PEARL Procedures

Igor procedures for the analysis of PEARL data

Version rev-distro-1.4.0-1-g0a436db-dirty

Thu Sep 21 2017 12:35:04

Contents

1 Introduction **1**

1.1 Introduction . . . 1

1.2 Installation . . . 1

1.3 License Information . . . 1

2 Projections **3**

3 Todo List **5**

4 Module Index **7**

4.1 Packages . . . 7

5 Namespace Index **9**

5.1 Namespace List . . . 9

6 Data Structure Index **11**

6.1 Data Structures . . . 11

7 File Index **13**

7.1 File List . . . 13

8 Module Documentation **15**

8.1 ARPES package . . . 15

8.1.1 Detailed Description . . . 15

9 Namespace Documentation **17**

9.1 PearlAnglescanProcess Namespace Reference . . . 17

9.1.1 Detailed Description . . . 17

9.2 PearlAreaDisplay Namespace Reference . . . 17

9.2.1 Detailed Description . . . 17

9.2.2 Miscellaneous . . . 17

9.3 PearlAreaImport Namespace Reference . . . 18

9.3.1 Detailed Description . . . 18

9.4 PearlAreaProfiles Namespace Reference . . . 18

9.4.1 Detailed Description . . . 18
9.5 PearlArpes Namespace Reference . . . 18
9.5.1 Detailed Description . . . 18
9.6 PearlDataExplorer Namespace Reference . . . 18
9.6.1 Detailed Description . . . 19
9.7 PearlElog Namespace Reference . . . 19
9.7.1 Detailed Description . . . 19
9.8 PearlMatrixImport Namespace Reference . . . 19
9.8.1 Detailed Description . . . 19
9.9 PearlPShellImport Namespace Reference . . . 19
9.9.1 Detailed Description . . . 19
9.10 PearlScientaPreprocess Namespace Reference . . . 19
9.10.1 Detailed Description . . . 19

10 Data Structure Documentation **21**
10.1 DoniachSunjicStruct Struct Reference . . . 21
10.1.1 Detailed Description . . . 21
10.1.2 Field Documentation . . . 21
10.1.2.1 broadening . . . 21
10.1.2.2 convolution . . . 21
10.1.2.3 model . . . 21
10.1.2.4 oversampling . . . 21
10.1.2.5 precision . . . 22
10.1.2.6 pw . . . 22
10.1.2.7 xdw . . . 22
10.1.2.8 xw . . . 22
10.1.2.9 yw . . . 22
10.2 errorCode Struct Reference . . . 22
10.2.1 Detailed Description . . . 22
10.2.2 Field Documentation . . . 23
10.2.2.1 ALREADY_FILE_OPEN . . . 23
10.2.2.2 EMPTY_RESULTFILE . . . 23
10.2.2.3 FILE_NOT_READABLE . . . 23
10.2.2.4 INTERNAL_ERROR_CONVERTING_DATA . . . 23
10.2.2.5 INVALID_RANGE . . . 23
10.2.2.6 NO_FILE_OPEN . . . 23
10.2.2.7 NO_NEW_BRICKLETS . . . 23
10.2.2.8 SUCCESS . . . 23
10.2.2.9 UNKNOWN_ERROR . . . 23
10.2.2.10 WAVE_EXIST . . . 23

10.2.2.11 WRONG_PARAMETER . . . 23

11 File Documentation **25**

11.1 fermi-edge-analysis.ipf File Reference . . . 25

11.1.1 Detailed Description . . . 25

11.1.2 Function Documentation . . . 26

11.1.2.1 analyse_curved_edge(wave data) . . . 26

11.1.2.2 analyser_energy_resolution(variable epass, variable slit) . . . 26

11.1.2.3 FermiFuncLinDOS2D_corr(variable w, threadsafe x, wave y) . . . 26

11.1.2.4 integrate_curved_edge(wave data, wave data_sig) . . . 26

11.1.2.5 record_results(variable index) . . . 26

11.1.2.6 show_shift(wave data) . . . 27

11.1.2.7 slit_correction(wave data, wave data_out, variable epass) . . . 27

11.1.2.8 slit_shift(variable ypix, variable epass) . . . 27

11.1.3 Variable Documentation . . . 27

11.1.3.1 hemi_radius_mm . . . 27

11.1.3.2 mcp_radius_epass . . . 27

11.1.3.3 mcp_radius_mm . . . 27

11.1.3.4 mcp_radius_pix . . . 27

11.2 mainpage.dox File Reference . . . 27

11.3 pearl-anglescan-process.ipf File Reference . . . 27

11.3.1 Detailed Description . . . 30

11.3.2 Function Documentation . . . 31

11.3.2.1 calc_graph_azi(variable x, variable y, variable projection=defaultValue, variable zeroAngle=defaultValue) . . . 31

11.3.2.2 calc_graph_polar(variable x, variable y, variable projection=defaultValue) . . . 32

11.3.2.3 calc_graph_radius(variable polar, variable projection=defaultValue) . . . 32

11.3.2.4 calc_nth(variable Theta_st, variable Theta_in, variable th, variable Phi_ran, variable Phi_ref, string Holomode) . . . 32

11.3.2.5 calc_phi_step(variable Theta_in, variable th, variable Theta_st, variable Phi_ran, variable Phi_ref, string Holomode) . . . 33

11.3.2.6 Calc_The_step(variable th, variable Theta_st, string Holomode) . . . 33

11.3.2.7 CalcN_Theta(string HoloMode, variable Theta_in, variable Theta_ran, variable Theta_st) . . . 33

11.3.2.8 check_contrast(wave values, variable pcmin, variable pcmax, variable *vmin, variable *vmax) . . . 33

11.3.2.9 clear_hemi_grid(string nickname) . . . 33

11.3.2.10 convert_angles_ttpa2polar(wave theta, wave tilt, wave phi, wave analyser, wave polar, wave azi) . . . 33

11.3.2.11 convert_angles_ttpd2polar(wave theta, wave tilt, wave phi, wave data, wave polar, wave azi) . . . 34

11.3.2.12 crop_strip(wave strip, variable xlo, variable xhi) . . . 34

11.3.2.13 display_hemi_scan(string nickname, variable projection=defaultValue, variable graphtype=defaultValue, variable do_ticks=defaultValue, variable do_grids=defaultValue, string graphname=defaultValue) . . . 34

11.3.2.14 display_polar_graph(string graphname, variable angle_offset=defaultValue, variable do_ticks=defaultValue) . . . 35

11.3.2.15 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) . . . 36

11.3.2.16 draw_diffraction_cone(string graphname, string groupname, variable theta_axis, variable theta_inner, variable phi) . . . 37

11.3.2.17 draw_hemi_axes(string graphname, variable do_grids=defaultValue) . . . 37

11.3.2.18 duplicate_hemi_scan(string source_nickname, dfref dest_folder, string dest_nickname, variable xpdplot=defaultValue) . . . 38

11.3.2.19 find_hemi_data(string nickname, string *prefix, string *intwave) . . . 38

11.3.2.20 get_hemi_nickname(wave w) . . . 38

11.3.2.21 get_hemi_prefix(wave w) . . . 39

11.3.2.22 hemi_add_anglescan(string nickname, wave values, wave polar, wave azi, wave weights=defaultValue) . . . 39

11.3.2.23 hemi_add_aziscan(string nickname, wave values, variable polar, wave azi, wave weights=defaultValue) . . . 39

11.3.2.24 hemi_azi_cut(string nickname, variable pol) . . . 39

11.3.2.25 hemi_polar_cut(string nickname, variable azim) . . . 40

11.3.2.26 import_tpi_scan(string nickname, wave theta, wave phi, wave intensity, variable folding=defaultValue, variable npolar=defaultValue, variable nograph=defaultValue, variable xpdplot=defaultValue) . . . 40

11.3.2.27 interpolate_hemi_scan(string nickname) . . . 41

11.3.2.28 line_average(wave source, wave dest) . . . 41

11.3.2.29 load_hemi_scan(string nickname, string pathname, string filename) . . . 41

11.3.2.30 make_hemi_grid(variable npol, string nickname, variable xpdplot=defaultValue) . . . 41

11.3.2.31 normalize_strip_2d(wave strip, wave theta, variable theta_offset=defaultValue, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue) . . . 42

11.3.2.32 normalize_strip_phi(wave strip, wave theta, wave phi, variable theta_offset=defaultValue, variable theta_range=defaultValue, variable check=defaultValue) . . . 42

11.3.2.33 normalize_strip_theta(wave strip, wave theta, variable theta_offset=defaultValue, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue) . . . 43

11.3.2.34 normalize_strip_x(wave strip, variable smooth_method=defaultValue, variable smooth_factor=defaultValue, variable check=defaultValue) . . . 44

11.3.2.35 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) 44

11.3.2.36 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) . . . 45

11.3.2.37 polar_graph_hook(WMWinHookStruct *s) . . . 46

11.3.2.38 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) . . . 46
11.3.2.39 rotate_hemi_scan(string nickname, variable angle) . . . 47
11.3.2.40 save_hemi_scan(string nickname, string pathname, string filename) . . . 47
11.3.2.41 set_contrast(variable pcmin, variable pcmax, string graphname=defaultValue, string colortable=defaultValue) . . . 47
11.3.2.42 set_polar_graph_cursor(string nickname, string cursorname, variable polar_↩ angle, variable azim_angle, string graphname=defaultValue) . . . 47
11.3.2.43 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) . . . 47
11.3.2.44 strip_delete_frames(wave strip, variable qlo, variable qhi, wave theta, wave tilt, wave phi) . . . 48
11.3.2.45 trim_hemi_scan(string nickname, variable theta_max) . . . 48
11.3.2.46 update_polar_info(string graphname) . . . 49
11.3.3 Variable Documentation . . . 49
11.3.3.1 kProjArea . . . 49
11.3.3.2 kProjDist . . . 49
11.3.3.3 kProjGnom . . . 49
11.3.3.4 kProjOrtho . . . 49
11.3.3.5 kProjScaleArea . . . 49
11.3.3.6 kProjScaleDist . . . 49
11.3.3.7 kProjScaleGnom . . . 50
11.3.3.8 kProjScaleOrtho . . . 50
11.3.3.9 kProjScaleStereo . . . 50
11.3.3.10 kProjStereo . . . 50
11.4 pearl-anglescan-tracker.ipf File Reference . . . 50
11.4.1 Function Documentation . . . 52
11.4.1.1 add_image_data(wave image, variable theta, variable tilt, variable phi) . . . 52
11.4.1.2 AfterCompiledHook() . . . 52
11.4.1.3 ast_add_image(wave image, variable theta, variable tilt, variable phi) . . . 52
11.4.1.4 ast_callback_data(variable chan) . . . 52
11.4.1.5 ast_callback_detector(variable chan) . . . 53
11.4.1.6 ast_callback_manip(variable chan) . . . 53
11.4.1.7 ast_close() . . . 53
11.4.1.8 ast_export(dfref folder, string nickname, variable xpdplot=defaultValue) . . . 53
11.4.1.9 ast_import(string nickname) . . . 53
11.4.1.10 ast_prepare(variable theta_offset=defaultValue, variable tilt_offset=defaultValue, variable phi_offset=defaultValue) . . . 53
11.4.1.11 ast_set_processing(string reduction_func, string reduction_params) . . . 54
11.4.1.12 ast_setup() . . . 54

11.4.1.13 ast_update_detector(variable theta, variable tilt, variable phi, variable range) . . 54
11.4.1.14 ast_window_hook(WMWinHookStruct *s) 54
11.4.1.15 bp_capture(WMButtonAction *ba) 55
11.4.1.16 edit_offsets() 55
11.4.1.17 edit_reduction_params() 55
11.4.1.18 epics_connect() 55
11.4.1.19 epics_disconnect() 55
11.4.1.20 epics_disconnect_chid(string chid_var_name) 55
11.4.1.21 export_tracker_data() 55
11.4.1.22 extend_data(variable num_slices) 55
11.4.1.23 IgorQuitHook(string app) 56
11.4.1.24 import_tracker_data() 56
11.4.1.25 init_package() 56
11.4.1.26 load_prefs() 56
11.4.1.27 load_tracker_data() 56
11.4.1.28 pmp_data(WMPopupAction *pa) 56
11.4.1.29 pmp_data_mouseup(WMPopupAction *pa) 56
11.4.1.30 pmp_parameters(WMPopupAction *pa) 56
11.4.1.31 pmp_parameters_mouseup(WMPopupAction *pa) 56
11.4.1.32 process_image_data() 56
11.4.1.33 save_prefs() 57
11.4.1.34 save_tracker_data() 57
11.4.1.35 setup_data() 57
11.4.1.36 setup_detector() 57
11.4.1.37 setup_graph() 57
11.4.1.38 toggle_capture() 57
11.4.1.39 update_capture() 57
11.4.1.40 update_data_graph() 57
11.4.1.41 update_detector(variable theta, variable tilt, variable phi, variable range) 57
11.4.1.42 update_detector_graph() 58
11.4.2 Variable Documentation 58
11.4.2.1 package_path 58
11.4.2.2 prefs_objects 58
11.4.2.3 version 58
11.5 pearl-area-display.ipf File Reference 58
11.5.1 Detailed Description 60
11.5.2 2D data 60
11.5.3 3D data 60
11.5.4 Function Documentation 60
11.5.4.1 ad_box_filter(wave image, string options) 60

11.5.4.2 ad_brick_slicer(wave data) . . . 61
11.5.4.3 ad_calc_cursor_profiles(wave image) . . . 61
11.5.4.4 ad_calc_histogram(wave image) . . . 61
11.5.4.5 ad_calc_profiles(wave image, variable pa, variable qa, variable pb, variable qb) . . . 61
11.5.4.6 ad_default_image_filter(wave image, string options) . . . 62
11.5.4.7 ad_display(wave image) . . . 62
11.5.4.8 ad_display_brick(wave data) . . . 62
11.5.4.9 ad_display_histogram(wave image) . . . 63
11.5.4.10 ad_display_profiles(wave image, string filter=defaultValue) . . . 63
11.5.4.11 ad_display_slice(wave data) . . . 63
11.5.4.12 ad_export_profile(wave view_image, variable dim, variable trace=defaultValue, variable show=defaultValue, variable overwrite=defaultValue) . . . 64
11.5.4.13 ad_gizmo_set_plane(wave brick, variable dim, variable value) . . . 64
11.5.4.14 ad_profiles_crosshairs(wave image, variable clear=defaultValue) . . . 65
11.5.4.15 ad_profiles_cursor_mode(wave image, variable mode) . . . 65
11.5.4.16 ad_profiles_hook(WMWinHookStruct *s) . . . 65
11.5.4.17 ad_profiles_set_cursor(wave image, string cursorname, variable xa, variable ya, variable pscale=defaultValue) . . . 66
11.5.4.18 ad_profiles_set_slice(wave brick, variable dim, variable value) . . . 66
11.5.4.19 ad_slicer_init_bg() . . . 66
11.5.4.20 ad_slicer_move_bg(WMBackgroundStruct *s) . . . 66
11.5.4.21 ad_slicer_start_bg(wave brick, variable dimension, string posvariable, variable delta) . . . 66
11.5.4.22 ad_slicer_stop_bg(string posvariable) . . . 67
11.5.4.23 ad_transpose_filter(wave image, string options) . . . 67
11.5.4.24 ad_update_profiles(wave image) . . . 67
11.5.4.25 bp_extract_slice(WMButtonAction *ba) . . . 67
11.5.4.26 bp_move_slice(WMButtonAction *ba) . . . 68
11.5.4.27 bp_move_slice_center(wave brick, variable dim, string posvariable) . . . 68
11.5.4.28 bp_reset_cursors(WMButtonAction *ba) . . . 68
11.5.4.29 get_source_image(wave view) . . . 68
11.5.4.30 get_view_folder(wave source) . . . 68
11.5.4.31 get_view_image(wave source) . . . 68
11.5.4.32 graphname_from_dfref(dfref df, string prefix) . . . 68
11.5.4.33 make_view_folder(wave source) . . . 69
11.5.4.34 pmp_export(WMPopupAction *pa) . . . 69
11.5.4.35 set_trace_colors(string graphname) . . . 69
11.5.4.36 slp_slice_position(WMSliderAction *sa) . . . 69
11.5.4.37 svp_slice_position(WMSetVariableAction *sva) . . . 69
11.5.4.38 svp_smoothing(WMSetVariableAction *sva) . . . 69
11.5.4.39 update_slice_info() . . . 69

11.6 pearl-area-import.ipf File Reference . . . 69
11.6.1 Detailed Description . . . 71
11.6.2 Function Documentation . . . 71
11.6.2.1 ad_load_dialog(string APathName) . . . 71
11.6.2.2 ad_suggest_foldername(string filename, variable ignoredate=defaultValue, string sourcename=defaultValue, variable unique=defaultValue) . . . 71
11.6.2.3 adh5_default_reduction(wave source, wave dest1, wave dest2, string *param) . . . 72
11.6.2.4 adh5_list_reduction_funcs() . . . 72
11.6.2.5 adh5_load_complete(string ANickName, string APathName, string AFileName, variable load_data=defaultValue, variable load_attr=defaultValue) . . . 72
11.6.2.6 adh5_load_detector(variable fileID, string detectorpath) . . . 73
11.6.2.7 adh5_load_detector_image(variable fileID, string detectorpath, variable dim2start, variable dim2count, variable dim3start, variable dim3count) . . . 73
11.6.2.8 adh5_load_detector_slabs(variable fileID, string detectorpath, variable progress=defaultValue) . . . 73
11.6.2.9 adh5_load_info(string APathName, string AFileName) . . . 74
11.6.2.10 adh5_load_preview(string ANickName, string APathName, string AFileName, variable load_data=defaultValue, variable load_attr=defaultValue) . . . 74
11.6.2.11 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) . . . 74
11.6.2.12 adh5_load_reduced_detector(variable fileID, string detectorpath, funcref reduction_func, string reduction_param, variable progress=defaultValue, variable nthreads=defaultValue) . . . 75
11.6.2.13 adh5_loadattr_all(variable fileID, string attributespath) . . . 75
11.6.2.14 adh5_redim(wave data) . . . 76
11.6.2.15 adh5_scale(wave data, string source=defaultValue) . . . 76
11.6.2.16 adh5_scale_scan(wave data) . . . 76
11.6.2.17 adh5_scale_scienta(wave data) . . . 76
11.6.2.18 adh5_setup_profile(wave image, wave profile, variable dim) . . . 76
11.6.2.19 adh5_test_reduction_func(wave source, wave dest1, wave dest2, funcref reduction_func, string reduction_param) . . . 77
11.6.2.20 BeforeFileOpenHook(variable refNum, string fileName, string path, string type, string creator, variable kind) . . . 77
11.6.2.21 GetAttrDataFolderDFR(wave data) . . . 77
11.6.2.22 read_attribute_info(string datawavename, string source, variable *idest) . . . 77
11.6.2.23 reduce_slab_image(wave slabdata, wave image, wave profile1, wave profile2, funcref reduction_func, string reduction_param) . . . 78
11.6.2.24 reduce_slab_worker(funcref reduction_func) . . . 78
11.7 pearl-area-profiles.ipf File Reference . . . 78
11.7.1 Detailed Description . . . 79
11.7.2 Function Documentation . . . 79
11.7.2.1 ad_collect_multiscan_y(wave dataset, wave positions, wave destwave, variable noavg=defaultValue) . . . 79

