

**dEdxRecon**

Generated by Doxygen 1.15.0



---

<b>1 Directory Hierarchy</b>	<b>1</b>
1.1 Directories . . . . .	1
<b>2 Namespace Index</b>	<b>5</b>
2.1 Namespace List . . . . .	5
<b>3 Hierarchical Index</b>	<b>7</b>
3.1 Class Hierarchy . . . . .	7
<b>4 Class Index</b>	<b>9</b>
4.1 Class List . . . . .	9
<b>5 File Index</b>	<b>11</b>
5.1 File List . . . . .	11
<b>6 Directory Documentation</b>	<b>15</b>
6.1 Apps Directory Reference . . . . .	15
6.2 Fitters Directory Reference . . . . .	15
6.3 Fitters/inc Directory Reference . . . . .	15
6.4 ModelEvent/inc Directory Reference . . . . .	16
6.5 ModelSignal/inc Directory Reference . . . . .	16
6.6 Reconstruction/inc Directory Reference . . . . .	17
6.7 Uploader/inc Directory Reference . . . . .	17
6.8 Utilities/inc Directory Reference . . . . .	18
6.9 ModelEvent Directory Reference . . . . .	18
6.10 ModelSignal Directory Reference . . . . .	18
6.11 Reconstruction Directory Reference . . . . .	18
6.12 Fitters/src Directory Reference . . . . .	18
6.13 ModelEvent/src Directory Reference . . . . .	19
6.14 ModelSignal/src Directory Reference . . . . .	19
6.15 Reconstruction/src Directory Reference . . . . .	19
6.16 Uploader/src Directory Reference . . . . .	19
6.17 Utilities/src Directory Reference . . . . .	20
6.18 Uploader Directory Reference . . . . .	20
6.19 Utilities Directory Reference . . . . .	20
<b>7 Namespace Documentation</b>	<b>21</b>
7.1 Reconstruction Namespace Reference . . . . .	21
7.1.1 Detailed Description . . . . .	23
7.1.2 * @file dEdx.h . . . . .	23
7.1.3 * . . . . .	23
7.1.4 * Contains declarations for data structures, helper classes and functions . . . . .	23
7.1.5 * used to compute energy loss estimates used by the reconstruction pipeline. . . . .	23
7.1.6 * . . . . .	23

7.1.7 *	23
7.1.8 * The implementation (dEdx.cxx) contains the core numerical routines and	23
7.1.9 * algorithms used to compute energy loss per unit length from reconstructed	23
7.1.10 * cluster and pad information, together with helper utilities for particle	23
7.1.11 * identification. Interfaces for computing WF and XP estimates, GigaWaveform	23
7.1.12 * helpers and truncated estimators are declared here; heavy numerical work	23
7.1.13 * and algorithmic details live in the source file.	23
7.1.14 autotoc_md13	23
7.1.15 Function Documentation	23
7.1.15.1 ClearVectors()	23
7.1.15.2 Correction()	24
7.1.15.3 DefaultAnalysis()	24
7.1.15.4 DrawCERN22Scan()	24
7.1.15.5 DrawMultipleScan()	24
7.1.15.6 DrawSingleScan()	24
7.1.15.7 Monitoring()	24
7.1.15.8 Settings()	24
7.1.15.9 WFCorrection()	25
7.1.16 Variable Documentation	25
7.1.16.1 CERN_drift	25
7.1.16.2 CERN_Escan	25
7.1.16.3 comment	25
7.1.16.4 correction_wf	25
7.1.16.5 corrFuncPath	25
7.1.16.6 dataPath	25
7.1.16.7 dataScanPath	25
7.1.16.8 dedx	25
7.1.16.9 dEdxPath	25
7.1.16.10 DESY_drift	26
7.1.16.11 DESY_phi	26
7.1.16.12 DESY_row	26
7.1.16.13 DESY_theta	26
7.1.16.14 DESY_yscan	26
7.1.16.15 DESY_zscan	26
7.1.16.16 DO_Comparison	26
7.1.16.17 DO_Control	26
7.1.16.18 DO_dEdx	26
7.1.16.19 Draw_CERN22Scan	26
7.1.16.20 Draw_Comparison	27
7.1.16.21 Draw_Control	27
7.1.16.22 Draw_Corrections	27
7.1.16.23 Draw_dEdx	27

---

7.1.16.24 Draw_DESY21MultScan . . . . .	27
7.1.16.25 Draw_DESY21SingleScan . . . . .	27
7.1.16.26 drawout_file . . . . .	27
7.1.16.27 drawoutMultiScanPath . . . . .	27
7.1.16.28 drawoutPath . . . . .	27
7.1.16.29 drawoutRunPath . . . . .	27
7.1.16.30 drawoutScanPath . . . . .	28
7.1.16.31 drawWhichMethods . . . . .	28
7.1.16.32 driftDist . . . . .	28
7.1.16.33 Dt . . . . .	28
7.1.16.34 DtB . . . . .	28
7.1.16.35 DtnoB . . . . .	28
7.1.16.36 DtwithBhere . . . . .	28
7.1.16.37 DtwithoutBhere . . . . .	28
7.1.16.38 fcorrectDrift . . . . .	28
7.1.16.39 fcorrectGain . . . . .	28
7.1.16.40 fcorrectRC . . . . .	29
7.1.16.41 fcorrectWF . . . . .	29
7.1.16.42 fsaveSelectOnly . . . . .	29
7.1.16.43 inputDir . . . . .	29
7.1.16.44 intUploader . . . . .	29
7.1.16.45 log_file . . . . .	29
7.1.16.46 moduleCase . . . . .	29
7.1.16.47 multiScanName . . . . .	29
7.1.16.48 p_dEdx . . . . .	29
7.1.16.49 p_DrawOuts . . . . .	29
7.1.16.50 p_lut . . . . .	30
7.1.16.51 p_uploader . . . . .	30
7.1.16.52 prototype . . . . .	30
7.1.16.53 PT . . . . .	30
7.1.16.54 pUploader . . . . .	30
7.1.16.55 rootout_file . . . . .	30
7.1.16.56 runvarstr . . . . .	30
7.1.16.57 scanIndex . . . . .	30
7.1.16.58 scanName . . . . .	30
7.1.16.59 selectionSet . . . . .	30
7.1.16.60 tag . . . . .	31
7.1.16.61 TB . . . . .	31
7.1.16.62 testbeam . . . . .	31
7.1.16.63 v_comments . . . . .	31
7.1.16.64 v_datafiles . . . . .	31
7.1.16.65 v_rootout_files . . . . .	31

---

7.1.16.66 v_scanspec . . . . .	31
7.1.16.67 vcorrFuncPaths . . . . .	31
7.1.16.68 vScanLabels . . . . .	31
7.1.16.69 vScanVals . . . . .	31
7.1.16.70 vTags . . . . .	31
<b>8 Class Documentation</b>	<b>33</b>
8.1 Cluster Class Reference . . . . .	33
8.1.1 Constructor & Destructor Documentation . . . . .	34
8.1.1.1 Cluster() . . . . .	34
8.1.1.2 ~Cluster() . . . . .	35
8.1.2 Member Function Documentation . . . . .	35
8.1.2.1 Add_Pad() . . . . .	35
8.1.2.2 Chi2_Diagonal() . . . . .	35
8.1.2.3 Chi2_Horizontal() . . . . .	35
8.1.2.4 DoClosure() . . . . .	35
8.1.2.5 Eval_Diagonal() . . . . .	35
8.1.2.6 Eval_Horizontal() . . . . .	35
8.1.2.7 FitRes_Diagonal_Get_NberOfTermsInChi2() . . . . .	35
8.1.2.8 FitRes_Diagonal_Get_Pad() . . . . .	36
8.1.2.9 FitRes_Diagonal_Get_Pull() . . . . .	36
8.1.2.10 FitRes_Diagonal_Get_Residual() . . . . .	36
8.1.2.11 FitRes_Horizontal_Get_NberOfTermsInChi2() . . . . .	36
8.1.2.12 FitRes_Horizontal_Get_Pad() . . . . .	36
8.1.2.13 FitRes_Horizontal_Get_Pull() . . . . .	36
8.1.2.14 FitRes_Horizontal_Get_Residual() . . . . .	36
8.1.2.15 Get_Acluster() . . . . .	36
8.1.2.16 Get_AMaxLeading() . . . . .	36
8.1.2.17 Get_Chi2Min() . . . . .	37
8.1.2.18 Get_EntryNber() . . . . .	37
8.1.2.19 Get_EventNber() . . . . .	37
8.1.2.20 Get_eXTrack() . . . . .	37
8.1.2.21 Get_eYTrack() . . . . .	37
8.1.2.22 Get_LeadingPad() . . . . .	37
8.1.2.23 Get_ModuleNber() . . . . .	37
8.1.2.24 Get_NberOfPads() . . . . .	37
8.1.2.25 Get_NextLeadingPad() . . . . .	37
8.1.2.26 Get_NextNextLeadingPad() . . . . .	37
8.1.2.27 Get_Pad() . . . . .	38
8.1.2.28 Get_TMaxLeading() . . . . .	38
8.1.2.29 Get_XTrack() . . . . .	38
8.1.2.30 Get_XTrack_BeforeMinimisation() . . . . .	38

