start_bg (wave brick, variable dimension, string posvariable, variable delta)

  start the animation.
- variable ad_slicer_stop_bg (string posvariable)

  stop the animation.

12.7.1 Detailed Description

visualization tools for 2D and 3D data.

these tools were initially developed for monitoring output from EPICS area detector software. they are, however, useful for any kind of intensity versus x,y(,z) data.

12.7.2 2D data

TO DO...

12.7.3 3D data

12.7.4 Function Documentation

12.7.4.1 ad_add_overlay()

```
wave ad_add_overlay (
            wave image,
            string rgba = defaultValue )
```

add an overlay on top of the displayed image

the function creates the overlay wave and returns it as function result. the name of the wave is "view_overlay" and is created in the same folder as the "view_image" wave.

Parameters

image	image wave that identifies the profiles window.

Returns

overlay overlay wave. same dimensions and scales as image, but unsigned binary. pixels that are 0 are overlaid with semi-transparent color. other pixels should be 64 (igor's mask convention).

Definition at line 329 of file pearl-area-display.ipf.

12.7.4.2 ad_box_filter()

```
variable ad_box_filter (
            wave image,
            string options )
```

boxcar smoothing filter.

filters the image in X and Y directions using Igor's Smooth operation.

Parameters

image	image to be filtered: original data and filter result.
options	smoothing factors in `key1=value1;key2=value2;...` format. • SmoothingX • SmoothingY

Definition at line 1103 of file pearl-area-display.ipf.

12.7.4.3 ad_brick_slicer()

```
variable ad_brick_slicer (
            wave data )
```

open a slicer panel for 3D data.

if a panel exists, bring it to the front.

Parameters

data	three-dimensional wave.

Definition at line 1256 of file pearl-area-display.ipf.

12.7.4.4 ad_calc_cursor_profiles()

```
variable ad_calc_cursor_profiles (
             wave image )
```

calculate profiles, statistics, and histogram of a cross-hair delimited region of interest.

Parameters

image	wave which contains the image data from the detector.

the function expects further objects as created by ad_display_profiles() in the same data folder as the image wave. the most recent profiles graph of the image must exist, and the cursors A and B must be set on the image.

Definition at line 785 of file pearl-area-display.ipf.

12.7.4.5 ad_calc_histogram()

```
variable ad_calc_histogram (
             wave image )
```

calculate the histogram.

Parameters

image	wave which contains the image data from the detector. the function expects further objects as created by ad_display_histogram() in the same data folder as the image wave.

Definition at line 1066 of file pearl-area-display.ipf.

12.7.4.6 ad_calc_profiles()

```
variable ad_calc_profiles (
             wave image,
             variable pa,
             variable qa,
             variable pb,
             variable qb )
```

calculate profiles, statistics, and histogram of a rectangular region of interest.

the region of interest a rectangle spanned by the two points A and B. pixels at these coordinates are included.

Parameters

image	wave which contains the image data.
pa	first point coordinate of A.
qa	second point coordinate of A.
pb	first point coordinate of B.
qb	second point coordinate of B.

the function expects further objects as created by ad_display_profiles() in the same data folder as the image wave.

this function does not require that the graph exists as long as the data folder is complete.

Definition at line 832 of file pearl-area-display.ipf.

12.7.4.7 ad_default_image_filter()

```
variable ad_default_image_filter (
             wave image,
             string options )
```

abstract filter function for image display.

this is a function prototype for filtering two-dimensional data for preview. to write your own filter, define a new function which has the same signature.

Parameters

image	image to be filtered: original data and filter result.
options	filter options in `key1=value1;key2=value2;...` format.

Returns

the result must be written to the incoming image wave.

Definition at line 1089 of file pearl-area-display.ipf.

12.7.4.8 ad_display()

```
string ad_display (
             wave image )
```

open a new graph window with a 2D image.

this is essentially `display; appendimage`. the graph is directly linked to the image wave. it is, thus, updated automatically.

Parameters

image	wave which contains the image data.

Returns

(string) name of the graph window

Definition at line 85 of file pearl-area-display.ipf.

12.7.4.9 ad_display_brick()

```
string ad_display_brick (
            wave data )
```

open a new "gizmo" window with three-dimensional data.

Parameters

data	three-dimensional wave.

Returns

name of the gizmo window.

Definition at line 1141 of file pearl-area-display.ipf.

12.7.4.10 ad_display_histogram()

```
string ad_display_histogram (
            wave image )
```

display the histogram of a 2D image.

the function will create additional objects in the same data folder as the image. this objects are displayed in the graph and are updated by calling ad_calc_profiles(). see the code.

Parameters

image	wave which contains the image data from the detector.

Returns

(string) name of the graph window

Definition at line 115 of file pearl-area-display.ipf.

12.7.4.11 ad_display_profiles()

```
string ad_display_profiles (
            wave image,
            string filter = defaultValue )
```

open a new profiles graph window.

opens an extended graph window with profiles for the specified image. the function copies/creates all necessary data structures in a subfolder of the one which contains the image wave. the data folder name is derived from the

image wave name by prefixing with "view_". there can be at most one profiles window of each image wave. the original wave must not be renamed while the graph window is used. to update the graph after modifying the original wave, call ad_update_profiles().

Parameters

image	wave which contains the image data.
filter	name of a filter function which maps the original data to the displayed data. the function must have the same parameters as ad_default_image_filter(). default: boxcar average (ad_box_filter()) using parameters view_filter_smoothing_x and _y.

Returns

name of the graph window

Definition at line 166 of file pearl-area-display.ipf.

12.7.4.12 ad_display_slice()

```
string ad_display_slice (
            wave data )
```

display three-dimensional data by 2D slice.

to select the slice data to display, call ad_profiles_set_slice(), or open a ad_brick_slicer() panel. do not modify the content of the created view_ data folder.

Parameters

data	three-dimensional wave.

Returns

name of the graph window.

Definition at line 1404 of file pearl-area-display.ipf.

12.7.4.13 ad_export_profile()

```
variable ad_export_profile (
            wave view_image,
            variable dim,
            variable trace = defaultValue,
            variable show = defaultValue,
            variable overwrite = defaultValue )
```

export a profile from a profiles graph to the source data folder.

this function does not require that the show exists as long as the view data folder is complete.

Parameters

view_image	wave which contains the view image (image wave on display in profiles window). the function expects further objects as created by ad_display_profiles() in the same data folder as the view_image wave.
dim	dimension index (0 = x, 1 = y).
trace	select profile trace: • 0 = cursor A • 1 = cursor B • 2 = average between cursors (default)
show	display mode: • 0 = do not show (default) • 1 = display in new graph, or append to existing graph • 2 = collate: common graph for all profiles of a dimension. rename graph manually to detach it from future additions.
overwrite	overwrite mode: • 0 = create new wave (default). wave name may get a suffix to be unique. • 1 = overwrite existing wave

Definition at line 934 of file pearl-area-display.ipf.

12.7.4.14 ad_gizmo_set_plane()

```
variable ad_gizmo_set_plane (
             wave brick,
             variable dim,
             variable value )
```

set the position of a slicing plane of a 3D brick in a Gizmo window.

Parameters

brick	original data wave.
dim	dimension index: 0, 1, or 2.
value	new coordinate of the slicing plane (axis scaling).

Returns

0 if successful, non-zero otherwise

Definition at line 1481 of file pearl-area-display.ipf.

12.7.4.15 ad_profiles_crosshairs()

```
variable ad_profiles_crosshairs (
```

```
            wave image,
            variable clear = defaultValue )
```

draw permanent crosshairs in a profiles graph.

adds dash-dotted horizontal and vertical crosshairs to a profiles graph. for each active cursor A and/or B, a pair of lines crossing at the cursor position is added. existing crosshairs are moved to the current cursor positions. optionally, existing crosshairs are removed from the graph.

in contrast to the cursors, these crosshairs will be exported and printed with the graph. they are drawn using Igor's DrawLine operation. all lines drawn by this function are part of the "crosshairs" draw group.

Parameters

image	image displayed in the graph. this is the original image, not the one in the view data folder.
clear	0 (default) = add/update lines. 1 = remove lines.

Definition at line 540 of file pearl-area-display.ipf.

12.7.4.16 ad_profiles_cursor_mode()

```
variable ad_profiles_cursor_mode (
            wave image,
            variable mode )
```

switch cursors on a profiles graph

the standard cursors allow to select the profiles to display in the profiles panes. additional cursors are shown in the profiles panes.

in the background selection mode, additional cursors allow the user to select the limits of the background and peak integration regions. the meaning of the cursors depends on the particular processing function.

Parameters

mode	cursor mode. • 0 (default) standard profile selection. cursors C-F on profile panes. • 1 background selection. cursors A-F on image.
image	image displayed in the graph. this is the original image, not the one in the view data folder.

Definition at line 427 of file pearl-area-display.ipf.

12.7.4.17 ad_profiles_hook()

```
variable ad_profiles_hook (
            WMWinHookStruct * s )
```

hook function for user events in the profiles window.

Definition at line 745 of file pearl-area-display.ipf.

12.7.4.18 ad_profiles_set_cursor()

```
variable ad_profiles_set_cursor (
             wave image,
             string cursorname,
             variable xa,
             variable ya,
             variable pscale = defaultValue )
```

move a cursor to the specified position in a profiles graph.

this function can only set cursors in the image part of the profiles graph.

Parameters

image	image displayed in the graph. this is the original image, not the one in the view data folder.
cursorname	name of the cursor, e.g. "A" or "B". other cursors are allowed but need to be activated separately.
xa	x-coordinate to move the cursor to. the position is coerced to the image scale. +/-inf is allowed.
ya	y-coordinate to move the cursor to. the position is coerced to the image scale. +/-inf is allowed.
pscale	scaling of the position argument • 0 (default) wave scaling • 1 point scaling

Definition at line 488 of file pearl-area-display.ipf.

12.7.4.19 ad_profiles_set_slice()

```
variable ad_profiles_set_slice (
             wave brick,
             variable dim,
             variable value )
```

set the position of the slicing plane of a 3D brick in a profiles window.

Parameters

brick	original data wave.
dim	dimension index: 0, 1, or 2.
value	new coordinate of the slicing plane (axis scaling).

Returns

0 if successful, non-zero otherwise

Definition at line 1517 of file pearl-area-display.ipf.

12.7.4.20 ad_slicer_init_bg()

```
variable ad_slicer_init_bg ( )
```

initialize the slice animation background task.

Definition at line 1776 of file pearl-area-display.ipf.

12.7.4.21 ad_slicer_move_bg()

```
static variable ad_slicer_move_bg (
            WMBackgroundStruct * s )  [static]
```

move a slice by one step (background task).

Definition at line 1737 of file pearl-area-display.ipf.

12.7.4.22 ad_slicer_start_bg()

```
variable ad_slicer_start_bg (
            wave brick,
            variable dimension,
            string posvariable,
            variable delta )
```

start the animation.

Parameters

brick	3D data wave
dimension	dimension to animate, 0, 1, or 2.
posvariable	full path to the global position variable.
delta	step increment, should be +/- dimdelta.

Definition at line 1800 of file pearl-area-display.ipf.

12.7.4.23 ad_slicer_stop_bg()

```
variable ad_slicer_stop_bg (
            string posvariable )
```

stop the animation.

Parameters

posvariable	full path to the global position variable.

Definition at line 1842 of file pearl-area-display.ipf.

12.7.4.24 ad_transpose_filter()

```
variable ad_transpose_filter (
            wave image,
            string options )
```

transpose image filter.

transposes the image.

Parameters

image	image to be transposed: original data and result.
options	not used. should be empty.

Definition at line 1125 of file pearl-area-display.ipf.

12.7.4.25 ad_update_profiles()

```
variable ad_update_profiles (
             wave image )
```

update a profiles graph with new data.

Parameters

image	wave which contains the image data. must be the same (by data folder and name) wave used with ad_display_profiles().

Definition at line 375 of file pearl-area-display.ipf.

12.7.4.26 bp_extract_slice()

```
static variable bp_extract_slice (
             WMButtonAction * ba )  [static]
```

export a slice (button procedure).

extract a slice and saves it in a separate wave.

Definition at line 1671 of file pearl-area-display.ipf.

12.7.4.27 bp_move_slice()

```
static variable bp_move_slice (
             WMButtonAction * ba )  [static]
```

move slice (button procedure).

Definition at line 1622 of file pearl-area-display.ipf.

12.7.4.28 bp_move_slice_center()

```
static variable bp_move_slice_center (
             wave brick,
             variable dim,
             string posvariable )  [static]
```

move the slice to the center of the dimension (button procedure).

Definition at line 1725 of file pearl-area-display.ipf.

12.7.4.29 bp_reset_cursors()

```
static variable bp_reset_cursors (
            WMButtonAction * ba )  [static]
```

Definition at line 677 of file pearl-area-display.ipf.

12.7.4.30 get_source_image()

```
static wave get_source_image (
            wave view )  [static]
```

find the source image wave corresponding to the given view.

Returns

wave reference of the original data wave. the reference may be invalid if the source wave cannot be found.

Definition at line 606 of file pearl-area-display.ipf.

12.7.4.31 get_view_folder()

```
static dfr get_view_folder (
            wave source )  [static]
```

find the view data folder corresponding to the given source.

the result data folder reference may be invalid if no view is currently open. use the built-in DataFolderRefStatus function to check for validity.

Parameters

source	wave which contains the image data. must be the same (by data folder and name) wave used with ad_display_profiles().

Definition at line 644 of file pearl-area-display.ipf.

12.7.4.32 get_view_image()

```
static wave get_view_image (
            wave source )  [static]
```

find the view image wave corresponding to the given source.

Parameters

source	wave which contains the image data. must be the same (by data folder and name) wave used with ad_display_profiles().

Definition at line 667 of file pearl-area-display.ipf.

12.7.4.33 graphname_from_dfref()

```
static string graphname_from_dfref (
            dfref df,
            string prefix )  [static]
```

compose a valid and unique graph name from a data folder reference

Definition at line 57 of file pearl-area-display.ipf.

12.7.4.34 make_view_folder()

```
static dfr make_view_folder (
            wave source )  [static]
```

create a view data folder.

Definition at line 620 of file pearl-area-display.ipf.

12.7.4.35 pmp_export()

```
static variable pmp_export (
            WMPopupAction * pa )  [static]
```

Definition at line 721 of file pearl-area-display.ipf.

12.7.4.36 set_trace_colors()

```
static variable set_trace_colors (
            string graphname )  [static]
```

Definition at line 1045 of file pearl-area-display.ipf.

12.7.4.37 slp_slice_position()

```
static variable slp_slice_position (
            WMSliderAction * sa )  [static]
```

set slice coordinate (slider procedure).

Definition at line 1558 of file pearl-area-display.ipf.

12.7.4.38 svp_slice_position()

```
static variable svp_slice_position (
            WMSetVariableAction * sva )  [static]
```

set slice coordinate (button procedure).

Definition at line 1590 of file pearl-area-display.ipf.

12.7.4.39 svp_smoothing()

```
static variable svp_smoothing (
            WMSetVariableAction * sva )  [static]
```

Definition at line 696 of file pearl-area-display.ipf.

12.7.4.40 update_slice_info()

```
static variable update_slice_info ( )  [static]
```

update controls with data scale limits.

current folder must be slicer info

Definition at line 1445 of file pearl-area-display.ipf.

12.8 pearl-area-import.ipf File Reference

HDF5 file import from EPICS area detectors.

```
#include <HDF5 Browser>
#include "pearl-compat"
#include "pearl-gui-tools"
```

Namespaces

- PearlAreaImport

  HDF5 file import from EPICS area detectors.

Functions

- static variable BeforeFileOpenHook (variable refNum, string fileName, string path, string type, string creator, variable kind)

  callback function for drag&drop of HDF5 files into Igor.
- string ad_suggest_foldername (string filename, variable ignoredate=defaultValue, string sourcename=default↩ Value, variable unique=defaultValue)

  generate the name of a data folder based on a file name.
- variable ad_load_dialog (string APathName)

  load area detector data files selected in a file dialog window
- string adh5_load_complete (string ANickName, string APathName, string AFileName, variable load_↩ data=defaultValue, variable load_attr=defaultValue)

  import everything from a HDF5 file created by the Area Detector software.
- string adh5_load_reduced (string ANickName, string APathName, string AFileName, funcref reduction↩ _func, string reduction_param, variable load_data=defaultValue, variable load_attr=defaultValue, variable progress=defaultValue)

  load and reduce a dataset from a HDF5 file created by the Area Detector software.

- string adh5_load_preview (string ANickName, string APathName, string AFileName, variable load_↩ data=defaultValue, variable load_attr=defaultValue)

  load a single image from a HDF5 file created by the Area Detector software.
- string adh5_load_info (string APathName, string AFileName)

  load descriptive info from a HDF5 file created by the Area Detector software.
- variable adh5_load_detector (variable fileID, string detectorpath)

  load the detector dataset from the open HDF5 file.
- variable adh5_redim (wave data)

  redimension a multi-dimensional area detector array loaded from HDF5.
- static dfr GetAttrDataFolderDFR (wave data)

  find the attributes data folder of an area detector dataset.
- variable adh5_scale (wave data, string source=defaultValue)

  set the dimension scales of an area detector dataset.
- variable adh5_load_detector_slabs (variable fileID, string detectorpath, variable progress=defaultValue)

  load the detector dataset from the open HDF5 file.
- variable adh5_load_detector_image (variable fileID, string detectorpath, variable dim2start, variable dim2count, variable dim3start, variable dim3count)

  load a single image from the detector dataset of the open HDF5 file
- string adh5_list_reduction_funcs ()

  get a list of functions which can be used as reduction functions.
- threadsafe wave adh5_default_reduction (wave source, string *param)

  function prototype for adh5_load_reduced_detector
- threadsafe variable adh5_setup_profile (wave image, wave profile, variable dim)

  set up a one-dimensional wave for a line profile based on a 2D original wave.
- string adh5_test_reduction_func (wave source, funcref reduction_func, string reduction_param, string result↩ _prefix)

  wrapper function for testing reduction functions from the command line.
- variable adh5_reduce_brick (wave source, funcref reduction_func, string reduction_param, string result_↩ prefix, variable progress=defaultValue, variable nthreads=defaultValue)

  reduce a three-dimensional intensity distribution
- static threadsafe variable reduce_brick_worker (funcref reduction_func)

  thread worker for adh5_reduce_brick
- threadsafe variable adh5_get_result_waves (wave results, string result_prefix, variable start_index)

  copy waves from wave reference wave into current data folder
- variable adh5_load_reduced_detector (variable fileID, string detectorpath, funcref reduction_func, string reduction_param, variable progress=defaultValue, variable nthreads=defaultValue)

  load a reduced detector dataset from the open HDF5 file.
- static threadsafe variable reduce_slab_worker (funcref reduction_func)
- static threadsafe wave reduce_slab_image (wave slabdata, wave image, funcref reduction_func, string reduction_param)
- variable adh5_loadattr_all (variable fileID, string attributespath)

  load an NDAttributes group from an open HDF5 file into the current data folder.
- static variable read_attribute_info (string datawavename, string source, variable *idest)

  sub-function of adh5_loadattr_all.
- variable adh5_scale_scienta (wave data)

  set the energy and angle scales of an area detector dataset from the Scienta analyser.
- variable adh5_scale_scan (wave data)

  scales the extra dimensions of an area detector dataset according to the EPICS scan

12.8.1 Detailed Description

HDF5 file import from EPICS area detectors.

HDF5 file import from EPICS area detectors such as CCD cameras, 2D electron analysers

as of Igor 6.3, Igor can open datasets of up to rank 4. i.e. the extra dimension Y of the file plugin cannot be used. the extra dimensions N and X are supported.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.8.2 Function Documentation

12.8.2.1 ad_load_dialog()

```
variable ad_load_dialog (
            string APathName )
```

load area detector data files selected in a file dialog window

Parameters

APathName	Igor symbolic path name. if empty, Igor will choose a folder on its own

Definition at line 167 of file pearl-area-import.ipf.

12.8.2.2 ad_suggest_foldername()

```
string ad_suggest_foldername (
            string filename,
            variable ignoredate = defaultValue,
            string sourcename = defaultValue,
            variable unique = defaultValue )
```

generate the name of a data folder based on a file name.

if the file name follows the naming convention source-date-index.extension, the function tries to generate the nick name as source_date_index. otherwise it's just a cleaned up version of the file name.

date must be in yyyymmdd or yymmdd format and is clipped to the short yymmdd format. index should be a running numeric index of up to 6 digits, or the time encoded as hhmmss. however, in the current version index can be any string that can be a valid Igor folder name.

Parameters

filename	file name, including extension. can also include a folder path (which is ignored). the extension is currently ignored, but may be used to select the parent folder in a later version.
ignoredate	if non-zero, the nick name will not include the date part. defaults to zero.
sourcename	nick name of the data source. by default, the function tries to detect the source from the file name. this option can be used to override auto-detection. the automatic source names are: sci (scienta by area detector), psh (pshell), sl (optics slit camera by area detector), es (end station camera by area detector), xy (unidentified).
unique	if non-zero, the resulting name is made a unique data folder name in the current data folder defaults to zero.

Definition at line 106 of file pearl-area-import.ipf.

12.8.2.3 adh5_default_reduction()

```
threadsafe wave adh5_default_reduction (
            wave source,
            string * param )
```

function prototype for adh5_load_reduced_detector

this is a prototype of custom functions that convert (reduce) a two-dimensional detector image into one or more one-dimensional waves. data processing can be tuned with a set of parameters.

reduction functions have a fixed signature (function arguments) so that the file import functions can call them efficiently on a series of detector images. pearl procedures comes with a number of pre-defined reduction functions but you may as well implement your own functions. if you write your own function, you must use the same declaration and arguments as this function except for the function name. you can do many things in a reduction function, e.g. integration over a region of interest, curve fitting, etc.

each destination wave is a one-dimensional intensity distribution. the function must redimension each of these waves to one of the image dimensions by calling the adh5_setup_profile() function. this function will also copy the scale information and dimension labels, which is important for the proper scaling of the result.

the meaning of the data in the result waves is up to the particular function, e.g. dest1 could hold the mean value and dest2 the one-sigma error, or dest1 could hold the X-profile, and dest2 the Y-profile.

Parameters

source	source wave. two-dimensional intensity distribution (image). the scales are carried over to the result waves.
param	string with optional parameters, shared between calls. this is a pass-by-reference argument, the function may modify the string.

Returns

a free wave containing references of the result waves. the result waves should as well be free waves. if an error occurred, the reference wave is empty.

Definition at line 1111 of file pearl-area-import.ipf.