11.7.2.2 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) . . . 79
11.7.2.3 ad_extract_rod_x(wave dataset, variable q1, variable q2, variable r1, variable r2, string destname, variable noavg=defaultValue, variable sdev=defaultValue) . . . 80
11.7.2.4 ad_extract_rod_y(wave dataset, variable p1, variable p2, variable r1, variable r2, string destname, variable noavg=defaultValue, variable sdev=defaultValue) . . . 80
11.7.2.5 ad_extract_rod_z(wave dataset, variable p1, variable p2, variable q1, variable q2, string destname, variable noavg=defaultValue, variable sdev=defaultValue) . . . 80
11.7.2.6 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) . . . 80
11.7.2.7 ad_extract_slab_x(wave dataset, variable p1, variable p2, string destname, variable noavg=defaultValue) . . . 81
11.7.2.8 ad_extract_slab_y(wave dataset, variable q1, variable q2, string destname, variable noavg=defaultValue) . . . 81
11.7.2.9 ad_extract_slab_z(wave dataset, variable r1, variable r2, string destname, variable noavg=defaultValue) . . . 81
11.7.2.10 ad_profile_x(wave dataset, variable q1, variable q2, string destname, variable noavg=defaultValue) . . . 81
11.7.2.11 ad_profile_x_w(wave dataset, variable q1, variable q2, wave destwave, variable noavg=defaultValue) . . . 81
11.7.2.12 ad_profile_y(wave dataset, variable p1, variable p2, string destname, variable noavg=defaultValue) . . . 81
11.7.2.13 ad_profile_y_w(wave dataset, variable p1, variable p2, wave destwave, variable noavg=defaultValue) . . . 81
11.7.2.14 calc_y_profile_mins(wave image) . . . 81
11.8 pearl-arpes.ipf File Reference . . . 81
11.8.1 Detailed Description . . . 82
11.8.2 Function Documentation . . . 82
11.8.2.1 AfterCompiledHook() . . . 82
11.8.2.2 UnloadPearlArpesPackage() . . . 83
11.9 pearl-data-explorer.ipf File Reference . . . 83
11.9.1 Detailed Description . . . 84
11.9.2 Function Documentation . . . 85
11.9.2.1 attributes_notebook(wave attr_names, wave attr_values, string title) . . . 85
11.9.2.2 bp_attr_notebook(WMButtonAction *ba) . . . 85
11.9.2.3 bp_browse_filepath(WMButtonAction *ba) . . . 85
11.9.2.4 bp_dataset_display(WMButtonAction *ba) . . . 85
11.9.2.5 bp_dataset_folder(WMButtonAction *ba) . . . 85
11.9.2.6 bp_dataset_next(WMButtonAction *ba) . . . 85
11.9.2.7 bp_dataset_prev(WMButtonAction *ba) . . . 85
11.9.2.8 bp_file_next(WMButtonAction *ba) . . . 85
11.9.2.9 bp_file_prev(WMButtonAction *ba) . . . 85

11.9.2.10 bp_load_files(WMButtonAction *ba) . . . 85
11.9.2.11 bp_load_files_opt(WMButtonAction *ba) . . . 85
11.9.2.12 bp_load_prefs(WMButtonAction *ba) . . . 85
11.9.2.13 bp_save_prefs(WMButtonAction *ba) . . . 86
11.9.2.14 bp_update_datasets(WMButtonAction *ba) . . . 86
11.9.2.15 bp_update_filelist(WMButtonAction *ba) . . . 86
11.9.2.16 display_dataset(string datasetname) . . . 86
11.9.2.17 display_preview_trace(wave xtrace, wave ytrace) . . . 86
11.9.2.18 extract_preview_image(wave data, wave preview) . . . 86
11.9.2.19 init_package() . . . 86
11.9.2.20 itx_suggest_foldername(string filename, variable ignoredate=defaultValue, string sourcename=defaultValue, variable unique=defaultValue) . . . 86
11.9.2.21 lbp_datasets(WMListboxAction *lba) . . . 86
11.9.2.22 lbp_filelist(WMListboxAction *lba) . . . 86
11.9.2.23 load_file(string filename, string options=defaultValue) . . . 86
11.9.2.24 load_hdf_file(string filename, string options=defaultValue) . . . 87
11.9.2.25 load_itx_file(string filename, string options=defaultValue) . . . 87
11.9.2.26 load_mtrx_file(string filename, string options=defaultValue) . . . 87
11.9.2.27 load_prefs() . . . 87
11.9.2.28 load_pshell_file(string filename, string options=defaultValue) . . . 87
11.9.2.29 load_selected_files(string options=defaultValue) . . . 87
11.9.2.30 notebook_add_attributes(string notebook_name, wave attr_filter, wave attr_↩ names, wave attr_values) . . . 87
11.9.2.31 pearl_data_explorer() . . . 87
11.9.2.32 pearl_file_type(string filename) . . . 87
11.9.2.33 PearlDataExplorer() . . . 88
11.9.2.34 preview_attributes(dfref attr_folder, dfref dest_folder=defaultValue, wave attr_↩ filter=defaultValue, variable include_datawaves=defaultValue, variable include_↩ infowaves=defaultValue) . . . 88
11.9.2.35 preview_datafolder() . . . 88
11.9.2.36 preview_dataset(string datasetname) . . . 88
11.9.2.37 preview_file(string filename) . . . 88
11.9.2.38 preview_hdf_file(string filename) . . . 88
11.9.2.39 preview_itx_file(string filename) . . . 88
11.9.2.40 preview_mtrx_file(string filename) . . . 88
11.9.2.41 preview_pshell_file(string filename) . . . 89
11.9.2.42 preview_setscale_x(wave data, wave preview) . . . 89
11.9.2.43 prompt_default_process(string *param) . . . 89
11.9.2.44 prompt_func_params(string func_name, string *func_param) . . . 89
11.9.2.45 prompt_hdf_options(string *options) . . . 89
11.9.2.46 save_prefs() . . . 90

11.9.2.47 show_preview_graph(wave data, wave xdata=defaultValue) . . . 90
11.9.2.48 test_attributes_notebook() . . . 90
11.9.2.49 update_datasets() . . . 90
11.9.2.50 update_filelist() . . . 90
11.9.3 Variable Documentation . . . 90
11.9.3.1 ks_filematch_adh5 . . . 90
11.9.3.2 ks_filematch_itx . . . 90
11.9.3.3 ks_filematch_mtrx . . . 90
11.9.3.4 ks_filematch_pshell . . . 90
11.9.3.5 package_name . . . 90
11.9.3.6 package_path . . . 90
11.10 pearl-elog.ipf File Reference . . . 91
11.10.1 Detailed Description . . . 93
11.10.2 Function Documentation . . . 94
11.10.2.1 AfterFileOpenHook(variable refNum, string file, string pathName, string type, string creator, variable kind) . . . 94
11.10.2.2 bp_attach(WMButtonAction *ba) . . . 94
11.10.2.3 bp_attach_allnone(WMButtonAction *ba) . . . 94
11.10.2.4 bp_attach_top(WMButtonAction *ba) . . . 94
11.10.2.5 bp_attach_updown(WMButtonAction *ba) . . . 94
11.10.2.6 bp_clear(WMButtonAction *ba) . . . 95
11.10.2.7 bp_login(WMButtonAction *ba) . . . 95
11.10.2.8 bp_logout(WMButtonAction *ba) . . . 95
11.10.2.9 bp_save_graphs(WMButtonAction *ba) . . . 95
11.10.2.10 bp_submit(WMButtonAction *ba) . . . 95
11.10.2.11 cleanup_temp_files() . . . 95
11.10.2.12 create_cmd_file(string cmd) . . . 95
11.10.2.13 create_graph_file(string graphname, variable fileindex) . . . 95
11.10.2.14 create_message_file(string message) . . . 96
11.10.2.15 elog_add_attachment(string logbook, variable id, string graphs) . . . 96
11.10.2.16 elog_config(string elog_path=defaultValue, string hostname=defaultValue, variable port=defaultValue, string subdir=defaultValue) . . . 97
11.10.2.17 elog_create_entry(string logbook, string attributes, string message, variable encoding=defaultValue, string graphs=defaultValue, variable replyto=defaultValue) 97
11.10.2.18 elog_create_logbook(string name, string template=defaultValue) . . . 97
11.10.2.19 elog_init_pearl_templates() . . . 98
11.10.2.20 elog_login(string logbook, string username, string password) . . . 98
11.10.2.21 elog_logout(string logbook) . . . 98
11.10.2.22 elog_panel_hook(WMWinHookStruct *s) . . . 98
11.10.2.23 elog_prompt_logbook() . . . 98
11.10.2.24 elog_prompt_login(string logbook) . . . 99

11.10.2.25 elog_validate_attributes(string logbook, string attributes) . . . 99
11.10.2.26 format_url(string logbook) . . . 99
11.10.2.27 get_default_panel_name() . . . 99
11.10.2.28 get_elog_df(string name, variable category) . . . 99
11.10.2.29 get_log_path() . . . 100
11.10.2.30 get_panel_attributes(string windowname) . . . 100
11.10.2.31 get_panel_graphs(string windowname) . . . 100
11.10.2.32 get_panel_message(string windowname) . . . 100
11.10.2.33 get_timestamp(string sep) . . . 100
11.10.2.34 IgorBeforeNewHook(string igorApplicationNameStr) . . . 101
11.10.2.35 IgorQuitHook(string igorApplicationNameStr) . . . 101
11.10.2.36 init_package(variable clean=defaultValue) . . . 101
11.10.2.37 init_volatile_vars() . . . 101
11.10.2.38 list_logbooks(variable templates=defaultValue) . . . 101
11.10.2.39 load_prefs() . . . 102
11.10.2.40 move_attach_item(string logbook, variable item, variable distance) . . . 102
11.10.2.41 parse_result() . . . 102
11.10.2.42 pearl_elog(string logbook) . . . 102
11.10.2.43 PearlElogPanel(string logbook) . . . 102
11.10.2.44 prepare_command_line(string logbook) . . . 102
11.10.2.45 prepare_graph_attachments(string graphs) . . . 103
11.10.2.46 save_prefs() . . . 103
11.10.2.47 set_panel_attributes(string windowname, string attributes, variable clear=defaultValue) . . . 103
11.10.2.48 set_panel_graphs(string windowname, string graphs) . . . 103
11.10.2.49 set_panel_message(string windowname, string message) . . . 104
11.10.2.50 update_attach_items(string logbook) . . . 104
11.10.3 Variable Documentation . . . 104
11.10.3.1 elog_parse_id . . . 104
11.10.3.2 elog_success_msg . . . 104
11.10.3.3 kAttachColName . . . 104
11.10.3.4 kAttachColSel . . . 104
11.10.3.5 kAttachColTitle . . . 104
11.10.3.6 kdfPersistent . . . 104
11.10.3.7 kdfRoot . . . 105
11.10.3.8 kdfTemplates . . . 105
11.10.3.9 kdfVolatile . . . 105
11.10.3.10 package_name . . . 105
11.10.3.11 package_path . . . 105
11.11 pearl-fitfuncs.ipf File Reference . . . 105

11.11.1 Function Documentation . . . 106
11.11.1.1 Au4f(wave w, variable x) . . . 106
11.11.1.2 Au4f_2p2(wave w, variable x) . . . 106
11.11.1.3 Au4f_2p3(wave w, variable x) . . . 106
11.11.1.4 Calc_DoniachSunjicBroad(wave pw, wave yw) . . . 106
11.11.1.5 DoniachSunjicBroad(wave pw, wave yw, wave xw) . . . 106
11.11.1.6 DoniachSunjicBroadS(DoniachSunjicStruct *s) . . . 106
11.11.1.7 ds1_bg(wave w, variable x) . . . 106
11.11.1.8 ds2_bg(wave w, variable x) . . . 106
11.11.1.9 ds4_bg(wave w, variable x) . . . 106
11.11.1.10 ds6_bg(wave w, variable x) . . . 106
11.11.1.11 FermiGaussConv(wave pw, wave yw, wave xw) . . . 106
11.11.1.12 Fit_DoniachSunjicBroad(wave pw, wave yw, wave xw, wave ww) . . . 107
11.11.1.13 ShowComponents_Au4f_2p2(wave coef_wave, wave fit_wave) . . . 107
11.11.1.14 ShowComponents_Au4f_2p3(wave coef_wave, wave fit_wave) . . . 107
11.11.2 Variable Documentation . . . 107
11.11.2.1 version . . . 107
11.12 pearl-gui-tools.ipf File Reference . . . 107
11.12.1 Function Documentation . . . 107
11.12.1.1 display_progress_panel(string title, string message, variable progress_max) . . . 107
11.12.1.2 kill_progress_panel() . . . 107
11.12.1.3 update_progress_panel(variable progress, string message=defaultValue, variable progress_max=defaultValue) . . . 107
11.13 pearl-matrix-import.ipf File Reference . . . 108
11.13.1 Detailed Description . . . 109
11.13.2 Function Documentation . . . 109
11.13.2.1 AfterFileOpenHook(variable refNum, string file, string pathName, string type, string creator, variable kind) . . . 109
11.13.2.2 BeforeFileOpenHook(variable refNum, string fileName, string path, string type, string creator, variable kind) . . . 109
11.13.2.3 check_package_folder() . . . 110
11.13.2.4 init_package() . . . 110
11.13.2.5 initStruct(errorCode *errorCode) . . . 110
11.13.2.6 matrix_format_elog_message(wave metadata) . . . 110
11.13.2.7 matrix_preview_2d(wave data, wave metadata) . . . 110
11.13.2.8 mtrx_create_folder(string fileName, dfref df_base=defaultValue) . . . 110
11.13.2.9 mtrx_file_brickletID(string resultFile, variable runCycle, variable scanCycle, string channel) . . . 111
11.13.2.10 mtrx_get_cycle_folder(dfref df_base=defaultValue, variable runCycle=defaultValue, variable scanCycle=defaultValue) . . . 111
11.13.2.11 mtrx_load_all() . . . 111

11.13.2.12 mtrx_load_file(string pathName, string fileName, string traces=defaultValue) . . . 111
11.13.2.13 mtrx_load_info(string APathName, string AFileName) . . . 112
11.13.2.14 mtrx_load_preview(string destName, string pathName, string fileName, string traces=defaultValue) . . . 112
11.13.2.15 mtrx_open_file(string pathName, string fileNameOrPath) . . . 113
11.13.2.16 mtrx_parse_filename(string fileName, string *resultFile, variable *runCycle, variable *scanCycle, string *channel) . . . 114
11.13.2.17 mtrx_scale_dataset(wave data) . . . 114
11.13.2.18 mtrx_split_filename(string fileName, string *prefix, string *datepart, string *timepart) . . . 114
11.13.2.19 preview_matrix_file(string filename) . . . 114
11.13.2.20 subtract_line_bg(wave img) . . . 115
11.13.3 Variable Documentation . . . 115
11.13.3.1 ks_filematch_mtrx . . . 115
11.13.3.2 package_name . . . 115
11.13.3.3 package_path . . . 115
11.14 pearl-menu.ipf File Reference . . . 115
11.14.1 Function Documentation . . . 115
11.14.1.1 Display2dProfiles() . . . 115
11.14.1.2 Display3dSlicer() . . . 115
11.14.1.3 DisplayGizmoSlicer() . . . 116
11.14.1.4 LoadPearlArpes() . . . 116
11.14.1.5 LoadPearlOptics() . . . 116
11.14.1.6 LoadPearlPreparation() . . . 116
11.14.1.7 PearlAnglescanTracker(string epicsname, string wbRGB) . . . 116
11.14.1.8 PearlLiveDisplay(string epicsname, string nickname, string wbRGB) . . . 116
11.14.1.9 PearlMenuEnableFunc(string funcname) . . . 116
11.15 pearl-otf-import.ipf File Reference . . . 116
11.15.1 Function Documentation . . . 116
11.15.1.1 gather_batch(string foldermatch, string ywavematch, string xwavematch, string destfolder) . . . 116
11.15.1.2 otf_gather_batch(string ywavematch, string xwavematch, string destfolder) . . . 117
11.15.1.3 otf_gather_iterator(dfref df, string *sdata) . . . 117
11.15.1.4 otf_interp(variable e1, variable e2, variable npts, variable smo) . . . 117
11.15.1.5 otf_load_itx(string pathname, string filename) . . . 117
11.15.1.6 otf_load_itx_all(string pathname) . . . 117
11.15.1.7 otf_load_itx_match(string pathname, string matchstr) . . . 117
11.15.1.8 otf_rename_folders(string pattern, variable unique_index=defaultValue, string new_suffix=defaultValue, string match_str=defaultValue) . . . 117
11.15.1.9 otf_rename_folders_iterator(dfref df, string *sdata) . . . 117
11.15.1.10 otf_smo_int(wave win, wave wout, wave wpe, variable smo) . . . 117

11.16pearl-polar-coordinates.ipf File Reference . . . 117
11.16.1 Function Documentation . . . 118
11.16.1.1 cart2polar(variable xx, variable yy, variable zz, variable *radius, variable *theta, variable *phi) . . . 118
11.16.1.2 cart2polar_wave(wave in, wave out) . . . 118
11.16.1.3 polar2cart(variable radius, variable theta, variable phi, variable *xx, variable *yy, variable *zz) . . . 118
11.16.1.4 polar2cart_wave(wave in, wave out) . . . 118
11.16.1.5 polar_distance(variable polar1, variable azim1, variable polar2, variable azim2) . . . 118
11.17pearl-pshell-import.ipf File Reference . . . 118
11.17.1 Detailed Description . . . 120
11.17.2 Function Documentation . . . 121
11.17.2.1 find_attr_folder(dfref dataDF) . . . 121
11.17.2.2 find_scale_wave(string name, dfref dataDF, dfref scanDF, dfref attrDF) . . . 121
11.17.2.3 find_scan_folder(dfref dataDF) . . . 121
11.17.2.4 ps_detect_scale(wave ax, wave lo, wave hi, wave un) . . . 121
11.17.2.5 ps_scale_dataset(wave data) . . . 122
11.17.2.6 ps_scale_dataset_2(wave data, wave ax, wave lo, wave hi, wave un) . . . 122
11.17.2.7 ps_scale_datasets() . . . 123
11.17.2.8 ps_set_dimlabels(wave data) . . . 123
11.17.2.9 ps_set_dimlabels2(wave data, string name) . . . 124
11.17.2.10 psh5_close_file(variable fileID) . . . 124
11.17.2.11 psh5_list_scan_datasets(variable fileID, string scanpath, variable include_↩ regions=defaultValue) . . . 124
11.17.2.12 psh5_list_scan_regions(variable fileID, string scanpath) . . . 125
11.17.2.13 psh5_list_scans(variable fileID) . . . 125
11.17.2.14 psh5_load_complete(string ANickName, string APathName, string AFileName, variable load_data=defaultValue, variable load_attr=defaultValue) . . . 125
11.17.2.15 psh5_load_dataset(variable fileID, string scanpath, string datasetname, variable set_scale=defaultValue) . . . 126
11.17.2.16 psh5_load_dataset_meta(variable fileID, string datapath, string datasetname, wave datawave) . . . 127
11.17.2.17 psh5_load_dataset_reduced(variable fileID, string scanpath, string datasetname, funcref reduction_func, string reduction_param, variable progress=defaultValue, variable nthreads=defaultValue) . . . 127
11.17.2.18 psh5_load_dataset_slab(variable fileID, string datapath, string datasetname, variable dim2start, variable dim2count, variable dim3start, variable dim3count) . . . 128
11.17.2.19 psh5_load_dataset_slabs(variable fileID, string datapath, string datasetname, variable progress=defaultValue) . . . 129
11.17.2.20 psh5_load_info(string APathName, string AFileName) . . . 129
11.17.2.21 psh5_load_preview(string APathName, string AFileName, variable load_↩ data=defaultValue, variable load_attr=defaultValue, string pref_scans=default↩ Value, string pref_datasets=defaultValue) . . . 129