---

8.1.2.31 Get_XWeight()	38
8.1.2.32 Get_YLeading()	38
8.1.2.33 Get_YTrack()	38
8.1.2.34 Get_YTrack_BeforeMinimisation()	38
8.1.2.35 Get_YWeight()	38
8.1.2.36 Invalidate()	38
8.1.2.37 IsValid()	39
8.1.2.38 Set_eXTrack()	39
8.1.2.39 Set_eYTrack()	39
8.1.2.40 Set_XTrack()	39
8.1.2.41 Set_YTrack()	39
8.1.2.42 SetEval_Diagonal()	39
8.1.2.43 SetEval_Horizontal()	39
8.1.2.44 SetParameter_Diagonal()	39
8.1.2.45 SetParameter_Horizontal()	39
8.1.2.46 SetResults_Diagonal()	40
8.1.2.47 SetResults_FailedFit_Diagonal()	40
8.1.2.48 SetResults_FailedFit_Horizontal()	40
8.1.2.49 SetResults_Horizontal()	40
8.1.2.50 StatusFit()	40
8.1.2.51 StatusFit_Diagonal()	40
8.1.2.52 StatusFit_Horizontal()	40
8.1.2.53 Validate()	40
8.1.2.54 WriteOut()	40
8.1.3 Member Data Documentation	41
8.1.3.1 m_AngleRot	41
8.2 ClusterFitter_Diagonal Class Reference	41
8.2.1 Constructor & Destructor Documentation	41
8.2.1.1 ClusterFitter_Diagonal()	41
8.2.1.2 ~ClusterFitter_Diagonal()	42
8.2.2 Member Function Documentation	42
8.2.2.1 Chi2()	42
8.2.2.2 DoMinimisation()	42
8.2.2.3 Set_Cluster()	42
8.2.3 Member Data Documentation	43
8.2.3.1 m_FitterName	43
8.2.3.2 p_Cluster	43
8.2.3.3 p_TVirtualFitter	43
8.3 ClusterFitter_Horizontal Class Reference	43
8.3.1 Constructor & Destructor Documentation	44
8.3.1.1 ClusterFitter_Horizontal()	44
8.3.1.2 ~ClusterFitter_Horizontal()	44

---

8.3.2 Member Function Documentation . . . . .	44
8.3.2.1 Chi2() . . . . .	44
8.3.2.2 DoMinimisation() . . . . .	45
8.3.2.3 Set_Cluster() . . . . .	45
8.3.3 Member Data Documentation . . . . .	45
8.3.3.1 m_FitterName . . . . .	45
8.3.3.2 p_Cluster . . . . .	45
8.3.3.3 p_TVirtualFitter . . . . .	45
8.4 Reconstruction::dEdx Class Reference . . . . .	45
8.4.1 Constructor & Destructor Documentation . . . . .	46
8.4.1.1 dEdx() . . . . .	46
8.4.1.2 ~dEdx() . . . . .	46
8.4.2 Member Function Documentation . . . . .	46
8.4.2.1 ComputedEdxGP() . . . . .	46
8.4.2.2 ComputedEdxGP1() . . . . .	46
8.4.2.3 ComputedEdxWF() . . . . .	47
8.4.2.4 ComputedEdxXP() . . . . .	47
8.4.2.5 DiscardedModule() . . . . .	47
8.4.2.6 GetGigaWaveform() . . . . .	47
8.4.2.7 GetTruncatedGigaWaveformGP1() . . . . .	47
8.4.2.8 Reconstruction() . . . . .	47
8.5 Reconstruction::DrawOuts Class Reference . . . . .	47
8.5.1 Detailed Description . . . . .	48
8.5.2 Constructor & Destructor Documentation . . . . .	48
8.5.2.1 DrawOuts() [1/3] . . . . .	48
8.5.2.2 DrawOuts() [2/3] . . . . .	49
8.5.2.3 DrawOuts() [3/3] . . . . .	50
8.5.2.4 ~DrawOuts() . . . . .	50
8.5.3 Member Function Documentation . . . . .	50
8.5.3.1 AmplitudeVSLength() . . . . .	50
8.5.3.2 CERN22Scan() . . . . .	50
8.5.3.3 CERN22ScanDraw() . . . . .	50
8.5.3.4 CERN22ScanFill() . . . . .	50
8.5.3.5 Control() . . . . .	51
8.5.3.6 DESY21MultiScan() . . . . .	51
8.5.3.7 DESY21ScanDraw() . . . . .	51
8.5.3.8 DESY21ScanFill() . . . . .	51
8.5.3.9 DESY21SingleScan() . . . . .	51
8.5.3.10 EnergyLoss() . . . . .	51
8.5.3.11 FileComparison() . . . . .	52
8.5.3.12 SetStyle() . . . . .	52
8.6 EramInfo Class Reference . . . . .	52

---

8.6.1 Detailed Description . . . . .	53
8.6.2 * @file LUTs.h . . . . .	53
8.6.3 * . . . . .	53
8.6.4 * Declares data structures and helpers for building and accessing LUTs . . . . .	53
8.6.5 * (geometry, calibration, etc.) used by dE/dx and track/cluster . . . . .	53
8.6.6 * reconstruction. . . . .	53
8.6.7 * . . . . .	53
8.6.8 * . . . . .	53
8.6.9 * The implementation file LUTs.cxx constructs ERAM maps, fills lookup tables . . . . .	53
8.6.10 * and exposes accessors used across the reconstruction pipeline. The source . . . . .	53
8.6.11 * also contains helpers to fill and repair maps (FillHoles), and code to . . . . .	53
8.6.12 * compute mean gains/RC values used in calibration steps. . . . .	53
8.6.13 autotoc_md26 . . . . .	53
8.6.14 Constructor & Destructor Documentation . . . . .	53
8.6.14.1 EramInfo() . . . . .	53
8.6.14.2 ~EramInfo() . . . . .	53
8.6.15 Member Data Documentation . . . . .	53
8.6.15.1 Endplate . . . . .	53
8.6.15.2 Gain . . . . .	54
8.6.15.3 Id . . . . .	54
8.6.15.4 InbTPC . . . . .	54
8.6.15.5 IntTPC . . . . .	54
8.6.15.6 Position . . . . .	54
8.6.15.7 RC . . . . .	54
8.6.15.8 Resolution . . . . .	54
8.6.15.9 XX . . . . .	54
8.6.15.10 YY . . . . .	54
8.7 Reconstruction::ERAMMaps Class Reference . . . . .	55
8.7.1 Constructor & Destructor Documentation . . . . .	55
8.7.1.1 ERAMMaps() . . . . .	55
8.7.1.2 ~ERAMMaps() . . . . .	55
8.7.2 Member Function Documentation . . . . .	55
8.7.2.1 Gain() . . . . .	55
8.7.2.2 ID() . . . . .	55
8.7.2.3 MeanGain() . . . . .	55
8.7.2.4 MeanRC() . . . . .	56
8.7.2.5 RC() . . . . .	56
8.7.2.6 Resolution() . . . . .	56
8.8 Event Class Reference . . . . .	56
8.8.1 Constructor & Destructor Documentation . . . . .	57
8.8.1.1 Event() . . . . .	57
8.8.1.2 ~Event() . . . . .	57