12.8.2.4 adh5_get_result_waves()

```
threadsafe variable adh5_get_result_waves (
```

```
            wave results,
            string result_prefix,
            variable start_index )
```

copy waves from wave reference wave into current data folder

this function copies waves that are referenced in a wave reference wave into the current data folder. the destination waves get new names consisting of a prefix and a numeric index. the index is the array index of the wave in results plus a chosen offset.

Parameters

results	a wave reference wave pointing to result waves from data reduction. the waves can be free or regular waves. results can be a free or regular wave.
result_prefix	name prefix of the copied waves.
start_index	start index (offset) of the copied waves.

Definition at line 1430 of file pearl-area-import.ipf.

12.8.2.5 adh5_list_reduction_funcs()

```
string adh5_list_reduction_funcs ( )
```

get a list of functions which can be used as reduction functions.

the function evaluates only the function arguments, it may thus include functions which are not suitable as reduction functions.

Definition at line 1041 of file pearl-area-import.ipf.

12.8.2.6 adh5_load_complete()

```
string adh5_load_complete (
            string ANickName,
            string APathName,
            string AFileName,
            variable load_data = defaultValue,
            variable load_attr = defaultValue )
```

import everything from a HDF5 file created by the Area Detector software.

if the data is from the electron analyser driver and some special attributes are included, the function will set the scales of the image dimensions.

Parameters

ANickName	destination folder name (top level under root)
APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up
load_data	1 (default): load data; 0: do not load data
load_attr	1 (default): load attributes; 0: do not load attributes for proper wave scaling, the attributes must be loaded

Definition at line 207 of file pearl-area-import.ipf.

12.8.2.7 adh5_load_detector()

```
variable adh5_load_detector (
            variable fileID,
            string detectorpath )
```

load the detector dataset from the open HDF5 file.

the function loads the whole dataset at once and redimensions it so that the image dimensions are X and Y

Parameters

fileID	ID of open HDF5 file from HDF5OpenFile
detectorpath	path to detector group in the HDF5 file

Definition at line 581 of file pearl-area-import.ipf.

12.8.2.8 adh5_load_detector_image()

```
variable adh5_load_detector_image (
            variable fileID,
            string detectorpath,
            variable dim2start,
            variable dim2count,
            variable dim3start,
            variable dim3count )
```

load a single image from the detector dataset of the open HDF5 file

the function can average over a region in the extra dimensions.

Parameters

fileID	ID of open HDF5 file from HDF5OpenFile
detectorpath	path to detector group in the HDF5 file
dim2start	2nd dimension coordinate of the first image note that the order of dimensions is reversed in the file 2nd dimension = N dimension in area detector = dimension 0 of the three-dimensional HDF dataset set to 0 if dimension may not be present
dim2count	number of subsequent images to average set to 1 if dimension may not be present
dim3start	3rd dimension coordinate of the first image note that the order of dimensions is reversed in the file 3rd dimension = extra X dimension in area detector = dimension 0 of the four-dimensional HDF dataset set to 0 if dimension may not be present
dim3count	number of subsequent images to average set to 1 if dimension may not be present

Definition at line 945 of file pearl-area-import.ipf.

12.8.2.9 adh5_load_detector_slabs()

```
variable adh5_load_detector_slabs (
            variable fileID,
```

```
            string detectorpath,
            variable progress = defaultValue )
```

load the detector dataset from the open HDF5 file.

the function loads the dataset image by image using the hyperslab option. this function gives the same result as adh5_load_detector. it is about 5% slower, and it depends on HDF5 Browser code. but it does not choke on large datasets (as long as the final wave fits into memory).

Parameters

fileID	ID of open HDF5 file from HDF5OpenFile.
detectorpath	path to detector group in the HDF5 file.
progress	1 (default): show progress window; 0: do not show progress window.

Returns

0 if successful, non-zero if an error occurred.

Definition at line 780 of file pearl-area-import.ipf.

12.8.2.10 adh5_load_info()

```
string adh5_load_info (
            string APathName,
            string AFileName )
```

load descriptive info from a HDF5 file created by the Area Detector software.

the information returned is the array size and active scans

Attention

EXPERIMENTAL this function should be merged with adh5_load_preview

Parameters

APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up

Definition at line 497 of file pearl-area-import.ipf.

12.8.2.11 adh5_load_preview()

```
string adh5_load_preview (
            string ANickName,
            string APathName,
            string AFileName,
            variable load_data = defaultValue,
            variable load_attr = defaultValue )
```

load a single image from a HDF5 file created by the Area Detector software.

the data wave is loaded into the current data folder. attributes are loaded into the attr subfolder. existing waves in attr are deleted.

Warning

EXPERIMENTAL this function uses the root:pearl_area:preview data folder. existing data there may be deleted!

Parameters

ANickName	destination wave name. the wave is created in the current data folder.
APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up
load_data	1 (default): load data; 0: do not load data
load_attr	1 (default): load attributes; 0: do not load attributes note: for correct scaling of the image, the attributes need to be loaded

Definition at line 387 of file pearl-area-import.ipf.

12.8.2.12 adh5_load_reduced()

```
string adh5_load_reduced (
             string ANickName,
             string APathName,
             string AFileName,
             funcref reduction_func,
             string reduction_param,
             variable load_data = defaultValue,
             variable load_attr = defaultValue,
             variable progress = defaultValue )
```

load and reduce a dataset from a HDF5 file created by the Area Detector software.

the resulting dataset is reduced in one image dimension by a user-defined reduction function, e.g. by region-of-interest integration, curve fitting, etc.

the function loads the dataset image by image using the hyperslab option and applies a custom reduction function to each image. the results from the reduction function are composed into one result wave. the raw data are discarded.

if the data is from the electron analyser driver and some special attributes are included, the function will set the scales of the image dimensions.

Parameters

ANickName	destination folder name (top level under root)
APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up
reduction_func	custom reduction function (any user-defined function which has the same parameters as adh5_default_reduction())
reduction_param	parameter string for the reduction function
load_data	1 (default): load data; 0: do not load data
load_attr	1 (default): load attributes; 0: do not load attributes for proper wave scaling, the attributes must be loaded
progress	1 (default): show progress window; 0: do not show progress window

Definition at line 302 of file pearl-area-import.ipf.

12.8.2.13 adh5_load_reduced_detector()

```
variable adh5_load_reduced_detector (
             variable fileID,
             string detectorpath,
             funcref reduction_func,
             string reduction_param,
             variable progress = defaultValue,
             variable nthreads = defaultValue )
```

load a reduced detector dataset from the open HDF5 file.

the function loads the dataset image by image using the hyperslab option and applies a custom reduction function to each image. the results from the reduction function are composed into one result wave. the raw data are discarded.

by default, the reduction function is called in separate threads to reduce the total loading time. (see the global variable adh5_perf_secs which reports the total run time of the function.) the effect varies depending on the balance between file loading (image size) and data processing (complexity of the reduction function). for debugging the reduction function, multi-threading can be disabled.

Parameters

fileID	ID of open HDF5 file from HDF5OpenFile
detectorpath	path to detector group in the HDF5 file
reduction_func	custom reduction function (any user-defined function which has the same parameters as adh5_default_reduction())
reduction_param	parameter string for the reduction function
progress	1 (default): show progress window; 0: do not show progress window
nthreads	-1 (default): use as many threads as there are processor cores (in addition to main thread) 0: use main thread only (e.g. for debugging the reduction function) >= 1: use a fixed number of (additional) threads

Definition at line 1466 of file pearl-area-import.ipf.

12.8.2.14 adh5_loadattr_all()

```
variable adh5_loadattr_all (
             variable fileID,
             string attributespath )
```

load an NDAttributes group from an open HDF5 file into the current data folder.

datasets contained in the group are loaded as waves. if a dataset contains only one data point, it is added to the IN, ID, IV, IU waves, where IN = EPICS channel name, ID = attribute name, IV = value, IU = unit (units are left empty as they are not saved in HDF5). attributes of the NDAttributes group are added to the IN, ID, IV, IU waves, however, IN and IU are left empty as this information is not saved in the HDF5 file.

Parameters

fileID	ID of open HDF5 file from HDF5OpenFile
attributespath	path to NDAttributes group in the HDF5 file

Definition at line 1790 of file pearl-area-import.ipf.

12.8.2.15 adh5_redim()

```
variable adh5_redim (
            wave data )
```

redimension a multi-dimensional area detector array loaded from HDF5.

so that the image dimensions are X and Y singleton dimensions are removed (most common cases only)

in the redimensioned array, the original dimension type is noted in the dimension label: AD_Dim0 = first image dimension AD_Dim1 = second image dimension AD_DimN = frame sequence AD_DimX = extra dimension X AD↩ _DimY = extra dimension Y (cannot be loaded in Igor)

Parameters

data	area detector data loaded from HDF5 to be redimensioned

Definition at line 626 of file pearl-area-import.ipf.

12.8.2.16 adh5_reduce_brick()

```
variable adh5_reduce_brick (
            wave source,
            funcref reduction_func,
            string reduction_param,
            string result_prefix,
            variable progress = defaultValue,
            variable nthreads = defaultValue )
```

reduce a three-dimensional intensity distribution

this function reduces a three-dimensional intensity distribution to a two-dimensional intensity map. the given reduction function is applied once on each Z section.

Parameters

source	source wave. three-dimensional intensity distribution (image). the scales are carried over to the result waves.
reduction_func	name of the reduction function to apply to the source data.
reduction_param	string with reduction parameters as required by the specific reduction function.
result_prefix	name prefix of result waves. a numeric index is appended to distinguish the results. the index starts at 1. existing waves are overwritten.

Definition at line 1198 of file pearl-area-import.ipf.

12.8.2.17 adh5_scale()

```
variable adh5_scale (
            wave data,
            string source = defaultValue )
```

set the dimension scales of an area detector dataset.

the intrinsic dimensions 0 and 1 are scaled according to the data source (currently supported: Prosilica cameras, Scienta electron analyser). the extra dimensions are scaled according to the scan. the latter requires that the positioner names and position values are available.

Definition at line 726 of file pearl-area-import.ipf.

12.8.2.18 adh5_scale_scan()

```
variable adh5_scale_scan (
             wave data )
```

scales the extra dimensions of an area detector dataset according to the EPICS scan

the scan positioner name and its values must be available

Todo incomplete

Definition at line 2030 of file pearl-area-import.ipf.

12.8.2.19 adh5_scale_scienta()

```
variable adh5_scale_scienta (
             wave data )
```

set the energy and angle scales of an area detector dataset from the Scienta analyser.

the dimension labels of the energy and angle scales must be set correctly: AD_Dim0 = energy dimension; AD_Dim1 = angle dimension. these dimensions must be the first two dimensions of a multi-dimensional dataset. normally, AD_Dim0 is the X dimension, and AD_Dim1 the Y dimension.

Definition at line 1925 of file pearl-area-import.ipf.

12.8.2.20 adh5_setup_profile()

```
threadsafe variable adh5_setup_profile (
             wave image,
             wave profile,
             variable dim )
```

set up a one-dimensional wave for a line profile based on a 2D original wave.

redimensions the profile wave to the given dimension. copies the scale and dimension label of the given dimension.

Definition at line 1134 of file pearl-area-import.ipf.

12.8.2.21 adh5_test_reduction_func()

```
string adh5_test_reduction_func (
             wave source,
             funcref reduction_func,
             string reduction_param,
             string result_prefix )
```

wrapper function for testing reduction functions from the command line.

reduction functions cannot be used on the command line because they require a pass-by-reference argument and return free waves. this function expects the reduction parameters in a normal string and copies the results into the current data folder. the prefix of the result names can be specified.

Parameters

source	source wave. two-dimensional intensity distribution (image). the scales are carried over to the result waves.
reduction_func	name of the reduction function to apply to the source data.
reduction_param	string with reduction parameters as required by the specific reduction function.
result_prefix	name prefix of result waves. a numeric index is appended to distinguish the results. the index starts at 1. existing waves are overwritten.

Returns

a copy of the reduction_param string, possibly modified by the reduction function.

Definition at line 1167 of file pearl-area-import.ipf.

12.8.2.22 BeforeFileOpenHook()

```
static variable BeforeFileOpenHook (
             variable refNum,
             string fileName,
             string path,
             string type,
             string creator,
             variable kind )  [static]
```

callback function for drag&drop of HDF5 files into Igor.

Definition at line 42 of file pearl-area-import.ipf.

12.8.2.23 GetAttrDataFolderDFR()

```
static dfr GetAttrDataFolderDFR (
             wave data )  [static]
```

find the attributes data folder of an area detector dataset.

since version 1.04 attributes should be stored in a subfolder named attr. earlier versions had the attributes in the same data folder as the actual dataset.

Parameters

data	wave containing the main dataset.

Returns

data folder reference of the attributes folder. the reference may be invalid (and default to root) if the folder cannot be found, cf. built-in DataFolderRefStatus function.

Definition at line 707 of file pearl-area-import.ipf.

12.8.2.24 read_attribute_info()

```
static variable read_attribute_info (
             string datawavename,
             string source,
             variable * idest )  [static]
```

sub-function of adh5_loadattr_all.

reads one attribute from a wave which was loaded from an HDF5 file into the info waves IN, ID, IV, IU. the attribute is read only if the input wave contains exactly one item, i.e. either the measurement is a single image, or the attribute has string type.

Parameters

datawavename	name of the attribute wave in the current folder. can be text or numeric.
source	source identifier (EPICS name) of the attribute.
idest	destination index in IN, ID, IV, IU where the results are written. the variable is incremented if data was written, otherwise it is left unchanged. make sure IN, ID, IV, IU have at least idest + 1 elements.

Definition at line 1872 of file pearl-area-import.ipf.

12.8.2.25 reduce_brick_worker()

```
static threadsafe variable reduce_brick_worker (
             funcref reduction_func )  [static]
```

thread worker for adh5_reduce_brick

this function polls job data folders from thread group 0 and calls the reduction function on their contents. the result waves have prefix "redw_" and are saved in the job folder.

Definition at line 1377 of file pearl-area-import.ipf.

12.8.2.26 reduce_slab_image()

```
static threadsafe wave reduce_slab_image (
             wave slabdata,
             wave image,
             funcref reduction_func,
             string reduction_param )  [static]
```

Definition at line 1757 of file pearl-area-import.ipf.

12.8.2.27 reduce_slab_worker()

```
static threadsafe variable reduce_slab_worker (
             funcref reduction_func )  [static]
```

Definition at line 1718 of file pearl-area-import.ipf.

12.9 pearl-area-profiles.ipf File Reference

profile extraction for multi-dimensional datasets acquired from area detectors.

Namespaces

- PearlAreaProfiles

  profile extraction for multi-dimensional datasets acquired from area detectors.

Functions

- threadsafe wave ad_extract_rod (wave dataset, variable x1, variable x2, variable y1, variable y2, variable z1, variable z2, string destname, variable noavg=defaultValue, variable sdev=defaultValue, variable pscale=defaultValue)

  1D cut through 3D dataset, integrate in normal dimensions
- threadsafe wave ad_extract_rod_x (wave dataset, variable q1, variable q2, variable r1, variable r2, string destname, variable noavg=defaultValue, variable sdev=defaultValue)

  1D cut through 3D dataset along X dimension.
- threadsafe wave ad_extract_rod_y (wave dataset, variable p1, variable p2, variable r1, variable r2, string destname, variable noavg=defaultValue, variable sdev=defaultValue)

  1D cut through 3D dataset along Y dimension.
- threadsafe wave ad_extract_rod_z (wave dataset, variable p1, variable p2, variable q1, variable q2, string destname, variable noavg=defaultValue, variable sdev=defaultValue)

  1D cut through 3D dataset along Z dimension.
- threadsafe wave ad_extract_slab (wave dataset, variable x1, variable x2, variable y1, variable y2, variable z1, variable z2, string destname, variable noavg=defaultValue, variable pscale=defaultValue)

  2D cut through 3D dataset, integrate in normal dimension
- threadsafe wave ad_extract_slab_x (wave dataset, variable p1, variable p2, string destname, variable noavg=defaultValue)
- threadsafe wave ad_extract_slab_y (wave dataset, variable q1, variable q2, string destname, variable noavg=defaultValue)
- threadsafe wave ad_extract_slab_z (wave dataset, variable r1, variable r2, string destname, variable noavg=defaultValue)
- threadsafe wave ad_profile_x (wave dataset, variable q1, variable q2, string destname, variable noavg=defaultValue)

  1D cut through 2D dataset along X dimension, new destination wave.
- threadsafe wave ad_profile_x_w (wave dataset, variable q1, variable q2, wave destwave, variable noavg=defaultValue)

  1D cut through 2D dataset along X dimension, existing destination wave.
- threadsafe wave ad_profile_y (wave dataset, variable p1, variable p2, string destname, variable noavg=defaultValue)

  1D cut through 2D dataset along Y dimension, new destination wave.
- threadsafe wave ad_profile_y_w (wave dataset, variable p1, variable p2, wave destwave, variable noavg=defaultValue)

  1D cut through 2D dataset along X dimension, existing destination wave.
- threadsafe variable calc_y_profile_mins (wave image)
- variable ad_collect_multiscan_y (wave dataset, wave positions, wave destwave, variable noavg=defaultValue)

  collect profiles from a multi-scan.

12.9.1 Detailed Description

profile extraction for multi-dimensional datasets acquired from area detectors.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.9.2 Function Documentation

12.9.2.1 ad_collect_multiscan_y()

```
variable ad_collect_multiscan_y (
             wave dataset,
             wave positions,
             wave destwave,
             variable noavg = defaultValue )
```

collect profiles from a multi-scan.

Warning

experimental: name and interface of this function may change.

Definition at line 621 of file pearl-area-profiles.ipf.

12.9.2.2 ad_extract_rod()

```
threadsafe wave ad_extract_rod (
             wave dataset,
             variable x1,
             variable x2,
             variable y1,
             variable y2,
             variable z1,
             variable z2,
             string destname,
             variable noavg = defaultValue,
             variable sdev = defaultValue,
             variable pscale = defaultValue )
```

1D cut through 3D dataset, integrate in normal dimensions

Parameters

dataset	

Parameters

x1,x2,y1,y2,z1,z2	coordinates of integration region by default, the coordinates use wave scaling coordinates of rod dimensions (to be preserved) must be nan coordinate pairs don't have to be ordered, i.e. both x1 <= x2 and x1 >= x2 are allowed.
destname	name of destination wave. to be created in current data folder. if empty, the function returns a free wave
noavg	non-zero = calculate the sum, default = 0 as of version 1.05, this option should rather be called "calc_sum" or similar. it is noavg for compatibility with older code.
sdev	non-zero = calculate the standard deviation, default = 0 by default, the function calculates the average of the integration region set either the noavg or sdev option to select the sum or the standard deviation, respectively. if both options are set, noavg (sum) takes precedence.
pscale	scaling of the slab coordinates x1, x2, ..., z2: zero or default = wave scaling, non-zero = point scaling

Remarks

- version 1.02: the specification of the destination coordinates has changed
- version 1.04: the function returns an empty wave reference if an error occurred

Definition at line 54 of file pearl-area-profiles.ipf.

12.9.2.3 ad_extract_rod_x()

```
threadsafe wave ad_extract_rod_x (
             wave dataset,
             variable q1,
             variable q2,
             variable r1,
             variable r2,
             string destname,
             variable noavg = defaultValue,
             variable sdev = defaultValue )
```

1D cut through 3D dataset along X dimension.

see ad_extract_rod() for descriptions of common parameters.

Definition at line 106 of file pearl-area-profiles.ipf.

12.9.2.4 ad_extract_rod_y()

```
threadsafe wave ad_extract_rod_y (
             wave dataset,
             variable p1,
             variable p2,
             variable r1,
             variable r2,
             string destname,
             variable noavg = defaultValue,
             variable sdev = defaultValue )
```

1D cut through 3D dataset along Y dimension.

see ad_extract_rod() for descriptions of common parameters.

Definition at line 168 of file pearl-area-profiles.ipf.

12.9.2.5 ad_extract_rod_z()

```
threadsafe wave ad_extract_rod_z (
             wave dataset,
             variable p1,
             variable p2,
             variable q1,
             variable q2,
             string destname,
             variable noavg = defaultValue,
             variable sdev = defaultValue )
```

1D cut through 3D dataset along Z dimension.

see ad_extract_rod() for descriptions of common parameters.

Definition at line 231 of file pearl-area-profiles.ipf.

12.9.2.6 ad_extract_slab()

```
threadsafe wave ad_extract_slab (
             wave dataset,
             variable x1,
             variable x2,
             variable y1,
             variable y2,
             variable z1,
             variable z2,
             string destname,
             variable noavg = defaultValue,
             variable pscale = defaultValue )
```

2D cut through 3D dataset, integrate in normal dimension

Parameters

dataset	
x1,x2,y1,y2,z1,z2	coordinates of integration region. by default, the coordinates use wave scaling. coordinates of slab dimensions (to be preserved) must be nan. coordinate pairs don't have to be ordered, i.e. both x1 <= x2 and x1 >= x2 are allowed. coordinates can be out of range (-inf and +inf allowed) to select the whole range.
destname	name of destination wave. to be created in current data folder. if empty, the function returns a free wave.
noavg	zero or default = average, non-zero = sum.
pscale	scaling of the slab coordinates x1, x2, ..., z2: zero or default = wave scaling, non-zero = point scaling.

Remarks

- version 1.02: the specification of the destination coordinates has changed
- version 1.04: the function returns an empty wave reference if an error occurred

Definition at line 313 of file pearl-area-profiles.ipf.

12.9.2.7 ad_extract_slab_x()

```
threadsafe wave ad_extract_slab_x (
             wave dataset,
             variable p1,
             variable p2,
             string destname,
             variable noavg = defaultValue )
```

Definition at line 358 of file pearl-area-profiles.ipf.

12.9.2.8 ad_extract_slab_y()

```
threadsafe wave ad_extract_slab_y (
             wave dataset,
             variable q1,
             variable q2,
             string destname,
             variable noavg = defaultValue )
```

Definition at line 398 of file pearl-area-profiles.ipf.

12.9.2.9 ad_extract_slab_z()

```
threadsafe wave ad_extract_slab_z (
             wave dataset,
             variable r1,
             variable r2,
             string destname,
             variable noavg = defaultValue )
```

Definition at line 438 of file pearl-area-profiles.ipf.

12.9.2.10 ad_profile_x()

```
threadsafe wave ad_profile_x (
             wave dataset,
             variable q1,
             variable q2,
             string destname,
             variable noavg = defaultValue )
```

1D cut through 2D dataset along X dimension, new destination wave.

Definition at line 480 of file pearl-area-profiles.ipf.

12.9.2.11 ad_profile_x_w()

```
threadsafe wave ad_profile_x_w (
             wave dataset,
             variable q1,
             variable q2,
             wave destwave,
             variable noavg = defaultValue )
```

1D cut through 2D dataset along X dimension, existing destination wave.