11.17.2.22psh5_load_reduced(string ANickName, string APathName, string AFileName, funcref reduction_func, string reduction_param, variable progress=defaultValue) 130
11.17.2.23psh5_load_scan_attrs(variable fileID, string scanpath, variable attr_sets=defaultValue) 131
11.17.2.24psh5_load_scan_complete(variable fileID, string scanpath, variable load_data=defaultValue, variable load_attr=defaultValue) 131
11.17.2.25psh5_load_scan_data(variable fileID, string scanpath) 132
11.17.2.26psh5_load_scan_info(variable fileID, string scanpath) 132
11.17.2.27psh5_load_scan_meta(variable fileID, string scanpath) 133
11.17.2.28psh5_load_scan_preview(variable fileID, string scanpath, variable set_scale=defaultValue, string pref_datasets=defaultValue) 133
11.17.2.29psh5_load_scan_section(variable fileID, string scanpath, variable dim, variable set_scale=defaultValue, string pref_datasets=defaultValue) 134
11.17.2.30psh5_open_file(string ANickName, string APathName, string AFileName) 134
11.17.2.31reduce_slab_image(wave slabdata, wave image, wave profile1, wave profile2, funcref reduction_func, string reduction_param) 135
11.17.2.32reduce_slab_worker(funcref reduction_func) 135
11.17.2.33select_dataset(string file_datasets, string pref_datasets) 135
11.17.2.34twave2list(wave wt, string sep) 135
11.17.2.35wave2list(wave w, string format, string sep) 135
11.17.3 Variable Documentation 136
11.17.3.1 kAngleDimLabel 136
11.17.3.2 kDataDimLabel 136
11.17.3.3 kDetectorSensitivity 136
11.17.3.4 kEnergyDimLabel 136
11.17.3.5 kPreviewDatasets 136
11.17.3.6 kScanDimLabel 136
11.17.3.7 kScientaScalingDatasets 136
11.17.3.8 kTransposedDatasets 136
11.18pearl-scienta-preprocess.ipf File Reference 137
11.18.1 Detailed Description 137
11.18.2 Function Documentation 138
11.18.2.1 capture_int_linbg_cursors() 138
11.18.2.2 csr_int_linbg_reduction(string win) 138
11.18.2.3 fit_scienta_ang_transm(wave data, wave params) 138
11.18.2.4 fit_scienta_poly_bg(wave data, wave params, variable bgterms) 138
11.18.2.5 gauss2_reduction(wave source, wave dest1, wave dest2, string *param) 138
11.18.2.6 int_linbg_reduction(wave source, wave dest1, wave dest2, string *param) 139
11.18.2.7 int_quadbg_reduction(wave source, wave dest1, wave dest2, string *param) 139
11.18.2.8 prompt_gauss2_reduction(string *param) 139
11.18.2.9 prompt_int_linbg_reduction(string *param) 139
11.18.2.10prompt_int_quadbg_reduction(string *param) 139

11.18.2.11 prompt_redim_linbg_reduction(string *param) . . . 139
11.18.2.12 prompt_Shockley_anglefit(string *param) . . . 139
11.18.2.13 redim_linbg_reduction(wave source, wave dest1, wave dest2, string *param) . . . 139
11.18.2.14 scienta_ang_transm(wave w, variable x) . . . 140
11.18.2.15 scienta_norm(wave w, variable x) . . . 140
11.18.2.16 scienta_poly_bg(wave w, variable e, variable a) . . . 140
11.18.2.17 Shockley_anglefit(wave source, wave dest1, wave dest2, string *param) . . . 140
11.18.2.18 test_gauss2_reduction(wave image) . . . 141
11.18.2.19 test_int_linbg(wave image) . . . 141
11.18.2.20 test_int_quadbg(wave image) . . . 141
11.18.2.21 test_shockley_anglefit(wave image, variable branch) . . . 141
11.19 pearl-tools.ipf File Reference . . . 141
11.19.1 Function Documentation . . . 141
11.19.1.1 AppendToGraphIterator(wave w, string *sdata) . . . 141
11.19.1.2 DefaultFolderIterator(dfref df, string *sdata) . . . 141
11.19.1.3 DefaultWaveIterator(wave w, string *sdata) . . . 141
11.19.1.4 IterateDataFolders(string matchStr, funcref iterator, string sdata, string progress_title=defaultValue) . . . 142
11.19.1.5 IterateWaves(string matchStr, funcref iterator, string sdata) . . . 142
11.19.1.6 SumWavesIterator(wave w, string *sdata) . . . 142
11.20 pearl-vector-operations.ipf File Reference . . . 142
11.20.1 Function Documentation . . . 142
11.20.1.1 create_rotation_matrix_free() . . . 142
11.20.1.2 rotate2d_x(variable xx, variable yy, variable angle) . . . 142
11.20.1.3 rotate2d_y(variable xx, variable yy, variable angle) . . . 142
11.20.1.4 rotate_x_wave(wave inout, variable angle) . . . 142
11.20.1.5 rotate_y_wave(wave inout, variable angle) . . . 142
11.20.1.6 rotate_z_wave(wave inout, variable angle) . . . 143
11.20.1.7 set_rotation_x(wave matrix, variable angle) . . . 143
11.20.1.8 set_rotation_y(wave matrix, variable angle) . . . 143
11.20.1.9 set_rotation_z(wave matrix, variable angle) . . . 143

Index **145**

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

An open distribution of PEARL Procedures is available under the `Apache License, Version 2.0` at `https://git.psi.ch/pearl-public/igor-procs`. Users of PEARL Procedures are requested to coordinate and share the development of the code with the original author. Please read and respect the respective license agreements.

Author

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

Version

This documentation is compiled from version rev-distro-1.4.0-1-g0a436db-dirty.

Copyright

2009-2016 by Paul Scherrer Institut
Licensed under the Apache License, Version 2.0

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

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.

Global psh5_load_reduced (string ANickName, string APathName, string AFileName, funcref reduction_↩ func, string reduction_param, variable progress=defaultValue)

load scan positions.

Chapter 4

Module Index

4.1 Packages

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

ARPES package 15

Chapter 5

Namespace Index

5.1 Namespace List

Here is a list of all namespaces with brief descriptions:

PearlAnglescanProcess
Processing and holographic mapping of angle scanned XPD data 17
PearlAreaDisplay
Instant visualization of angle scan and manipulator position 17
PearlAreaImport
HDF5 file import from EPICS area detectors 18
PearlAreaProfiles
Profile extraction for multi-dimensional datasets acquired from area detectors 18
PearlArpes
Data acquisition and analysis package for ARPES at PEARL 18
PearlDataExplorer
Preview and import panel for PEARL data 18
PearlElog
Interface for writing ELOG entries with Igor graphs as attachment 19
PearlMatrixImport
Data file import for omicron matrix (STM) files 19
PearlPShellImport
Import data from PShell 19
PearlScientaPreprocess
Preprocessing functions for Scienta detector images 19

Chapter 6

Data Structure Index

6.1 Data Structures

Here are the data structures with brief descriptions:

DoniachSunjicStruct 21
errorCode
From matrixfilereader help 22

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

fermi-edge-analysis.ipf
: Tools for analysing the Fermi edge measured by the Scienta EW4000 analyser . . . 25

pearl-anglescan-process.ipf
: Processing and holographic mapping of angle scanned XPD data . . . 27

pearl-anglescan-tracker.ipf . . . 50

pearl-area-display.ipf
: Visualization tools for 2D and 3D data . . . 58

pearl-area-import.ipf
: HDF5 file import from EPICS area detectors . . . 69

pearl-area-profiles.ipf
: Profile extraction for multi-dimensional datasets acquired from area detectors . . . 78

pearl-arpes.ipf
: Data acquisition and analysis package for ARPES at PEARL . . . 81

pearl-data-explorer.ipf
: Preview and import panel for PEARL data . . . 83

pearl-elog.ipf
: Interface for writing ELOG entries with Igor graphs as attachment . . . 91

pearl-fitfuncs.ipf . . . 105

pearl-gui-tools.ipf . . . 107

pearl-matrix-import.ipf
: Data file import for omicron matrix (STM) files . . . 108

pearl-menu.ipf . . . 115

pearl-otf-import.ipf . . . 116

pearl-polar-coordinates.ipf . . . 117

pearl-pshell-import.ipf
: Import data from PShell . . . 118

pearl-scienta-preprocess.ipf
: Preprocessing functions for Scienta detector images . . . 137

pearl-tools.ipf . . . 141

pearl-vector-operations.ipf . . . 142

Chapter 8

Module Documentation

8.1 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-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-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-pshell-import.ipf

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

  preprocessing functions for Scienta detector images.

8.1.1 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-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

Chapter 9

Namespace Documentation

9.1 PearlAnglescanProcess Namespace Reference

processing and holographic mapping of angle scanned XPD data.

9.1.1 Detailed Description

processing and holographic mapping of angle scanned XPD data.

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

9.2 PearlAreaDisplay Namespace Reference

instant visualization of angle scan and manipulator position.

9.2.1 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

9.2.2 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.

9.3 PearlAreaImport Namespace Reference

HDF5 file import from EPICS area detectors.

9.3.1 Detailed Description

HDF5 file import from EPICS area detectors.

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

9.4 PearlAreaProfiles Namespace Reference

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

9.4.1 Detailed Description

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

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

9.5 PearlArpes Namespace Reference

data acquisition and analysis package for ARPES at PEARL.

9.5.1 Detailed Description

data acquisition and analysis package for ARPES at PEARL.

PearlArpes is declared in pearl-arpes.ipf.

9.6 PearlDataExplorer Namespace Reference

preview and import panel for PEARL data

9.6.1 Detailed Description

preview and import panel for PEARL data

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

9.7 PearlElog Namespace Reference

interface for writing ELOG entries with Igor graphs as attachment.

9.7.1 Detailed Description

interface for writing ELOG entries with Igor graphs as attachment.

PearlElog is declared in pearl-elog.ipf.

9.8 PearlMatrixImport Namespace Reference

data file import for omicron matrix (STM) files

9.8.1 Detailed Description

data file import for omicron matrix (STM) files

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

9.9 PearlPShellImport Namespace Reference

import data from PShell

9.9.1 Detailed Description

import data from PShell

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

9.10 PearlScientaPreprocess Namespace Reference

preprocessing functions for Scienta detector images.

9.10.1 Detailed Description

preprocessing functions for Scienta detector images.

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

Chapter 10

Data Structure Documentation

10.1 DoniachSunjicStruct Struct Reference

Data Fields

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

10.1.1 Detailed Description

Definition at line 191 of file pearl-fitfuncs.ipf.

10.1.2 Field Documentation

10.1.2.1 wave DoniachSunjicStruct::broadening

Definition at line 207 of file pearl-fitfuncs.ipf.

10.1.2.2 wave DoniachSunjicStruct::convolution

Definition at line 208 of file pearl-fitfuncs.ipf.

10.1.2.3 wave DoniachSunjicStruct::model

Definition at line 206 of file pearl-fitfuncs.ipf.

10.1.2.4 variable DoniachSunjicStruct::oversampling

Definition at line 201 of file pearl-fitfuncs.ipf.

10.1.2.5 variable DoniachSunjicStruct::precision

Definition at line 200 of file pearl-fitfuncs.ipf.

10.1.2.6 wave DoniachSunjicStruct::pw

Definition at line 191 of file pearl-fitfuncs.ipf.

10.1.2.7 wave DoniachSunjicStruct::xdw

Definition at line 205 of file pearl-fitfuncs.ipf.

10.1.2.8 wave DoniachSunjicStruct::xw

Definition at line 197 of file pearl-fitfuncs.ipf.

10.1.2.9 wave DoniachSunjicStruct::yw

Definition at line 196 of file pearl-fitfuncs.ipf.

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

- pearl-fitfuncs.ipf

10.2 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

10.2.1 Detailed Description

from matrixfilereader help

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

10.2.2 Field Documentation

10.2.2.1 int32 errorCode::ALREADY_FILE_OPEN

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

10.2.2.2 int32 errorCode::EMPTY_RESULTFILE

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

10.2.2.3 int32 errorCode::FILE_NOT_READABLE

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

10.2.2.4 int32 errorCode::INTERNAL_ERROR_CONVERTING_DATA

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

10.2.2.5 int32 errorCode::INVALID_RANGE

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

10.2.2.6 int32 errorCode::NO_FILE_OPEN

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

10.2.2.7 int32 errorCode::NO_NEW_BRICKLETS

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

10.2.2.8 int32 errorCode::SUCCESS

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

10.2.2.9 int32 errorCode::UNKNOWN_ERROR

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

10.2.2.10 int32 errorCode::WAVE_EXIST

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

10.2.2.11 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 11

File Documentation

11.1 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

11.1.1 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

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

11.1.2 Function Documentation

11.1.2.1 variable analyse_curved_edge (wave *data*)

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

11.1.2.2 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.

11.1.2.3 threadsafe variable FermiFuncLinDOS2D_corr (variable *w*, threadsafe *x*, wave *y*)

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

11.1.2.4 variable integrate_curved_edge (wave *data*, wave *data_sig*)

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

11.1.2.5 variable record_results (variable *index*)

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

11.1.2.6 variable show_shift (wave *data*)

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

11.1.2.7 variable slit_correction (wave *data,* wave *data_out,* variable *epass*)

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

11.1.2.8 threadsafe variable slit_shift (variable *ypix,* variable *epass*)

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

11.1.3 Variable Documentation

11.1.3.1 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.

11.1.3.2 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.

11.1.3.3 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.

11.1.3.4 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.

11.2 mainpage.dox File Reference

11.3 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_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 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=default↩ Value)

  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=default↩ Value)

  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 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)

  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=defaultValue)

  set the pseudocolor contrast by percentile.

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

11.3.1 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

2013-17 Paul Scherrer Institut
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Version

1.8 canonical orientation of spherical coordinate system.

11.3.2 Function Documentation

11.3.2.1 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 2129 of file pearl-anglescan-process.ipf.

11.3.2.2 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 2076 of file pearl-anglescan-process.ipf.

11.3.2.3 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 2031 of file pearl-anglescan-process.ipf.

11.3.2.4 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 935 of file pearl-anglescan-process.ipf.

11.3.2.5 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 964 of file pearl-anglescan-process.ipf.

11.3.2.6 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 1008 of file pearl-anglescan-process.ipf.

11.3.2.7 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 1039 of file pearl-anglescan-process.ipf.

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

Definition at line 2925 of file pearl-anglescan-process.ipf.

11.3.2.9 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 1318 of file pearl-anglescan-process.ipf.

11.3.2.10 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

Parameters

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 874 of file pearl-anglescan-process.ipf.

11.3.2.11 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 835 of file pearl-anglescan-process.ipf.

11.3.2.12 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 495 of file pearl-anglescan-process.ipf.

11.3.2.13 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.

Parameters

projection	mapping function from polar to cartesian coordinates • 0 linear • 1 stereographic (default) • 2 azimuthal • 3 gnomonic (0 <= polar < 90).
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.
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 1528 of file pearl-anglescan-process.ipf.

11.3.2.14 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.
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 1665 of file pearl-anglescan-process.ipf.

11.3.2.15 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 1916 of file pearl-anglescan-process.ipf.

11.3.2.16 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 1852 of file pearl-anglescan-process.ipf.

11.3.2.17 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.
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 1780 of file pearl-anglescan-process.ipf.

11.3.2.18 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 1365 of file pearl-anglescan-process.ipf.

11.3.2.19 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 1284 of file pearl-anglescan-process.ipf.

11.3.2.20 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 1228 of file pearl-anglescan-process.ipf.

11.3.2.21 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 1253 of file pearl-anglescan-process.ipf.

11.3.2.22 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 2275 of file pearl-anglescan-process.ipf.

11.3.2.23 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 2348 of file pearl-anglescan-process.ipf.

11.3.2.24 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 2874 of file pearl-anglescan-process.ipf.

11.3.2.25 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 2790 of file pearl-anglescan-process.ipf.

11.3.2.26 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

Parameters

xpdplot	XPDplot compatibility • 0 (default) create waves in child folder $nickname • 1 create waves in root folder (compatible with XPDplot)

Definition at line 2699 of file pearl-anglescan-process.ipf.

11.3.2.27 variable interpolate_hemi_scan (string *nickname*)

interpolate a hemispherical scan onto a rectangular grid

Warning

experimental this function has been tested for one specific set of scan parameters. the interface and code may change at any time. the function depends on the ster_x and ster_y waves that are created by display_↩ hemi_scan.

Definition at line 2449 of file pearl-anglescan-process.ipf.

11.3.2.28 static variable line_average (wave *source,* wave *dest*) [static]

Definition at line 917 of file pearl-anglescan-process.ipf.

11.3.2.29 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 2650 of file pearl-anglescan-process.ipf.

11.3.2.30 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 1077 of file pearl-anglescan-process.ipf.

11.3.2.31 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 435 of file pearl-anglescan-process.ipf.

11.3.2.32 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
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 270 of file pearl-anglescan-process.ipf.

11.3.2.33 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.
`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 353 of file pearl-anglescan-process.ipf.

11.3.2.34 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.

Parameters

in,out	*strip*	2D data, X-axis = analyser angle, Y-axis = arbitrary manipulator scan
in	*smooth_method*	smoothing method • 0 none • 1 (default) binomial, see Igor's Smooth operation • 2 boxcar, see Igor's Smooth operation • 3 scienta_ang_transm() function fit • 4 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 185 of file pearl-anglescan-process.ipf.

11.3.2.35 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.

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_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 555 of file pearl-anglescan-process.ipf.

11.3.2.36 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.

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.
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 648 of file pearl-anglescan-process.ipf.

11.3.2.37 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 2209 of file pearl-anglescan-process.ipf.

11.3.2.38 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 2503 of file pearl-anglescan-process.ipf.

11.3.2.39 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 1449 of file pearl-anglescan-process.ipf.