8.8.2 Member Function Documentation . . . . .	57
8.8.2.1 Add_Module() . . . . .	57
8.8.2.2 Clear_Modules() . . . . .	57
8.8.2.3 Get_Cluster_Copy() . . . . .	57
8.8.2.4 Get_EntryNber() . . . . .	58
8.8.2.5 Get_EventNber() . . . . .	58
8.8.2.6 Get_Model_Charge() . . . . .	58
8.8.2.7 Get_Model_Electronics() . . . . .	58
8.8.2.8 Get_Model_ReadOutGeometry() . . . . .	58
8.8.2.9 Get_Module_InArray() . . . . .	58
8.8.2.10 Get_NberOfModule() . . . . .	58
8.8.2.11 Get_ThisModule() . . . . .	58
8.8.2.12 Get_ThisModuleConst() . . . . .	58
8.8.2.13 GiveMe_AnUnfittedTrack_ForThisModule() . . . . .	58
8.8.2.14 GiveMe_Clusters_ForThisModule() . . . . .	59
8.8.2.15 GiveMe_CutInterCeptXZ_ForThisModule() . . . . .	59
8.8.2.16 GiveMe_CutSlopeXY_ForThisModule() . . . . .	59
8.8.2.17 GiveMe_CutSlopeXYZ_ForThisModule() . . . . .	59
8.8.2.18 GiveMe_CutSlopeXZ_ForThisModule() . . . . .	59
8.8.2.19 GiveMe_Track_ForThisModule() . . . . .	59
8.8.2.20 Invalidate() . . . . .	59
8.8.2.21 Invalidate_ThisModule() . . . . .	59
8.8.2.22 IsValid() . . . . .	59
8.8.2.23 Replace_Clusters_ForThisModule() . . . . .	60
8.8.2.24 Set_Track_ForThisModule() . . . . .	60
8.8.2.25 SmallDump() . . . . .	60
8.8.2.26 Validate() . . . . .	60
8.8.2.27 Validate_ThisModule() . . . . .	60
8.8.2.28 Validity_ForThisModule() . . . . .	60
8.8.2.29 WriteOut() . . . . .	60
8.9 FitOutput Class Reference . . . . .	60
8.9.1 Constructor & Destructor Documentation . . . . .	61
8.9.1.1 FitOutput() [1/2] . . . . .	61
8.9.1.2 ~FitOutput() . . . . .	61
8.9.1.3 FitOutput() [2/2] . . . . .	61
8.9.2 Member Function Documentation . . . . .	62
8.9.2.1 operator=() . . . . .	62
8.9.2.2 PrintFitOutput() . . . . .	62
8.9.2.3 PrintFitOutputInCanvas() . . . . .	62
8.9.2.4 Reset() . . . . .	62
8.9.2.5 Set() . . . . .	62
8.9.2.6 SetResults() [1/2] . . . . .	62

---

8.9.2.7 SetResults() [2/2] . . . . .	62
8.9.3 Member Data Documentation . . . . .	62
8.9.3.1 m_Description . . . . .	62
8.9.3.2 m_MinnLL . . . . .	63
8.9.3.3 m_NameOfTheModel . . . . .	63
8.9.3.4 m_NberOfDataPoints . . . . .	63
8.9.3.5 m_NberOfModelParameters . . . . .	63
8.9.3.6 p_CovMatrix . . . . .	63
8.9.3.7 p_eparminus . . . . .	63
8.9.3.8 p_eparparab . . . . .	63
8.9.3.9 p_eparplus . . . . .	63
8.9.3.10 p_par . . . . .	64
8.9.3.11 p_parName . . . . .	64
8.10 GaussFunction Class Reference . . . . .	64
8.10.1 Constructor & Destructor Documentation . . . . .	64
8.10.1.1 GaussFunction() . . . . .	64
8.10.1.2 ~GaussFunction() . . . . .	64
8.10.2 Member Function Documentation . . . . .	65
8.10.2.1 Get_Mean() . . . . .	65
8.10.2.2 Get_Norm() . . . . .	65
8.10.2.3 Get_Sigma() . . . . .	65
8.10.2.4 operator()() . . . . .	65
8.10.2.5 Set_Mean() . . . . .	65
8.10.2.6 Set_Norm() . . . . .	65
8.10.2.7 Set_Sigma() . . . . .	65
8.11 GlobalChi2_4 Struct Reference . . . . .	66
8.11.1 Constructor & Destructor Documentation . . . . .	66
8.11.1.1 GlobalChi2_4() . . . . .	66
8.11.2 Member Function Documentation . . . . .	66
8.11.2.1 operator()() . . . . .	66
8.11.3 Member Data Documentation . . . . .	66
8.11.3.1 fChi2_1 . . . . .	66
8.11.3.2 fChi2_2 . . . . .	66
8.11.3.3 fChi2_3 . . . . .	66
8.11.3.4 fChi2_4 . . . . .	67
8.12 Reconstruction::LUT Class Reference . . . . .	67
8.12.1 Constructor & Destructor Documentation . . . . .	67
8.12.1.1 LUT() [1/2] . . . . .	67
8.12.1.2 LUT() [2/2] . . . . .	67
8.12.1.3 ~LUT() . . . . .	67
8.12.2 Member Function Documentation . . . . .	68
8.12.2.1 getRatio() . . . . .	68

---

8.13 Model_Charge0D Class Reference . . . . .	68
8.13.1 Constructor & Destructor Documentation . . . . .	69
8.13.1.1 Model_Charge0D() . . . . .	69
8.13.1.2 ~Model_Charge0D() . . . . .	69
8.13.2 Member Function Documentation . . . . .	69
8.13.2.1 Get_Qpad() . . . . .	69
8.13.2.2 Get_Qprim() . . . . .	70
8.13.2.3 Set_Ion() . . . . .	70
8.13.2.4 Set_Qprim() . . . . .	70
8.13.2.5 SetSecondaries() . . . . .	70
8.13.2.6 WriteOut() . . . . .	70
8.13.3 Member Data Documentation . . . . .	70
8.13.3.1 m_Dolons . . . . .	70
8.13.3.2 m_Qprim . . . . .	70
8.13.3.3 m_RFactor . . . . .	70
8.13.3.4 m_RTrue . . . . .	71
8.13.3.5 m_Tion . . . . .	71
8.14 Model_Charge1D Class Reference . . . . .	71
8.14.1 Constructor & Destructor Documentation . . . . .	72
8.14.1.1 Model_Charge1D() . . . . .	72
8.14.1.2 ~Model_Charge1D() . . . . .	72
8.14.2 Member Function Documentation . . . . .	72
8.14.2.1 Get_Lambda() . . . . .	72
8.14.2.2 Get_Qpad() . . . . .	73
8.14.2.3 Set_Lambda() . . . . .	73
8.14.2.4 SetSecondaries() . . . . .	73
8.14.2.5 WriteOut() . . . . .	73
8.14.3 Member Data Documentation . . . . .	73
8.14.3.1 m_Lambda . . . . .	73
8.15 Model_Charge1 Class Reference . . . . .	74
8.15.1 Constructor & Destructor Documentation . . . . .	74
8.15.1.1 Model_Charge1() . . . . .	74
8.15.1.2 ~Model_Charge1() . . . . .	75
8.15.2 Member Function Documentation . . . . .	75
8.15.2.1 Get_Gain() . . . . .	75
8.15.2.2 Get_Qpad() . . . . .	75
8.15.2.3 Get_RC() . . . . .	75
8.15.2.4 Get_Width() . . . . .	75
8.15.2.5 Set_Gain() . . . . .	75
8.15.2.6 Set_RC() . . . . .	75
8.15.2.7 Set_Width() . . . . .	75
8.15.2.8 Set_WidthFromDriftDistance() . . . . .	76