Definition at line 504 of file pearl-area-profiles.ipf.

12.9.2.12 ad_profile_y()

```
threadsafe wave ad_profile_y (
             wave dataset,
             variable p1,
             variable p2,
             string destname,
             variable noavg = defaultValue )
```

1D cut through 2D dataset along Y dimension, new destination wave.

Definition at line 542 of file pearl-area-profiles.ipf.

12.9.2.13 ad_profile_y_w()

```
threadsafe wave ad_profile_y_w (
             wave dataset,
             variable p1,
             variable p2,
             wave destwave,
             variable noavg = defaultValue )
```

1D cut through 2D dataset along X dimension, existing destination wave.

Definition at line 567 of file pearl-area-profiles.ipf.

12.9.2.14 calc_y_profile_mins()

```
threadsafe variable calc_y_profile_mins (
             wave image )
```

Definition at line 603 of file pearl-area-profiles.ipf.

12.10 pearl-arpes.ipf File Reference

data acquisition and analysis package for ARPES at PEARL.

```
#include "pearl-area-display"
#include "pearl-area-profiles"
#include "pearl-area-import"
#include "pearl-pshell-import"
```

```
#include "pearl-compat"
#include "pearl-data-explorer"
#include "pearl-anglescan-process"
#include "pearl-anglescan-panel"
#include "pearl-anglescan-tracker"
#include "pearl-scienta-preprocess"
#include "pearl-elog"
```

Namespaces

- PearlArpes

  data acquisition and analysis package for ARPES at PEARL.

Functions

- static variable AfterCompiledHook ()

  initializes package data once when the procedure is first loaded
- variable UnloadPearlArpesPackage ()

12.10.1 Detailed Description

data acquisition and analysis package for ARPES at PEARL.

this procedure defines the PEARL ARPES package the main purpose of this file is to load the necessary dependent procedures (see the include statements at the top)

Precondition

- HDF5 XOP must be loaded.
- on-line data acquisition functionality requires the EPICS XOP to be loaded

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.10.2 Function Documentation

12.10.2.1 AfterCompiledHook()

```
static variable AfterCompiledHook ( )  [static]
```

initializes package data once when the procedure is first loaded

Definition at line 77 of file pearl-arpes.ipf.

12.10.2.2 UnloadPearlArpesPackage()

```
variable UnloadPearlArpesPackage ( )
```

Definition at line 84 of file pearl-arpes.ipf.

12.11 pearl-compat.ipf File Reference

compatibility procedures for igor 8

Namespaces

- PearlCompat

  compatibility procedures for igor 8

Functions

- string PearlCleanupName (string name)

12.11.1 Detailed Description

compatibility procedures for igor 8

the compatibility procedures ensure that igor experiments created with the PEARL procedures under igor 8 can be opened with earlier igor versions (>= 6.34).

the following possible issues are addressed:

- length of object names

12.11.2 Function Documentation

12.11.2.1 PearlCleanupName()

```
string PearlCleanupName (
            string name )
```

Definition at line 47 of file pearl-compat.ipf.

12.12 pearl-data-explorer.ipf File Reference

preview and import panel for PEARL data

```
#include "pearl-area-import"
#include "pearl-area-profiles"
#include "pearl-area-display"
#include "pearl-compat"
#include "pearl-pshell-import"
```

Namespaces

- PearlDataExplorer

  preview and import panel for PEARL data

Functions

- variable pearl_data_explorer ()
- static variable init_package ()

  initialize the global variables of the data explorer.
- static variable save_prefs ()

  save persistent package data to the preferences file.
- static variable load_prefs ()
- static variable pearl_file_type (string filename)

  check whether a file can be imported by this module.
- static variable update_filelist ()

  read a list of PEARL files from the file system
- static variable update_datasets ()
- static variable preview_file (string filename)
- static wave preview_pshell_file (string filename)

  load the preview of a PShell HDF5 file.
- static wave preview_hdf_file (string filename)

  load the preview of a PEARL HDF5 file.
- static wave preview_itx_file (string filename)

  load the preview of a general ITX file.
- static wave preview_mtrx_file (string filename)

  load the preview of a Matrix STM file.
- static variable extract_preview_image (wave data, wave preview)
- static variable preview_dataset (string datasetname)
- static variable preview_datafolder ()
- static variable preview_setscale_x (wave data, wave preview)
- static variable preview_attributes (dfref attr_folder, dfref dest_folder=defaultValue, wave attr_filter=default↩ Value, variable include_datawaves=defaultValue, variable include_infowaves=defaultValue)
- static variable display_dataset (string datasetname)
- variable test_attributes_notebook ()
- static variable attributes_notebook (wave attr_names, wave attr_values, string title)
- static variable notebook_add_attributes (string notebook_name, wave attr_filter, wave attr_names, wave attr_values)
- static string show_preview_graph (wave data, wave xdata=defaultValue)
- static string display_preview_trace (wave xtrace, wave ytrace)
- static variable load_selected_files (string options=defaultValue)
- static variable load_file (string filename, string options=defaultValue)
- static variable prompt_hdf_options (string ∗options)
- variable prompt_default_process (string ∗param)

  prototype for prompting for processing function parameters.
- variable prompt_func_params (string func_name, string ∗func_param)
- static dfr load_pshell_file (string filename, string options=defaultValue)
- static dfr load_hdf_file (string filename, string options=defaultValue)
- static dfr load_itx_file (string filename, string options=defaultValue)
- static dfr load_mtrx_file (string filename, string options=defaultValue)

  load a matrix (STM) data file
- string itx_suggest_foldername (string filename, variable ignoredate=defaultValue, string sourcename=default↩ Value, variable unique=defaultValue)

- void PearlDataExplorer ()
- static variable bp_load_prefs (WMButtonAction *ba)
- static variable bp_save_prefs (WMButtonAction *ba)
- static variable bp_browse_filepath (WMButtonAction *ba)
- static variable bp_update_filelist (WMButtonAction *ba)
- static variable bp_load_files (WMButtonAction *ba)
- static variable bp_load_files_opt (WMButtonAction *ba)
- static variable bp_file_next (WMButtonAction *ba)
- static variable bp_file_prev (WMButtonAction *ba)
- static variable lbp_filelist (WMListboxAction *lba)
- static variable bp_update_datasets (WMButtonAction *ba)
- static variable bp_dataset_folder (WMButtonAction *ba)
- static variable bp_dataset_display (WMButtonAction *ba)
- static variable bp_dataset_next (WMButtonAction *ba)
- static variable bp_dataset_prev (WMButtonAction *ba)
- static variable lbp_datasets (WMListboxAction *lba)
- static variable bp_attr_notebook (WMButtonAction *ba)

Variables

- static const string package_name = "pearl_explorer"
- static const string package_path = "root:packages:pearl_explorer:"
- static const string ks_filematch_adh5 = "*.h5"
- static const string ks_filematch_pshell = "psh*.h5"
- static const string ks_filematch_itx = "*.itx"
- static const string ks_filematch_mtrx = "*_mtrx"

12.12.1 Detailed Description

preview and import panel for PEARL data

preview and import panel for PEARL data:

- area detector (HDF5) files from scienta analyser and prosilica cameras (if HDF5.xop is installed).
- igor text files from s-scans and otf-scans.
- pshell (HDF5) data files (if HDF5.xop is installed).
- matrix STM files (if MatrixFileReader.xop is installed).

12.12.2 Function Documentation

12.12.2.1 attributes_notebook()

```
static variable attributes_notebook (
            wave attr_names,
            wave attr_values,
            string title )  [static]
```

Definition at line 822 of file pearl-data-explorer.ipf.

12.12.2.2 bp_attr_notebook()

```
static variable bp_attr_notebook (
            WMButtonAction * ba )  [static]
```

Definition at line 1750 of file pearl-data-explorer.ipf.

12.12.2.3 bp_browse_filepath()

```
static variable bp_browse_filepath (
            WMButtonAction * ba )  [static]
```

Definition at line 1436 of file pearl-data-explorer.ipf.

12.12.2.4 bp_dataset_display()

```
static variable bp_dataset_display (
            WMButtonAction * ba )  [static]
```

Definition at line 1641 of file pearl-data-explorer.ipf.

12.12.2.5 bp_dataset_folder()

```
static variable bp_dataset_folder (
            WMButtonAction * ba )  [static]
```

Definition at line 1615 of file pearl-data-explorer.ipf.

12.12.2.6 bp_dataset_next()

```
static variable bp_dataset_next (
            WMButtonAction * ba )  [static]
```

Definition at line 1661 of file pearl-data-explorer.ipf.

12.12.2.7 bp_dataset_prev()

```
static variable bp_dataset_prev (
            WMButtonAction * ba )  [static]
```

Definition at line 1688 of file pearl-data-explorer.ipf.

12.12.2.8 bp_file_next()

```
static variable bp_file_next (
            WMButtonAction * ba )  [static]
```

Definition at line 1502 of file pearl-data-explorer.ipf.

12.12.2.9 bp_file_prev()

```
static variable bp_file_prev (
            WMButtonAction * ba )  [static]
```

Definition at line 1534 of file pearl-data-explorer.ipf.

12.12.2.10 bp_load_files()

```
static variable bp_load_files (
            WMButtonAction * ba )  [static]
```

Definition at line 1474 of file pearl-data-explorer.ipf.

12.12.2.11 bp_load_files_opt()

```
static variable bp_load_files_opt (
            WMButtonAction * ba )  [static]
```

Definition at line 1488 of file pearl-data-explorer.ipf.

12.12.2.12 bp_load_prefs()

```
static variable bp_load_prefs (
            WMButtonAction * ba )  [static]
```

Definition at line 1408 of file pearl-data-explorer.ipf.

12.12.2.13 bp_save_prefs()

```
static variable bp_save_prefs (
            WMButtonAction * ba )  [static]
```

Definition at line 1422 of file pearl-data-explorer.ipf.

12.12.2.14 bp_update_datasets()

```
static variable bp_update_datasets (
            WMButtonAction * ba )  [static]
```

Definition at line 1601 of file pearl-data-explorer.ipf.

12.12.2.15 bp_update_filelist()

```
static variable bp_update_filelist (
            WMButtonAction * ba )  [static]
```

Definition at line 1460 of file pearl-data-explorer.ipf.

12.12.2.16 display_dataset()

```
static variable display_dataset (
            string datasetname )  [static]
```

Definition at line 782 of file pearl-data-explorer.ipf.

12.12.2.17 display_preview_trace()

```
static string display_preview_trace (
            wave xtrace,
            wave ytrace )  [static]
```

Definition at line 939 of file pearl-data-explorer.ipf.

12.12.2.18 extract_preview_image()

```
static variable extract_preview_image (
            wave data,
            wave preview )  [static]
```

Definition at line 495 of file pearl-data-explorer.ipf.

12.12.2.19 init_package()

```
static variable init_package ( )  [static]
```

initialize the global variables of the data explorer.

initializes the global variables and data folder for this procedure file must be called once before the panel is created warning: this function overwrites previous values

Definition at line 56 of file pearl-data-explorer.ipf.

12.12.2.20 itx_suggest_foldername()

```
string itx_suggest_foldername (
            string filename,
            variable ignoredate = defaultValue,
            string sourcename = defaultValue,
            variable unique = defaultValue )
```

Definition at line 1258 of file pearl-data-explorer.ipf.

12.12.2.21 lbp_datasets()

```
static variable lbp_datasets (
            WMListboxAction * lba )  [static]
```

Definition at line 1715 of file pearl-data-explorer.ipf.

12.12.2.22 lbp_filelist()

```
static variable lbp_filelist (
            WMListboxAction * lba )  [static]
```

Definition at line 1566 of file pearl-data-explorer.ipf.

12.12.2.23 load_file()

```
static variable load_file (
            string filename,
            string options = defaultValue )  [static]
```

Definition at line 996 of file pearl-data-explorer.ipf.

12.12.2.24 load_hdf_file()

```
static dfr load_hdf_file (
            string filename,
            string options = defaultValue )  [static]
```

Definition at line 1142 of file pearl-data-explorer.ipf.

12.12.2.25 load_itx_file()

```
static dfr load_itx_file (
            string filename,
            string options = defaultValue )  [static]
```

Definition at line 1192 of file pearl-data-explorer.ipf.

12.12.2.26 load_mtrx_file()

```
static dfr load_mtrx_file (
            string filename,
            string options = defaultValue )  [static]
```

load a matrix (STM) data file

Definition at line 1235 of file pearl-data-explorer.ipf.

12.12.2.27 load_prefs()

```
static variable load_prefs ( )  [static]
```

Definition at line 118 of file pearl-data-explorer.ipf.

12.12.2.28 load_pshell_file()

```
static dfr load_pshell_file (
             string filename,
             string options = defaultValue )  [static]
```

Definition at line 1090 of file pearl-data-explorer.ipf.

12.12.2.29 load_selected_files()

```
static variable load_selected_files (
             string options = defaultValue )  [static]
```

Definition at line 972 of file pearl-data-explorer.ipf.

12.12.2.30 notebook_add_attributes()

```
static variable notebook_add_attributes (
             string notebook_name,
             wave attr_filter,
             wave attr_names,
             wave attr_values )  [static]
```

Definition at line 861 of file pearl-data-explorer.ipf.

12.12.2.31 pearl_data_explorer()

```
variable pearl_data_explorer ( )
```

Definition at line 45 of file pearl-data-explorer.ipf.

12.12.2.32 pearl_file_type()

```
static variable pearl_file_type (
             string filename )  [static]
```

check whether a file can be imported by this module.

the file type is determined by the extension of the file name.

Returns

file type

- 0 not a recognized file type
- 1 PShell file (HDF5, name starts with psh_)

- 2 area detector HDF5 file
- 3 Igor text (itx) file
- 4 Matrix STM file (∗_mtrx)

Definition at line 163 of file pearl-data-explorer.ipf.

12.12.2.33 PearlDataExplorer()

```
void PearlDataExplorer ( )
```

Definition at line 1317 of file pearl-data-explorer.ipf.

12.12.2.34 preview_attributes()

```
static variable preview_attributes (
             dfref attr_folder,
             dfref dest_folder = defaultValue,
             wave attr_filter = defaultValue,
             variable include_datawaves = defaultValue,
             variable include_infowaves = defaultValue )  [static]
```

Definition at line 648 of file pearl-data-explorer.ipf.

12.12.2.35 preview_datafolder()

```
static variable preview_datafolder ( )  [static]
```

Definition at line 577 of file pearl-data-explorer.ipf.

12.12.2.36 preview_dataset()

```
static variable preview_dataset (
             string datasetname )  [static]
```

Definition at line 534 of file pearl-data-explorer.ipf.

12.12.2.37 preview_file()

```
static variable preview_file (
             string filename )  [static]
```

Definition at line 257 of file pearl-data-explorer.ipf.

12.12.2.38 preview_hdf_file()

```
static wave preview_hdf_file (
             string filename )  [static]
```

load the preview of a PEARL HDF5 file.

the preview is an arbitrary detector image extracted from the file, see adh5_load_preview(). the preview is loaded to the preview_image wave in the pear_explorer data folder.

the s_file_info string is updated with information about the scan dimensions.

Parameters

filename	name of a file in the directory specified by the pearl_explorer_filepath path object.

Returns

wave reference of the preview image

Definition at line 365 of file pearl-data-explorer.ipf.

12.12.2.39 preview_itx_file()

```
static wave preview_itx_file (
             string filename )  [static]
```

load the preview of a general ITX file.

the function is designed for PEARL OTF and EPICS scan data converted from MDA files. the function picks the first wave whose PV note matches one from the global string s_preview_pvs (see preview_datafolder and init_package).

the preview is loaded to the preview_image wave in the pearl_explorer data folder. the s_file_info string is updated with information about the scan dimensions.

Note

: the ITX files should load their waves into the current data folder (a "free" data folder). some early versions of PEARL ITX data files created a data folder of their own. both ways are allowed, while the first one is preferred. on return, the current data folder must point to either the original free folder or the newly created one.

Parameters

filename	name of a file in the directory specified by the pearl_explorer_filepath path object.

Returns

wave reference of the preview trace. empty wave reference if the function failed.

Definition at line 416 of file pearl-data-explorer.ipf.

12.12.2.40 preview_mtrx_file()

```
static wave preview_mtrx_file (
             string filename )  [static]
```

load the preview of a Matrix STM file.

the preview is loaded to the preview_image wave in the pearl_explorer data folder.

the s_file_info string is updated with information about the scan dimensions.

this function requires the MatrixFileReader.xop and pearl-matrix-import.ipf to be loaded. otherwise it will return an empty wave reference.

Parameters

filename	name of a file in the directory specified by the pearl_explorer_filepath path object.

Returns

wave reference of the preview image. empty wave reference if the function failed.

Definition at line 452 of file pearl-data-explorer.ipf.

12.12.2.41 preview_pshell_file()

```
static wave preview_pshell_file (
            string filename )  [static]
```

load the preview of a PShell HDF5 file.

the preview is an arbitrary detector image extracted from the file, see adh5_load_preview(). the preview is loaded to the preview_image wave in the pear_explorer data folder.

the s_file_info string is updated with information about the scan dimensions.

Parameters

filename	name of a file in the directory specified by the pearl_explorer_filepath path object.

Returns

wave reference of the preview image

Definition at line 309 of file pearl-data-explorer.ipf.

12.12.2.42 preview_setscale_x()

```
static variable preview_setscale_x (
            wave data,
            wave preview )  [static]
```

Definition at line 618 of file pearl-data-explorer.ipf.

12.12.2.43 prompt_default_process()

```
variable prompt_default_process (
            string * param )
```

prototype for prompting for processing function parameters.

the function should prompt the user for function parameters, and update the param argument if the user clicked OK. returns 0 if the user clicked OK, 1 if the user cancelled.

prompt functions must have the same name as the corresponding reduction function with the prefix "prompt_". be aware of the limited length of function names in Igor.

this function is a prototype. it does nothing but returns OK.

Definition at line 1070 of file pearl-data-explorer.ipf.

12.12.2.44 prompt_func_params()

```
variable prompt_func_params (
            string func_name,
            string * func_param )
```

Definition at line 1076 of file pearl-data-explorer.ipf.

12.12.2.45 prompt_hdf_options()

```
static variable prompt_hdf_options (
            string * options )  [static]
```

Definition at line 1031 of file pearl-data-explorer.ipf.

12.12.2.46 save_prefs()

```
static variable save_prefs ( )  [static]
```

save persistent package data to the preferences file.

this function is called when the user clicks the corresponding button. the data saved in the file are: data file path, attributes filter

Definition at line 102 of file pearl-data-explorer.ipf.

12.12.2.47 show_preview_graph()

```
static string show_preview_graph (
            wave data,
            wave xdata = defaultValue )  [static]
```

Definition at line 888 of file pearl-data-explorer.ipf.

12.12.2.48 test_attributes_notebook()

```
variable test_attributes_notebook ( )
```

Definition at line 815 of file pearl-data-explorer.ipf.

12.12.2.49 update_datasets()

```
static variable update_datasets ( )  [static]
```

Definition at line 214 of file pearl-data-explorer.ipf.

12.12.2.50 update_filelist()

```
static variable update_filelist ( )  [static]
```

read a list of PEARL files from the file system

wtFiles and wSelectedFiles in the package data folder are updated. only files for which pearl_file_type() returns non-zero are listed.

Definition at line 186 of file pearl-data-explorer.ipf.

12.12.3 Variable Documentation

12.12.3.1 ks_filematch_adh5

```
const string ks_filematch_adh5 = "*.h5"  [static]
```

Definition at line 40 of file pearl-data-explorer.ipf.

12.12.3.2 ks_filematch_itx

```
const string ks_filematch_itx = "*.itx"  [static]
```

Definition at line 42 of file pearl-data-explorer.ipf.

12.12.3.3 ks_filematch_mtrx

```
const string ks_filematch_mtrx = "*_mtrx"  [static]
```

Definition at line 43 of file pearl-data-explorer.ipf.

12.12.3.4 ks_filematch_pshell

```
const string ks_filematch_pshell = "psh*.h5"  [static]
```

Definition at line 41 of file pearl-data-explorer.ipf.

12.12.3.5 package_name

```
const string package_name = "pearl_explorer"  [static]
```

Definition at line 37 of file pearl-data-explorer.ipf.

12.12.3.6 package_path

```
const string package_path = "root:packages:pearl_explorer:"  [static]
```

Definition at line 38 of file pearl-data-explorer.ipf.

12.13 pearl-elog.ipf File Reference

interface for writing ELOG entries with Igor graphs as attachment.

Namespaces

- PearlElog

  interface for writing ELOG entries with Igor graphs as attachment.

Functions

- variable pearl_elog (string logbook)

  main function to initialize ELOG and to open an ELOG panel.
- static variable IgorBeforeNewHook (string igorApplicationNameStr)

  save preferences and recent values before Igor opens a new experiment.
- static variable IgorQuitHook (string igorApplicationNameStr)

  save preferences and recent values before Igor quits.
- static variable AfterFileOpenHook (variable refNum, string file, string pathName, string type, string creator, variable kind)

  initialize the package and reload preferences after an experiment is loaded.
- static dfr get_elog_df (string name, variable category)

  get the package, logbook, or template datafolder.
- static variable init_package (variable clean=defaultValue)

  initialize the package data folder.
- variable elog_init_pearl_templates ()

  setup PEARL template logbooks.
- static variable init_volatile_vars ()

  initialize volatile variables.
- variable elog_create_logbook (string name, string template=defaultValue)

  create a new logbook.
- variable elog_config (string elog_path=defaultValue, string hostname=defaultValue, variable port=default↩ Value, string subdir=defaultValue)

  set global module configuration parameters
- variable elog_login (string logbook, string username, string password)

  set username and password for login to a logbook
- variable elog_logout (string logbook)

  clear username and password of a logbook or all logbooks.
- static variable save_prefs ()

  save persistent package data to the preferences file.
- static variable load_prefs ()

  load persistent package data from the preferences file.
- static string list_logbooks (variable templates=defaultValue)

  get a list of configured logbooks or templates.
- variable elog_validate_attributes (string logbook, string attributes)

validate attributes

- variable elog_create_entry (string logbook, string attributes, string message, variable encoding=defaultValue, string graphs=defaultValue, variable replyto=defaultValue)

  create a new entry in ELOG
- variable elog_add_attachment (string logbook, variable id, string graphs)

  add one or more graphs to an existing ELOG entry
- static string prepare_command_line (string logbook)

  format the ELOG command and essential address arguments.
- static string format_url (string logbook)

  format the URL for display to the user
- static string prepare_graph_attachments (string graphs)

  prepare screenshots of graph windows for attachments
- static string get_timestamp (string sep)
- static string create_message_file (string message)

  save the message to a temporary text file
- static string create_graph_file (string graphname, variable fileindex)

  save a graph to a temporary graphics file
- static string create_cmd_file (string cmd)

  write the command line to a file.
- static string get_log_path ()
- static variable cleanup_temp_files ()

  delete temporary files created by the ELOG module.
- static variable parse_result ()

  parse the result file from an elog invokation.
- string elog_prompt_logbook ()

  prompt to open or create a logbook
- variable elog_prompt_login (string logbook)

  prompt the user for login to a logbook
- string PearlElogPanel (string logbook)

  open a new panel for submitting data to ELOG.
- static variable elog_panel_hook (WMWinHookStruct *s)
- static variable update_attach_items (string logbook)

  update the list of attachments
- static variable move_attach_item (string logbook, variable item, variable distance)

  move an attachment item in the list of attachments
- static variable bp_attach_updown (WMButtonAction *ba)

  button procedure for the attachment up and down buttons
- static variable bp_submit (WMButtonAction *ba)

  button procedure for the Submit and Reply buttons
- static variable bp_attach_top (WMButtonAction *ba)

  select top graph window for attachment
- static variable bp_attach_allnone (WMButtonAction *ba)

  select/deselect all graph windows for attachment
- static variable bp_attach (WMButtonAction *ba)
- static variable bp_save_graphs (WMButtonAction *ba)
- static variable bp_clear (WMButtonAction *ba)
- static variable bp_login (WMButtonAction *ba)
- static variable bp_logout (WMButtonAction *ba)
- static string get_default_panel_name ()
- static string get_panel_attributes (string windowname)

  get a list of attributes from the fields of the ELOG panel.