11.3.2.40 variable save_hemi_scan (string *nickname,* string *pathname,* string *filename*)

save a hemispherical scan to an Igor text file

Definition at line 2614 of file pearl-anglescan-process.ipf.

11.3.2.41 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 2961 of file pearl-anglescan-process.ipf.

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

Definition at line 2229 of file pearl-anglescan-process.ipf.

11.3.2.43 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

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 765 of file pearl-anglescan-process.ipf.

11.3.2.44 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.

11.3.2.45 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 2748 of file pearl-anglescan-process.ipf.

11.3.2.46 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 2174 of file pearl-anglescan-process.ipf.

11.3.3 Variable Documentation

11.3.3.1 const variable kProjArea = 2

Definition at line 2006 of file pearl-anglescan-process.ipf.

11.3.3.2 const variable kProjDist = 0

Definition at line 2004 of file pearl-anglescan-process.ipf.

11.3.3.3 const variable kProjGnom = 3

Definition at line 2007 of file pearl-anglescan-process.ipf.

11.3.3.4 const variable kProjOrtho = 4

Definition at line 2008 of file pearl-anglescan-process.ipf.

11.3.3.5 const variable kProjScaleArea = 2 [static]

Definition at line 2012 of file pearl-anglescan-process.ipf.

11.3.3.6 const variable kProjScaleDist = 2 [static]

Definition at line 2010 of file pearl-anglescan-process.ipf.

11.3.3.7 const variable kProjScaleGnom = 0.06744519021 [static]

Definition at line 2014 of file pearl-anglescan-process.ipf.

11.3.3.8 const variable kProjScaleOrtho = 2 [static]

Definition at line 2015 of file pearl-anglescan-process.ipf.

11.3.3.9 const variable kProjScaleStereo = 2 [static]

Definition at line 2011 of file pearl-anglescan-process.ipf.

11.3.3.10 const variable kProjStereo = 1

Definition at line 2005 of file pearl-anglescan-process.ipf.

11.4 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

11.4.1 Function Documentation

11.4.1.1 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.

11.4.1.2 static variable AfterCompiledHook ( ) [static]

initialize package data once when the procedure is first loaded

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

11.4.1.3 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.

11.4.1.4 variable ast_callback_data (variable *chan*)

callback function for new analyser data from EPICS.

Definition at line 846 of file pearl-anglescan-tracker.ipf.

11.4.1.5 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 951 of file pearl-anglescan-tracker.ipf.

11.4.1.6 variable ast_callback_manip (variable *chan*)

callback function for new manipulator position from EPICS.

Definition at line 980 of file pearl-anglescan-tracker.ipf.

11.4.1.7 variable ast_close ( )

stop tracker, close graph, release data structures.

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

11.4.1.8 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.

11.4.1.9 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.

11.4.1.10 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.

11.4.1.11 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.

11.4.1.12 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.

11.4.1.13 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
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.

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

window hook

disconnects from EPICS when the window is closed.

Definition at line 831 of file pearl-anglescan-tracker.ipf.

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

Definition at line 1013 of file pearl-anglescan-tracker.ipf.

11.4.1.16 static variable edit_offsets ( ) [static]

Definition at line 1234 of file pearl-anglescan-tracker.ipf.

11.4.1.17 static variable edit_reduction_params ( ) [static]

Definition at line 1218 of file pearl-anglescan-tracker.ipf.

11.4.1.18 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 675 of file pearl-anglescan-tracker.ipf.

11.4.1.19 static variable epics_disconnect ( ) [static]

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

11.4.1.20 static variable epics_disconnect_chid (string *chid_var_name*) [static]

Definition at line 787 of file pearl-anglescan-tracker.ipf.

11.4.1.21 static variable export_tracker_data ( ) [static]

export tracker data (with prompt)

Definition at line 1095 of file pearl-anglescan-tracker.ipf.

11.4.1.22 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.

11.4.1.23 static variable IgorQuitHook (string *app*) [static]

disconnect EPICS channels before Igor quits.

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

11.4.1.24 static variable import_tracker_data ( ) [static]

import tracker data (with prompt)

Definition at line 1125 of file pearl-anglescan-tracker.ipf.

11.4.1.25 static variable init_package ( ) [static]

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

11.4.1.26 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.

11.4.1.27 static variable load_tracker_data ( ) [static]

import tracker data from file (with prompt)

Definition at line 1165 of file pearl-anglescan-tracker.ipf.

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

Definition at line 1061 of file pearl-anglescan-tracker.ipf.

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

Definition at line 1075 of file pearl-anglescan-tracker.ipf.

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

Definition at line 1184 of file pearl-anglescan-tracker.ipf.

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

Definition at line 1198 of file pearl-anglescan-tracker.ipf.

11.4.1.32 static variable process_image_data ( ) [static]

process the data buffer to generate the tracker dataset.

Definition at line 502 of file pearl-anglescan-tracker.ipf.

11.4.1.33 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.

11.4.1.34 static variable save_tracker_data ( ) [static]

save tracker data to file (with prompt)

Definition at line 1154 of file pearl-anglescan-tracker.ipf.

11.4.1.35 static variable setup_data ( ) [static]

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

11.4.1.36 static variable setup_detector ( ) [static]

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

11.4.1.37 static variable setup_graph ( ) [static]

create the graph window.

Definition at line 589 of file pearl-anglescan-tracker.ipf.

11.4.1.38 static variable toggle_capture ( ) [static]

Definition at line 1027 of file pearl-anglescan-tracker.ipf.

11.4.1.39 static variable update_capture ( ) [static]

Definition at line 1045 of file pearl-anglescan-tracker.ipf.

11.4.1.40 static variable update_data_graph ( ) [static]

Definition at line 640 of file pearl-anglescan-tracker.ipf.

11.4.1.41 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 554 of file pearl-anglescan-tracker.ipf.

11.4.1.42 static variable update_detector_graph ( ) [static]

Definition at line 652 of file pearl-anglescan-tracker.ipf.

11.4.2 Variable Documentation

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

data folder path

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

11.4.2.2 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.

11.4.2.3 version

Initial value:

```
= 1.6
```

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

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

11.5 pearl-area-display.ipf File Reference

visualization tools for 2D and 3D data.

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.
- 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.

11.5.1 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.

11.5.2 2D data

TO DO...

11.5.3 3D data

11.5.4 Function Documentation

11.5.4.1 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.

Parameters

options	smoothing factors in `key1=value1;key2=value2;...` format. • SmoothingX • SmoothingY

Definition at line 1044 of file pearl-area-display.ipf.

11.5.4.2 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 1197 of file pearl-area-display.ipf.

11.5.4.3 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 726 of file pearl-area-display.ipf.

11.5.4.4 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 1007 of file pearl-area-display.ipf.

11.5.4.5 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 773 of file pearl-area-display.ipf.

11.5.4.6 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 1030 of file pearl-area-display.ipf.

11.5.4.7 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 84 of file pearl-area-display.ipf.

11.5.4.8 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 1082 of file pearl-area-display.ipf.

11.5.4.9 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 114 of file pearl-area-display.ipf.

11.5.4.10 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 165 of file pearl-area-display.ipf.

11.5.4.11 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 1345 of file pearl-area-display.ipf.

11.5.4.12 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 875 of file pearl-area-display.ipf.

11.5.4.13 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.

Parameters

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 1422 of file pearl-area-display.ipf.

11.5.4.14 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 481 of file pearl-area-display.ipf.

11.5.4.15 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 368 of file pearl-area-display.ipf.

11.5.4.16 variable ad_profiles_hook (WMWinHookStruct * *s*)

hook function for user events in the profiles window.

Definition at line 686 of file pearl-area-display.ipf.

11.5.4.17 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 429 of file pearl-area-display.ipf.

11.5.4.18 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 1458 of file pearl-area-display.ipf.

11.5.4.19 variable ad_slicer_init_bg ( )

initialize the slice animation background task.

Definition at line 1717 of file pearl-area-display.ipf.

11.5.4.20 static variable ad_slicer_move_bg (WMBackgroundStruct * *s*) [static]

move a slice by one step (background task).

Definition at line 1678 of file pearl-area-display.ipf.

11.5.4.21 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 1741 of file pearl-area-display.ipf.

11.5.4.22 variable ad_slicer_stop_bg (string *posvariable*)

stop the animation.

Parameters

posvariable	full path to the global position variable.

Definition at line 1783 of file pearl-area-display.ipf.

11.5.4.23 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 1066 of file pearl-area-display.ipf.

11.5.4.24 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 321 of file pearl-area-display.ipf.

11.5.4.25 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 1612 of file pearl-area-display.ipf.

11.5.4.26 static variable bp_move_slice (WMButtonAction * *ba*) [static]

move slice (button procedure).

Definition at line 1563 of file pearl-area-display.ipf.

11.5.4.27 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 1666 of file pearl-area-display.ipf.

11.5.4.28 static variable bp_reset_cursors (WMButtonAction * *ba*) [static]

Definition at line 618 of file pearl-area-display.ipf.

11.5.4.29 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 547 of file pearl-area-display.ipf.

11.5.4.30 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 585 of file pearl-area-display.ipf.

11.5.4.31 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 608 of file pearl-area-display.ipf.

11.5.4.32 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 56 of file pearl-area-display.ipf.

11.5.4.33 static dfr make_view_folder (wave *source*) [static]

create a view data folder.

Definition at line 561 of file pearl-area-display.ipf.

11.5.4.34 static variable pmp_export (WMPopupAction * *pa*) [static]

Definition at line 662 of file pearl-area-display.ipf.

11.5.4.35 static variable set_trace_colors (string *graphname*) [static]

Definition at line 986 of file pearl-area-display.ipf.

11.5.4.36 static variable slp_slice_position (WMSliderAction * *sa*) [static]

set slice coordinate (slider procedure).

Definition at line 1499 of file pearl-area-display.ipf.

11.5.4.37 static variable svp_slice_position (WMSetVariableAction * *sva*) [static]

set slice coordinate (button procedure).

Definition at line 1531 of file pearl-area-display.ipf.

11.5.4.38 static variable svp_smoothing (WMSetVariableAction * *sva*) [static]

Definition at line 637 of file pearl-area-display.ipf.

11.5.4.39 static variable update_slice_info ( ) [static]

update controls with data scale limits.

current folder must be slicer info

Definition at line 1386 of file pearl-area-display.ipf.

11.6 pearl-area-import.ipf File Reference

HDF5 file import from EPICS area detectors.

```
#include <HDF5 Browser>
#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=defaultValue, 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 variable adh5_default_reduction (wave source, wave dest1, wave dest2, 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, wave dest1, wave dest2, funcref reduction_func, string reduction_param)

  wrapper function for testing reduction functions from the command line.
- 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 variable reduce_slab_image (wave slabdata, wave image, wave profile1, wave profile2, 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

11.6.1 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.

11.6.2 Function Documentation

11.6.2.1 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 159 of file pearl-area-import.ipf.

11.6.2.2 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 98 of file pearl-area-import.ipf.

11.6.2.3 threadsafe variable adh5_default_reduction (wave *source*, wave *dest1*, wave *dest2*, string * *param*)

function prototype for adh5_load_reduced_detector

derived functions reduce a two-dimensional dataset to a one-dimensional dataset, e.g. by ROI-integration, 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 dest1 and dest2 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)
dest1,dest2	destination waves
param	string with optional parameters, shared between calls. this is a pass-by-reference argument, the function may modify the string

Returns

zero if successful, non-zero if an error occurs.

Definition at line 1093 of file pearl-area-import.ipf.

11.6.2.4 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 1033 of file pearl-area-import.ipf.

11.6.2.5 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 199 of file pearl-area-import.ipf.

11.6.2.6 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 573 of file pearl-area-import.ipf.

11.6.2.7 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 937 of file pearl-area-import.ipf.

11.6.2.8 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 772 of file pearl-area-import.ipf.

11.6.2.9 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 489 of file pearl-area-import.ipf.

11.6.2.10 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 379 of file pearl-area-import.ipf.

11.6.2.11 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 294 of file pearl-area-import.ipf.

11.6.2.12 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 1162 of file pearl-area-import.ipf.

11.6.2.13 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 1479 of file pearl-area-import.ipf.

11.6.2.14 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 618 of file pearl-area-import.ipf.

11.6.2.15 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 718 of file pearl-area-import.ipf.

11.6.2.16 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 1719 of file pearl-area-import.ipf.

11.6.2.17 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 1614 of file pearl-area-import.ipf.

11.6.2.18 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 1113 of file pearl-area-import.ipf.

11.6.2.19 string adh5_test_reduction_func (wave *source,* wave *dest1,* wave *dest2,* funcref *reduction_func,* string *reduction_param*)

wrapper function for testing reduction functions from the command line.

Igor does not allow global variables as pass-by-reference parameter for reduction_param.

Definition at line 1128 of file pearl-area-import.ipf.

11.6.2.20 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 34 of file pearl-area-import.ipf.

11.6.2.21 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 699 of file pearl-area-import.ipf.

11.6.2.22 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 1561 of file pearl-area-import.ipf.

11.6.2.23 static threadsafe variable reduce_slab_image (wave *slabdata,* wave *image,* wave *profile1,* wave *profile2,* funcref *reduction_func,* string *reduction_param*) [static]

Definition at line 1444 of file pearl-area-import.ipf.

11.6.2.24 static threadsafe variable reduce_slab_worker (funcref *reduction_func*) [static]

Definition at line 1405 of file pearl-area-import.ipf.

11.7 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.

11.7.1 Detailed Description

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

Author

matthias muntwiler, matthias.muntwiler@psi.ch

Copyright

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

11.7.2 Function Documentation

11.7.2.1 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.

11.7.2.2 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	
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.

11.7.2.3 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.

11.7.2.4 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.

11.7.2.5 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.

11.7.2.6 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.

11.7.2.7 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.

11.7.2.8 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.

11.7.2.9 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.

11.7.2.10 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.

11.7.2.11 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.

11.7.2.12 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.

11.7.2.13 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.

11.7.2.14 threadsafe variable calc_y_profile_mins (wave *image*)

Definition at line 603 of file pearl-area-profiles.ipf.

11.8 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-data-explorer"
#include "pearl-anglescan-process"
#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 ()

11.8.1 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

2012-15 Paul Scherrer Institut
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

11.8.2 Function Documentation

11.8.2.1 static variable AfterCompiledHook ( ) [static]

initializes package data once when the procedure is first loaded

Definition at line 81 of file pearl-arpes.ipf.

11.8.2.2 variable UnloadPearlArpesPackage ( )

Definition at line 88 of file pearl-arpes.ipf.

11.9 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-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 ()
- 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)
- 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"

11.9.1 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).

11.9.2 Function Documentation

11.9.2.1 static variable attributes_notebook (wave *attr_names,* wave *attr_values,* string *title*) [static]

Definition at line 805 of file pearl-data-explorer.ipf.

11.9.2.2 static variable bp_attr_notebook (WMButtonAction * *ba*) [static]

Definition at line 1733 of file pearl-data-explorer.ipf.

11.9.2.3 static variable bp_browse_filepath (WMButtonAction * *ba*) [static]

Definition at line 1419 of file pearl-data-explorer.ipf.

11.9.2.4 static variable bp_dataset_display (WMButtonAction * *ba*) [static]

Definition at line 1624 of file pearl-data-explorer.ipf.

11.9.2.5 static variable bp_dataset_folder (WMButtonAction * *ba*) [static]

Definition at line 1598 of file pearl-data-explorer.ipf.

11.9.2.6 static variable bp_dataset_next (WMButtonAction * *ba*) [static]

Definition at line 1644 of file pearl-data-explorer.ipf.

11.9.2.7 static variable bp_dataset_prev (WMButtonAction * *ba*) [static]

Definition at line 1671 of file pearl-data-explorer.ipf.

11.9.2.8 static variable bp_file_next (WMButtonAction * *ba*) [static]

Definition at line 1485 of file pearl-data-explorer.ipf.

11.9.2.9 static variable bp_file_prev (WMButtonAction * *ba*) [static]

Definition at line 1517 of file pearl-data-explorer.ipf.

11.9.2.10 static variable bp_load_files (WMButtonAction * *ba*) [static]

Definition at line 1457 of file pearl-data-explorer.ipf.

11.9.2.11 static variable bp_load_files_opt (WMButtonAction * *ba*) [static]

Definition at line 1471 of file pearl-data-explorer.ipf.

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

Definition at line 1391 of file pearl-data-explorer.ipf.

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

Definition at line 1405 of file pearl-data-explorer.ipf.

11.9.2.14 static variable bp_update_datasets (WMButtonAction * *ba*) [static]

Definition at line 1584 of file pearl-data-explorer.ipf.

11.9.2.15 static variable bp_update_filelist (WMButtonAction * *ba*) [static]

Definition at line 1443 of file pearl-data-explorer.ipf.

11.9.2.16 static variable display_dataset (string *datasetname*) [static]

Definition at line 765 of file pearl-data-explorer.ipf.

11.9.2.17 static string display_preview_trace (wave *xtrace,* wave *ytrace*) [static]

Definition at line 922 of file pearl-data-explorer.ipf.

11.9.2.18 static variable extract_preview_image (wave *data,* wave *preview*) [static]

Definition at line 478 of file pearl-data-explorer.ipf.

11.9.2.19 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 55 of file pearl-data-explorer.ipf.

11.9.2.20 string itx_suggest_foldername (string *filename,* variable *ignoredate* = defaultValue, string *sourcename* = defaultValue, variable *unique* = defaultValue)

Definition at line 1241 of file pearl-data-explorer.ipf.

11.9.2.21 static variable lbp_datasets (WMListboxAction * *lba*) [static]

Definition at line 1698 of file pearl-data-explorer.ipf.

11.9.2.22 static variable lbp_filelist (WMListboxAction * *lba*) [static]

Definition at line 1549 of file pearl-data-explorer.ipf.

11.9.2.23 static variable load_file (string *filename,* string *options* = defaultValue) [static]

Definition at line 979 of file pearl-data-explorer.ipf.

11.9.2.24 static dfr load_hdf_file (string *filename,* string *options =* defaultValue) [static]

Definition at line 1125 of file pearl-data-explorer.ipf.

11.9.2.25 static dfr load_itx_file (string *filename,* string *options =* defaultValue) [static]

Definition at line 1175 of file pearl-data-explorer.ipf.

11.9.2.26 static dfr load_mtrx_file (string *filename,* string *options =* defaultValue) [static]

load a matrix (STM) data file

Definition at line 1218 of file pearl-data-explorer.ipf.

11.9.2.27 static variable load_prefs ( ) [static]

Definition at line 116 of file pearl-data-explorer.ipf.

11.9.2.28 static dfr load_pshell_file (string *filename,* string *options =* defaultValue) [static]

Definition at line 1073 of file pearl-data-explorer.ipf.

11.9.2.29 static variable load_selected_files (string *options =* defaultValue) [static]

Definition at line 955 of file pearl-data-explorer.ipf.

11.9.2.30 static variable notebook_add_attributes (string *notebook_name,* wave *attr_filter,* wave *attr_names,* wave *attr_values*) [static]

Definition at line 844 of file pearl-data-explorer.ipf.

11.9.2.31 variable pearl_data_explorer ( )

Definition at line 44 of file pearl-data-explorer.ipf.

11.9.2.32 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 161 of file pearl-data-explorer.ipf.

11.9.2.33 void PearlDataExplorer ( )

Definition at line 1300 of file pearl-data-explorer.ipf.

11.9.2.34 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 631 of file pearl-data-explorer.ipf.

11.9.2.35 static variable preview_datafolder ( ) [static]

Definition at line 560 of file pearl-data-explorer.ipf.

11.9.2.36 static variable preview_dataset (string *datasetname*) [static]

Definition at line 517 of file pearl-data-explorer.ipf.

11.9.2.37 static variable preview_file (string *filename*) [static]

Definition at line 255 of file pearl-data-explorer.ipf.

11.9.2.38 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 363 of file pearl-data-explorer.ipf.

11.9.2.39 static wave preview_itx_file (string *filename*) [static]

Definition at line 395 of file pearl-data-explorer.ipf.

11.9.2.40 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 435 of file pearl-data-explorer.ipf.

11.9.2.41 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 307 of file pearl-data-explorer.ipf.

11.9.2.42 static variable preview_setscale_x (wave *data,* wave *preview*) [static]

Definition at line 601 of file pearl-data-explorer.ipf.

11.9.2.43 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 1053 of file pearl-data-explorer.ipf.

11.9.2.44 variable prompt_func_params (string *func_name,* string ∗ *func_param*)

Definition at line 1059 of file pearl-data-explorer.ipf.

11.9.2.45 static variable prompt_hdf_options (string ∗ *options*) [static]

Definition at line 1014 of file pearl-data-explorer.ipf.

11.9.2.46 static variable save_prefs ( ) [static]

Definition at line 96 of file pearl-data-explorer.ipf.

11.9.2.47 static string show_preview_graph (wave *data*, wave *xdata* = defaultValue) [static]

Definition at line 871 of file pearl-data-explorer.ipf.

11.9.2.48 variable test_attributes_notebook ( )

Definition at line 798 of file pearl-data-explorer.ipf.

11.9.2.49 static variable update_datasets ( ) [static]

Definition at line 212 of file pearl-data-explorer.ipf.

11.9.2.50 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 184 of file pearl-data-explorer.ipf.

11.9.3 Variable Documentation

11.9.3.1 const string ks_filematch_adh5 = "*.h5" [static]

Definition at line 39 of file pearl-data-explorer.ipf.

11.9.3.2 const string ks_filematch_itx = "*.itx" [static]

Definition at line 41 of file pearl-data-explorer.ipf.

11.9.3.3 const string ks_filematch_mtrx = "*_mtrx" [static]

Definition at line 42 of file pearl-data-explorer.ipf.

11.9.3.4 const string ks_filematch_pshell = "psh*.h5" [static]

Definition at line 40 of file pearl-data-explorer.ipf.

11.9.3.5 const string package_name = "pearl_explorer" [static]

Definition at line 36 of file pearl-data-explorer.ipf.

11.9.3.6 const string package_path = "root:packages:pearl_explorer:" [static]

Definition at line 37 of file pearl-data-explorer.ipf.

11.10 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

11.10.1 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

2013-17 Paul Scherrer Institut
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

11.10.2 Function Documentation

11.10.2.1 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.

11.10.2.2 static variable bp_attach (WMButtonAction * *ba*) [static]

Definition at line 1634 of file pearl-elog.ipf.

11.10.2.3 static variable bp_attach_allnone (WMButtonAction * *ba*) [static]

select/deselect all graph windows for attachment

Definition at line 1613 of file pearl-elog.ipf.

11.10.2.4 static variable bp_attach_top (WMButtonAction * *ba*) [static]

select top graph window for attachment

Definition at line 1597 of file pearl-elog.ipf.

11.10.2.5 static variable bp_attach_updown (WMButtonAction * *ba*) [static]

button procedure for the attachment up and down buttons

Definition at line 1518 of file pearl-elog.ipf.

11.10.2.6 static variable bp_clear (WMButtonAction * *ba*) [static]

Definition at line 1695 of file pearl-elog.ipf.

11.10.2.7 static variable bp_login (WMButtonAction * *ba*) [static]

Definition at line 1711 of file pearl-elog.ipf.

11.10.2.8 static variable bp_logout (WMButtonAction * *ba*) [static]

Definition at line 1729 of file pearl-elog.ipf.

11.10.2.9 static variable bp_save_graphs (WMButtonAction * *ba*) [static]

Definition at line 1667 of file pearl-elog.ipf.

11.10.2.10 static variable bp_submit (WMButtonAction * *ba*) [static]

button procedure for the Submit and Reply buttons

Definition at line 1550 of file pearl-elog.ipf.

11.10.2.11 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.

11.10.2.12 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.

11.10.2.13 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.

11.10.2.14 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.

11.10.2.15 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.

11.10.2.16 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.

11.10.2.17 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.

11.10.2.18 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.

11.10.2.19 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.

11.10.2.20 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.

11.10.2.21 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.

11.10.2.22 static variable elog_panel_hook (WMWinHookStruct * *s*) [static]

Definition at line 1394 of file pearl-elog.ipf.

11.10.2.23 string elog_prompt_logbook ( )

prompt to open or create a logbook

Definition at line 1184 of file pearl-elog.ipf.

11.10.2.24 variable elog_prompt_login (string *logbook*)

prompt the user for login to a logbook

Definition at line 1214 of file pearl-elog.ipf.

11.10.2.25 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.

11.10.2.26 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.

11.10.2.27 static string get_default_panel_name ( ) [static]

Definition at line 1746 of file pearl-elog.ipf.

11.10.2.28 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.

11.10.2.29 static string get_log_path ( ) [static]

Definition at line 1095 of file pearl-elog.ipf.

11.10.2.30 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 1759 of file pearl-elog.ipf.

11.10.2.31 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 1914 of file pearl-elog.ipf.

11.10.2.32 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 1868 of file pearl-elog.ipf.

11.10.2.33 static string get_timestamp (string *sep*) [static]

Definition at line 958 of file pearl-elog.ipf.

11.10.2.34 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.

11.10.2.35 static variable IgorQuitHook (string *igorApplicationNameStr*) [static]

save preferences and recent values before Igor quits.

Definition at line 135 of file pearl-elog.ipf.

11.10.2.36 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.

11.10.2.37 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.

11.10.2.38 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.

11.10.2.39 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.

11.10.2.40 static variable move_attach_item (string *logbook,* variable *item,* variable *distance*) [static]

move an attachment item in the list of attachments

Definition at line 1494 of file pearl-elog.ipf.

11.10.2.41 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.

11.10.2.42 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.

11.10.2.43 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.

11.10.2.44 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.

11.10.2.45 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.

11.10.2.46 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.

11.10.2.47 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 1806 of file pearl-elog.ipf.

11.10.2.48 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 1954 of file pearl-elog.ipf.

11.10.2.49 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 1894 of file pearl-elog.ipf.

11.10.2.50 static variable update_attach_items (string *logbook*) [static]

update the list of attachments

Definition at line 1433 of file pearl-elog.ipf.

11.10.3 Variable Documentation

11.10.3.1 const string elog_parse_id = "ID=%u" [static]

Definition at line 1134 of file pearl-elog.ipf.

11.10.3.2 const string elog_success_msg = "Message successfully transmitted" [static]

Definition at line 1133 of file pearl-elog.ipf.

11.10.3.3 const variable kAttachColName = 2 [static]

Definition at line 1430 of file pearl-elog.ipf.

11.10.3.4 const variable kAttachColSel = 0 [static]

Definition at line 1428 of file pearl-elog.ipf.

11.10.3.5 const variable kAttachColTitle = 1 [static]

Definition at line 1429 of file pearl-elog.ipf.

11.10.3.6 const variable kdfPersistent = 2 [static]

Definition at line 155 of file pearl-elog.ipf.

11.10.3.7 const variable kdfRoot = 0 [static]

Definition at line 153 of file pearl-elog.ipf.

11.10.3.8 const variable kdfTemplates = 3 [static]

Definition at line 156 of file pearl-elog.ipf.

11.10.3.9 const variable kdfVolatile = 1 [static]

Definition at line 154 of file pearl-elog.ipf.

11.10.3.10 const string package_name = "pearl_elog" [static]

Definition at line 85 of file pearl-elog.ipf.

11.10.3.11 const string package_path = "root:packages:pearl_elog:" [static]

Definition at line 86 of file pearl-elog.ipf.

11.11 pearl-fitfuncs.ipf File Reference