---

8.15.2.9 SetSecondaries()	76
8.15.2.10 WidthFromDriftDistance()	76
8.15.2.11 WriteOut()	76
8.15.3 Member Data Documentation	76
8.15.3.1 m_2RCinv	76
8.15.3.2 m_Gain	76
8.15.3.3 m_NormQPad	76
8.15.3.4 m_RC	76
8.15.3.5 m_Width	76
8.15.3.6 m_Width2	77
8.16 Model_Electronics Class Reference	77
8.16.1 Constructor & Destructor Documentation	77
8.16.1.1 Model_Electronics()	77
8.16.1.2 ~Model_Electronics()	77
8.16.2 Member Function Documentation	77
8.16.2.1 Derive_Response_Base()	77
8.16.2.2 Get_Amplitude()	78
8.16.2.3 Get_QValue()	78
8.16.2.4 Get_ws()	78
8.16.2.5 Response_Base()	78
8.16.2.6 Set_Amplitude()	78
8.16.2.7 Set_TimeShape()	78
8.16.2.8 WriteOut()	78
8.17 Model_ReadOutGeometry Class Reference	78
8.17.1 Constructor & Destructor Documentation	79
8.17.1.1 Model_ReadOutGeometry()	79
8.17.1.2 ~Model_ReadOutGeometry()	79
8.17.2 Member Function Documentation	79
8.17.2.1 Get_LX()	79
8.17.2.2 Get LY()	80
8.17.2.3 Get_Nx()	80
8.17.2.4 Get_Ny()	80
8.17.2.5 Get_XcPad()	80
8.17.2.6 Get_XHPad()	80
8.17.2.7 Get_XLPad()	80
8.17.2.8 Get_Xpad_min()	80
8.17.2.9 Get_YcPad()	80
8.17.2.10 Get_YHPad()	81
8.17.2.11 Get_YLPad()	81
8.17.2.12 Get_Ypad_min()	81
8.17.2.13 GetPadEdges()	81
8.17.2.14 Set_LX()	81

---

8.17.2.15 Set_LY()	81
8.17.2.16 Set_Nx()	81
8.17.2.17 Set_Ny()	82
8.17.2.18 Set_Xpad_min()	82
8.17.2.19 Set_Ypad_min()	82
8.17.2.20 SquareGeometry()	82
8.17.2.21 SwapGeometry()	82
8.17.2.22 WriteOut()	82
8.18 Module Class Reference	82
8.18.1 Constructor & Destructor Documentation	84
8.18.1.1 Module()	84
8.18.1.2 ~Module()	84
8.18.2 Member Function Documentation	84
8.18.2.1 Add_Cluster()	84
8.18.2.2 Add_Pad()	84
8.18.2.3 Clear_Clusters()	84
8.18.2.4 Get_Cluster()	85
8.18.2.5 Get_Cluster_Copy()	85
8.18.2.6 Get_EntryNber()	85
8.18.2.7 Get_EventNber()	85
8.18.2.8 Get_ModuleNber()	85
8.18.2.9 Get_NberOfCluster()	85
8.18.2.10 GiveMe_AnUnfittedTrack()	85
8.18.2.11 GiveMe_Clusters_ForThisModule()	85
8.18.2.12 GiveMe_ModuleTrack()	85
8.18.2.13 Invalidate()	85
8.18.2.14 IsValid()	86
8.18.2.15 ReplaceClusters()	86
8.18.2.16 Set_ModuleTrack()	86
8.18.2.17 SmallDump()	86
8.18.2.18 Validate()	86
8.18.2.19 WriteOut()	86
8.19 Pad Class Reference	86
8.19.1 Constructor & Destructor Documentation	88
8.19.1.1 Pad() [1/2]	88
8.19.1.2 Pad() [2/2]	88
8.19.1.3 ~Pad()	89
8.19.2 Member Function Documentation	89
8.19.2.1 Clear_ADC()	89
8.19.2.2 Get_AMax()	89
8.19.2.3 Get_AMax_FIT()	89
8.19.2.4 Get_AMax_FromSet()	89

---

8.19.2.5 Get_AMax_True()	89
8.19.2.6 Get_AMax_WF()	89
8.19.2.7 Get_AMax_WF_01()	89
8.19.2.8 Get_APad()	89
8.19.2.9 Get_EntryNber()	90
8.19.2.10 Get_EventNber()	90
8.19.2.11 Get_FIT_A0M()	90
8.19.2.12 Get_FIT_A0P()	90
8.19.2.13 Get_FIT_Status()	90
8.19.2.14 Get_FIT_X0()	90
8.19.2.15 Get_FIT_Xmax()	90
8.19.2.16 Get_FIT_Xmin()	90
8.19.2.17 Get_FIT_Y0()	90
8.19.2.18 Get_iX()	90
8.19.2.19 Get_iY()	91
8.19.2.20 Get_LX()	91
8.19.2.21 Get LY()	91
8.19.2.22 Get_Model_Charge()	91
8.19.2.23 Get_Model_Electronics()	91
8.19.2.24 Get_Model_ReadOutGeometry()	91
8.19.2.25 Get_ModuleNber()	91
8.19.2.26 Get_PadName()	91
8.19.2.27 Get_Qpad()	91
8.19.2.28 Get_Time0()	91
8.19.2.29 Get_TMax()	92
8.19.2.30 Get_TMax_FIT()	92
8.19.2.31 Get_TMax_FromSet()	92
8.19.2.32 Get_TMax_True()	92
8.19.2.33 Get_TMax_WF()	92
8.19.2.34 Get_TMax_WF_01()	92
8.19.2.35 Get_vADC()	92
8.19.2.36 Get_XH()	92
8.19.2.37 Get_XL()	92
8.19.2.38 Get_XPad()	92
8.19.2.39 Get_XTrue()	93
8.19.2.40 Get_YH()	93
8.19.2.41 Get_YL()	93
8.19.2.42 Get_YPad()	93
8.19.2.43 Get_YTrue()	93
8.19.2.44 Invalidate()	93
8.19.2.45 IsValid()	93
8.19.2.46 Set_ADC()	93

8.19.2.47 Set_AMax() . . . . .	93
8.19.2.48 Set_AMax_True() . . . . .	93
8.19.2.49 Set_Data_2Use() . . . . .	94
8.19.2.50 Set_TMax() . . . . .	94
8.19.2.51 Set_TMax_True() . . . . .	94
8.19.2.52 SetSignalModel() . . . . .	94
8.19.2.53 Validate() . . . . .	94
8.19.2.54 WF_DoClosure() . . . . .	94
8.19.2.55 WriteOut() . . . . .	94
8.20 ParabolaFunction Class Reference . . . . .	94
8.20.1 Constructor & Destructor Documentation . . . . .	95
8.20.1.1 ParabolaFunction() . . . . .	95
8.20.1.2 ~ParabolaFunction() . . . . .	95
8.20.2 Member Function Documentation . . . . .	95
8.20.2.1 Get_A0() . . . . .	95
8.20.2.2 Get_X0() . . . . .	95
8.20.2.3 Get_Y0() . . . . .	95
8.20.2.4 operator()() . . . . .	95
8.20.2.5 Set_A0() . . . . .	96
8.20.2.6 Set_X0() . . . . .	96
8.20.2.7 Set_Y0() . . . . .	96
8.21 ParabolaFunctionNG Class Reference . . . . .	96
8.21.1 Constructor & Destructor Documentation . . . . .	96
8.21.1.1 ParabolaFunctionNG() . . . . .	96
8.21.1.2 ~ParabolaFunctionNG() . . . . .	97
8.21.2 Member Function Documentation . . . . .	97
8.21.2.1 Get_A0M() . . . . .	97
8.21.2.2 Get_A0P() . . . . .	97
8.21.2.3 Get_X0() . . . . .	97
8.21.2.4 Get_Y0() . . . . .	97
8.21.2.5 operator()() . . . . .	97
8.21.2.6 Set_A0M() . . . . .	97
8.21.2.7 Set_A0P() . . . . .	97
8.21.2.8 Set_X0() . . . . .	97
8.21.2.9 Set_Y0() . . . . .	98
8.22 PRFParameters Class Reference . . . . .	98
8.22.1 Constructor & Destructor Documentation . . . . .	98
8.22.1.1 PRFParameters() . . . . .	98
8.22.1.2 ~PRFParameters() . . . . .	98
8.22.2 Member Function Documentation . . . . .	98
8.22.2.1 Eval() . . . . .	98
8.22.2.2 operator()() . . . . .	99