- static string set_panel_attributes (string windowname, string attributes, variable clear=defaultValue)

  set the fields of the ELOG panel
- static string get_panel_message (string windowname)

  get the message field of the ELOG panel
- static string set_panel_message (string windowname, string message)

  set the message field of the ELOG panel
- static string get_panel_graphs (string windowname)

  get the names of the graphs selected for attachment
- static string set_panel_graphs (string windowname, string graphs)

  update selection of graphs for attachment

Variables

- static const string package_name = "pearl_elog"
- static const string package_path = "root:packages:pearl_elog:"
- static const variable kdfRoot = 0
- static const variable kdfVolatile = 1
- static const variable kdfPersistent = 2
- static const variable kdfTemplates = 3
- static const string elog_success_msg = "Message successfully transmitted"
- static const string elog_parse_id = "ID=%u"
- static const variable kAttachColSel = 0
- static const variable kAttachColTitle = 1
- static const variable kAttachColName = 2

12.13.1 Detailed Description

interface for writing ELOG entries with Igor graphs as attachment.

the functions in this module support the following ELOG features:

- submit new entries and replies to existing entries.
- text field, list box, and check box attributes.
- attach any Igor graph to ELOG.
- configurable logbook templates for logbooks that share the same configuration.
- common server configurations available on the ELOG command line (hostname, port, SSL, username, password, sub-directory).
- not specific to the configuration at PEARL. PEARL code is concentrated in the elog_init_pearl_templates() function.
- the configuration of the ELOG server and logbooks as well as the most recently used attributes are persisted in the preference file.

usage:

1. the administrator of the ELOG server creates logbook templates according to the configuration of the logbooks. the templates are written in Igor code.
2. the user opens logbooks via the *Open ELOG panel* menu item. before first use, select a template and enter a name for the logbook. the new logbook is written to the preference file, and can afterwards be opened directly.
3. if the server requires a user name and password, click the login button.

4. edit the message, attributes and attachments as necessary, and submit to ELOG.
5. log out before saving the experiment to clear the password.

Attention

the user name and password are stored in the global data tree of an experiment. it is not possible to handle passwords safely in Igor. they can be read by anyone having access to an open Igor experiment or a saved experiment file (unless the password is reset before saving). therefore:

- use a password for the ELOG server which is different from your other passwords.
- clear the password (logout button in the panel) before saving an experiment.

elog command line

```
*  elog -h <hostname> [-p port] [-d subdir]
*                               Location where elogd is running
*       -l logbook              Name of logbook
*       -s                      Use SSL for communication
*       [-v]                    For verbose output
*       [-w password]           Write password defined on server
*       [-u username password]  User name and password
*       [-f <attachment>]       Up to 50 attachments
*       -a <attribute>=<value>  Up to 50 attributes
*       [-r <id>]               Reply to existing message
*       [-q]                    Quote original text on reply
*       [-e <id>]               Edit existing message
*       [-x]                    Suppress email notification
*       [-n 0|1|2]              Encoding: 0:ELcode,1:plain,2:HTML
*       -m <textfile>] | <text>
*
```

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.13.2 Function Documentation

12.13.2.1 AfterFileOpenHook()

```
static variable AfterFileOpenHook (
             variable refNum,
             string file,
             string pathName,
             string type,
             string creator,
             variable kind )  [static]
```

initialize the package and reload preferences after an experiment is loaded.

Definition at line 143 of file pearl-elog.ipf.

12.13.2.2 bp_attach()

```
static variable bp_attach (
             WMButtonAction * ba )  [static]
```

Definition at line 1635 of file pearl-elog.ipf.

12.13.2.3 bp_attach_allnone()

```
static variable bp_attach_allnone (
             WMButtonAction * ba )  [static]
```

select/deselect all graph windows for attachment

Definition at line 1614 of file pearl-elog.ipf.

12.13.2.4 bp_attach_top()

```
static variable bp_attach_top (
             WMButtonAction * ba )  [static]
```

select top graph window for attachment

Definition at line 1598 of file pearl-elog.ipf.

12.13.2.5 bp_attach_updown()

```
static variable bp_attach_updown (
             WMButtonAction * ba )  [static]
```

button procedure for the attachment up and down buttons

Definition at line 1519 of file pearl-elog.ipf.

12.13.2.6 bp_clear()

```
static variable bp_clear (
             WMButtonAction * ba )  [static]
```

Definition at line 1696 of file pearl-elog.ipf.

12.13.2.7 bp_login()

```
static variable bp_login (
             WMButtonAction * ba )  [static]
```

Definition at line 1712 of file pearl-elog.ipf.

12.13.2.8 bp_logout()

```
static variable bp_logout (
            WMButtonAction * ba )  [static]
```

Definition at line 1730 of file pearl-elog.ipf.

12.13.2.9 bp_save_graphs()

```
static variable bp_save_graphs (
            WMButtonAction * ba )  [static]
```

Definition at line 1668 of file pearl-elog.ipf.

12.13.2.10 bp_submit()

```
static variable bp_submit (
            WMButtonAction * ba )  [static]
```

button procedure for the Submit and Reply buttons

Definition at line 1551 of file pearl-elog.ipf.

12.13.2.11 cleanup_temp_files()

```
static variable cleanup_temp_files ( )  [static]
```

delete temporary files created by the ELOG module.

this deletes all temporary graph files that are referenced by the volatile temp_graph_files list. temp_graph_files is a semicolon-delimited string. items are added by create_graph_file().

this function should be called before a new experiment is loaded or igor quits.

Definition at line 1115 of file pearl-elog.ipf.

12.13.2.12 create_cmd_file()

```
static string create_cmd_file (
            string cmd )  [static]
```

write the command line to a file.

the command script changes the working directory to the Temporary directory. it also deletes a previous elog.log file.

Note

somewhere the command line (even inside command files) is limited to 1024 bytes. for this reason all files should now be in the Temporary directory assigned by igor.

Definition at line 1059 of file pearl-elog.ipf.

12.13.2.13 create_graph_file()

```
static string create_graph_file (
             string graphname,
             variable fileindex )  [static]
```

save a graph to a temporary graphics file

the file is saved to the Temporary directory returned by igor's SpecialDirPath function. the file name contains a time stamp and the specified file index to make it unique. the function returns the name of the file (excluding path!)

the full path is added to the temp_graph_files global list. a hook function will delete the files listed there when igor quits.

Parameters

graphname	object name of the graph to save.
fileindex	incrememtal index of the file within one submission. the file name is made unique by a time stamp and this file index. submissions within the same second must have a unique file index.

Returns

(string) name of the created file. empty string if unsuccessful.

Version

1.41 the return value has changed from full path to file name only due to the limited length of the command line (1024 bytes).

Definition at line 1023 of file pearl-elog.ipf.

12.13.2.14 create_message_file()

```
static string create_message_file (
             string message )  [static]
```

save the message to a temporary text file

the file is saved to the Temporary directory returned by igor's SpecialDirPath function under the file name "elog_↩ temp_message.txt". the function returns the name of the file (excluding path!)

Note

percent characters (%) cannot be passed to elog. they are removed silently from the message.

Parameters

message	text message to save to the file.

Returns

(string) name of the created file. empty string if unsuccessful.

Version

1.41 the return value has changed from full path to file name only due to the limited length of the command line (1024 bytes).

Definition at line 982 of file pearl-elog.ipf.

12.13.2.15 elog_add_attachment()

```
variable elog_add_attachment (
             string logbook,
             variable id,
             string graphs )
```

add one or more graphs to an existing ELOG entry

Parameters

logbook	name of the target logbook
id	identification number of the existing entry
graphs	names of graph windows to be added as attachments, semicolon separated

Warning

this will delete all existing attachments of the entry!

Definition at line 792 of file pearl-elog.ipf.

12.13.2.16 elog_config()

```
variable elog_config (
             string elog_path = defaultValue,
             string hostname = defaultValue,
             variable port = defaultValue,
             string subdir = defaultValue )
```

set global module configuration parameters

Definition at line 473 of file pearl-elog.ipf.

12.13.2.17 elog_create_entry()

```
variable elog_create_entry (
             string logbook,
             string attributes,
             string message,
             variable encoding = defaultValue,
             string graphs = defaultValue,
             variable replyto = defaultValue )
```

create a new entry in ELOG

this is the main function to create a new entry in a logbook.

Parameters

logbook	name of the target logbook.
attributes	key=value list of attributes, semicolon separated.
message	free text part of the entry.
encoding	encoding of message, 0:ELcode, 1:plain (default), 2:HTML.
graphs	names of graph windows to be added as attachments, semicolon separated.
replyto	existing message ID ($>$ 1) to follow up on. 0 or default: start new thread.

Returns

ID number of the new entry ($>$ 0), or error code ($<$ 0).

- -1: failed to save temporary message file.
- -2: invalid/missing command line.
- -3: invalid/missing attributes.
- -4: elog returned error

Definition at line 685 of file pearl-elog.ipf.

12.13.2.18 elog_create_logbook()

```
variable elog_create_logbook (
             string name,
             string template = defaultValue )
```

create a new logbook.

create a new empty logbook or duplicate from a template.

Parameters

name	name of the new logbook. if the logbook exists, the existing logbook folder is killed and replaced by a new one. this may fail if a window is still open.
template	name of the template. if empty string, a new empty logbook is created.

Definition at line 414 of file pearl-elog.ipf.

12.13.2.19 elog_init_pearl_templates()

```
variable elog_init_pearl_templates ( )
```

setup PEARL template logbooks.

template logbooks for PEARL.

Remarks

this function is specific to the setup at PEARL.

Definition at line 262 of file pearl-elog.ipf.

12.13.2.20 elog_login()

```
variable elog_login (
             string logbook,
             string username,
             string password )
```

set username and password for login to a logbook

the username and password are stored (in plain text) in global strings under the selected logbook folder. this is necessary for sending data to the ELOG server.

call elog_logout() to clear the password variables and to avoid unintended use of your credentials.

Warning

igor does not have a built-in mechanism to protect passwords. user names and passwords are stored in plain text in the data folder tree. as such they are saved to experiment files and preferences.

Parameters

logbook	name of the target logbook.

Definition at line 513 of file pearl-elog.ipf.

12.13.2.21 elog_logout()

```
variable elog_logout (
             string logbook )
```

clear username and password of a logbook or all logbooks.

the username and password are stored (in plain text) in global strings under the selected logbook folder. this function resets the username and password strings.

Parameters

logbook	name of the target logbook. if empty, the passwords of all logbooks are cleared.

Definition at line 533 of file pearl-elog.ipf.

12.13.2.22 elog_panel_hook()

```
static variable elog_panel_hook (
             WMWinHookStruct * s )  [static]
```

Definition at line 1395 of file pearl-elog.ipf.

12.13.2.23 elog_prompt_logbook()

```
string elog_prompt_logbook ( )
```

prompt to open or create a logbook

Definition at line 1184 of file pearl-elog.ipf.

12.13.2.24 elog_prompt_login()

```
variable elog_prompt_login (
             string logbook )
```

prompt the user for login to a logbook

Definition at line 1214 of file pearl-elog.ipf.

12.13.2.25 elog_validate_attributes()

```
variable elog_validate_attributes (
             string logbook,
             string attributes )
```

validate attributes

Returns

0 if all required attributes are present and enumerated items are correct. non-zero if a violation is detected.

Todo function currently not implemented, always returns 0

Definition at line 653 of file pearl-elog.ipf.

12.13.2.26 format_url()

```
static string format_url (
             string logbook )  [static]
```

format the URL for display to the user

Parameters

logbook	name of the target logbook

Definition at line 902 of file pearl-elog.ipf.

12.13.2.27 get_default_panel_name()

```
static string get_default_panel_name ( )  [static]
```

Definition at line 1747 of file pearl-elog.ipf.

12.13.2.28 get_elog_df()

```
static dfr get_elog_df (
```

```
            string name,
            variable category )  [static]
```

get the package, logbook, or template datafolder.

Parameters

name	name of logbook or template, or empty string for respective parent folder.
category	parameter category: • kdfRoot package root • kdfVolatile volatile • kdfPersistent persistent • kdfTemplates template

Returns

data folder reference

Definition at line 170 of file pearl-elog.ipf.

12.13.2.29 get_log_path()

```
static string get_log_path ( )  [static]
```

Definition at line 1095 of file pearl-elog.ipf.

12.13.2.30 get_panel_attributes()

```
static string get_panel_attributes (
            string windowname )  [static]
```

get a list of attributes from the fields of the ELOG panel.

Parameters

windowname	window name of the ELOG panel if empty, use default name "PearlElogPanel"

Returns

list of attributes to in the format "`key1=value1;key2=value2`".

Definition at line 1760 of file pearl-elog.ipf.

12.13.2.31 get_panel_graphs()

```
static string get_panel_graphs (
            string windowname )  [static]
```

get the names of the graphs selected for attachment

Parameters

windowname	panel window name

Returns

a semicolon-separated list, or the empty string if the selection is not valid.

Definition at line 1926 of file pearl-elog.ipf.

12.13.2.32 get_panel_message()

```
static string get_panel_message (
             string windowname )  [static]
```

get the message field of the ELOG panel

Parameters

windowname	window name of the ELOG panel if empty, use default name "PearlElogPanel"

Returns

message text

Definition at line 1880 of file pearl-elog.ipf.

12.13.2.33 get_timestamp()

```
static string get_timestamp (
             string sep )  [static]
```

Definition at line 958 of file pearl-elog.ipf.

12.13.2.34 IgorBeforeNewHook()

```
static variable IgorBeforeNewHook (
             string igorApplicationNameStr )  [static]
```

save preferences and recent values before Igor opens a new experiment.

Definition at line 127 of file pearl-elog.ipf.

12.13.2.35 IgorQuitHook()

```
static variable IgorQuitHook (
             string igorApplicationNameStr )  [static]
```

save preferences and recent values before Igor quits.

Definition at line 135 of file pearl-elog.ipf.

12.13.2.36 init_package()

```
static variable init_package (
             variable clean = defaultValue )  [static]
```

initialize the package data folder.

the data folder is initialized with a default, local configuration without any logbooks. the server configuration should be set in the preferences.

Parameters

clean	decides what to do if the package configuration exists. • 0 (default) keep existing configuration. • 1 overwrite existing configuration.

Definition at line 217 of file pearl-elog.ipf.

12.13.2.37 init_volatile_vars()

```
static variable init_volatile_vars ( )  [static]
```

initialize volatile variables.

create and initialize all volatile variables for the configured notebooks. values of existing variables are not changed.

this function must be called after new logbooks have been configured, specifically by elog_create_logbook() and load_prefs().

Definition at line 340 of file pearl-elog.ipf.

12.13.2.38 list_logbooks()

```
static string list_logbooks (
             variable templates = defaultValue )  [static]
```

get a list of configured logbooks or templates.

this is list of data folder names under persistent:logbooks (or persistent:templates). the function does not check whether the folders contain valid data.

Parameters

templates	select whether logbooks (0, default) or templates (1) are returned.

Returns

semicolon-separated list of logbooks

Definition at line 618 of file pearl-elog.ipf.

12.13.2.39 load_prefs()

```
static variable load_prefs ( )  [static]
```

load persistent package data from the preferences file.

the preferences file is an Igor packed experiment file in a special preferences folder

Definition at line 584 of file pearl-elog.ipf.

12.13.2.40 move_attach_item()

```
static variable move_attach_item (
             string logbook,
             variable item,
             variable distance )  [static]
```

move an attachment item in the list of attachments

Definition at line 1495 of file pearl-elog.ipf.

12.13.2.41 parse_result()

```
static variable parse_result ( )  [static]
```

parse the result file from an elog invokation.

Returns

the ID of the generated message, or a value <= 0 if an error occurred.

Definition at line 1141 of file pearl-elog.ipf.

12.13.2.42 pearl_elog()

```
variable pearl_elog (
             string logbook )
```

main function to initialize ELOG and to open an ELOG panel.

this function takes care of all necessary initialization, configuration, and preferences. if a panel exists, it will be moved to the front.

Parameters

logbook	name of the logbook if empty, the user is prompted to select or create a logbook by elog_prompt_logbook().

Definition at line 97 of file pearl-elog.ipf.

12.13.2.43 PearlElogPanel()

```
string PearlElogPanel (
```

```
            string logbook )
```

open a new panel for submitting data to ELOG.

this function creates only the panel but not the necessary data folders. call init_package() and load_prefs() once before creating panels.

Parameters

logbook	name of the target logbook

Definition at line 1241 of file pearl-elog.ipf.

12.13.2.44 prepare_command_line()

```
static string prepare_command_line (
            string logbook )  [static]
```

format the ELOG command and essential address arguments.

the following arguments are included (from global variables) if applicable: host name, port, SSL, sub-dir, username, password the result string does not include leading or trailing space

Parameters

logbook	name of the target logbook

Definition at line 853 of file pearl-elog.ipf.

12.13.2.45 prepare_graph_attachments()

```
static string prepare_graph_attachments (
            string graphs )  [static]
```

prepare screenshots of graph windows for attachments

prepares the attachment files from Igor graph windows and returns the arguments to the elog command to attach the files. file names are returned without path. the result string does not include leading or trailing space.

Parameters

graphs	names of graph windows to be added as attachments, semicolon separated

Definition at line 939 of file pearl-elog.ipf.

12.13.2.46 save_prefs()

```
static variable save_prefs ( )  [static]
```

save persistent package data to the preferences file.

saves everything under the persistent folder of the package.

Definition at line 564 of file pearl-elog.ipf.

12.13.2.47 set_panel_attributes()

```
static string set_panel_attributes (
             string windowname,
             string attributes,
             variable clear = defaultValue )  [static]
```

set the fields of the ELOG panel

Parameters

windowname	window name of the ELOG panel if empty, use default name "PearlElogPanel"
attributes	list of attributes to set (format "key1=value1;key2=value2")
clear	what to do if a key is missing in attributes? • 0 (default) leave the field unchanged • 1 clear the field

Definition at line 1807 of file pearl-elog.ipf.

12.13.2.48 set_panel_graphs()

```
static string set_panel_graphs (
             string windowname,
             string graphs )  [static]
```

update selection of graphs for attachment

Parameters

windowname	panel window name. looks for default panel if empty.
graphs	semicolon-separated list of names of graph windows to select for attachment.

Definition at line 1966 of file pearl-elog.ipf.

12.13.2.49 set_panel_message()

```
static string set_panel_message (
             string windowname,
             string message )  [static]
```

set the message field of the ELOG panel

Parameters

windowname	window name of the ELOG panel if empty, use default name "PearlElogPanel"
message	message text that can be passed to the `Notebook` operation.

Returns

original message (unchanged)

Definition at line 1906 of file pearl-elog.ipf.

12.13.2.50 update_attach_items()

```
static variable update_attach_items (
             string logbook )  [static]
```

update the list of attachments

Definition at line 1434 of file pearl-elog.ipf.

12.13.3 Variable Documentation

12.13.3.1 elog_parse_id

```
const string elog_parse_id = "ID=%u"  [static]
```

Definition at line 1134 of file pearl-elog.ipf.

12.13.3.2 elog_success_msg

```
const string elog_success_msg = "Message successfully transmitted"  [static]
```

Definition at line 1133 of file pearl-elog.ipf.

12.13.3.3 kAttachColName

```
const variable kAttachColName = 2  [static]
```

Definition at line 1431 of file pearl-elog.ipf.

12.13.3.4 kAttachColSel

```
const variable kAttachColSel = 0  [static]
```

Definition at line 1429 of file pearl-elog.ipf.

12.13.3.5 kAttachColTitle

```
const variable kAttachColTitle = 1  [static]
```

Definition at line 1430 of file pearl-elog.ipf.

12.13.3.6 kdfPersistent

```
const variable kdfPersistent = 2  [static]
```

Definition at line 155 of file pearl-elog.ipf.

12.13.3.7 kdfRoot

```
const variable kdfRoot = 0  [static]
```

Definition at line 153 of file pearl-elog.ipf.

12.13.3.8 kdfTemplates

```
const variable kdfTemplates = 3  [static]
```

Definition at line 156 of file pearl-elog.ipf.

12.13.3.9 kdfVolatile

```
const variable kdfVolatile = 1  [static]
```

Definition at line 154 of file pearl-elog.ipf.

12.13.3.10 package_name

```
const string package_name = "pearl_elog"  [static]
```

Definition at line 85 of file pearl-elog.ipf.

12.13.3.11 package_path

```
const string package_path = "root:packages:pearl_elog:"  [static]
```

Definition at line 86 of file pearl-elog.ipf.

12.14 pearl-fitfuncs.ipf File Reference

various fit functions for photoelectron spectroscopy.

```
#include "mm-physconst"
```

Data Structures

- struct DoniachSunjicStruct

Namespaces

- PearlFitFuncs

  various fit functions for photoelectron spectroscopy.