```
#include "mm-physconst"
```

Data Structures

- struct DoniachSunjicStruct

Functions

- 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.

Variables

- version

11.11.1 Function Documentation

11.11.1.1 variable Au4f (wave *w,* variable *x*)

Definition at line 419 of file pearl-fitfuncs.ipf.

11.11.1.2 variable Au4f_2p2 (wave *w,* variable *x*)

Definition at line 464 of file pearl-fitfuncs.ipf.

11.11.1.3 variable Au4f_2p3 (wave *w,* variable *x*)

Definition at line 549 of file pearl-fitfuncs.ipf.

11.11.1.4 variable Calc_DoniachSunjicBroad (wave *pw,* wave *yw*)

Definition at line 362 of file pearl-fitfuncs.ipf.

11.11.1.5 variable DoniachSunjicBroad (wave *pw,* wave *yw,* wave *xw*)

Definition at line 301 of file pearl-fitfuncs.ipf.

11.11.1.6 threadsafe variable DoniachSunjicBroadS (DoniachSunjicStruct * *s*)

Definition at line 212 of file pearl-fitfuncs.ipf.

11.11.1.7 threadsafe variable ds1_bg (wave *w,* variable *x*)

Definition at line 38 of file pearl-fitfuncs.ipf.

11.11.1.8 threadsafe variable ds2_bg (wave *w,* variable *x*)

Definition at line 61 of file pearl-fitfuncs.ipf.

11.11.1.9 variable ds4_bg (wave *w,* variable *x*)

Definition at line 89 of file pearl-fitfuncs.ipf.

11.11.1.10 variable ds6_bg (wave *w,* variable *x*)

Definition at line 130 of file pearl-fitfuncs.ipf.

11.11.1.11 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 672 of file pearl-fitfuncs.ipf.

11.11.1.12 variable Fit_DoniachSunjicBroad (wave *pw,* wave *yw,* wave *xw,* wave *ww*)

Definition at line 383 of file pearl-fitfuncs.ipf.

11.11.1.13 variable ShowComponents_Au4f_2p2 (wave *coef_wave,* wave *fit_wave*)

Definition at line 518 of file pearl-fitfuncs.ipf.

11.11.1.14 variable ShowComponents_Au4f_2p3 (wave *coef_wave,* wave *fit_wave*)

Definition at line 617 of file pearl-fitfuncs.ipf.

11.11.2 Variable Documentation

11.11.2.1 version

Definition at line 5 of file pearl-fitfuncs.ipf.

11.12 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 ()

11.12.1 Function Documentation

11.12.1.1 variable display_progress_panel (string *title,* string *message,* variable *progress_max*)

Definition at line 18 of file pearl-gui-tools.ipf.

11.12.1.2 variable kill_progress_panel ( )

Definition at line 50 of file pearl-gui-tools.ipf.

11.12.1.3 variable update_progress_panel (variable *progress,* string *message* = `defaultValue`, variable *progress_max* = `defaultValue`)

Definition at line 32 of file pearl-gui-tools.ipf.

11.13 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 scan↩Cycle=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"

11.13.1 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

2016 Paul Scherrer Institut
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

11.13.2 Function Documentation

11.13.2.1 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.

11.13.2.2 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.

11.13.2.3 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.

11.13.2.4 static variable init_package ( ) [static]

initialize the package data folder.

Definition at line 45 of file pearl-matrix-import.ipf.

11.13.2.5 static variable initStruct (errorCode * *errorCode*) [static]

from matrixfilereader help

Definition at line 212 of file pearl-matrix-import.ipf.

11.13.2.6 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.

11.13.2.7 variable matrix_preview_2d (wave *data,* wave *metadata*)

Definition at line 145 of file pearl-matrix-import.ipf.

11.13.2.8 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.

11.13.2.9 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.

11.13.2.10 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.

11.13.2.11 variable mtrx_load_all ( )

load all data from a Matrix data file.

Definition at line 231 of file pearl-matrix-import.ipf.

11.13.2.12 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.

11.13.2.13 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.

11.13.2.14 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.

11.13.2.15 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.

11.13.2.16 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.

11.13.2.17 variable mtrx_scale_dataset (wave *data*)

Definition at line 829 of file pearl-matrix-import.ipf.

11.13.2.18 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.

11.13.2.19 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.

11.13.2.20 variable subtract_line_bg (wave *img*)

remove linear background line-by-line

Definition at line 886 of file pearl-matrix-import.ipf.

11.13.3 Variable Documentation

11.13.3.1 const string ks_filematch_mtrx = "*_mtrx" [static]

Definition at line 40 of file pearl-matrix-import.ipf.

11.13.3.2 const string package_name = "pearl_matrix_import" [static]

Definition at line 37 of file pearl-matrix-import.ipf.

11.13.3.3 const string package_path = "root:packages:pearl_matrix_import:" [static]

Definition at line 38 of file pearl-matrix-import.ipf.

11.14 pearl-menu.ipf File Reference

Functions

- string PearlMenuEnableFunc (string funcname)
- variable LoadPearlOptics ()
- variable LoadPearlArpes ()
- variable LoadPearlPreparation ()
- variable Display2dProfiles ()
- variable Display3dSlicer ()
- variable DisplayGizmoSlicer ()
- variable PearlLiveDisplay (string epicsname, string nickname, string wbRGB)
- variable PearlAnglescanTracker (string epicsname, string wbRGB)

11.14.1 Function Documentation

11.14.1.1 variable Display2dProfiles ( )

Definition at line 108 of file pearl-menu.ipf.

11.14.1.2 variable Display3dSlicer ( )

Definition at line 125 of file pearl-menu.ipf.

11.14.1.3 variable DisplayGizmoSlicer ( )

Definition at line 144 of file pearl-menu.ipf.

11.14.1.4 variable LoadPearlArpes ( )

Definition at line 96 of file pearl-menu.ipf.

11.14.1.5 variable LoadPearlOptics ( )

Definition at line 89 of file pearl-menu.ipf.

11.14.1.6 variable LoadPearlPreparation ( )

Definition at line 102 of file pearl-menu.ipf.

11.14.1.7 variable PearlAnglescanTracker (string *epicsname,* string *wbRGB*)

Definition at line 182 of file pearl-menu.ipf.

11.14.1.8 variable PearlLiveDisplay (string *epicsname,* string *nickname,* string *wbRGB*)

Definition at line 163 of file pearl-menu.ipf.

11.14.1.9 string PearlMenuEnableFunc (string *funcname*)

Definition at line 77 of file pearl-menu.ipf.

11.15 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)

11.15.1 Function Documentation

11.15.1.1 variable gather_batch (string *foldermatch,* string *ywavematch,* string *xwavematch,* string *destfolder*)

Definition at line 162 of file pearl-otf-import.ipf.

11.15.1.2 variable otf_gather_batch (string *ywavematch,* string *xwavematch,* string *destfolder*)

Definition at line 152 of file pearl-otf-import.ipf.

11.15.1.3 variable otf_gather_iterator (dfref *df,* string ∗ *sdata*)

Definition at line 106 of file pearl-otf-import.ipf.

11.15.1.4 variable otf_interp (variable *e1,* variable *e2,* variable *npts,* variable *smo*)

Definition at line 271 of file pearl-otf-import.ipf.

11.15.1.5 variable otf_load_itx (string *pathname,* string *filename*)

Definition at line 76 of file pearl-otf-import.ipf.

11.15.1.6 variable otf_load_itx_all (string *pathname*)

Definition at line 31 of file pearl-otf-import.ipf.

11.15.1.7 variable otf_load_itx_match (string *pathname,* string *matchstr*)

Definition at line 53 of file pearl-otf-import.ipf.

11.15.1.8 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.

11.15.1.9 variable otf_rename_folders_iterator (dfref *df,* string ∗ *sdata*)

Definition at line 185 of file pearl-otf-import.ipf.

11.15.1.10 variable otf_smo_int (wave *win,* wave *wout,* wave *wpe,* variable *smo*)

Definition at line 304 of file pearl-otf-import.ipf.

11.16 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)

11.16.1 Function Documentation

11.16.1.1 variable cart2polar (variable *xx,* variable *yy,* variable *zz,* variable * *radius,* variable * *theta,* variable * *phi*)

Definition at line 10 of file pearl-polar-coordinates.ipf.

11.16.1.2 variable cart2polar_wave (wave *in,* wave *out*)

Definition at line 36 of file pearl-polar-coordinates.ipf.

11.16.1.3 variable polar2cart (variable *radius,* variable *theta,* variable *phi,* variable * *xx,* variable * *yy,* variable * *zz*)

Definition at line 48 of file pearl-polar-coordinates.ipf.

11.16.1.4 variable polar2cart_wave (wave *in,* wave *out*)

Definition at line 58 of file pearl-polar-coordinates.ipf.

11.16.1.5 variable polar_distance (variable *polar1,* variable *azim1,* variable *polar2,* variable *azim2*)

Definition at line 69 of file pearl-polar-coordinates.ipf.

11.17 pearl-pshell-import.ipf File Reference

import data from PShell

```
#include <HDF5 Browser>
#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)

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 variable reduce_slab_image (wave slabdata, wave image, wave profile1, wave profile2, 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 = 4

  multiply scienta detector intensity by this value to get actual counts.

11.17.1 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()

11.17.2 Function Documentation

11.17.2.1 static dfr find_attr_folder (dfref *dataDF*) [static]

find the attributes data folder

this is the :attr folder.

Definition at line 1451 of file pearl-pshell-import.ipf.

11.17.2.2 static wave find_scale_wave (string *name,* dfref *dataDF,* dfref *scanDF,* dfref *attrDF*) [static]

Definition at line 1539 of file pearl-pshell-import.ipf.

11.17.2.3 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 1434 of file pearl-pshell-import.ipf.

11.17.2.4 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 1596 of file pearl-pshell-import.ipf.

11.17.2.5 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 1525 of file pearl-pshell-import.ipf.

11.17.2.6 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.

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 1750 of file pearl-pshell-import.ipf.

11.17.2.7 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 1479 of file pearl-pshell-import.ipf.

11.17.2.8 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 1358 of file pearl-pshell-import.ipf.

11.17.2.9 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 1377 of file pearl-pshell-import.ipf.

11.17.2.10 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 132 of file pearl-pshell-import.ipf.

11.17.2.11 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 432 of file pearl-pshell-import.ipf.

11.17.2.12 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 473 of file pearl-pshell-import.ipf.

11.17.2.13 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 397 of file pearl-pshell-import.ipf.

11.17.2.14 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 162 of file pearl-pshell-import.ipf.

11.17.2.15 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".
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 681 of file pearl-pshell-import.ipf.

11.17.2.16 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 1066 of file pearl-pshell-import.ipf.

11.17.2.17 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` and `ReducedData2` 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".