---

8.22.2.3 SetPRF() . . . . .	99
8.23 Reconstruction::RecoCluster Class Reference . . . . .	99
8.23.1 Constructor & Destructor Documentation . . . . .	100
8.23.1.1 ~RecoCluster() . . . . .	100
8.23.2 Member Function Documentation . . . . .	100
8.23.2.1 ClassDef() . . . . .	100
8.23.3 Member Data Documentation . . . . .	100
8.23.3.1 ADCmax_base . . . . .	100
8.23.3.2 ALead_base . . . . .	100
8.23.3.3 ALLead_GCorr . . . . .	101
8.23.3.4 charge . . . . .	101
8.23.3.5 dEdxWF . . . . .	101
8.23.3.6 length . . . . .	101
8.23.3.7 LUTrhoLead . . . . .	101
8.23.3.8 NPads . . . . .	101
8.23.3.9 ratioCorr . . . . .	101
8.23.3.10 TLead . . . . .	101
8.23.3.11 v_pads . . . . .	101
8.23.3.12 yCluster . . . . .	101
8.23.3.13 yWeight . . . . .	102
8.24 Reconstruction::RecoEvent Class Reference . . . . .	102
8.24.1 Constructor & Destructor Documentation . . . . .	103
8.24.1.1 RecoEvent() . . . . .	103
8.24.1.2 ~RecoEvent() . . . . .	103
8.24.2 Member Function Documentation . . . . .	103
8.24.2.1 ClassDef() . . . . .	103
8.24.2.2 Clear() . . . . .	103
8.24.3 Member Data Documentation . . . . .	103
8.24.3.1 avg_pad_mult . . . . .	103
8.24.3.2 dEdxGP1 . . . . .	103
8.24.3.3 dEdxGP2 . . . . .	103
8.24.3.4 dEdxGP3 . . . . .	103
8.24.3.5 dEdxGP4 . . . . .	104
8.24.3.6 dEdxGP5 . . . . .	104
8.24.3.7 dEdxWF . . . . .	104
8.24.3.8 dEdxWFnoTrunc . . . . .	104
8.24.3.9 dEdxXP . . . . .	104
8.24.3.10 dEdxXPnoTrunc . . . . .	104
8.24.3.11 eventNbr . . . . .	104
8.24.3.12 GWF . . . . .	104
8.24.3.13 GWFtruncatedGP1 . . . . .	104
8.24.3.14 lengthWF . . . . .	104

---

8.24.3.15 lengthXP . . . . .	105
8.24.3.16 NClusters . . . . .	105
8.24.3.17 NCrossedPads . . . . .	105
8.24.3.18 NPads . . . . .	105
8.24.3.19 numberOfModules . . . . .	105
8.24.3.20 peakingTime . . . . .	105
8.24.3.21 selected . . . . .	105
8.24.3.22 timeBinSize . . . . .	105
8.24.3.23 v_modules . . . . .	105
8.24.3.24 v_modules_position . . . . .	105
8.25 Reconstruction::RecoModule Class Reference . . . . .	106
8.25.1 Constructor & Destructor Documentation . . . . .	106
8.25.1.1 ~RecoModule() . . . . .	106
8.25.2 Member Function Documentation . . . . .	106
8.25.2.1 ClassDef() . . . . .	106
8.25.3 Member Data Documentation . . . . .	107
8.25.3.1 avg_pad_mult . . . . .	107
8.25.3.2 dEdxWF . . . . .	107
8.25.3.3 dEdxWFnoTrunc . . . . .	107
8.25.3.4 dEdxXP . . . . .	107
8.25.3.5 dEdxXPnoTrunc . . . . .	107
8.25.3.6 ID . . . . .	107
8.25.3.7 lengthWF . . . . .	107
8.25.3.8 lengthXP . . . . .	107
8.25.3.9 NClusters . . . . .	107
8.25.3.10 NCrossedPads . . . . .	107
8.25.3.11 NPads . . . . .	108
8.25.3.12 phi . . . . .	108
8.25.3.13 position . . . . .	108
8.25.3.14 selected . . . . .	108
8.25.3.15 Track . . . . .	108
8.25.3.16 v_clusters . . . . .	108
8.26 Reconstruction::RecoPad Class Reference . . . . .	108
8.26.1 Constructor & Destructor Documentation . . . . .	109
8.26.1.1 ~RecoPad() . . . . .	109
8.26.2 Member Function Documentation . . . . .	109
8.26.2.1 ClassDef() . . . . .	109
8.26.3 Member Data Documentation . . . . .	109
8.26.3.1 ADCmax . . . . .	109
8.26.3.2 ADCmax_base . . . . .	110
8.26.3.3 charge . . . . .	110
8.26.3.4 d . . . . .	110

---

8.26.3.5 dd . . . . .	110
8.26.3.6 dEdxXP . . . . .	110
8.26.3.7 driftDistance . . . . .	110
8.26.3.8 dy . . . . .	110
8.26.3.9 gain . . . . .	110
8.26.3.10 GainCorrection . . . . .	110
8.26.3.11 ix . . . . .	110
8.26.3.12 iy . . . . .	111
8.26.3.13 leading . . . . .	111
8.26.3.14 length . . . . .	111
8.26.3.15 phi . . . . .	111
8.26.3.16 ratio . . . . .	111
8.26.3.17 ratioDrift . . . . .	111
8.26.3.18 ratioFile . . . . .	111
8.26.3.19 RC . . . . .	111
8.26.3.20 T0 . . . . .	111
8.26.3.21 TMax . . . . .	111
8.26.3.22 xPad . . . . .	112
8.26.3.23 yPad . . . . .	112
8.27 ROBoard Class Reference . . . . .	112
8.27.1 Detailed Description . . . . .	113
8.27.2 Constructor & Destructor Documentation . . . . .	113
8.27.2.1 ROBoard() . . . . .	113
8.27.2.2 ~ROBoard() . . . . .	113
8.27.3 Member Function Documentation . . . . .	113
8.27.3.1 Add_Pad() . . . . .	113
8.27.3.2 Get_IsThisPadExisting() . . . . .	113
8.27.3.3 Get_Model_Charge() . . . . .	113
8.27.3.4 Get_Model_Electronics() . . . . .	114
8.27.3.5 Get_Model_ReadOutGeometry() . . . . .	114
8.27.3.6 Get_NberOfPads() . . . . .	114
8.27.3.7 Get_Pad() [1/3] . . . . .	114
8.27.3.8 Get_Pad() [2/3] . . . . .	114
8.27.3.9 Get_Pad() [3/3] . . . . .	114
8.27.3.10 GetLinearIndex() . . . . .	114
8.27.3.11 Ini_Models() . . . . .	114
8.27.4 Member Data Documentation . . . . .	115
8.27.4.1 m_Nx . . . . .	115
8.27.4.2 m_Ny . . . . .	115
8.27.4.3 p_Model_Charge . . . . .	115
8.27.4.4 p_Model_Electronics . . . . .	115
8.27.4.5 p_Model_ReadOutGeometry . . . . .	115

---

8.27.4.6 V_ExisFlag2D . . . . .	115
8.27.4.7 V_Pad . . . . .	115
8.28 Sample Class Reference . . . . .	115
8.28.1 Constructor & Destructor Documentation . . . . .	116
8.28.1.1 Sample() [1/2] . . . . .	116
8.28.1.2 Sample() [2/2] . . . . .	116
8.28.1.3 ~Sample() . . . . .	116
8.28.2 Member Function Documentation . . . . .	117
8.28.2.1 Add_Event() . . . . .	117
8.28.2.2 Get_Chi2Min() . . . . .	117
8.28.2.3 Get_DD() . . . . .	117
8.28.2.4 Get_DD_BeforeMinimisation() . . . . .	117
8.28.2.5 Get_eDD() . . . . .	117
8.28.2.6 Get_eRC() . . . . .	117
8.28.2.7 Get_Event() . . . . .	117
8.28.2.8 Get_Model_Charge() . . . . .	117
8.28.2.9 Get_Model_Electronics() . . . . .	117
8.28.2.10 Get_Model_ReadOutGeometry() . . . . .	118
8.28.2.11 Get_NberOfEvents() . . . . .	118
8.28.2.12 Get_RC() . . . . .	118
8.28.2.13 Get_RC_BeforeMinimisation() . . . . .	118
8.28.2.14 Get_SetOfTracks_ForThisModule() . . . . .	118
8.28.2.15 GetFilePRF() . . . . .	118
8.28.2.16 Set_DD() . . . . .	118
8.28.2.17 Set_Model_Charge() . . . . .	118
8.28.2.18 Set_Model_Electronics() . . . . .	118
8.28.2.19 Set_Model_ReadOutGeometry() . . . . .	119
8.28.2.20 Set_RC() . . . . .	119
8.28.2.21 SetFilePRF() . . . . .	119
8.28.2.22 SmallDump() . . . . .	119
8.28.2.23 StatusFit() . . . . .	119
8.28.2.24 WriteOut() . . . . .	119
8.29 Selector Class Reference . . . . .	119
8.29.1 Detailed Description . . . . .	120
8.29.2 Constructor & Destructor Documentation . . . . .	121
8.29.2.1 Selector() [1/2] . . . . .	121
8.29.2.2 Selector() [2/2] . . . . .	121
8.29.2.3 ~Selector() . . . . .	121
8.29.3 Member Function Documentation . . . . .	121
8.29.3.1 Add_Selection() . . . . .	121
8.29.3.2 Apply_ASelection() [1/2] . . . . .	121
8.29.3.3 Apply_ASelection() [2/2] . . . . .	121