Functions

- threadsafe variable MultiGaussLinBG (wave w, variable x)

  multiple gaussian peaks on a linear background fit function.
- threadsafe variable MultiGaussLinBG_AO (wave pw, wave yw, wave xw)

  multiple gaussian peaks on a linear background fit function (all at once).
- threadsafe variable DoubletGaussLinBG_AO (wave pw, wave yw, wave xw)

  doublet gaussian peaks on a linear background fit function (all at once).
- variable MultiVoigtLinBG (wave w, variable x)

  multiple voigt peaks on a linear background fit function.
- threadsafe variable DoniachSunjic (variable x, variable amp, variable pos, variable sing, variable fwhm)

  Doniach-Sunjic line shape.
- variable MultiDoniachSunjicLinBG (wave w, variable x)

  multiple doniach-sunjic peaks on a linear background fit function.
- threadsafe variable ds1_bg (wave w, variable x)
- threadsafe variable ds2_bg (wave w, variable x)
- variable ds4_bg (wave w, variable x)
- variable ds6_bg (wave w, variable x)
- threadsafe variable DoniachSunjicBroadS (DoniachSunjicStruct *s)
- variable DoniachSunjicBroad (wave pw, wave yw, wave xw)
- variable Calc_DoniachSunjicBroad (wave pw, wave yw)
- variable Fit_DoniachSunjicBroad (wave pw, wave yw, wave xw, wave ww)
- variable Au4f (wave w, variable x)
- variable Au4f_2p2 (wave w, variable x)
- variable ShowComponents_Au4f_2p2 (wave coef_wave, wave fit_wave)
- variable Au4f_2p3 (wave w, variable x)
- variable ShowComponents_Au4f_2p3 (wave coef_wave, wave fit_wave)
- variable FermiGaussConv (wave pw, wave yw, wave xw)

  convolution of Fermi-Dirac distribution and a Gaussian.
- variable ShirleyBG (wave w, wave bg, variable p1, variable p2)

  calculate the shirley background

12.14.1 Detailed Description

various fit functions for photoelectron spectroscopy.

this procedure contains various functions for curve fitting.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.14.2 Function Documentation

12.14.2.1 Au4f()

```
variable Au4f (
             wave w,
             variable x )
```

Definition at line 598 of file pearl-fitfuncs.ipf.

12.14.2.2 Au4f_2p2()

```
variable Au4f_2p2 (
             wave w,
             variable x )
```

Definition at line 643 of file pearl-fitfuncs.ipf.

12.14.2.3 Au4f_2p3()

```
variable Au4f_2p3 (
             wave w,
             variable x )
```

Definition at line 728 of file pearl-fitfuncs.ipf.

12.14.2.4 Calc_DoniachSunjicBroad()

```
variable Calc_DoniachSunjicBroad (
             wave pw,
             wave yw )
```

Definition at line 541 of file pearl-fitfuncs.ipf.

12.14.2.5 DoniachSunjic()

```
threadsafe variable DoniachSunjic (
             variable x,
             variable amp,
             variable pos,
             variable sing,
             variable fwhm )
```

Doniach-Sunjic line shape.

[S. Doniach, M. Sunjic, J. Phys. C 3 (1970) 285]

Parameters

x	independent variable

Parameters

amp	amplitude
pos	position
sing	singularity index (0 <= sing < 1)
fwhm	full width at half maximum

Definition at line 174 of file pearl-fitfuncs.ipf.

12.14.2.6 DoniachSunjicBroad()

```
variable DoniachSunjicBroad (
             wave pw,
             wave yw,
             wave xw )
```

Definition at line 480 of file pearl-fitfuncs.ipf.

12.14.2.7 DoniachSunjicBroadS()

```
threadsafe variable DoniachSunjicBroadS (
             DoniachSunjicStruct * s )
```

Definition at line 391 of file pearl-fitfuncs.ipf.

12.14.2.8 DoubletGaussLinBG_AO()

```
threadsafe variable DoubletGaussLinBG_AO (
             wave pw,
             wave yw,
             wave xw )
```

doublet gaussian peaks on a linear background fit function (all at once).

this fits two gaussian peaks. peak positions are specified by center and distance rather than individually. amplitude and width are specified as absolute values for the upper peak (in x), and relative values for the lower peak.

Note

FWHM = width $*$ 2 $*$ sqrt(ln(2)) = width $*$ 1.665

Parameters

pw	shape parameters. the length of the wave defines the number of peaks. • pw[0] = constant coefficient of background • pw[1] = linear coefficient of background • pw[2] = amplitude of peak 1 (higher x) • pw[3] = amplitude of peak 2 relative to peak 1 • pw[4] = center position • pw[5] = distance between peaks (splitting) • pw[6] = width of peak 1 (see note) • pw[7] = width of peak 2 relative to peak 1
yw	y (dependent) values.
xw	x (independent) independent values.

Definition at line 117 of file pearl-fitfuncs.ipf.

12.14.2.9 ds1_bg()

```
threadsafe variable ds1_bg (
             wave w,
             variable x )
```

Definition at line 217 of file pearl-fitfuncs.ipf.

12.14.2.10 ds2_bg()

```
threadsafe variable ds2_bg (
             wave w,
             variable x )
```

Definition at line 240 of file pearl-fitfuncs.ipf.

12.14.2.11 ds4_bg()

```
variable ds4_bg (
             wave w,
             variable x )
```

Definition at line 268 of file pearl-fitfuncs.ipf.

12.14.2.12 ds6_bg()

```
variable ds6_bg (
             wave w,
```

```
            variable x )
```

Definition at line 309 of file pearl-fitfuncs.ipf.

12.14.2.13 FermiGaussConv()

```
variable FermiGaussConv (
            wave pw,
            wave yw,
            wave xw )
```

convolution of Fermi-Dirac distribution and a Gaussian.

- pw[0] = constant background
- pw[1] = linear background
- pw[2] = amplitude
- pw[3] = Fermi level in eV
- pw[4] = temperature in K
- pw[5] = gaussian width = FWHM / 1.66511

Definition at line 851 of file pearl-fitfuncs.ipf.

12.14.2.14 Fit_DoniachSunjicBroad()

```
variable Fit_DoniachSunjicBroad (
            wave pw,
            wave yw,
            wave xw,
            wave ww )
```

Definition at line 562 of file pearl-fitfuncs.ipf.

12.14.2.15 MultiDoniachSunjicLinBG()

```
variable MultiDoniachSunjicLinBG (
            wave w,
            variable x )
```

multiple doniach-sunjic peaks on a linear background fit function.

Parameters

w	shape parameters. the length of the wave defines the number of peaks. • w[0] = constant coefficient of background • w[1] = linear coefficient of background • w[2 + (i-1) ∗ 4] = amplitude of peak i • w[3 + (i-1) ∗ 4] = position of peak i • w[4 + (i-1) ∗ 4] = width (fwhm) of peak i • w[5 + (i-1) ∗ 4] = singularity index (0...1) of peak i
x	independent variable

Definition at line 201 of file pearl-fitfuncs.ipf.

12.14.2.16 MultiGaussLinBG()

```
threadsafe variable MultiGaussLinBG (
             wave w,
             variable x )
```

multiple gaussian peaks on a linear background fit function.

Note

FWHM = width ∗ 2 ∗ sqrt(ln(2)) = width ∗ 1.665

Parameters

w	shape parameters. the length of the wave defines the number of peaks. • w[0] = constant coefficient of background • w[1] = linear coefficient of background • w[2 + (i-1) ∗ 3] = amplitude of peak i • w[3 + (i-1) ∗ 3] = position of peak i • w[4 + (i-1) ∗ 3] = width of peak i (see note)
x	independent variable

Definition at line 44 of file pearl-fitfuncs.ipf.

12.14.2.17 MultiGaussLinBG_AO()

```
threadsafe variable MultiGaussLinBG_AO (
             wave pw,
             wave yw,
             wave xw )
```

multiple gaussian peaks on a linear background fit function (all at once).

this is the all-at-once version of MultiGaussLinBG. it runs about 15% faster compared to the point-by-point function (measured on a 200 point spectrum with 3 peaks).

Note

FWHM = width ∗ 2 ∗ sqrt(ln(2)) = width ∗ 1.665

Parameters

pw	shape parameters. the length of the wave defines the number of peaks. • pw[0] = constant coefficient of background • pw[1] = linear coefficient of background • pw[2 + (i-1) ∗ 3] = amplitude of peak i • pw[3 + (i-1) ∗ 3] = position of peak i • pw[4 + (i-1) ∗ 3] = width of peak i (see note)
yw	y (dependent) values.
xw	x (independent) independent values.

Definition at line 79 of file pearl-fitfuncs.ipf.

12.14.2.18 MultiVoigtLinBG()

```
variable MultiVoigtLinBG (
             wave w,
             variable x )
```

multiple voigt peaks on a linear background fit function.

Parameters

w	shape parameters. the length of the wave defines the number of peaks. • w[0] = constant coefficient of background • w[1] = linear coefficient of background • w[2 + (i-1) ∗ 4] = amplitude of peak i • w[3 + (i-1) ∗ 4] = position of peak i • w[4 + (i-1) ∗ 4] = width of peak i • w[5 + (i-1) ∗ 4] = shape of peak i
x	independent variable

Definition at line 144 of file pearl-fitfuncs.ipf.

12.14.2.19 ShirleyBG()

```
variable ShirleyBG (
             wave w,
             wave bg,
             variable p1,
             variable p2 )
```

calculate the shirley background

Definition at line 901 of file pearl-fitfuncs.ipf.

12.14.2.20 ShowComponents_Au4f_2p2()

```
variable ShowComponents_Au4f_2p2 (
             wave coef_wave,
             wave fit_wave )
```

Definition at line 697 of file pearl-fitfuncs.ipf.

12.14.2.21 ShowComponents_Au4f_2p3()

```
variable ShowComponents_Au4f_2p3 (
             wave coef_wave,
             wave fit_wave )
```

Definition at line 796 of file pearl-fitfuncs.ipf.

12.15 pearl-gui-tools.ipf File Reference

Functions

- variable display_progress_panel (string title, string message, variable progress_max)
- variable update_progress_panel (variable progress, string message=defaultValue, variable progress_↩ max=defaultValue)
- variable kill_progress_panel ()

12.15.1 Function Documentation

12.15.1.1 display_progress_panel()

```
variable display_progress_panel (
             string title,
             string message,
             variable progress_max )
```

Definition at line 18 of file pearl-gui-tools.ipf.

12.15.1.2 kill_progress_panel()

```
variable kill_progress_panel ( )
```

Definition at line 50 of file pearl-gui-tools.ipf.

12.15.1.3 update_progress_panel()

```
variable update_progress_panel (
             variable progress,
             string message = defaultValue,
             variable progress_max = defaultValue )
```

Definition at line 32 of file pearl-gui-tools.ipf.

12.16 pearl-matrix-import.ipf File Reference

data file import for omicron matrix (STM) files

Data Structures

- struct errorCode

  from matrixfilereader help

Namespaces

- PearlMatrixImport

  data file import for omicron matrix (STM) files

Functions

- static variable init_package ()

  initialize the package data folder.
- static variable check_package_folder ()

  check that the package data folder exists
- static variable AfterFileOpenHook (variable refNum, string file, string pathName, string type, string creator, variable kind)

  initialize the package and reload preferences after an experiment is loaded.
- static variable BeforeFileOpenHook (variable refNum, string fileName, string path, string type, string creator, variable kind)

  open a matrix file that was dropped into Igor.
- string matrix_format_elog_message (wave metadata)

  generate elog message from bricklet metadata
- variable matrix_preview_2d (wave data, wave metadata)
- static wave preview_matrix_file (string filename)

  load the preview of a Matrix data file
- static variable initStruct (errorCode *errorCode)

  from matrixfilereader help
- variable mtrx_load_all ()

  load all data from a Matrix data file.

- variable mtrx_parse_filename (string fileName, string *resultFile, variable *runCycle, variable *scanCycle, string *channel)

  parse matrix file names
- string mtrx_split_filename (string fileName, string *prefix, string *datepart, string *timepart)

  split a matrix filename and return the first three parts
- dfr mtrx_create_folder (string fileName, dfref df_base=defaultValue)

  create or look up a data folder based on a matrix file name.
- dfr mtrx_get_cycle_folder (dfref df_base=defaultValue, variable runCycle=defaultValue, variable scanCycle=defaultValue)

  create a data folder for bricklet data.
- variable mtrx_file_brickletID (string resultFile, variable runCycle, variable scanCycle, string channel)

  find out bricklet ID of a file
- variable mtrx_open_file (string pathName, string fileNameOrPath)

  open a matrix result or data file
- string mtrx_load_preview (string destName, string pathName, string fileName, string traces=defaultValue)

  load a preview image from a Matrix data file.
- string mtrx_load_file (string pathName, string fileName, string traces=defaultValue)

  load all data from a Matrix data file.
- variable mtrx_scale_dataset (wave data)
- string mtrx_load_info (string APathName, string AFileName)

  load descriptive info from a Matrix data file.
- variable subtract_line_bg (wave img)

  remove linear background line-by-line

Variables

- static const string package_name = "pearl_matrix_import"
- static const string package_path = "root:packages:pearl_matrix_import:"
- static const string ks_filematch_mtrx = "*_mtrx"

12.16.1 Detailed Description

data file import for omicron matrix (STM) files

the matrix file import requires the matrix file reader XOP by thomas braun (http://www.igorexchange.com/project/matrixFileReader) which in turn requires an installation of vernissage by omicron nanotechnology.

Warning

EXPERIMENTAL the matrix import module and its interface may change radically in future revisions!

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.16.2 Function Documentation

12.16.2.1 AfterFileOpenHook()

```
static variable AfterFileOpenHook (
            variable refNum,
            string file,
            string pathName,
            string type,
            string creator,
            variable kind )  [static]
```

initialize the package and reload preferences after an experiment is loaded.

Definition at line 85 of file pearl-matrix-import.ipf.

12.16.2.2 BeforeFileOpenHook()

```
static variable BeforeFileOpenHook (
            variable refNum,
            string fileName,
            string path,
            string type,
            string creator,
            variable kind )  [static]
```

open a matrix file that was dropped into Igor.

preliminary implementation. this should rather load the entire file and display a preview. graph windows should be reused by subsequent loads. also decide on a data saving location.

Definition at line 102 of file pearl-matrix-import.ipf.

12.16.2.3 check_package_folder()

```
static variable check_package_folder ( )  [static]
```

check that the package data folder exists

initialize the package if the folder does not exist.

Definition at line 72 of file pearl-matrix-import.ipf.

12.16.2.4 init_package()

```
static variable init_package ( )  [static]
```

initialize the package data folder.

Definition at line 45 of file pearl-matrix-import.ipf.

12.16.2.5 initStruct()

```
static variable initStruct (
             errorCode * errorCode )  [static]
```

from matrixfilereader help

Definition at line 212 of file pearl-matrix-import.ipf.

12.16.2.6 matrix_format_elog_message()

```
string matrix_format_elog_message (
             wave metadata )
```

generate elog message from bricklet metadata

Parameters

metadata	two-column text wave

Definition at line 120 of file pearl-matrix-import.ipf.

12.16.2.7 matrix_preview_2d()

```
variable matrix_preview_2d (
             wave data,
             wave metadata )
```

Definition at line 145 of file pearl-matrix-import.ipf.

12.16.2.8 mtrx_create_folder()

```
dfr mtrx_create_folder (
             string fileName,
             dfref df_base = defaultValue )
```

create or look up a data folder based on a matrix file name.

the name of the folder is mtrx_date_time, where date and time are parsed from the file name. for this to work, the file name must consist of at least three parts that are separated by dash or underscore. the second (third) part contains the date (time). date and time are copied as strings.

if the data folder exists, a reference to the existing folder is returned.

Parameters

fileName	name of the result or data file.
df_base	(optional) base data folder. default: current folder.

Returns

reference of the newly created or existing data folder.

Definition at line 360 of file pearl-matrix-import.ipf.

12.16.2.9 mtrx_file_brickletID()

```
variable mtrx_file_brickletID (
             string resultFile,
             variable runCycle,
             variable scanCycle,
             string channel )
```

find out bricklet ID of a file

Warning

EXPERIMENTAL the code of this function is inefficient. the function may be removed in a later version.

Parameters

resultFile	base name of result file without chain link number and extension. as returned by mtrx_parse_filename.
runCycle	requested run cycle. 0 = first available.
scanCycle	requested scan cycle. 0 = first available.
channel	channel name. for example: "I", "Z", "Aux(V)", etc. empty string: first available.

Returns

bricklet ID, or -1 if an error occurred.

Definition at line 461 of file pearl-matrix-import.ipf.

12.16.2.10 mtrx_get_cycle_folder()

```
dfr mtrx_get_cycle_folder (
             dfref df_base = defaultValue,
             variable runCycle = defaultValue,
             variable scanCycle = defaultValue )
```

create a data folder for bricklet data.

the name of the folder is, for example "r23s2" where the first (second) number is the run (scan) cycle. run cycle and scan cycle numbers are taken from the open matrix file unless overridden by optional arguments.

if the data folder exists, a reference to the existing folder is returned. if one of the run or scan cycle numbers is lower than 1, the base folder is returned.

Parameters

df_base	(optional) base data folder. default: current folder.
runCycle	(optional) run cycle number. must be >= 1. default: from last mtrx_open_file call.
scanCycle	(optional) scan cycle number. must be >= 1. default: from last mtrx_open_file call.

Returns

reference of the newly created or existing data folder.

Definition at line 405 of file pearl-matrix-import.ipf.

12.16.2.11 mtrx_load_all()

```
variable mtrx_load_all ( )
```

load all data from a Matrix data file.

Definition at line 231 of file pearl-matrix-import.ipf.

12.16.2.12 mtrx_load_file()

```
string mtrx_load_file (
             string pathName,
             string fileName,
             string traces = defaultValue )
```

load all data from a Matrix data file.

the data wave is loaded into a sub-subfolder the current data folder. the relative path has the format ":mtrx_{date}↩ _{time}:r{run_cycle}s{scan_cycle}", where the parameters {date}, {time}, {run_cycle} and {scan_cycle} are copied from the file name. the file name must be formatted according to the specifications set out below.

Parameters

pathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
fileName	if empty a dialog box shows up the file name must adhere to the format "{prefix}-{date}-{time}-{anything}--{run_cycle}_{scan_cycle}.{extension}". the first three seperators can alternatively be underscores. it may be necessary to change the configuration of the Matrix application.
traces	(currently not used) semicolon-separated list of preferred traces. the items of the list are match strings for the Igor StringMatch function. only matching traces are loaded from the file. default: "∗Up;∗Down;∗ReUp;∗ReDown;"

Returns

semicolon-separated list of loaded waves including partial path from current data folder.

Definition at line 767 of file pearl-matrix-import.ipf.

12.16.2.13 mtrx_load_info()

```
string mtrx_load_info (
             string APathName,
             string AFileName )
```

load descriptive info from a Matrix data file.

the info string lists the following information for each scan contained in the file:

- path of the scan group inside the file.
- number of scan positions.
- dataset names of scan positioners.
- dataset names of detectors.

Parameters

APathName	igor symbolic path name. can be empty if the path is specified in AFileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up

Returns

newline terminated string.

Definition at line 863 of file pearl-matrix-import.ipf.

12.16.2.14 mtrx_load_preview()

```
string mtrx_load_preview (
             string destName,
             string pathName,
             string fileName,
             string traces = defaultValue )
```

load a preview image from a Matrix data file.

the data wave is loaded into the current data folder.

Parameters

destName	destination wave name. the wave is created in the current data folder.
pathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
fileName	if empty a dialog box shows up the file name must adhere to the format "{prefix}-{date}-{time}-{anything}--{run_cycle}_{scan_cycle}.{extension}". the first three seperators can alternatively be underscores. it may be necessary to change the configuration of the Matrix application.
traces	(currently not used) semicolon-separated list of preferred traces. the items of the list are match strings for the Igor StringMatch function. only the first matching trace is loaded from the file. default: "∗Up;∗Down;∗ReUp;∗ReDown;"

Returns

semicolon-separated list of loaded waves including partial path from current data folder.

Definition at line 682 of file pearl-matrix-import.ipf.

12.16.2.15 mtrx_open_file()

```
variable mtrx_open_file (
```

```
            string pathName,
            string fileNameOrPath )
```

open a matrix result or data file

this function opens a matrix result file (.mtrx) or data file (.*_mtrx).

if a data file is selected, the function locates the corresponding result file, opens it, and looks up the bricklet ID of the data file. if a result file is selected, the function opens it but does not look up bricklet IDs.

the result file remains open and can be accessed using the mtrx_ functions or MFR_ operations. once a result file is open, you can easily access any bricklets linked to it, i.e., any run cycle, scan cycle, and channel.

the function stores information about the opened file in a global package data folder. if the same result file is opened again later, the information is reused and the file not read again. this may cause problems if the file has been modified in the meantime, or if the cached data become corrupt for some reason. the function detects if a data file is not linked in the open result file, and updates the cache. in other situations it may be necessary to force a reload.

Todo fix possible cache issues, add an option to override the cache.

Parameters

pathName	igor path name or empty string.
fileName	file name, with or without path, or empty string.

Returns

file type

- 0 result file (logbook)
- 1 result data file (bricklet)
- -1 error, no data loaded
- -2 matrixfilereader.xop not installed

Definition at line 545 of file pearl-matrix-import.ipf.

12.16.2.16 mtrx_parse_filename()

```
variable mtrx_parse_filename (
            string fileName,
            string * resultFile,
            variable * runCycle,
            variable * scanCycle,
            string * channel )
```

parse matrix file names

parse matrix file names for result name, run cycle, scan cycle, and channel.

Parameters

fileName	matrix result or data file name (without path).
resultFile	(out) base name of the result file. append "_%04u.mtrx" to get the actual result file. we do not know the chain link number at this stage.
runCycle	(out) run cycle number. necessary to look up the bricklet ID.
scanCycle	(out) scan cycle number. necessary to look up the bricklet ID.
channel	(out) channel name.

Returns

file type

- 0 result file (logbook)
- 1 result data file (bricklet)

result file names look like: default_2015Apr20-124353_STM-STM_AtomManipulation_0001.mtrx, default_↩ 2015Apr20-124353_STM-STM_AtomManipulation_0002.mtrx, etc. the function returns the first part up to the experiment name ("AtomManipulation" in the examples). all other return values set to defaults and must not be regarded.

result data files look like: default_2015Apr20-124353_STM-STM_AtomManipulation–136_1.Aux1(V)_mtrx, default_2015Apr20-124353_STM-STM_AtomManipulation–136_1.I(V)_mtrx, default_2015Apr20-124353_STM-STM_AtomManipulation–14_1.I_mtrx, default_2015Apr20-124353_STM-STM_AtomManipulation–14_1.Z_mtrx, etc. the function returns all results as described in the parameter list.

Definition at line 294 of file pearl-matrix-import.ipf.

12.16.2.17 mtrx_scale_dataset()

```
variable mtrx_scale_dataset (
             wave data )
```

Definition at line 829 of file pearl-matrix-import.ipf.