Parameters

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	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 (e.g. for debugging the reduction function). • >= 1 use a fixed number of (additional) threads.

Returns

semicolon-separated list of the loaded waves, `ReducedData1` and `ReducedData2` if successful. 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 1970 of file pearl-pshell-import.ipf.

11.17.2.18 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 1259 of file pearl-pshell-import.ipf.

11.17.2.19 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 1123 of file pearl-pshell-import.ipf.

11.17.2.20 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 2286 of file pearl-pshell-import.ipf.

11.17.2.21 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
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 243 of file pearl-pshell-import.ipf.

11.17.2.22 string psh5_load_reduced (string *ANickName,* string *APathName,* string *AFileName,* funcref *reduction_func,* string *reduction_param,* variable *progress* = `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.

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.
progress	progress window. • 1 (default) show progress window • 0 do not show progress window

Returns

semicolon-separated list of the loaded waves, `ReducedData1` and `ReducedData2` if successful. 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.

Todo load scan positions.

Definition at line 1846 of file pearl-pshell-import.ipf.

11.17.2.23 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 546 of file pearl-pshell-import.ipf.

11.17.2.24 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.

Parameters

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 353 of file pearl-pshell-import.ipf.

11.17.2.25 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 506 of file pearl-pshell-import.ipf.

11.17.2.26 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 2333 of file pearl-pshell-import.ipf.

11.17.2.27 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)

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 614 of file pearl-pshell-import.ipf.

11.17.2.28 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.

Parameters

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 815 of file pearl-pshell-import.ipf.

11.17.2.29 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 928 of file pearl-pshell-import.ipf.

11.17.2.30 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).
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 102 of file pearl-pshell-import.ipf.

11.17.2.31 static threadsafe variable reduce_slab_image (wave *slabdata,* wave *image,* wave *profile1,* wave *profile2,* funcref *reduction_func,* string *reduction_param*) [static]

Definition at line 2258 of file pearl-pshell-import.ipf.

11.17.2.32 static threadsafe variable reduce_slab_worker (funcref *reduction_func*) [static]

Definition at line 2219 of file pearl-pshell-import.ipf.

11.17.2.33 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 753 of file pearl-pshell-import.ipf.

11.17.2.34 static string twave2list (wave *wt,* string *sep*) [static]

convert text wave to list.

Definition at line 2392 of file pearl-pshell-import.ipf.

11.17.2.35 static string wave2list (wave *w,* string *format,* string *sep*) [static]

convert numeric wave to list.

Definition at line 2409 of file pearl-pshell-import.ipf.

11.17.3 Variable Documentation

11.17.3.1 const string kAngleDimLabel = "angle"

Dimension label for the angle dispersive dimension of multi-dimensional datasets.

Definition at line 60 of file pearl-pshell-import.ipf.

11.17.3.2 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 67 of file pearl-pshell-import.ipf.

11.17.3.3 const variable kDetectorSensitivity = 4

multiply scienta detector intensity by this value to get actual counts.

Definition at line 79 of file pearl-pshell-import.ipf.

11.17.3.4 const string kEnergyDimLabel = "energy"

Dimension label for the energy dispersive dimension of multi-dimensional datasets.

Definition at line 57 of file pearl-pshell-import.ipf.

11.17.3.5 const string kPreviewDatasets = "ScientaImage;ScientaSpectrum;ImageAngleDistribution;ImageEnergy↩ Distribution;Counts;SampleCurrent;"

List of preferred datasets to load for preview.

Definition at line 70 of file pearl-pshell-import.ipf.

11.17.3.6 const string kScanDimLabel = "scan"

Dimension label for the scan dimension of multi-dimensional datasets.

Definition at line 63 of file pearl-pshell-import.ipf.

11.17.3.7 const string kScientaScalingDatasets = "LensMode;ScientaChannelBegin;ScientaChannelEnd;ScientaSlice↩ Begin;ScientaSliceEnd;"

List of datasets that must be loaded to determine the axis scaling of a Scienta image.

Definition at line 73 of file pearl-pshell-import.ipf.

11.17.3.8 const string kTransposedDatasets = "ScientaImage;"

List of datasets that should be transposed upon loading.

Definition at line 76 of file pearl-pshell-import.ipf.

11.18 pearl-scienta-preprocess.ipf File Reference

preprocessing functions for Scienta detector images.

```
#include "mm-fitfuncs"
```

Namespaces

- PearlScientaPreprocess
  preprocessing functions for Scienta detector images.

Functions

- variable prompt_int_linbg_reduction (string *param)
- string capture_int_linbg_cursors ()
- string csr_int_linbg_reduction (string win)
- variable test_int_linbg (wave image)
- threadsafe variable int_linbg_reduction (wave source, wave dest1, wave dest2, string *param)
- variable test_shockley_anglefit (wave image, variable branch)
- variable prompt_Shockley_anglefit (string *param)
- threadsafe variable Shockley_anglefit (wave source, wave dest1, wave dest2, string *param)
- variable prompt_int_quadbg_reduction (string *param)
- variable test_int_quadbg (wave image)
- threadsafe variable int_quadbg_reduction (wave source, wave dest1, wave dest2, string *param)
- variable scienta_norm (wave w, variable x)
- wave fit_scienta_ang_transm (wave data, wave params)
- threadsafe variable scienta_ang_transm (wave w, variable x)
- wave fit_scienta_poly_bg (wave data, wave params, variable bgterms)
- variable scienta_poly_bg (wave w, variable e, variable a)
- variable prompt_redim_linbg_reduction (string *param)
  parameter dialog for the redim_linbg_reduction() function
- threadsafe variable redim_linbg_reduction (wave source, wave dest1, wave dest2, string *param)
  linear background reduction function for incorrectly dimensioned scienta image
- variable test_gauss2_reduction (wave image)
  apply the gauss2_reduction function to a single image
- variable prompt_gauss2_reduction (string *param)
  prompt for the gauss2_reduction parameters
- threadsafe variable gauss2_reduction (wave source, wave dest1, wave dest2, string *param)
  fit horizontal cuts of an image with two gaussian peaks on a linear background

11.18.1 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

2013-17 Paul Scherrer Institut
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

11.18.2 Function Documentation

11.18.2.1 string capture_int_linbg_cursors ( )

Definition at line 66 of file pearl-scienta-preprocess.ipf.

11.18.2.2 string csr_int_linbg_reduction (string *win*)

Definition at line 76 of file pearl-scienta-preprocess.ipf.

11.18.2.3 wave fit_scienta_ang_transm (wave *data,* wave *params*)

Definition at line 551 of file pearl-scienta-preprocess.ipf.

11.18.2.4 wave fit_scienta_poly_bg (wave *data,* wave *params,* variable *bgterms*)

Definition at line 610 of file pearl-scienta-preprocess.ipf.

11.18.2.5 threadsafe variable gauss2_reduction (wave *source,* wave *dest1,* wave *dest2,* string ∗ *param*)

fit horizontal cuts of an image with two gaussian peaks on a linear background

the function fits each horizontal profile (EDC) with two 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 either of peak 1 or peak 2. 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).
dest1	(out) peak area
dest2	(out) error estimate of peak area
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 • return select result to return, either "int1" or "int2". • ybox box size of averaging in y direction, must be 1 or 3. other values lead to corrupt data.

Returns

zero if successful, non-zero if an error occurs.

Definition at line 903 of file pearl-scienta-preprocess.ipf.

11.18.2.6 threadsafe variable int_linbg_reduction (wave *source*, wave *dest1*, wave *dest2*, string * *param*)

Definition at line 197 of file pearl-scienta-preprocess.ipf.

11.18.2.7 threadsafe variable int_quadbg_reduction (wave *source*, wave *dest1*, wave *dest2*, string * *param*)

Definition at line 457 of file pearl-scienta-preprocess.ipf.

11.18.2.8 variable prompt_gauss2_reduction (string * *param*)

prompt for the gauss2_reduction parameters

useful for testing or manual processing. to debug, (temporarily) remove the threadsafe attribute from the gauss2_reduction function.

Definition at line 834 of file pearl-scienta-preprocess.ipf.

11.18.2.9 variable prompt_int_linbg_reduction (string * *param*)

Definition at line 36 of file pearl-scienta-preprocess.ipf.

11.18.2.10 variable prompt_int_quadbg_reduction (string * *param*)

Definition at line 410 of file pearl-scienta-preprocess.ipf.

11.18.2.11 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 716 of file pearl-scienta-preprocess.ipf.

11.18.2.12 variable prompt_Shockley_anglefit (string * *param*)

Definition at line 322 of file pearl-scienta-preprocess.ipf.

11.18.2.13 threadsafe variable redim_linbg_reduction (wave *source*, wave *dest1*, wave *dest2*, 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
dest1	destination wave 1
dest2	destination wave 2 each wave is a one-dimensional intensity distribution the function may redimension these waves to one of the image dimensions (it must be clear to the user which dimension this is). the meaning of dest1 and dest2 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.
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

zero if successful, non-zero if an error occurs.

Definition at line 787 of file pearl-scienta-preprocess.ipf.

11.18.2.14 threadsafe variable scienta_ang_transm (wave *w,* variable *x*)

Definition at line 578 of file pearl-scienta-preprocess.ipf.

11.18.2.15 variable scienta_norm (wave *w,* variable *x*)

Definition at line 544 of file pearl-scienta-preprocess.ipf.

11.18.2.16 variable scienta_poly_bg (wave *w,* variable *e,* variable *a*)

Definition at line 654 of file pearl-scienta-preprocess.ipf.

11.18.2.17 threadsafe variable Shockley_anglefit (wave *source,* wave *dest1,* wave *dest2,* string $*$ *param*)

Definition at line 337 of file pearl-scienta-preprocess.ipf.

11.18.2.18 variable test_gauss2_reduction (wave *image*)

apply the gauss2_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 810 of file pearl-scienta-preprocess.ipf.

11.18.2.19 variable test_int_linbg (wave *image*)

Definition at line 180 of file pearl-scienta-preprocess.ipf.

11.18.2.20 variable test_int_quadbg (wave *image*)

Definition at line 440 of file pearl-scienta-preprocess.ipf.

11.18.2.21 variable test_shockley_anglefit (wave *image,* variable *branch*)

Definition at line 297 of file pearl-scienta-preprocess.ipf.

11.19 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)

11.19.1 Function Documentation

11.19.1.1 variable AppendToGraphIterator (wave *w,* string ∗ *sdata*)

Definition at line 26 of file pearl-tools.ipf.

11.19.1.2 variable DefaultFolderIterator (dfref *df,* string ∗ *sdata*)

Definition at line 65 of file pearl-tools.ipf.

11.19.1.3 variable DefaultWaveIterator (wave *w,* string ∗ *sdata*)

Definition at line 18 of file pearl-tools.ipf.

11.19.1.4 string IterateDataFolders (string *matchStr,* funcref *iterator,* string *sdata,* string *progress_title* = `defaultValue`)

Definition at line 79 of file pearl-tools.ipf.

11.19.1.5 string IterateWaves (string *matchStr,* funcref *iterator,* string *sdata*)

Definition at line 45 of file pearl-tools.ipf.

11.19.1.6 variable SumWavesIterator (wave *w,* string ∗ *sdata*)

Definition at line 34 of file pearl-tools.ipf.

11.20 pearl-vector-operations.ipf File Reference

Functions

- variable rotate2d_x (variable xx, variable yy, variable angle)
- variable rotate2d_y (variable xx, variable yy, variable angle)
- wave create_rotation_matrix_free ()
- wave set_rotation_x (wave matrix, variable angle)
- wave set_rotation_y (wave matrix, variable angle)
- wave set_rotation_z (wave matrix, variable angle)
- variable rotate_x_wave (wave inout, variable angle)
- variable rotate_y_wave (wave inout, variable angle)
- variable rotate_z_wave (wave inout, variable angle)

11.20.1 Function Documentation

11.20.1.1 wave create_rotation_matrix_free ( )

Definition at line 26 of file pearl-vector-operations.ipf.

11.20.1.2 variable rotate2d_x (variable *xx,* variable *yy,* variable *angle*)

Definition at line 10 of file pearl-vector-operations.ipf.

11.20.1.3 variable rotate2d_y (variable *xx,* variable *yy,* variable *angle*)

Definition at line 18 of file pearl-vector-operations.ipf.

11.20.1.4 variable rotate_x_wave (wave *inout,* variable *angle*)

Definition at line 83 of file pearl-vector-operations.ipf.

11.20.1.5 variable rotate_y_wave (wave *inout,* variable *angle*)

Definition at line 101 of file pearl-vector-operations.ipf.

11.20.1.6 variable rotate_z_wave (wave *inout,* variable *angle*)

Definition at line 119 of file pearl-vector-operations.ipf.

11.20.1.7 wave set_rotation_x (wave *matrix,* variable *angle*)

Definition at line 35 of file pearl-vector-operations.ipf.

11.20.1.8 wave set_rotation_y (wave *matrix,* variable *angle*)

Definition at line 51 of file pearl-vector-operations.ipf.

11.20.1.9 wave set_rotation_z (wave *matrix,* variable *angle*)

Definition at line 67 of file pearl-vector-operations.ipf.

Index

ALREADY_FILE_OPEN
    errorCode, 23
ARPES package, 15
ad_box_filter
    pearl-area-display.ipf, 60
ad_brick_slicer
    pearl-area-display.ipf, 61
ad_calc_cursor_profiles
    pearl-area-display.ipf, 61
ad_calc_histogram
    pearl-area-display.ipf, 61
ad_calc_profiles
    pearl-area-display.ipf, 61
ad_collect_multiscan_y
    pearl-area-profiles.ipf, 79
ad_default_image_filter
    pearl-area-display.ipf, 62
ad_display
    pearl-area-display.ipf, 62
ad_display_brick
    pearl-area-display.ipf, 62
ad_display_histogram
    pearl-area-display.ipf, 63
ad_display_profiles
    pearl-area-display.ipf, 63
ad_display_slice
    pearl-area-display.ipf, 63
ad_export_profile
    pearl-area-display.ipf, 64
ad_extract_rod
    pearl-area-profiles.ipf, 79
ad_extract_rod_x
    pearl-area-profiles.ipf, 80
ad_extract_rod_y
    pearl-area-profiles.ipf, 80
ad_extract_rod_z
    pearl-area-profiles.ipf, 80
ad_extract_slab
    pearl-area-profiles.ipf, 80
ad_extract_slab_x
    pearl-area-profiles.ipf, 80
ad_extract_slab_y
    pearl-area-profiles.ipf, 81
ad_extract_slab_z
    pearl-area-profiles.ipf, 81
ad_gizmo_set_plane
    pearl-area-display.ipf, 64
ad_load_dialog
    pearl-area-import.ipf, 71
ad_profile_x
    pearl-area-profiles.ipf, 81
ad_profile_x_w
    pearl-area-profiles.ipf, 81
ad_profile_y
    pearl-area-profiles.ipf, 81
ad_profile_y_w
    pearl-area-profiles.ipf, 81
ad_profiles_crosshairs
    pearl-area-display.ipf, 65
ad_profiles_cursor_mode
    pearl-area-display.ipf, 65
ad_profiles_hook
    pearl-area-display.ipf, 65
ad_profiles_set_cursor
    pearl-area-display.ipf, 65
ad_profiles_set_slice
    pearl-area-display.ipf, 66
ad_slicer_init_bg
    pearl-area-display.ipf, 66
ad_slicer_move_bg
    pearl-area-display.ipf, 66
ad_slicer_start_bg
    pearl-area-display.ipf, 66
ad_slicer_stop_bg
    pearl-area-display.ipf, 67
ad_suggest_foldername
    pearl-area-import.ipf, 71
ad_transpose_filter
    pearl-area-display.ipf, 67
ad_update_profiles
    pearl-area-display.ipf, 67
add_image_data
    pearl-anglescan-tracker.ipf, 52
adh5_default_reduction
    pearl-area-import.ipf, 71
adh5_list_reduction_funcs
    pearl-area-import.ipf, 72
adh5_load_complete
    pearl-area-import.ipf, 72
adh5_load_detector
    pearl-area-import.ipf, 72
adh5_load_detector_image
    pearl-area-import.ipf, 73
adh5_load_detector_slabs
    pearl-area-import.ipf, 73
adh5_load_info
    pearl-area-import.ipf, 73
adh5_load_preview

pearl-area-import.ipf, 74
adh5_load_reduced
pearl-area-import.ipf, 74
adh5_load_reduced_detector
pearl-area-import.ipf, 75
adh5_loadattr_all
pearl-area-import.ipf, 75
adh5_redim
pearl-area-import.ipf, 76
adh5_scale
pearl-area-import.ipf, 76
adh5_scale_scan
pearl-area-import.ipf, 76
adh5_scale_scienta
pearl-area-import.ipf, 76
adh5_setup_profile
pearl-area-import.ipf, 76
adh5_test_reduction_func
pearl-area-import.ipf, 77
AfterCompiledHook
pearl-anglescan-tracker.ipf, 52
pearl-arpes.ipf, 82
AfterFileOpenHook
pearl-elog.ipf, 94
pearl-matrix-import.ipf, 109
analyse_curved_edge
fermi-edge-analysis.ipf, 26
analyser_energy_resolution
fermi-edge-analysis.ipf, 26
AppendToGraphIterator
pearl-tools.ipf, 141
ast_add_image
pearl-anglescan-tracker.ipf, 52
ast_callback_data
pearl-anglescan-tracker.ipf, 52
ast_callback_detector
pearl-anglescan-tracker.ipf, 52
ast_callback_manip
pearl-anglescan-tracker.ipf, 53
ast_close
pearl-anglescan-tracker.ipf, 53
ast_export
pearl-anglescan-tracker.ipf, 53
ast_import
pearl-anglescan-tracker.ipf, 53
ast_prepare
pearl-anglescan-tracker.ipf, 53
ast_set_processing
pearl-anglescan-tracker.ipf, 54
ast_setup
pearl-anglescan-tracker.ipf, 54
ast_update_detector
pearl-anglescan-tracker.ipf, 54
ast_window_hook
pearl-anglescan-tracker.ipf, 54
attributes_notebook
pearl-data-explorer.ipf, 85
Au4f
pearl-fitfuncs.ipf, 106
Au4f_2p2
pearl-fitfuncs.ipf, 106
Au4f_2p3
pearl-fitfuncs.ipf, 106