---

8.29.3.4 ApplySelection() [1/2] . . . . .	122
8.29.3.5 ApplySelection() [2/2] . . . . .	122
8.29.3.6 Get_Cut_Stage2_EventBased() . . . . .	122
8.29.3.7 Get_Cut_Stage3_THigh() . . . . .	122
8.29.3.8 Get_Cut_Stage3_TLow() . . . . .	122
8.29.3.9 Get_Cut_Stage4_APM_High() . . . . .	122
8.29.3.10 Get_Cut_Stage4_APM_Low() . . . . .	122
8.29.3.11 Get_Cut_Stage5_Npads_Hig() . . . . .	122
8.29.3.12 Get_Cut_Stage6_Amax_Hig() . . . . .	123
8.29.3.13 Get_Cut_Stage6_Amax_Low() . . . . .	123
8.29.3.14 Get_Cut_StageFinal_NCluster_Low() . . . . .	123
8.29.3.15 Get_SelectionName() . . . . .	123
8.29.3.16 NberOfSelections() . . . . .	123
8.29.3.17 PrintStat() . . . . .	123
8.29.3.18 Reset_Selection() . . . . .	123
8.29.3.19 Reset_StatCounters() . . . . .	123
8.29.3.20 Set_Cut_Stage2_EventBased() . . . . .	123
8.29.3.21 Set_Cut_Stage3_THigh() . . . . .	124
8.29.3.22 Set_Cut_Stage3_TLow() . . . . .	124
8.29.3.23 Set_Cut_Stage4_APM_High() . . . . .	124
8.29.3.24 Set_Cut_Stage4_APM_Low() . . . . .	124
8.29.3.25 Set_Cut_Stage5_Npads_Hig() . . . . .	124
8.29.3.26 Set_Cut_Stage6_Amax_Hig() . . . . .	124
8.29.3.27 Set_Cut_Stage6_Amax_Low() . . . . .	124
8.29.3.28 Set_Cut_StageFinal_NCluster_Low() . . . . .	124
8.29.3.29 Tell_Selection() . . . . .	125
8.30 SetOfTracks Class Reference . . . . .	125
8.30.1 Constructor & Destructor Documentation . . . . .	126
8.30.1.1 SetOfTracks() . . . . .	126
8.30.1.2 ~SetOfTracks() . . . . .	126
8.30.2 Member Function Documentation . . . . .	126
8.30.2.1 Add_Track() . . . . .	126
8.30.2.2 DumpRec() . . . . .	126
8.30.2.3 Get_MinimalNberOfEntries() . . . . .	126
8.30.2.4 Get_NberOfTrack() . . . . .	126
8.30.2.5 Get_Track() . . . . .	126
8.30.2.6 GiveMe_pTH1F_Ch2Min() . . . . .	127
8.30.2.7 GiveMe_pTH1F_Ch2MinPerNODF() . . . . .	127
8.30.2.8 GiveMe_pTH1F_Pull() [1/5] . . . . .	127
8.30.2.9 GiveMe_pTH1F_Pull() [2/5] . . . . .	127
8.30.2.10 GiveMe_pTH1F_Pull() [3/5] . . . . .	127
8.30.2.11 GiveMe_pTH1F_Pull() [4/5] . . . . .	127

---

8.30.2.12 GiveMe_pTH1F_Pull() [5/5] . . . . .	128
8.30.2.13 GiveMe_pTH1F_Residual() [1/3] . . . . .	128
8.30.2.14 GiveMe_pTH1F_Residual() [2/3] . . . . .	128
8.30.2.15 GiveMe_pTH1F_Residual() [3/3] . . . . .	128
8.30.2.16 GiveMe_pTH1F_TrackDeltaT() . . . . .	128
8.30.2.17 GiveMe_pTH1F_YFitCol() [1/4] . . . . .	129
8.30.2.18 GiveMe_pTH1F_YFitCol() [2/4] . . . . .	129
8.30.2.19 GiveMe_pTH1F_YFitCol() [3/4] . . . . .	129
8.30.2.20 GiveMe_pTH1F_YFitCol() [4/4] . . . . .	129
8.30.2.21 GiveMe_pTH1F_YTrackInTracks() [1/4] . . . . .	129
8.30.2.22 GiveMe_pTH1F_YTrackInTracks() [2/4] . . . . .	129
8.30.2.23 GiveMe_pTH1F_YTrackInTracks() [3/4] . . . . .	129
8.30.2.24 GiveMe_pTH1F_YTrackInTracks() [4/4] . . . . .	130
8.30.2.25 GiveMe_pTH1F_YTrackYPadInTracks() [1/2] . . . . .	130
8.30.2.26 GiveMe_pTH1F_YTrackYPadInTracks() [2/2] . . . . .	130
8.30.2.27 GiveMe_pTH1F_YTrackYPadLeadingInTracks() [1/2] . . . . .	130
8.30.2.28 GiveMe_pTH1F_YTrackYPadLeadingInTracks() [2/2] . . . . .	130
8.30.2.29 Set_MinimalNberOfEntries() . . . . .	130
8.31 StaticClusterFitter_Diagonal Class Reference . . . . .	131
8.31.1 Constructor & Destructor Documentation . . . . .	131
8.31.1.1 StaticClusterFitter_Diagonal() . . . . .	131
8.31.1.2 ~StaticClusterFitter_Diagonal() . . . . .	131
8.31.2 Member Function Documentation . . . . .	131
8.31.2.1 Set() . . . . .	131
8.31.3 Member Data Documentation . . . . .	131
8.31.3.1 p_ClusterFitter_Diagonal . . . . .	131
8.32 StaticClusterFitter_Horizontal Class Reference . . . . .	132
8.32.1 Constructor & Destructor Documentation . . . . .	132
8.32.1.1 StaticClusterFitter_Horizontal() . . . . .	132
8.32.1.2 ~StaticClusterFitter_Horizontal() . . . . .	132
8.32.2 Member Function Documentation . . . . .	132
8.32.2.1 Set() . . . . .	132
8.32.3 Member Data Documentation . . . . .	132
8.32.3.1 p_ClusterFitter_Horizontal . . . . .	132
8.33 StaticTrackFitter Class Reference . . . . .	133
8.33.1 Constructor & Destructor Documentation . . . . .	133
8.33.1.1 StaticTrackFitter() . . . . .	133
8.33.1.2 ~StaticTrackFitter() . . . . .	133
8.33.2 Member Function Documentation . . . . .	133
8.33.2.1 Set() . . . . .	133
8.33.3 Member Data Documentation . . . . .	133
8.33.3.1 p_TrackFitter . . . . .	133