12.16.2.18 mtrx_split_filename()

```
string mtrx_split_filename (
             string fileName,
             string * prefix,
             string * datepart,
             string * timepart )
```

split a matrix filename and return the first three parts

we assume that the second (third) part contains the date (time). the parts are separated by dash or underscore.

Definition at line 332 of file pearl-matrix-import.ipf.

12.16.2.19 preview_matrix_file()

```
static wave preview_matrix_file (
             string filename )  [static]
```

load the preview of a Matrix data file

the preview is loaded to the preview_image wave in the pearl_explorer data folder.

the s_file_info string is updated with information about the scan dimensions.

Parameters

filename	name of a file in the directory specified by the pearl_explorer_filepath path object.

Returns

wave reference of the preview image

Definition at line 170 of file pearl-matrix-import.ipf.

12.16.2.20 subtract_line_bg()

```
variable subtract_line_bg (
             wave img )
```

remove linear background line-by-line

Definition at line 886 of file pearl-matrix-import.ipf.

12.16.3 Variable Documentation

12.16.3.1 ks_filematch_mtrx

```
const string ks_filematch_mtrx = "*_mtrx"  [static]
```

Definition at line 40 of file pearl-matrix-import.ipf.

12.16.3.2 package_name

```
const string package_name = "pearl_matrix_import"  [static]
```

Definition at line 37 of file pearl-matrix-import.ipf.

12.16.3.3 package_path

```
const string package_path = "root:packages:pearl_matrix_import:"  [static]
```

Definition at line 38 of file pearl-matrix-import.ipf.

12.17 pearl-menu.ipf File Reference

Functions

- string PearlMenuEnableFunc (string funcname)

  check whether a function name exists
- variable LoadPearlOptics ()
- variable LoadPearlArpes ()
- variable LoadPearlPreparation ()
- variable Display2dProfiles ()
- variable Display3dSlicer ()
- variable DisplayGizmoSlicer ()
- variable PearlLiveDisplay (string epicsname, string nickname, string wbRGB)

area detector live display

- variable PearlCameraDisplay (string epicsname, string nickname, string wbRGB)

  area detector surveillance camera display
- variable PearlAnglescanTracker (string epicsname, string wbRGB)

  display the angle scan tracker window
- variable PearlSampleTracker (variable action)

  display the sample tracker window

12.17.1 Function Documentation

12.17.1.1 Display2dProfiles()

```
variable Display2dProfiles ( )
```

Definition at line 116 of file pearl-menu.ipf.

12.17.1.2 Display3dSlicer()

```
variable Display3dSlicer ( )
```

Definition at line 133 of file pearl-menu.ipf.

12.17.1.3 DisplayGizmoSlicer()

```
variable DisplayGizmoSlicer ( )
```

Definition at line 152 of file pearl-menu.ipf.

12.17.1.4 LoadPearlArpes()

```
variable LoadPearlArpes ( )
```

Definition at line 104 of file pearl-menu.ipf.

12.17.1.5 LoadPearlOptics()

```
variable LoadPearlOptics ( )
```

Definition at line 98 of file pearl-menu.ipf.

12.17.1.6 LoadPearlPreparation()

```
variable LoadPearlPreparation ( )
```

Definition at line 110 of file pearl-menu.ipf.

12.17.1.7 PearlAnglescanTracker()

```
variable PearlAnglescanTracker (
            string epicsname,
            string wbRGB )
```

display the angle scan tracker window

Parameters

epicsname	base name of the detector, e.g. X03DA-SCIENTA: image1: and cam1: are appended by the function. see ad_connect().
wbRGB	window background color, e.g. "(32768,49152,55296)"

Definition at line 242 of file pearl-menu.ipf.

12.17.1.8 PearlCameraDisplay()

```
variable PearlCameraDisplay (
            string epicsname,
            string nickname,
            string wbRGB )
```

area detector surveillance camera display

display an area detector channel in a simple image window without any interactive controls.

Parameters

epicsname	base name of the detector, e.g. X03DA-SCIENTA: image1: and cam1: are appended by the function. see ad_connect().
nickname	nick name under which this detector is referred to in Igor. must be a valid name for a data folder. see ad_connect().
wbRGB	window background color, e.g. "(32768,49152,55296)"

Definition at line 218 of file pearl-menu.ipf.

12.17.1.9 PearlLiveDisplay()

```
variable PearlLiveDisplay (
            string epicsname,
            string nickname,
            string wbRGB )
```

area detector live display

display an area detector channel in an ad_display_profiles() window.

Parameters

epicsname	base name of the detector, e.g. X03DA-SCIENTA: image1: and cam1: are appended by the function. see ad_connect().
nickname	nick name under which this detector is referred to in Igor. must be a valid name for a data folder. see ad_connect().

Parameters

wbRGB	window background color, e.g. "(32768,49152,55296)"

Definition at line 185 of file pearl-menu.ipf.

12.17.1.10 PearlMenuEnableFunc()

```
string PearlMenuEnableFunc (
            string funcname )
```

check whether a function name exists

return a prefix which disables the menu item if the function does not exist

Definition at line 89 of file pearl-menu.ipf.

12.17.1.11 PearlSampleTracker()

```
variable PearlSampleTracker (
            variable action )
```

display the sample tracker window

Parameters

epicsname	base name of the detector, e.g. X03DA-SCIENTA: image1: and cam1: are appended by the function. see ad_connect().
wbRGB	window background color, e.g. "(32768,49152,55296)"

Definition at line 261 of file pearl-menu.ipf.

12.18 pearl-otf-import.ipf File Reference

Functions

- variable otf_load_itx_all (string pathname)
- variable otf_load_itx_match (string pathname, string matchstr)
- variable otf_load_itx (string pathname, string filename)
- variable otf_gather_iterator (dfref df, string ∗sdata)
- variable otf_gather_batch (string ywavematch, string xwavematch, string destfolder)
- variable gather_batch (string foldermatch, string ywavematch, string xwavematch, string destfolder)
- variable otf_rename_folders_iterator (dfref df, string ∗sdata)
- variable otf_rename_folders (string pattern, variable unique_index=defaultValue, string new_suffix=default↩ Value, string match_str=defaultValue)
- variable otf_interp (variable e1, variable e2, variable npts, variable smo)
- variable otf_smo_int (wave win, wave wout, wave wpe, variable smo)

12.18.1 Function Documentation

12.18.1.1 gather_batch()

```
variable gather_batch (
             string foldermatch,
             string ywavematch,
             string xwavematch,
             string destfolder )
```

Definition at line 162 of file pearl-otf-import.ipf.

12.18.1.2 otf_gather_batch()

```
variable otf_gather_batch (
             string ywavematch,
             string xwavematch,
             string destfolder )
```

Definition at line 152 of file pearl-otf-import.ipf.

12.18.1.3 otf_gather_iterator()

```
variable otf_gather_iterator (
             dfref df,
             string * sdata )
```

Definition at line 106 of file pearl-otf-import.ipf.

12.18.1.4 otf_interp()

```
variable otf_interp (
             variable e1,
             variable e2,
             variable npts,
             variable smo )
```

Definition at line 271 of file pearl-otf-import.ipf.

12.18.1.5 otf_load_itx()

```
variable otf_load_itx (
             string pathname,
             string filename )
```

Definition at line 76 of file pearl-otf-import.ipf.

12.18.1.6 otf_load_itx_all()

```
variable otf_load_itx_all (
             string pathname )
```

Definition at line 31 of file pearl-otf-import.ipf.

12.18.1.7 otf_load_itx_match()

```
variable otf_load_itx_match (
            string pathname,
            string matchstr )
```

Definition at line 53 of file pearl-otf-import.ipf.

12.18.1.8 otf_rename_folders()

```
variable otf_rename_folders (
            string pattern,
            variable unique_index = defaultValue,
            string new_suffix = defaultValue,
            string match_str = defaultValue )
```

Definition at line 231 of file pearl-otf-import.ipf.

12.18.1.9 otf_rename_folders_iterator()

```
variable otf_rename_folders_iterator (
            dfref df,
            string * sdata )
```

Definition at line 185 of file pearl-otf-import.ipf.

12.18.1.10 otf_smo_int()

```
variable otf_smo_int (
            wave win,
            wave wout,
            wave wpe,
            variable smo )
```

Definition at line 304 of file pearl-otf-import.ipf.

12.19 pearl-pmsco-import.ipf File Reference

data import/export procedures for multiple scattering calculations.

Namespaces

- PearlPmscoImport

  data import/export procedures for multiple scattering calculations.

Functions

- string pmsco_save_scan (string pathname, string filename, string energy, string theta, string phi, string alpha, string intensity, string sigma, dfref sdfr=defaultValue)

  save waves in a PMSCO scan data file.
- static string save_scan_helper (string destname, string value, wave template, dfref destdfr, string wavenames)

  helper function for save_pmsco_scan()
- string pmsco_load_xyz (string pathname, string filename)

  load an xyz cluster file

12.19.1 Detailed Description

data import/export procedures for multiple scattering calculations.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.19.2 Function Documentation

12.19.2.1 pmsco_load_xyz()

```
string pmsco_load_xyz (
            string pathname,
            string filename )
```

load an xyz cluster file

load an xyz cluster file into the current data folder the wave names are at (atom types), xx, yy, and zz. at is a text wave containing chemical symbols. existing waves are overwritten.

Parameters

pathname	name of igor symbolic path. can be empty (path is taken from filename argument).
filename	file system path. can be empty (will open dialog).

Definition at line 188 of file pearl-pmsco-import.ipf.

12.19.2.2 pmsco_save_scan()

```
string pmsco_save_scan (
            string pathname,
            string filename,
```

```
    string energy,
    string theta,
    string phi,
    string alpha,
    string intensity,
    string sigma,
    dfref sdfr = defaultValue )
```

save waves in a PMSCO scan data file.

Warning

experimental. this function is work in progress.

cases

- phd scan: separate energy, theta, phi, alpha, intensity waves
- hemi scan: separate energy, theta, phi, intensity waves
- polar/azi scan: intensity wave, angle is in x scale

options

- sigma wave

the data arguments are strings and can be

- the name of an existing wave (optionally including a path relative to the specified source folder),
- the string representation of a constant numeric value,
- a dimension specifier ("x", "y", "z" or "t") referring to the dimension scale of the intensity wave, or
- an empty string if the corresponding axis should not be saved.

wave names can include a path relative to the specified source data folder. by default, the function looks in the folder specified by the sdfr argument.

Parameters

pathname	name of igor symbolic path to destination folder. prompt user if empty.
filename	requested file name. prompt user if empty.

Note

the extension should include the symbols of the included parameters in the order "etpais". if the intensity wave contains a modulation function, ".modf" should be inserted before the extension. in interactive mode, igor tends to override the file extension with a standard one like ".txt".

Parameters

energy	energy specification. see description above.
theta	theta specification. see description above.
phi	phi specification. see description above.
alpha	alpha specification. see description above.
intensity	name of intensity (or modulation) wave. this parameter is mandatory and must refer to an existing wave.
sigma	sigma specification. see description above.

Parameters

sdfr	source data folder reference. default: current data folder.

Returns

file name

Definition at line 88 of file pearl-pmsco-import.ipf.

12.19.2.3 save_scan_helper()

```
static string save_scan_helper (
             string destname,
             string value,
             wave template,
             dfref destdfr,
             string wavenames )  [static]
```

helper function for save_pmsco_scan()

Definition at line 126 of file pearl-pmsco-import.ipf.

12.20 pearl-polar-coordinates.ipf File Reference

Functions

- variable cart2polar (variable xx, variable yy, variable zz, variable ∗radius, variable ∗theta, variable ∗phi)
- variable cart2polar_wave (wave in, wave out)
- variable polar2cart (variable radius, variable theta, variable phi, variable ∗xx, variable ∗yy, variable ∗zz)
- variable polar2cart_wave (wave in, wave out)
- variable polar_distance (variable polar1, variable azim1, variable polar2, variable azim2)

12.20.1 Function Documentation

12.20.1.1 cart2polar()

```
variable cart2polar (
             variable xx,
             variable yy,
             variable zz,
             variable * radius,
             variable * theta,
             variable * phi )
```

Definition at line 10 of file pearl-polar-coordinates.ipf.

12.20.1.2 cart2polar_wave()

```
variable cart2polar_wave (
             wave in,
             wave out )
```

Definition at line 36 of file pearl-polar-coordinates.ipf.

12.20.1.3 polar2cart()

```
variable polar2cart (
             variable radius,
             variable theta,
             variable phi,
             variable * xx,
             variable * yy,
             variable * zz )
```

Definition at line 48 of file pearl-polar-coordinates.ipf.

12.20.1.4 polar2cart_wave()

```
variable polar2cart_wave (
             wave in,
             wave out )
```

Definition at line 58 of file pearl-polar-coordinates.ipf.

12.20.1.5 polar_distance()

```
variable polar_distance (
             variable polar1,
             variable azim1,
             variable polar2,
             variable azim2 )
```

Definition at line 69 of file pearl-polar-coordinates.ipf.

12.21 pearl-pshell-import.ipf File Reference

import data from PShell

```
#include <HDF5 Browser>
#include "pearl-compat"
#include "pearl-gui-tools"
#include "pearl-area-import"
```

Namespaces

- PearlPShellImport
  import data from PShell

Functions

- variable psh5_open_file (string ANickName, string APathName, string AFileName)

  open a HDF5 file created by the PShell data acquisition program and prepare the data folder.
- variable psh5_close_file (variable fileID)

  close a HDF5 file opened by psh5_open_file.
- string psh5_load_complete (string ANickName, string APathName, string AFileName, variable load_↩ data=defaultValue, variable load_attr=defaultValue)

  load everything from a PShell data file.
- string psh5_load_preview (string APathName, string AFileName, variable load_data=defaultValue, variable load_attr=defaultValue, string pref_scans=defaultValue, string pref_datasets=defaultValue)

  load a preview image from a PShell data file.
- string psh5_load_scan_complete (variable fileID, string scanpath, variable load_data=defaultValue, variable load_attr=defaultValue)

  load all data of a selected scan from a PShell data file.
- string psh5_list_scans (variable fileID)

  list scan groups of a PShell data file.
- string psh5_list_scan_datasets (variable fileID, string scanpath, variable include_regions=defaultValue)

  list datasets of a PShell scan group.
- string psh5_list_scan_regions (variable fileID, string scanpath)

  list regions of a PShell scan group.
- string psh5_load_scan_data (variable fileID, string scanpath)

  load all datasets of a PShell scan group.
- string psh5_load_scan_attrs (variable fileID, string scanpath, variable attr_sets=defaultValue)

  load attributes of a PShell scan group.
- string psh5_load_scan_meta (variable fileID, string scanpath)

  load metadata of a PShell scan group.
- string psh5_load_dataset (variable fileID, string scanpath, string datasetname, variable set_scale=default↩ Value)

  load a dataset from an open PShell HDF5 file.
- static string select_dataset (string file_datasets, string pref_datasets)

  select the preferred dataset from a list of available datasets.
- string psh5_load_scan_preview (variable fileID, string scanpath, variable set_scale=defaultValue, string pref_datasets=defaultValue)

  load a preview dataset from an open PShell HDF5 file.
- string psh5_load_scan_section (variable fileID, string scanpath, variable dim, variable set_scale=default↩ Value, string pref_datasets=defaultValue)

  load a longitudinal section of a scan from an open PShell HDF5 file.
- variable psh5_load_dataset_meta (variable fileID, string datapath, string datasetname, wave datawave)

  load metadata of a PShell dataset.
- string psh5_load_dataset_slabs (variable fileID, string datapath, string datasetname, variable progress=default↩ Value)

  load a dataset slab-wise from the open PShell HDF5 file.
- string psh5_load_dataset_slab (variable fileID, string datapath, string datasetname, variable dim2start, variable dim2count, variable dim3start, variable dim3count)

  load a single image from the open PShell data file.
- variable ps_set_dimlabels (wave data)

  set dimension labels according to the axis type
- variable ps_set_dimlabels2 (wave data, string name)

  set dimension labels according to the axis type
- static dfr find_scan_folder (dfref dataDF)

  find the scan folder

- static dfr find_attr_folder (dfref dataDF)

  find the attributes data folder
- variable ps_scale_datasets ()

  set the dimension scales of loaded PShell Scienta datasets according to attributes.
- variable ps_scale_dataset (wave data)

  set the dimension scales of a loaded PShell Scienta dataset according to attributes.
- static wave find_scale_wave (string name, dfref dataDF, dfref scanDF, dfref attrDF)
- variable ps_detect_scale (wave ax, wave lo, wave hi, wave un)

  detect the dimension scales from attributes.
- variable ps_scale_dataset_2 (wave data, wave ax, wave lo, wave hi, wave un)

  set the dimension scales of a dataset.
- string psh5_load_reduced (string ANickName, string APathName, string AFileName, funcref reduction_func, string reduction_param, variable progress=defaultValue, variable nthreads=defaultValue)

  load and reduce the ScientaImage dataset of the first scan of a PShell data file.
- string psh5_load_dataset_reduced (variable fileID, string scanpath, string datasetname, funcref reduction_↩ func, string reduction_param, variable progress=defaultValue, variable nthreads=defaultValue)

  load a reduced dataset from the open PShell HDF5 file.
- static threadsafe variable reduce_slab_worker (funcref reduction_func)
- static threadsafe wave reduce_slab_image (wave slabdata, wave image, funcref reduction_func, string reduction_param)
- string psh5_load_info (string APathName, string AFileName)

  load descriptive info from a PShell data file.
- string psh5_load_scan_info (variable fileID, string scanpath)

  load descriptive info from a PShell scan.
- static string twave2list (wave wt, string sep)

  convert text wave to list.
- static string wave2list (wave w, string format, string sep)

  convert numeric wave to list.

Variables

- const string kEnergyDimLabel = "energy"

  Dimension label for the energy dispersive dimension of multi-dimensional datasets.
- const string kAngleDimLabel = "angle"

  Dimension label for the angle dispersive dimension of multi-dimensional datasets.
- const string kScanDimLabel = "scan"

  Dimension label for the scan dimension of multi-dimensional datasets.
- const string kDataDimLabel = "data"

  Dimension label for the data dimension.
- const string kPreviewDatasets = "ScientaImage;ScientaSpectrum;ImageAngleDistribution;ImageEnergy↩ Distribution;Counts;SampleCurrent;"

  List of preferred datasets to load for preview.
- const string kScientaScalingDatasets = "LensMode;ScientaChannelBegin;ScientaChannelEnd;Scienta↩ SliceBegin;ScientaSliceEnd;"

  List of datasets that must be loaded to determine the axis scaling of a Scienta image.
- const string kTransposedDatasets = "ScientaImage;"

  List of datasets that should be transposed upon loading.
- const variable kDetectorSensitivity = 1

  multiply scienta detector intensity by this value to get actual counts.

12.21.1 Detailed Description

import data from PShell

HDF5 file import from the PShell data acquisition program. the main import functions are:

- psh5_load_complete() load all scans and datasets from a file.
- psh5_load_reduced() load the ScientaImage dataset of the first scan and reduce its dimensionality.
- psh5_load_scan_complete() load all datasets of a selected scan.
- psh5_load_scan_preview() load a preview of a selected scan.
- psh5_load_dataset() load a selected dataset.
- psh5_load_dataset_reduced() load a selected dataset and reduce its dimensionality.

the following helper functions are also needed:

- psh5_open_file()
- psh5_close_file()
- psh5_list_scans()
- psh5_list_scan_datasets()
- psh5_load_scan_meta()
- psh5_load_scan_attrs()

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.21.2 Function Documentation

12.21.2.1 find_attr_folder()

```
static dfr find_attr_folder (
            dfref dataDF )  [static]
```

find the attributes data folder

this is the :attr folder.

Definition at line 1476 of file pearl-pshell-import.ipf.

12.21.2.2 find_scale_wave()

```
static wave find_scale_wave (
             string name,
             dfref dataDF,
             dfref scanDF,
             dfref attrDF )  [static]
```

Definition at line 1564 of file pearl-pshell-import.ipf.

12.21.2.3 find_scan_folder()

```
static dfr find_scan_folder (
             dfref dataDF )  [static]
```

find the scan folder

the scan folder is the one that contains the :attr folder the data and scan folders may refer to the same folder.

Definition at line 1459 of file pearl-pshell-import.ipf.

12.21.2.4 ps_detect_scale()

```
variable ps_detect_scale (
             wave ax,
             wave lo,
             wave hi,
             wave un )
```

detect the dimension scales from attributes.

the function checks the data , scan and attributes folders for scan parameters. the results are written to the provided waves. the function is normally called by ps_scale_datasets() but can also be used independently.

the current datafolder must be the data or the scan folder. the data folder contains the waves that are to be scaled. the scan folder contains the scan positions and the :attr folder.

the provided waves are redimensioned by the function, and dimension labels are set. the scale parameters can then be extracted by keyword, e.g.,

- `lo[%energy]` analyser energy dimension.
- `lo[%angle]` analyser angle dimension.
- `lo[%scan]` scan dimension.
- `lo[%data]` data dimension.

the function tries to read the following waves, in the data, scan, and attributes folders, where the first folder in the list takes precedence. it may fall back to more or less reasonable default values if no data is not found.

- `LensMode`
- `ScientaChannelBegin`
- `ScientaChannelEnd`
- `ScientaSliceBegin`
- `ScientaSliceEnd`

- `ScanWritables`
- wave referenced by `ScanWritables[0]`

Parameters

ax	text wave to receive the axis labels.
lo	wave to receive the lower limits.
hi	wave to receive the upper limits.
un	text wave to receive the unit labels.

Returns

the function results are written to the lo, hi, un, and ax waves.

Version

this function supports regions from version 1.03. check that you're in the correct data folder!

Definition at line 1621 of file pearl-pshell-import.ipf.

12.21.2.5 ps_scale_dataset()

```
variable ps_scale_dataset (
             wave data )
```

set the dimension scales of a loaded PShell Scienta dataset according to attributes.

the current datafolder must contain the :attr folder. the data wave can be in the current folder or a sub-folder.

the dimension labels of the dataset waves must have been set correctly, e.g. by ps_set_dimlabels(). this is implicitly done by the high-level load functions.

the function is useful if a single dataset is loaded and scaled. if multiple datasets are loaded, ps_scale_datasets() is slightly more efficient.

Parameters

data	data wave to be scaled. dimension labels (index -1) must be set correctly, cf. ps_set_dimlabels().

Version

this function supports regions from version 1.03.

Definition at line 1550 of file pearl-pshell-import.ipf.