BeforeFileOpenHook
pearl-area-import.ipf, 77
pearl-matrix-import.ipf, 109
bp_attach
pearl-elog.ipf, 94
bp_attach_allnone
pearl-elog.ipf, 94
bp_attach_top
pearl-elog.ipf, 94
bp_attach_updown
pearl-elog.ipf, 94
bp_attr_notebook
pearl-data-explorer.ipf, 85
bp_browse_filepath
pearl-data-explorer.ipf, 85
bp_capture
pearl-anglescan-tracker.ipf, 54
bp_clear
pearl-elog.ipf, 94
bp_dataset_display
pearl-data-explorer.ipf, 85
bp_dataset_folder
pearl-data-explorer.ipf, 85
bp_dataset_next
pearl-data-explorer.ipf, 85
bp_dataset_prev
pearl-data-explorer.ipf, 85
bp_extract_slice
pearl-area-display.ipf, 67
bp_file_next
pearl-data-explorer.ipf, 85
bp_file_prev
pearl-data-explorer.ipf, 85
bp_load_files
pearl-data-explorer.ipf, 85
bp_load_files_opt
pearl-data-explorer.ipf, 85
bp_load_prefs
pearl-data-explorer.ipf, 85
bp_login
pearl-elog.ipf, 95
bp_logout
pearl-elog.ipf, 95
bp_move_slice
pearl-area-display.ipf, 67
bp_move_slice_center
pearl-area-display.ipf, 68
bp_reset_cursors
pearl-area-display.ipf, 68
bp_save_graphs
pearl-elog.ipf, 95
bp_save_prefs
pearl-data-explorer.ipf, 85

bp_submit
pearl-elog.ipf, 95
bp_update_datasets
pearl-data-explorer.ipf, 86
bp_update_filelist
pearl-data-explorer.ipf, 86
broadening
DoniachSunjicStruct, 21

Calc_DoniachSunjicBroad
pearl-fitfuncs.ipf, 106
Calc_The_step
pearl-anglescan-process.ipf, 33
calc_graph_azi
pearl-anglescan-process.ipf, 31
calc_graph_polar
pearl-anglescan-process.ipf, 31
calc_graph_radius
pearl-anglescan-process.ipf, 32
calc_nth
pearl-anglescan-process.ipf, 32
calc_phi_step
pearl-anglescan-process.ipf, 33
calc_y_profile_mins
pearl-area-profiles.ipf, 81
CalcN_Theta
pearl-anglescan-process.ipf, 33
capture_int_linbg_cursors
pearl-scienta-preprocess.ipf, 138
cart2polar
pearl-polar-coordinates.ipf, 118
cart2polar_wave
pearl-polar-coordinates.ipf, 118
check_contrast
pearl-anglescan-process.ipf, 33
check_package_folder
pearl-matrix-import.ipf, 109
cleanup_temp_files
pearl-elog.ipf, 95
clear_hemi_grid
pearl-anglescan-process.ipf, 33
convert_angles_ttpa2polar
pearl-anglescan-process.ipf, 33
convert_angles_ttpd2polar
pearl-anglescan-process.ipf, 34
convolution
DoniachSunjicStruct, 21
create_cmd_file
pearl-elog.ipf, 95
create_graph_file
pearl-elog.ipf, 95
create_message_file
pearl-elog.ipf, 96
create_rotation_matrix_free
pearl-vector-operations.ipf, 142
crop_strip
pearl-anglescan-process.ipf, 34
csr_int_linbg_reduction
pearl-scienta-preprocess.ipf, 138

DefaultFolderIterator
pearl-tools.ipf, 141
DefaultWaveIterator
pearl-tools.ipf, 141
Display2dProfiles
pearl-menu.ipf, 115
Display3dSlicer
pearl-menu.ipf, 115
display_dataset
pearl-data-explorer.ipf, 86
display_hemi_scan
pearl-anglescan-process.ipf, 34
display_polar_graph
pearl-anglescan-process.ipf, 35
display_preview_trace
pearl-data-explorer.ipf, 86
display_progress_panel
pearl-gui-tools.ipf, 107
display_scanlines
pearl-anglescan-process.ipf, 36
DisplayGizmoSlicer
pearl-menu.ipf, 115
DoniachSunjicBroad
pearl-fitfuncs.ipf, 106
DoniachSunjicBroadS
pearl-fitfuncs.ipf, 106
DoniachSunjicStruct, 21
broadening, 21
convolution, 21
model, 21
oversampling, 21
precision, 21
pw, 22
xdw, 22
xw, 22
yw, 22
draw_diffraction_cone
pearl-anglescan-process.ipf, 36
draw_hemi_axes
pearl-anglescan-process.ipf, 37
ds1_bg
pearl-fitfuncs.ipf, 106
ds2_bg
pearl-fitfuncs.ipf, 106
ds4_bg
pearl-fitfuncs.ipf, 106
ds6_bg
pearl-fitfuncs.ipf, 106
duplicate_hemi_scan
pearl-anglescan-process.ipf, 37

EMPTY_RESULTFILE
errorCode, 23
edit_offsets
pearl-anglescan-tracker.ipf, 55
edit_reduction_params
pearl-anglescan-tracker.ipf, 55
elog_add_attachment
pearl-elog.ipf, 96

elog_config
pearl-elog.ipf, 97
elog_create_entry
pearl-elog.ipf, 97
elog_create_logbook
pearl-elog.ipf, 97
elog_init_pearl_templates
pearl-elog.ipf, 97
elog_login
pearl-elog.ipf, 98
elog_logout
pearl-elog.ipf, 98
elog_panel_hook
pearl-elog.ipf, 98
elog_parse_id
pearl-elog.ipf, 104
elog_prompt_logbook
pearl-elog.ipf, 98
elog_prompt_login
pearl-elog.ipf, 98
elog_success_msg
pearl-elog.ipf, 104
elog_validate_attributes
pearl-elog.ipf, 99
epics_connect
pearl-anglescan-tracker.ipf, 55
epics_disconnect
pearl-anglescan-tracker.ipf, 55
epics_disconnect_chid
pearl-anglescan-tracker.ipf, 55
errorCode, 22
ALREADY_FILE_OPEN, 23
EMPTY_RESULTFILE, 23
FILE_NOT_READABLE, 23
INTERNAL_ERROR_CONVERTING_DATA, 23
INVALID_RANGE, 23
NO_FILE_OPEN, 23
NO_NEW_BRICKLETS, 23
SUCCESS, 23
UNKNOWN_ERROR, 23
WAVE_EXIST, 23
WRONG_PARAMETER, 23
export_tracker_data
pearl-anglescan-tracker.ipf, 55
extend_data
pearl-anglescan-tracker.ipf, 55
extract_preview_image
pearl-data-explorer.ipf, 86

FILE_NOT_READABLE
errorCode, 23
fermi-edge-analysis.ipf, 25
analyse_curved_edge, 26
analyser_energy_resolution, 26
FermiFuncLinDOS2D_corr, 26
hemi_radius_mm, 27
integrate_curved_edge, 26
mcp_radius_epass, 27
mcp_radius_mm, 27
mcp_radius_pix, 27
record_results, 26
show_shift, 26
slit_correction, 27
slit_shift, 27
FermiFuncLinDOS2D_corr
fermi-edge-analysis.ipf, 26
FermiGaussConv
pearl-fitfuncs.ipf, 106
find_attr_folder
pearl-pshell-import.ipf, 121
find_hemi_data
pearl-anglescan-process.ipf, 38
find_scale_wave
pearl-pshell-import.ipf, 121
find_scan_folder
pearl-pshell-import.ipf, 121
Fit_DoniachSunjicBroad
pearl-fitfuncs.ipf, 107
fit_scienta_ang_transm
pearl-scienta-preprocess.ipf, 138
fit_scienta_poly_bg
pearl-scienta-preprocess.ipf, 138
format_url
pearl-elog.ipf, 99

gather_batch
pearl-otf-import.ipf, 116
gauss2_reduction
pearl-scienta-preprocess.ipf, 138
get_default_panel_name
pearl-elog.ipf, 99
get_elog_df
pearl-elog.ipf, 99
get_hemi_nickname
pearl-anglescan-process.ipf, 38
get_hemi_prefix
pearl-anglescan-process.ipf, 39
get_log_path
pearl-elog.ipf, 99
get_panel_attributes
pearl-elog.ipf, 100
get_panel_graphs
pearl-elog.ipf, 100
get_panel_message
pearl-elog.ipf, 100
get_source_image
pearl-area-display.ipf, 68
get_timestamp
pearl-elog.ipf, 100
get_view_folder
pearl-area-display.ipf, 68
get_view_image
pearl-area-display.ipf, 68
GetAttrDataFolderDFR
pearl-area-import.ipf, 77
graphname_from_dfref
pearl-area-display.ipf, 68

hemi_add_anglescan
pearl-anglescan-process.ipf, 39
hemi_add_aziscan
pearl-anglescan-process.ipf, 39
hemi_azi_cut
pearl-anglescan-process.ipf, 39
hemi_polar_cut
pearl-anglescan-process.ipf, 40
hemi_radius_mm
fermi-edge-analysis.ipf, 27

INTERNAL_ERROR_CONVERTING_DATA
errorCode, 23
INVALID_RANGE
errorCode, 23
IgorBeforeNewHook
pearl-elog.ipf, 100
IgorQuitHook
pearl-anglescan-tracker.ipf, 55
pearl-elog.ipf, 101
import_tpi_scan
pearl-anglescan-process.ipf, 40
import_tracker_data
pearl-anglescan-tracker.ipf, 56
init_package
pearl-anglescan-tracker.ipf, 56
pearl-data-explorer.ipf, 86
pearl-elog.ipf, 101
pearl-matrix-import.ipf, 110
init_volatile_vars
pearl-elog.ipf, 101
initStruct
pearl-matrix-import.ipf, 110
int_linbg_reduction
pearl-scienta-preprocess.ipf, 139
int_quadbg_reduction
pearl-scienta-preprocess.ipf, 139
integrate_curved_edge
fermi-edge-analysis.ipf, 26
interpolate_hemi_scan
pearl-anglescan-process.ipf, 41
IterateDataFolders
pearl-tools.ipf, 141
IterateWaves
pearl-tools.ipf, 142
itx_suggest_foldername
pearl-data-explorer.ipf, 86

kAngleDimLabel
pearl-pshell-import.ipf, 136
kAttachColName
pearl-elog.ipf, 104
kAttachColSel
pearl-elog.ipf, 104
kAttachColTitle
pearl-elog.ipf, 104
kDataDimLabel
pearl-pshell-import.ipf, 136
kDetectorSensitivity
pearl-pshell-import.ipf, 136
kEnergyDimLabel
pearl-pshell-import.ipf, 136
kPreviewDatasets
pearl-pshell-import.ipf, 136
kProjArea
pearl-anglescan-process.ipf, 49
kProjDist
pearl-anglescan-process.ipf, 49
kProjGnom
pearl-anglescan-process.ipf, 49
kProjOrtho
pearl-anglescan-process.ipf, 49
kProjScaleArea
pearl-anglescan-process.ipf, 49
kProjScaleDist
pearl-anglescan-process.ipf, 49
kProjScaleGnom
pearl-anglescan-process.ipf, 49
kProjScaleOrtho
pearl-anglescan-process.ipf, 50
kProjScaleStereo
pearl-anglescan-process.ipf, 50
kProjStereo
pearl-anglescan-process.ipf, 50
kScanDimLabel
pearl-pshell-import.ipf, 136
kScientaScalingDatasets
pearl-pshell-import.ipf, 136
kTransposedDatasets
pearl-pshell-import.ipf, 136
kdfPersistent
pearl-elog.ipf, 104
kdfRoot
pearl-elog.ipf, 104
kdfTemplates
pearl-elog.ipf, 105
kdfVolatile
pearl-elog.ipf, 105
kill_progress_panel
pearl-gui-tools.ipf, 107
ks_filematch_adh5
pearl-data-explorer.ipf, 90
ks_filematch_itx
pearl-data-explorer.ipf, 90
ks_filematch_mtrx
pearl-data-explorer.ipf, 90
pearl-matrix-import.ipf, 115
ks_filematch_pshell
pearl-data-explorer.ipf, 90

lbp_datasets
pearl-data-explorer.ipf, 86
lbp_filelist
pearl-data-explorer.ipf, 86
line_average
pearl-anglescan-process.ipf, 41
list_logbooks
pearl-elog.ipf, 101

load_file
pearl-data-explorer.ipf, 86
load_hdf_file
pearl-data-explorer.ipf, 86
load_hemi_scan
pearl-anglescan-process.ipf, 41
load_itx_file
pearl-data-explorer.ipf, 87
load_mtrx_file
pearl-data-explorer.ipf, 87
load_prefs
pearl-anglescan-tracker.ipf, 56
pearl-data-explorer.ipf, 87
pearl-elog.ipf, 101
load_pshell_file
pearl-data-explorer.ipf, 87
load_selected_files
pearl-data-explorer.ipf, 87
load_tracker_data
pearl-anglescan-tracker.ipf, 56
LoadPearlArpes
pearl-menu.ipf, 116
LoadPearlOptics
pearl-menu.ipf, 116
LoadPearlPreparation
pearl-menu.ipf, 116

mainpage.dox, 27
make_hemi_grid
pearl-anglescan-process.ipf, 41
make_view_folder
pearl-area-display.ipf, 69
matrix_format_elog_message
pearl-matrix-import.ipf, 110
matrix_preview_2d
pearl-matrix-import.ipf, 110
mcp_radius_epass
fermi-edge-analysis.ipf, 27
mcp_radius_mm
fermi-edge-analysis.ipf, 27
mcp_radius_pix
fermi-edge-analysis.ipf, 27
model
DoniachSunjicStruct, 21
move_attach_item
pearl-elog.ipf, 102
mtrx_create_folder
pearl-matrix-import.ipf, 110
mtrx_file_brickletID
pearl-matrix-import.ipf, 110
mtrx_get_cycle_folder
pearl-matrix-import.ipf, 111
mtrx_load_all
pearl-matrix-import.ipf, 111
mtrx_load_file
pearl-matrix-import.ipf, 111
mtrx_load_info
pearl-matrix-import.ipf, 112
mtrx_load_preview
pearl-matrix-import.ipf, 112
mtrx_open_file
pearl-matrix-import.ipf, 113
mtrx_parse_filename
pearl-matrix-import.ipf, 113
mtrx_scale_dataset
pearl-matrix-import.ipf, 114
mtrx_split_filename
pearl-matrix-import.ipf, 114

NO_FILE_OPEN
errorCode, 23
NO_NEW_BRICKLETS
errorCode, 23
normalize_strip_2d
pearl-anglescan-process.ipf, 42
normalize_strip_phi
pearl-anglescan-process.ipf, 42
normalize_strip_theta
pearl-anglescan-process.ipf, 43
normalize_strip_x
pearl-anglescan-process.ipf, 43
notebook_add_attributes
pearl-data-explorer.ipf, 87

otf_gather_batch
pearl-otf-import.ipf, 116
otf_gather_iterator
pearl-otf-import.ipf, 117
otf_interp
pearl-otf-import.ipf, 117
otf_load_itx
pearl-otf-import.ipf, 117
otf_load_itx_all
pearl-otf-import.ipf, 117
otf_load_itx_match
pearl-otf-import.ipf, 117
otf_rename_folders
pearl-otf-import.ipf, 117
otf_rename_folders_iterator
pearl-otf-import.ipf, 117
otf_smo_int
pearl-otf-import.ipf, 117
oversampling
DoniachSunjicStruct, 21

package_name
pearl-data-explorer.ipf, 90
pearl-elog.ipf, 105
pearl-matrix-import.ipf, 115
package_path
pearl-anglescan-tracker.ipf, 58
pearl-data-explorer.ipf, 90
pearl-elog.ipf, 105
pearl-matrix-import.ipf, 115
parse_result
pearl-elog.ipf, 102
pearl-anglescan-process.ipf, 27
Calc_The_step, 33

calc_graph_azi, 31
calc_graph_polar, 31
calc_graph_radius, 32
calc_nth, 32
calc_phi_step, 33
CalcN_Theta, 33
check_contrast, 33
clear_hemi_grid, 33
convert_angles_ttpa2polar, 33
convert_angles_ttpd2polar, 34
crop_strip, 34
display_hemi_scan, 34
display_polar_graph, 35
display_scanlines, 36
draw_diffraction_cone, 36
draw_hemi_axes, 37
duplicate_hemi_scan, 37
find_hemi_data, 38
get_hemi_nickname, 38
get_hemi_prefix, 39
hemi_add_anglescan, 39
hemi_add_aziscan, 39
hemi_azi_cut, 39
hemi_polar_cut, 40
import_tpi_scan, 40
interpolate_hemi_scan, 41
kProjArea, 49
kProjDist, 49
kProjGnom, 49
kProjOrtho, 49
kProjScaleArea, 49
kProjScaleDist, 49
kProjScaleGnom, 49
kProjScaleOrtho, 50
kProjScaleStereo, 50
kProjStereo, 50
line_average, 41
load_hemi_scan, 41
make_hemi_grid, 41
normalize_strip_2d, 42
normalize_strip_phi, 42
normalize_strip_theta, 43
normalize_strip_x, 43
pizza_service, 44
pizza_service_2, 45
polar_graph_hook, 46
quick_pizza_image, 46
rotate_hemi_scan, 46
save_hemi_scan, 47
set_contrast, 47
set_polar_graph_cursor, 47
show_analyser_line, 47
strip_delete_frames, 48
trim_hemi_scan, 48
update_polar_info, 49
pearl-anglescan-tracker.ipf, 50
add_image_data, 52
AfterCompiledHook, 52
ast_add_image, 52
ast_callback_data, 52
ast_callback_detector, 52
ast_callback_manip, 53
ast_close, 53
ast_export, 53
ast_import, 53
ast_prepare, 53
ast_set_processing, 54
ast_setup, 54
ast_update_detector, 54
ast_window_hook, 54
bp_capture, 54
edit_offsets, 55
edit_reduction_params, 55
epics_connect, 55
epics_disconnect, 55
epics_disconnect_chid, 55
export_tracker_data, 55
extend_data, 55
IgorQuitHook, 55
import_tracker_data, 56
init_package, 56
load_prefs, 56
load_tracker_data, 56
package_path, 58
pmp_data, 56
pmp_data_mouseup, 56
pmp_parameters, 56
pmp_parameters_mouseup, 56
prefs_objects, 58
process_image_data, 56
save_prefs, 56
save_tracker_data, 57
setup_data, 57
setup_detector, 57
setup_graph, 57
toggle_capture, 57
update_capture, 57
update_data_graph, 57
update_detector, 57
update_detector_graph, 58
version, 58
pearl-area-display.ipf, 58
ad_box_filter, 60
ad_brick_slicer, 61
ad_calc_cursor_profiles, 61
ad_calc_histogram, 61
ad_calc_profiles, 61
ad_default_image_filter, 62
ad_display, 62
ad_display_brick, 62
ad_display_histogram, 63
ad_display_profiles, 63
ad_display_slice, 63
ad_export_profile, 64
ad_gizmo_set_plane, 64
ad_profiles_crosshairs, 65