---

8.34 Track Class Reference . . . . .	134
8.34.1 Constructor & Destructor Documentation . . . . .	134
8.34.1.1 Track() . . . . .	134
8.34.1.2 ~Track() . . . . .	135
8.34.2 Member Function Documentation . . . . .	135
8.34.2.1 Add_Cluster() . . . . .	135
8.34.2.2 Chi2() . . . . .	135
8.34.2.3 DoClosure() . . . . .	135
8.34.2.4 DumpRec() . . . . .	135
8.34.2.5 Get_Chi2Min() . . . . .	135
8.34.2.6 Get_Cluster() . . . . .	135
8.34.2.7 Get_CovMatrix() . . . . .	135
8.34.2.8 Get_EntryNber() . . . . .	136
8.34.2.9 Get_EventNber() . . . . .	136
8.34.2.10 Get_ModuleNber() . . . . .	136
8.34.2.11 Get_NberOfCluster() . . . . .	136
8.34.2.12 Get_ParameterError() . . . . .	136
8.34.2.13 Get_ParameterName() . . . . .	136
8.34.2.14 Get_ParameterValue() . . . . .	136
8.34.2.15 Get_ParameterValue_BeforeMinimisation() . . . . .	136
8.34.2.16 Get_Pull() . . . . .	137
8.34.2.17 Get_Residual() . . . . .	137
8.34.2.18 GetNberOfParameters() . . . . .	137
8.34.2.19 SetNberOfParameters() . . . . .	137
8.34.2.20 SetParameter() . . . . .	137
8.34.2.21 SetParameters_Internal() . . . . .	137
8.34.2.22 SetResults() . . . . .	137
8.34.2.23 Y_Position() . . . . .	137
8.34.3 Member Data Documentation . . . . .	137
8.34.3.1 m_NberOfClusters . . . . .	137
8.34.3.2 V_Cluster . . . . .	138
8.34.3.3 V_Pull . . . . .	138
8.34.3.4 V_Residual . . . . .	138
8.35 TrackFitter Class Reference . . . . .	138
8.35.1 Constructor & Destructor Documentation . . . . .	138
8.35.1.1 TrackFitter() . . . . .	138
8.35.1.2 ~TrackFitter() . . . . .	139
8.35.2 Member Function Documentation . . . . .	139
8.35.2.1 Chi2() . . . . .	139
8.35.2.2 DoMinimisation() . . . . .	139
8.35.2.3 Set_Track() . . . . .	139
8.35.3 Member Data Documentation . . . . .	140

---

8.35.3.1 m_FitterName . . . . .	140
8.35.3.2 m_NberOfParameters . . . . .	140
8.35.3.3 p_Track . . . . .	140
8.35.3.4 p_TVirtualFitter . . . . .	140
8.36 Uploader Class Reference . . . . .	140
8.36.1 Constructor & Destructor Documentation . . . . .	141
8.36.1.1 Uploader() . . . . .	141
8.36.1.2 ~Uploader() . . . . .	141
8.36.2 Member Function Documentation . . . . .	142
8.36.2.1 Get_a2() . . . . .	142
8.36.2.2 Get_a4() . . . . .	142
8.36.2.3 Get_b2() . . . . .	142
8.36.2.4 Get_b4() . . . . .	142
8.36.2.5 Get_Model_Charge1D() . . . . .	142
8.36.2.6 Get_Model_Electronics() . . . . .	142
8.36.2.7 Get_Model_ReadOutGeometry() . . . . .	142
8.36.2.8 Get_NberOfEvent() . . . . .	142
8.36.2.9 Get_Norm() . . . . .	142
8.36.2.10 Get_PRF_exist() . . . . .	142
8.36.2.11 Get_SampleFile() . . . . .	143
8.36.2.12 GiveMe_Event() . . . . .	143
8.36.3 Member Data Documentation . . . . .	143
8.36.3.1 m_a2 . . . . .	143
8.36.3.2 m_a4 . . . . .	143
8.36.3.3 m_b2 . . . . .	143
8.36.3.4 m_b4 . . . . .	143
8.36.3.5 m_NberOfEvent . . . . .	143
8.36.3.6 m_Norm . . . . .	143
8.36.3.7 m_PRF_exist . . . . .	143
8.36.3.8 m_SampleFile . . . . .	144
8.36.3.9 p_Model_Charge1D . . . . .	144
8.36.3.10 p_Model_Electronics . . . . .	144
8.36.3.11 p_Model_ReadOutGeometry . . . . .	144
8.36.3.12 p_TFile . . . . .	144
8.36.3.13 p_TTree . . . . .	144
8.37 Uploader_ERAM01 Class Reference . . . . .	144
8.37.1 Constructor & Destructor Documentation . . . . .	145
8.37.1.1 Uploader_ERAM01() . . . . .	145
8.37.1.2 ~Uploader_ERAM01() . . . . .	146
8.37.2 Member Function Documentation . . . . .	146
8.37.2.1 GiveMe_Event() [1/2] . . . . .	146
8.37.2.2 GiveMe_Event() [2/2] . . . . .	146

---

8.38 Uploader_MockUp_V1 Class Reference . . . . .	146
8.38.1 Constructor & Destructor Documentation . . . . .	147
8.38.1.1 Uploader_MockUp_V1() . . . . .	147
8.38.1.2 ~Uploader_MockUp_V1() . . . . .	147
8.38.2 Member Function Documentation . . . . .	148
8.38.2.1 GiveMe_Event() [1/2] . . . . .	148
8.38.2.2 GiveMe_Event() [2/2] . . . . .	148
8.39 Uploader_MockUp_V2 Class Reference . . . . .	148
8.39.1 Constructor & Destructor Documentation . . . . .	151
8.39.1.1 Uploader_MockUp_V2() . . . . .	151
8.39.1.2 ~Uploader_MockUp_V2() . . . . .	151
8.39.2 Member Function Documentation . . . . .	151
8.39.2.1 GiveMe_Event() [1/2] . . . . .	151
8.39.2.2 GiveMe_Event() [2/2] . . . . .	151
8.39.2.3 Init() . . . . .	151
8.39.2.4 SetBeforeMerging() . . . . .	151
8.39.2.5 Setswap_XY() . . . . .	151
8.39.3 Member Data Documentation . . . . .	152
8.39.3.1 angle_xy . . . . .	152
8.39.3.2 angle_yz . . . . .	152
8.39.3.3 b_angle_xy . . . . .	152
8.39.3.4 b_angle_yz . . . . .	152
8.39.3.5 b_beforeMerging . . . . .	152
8.39.3.6 b_charge . . . . .	152
8.39.3.7 b_clust_pos . . . . .	152
8.39.3.8 b_clust_pos_err . . . . .	152
8.39.3.9 b_date . . . . .	152
8.39.3.10 b_dEdx . . . . .	152
8.39.3.11 b_dx . . . . .	153
8.39.3.12 b_ev . . . . .	153
8.39.3.13 b_max_mult . . . . .	153
8.39.3.14 b_mean_mult . . . . .	153
8.39.3.15 b_module . . . . .	153
8.39.3.16 b_mom . . . . .	153
8.39.3.17 b_multiplicity . . . . .	153
8.39.3.18 b_offset . . . . .	153
8.39.3.19 b_pad_charge . . . . .	153
8.39.3.20 b_pad_time . . . . .	153
8.39.3.21 b_pad_wf_q . . . . .	154
8.39.3.22 b_pad_x . . . . .	154
8.39.3.23 b_pad_y . . . . .	154
8.39.3.24 b_qfrac . . . . .	154

---

8.39.3.25 b_quality . . . . .	154
8.39.3.26 b_residual . . . . .	154
8.39.3.27 b_residual_corr . . . . .	154
8.39.3.28 b_rob_clusters . . . . .	154
8.39.3.29 b_sina . . . . .	154
8.39.3.30 b_time . . . . .	154
8.39.3.31 b_track . . . . .	155
8.39.3.32 b_track_pos . . . . .	155
8.39.3.33 b_wf_fwhm . . . . .	155
8.39.3.34 b_wf_width . . . . .	155
8.39.3.35 beforeMerging . . . . .	155
8.39.3.36 charge . . . . .	155
8.39.3.37 clust_pos . . . . .	155
8.39.3.38 clust_pos_err . . . . .	155
8.39.3.39 date . . . . .	155
8.39.3.40 dEdx . . . . .	155
8.39.3.41 dx . . . . .	156
8.39.3.42 ev . . . . .	156
8.39.3.43 fCurrent . . . . .	156
8.39.3.44 max_mult . . . . .	156
8.39.3.45 mean_mult . . . . .	156
8.39.3.46 module . . . . .	156
8.39.3.47 mom . . . . .	156
8.39.3.48 multiplicity . . . . .	156
8.39.3.49 offset . . . . .	156
8.39.3.50 pad_charge . . . . .	156
8.39.3.51 pad_time . . . . .	157
8.39.3.52 pad_wf_q . . . . .	157
8.39.3.53 pad_x . . . . .	157
8.39.3.54 pad_y . . . . .	157
8.39.3.55 qfrac . . . . .	157
8.39.3.56 quality . . . . .	157
8.39.3.57 residual . . . . .	157
8.39.3.58 residual_corr . . . . .	157
8.39.3.59 rob_clusters . . . . .	157
8.39.3.60 sina . . . . .	157
8.39.3.61 time . . . . .	158
8.39.3.62 track . . . . .	158
8.39.3.63 track_pos . . . . .	158
8.39.3.64 wf_fwhm . . . . .	158
8.39.3.65 wf_width . . . . .	158
8.40 Uploader_Prototype Class Reference . . . . .	158