12.21.2.6 ps_scale_dataset_2()

```
variable ps_scale_dataset_2 (
             wave data,
             wave ax,
             wave lo,
```

```
wave hi,
wave un )
```

set the dimension scales of a dataset.

the function is normally called by ps_scale_datasets() but can also be used independently. the limits and units must be given as function arguments with proper dimension labels.

the provided limit and unit waves must have dimension labels matching the -1 index dimension labels of the data wave, such as set by the ps_detect_scale() function. the scale parameters are extracted by keyword, e.g.,

- `lo[%energy]` analyser energy dimension.
- `lo[%angle]` analyser angle dimension.
- `lo[%scan]` scan dimension.
- `lo[%data]` data dimension.

if the data dimension labels and units are at their defaults ("value" and "arb.", respectively), the function tries to read them from the existing wave note ("AxisLabelD" and "AxisUnitD"), or based on the wave name if the name is one of the known measurement variables: "ScientaImage", "ImageAngleDistribution", "ScientaAngleDistribution", "↩ ScientaSpectrum", "ImageEnergyDistribution", "ScientaEnergyDistribution", "SampleCurrent", "RefCurrent", "Aux↩ Current", "MachineCurrent".

Parameters

data	data wave to be scaled. dimension labels (index -1) must be set to match the limit waves.
ax	axis labels. the axis labels are written to the wave note in the format `AxisLabel%s=%s` where `X`, `Y`, `Z`, `D` is substituted for the first place holder and the label for the second one.
lo	lower limits. the lower limits are applied using the SetScale operation.
hi	upper limits. the upper limits are applied using the SetScale operation.
un	unit labels. the unit labels are applied using the SetScale operation.

Version

this function supports regions from version 1.03.

Definition at line 1781 of file pearl-pshell-import.ipf.

12.21.2.7 ps_scale_datasets()

```
variable ps_scale_datasets ( )
```

set the dimension scales of loaded PShell Scienta datasets according to attributes.

datasets listed in the ScanReadables waves are scaled according to the attribute waves in the data, scan, and attributes folders, whichever is found first.

the current datafolder must contain the ScanReadables wave and the :attr folder. the ScanReadables text wave contains names of the waves to scale. wave names can include a relative path to a sub-folder. the path separator is "/".

the dimension labels of the dataset waves must have been set correctly, e.g. by ps_set_dimlabels(). this is implicitly done by the high-level load functions.

Version

this function supports regions from version 1.03. check that you're in the correct data folder!

Definition at line 1504 of file pearl-pshell-import.ipf.

12.21.2.8 ps_set_dimlabels()

```
variable ps_set_dimlabels (
             wave data )
```

set dimension labels according to the axis type

this function asserts a particular ordering of dimensions types based on the name of the wave for ScientaImage, ScientaSpectrum, ImageAngleDistribution, ImageEnergyDistribution. all other waves must be one-dimensional, and the dimension must be the scan dimension.

dimension labels are required by scaling functions.

Parameters

data	data wave as loaded from PShell file

Returns

- 0 all labels set successfully.
- 1 unidentified data source.
- 2 wave does not contain data.

Definition at line 1383 of file pearl-pshell-import.ipf.

12.21.2.9 ps_set_dimlabels2()

```
variable ps_set_dimlabels2 (
             wave data,
             string name )
```

set dimension labels according to the axis type

same as ps_set_dimlabels() except that the dimension labels are set according to a separate name argument instead of the wave name.

Parameters

data	data wave as loaded from PShell file.
name	original name of the dataset in the PShell file.

Returns

- 0 all labels set successfully.
- 1 unidentified data source.
- 2 wave does not contain data.

Definition at line 1402 of file pearl-pshell-import.ipf.

12.21.2.10 psh5_close_file()

```
variable psh5_close_file (
            variable fileID )
```

close a HDF5 file opened by psh5_open_file.

this function just closes the HDF5 file. no change is made to the loaded data.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().

Definition at line 140 of file pearl-pshell-import.ipf.

12.21.2.11 psh5_list_scan_datasets()

```
string psh5_list_scan_datasets (
            variable fileID,
            string scanpath,
            variable include_regions = defaultValue )
```

list datasets of a PShell scan group.

the function returns a list of all datasets of the selected scan. this does not include datasets from the attributes sub-group.

Note

in a future version, an option may be introduced to filter datasets by function (*Readable* and/or *Writable*).

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".

Returns

semicolon-separated list of dataset paths.

Version

since version 1.03 this function returns paths relative to scanpath.

Definition at line 440 of file pearl-pshell-import.ipf.

12.21.2.12 psh5_list_scan_regions()

```
string psh5_list_scan_regions (
            variable fileID,
            string scanpath )
```

list regions of a PShell scan group.

the function returns a list of all region groups of the selected scan.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".

Returns

semicolon-separated list of datagroup paths.

Definition at line 481 of file pearl-pshell-import.ipf.

12.21.2.13 psh5_list_scans()

```
string psh5_list_scans (
             variable fileID )
```

list scan groups of a PShell data file.

the function returns a list of all top-level groups whose name starts with "scan".

Parameters

fileID	ID of open HDF5 file from psh5_open_file().

Returns

semicolon-separated list of group paths.

Definition at line 405 of file pearl-pshell-import.ipf.

12.21.2.14 psh5_load_complete()

```
string psh5_load_complete (
             string ANickName,
             string APathName,
             string AFileName,
             variable load_data = defaultValue,
             variable load_attr = defaultValue )
```

load everything from a PShell data file.

Parameters

ANickName	destination folder name (top level under root)
APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up

Parameters

load_data	select whether datasets (positioners and detectors) are loaded. • 1 (default) load data. • 0 do not load data.
load_attr	select whether attributes (auxiliary device readbacks) are loaded. for proper wave scaling, the attributes must be loaded. • 1 (default) load attributes. • 0 do not load attributes.

Returns

complete path of the loaded file if successful. empty string otherwise.
global string s_filepath in new data folder contains the full file path on disk.
global string s_scanpaths in new data folder contains a list of scan groups inside the file.

Definition at line 170 of file pearl-pshell-import.ipf.

12.21.2.15 psh5_load_dataset()

```
string psh5_load_dataset (
             variable fileID,
             string scanpath,
             string datasetname,
             variable set_scale = defaultValue )
```

load a dataset from an open PShell HDF5 file.

if the dataset has a maximum of two dimensions, the function loads it at once. if it has more than two dimension, the function calls psh5_load_dataset_slabs() to load the data slab by slab.

- the metadata (HDF5 attributes) are loaded into the wave note, cf. psh5_load_dataset_meta().
- dimension labels are set according the dataset name, cf. ps_set_dimlabels().
- wave scaling is set if the necessary scan attributes have been loaded and the `set_scale` option is selected (default). the attributes must be loaded by psh5_load_scan_meta() and psh5_load_scan_attrs() (attr_sets=2).

the dataset is loaded into the current data folder unless datasetname contains a region specifier. in the latter case, the dataset is loaded into sub-folder with the name of the region. the function returns from the original data folder.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".
datasetname	name of the dataset. the name of the loaded wave is a cleaned up version of the dataset name. the name can include the region name as a relative path, e.g. "region1/ScientaSpectrum". in this case, the dataset is loaded into a sub-folder named "region1".

Parameters

set_scale	by default, the function tries to set the wave scaling if the attributes have been loaded. if multiple datasets are loaded from a file, it is more efficient to set the scaling of all loaded datasets at the end by calling ps_scale_datasets(). • 1 (default) set the wave scaling. • 0 do not set the wave scaling.

Returns

name of loaded wave if successful. empty string otherwise.

Version

this function supports regions as of version 1.03.

Definition at line 706 of file pearl-pshell-import.ipf.

12.21.2.16 psh5_load_dataset_meta()

```
variable psh5_load_dataset_meta (
             variable fileID,
             string datapath,
             string datasetname,
             wave datawave )
```

load metadata of a PShell dataset.

"metadata" are the HDF5 attributes attached to the scan dataset.

data is added to the wave note.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
datapath	path to the containing group in the HDF5 file. path separator is the slash "/".
datasetname	name of the dataset. may include relative path.
datawave	metadata is added to the wave note of this wave.

Returns

0 if successful, non-zero if an error occurred.

Definition at line 1091 of file pearl-pshell-import.ipf.

12.21.2.17 psh5_load_dataset_reduced()

```
string psh5_load_dataset_reduced (
             variable fileID,
             string scanpath,
             string datasetname,
```

```
funcref reduction_func,
string reduction_param,
variable progress = defaultValue,
variable nthreads = defaultValue )
```

load a reduced dataset from the open PShell HDF5 file.

the function loads the dataset image by image using the hyperslab option and applies a custom reduction function to each image. the results from the reduction function are written to the `ReducedData1`, `ReducedData2`, etc. waves. the raw data are discarded.

by default, the reduction function is called in separate threads to reduce the total loading time. (see the global variable psh5_perf_secs which reports the total run time of the function.) the effect varies depending on the balance between file loading (image size) and data processing (complexity of the reduction function). for debugging the reduction function, multi-threading can be disabled.

if the reduction function requires the image waves to be scaled properly, the attributes must have been loaded by psh5_load_scan_attrs() before. in this case, the scales of the result waves are also set by the function. otherwise, the results can also be scaled by ps_scale_dataset() later.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to scan group in the HDF5 file.
datasetname	name of the dataset. this must currently be "ScientaImage", other data is not supported. the name of the loaded wave is a cleaned up version of the dataset name. the name can include the region name as a relative path, e.g. "region1/ScientaImage". in this case, the dataset is loaded into a sub-folder named "region1".
reduction_func	custom data reduction function. this can be any user-defined function which has the same parameters as adh5_default_reduction. some reduction functions are predefined in the PearlScientaPreprocess module.
reduction_param	parameter string for the reduction function.
progress	progress window. • 1 (default) show progress window • 0 do not show progress window
nthreads	• -1 (default) use as many threads as there are processor cores (in addition to main thread). • 0 use main thread only (for debugging and profiling). • >= 1 use a fixed number of (additional) threads.

Returns

semicolon-separated list of the loaded dataset `ReducedData1`, `ReducedData2`, etc. if successful. auxiliary waves, scan positions, attributes are loaded but not listed in the string. empty string if an error occurred. error messages are printed to the history.

Version

this function supports regions as of version 1.03.

Definition at line 2040 of file pearl-pshell-import.ipf.

12.21.2.18 psh5_load_dataset_slab()

```
string psh5_load_dataset_slab (
             variable fileID,
             string datapath,
             string datasetname,
             variable dim2start,
             variable dim2count,
             variable dim3start,
             variable dim3count )
```

load a single image from the open PShell data file.

the function can average over a region in the extra dimensions.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
datapath	path to the containing group in the HDF5 file. path separator is the slash "/".
dataset	name of the dataset. also defines the name of the loaded wave.
dim2start	2nd dimension coordinate of the first image set to 0 if dimension may not be present
dim2count	number of subsequent images to average set to 1 if dimension may not be present
dim3start	3rd dimension coordinate of the first image set to 0 if dimension may not be present
dim3count	number of subsequent images to average set to 1 if dimension may not be present

Returns

name of loaded wave if successful. empty string otherwise.

Definition at line 1284 of file pearl-pshell-import.ipf.

12.21.2.19 psh5_load_dataset_slabs()

```
string psh5_load_dataset_slabs (
             variable fileID,
             string datapath,
             string datasetname,
             variable progress = defaultValue )
```

load a dataset slab-wise from the open PShell HDF5 file.

the function loads the dataset image by image using the hyperslab option.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
datapath	path to the containing group in the HDF5 file. path separator is the slash "/".
dataset	name of the dataset. also defines the name of the loaded wave.
progress	select whether a progress window is displayed during the process. • 1 (default) show progress window. • 0 do not show progress window.

Returns

name of loaded wave if successful. empty string otherwise.

Definition at line 1148 of file pearl-pshell-import.ipf.

12.21.2.20 psh5_load_info()

```
string psh5_load_info (
             string APathName,
             string AFileName )
```

load descriptive info from a PShell data file.

the info string lists the following information for each scan contained in the file:

- path of the scan group inside the file.
- number of scan positions.
- dataset names of scan positioners.
- dataset names of detectors.

Parameters

APathName	igor symbolic path name. can be empty if the path is specified in AFileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up

Returns

newline terminated string.

Definition at line 2367 of file pearl-pshell-import.ipf.

12.21.2.21 psh5_load_preview()

```
string psh5_load_preview (
             string APathName,
             string AFileName,
             variable load_data = defaultValue,
             variable load_attr = defaultValue,
             string pref_scans = defaultValue,
             string pref_datasets = defaultValue )
```

load a preview image from a PShell data file.

the data wave is loaded into the current data folder. attributes are loaded into the attr subfolder. existing waves in attr are deleted.

Parameters

APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up

Parameters

load_data	1 (default): load data; 0: do not load data
load_attr	1 (default): load attributes; 0: do not load attributes note: for correct scaling of the image, the attributes need to be loaded
pref_scans	semicolon-separated list of preferred scans. the items of the list are match strings for the Igor StringMatch function. the first matching scan (i.e. top-level HDF5 group with a matching name) is loaded from the file. if no match is found, the first scan is loaded.
pref_datasets	semicolon-separated list of preferred datasets. the items of the list are match strings for the Igor StringMatch function. the first matching dataset is loaded from the file. if no match is found, the first dataset listed in the file is loaded.

Returns

name of loaded preview wave.

Definition at line 251 of file pearl-pshell-import.ipf.

12.21.2.22 psh5_load_reduced()

```
string psh5_load_reduced (
              string ANickName,
              string APathName,
              string AFileName,
              funcref reduction_func,
              string reduction_param,
              variable progress = defaultValue,
              variable nthreads = defaultValue )
```

load and reduce the ScientaImage dataset of the first scan of a PShell data file.

the resulting dataset is reduced in one image dimension by a user-defined reduction function, e.g. by region-of-interest integration, curve fitting, etc. cf. adh5_default_reduction for further details.

the function loads the dataset image by image using the hyperslab option and applies a custom reduction function to each image. the results from the reduction function are composed into one result wave. the raw data are discarded.

if the data is from the electron analyser driver and some special attributes are included, the function will set the scales of the image dimensions.

by default, the reduction function is called in separate threads to reduce the total loading time. (see the global variable psh5_perf_secs which reports the total run time of the function.) the effect varies depending on the balance between file loading (image size) and data processing (complexity of the reduction function). for debugging the reduction function, multi-threading can be disabled.

Parameters

ANickName	destination folder name (top level under root).
APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed.
AFileName	if empty a dialog box shows up.
reduction_func	custom data reduction function. this can be any user-defined function which has the same parameters as adh5_default_reduction. some reduction functions are predefined in the PearlScientaPreprocess module.
reduction_param	parameter string for the reduction function.

Parameters

progress	progress window. • 1 (default) show progress window • 0 do not show progress window
nthreads	• -1 (default) use as many threads as there are processor cores (in addition to main thread). • 0 use main thread only (for debugging and profiling). • >= 1 use a fixed number of (additional) threads.

Returns

semicolon-separated list of the loaded dataset `ReducedData1`, `ReducedData2`, etc. if successful. auxiliary waves, scan positions, attributes are loaded but not listed in the string. empty string if an error occurred. error messages are printed to the history.
global string s_filepath in new data folder contains the full file path on disk.
global string s_scanpaths in new data folder contains a list of scan groups inside the file.

Definition at line 1911 of file pearl-pshell-import.ipf.

12.21.2.23 psh5_load_scan_attrs()

```
string psh5_load_scan_attrs (
             variable fileID,
             string scanpath,
             variable attr_sets = defaultValue )
```

load attributes of a PShell scan group.

"attributes" are the auxiliary data inside the attrs group. do not confuse with HDF5 attributes! HDF5 attributes are loaded by the psh5_load_scan_meta() function.

data is loaded into the current data folder. this should normally be the `:attr` folder inside the respective scan folder.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".
attr_sets	specify the attribute sets to be loaded. this value can be an arithmetic OR of the following constants. by default, all attributes are loaded. • 1 all datasets that are present in the file. • 2 datasets relevant for wave scaling of Scienta data.

Returns

semicolon-separated list of the loaded waves.

Definition at line 554 of file pearl-pshell-import.ipf.

12.21.2.24 psh5_load_scan_complete()

```
string psh5_load_scan_complete (
             variable fileID,
             string scanpath,
             variable load_data = defaultValue,
             variable load_attr = defaultValue )
```

load all data of a selected scan from a PShell data file.

data is loaded into the current data folder. attribute datasets are loaded into sub-folder `attr`. region datasets are loaded into region sub-folders. existing data, if present, is overwritten.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".
load_data	select whether datasets (positioners and detectors) are loaded. • 1 (default) load data. • 0 do not load data.
load_attr	select whether attributes (auxiliary device readbacks) are loaded. for proper wave scaling, the attributes must be loaded. • 1 (default) load attributes. • 0 do not load attributes.

Returns

semicolon-separated list of the loaded data waves (excluding attributes).

Definition at line 361 of file pearl-pshell-import.ipf.

12.21.2.25 psh5_load_scan_data()

```
string psh5_load_scan_data (
             variable fileID,
             string scanpath )
```

load all datasets of a PShell scan group.

data is loaded into the current data folder. region datasets are loaded into the respective region sub-folders.

this function does not scale the datasets. call ps_scale_datasets() separately.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".

Returns

semicolon-separated list of the loaded waves.

Definition at line 514 of file pearl-pshell-import.ipf.

12.21.2.26 psh5_load_scan_info()

```
string psh5_load_scan_info (
             variable fileID,
             string scanpath )
```

load descriptive info from a PShell scan.

the info string contains up to three lines which are made up of the following information:

- number of scan positions.
- dataset names of scan positioners.
- dataset names of detectors (without region names).
- region names

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to scan group in the HDF5 file.

Returns

newline terminated string.

Definition at line 2414 of file pearl-pshell-import.ipf.

12.21.2.27 psh5_load_scan_meta()

```
string psh5_load_scan_meta (
             variable fileID,
             string scanpath )
```

load metadata of a PShell scan group.

metadata are the HDF5 attributes attached to the scan group. the following attributes are loaded. the respective wave names under Igor are given in parentheses.

- Dimensions (ScanDimensions)
- Writables (ScanWritables)
- Readables (ScanReadables)
- Steps (ScanSteps)
- Iterations (ScanIterations) - if present (XPSSpectrum script)
- Step Size (ScanStepSize) - if present (XPSSpectrum script)

- Step Time (ScanStepTime) - if present (XPSSpectrum script)

if they are missing in the file, `ScanDimensions` and `ScanReadables` are set to default values assuming the file contains a single spectrum.

data is loaded into the current data folder.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".

Returns

semicolon-separated list of the loaded waves.

Definition at line 625 of file pearl-pshell-import.ipf.

12.21.2.28 psh5_load_scan_preview()

```
string psh5_load_scan_preview (
             variable fileID,
             string scanpath,
             variable set_scale = defaultValue,
             string pref_datasets = defaultValue )
```

load a preview dataset from an open PShell HDF5 file.

if the dataset has a maximum of two dimensions, the function loads it at once. if it has more than two dimension, the function selects and loads one two-dimensional slab.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".
set_scale	by default, the function tries to set the wave scaling if the attributes have been loaded. if multiple datasets are loaded from a file, it is more efficient to set the scaling of all loaded datasets at the end by calling ps_scale_datasets(). • 1 (default) set the wave scaling. • 0 do not set the wave scaling.
pref_datasets	semicolon-separated list of preferred datasets. the items of the list are match strings for the Igor StringMatch function. the first matching dataset is loaded from the file. if no match is found, the first dataset listed in the file is loaded. if empty, a hard-coded default preference list is used.

Returns

name of loaded wave if successful. empty string otherwise.

Definition at line 840 of file pearl-pshell-import.ipf.

12.21.2.29 psh5_load_scan_section()

```
string psh5_load_scan_section (
            variable fileID,
            string scanpath,
            variable dim,
            variable set_scale = defaultValue,
            string pref_datasets = defaultValue )
```

load a longitudinal section of a scan from an open PShell HDF5 file.

the dataset must have three dimensions.

Parameters

fileID	ID of open HDF5 file from psh5_open_file().
scanpath	path to the scan group in the HDF5 file, e.g. "/scan 1".
dim	reserved, must be 0.
set_scale	by default, the function tries to set the wave scaling if the attributes have been loaded. if multiple datasets are loaded from a file, it is more efficient to set the scaling of all loaded datasets at the end by calling ps_scale_datasets(). • 1 (default) set the wave scaling. • 0 do not set the wave scaling.
pref_datasets	semicolon-separated list of preferred datasets. the items of the list are match strings for the Igor StringMatch function. the first matching dataset is loaded from the file. if no match is found, the first dataset listed in the file is loaded. if empty, a hard-coded default preference list is used.

Returns

name of loaded wave if successful. empty string otherwise.

Warning

EXPERIMENTAL: this function is under development.

Definition at line 953 of file pearl-pshell-import.ipf.

12.21.2.30 psh5_open_file()

```
variable psh5_open_file (
            string ANickName,
            string APathName,
            string AFileName )
```

open a HDF5 file created by the PShell data acquisition program and prepare the data folder.

the function opens a specified or interactively selected HDF5 file, creates a data folder `$ANickName` under root, and changes to the new data folder.

the file must be closed by psh5_close_file() after use.

Parameters

ANickName	destination folder name (top level under root).

Parameters

APathName	igor symbolic path name. can be empty if the path is specified in FileName or a dialog box should be displayed
AFileName	if empty a dialog box shows up

Returns

ID of open HDF5 file from HDF5OpenFile. zero if an error occurred.
global string s_filepath in new data folder contains the full file path on disk.
global string s_scanpaths in new data folder contains a list of scan groups inside the file.

Definition at line 110 of file pearl-pshell-import.ipf.

12.21.2.31 reduce_slab_image()

```
static threadsafe wave reduce_slab_image (
             wave slabdata,
             wave image,
             funcref reduction_func,
             string reduction_param )  [static]
```

Definition at line 2341 of file pearl-pshell-import.ipf.

12.21.2.32 reduce_slab_worker()

```
static threadsafe variable reduce_slab_worker (
             funcref reduction_func )  [static]
```

Definition at line 2302 of file pearl-pshell-import.ipf.

12.21.2.33 select_dataset()

```
static string select_dataset (
             string file_datasets,
             string pref_datasets )  [static]
```

select the preferred dataset from a list of available datasets.

Parameters

file_datasets	semicolon-separated list of datasets that are available in the file. the items may include a path separated by slashes "/". only the last component of the path is checked.
pref_datasets	semicolon-separated list of preferred datasets. the items of the list are match strings for the Igor StringMatch function. the first matching dataset is loaded from the file. if no match is found, the first file dataset is selected.

Returns

selected dataset.

Definition at line 778 of file pearl-pshell-import.ipf.

12.21.2.34 twave2list()

```
static string twave2list (
             wave wt,
             string sep )  [static]
```

convert text wave to list.

Definition at line 2473 of file pearl-pshell-import.ipf.