ad_profiles_cursor_mode, 65
ad_profiles_hook, 65
ad_profiles_set_cursor, 65
ad_profiles_set_slice, 66
ad_slicer_init_bg, 66
ad_slicer_move_bg, 66
ad_slicer_start_bg, 66
ad_slicer_stop_bg, 67
ad_transpose_filter, 67
ad_update_profiles, 67
bp_extract_slice, 67
bp_move_slice, 67
bp_move_slice_center, 68
bp_reset_cursors, 68
get_source_image, 68
get_view_folder, 68
get_view_image, 68
graphname_from_dfref, 68
make_view_folder, 69
pmp_export, 69
set_trace_colors, 69
slp_slice_position, 69
svp_slice_position, 69
svp_smoothing, 69
update_slice_info, 69
pearl-area-import.ipf, 69
ad_load_dialog, 71
ad_suggest_foldername, 71
adh5_default_reduction, 71
adh5_list_reduction_funcs, 72
adh5_load_complete, 72
adh5_load_detector, 72
adh5_load_detector_image, 73
adh5_load_detector_slabs, 73
adh5_load_info, 73
adh5_load_preview, 74
adh5_load_reduced, 74
adh5_load_reduced_detector, 75
adh5_loadattr_all, 75
adh5_redim, 76
adh5_scale, 76
adh5_scale_scan, 76
adh5_scale_scienta, 76
adh5_setup_profile, 76
adh5_test_reduction_func, 77
BeforeFileOpenHook, 77
GetAttrDataFolderDFR, 77
read_attribute_info, 77
reduce_slab_image, 77
reduce_slab_worker, 78
pearl-area-profiles.ipf, 78
ad_collect_multiscan_y, 79
ad_extract_rod, 79
ad_extract_rod_x, 80
ad_extract_rod_y, 80
ad_extract_rod_z, 80
ad_extract_slab, 80
ad_extract_slab_x, 80
ad_extract_slab_y, 81
ad_extract_slab_z, 81
ad_profile_x, 81
ad_profile_x_w, 81
ad_profile_y, 81
ad_profile_y_w, 81
calc_y_profile_mins, 81
pearl-arpes.ipf, 81
AfterCompiledHook, 82
UnloadPearlArpesPackage, 82
pearl-data-explorer.ipf, 83
attributes_notebook, 85
bp_attr_notebook, 85
bp_browse_filepath, 85
bp_dataset_display, 85
bp_dataset_folder, 85
bp_dataset_next, 85
bp_dataset_prev, 85
bp_file_next, 85
bp_file_prev, 85
bp_load_files, 85
bp_load_files_opt, 85
bp_load_prefs, 85
bp_save_prefs, 85
bp_update_datasets, 86
bp_update_filelist, 86
display_dataset, 86
display_preview_trace, 86
extract_preview_image, 86
init_package, 86
itx_suggest_foldername, 86
ks_filematch_adh5, 90
ks_filematch_itx, 90
ks_filematch_mtrx, 90
ks_filematch_pshell, 90
lbp_datasets, 86
lbp_filelist, 86
load_file, 86
load_hdf_file, 86
load_itx_file, 87
load_mtrx_file, 87
load_prefs, 87
load_pshell_file, 87
load_selected_files, 87
notebook_add_attributes, 87
package_name, 90
package_path, 90
pearl_data_explorer, 87
pearl_file_type, 87
PearlDataExplorer, 87
preview_attributes, 88
preview_datafolder, 88
preview_dataset, 88
preview_file, 88
preview_hdf_file, 88
preview_itx_file, 88
preview_mtrx_file, 88
preview_pshell_file, 89

preview_setscale_x, 89
prompt_default_process, 89
prompt_func_params, 89
prompt_hdf_options, 89
save_prefs, 89
show_preview_graph, 90
test_attributes_notebook, 90
update_datasets, 90
update_filelist, 90
pearl-elog.ipf, 91
AfterFileOpenHook, 94
bp_attach, 94
bp_attach_allnone, 94
bp_attach_top, 94
bp_attach_updown, 94
bp_clear, 94
bp_login, 95
bp_logout, 95
bp_save_graphs, 95
bp_submit, 95
cleanup_temp_files, 95
create_cmd_file, 95
create_graph_file, 95
create_message_file, 96
elog_add_attachment, 96
elog_config, 97
elog_create_entry, 97
elog_create_logbook, 97
elog_init_pearl_templates, 97
elog_login, 98
elog_logout, 98
elog_panel_hook, 98
elog_parse_id, 104
elog_prompt_logbook, 98
elog_prompt_login, 98
elog_success_msg, 104
elog_validate_attributes, 99
format_url, 99
get_default_panel_name, 99
get_elog_df, 99
get_log_path, 99
get_panel_attributes, 100
get_panel_graphs, 100
get_panel_message, 100
get_timestamp, 100
IgorBeforeNewHook, 100
IgorQuitHook, 101
init_package, 101
init_volatile_vars, 101
kAttachColName, 104
kAttachColSel, 104
kAttachColTitle, 104
kdfPersistent, 104
kdfRoot, 104
kdfTemplates, 105
kdfVolatile, 105
list_logbooks, 101
load_prefs, 101
move_attach_item, 102
package_name, 105
package_path, 105
parse_result, 102
pearl_elog, 102
PearlElogPanel, 102
prepare_command_line, 102
prepare_graph_attachments, 103
save_prefs, 103
set_panel_attributes, 103
set_panel_graphs, 103
set_panel_message, 104
update_attach_items, 104
pearl-fitfuncs.ipf, 105
Au4f, 106
Au4f_2p2, 106
Au4f_2p3, 106
Calc_DoniachSunjicBroad, 106
DoniachSunjicBroad, 106
DoniachSunjicBroadS, 106
ds1_bg, 106
ds2_bg, 106
ds4_bg, 106
ds6_bg, 106
FermiGaussConv, 106
Fit_DoniachSunjicBroad, 107
ShowComponents_Au4f_2p2, 107
ShowComponents_Au4f_2p3, 107
version, 107
pearl-gui-tools.ipf, 107
display_progress_panel, 107
kill_progress_panel, 107
update_progress_panel, 107
pearl-matrix-import.ipf, 108
AfterFileOpenHook, 109
BeforeFileOpenHook, 109
check_package_folder, 109
init_package, 110
initStruct, 110
ks_filematch_mtrx, 115
matrix_format_elog_message, 110
matrix_preview_2d, 110
mtrx_create_folder, 110
mtrx_file_brickletID, 110
mtrx_get_cycle_folder, 111
mtrx_load_all, 111
mtrx_load_file, 111
mtrx_load_info, 112
mtrx_load_preview, 112
mtrx_open_file, 113
mtrx_parse_filename, 113
mtrx_scale_dataset, 114
mtrx_split_filename, 114
package_name, 115
package_path, 115
preview_matrix_file, 114
subtract_line_bg, 115
pearl-menu.ipf, 115

Display2dProfiles, 115
Display3dSlicer, 115
DisplayGizmoSlicer, 115
LoadPearlArpes, 116
LoadPearlOptics, 116
LoadPearlPreparation, 116
PearlAnglescanTracker, 116
PearlLiveDisplay, 116
PearlMenuEnableFunc, 116
pearl-otf-import.ipf, 116
gather_batch, 116
otf_gather_batch, 116
otf_gather_iterator, 117
otf_interp, 117
otf_load_itx, 117
otf_load_itx_all, 117
otf_load_itx_match, 117
otf_rename_folders, 117
otf_rename_folders_iterator, 117
otf_smo_int, 117
pearl-polar-coordinates.ipf, 117
cart2polar, 118
cart2polar_wave, 118
polar2cart, 118
polar2cart_wave, 118
polar_distance, 118
pearl-pshell-import.ipf, 118
find_attr_folder, 121
find_scale_wave, 121
find_scan_folder, 121
kAngleDimLabel, 136
kDataDimLabel, 136
kDetectorSensitivity, 136
kEnergyDimLabel, 136
kPreviewDatasets, 136
kScanDimLabel, 136
kScientaScalingDatasets, 136
kTransposedDatasets, 136
ps_detect_scale, 121
ps_scale_dataset, 122
ps_scale_dataset_2, 122
ps_scale_datasets, 123
ps_set_dimlabels, 123
ps_set_dimlabels2, 124
psh5_close_file, 124
psh5_list_scan_datasets, 124
psh5_list_scan_regions, 125
psh5_list_scans, 125
psh5_load_complete, 125
psh5_load_dataset, 126
psh5_load_dataset_meta, 127
psh5_load_dataset_reduced, 127
psh5_load_dataset_slab, 128
psh5_load_dataset_slabs, 128
psh5_load_info, 129
psh5_load_preview, 129
psh5_load_reduced, 130
psh5_load_scan_attrs, 131
psh5_load_scan_complete, 131
psh5_load_scan_data, 132
psh5_load_scan_info, 132
psh5_load_scan_meta, 133
psh5_load_scan_preview, 133
psh5_load_scan_section, 134
psh5_open_file, 134
reduce_slab_image, 135
reduce_slab_worker, 135
select_dataset, 135
twave2list, 135
wave2list, 135
pearl-scienta-preprocess.ipf, 137
capture_int_linbg_cursors, 138
csr_int_linbg_reduction, 138
fit_scienta_ang_transm, 138
fit_scienta_poly_bg, 138
gauss2_reduction, 138
int_linbg_reduction, 139
int_quadbg_reduction, 139
prompt_Shockley_anglefit, 139
prompt_gauss2_reduction, 139
prompt_int_linbg_reduction, 139
prompt_int_quadbg_reduction, 139
prompt_redim_linbg_reduction, 139
redim_linbg_reduction, 139
scienta_ang_transm, 140
scienta_norm, 140
scienta_poly_bg, 140
Shockley_anglefit, 140
test_gauss2_reduction, 140
test_int_linbg, 141
test_int_quadbg, 141
test_shockley_anglefit, 141
pearl-tools.ipf, 141
AppendToGraphIterator, 141
DefaultFolderIterator, 141
DefaultWaveIterator, 141
IterateDataFolders, 141
IterateWaves, 142
SumWavesIterator, 142
pearl-vector-operations.ipf, 142
create_rotation_matrix_free, 142
rotate2d_x, 142
rotate2d_y, 142
rotate_x_wave, 142
rotate_y_wave, 142
rotate_z_wave, 142
set_rotation_x, 143
set_rotation_y, 143
set_rotation_z, 143
pearl_data_explorer
pearl-data-explorer.ipf, 87
pearl_elog
pearl-elog.ipf, 102
pearl_file_type
pearl-data-explorer.ipf, 87
PearlAnglescanProcess, 17

PearlAnglescanTracker
pearl-menu.ipf, 116
PearlAreaDisplay, 17
PearlAreaImport, 18
PearlAreaProfiles, 18
PearlArpes, 18
PearlDataExplorer, 18
pearl-data-explorer.ipf, 87
PearlElog, 19
PearlElogPanel
pearl-elog.ipf, 102
PearlLiveDisplay
pearl-menu.ipf, 116
PearlMatrixImport, 19
PearlMenuEnableFunc
pearl-menu.ipf, 116
PearlPShellImport, 19
PearlScientaPreprocess, 19
pizza_service
pearl-anglescan-process.ipf, 44
pizza_service_2
pearl-anglescan-process.ipf, 45
pmp_data
pearl-anglescan-tracker.ipf, 56
pmp_data_mouseup
pearl-anglescan-tracker.ipf, 56
pmp_export
pearl-area-display.ipf, 69
pmp_parameters
pearl-anglescan-tracker.ipf, 56
pmp_parameters_mouseup
pearl-anglescan-tracker.ipf, 56
polar2cart
pearl-polar-coordinates.ipf, 118
polar2cart_wave
pearl-polar-coordinates.ipf, 118
polar_distance
pearl-polar-coordinates.ipf, 118
polar_graph_hook
pearl-anglescan-process.ipf, 46
precision
DoniachSunjicStruct, 21
prefs_objects
pearl-anglescan-tracker.ipf, 58
prepare_command_line
pearl-elog.ipf, 102
prepare_graph_attachments
pearl-elog.ipf, 103
preview_attributes
pearl-data-explorer.ipf, 88
preview_datafolder
pearl-data-explorer.ipf, 88
preview_dataset
pearl-data-explorer.ipf, 88
preview_file
pearl-data-explorer.ipf, 88
preview_hdf_file
pearl-data-explorer.ipf, 88
preview_itx_file
pearl-data-explorer.ipf, 88
preview_matrix_file
pearl-matrix-import.ipf, 114
preview_mtrx_file
pearl-data-explorer.ipf, 88
preview_pshell_file
pearl-data-explorer.ipf, 89
preview_setscale_x
pearl-data-explorer.ipf, 89
process_image_data
pearl-anglescan-tracker.ipf, 56
prompt_Shockley_anglefit
pearl-scienta-preprocess.ipf, 139
prompt_default_process
pearl-data-explorer.ipf, 89
prompt_func_params
pearl-data-explorer.ipf, 89
prompt_gauss2_reduction
pearl-scienta-preprocess.ipf, 139
prompt_hdf_options
pearl-data-explorer.ipf, 89
prompt_int_linbg_reduction
pearl-scienta-preprocess.ipf, 139
prompt_int_quadbg_reduction
pearl-scienta-preprocess.ipf, 139
prompt_redim_linbg_reduction
pearl-scienta-preprocess.ipf, 139
ps_detect_scale
pearl-pshell-import.ipf, 121
ps_scale_dataset
pearl-pshell-import.ipf, 122
ps_scale_dataset_2
pearl-pshell-import.ipf, 122
ps_scale_datasets
pearl-pshell-import.ipf, 123
ps_set_dimlabels
pearl-pshell-import.ipf, 123
ps_set_dimlabels2
pearl-pshell-import.ipf, 124
psh5_close_file
pearl-pshell-import.ipf, 124
psh5_list_scan_datasets
pearl-pshell-import.ipf, 124
psh5_list_scan_regions
pearl-pshell-import.ipf, 125
psh5_list_scans
pearl-pshell-import.ipf, 125
psh5_load_complete
pearl-pshell-import.ipf, 125
psh5_load_dataset
pearl-pshell-import.ipf, 126
psh5_load_dataset_meta
pearl-pshell-import.ipf, 127
psh5_load_dataset_reduced
pearl-pshell-import.ipf, 127
psh5_load_dataset_slab
pearl-pshell-import.ipf, 128

psh5_load_dataset_slabs
pearl-pshell-import.ipf, 128
psh5_load_info
pearl-pshell-import.ipf, 129
psh5_load_preview
pearl-pshell-import.ipf, 129
psh5_load_reduced
pearl-pshell-import.ipf, 130
psh5_load_scan_attrs
pearl-pshell-import.ipf, 131
psh5_load_scan_complete
pearl-pshell-import.ipf, 131
psh5_load_scan_data
pearl-pshell-import.ipf, 132
psh5_load_scan_info
pearl-pshell-import.ipf, 132
psh5_load_scan_meta
pearl-pshell-import.ipf, 133
psh5_load_scan_preview
pearl-pshell-import.ipf, 133
psh5_load_scan_section
pearl-pshell-import.ipf, 134
psh5_open_file
pearl-pshell-import.ipf, 134
pw
DoniachSunjicStruct, 22

quick_pizza_image
pearl-anglescan-process.ipf, 46

read_attribute_info
pearl-area-import.ipf, 77
record_results
fermi-edge-analysis.ipf, 26
redim_linbg_reduction
pearl-scienta-preprocess.ipf, 139
reduce_slab_image
pearl-area-import.ipf, 77
pearl-pshell-import.ipf, 135
reduce_slab_worker
pearl-area-import.ipf, 78
pearl-pshell-import.ipf, 135
rotate2d_x
pearl-vector-operations.ipf, 142
rotate2d_y
pearl-vector-operations.ipf, 142
rotate_hemi_scan
pearl-anglescan-process.ipf, 46
rotate_x_wave
pearl-vector-operations.ipf, 142
rotate_y_wave
pearl-vector-operations.ipf, 142
rotate_z_wave
pearl-vector-operations.ipf, 142

SUCCESS
errorCode, 23
save_hemi_scan
pearl-anglescan-process.ipf, 47
save_prefs
pearl-anglescan-tracker.ipf, 56
pearl-data-explorer.ipf, 89
pearl-elog.ipf, 103
save_tracker_data
pearl-anglescan-tracker.ipf, 57
scienta_ang_transm
pearl-scienta-preprocess.ipf, 140
scienta_norm
pearl-scienta-preprocess.ipf, 140
scienta_poly_bg
pearl-scienta-preprocess.ipf, 140
select_dataset
pearl-pshell-import.ipf, 135
set_contrast
pearl-anglescan-process.ipf, 47
set_panel_attributes
pearl-elog.ipf, 103
set_panel_graphs
pearl-elog.ipf, 103
set_panel_message
pearl-elog.ipf, 104
set_polar_graph_cursor
pearl-anglescan-process.ipf, 47
set_rotation_x
pearl-vector-operations.ipf, 143
set_rotation_y
pearl-vector-operations.ipf, 143
set_rotation_z
pearl-vector-operations.ipf, 143
set_trace_colors
pearl-area-display.ipf, 69
setup_data
pearl-anglescan-tracker.ipf, 57
setup_detector
pearl-anglescan-tracker.ipf, 57
setup_graph
pearl-anglescan-tracker.ipf, 57
Shockley_anglefit
pearl-scienta-preprocess.ipf, 140
show_analyser_line
pearl-anglescan-process.ipf, 47
show_preview_graph
pearl-data-explorer.ipf, 90
show_shift
fermi-edge-analysis.ipf, 26
ShowComponents_Au4f_2p2
pearl-fitfuncs.ipf, 107
ShowComponents_Au4f_2p3
pearl-fitfuncs.ipf, 107
slit_correction
fermi-edge-analysis.ipf, 27
slit_shift
fermi-edge-analysis.ipf, 27
slp_slice_position
pearl-area-display.ipf, 69
strip_delete_frames
pearl-anglescan-process.ipf, 48

subtract_line_bg
pearl-matrix-import.ipf, 115
SumWavesIterator
pearl-tools.ipf, 142
svp_slice_position
pearl-area-display.ipf, 69
svp_smoothing
pearl-area-display.ipf, 69

test_attributes_notebook
pearl-data-explorer.ipf, 90
test_gauss2_reduction
pearl-scienta-preprocess.ipf, 140
test_int_linbg
pearl-scienta-preprocess.ipf, 141
test_int_quadbg
pearl-scienta-preprocess.ipf, 141
test_shockley_anglefit
pearl-scienta-preprocess.ipf, 141
toggle_capture
pearl-anglescan-tracker.ipf, 57
trim_hemi_scan
pearl-anglescan-process.ipf, 48
twave2list
pearl-pshell-import.ipf, 135

UNKNOWN_ERROR
errorCode, 23
UnloadPearlArpesPackage
pearl-arpes.ipf, 82
update_attach_items
pearl-elog.ipf, 104
update_capture
pearl-anglescan-tracker.ipf, 57
update_data_graph
pearl-anglescan-tracker.ipf, 57
update_datasets
pearl-data-explorer.ipf, 90
update_detector
pearl-anglescan-tracker.ipf, 57
update_detector_graph
pearl-anglescan-tracker.ipf, 58
update_filelist
pearl-data-explorer.ipf, 90
update_polar_info
pearl-anglescan-process.ipf, 49
update_progress_panel
pearl-gui-tools.ipf, 107
update_slice_info
pearl-area-display.ipf, 69

version
pearl-anglescan-tracker.ipf, 58
pearl-fitfuncs.ipf, 107

WAVE_EXIST
errorCode, 23
WRONG_PARAMETER
errorCode, 23

wave2list
pearl-pshell-import.ipf, 135

xdw
DoniachSunjicStruct, 22
xw
DoniachSunjicStruct, 22

yw
DoniachSunjicStruct, 22