12.21.2.35 wave2list()

```
static string wave2list (
             wave w,
             string format,
             string sep )  [static]
```

convert numeric wave to list.

Definition at line 2490 of file pearl-pshell-import.ipf.

12.21.3 Variable Documentation

12.21.3.1 kAngleDimLabel

```
const string kAngleDimLabel = "angle"
```

Dimension label for the angle dispersive dimension of multi-dimensional datasets.

Definition at line 68 of file pearl-pshell-import.ipf.

12.21.3.2 kDataDimLabel

```
const string kDataDimLabel = "data"
```

Dimension label for the data dimension.

This label may be used to store the parameters for the `setscale d` operation.

Definition at line 75 of file pearl-pshell-import.ipf.

12.21.3.3 kDetectorSensitivity

```
const variable kDetectorSensitivity = 1
```

multiply scienta detector intensity by this value to get actual counts.

Definition at line 87 of file pearl-pshell-import.ipf.

12.21.3.4 kEnergyDimLabel

```
const string kEnergyDimLabel = "energy"
```

Dimension label for the energy dispersive dimension of multi-dimensional datasets.

Definition at line 65 of file pearl-pshell-import.ipf.

12.21.3.5 kPreviewDatasets

```
const string kPreviewDatasets = "ScientaImage;ScientaSpectrum;ImageAngleDistribution;Image↩
EnergyDistribution;Counts;SampleCurrent;"
```

List of preferred datasets to load for preview.

Definition at line 78 of file pearl-pshell-import.ipf.

12.21.3.6 kScanDimLabel

```
const string kScanDimLabel = "scan"
```

Dimension label for the scan dimension of multi-dimensional datasets.

Definition at line 71 of file pearl-pshell-import.ipf.

12.21.3.7 kScientaScalingDatasets

```
const string kScientaScalingDatasets = "LensMode;ScientaChannelBegin;ScientaChannelEnd;Scienta↩
SliceBegin;ScientaSliceEnd;"
```

List of datasets that must be loaded to determine the axis scaling of a Scienta image.

Definition at line 81 of file pearl-pshell-import.ipf.

12.21.3.8 kTransposedDatasets

```
const string kTransposedDatasets = "ScientaImage;"
```

List of datasets that should be transposed upon loading.

Definition at line 84 of file pearl-pshell-import.ipf.

12.22 pearl-scienta-countrate.ipf File Reference

count rate functions for Scienta detector images.

```
#include "pearl-area-display"
```

Namespaces

- PearlScientaCountrate

  count rate functions for Scienta detector images.

Functions

- variable ScientaLiveDisplay (string epicsname, string nickname, string wbRGB)

  open live display of most recent scienta measurement
- variable check_exposure_opt (wave image, wave outmask, variable dwelltime, dfref calc_df=defaultValue)

  optimized check exposure and calculate overexposure indicator mask

12.22.1 Detailed Description

count rate functions for Scienta detector images.

this procedure contains functions for working with true count rates.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.22.2 Function Documentation

12.22.2.1 check_exposure_opt()

```
variable check_exposure_opt (
            wave image,
            wave outmask,
            variable dwelltime,
            dfref calc_df = defaultValue )
```

optimized check exposure and calculate overexposure indicator mask

calculate the local count rate density and return a mask to indicate where the maximum count rate is exceeded. the raw image is filtered by FFT with a gaussian kernel.

the raw image must have been acquired in fixed mode. slicing and dwell time are accounted for.

this function does the same as check_exposure() but keeps intermediate waves for time-optimized processing. moreover it is compatible with igor 6.

Parameters

Definition at line 150 of file pearl-scienta-countrate.ipf.

12.22.2.2 ScientaLiveDisplay()

```
variable ScientaLiveDisplay (
            string epicsname,
            string nickname,
            string wbRGB )
```

open live display of most recent scienta measurement

Parameters

epicsname	base name of the detector, e.g. X03DA-SCIENTA: image1: and cam1: are appended by the function. see ad_connect.
nickname	nick name under which this detector is referred to in Igor. must be a valid data folder name. see ad_connect.
wbRGB	window background color, e.g. "32768,49152,55296"

Definition at line 43 of file pearl-scienta-countrate.ipf.

12.23 pearl-scienta-preprocess.ipf File Reference

preprocessing functions for Scienta detector images.

```
#include "pearl-fitfuncs"
```

Namespaces

- PearlScientaPreprocess

  preprocessing functions for Scienta detector images.

Functions

- variable prompt_int_linbg_reduction (string ∗param)

  prompt the user for integrate on linear background reduction parameters.
- string capture_int_linbg_cursors ()

  capture linear background reduction parameters from cursors in a graph.
- string csr_int_linbg_reduction (string win)

  calculate linear background reduction parameters from cursors in a graph.
- threadsafe wave int_linbg_reduction (wave source, string ∗param)

  linear-background subtracted integration reduction function.
- variable prompt_int_quadbg_reduction (string ∗param)
- threadsafe wave int_quadbg_reduction (wave source, string ∗param)

  integrate peak area minus a quadratic background
- variable prompt_redim_linbg_reduction (string ∗param)

  parameter dialog for the redim_linbg_reduction() function
- threadsafe wave redim_linbg_reduction (wave source, string ∗param)

  linear background reduction function for incorrectly dimensioned scienta image
- variable test_gauss4_reduction (wave image)

apply the gauss4_reduction function to a single image

- variable prompt_gauss4_reduction (string *param)

  prompt for the gauss4_reduction parameters

- threadsafe wave gauss4_reduction (wave source, string *param)

  fit horizontal cuts of an image with up to four gaussian peaks on a linear background

12.23.1 Detailed Description

preprocessing functions for Scienta detector images.

this procedure contains functions for data reduction and instrument-specific normalization.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.23.2 Function Documentation

12.23.2.1 capture_int_linbg_cursors()

```
string capture_int_linbg_cursors ( )
```

capture linear background reduction parameters from cursors in a graph.

PRELIMINARY - function arguments may change

sets reduction parameters from cursors in a graph. the resulting parameters are copied to the global `s_reduction_params` string used by the data explorer.

an even number of cursors (2 or more) must be set on the image. cursor names and order do not matter, except that the alphabetically first cursor which is attached to an image selects the image. the cursors mark the following positions, from innermost to outermost pair:

1. low and high limits of peak region. if no cursors are present, the limits are set at 40% and 60% of the x-scale.
2. peak-side boundary of lower and upper background region. if two or less cursors are present, the default background region applies, which extends from the peak limits up to the default cropping region. the background region extends up to the cropping region defined by the third pair.
3. lower and upper cropping region. if four or less cursors are present, the default cropping region applies, which is 11% on either side of the image in fixed mode, and 0% otherwise. fixed mode is detected by the number of pixels (>= 992).

Note

on profile graphs, the necessary cursors can be configured easily by calling the ad_profile_cursor_mode() function, e.g. `ad_profiles_cursor_mode(root:packages:pearl_explorer:preview_image, 1)`.

Definition at line 93 of file pearl-scienta-preprocess.ipf.

12.23.2.2 csr_int_linbg_reduction()

```
string csr_int_linbg_reduction (
            string win )
```

calculate linear background reduction parameters from cursors in a graph.

PRELIMINARY - function arguments may change

calculates reduction parameters from cursors in a graph. the resulting parameters are returned in a string.

an even number of cursors (2 or more) must be set on the image. cursor names and order do not matter, except that the alphabetically first cursor which is attached to an image selects the image. the cursors mark the following positions, from innermost to outermost pair:

1. low and high limits of peak region. if no cursors are present, the limits are set at 40% and 60% of the x-scale.
2. peak-side boundary of lower and upper background region. if two or less cursors are present, the default background region applies, which extends from the peak limits up to the default cropping region. the background region extends up to the cropping region defined by the third pair.
3. lower and upper cropping region. if four or less cursors are present, the default cropping region applies, which is 11% on either side of the image in fixed mode, and 0% otherwise. fixed mode is detected by the number of pixels (>= 992).

Note

on profile graphs, the necessary cursors can be configured easily by calling the ad_profile_cursor_mode() function, e.g. `ad_profiles_cursor_mode(root:packages:pearl_explorer:preview_↩image, 1)`.

Parameters

win	graph window name or empty string for top window.

Returns

parameter string for linear background subtraction

Definition at line 133 of file pearl-scienta-preprocess.ipf.

12.23.2.3 gauss4_reduction()

```
threadsafe wave gauss4_reduction (
            wave source,
            string * param )
```

fit horizontal cuts of an image with up to four gaussian peaks on a linear background

the function fits each horizontal profile (EDC) with four gaussian peaks on a linear background. the position and width of the peaks is kept fixed according to input parameters. the peak amplitude is constrained to positive value.

the width parameter is defined as in Igor's gauss curve fit function (standard deviation divided by the square root of two). the return value in dest1 is the integrated peak of one of the peaks. dest2 returns the corresponding error estimate.

Parameters

source	source wave. two-dimensional distribution of counts. for correct weighting and error estimation it is important that the source wave contains actual counts (Poisson statistics).
param	(in, out) semicolon-separated key=value list of processing parameters. this is a pass-by-reference argument. the following parameters are required. position, width and limit parameters are on the x (energy) scale. • rngl low limit of fit interval • rngh high limit of fit interval • pos1 position of peak 1 • wid1 width of peak 1 • pos2 position of peak 2 • wid2 width of peak 2 • pos3 position of peak 3 • wid3 width of peak 3 • pos4 position of peak 4 • wid4 width of peak 4 • npeaks number of peaks to fit: 1...4 the others are held at amplitude 0. • ybox box size of averaging in y direction, must be 1 or 3. other values lead to corrupt data.

Returns

free wave containing references of the result waves. the number of waves is two times the number of peaks that are fit. the first npeaks waves contain the peak integrals, the second npeaks waves the corresponding error estimates.

Definition at line 718 of file pearl-scienta-preprocess.ipf.

12.23.2.4 int_linbg_reduction()

```
threadsafe wave int_linbg_reduction (
             wave source,
             string * param )
```

linear-background subtracted integration reduction function.

data reduction function for adh5_load_reduced_detector. cf. adh5_default_reduction for an explanation of reduction functions.

this function calculates the average pixel value of each angular slice in one center and two background intervals. a background value is calculated at the center position by linear interpolation from the two background values. returns the center minus linear background in dest1. returns the Poisson one-sigma error in dest2.

typical values (peak centered on detector, FWHM $\sim$ 20 % of image) Lcrop=0.11;Hcrop=0.11;Lsize=0.2;Hsize=0.↩2;Cpos=0.5;Csize=0.2

Parameters

source	scienta detector image, energy axis along X, angle axis along Y. two-dimensional intensity distribution (image). the scales are carried over to the result waves.
param	parameters in a key1=value1;key2=value2;... list. all region parameters are relative to the image size (0...1). • Lcrop = size of the lower cropping region • Hcrop = size of the upper cropping region • Lsize = size of the lower background integration region • Hsize = size of the upper background integration region • Cpos = center position of the of the peak integration region • Csize = size of the peak integration region

Returns

free wave containing references of the two result waves. the first wave is the integral minus linear background. the second wave is the Poisson one-sigma error.

Definition at line 260 of file pearl-scienta-preprocess.ipf.

12.23.2.5 int_quadbg_reduction()

```
threadsafe wave int_quadbg_reduction (
            wave source,
            string * param )
```

integrate peak area minus a quadratic background

data reduction function for adh5_load_reduced_detector. cf. adh5_default_reduction for an explanation of reduction functions.

this function calculates the average pixel value of each angular slice in one center and two background intervals. a background value is calculated at the center position by linear interpolation from the two background values. returns the center minus linear background in dest1. returns the Poisson one-sigma error in dest2.

typical values (peak centered on detector, FWHM $\sim$ 20 % of image) Lcrop=0.11;Hcrop=0.11;Lsize=0.2;Hsize=0.↩2;Cpos=0.5;Csize=0.2

Parameters

source	scienta detector image, energy axis along X, angle axis along Y. two-dimensional intensity distribution (image). the scales are carried over to the result waves.

Parameters

param	parameters in a key1=value1;key2=value2;... list. all region parameters are relative to the image size (0...1). • Lcrop = size of the lower cropping region • Hcrop = size of the upper cropping region • Lsize = size of the lower background integration region • Hsize = size of the upper background integration region • Cpos = center position of the of the peak integration region • Csize = size of the peak integration region

Returns

free wave containing references of the two result waves. the first wave is the integral minus linear background. the second wave is the Poisson one-sigma error.

Definition at line 413 of file pearl-scienta-preprocess.ipf.

12.23.2.6 prompt_gauss4_reduction()

```
variable prompt_gauss4_reduction (
             string * param )
```

prompt for the gauss4_reduction parameters

Definition at line 626 of file pearl-scienta-preprocess.ipf.

12.23.2.7 prompt_int_linbg_reduction()

```
variable prompt_int_linbg_reduction (
             string * param )
```

prompt the user for integrate on linear background reduction parameters.

Definition at line 35 of file pearl-scienta-preprocess.ipf.

12.23.2.8 prompt_int_quadbg_reduction()

```
variable prompt_int_quadbg_reduction (
             string * param )
```

Definition at line 351 of file pearl-scienta-preprocess.ipf.

12.23.2.9 prompt_redim_linbg_reduction()

```
variable prompt_redim_linbg_reduction (
             string * param )
```

parameter dialog for the redim_linbg_reduction() function

Parameters

param	parameter string in a key1=value1;key2=value2;... list. the parameter string is passed by reference. see redim_linbg_reduction() for a description of parameters.

Returns

zero if the user clicked OK, non-zero if the user clicked Cancel.

Definition at line 509 of file pearl-scienta-preprocess.ipf.

12.23.2.10 redim_linbg_reduction()

```
threadsafe wave redim_linbg_reduction (
             wave source,
             string * param )
```

linear background reduction function for incorrectly dimensioned scienta image

if the energy step size does not divide the energy range to an integer number, the scienta image is exported with the wrong array size. this can be fixed by redimensioning the array.

the current implementation works in the case where dimension 0 needs to be incremented. the function may be generalized to dimension 1 and/or decrementing by additional parameters. it is not known yet whether a generalization is needed or whether it can cover all cases.

background subtraction and peak integration is the same as by the int_linbg_reduction() function.

Parameters

source	source wave Scienta detector image, energy axis along X, angle axis along Y
param	parameter string in a key1=value1;key2=value2;... list. the parameter string is passed by reference.

all region parameters are relative to the image size (0...1).

- Lcrop size of the lower cropping region
- Hcrop size of the upper cropping region
- Lsize size of the lower background integration region
- Hsize size of the upper background integration region
- Cpos center position of the of the peak integration region
- Csize size of the peak integration region

typical values (peak centered on detector, FWHM $\sim$ 20 % of image) Lcrop=0.11;Hcrop=0.11;Lsize=0.2;Hsize=0.↩2;Cpos=0.5;Csize=0.2

Returns

free wave containing references of the two result waves. the first wave is the integral minus linear background. the second wave is the Poisson one-sigma error.

Definition at line 572 of file pearl-scienta-preprocess.ipf.

12.23.2.11 test_gauss4_reduction()

```
variable test_gauss4_reduction (
            wave image )
```

apply the gauss4_reduction function to a single image

useful for testing or manual processing. to debug, (temporarily) remove the threadsafe attribute from the gauss2↩ _reduction function.

Definition at line 592 of file pearl-scienta-preprocess.ipf.

12.24 pearl-tools.ipf File Reference

```
#include "pearl-gui-tools"
```

Functions

- variable DefaultWaveIterator (wave w, string ∗sdata)
- variable AppendToGraphIterator (wave w, string ∗sdata)
- variable SumWavesIterator (wave w, string ∗sdata)
- string IterateWaves (string matchStr, funcref iterator, string sdata)
- variable DefaultFolderIterator (dfref df, string ∗sdata)
- string IterateDataFolders (string matchStr, funcref iterator, string sdata, string progress_title=defaultValue)

12.24.1 Function Documentation

12.24.1.1 AppendToGraphIterator()

```
variable AppendToGraphIterator (
            wave w,
            string * sdata )
```

Definition at line 26 of file pearl-tools.ipf.

12.24.1.2 DefaultFolderIterator()

```
variable DefaultFolderIterator (
            dfref df,
            string * sdata )
```

Definition at line 65 of file pearl-tools.ipf.

12.24.1.3 DefaultWaveIterator()

```
variable DefaultWaveIterator (
            wave w,
            string * sdata )
```

Definition at line 18 of file pearl-tools.ipf.

12.24.1.4 IterateDataFolders()

```
string IterateDataFolders (
             string matchStr,
             funcref iterator,
             string sdata,
             string progress_title = defaultValue )
```

Definition at line 79 of file pearl-tools.ipf.

12.24.1.5 IterateWaves()

```
string IterateWaves (
             string matchStr,
             funcref iterator,
             string sdata )
```

Definition at line 45 of file pearl-tools.ipf.

12.24.1.6 SumWavesIterator()

```
variable SumWavesIterator (
             wave w,
             string * sdata )
```

Definition at line 34 of file pearl-tools.ipf.

12.25 pearl-vector-operations.ipf File Reference

basic vector geometry operations.

Namespaces

- PearlVectorOperations

  basic vector geometry operations.

Functions

- variable rotate2d_x (variable xx, variable yy, variable angle)

  rotate a 2D cartesian vector and returns its x component.
- variable rotate2d_y (variable xx, variable yy, variable angle)

  rotate a 2D cartesian vector and returns its y component.
- wave create_rotation_matrix_free ()

  create a free matrix wave which represents the 3-vector identity.
- wave set_rotation_x (wave matrix, variable angle)

  calculate a matrix representing a 3-vector rotation around the x axis.

- wave set_rotation_y (wave matrix, variable angle)

  calculate a matrix representing a 3-vector rotation around the y axis
- wave set_rotation_z (wave matrix, variable angle)

  calculate a matrix representing a 3-vector rotation around the z axis
- variable rotate_x_wave (wave inout, variable angle)

  rotate a wave of 3-vectors about the x axis.
- variable rotate_y_wave (wave inout, variable angle)

  rotates a wave of 3-vectors about the y axis
- variable rotate_z_wave (wave inout, variable angle)

  rotates a wave of 3-vectors about the z axis

12.25.1 Detailed Description

basic vector geometry operations.

this procedure file contains basic vector geometry functions, such as rotations.

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

12.25.2 Function Documentation

12.25.2.1 create_rotation_matrix_free()

```
wave create_rotation_matrix_free ( )
```

create a free matrix wave which represents the 3-vector identity.

the matrix is initialized as identity.

Returns

3x3 identity matrix in a free wave.

Definition at line 73 of file pearl-vector-operations.ipf.

12.25.2.2 rotate2d_x()

```
variable rotate2d_x (
            variable xx,
            variable yy,
            variable angle )
```

rotate a 2D cartesian vector and returns its x component.

Parameters

xx	x coordinate.
yy	y coordinate.
angle	rotation angle in degrees.

Returns

x coordinate of the rotated vector.

Definition at line 45 of file pearl-vector-operations.ipf.

12.25.2.3 rotate2d_y()

```
variable rotate2d_y (
             variable xx,
             variable yy,
             variable angle )
```

rotate a 2D cartesian vector and returns its y component.

Parameters

xx	x coordinate.
yy	y coordinate.
angle	rotation angle in degrees.

Returns

y coordinate of the rotated vector.

Definition at line 60 of file pearl-vector-operations.ipf.

12.25.2.4 rotate_x_wave()

```
variable rotate_x_wave (
             wave inout,
             variable angle )
```

rotate a wave of 3-vectors about the x axis.

this function rotates multiple vectors.

Parameters

in,out	*inout*	wave with dimensions (M, N), M >= 3 (x, y, z), N >= 1. the result will be in same wave. only the first three rows of dimension 0 are used, extra rows are left unchanged.
in	*angle*	rotation angle in degrees.

Returns

none

Definition at line 175 of file pearl-vector-operations.ipf.

12.25.2.5 rotate_y_wave()

```
variable rotate_y_wave (
            wave inout,
            variable angle )
```

rotates a wave of 3-vectors about the y axis

this function rotates multiple vectors.

Parameters

in,out	*inout*	wave with dimensions (M, N), M >= 3 (x, y, z), N >= 1. the result will be in same wave. only the first three rows of dimension 0 are used, extra rows are left unchanged.
in	*angle*	rotation angle in degrees.

Returns

none

Definition at line 203 of file pearl-vector-operations.ipf.

12.25.2.6 rotate_z_wave()

```
variable rotate_z_wave (
            wave inout,
            variable angle )
```

rotates a wave of 3-vectors about the z axis

this function rotates multiple vectors.

Parameters

in,out	*inout*	wave with dimensions (M, N), M >= 3 (x, y, z), N >= 1. the result will be in same wave. only the first three rows of dimension 0 are used, extra rows are left unchanged.
in	*angle*	rotation angle in degrees.

Returns

none

Definition at line 231 of file pearl-vector-operations.ipf.

12.25.2.7 set_rotation_x()

```
wave set_rotation_x (
```

```
            wave matrix,
            variable angle )
```

calculate a matrix representing a 3-vector rotation around the x axis.

the function calculates the matrix elements of a rotation about the x axis.

Parameters

in,out	*matrix*	3x3 wave to receive the rotation matrix elements. the function calculates only the 2x2 block of the rotation. the other elements must be initialized by the caller, e.g. set to the identity matrix.
in	*angle*	rotation angle in degrees.

Returns

rotation matrix. this is the same wave instance as the matrix input.

Definition at line 92 of file pearl-vector-operations.ipf.

12.25.2.8 set_rotation_y()

```
wave set_rotation_y (
            wave matrix,
            variable angle )
```

calculate a matrix representing a 3-vector rotation around the y axis

the function calculates the matrix elements of a rotation about the y axis.

Parameters

in,out	*matrix*	3x3 wave to receive the rotation matrix elements. the function calculates only the 2x2 block of the rotation. the other elements must be initialized by the caller, e.g. set to the identity matrix.
in	*angle*	rotation angle in degrees.

Returns

rotation matrix. this is the same wave instance as the matrix input.

Definition at line 120 of file pearl-vector-operations.ipf.

12.25.2.9 set_rotation_z()

```
wave set_rotation_z (
            wave matrix,
            variable angle )
```

calculate a matrix representing a 3-vector rotation around the z axis

the function calculates the matrix elements of a rotation about the z axis.

Parameters

`in,out`	*matrix*	3x3 wave to receive the rotation matrix elements. the function calculates only the 2x2 block of the rotation. the other elements must be initialized by the caller, e.g. set to the identity matrix.
`in`	*angle*	rotation angle in degrees.

Returns

rotation matrix. this is the same wave instance as the matrix input.

Definition at line 148 of file pearl-vector-operations.ipf.

Index