---

8.40.1 Constructor & Destructor Documentation . . . . .	159
8.40.1.1 Uploader_Protoype() . . . . .	159
8.40.1.2 ~Uploader_Protoype() . . . . .	159
8.40.2 Member Function Documentation . . . . .	160
8.40.2.1 GiveMe_Event() [1/2] . . . . .	160
8.40.2.2 GiveMe_Event() [2/2] . . . . .	160
<b>9 File Documentation</b> . . . . .	<b>161</b>
9.1 Apps/Analysiscxx File Reference . . . . .	161
9.1.1 Function Documentation . . . . .	161
9.1.1.1 main() . . . . .	161
9.2 Apps/DrawOutcxx File Reference . . . . .	161
9.2.1 Function Documentation . . . . .	162
9.2.1.1 main() . . . . .	162
9.3 Fitters/inc/ClusterFitter.h File Reference . . . . .	162
9.3.1 Detailed Description . . . . .	162
9.3.2 Function Documentation . . . . .	163
9.3.2.1 ClusterFit_Diagonal() . . . . .	163
9.3.2.2 ClusterFit_Diagonal_Cluster() . . . . .	163
9.3.2.3 ClusterFit_Diagonal_Event() . . . . .	163
9.3.2.4 ClusterFit_Horizontal() . . . . .	163
9.3.2.5 ClusterFit_Horizontal_Cluster() . . . . .	163
9.3.2.6 ClusterFit_Horizontal_Event() . . . . .	164
9.4 ClusterFitter.h . . . . .	164
9.5 Fitters/inc/PRFParameters.h File Reference . . . . .	165
9.5.1 Detailed Description . . . . .	165
9.6 PRFParameters.h . . . . .	166
9.7 Fitters/inc/TrackFitter.h File Reference . . . . .	166
9.7.1 Detailed Description . . . . .	166
9.8 TrackFitter.h . . . . .	167
9.9 Fitters/inc/TrackRecon.h File Reference . . . . .	167
9.9.1 Detailed Description . . . . .	168
9.9.2 Function Documentation . . . . .	168
9.9.2.1 TrackRecon() . . . . .	168
9.9.2.2 TrackRecon_Event() . . . . .	168
9.10 TrackRecon.h . . . . .	169
9.11 Fitters/src/ClusterFittercxx File Reference . . . . .	169
9.11.1 Function Documentation . . . . .	169
9.11.1.1 ClusterFit_Diagonal() . . . . .	169
9.11.1.2 ClusterFit_Diagonal_Cluster() . . . . .	170
9.11.1.3 ClusterFit_Diagonal_Event() . . . . .	170
9.11.1.4 ClusterFit_Horizontal() . . . . .	170

---

9.11.1.5 ClusterFit_Horizontal_Cluster()	170
9.11.1.6 ClusterFit_Horizontal_Event()	170
9.11.1.7 ClusterFitter_DiagonalFunction()	170
9.11.1.8 ClusterFitter_HorizontalFunction()	171
9.12 Fitters/src/PRFParameters.cxx File Reference	171
9.13 Fitters/src/TrackFitter.cxx File Reference	171
9.13.1 Function Documentation	171
9.13.1.1 TrackFitterFunction()	171
9.14 Fitters/src/TrackRecon.cxx File Reference	171
9.14.1 Function Documentation	172
9.14.1.1 TrackRecon()	172
9.14.1.2 TrackRecon_Event()	172
9.15 ModelEvent/inc/AmplitudeError.h File Reference	173
9.15.1 Detailed Description	173
9.15.2 Function Documentation	173
9.15.2.1 AmplitudeError()	173
9.16 AmplitudeError.h	173
9.17 ModelEvent/inc/Cluster.h File Reference	173
9.17.1 Detailed Description	174
9.18 Cluster.h	174
9.19 ModelEvent/inc/Event.h File Reference	176
9.19.1 Detailed Description	176
9.20 Event.h	176
9.21 ModelEvent/inc/FitOutput.h File Reference	178
9.21.1 Detailed Description	178
9.22 FitOutput.h	178
9.23 ModelEvent/inc/Module.h File Reference	179
9.23.1 Detailed Description	179
9.24 Module.h	179
9.25 ModelEvent/inc/Pad.h File Reference	180
9.25.1 Detailed Description	181
9.26 Pad.h	181
9.27 ModelEvent/inc/ROBoard.h File Reference	183
9.27.1 Detailed Description	184
9.27.2 Macro Definition Documentation	184
9.27.2.1 ROBoard_H	184
9.28 ROBoard.h	184
9.29 ModelEvent/inc/Sample.h File Reference	185
9.29.1 Detailed Description	185
9.30 Sample.h	185
9.31 ModelEvent/inc/SetOfTracks.h File Reference	187
9.31.1 Detailed Description	187

---

9.32 SetOfTracks.h . . . . .	187
9.33 ModelEvent/inc/TimeError.h File Reference . . . . .	189
9.33.1 Detailed Description . . . . .	189
9.33.2 Function Documentation . . . . .	189
9.33.2.1 TimeError() [1/2] . . . . .	189
9.33.2.2 TimeError() [2/2] . . . . .	189
9.34 TimeError.h . . . . .	189
9.35 ModelEvent/inc/Track.h File Reference . . . . .	190
9.35.1 Detailed Description . . . . .	190
9.36 Track.h . . . . .	190
9.37 ModelEvent/src/AmplitudeError.cxx File Reference . . . . .	191
9.37.1 Function Documentation . . . . .	191
9.37.1.1 AmplitudeError() . . . . .	191
9.37.1.2 AmplitudeError_Leading() . . . . .	192
9.37.1.3 AmplitudeError_NextLeading() . . . . .	192
9.37.1.4 AmplitudeError_NextNextLeading() . . . . .	192
9.38 ModelEvent/src/Cluster.cxx File Reference . . . . .	192
9.39 ModelEvent/src/Event.cxx File Reference . . . . .	192
9.40 ModelEvent/src/FitOutput.cxx File Reference . . . . .	192
9.41 ModelEvent/src/Module.cxx File Reference . . . . .	192
9.42 ModelEvent/src/Pad.cxx File Reference . . . . .	193
9.43 ModelEvent/src/ROBoard.cxx File Reference . . . . .	193
9.44 ModelEvent/src/Sample.cxx File Reference . . . . .	193
9.45 ModelEvent/src/SetOfTracks.cxx File Reference . . . . .	193
9.46 ModelEvent/src/TimeError.cxx File Reference . . . . .	193
9.46.1 Function Documentation . . . . .	194
9.46.1.1 TimeError() [1/2] . . . . .	194
9.46.1.2 TimeError() [2/2] . . . . .	194
9.46.1.3 TimeError_BASE() . . . . .	194
9.46.1.4 TimeError_Leading() . . . . .	194
9.46.1.5 TimeError_NextLeading() . . . . .	194
9.46.1.6 TimeError_NextNextLeading() . . . . .	194
9.47 ModelEvent/src/Track.cxx File Reference . . . . .	195
9.48 ModelSignal/inc/Model_Charge0D.h File Reference . . . . .	195
9.48.1 Detailed Description . . . . .	195
9.49 Model_Charge0D.h . . . . .	196
9.50 ModelSignal/inc/Model_Charge1D.h File Reference . . . . .	196
9.50.1 Detailed Description . . . . .	196
9.51 Model_Charge1D.h . . . . .	197
9.52 ModelSignal/inc/Model_Charge1.h File Reference . . . . .	197
9.52.1 Detailed Description . . . . .	197
9.53 Model_Charge1.h . . . . .	198

---

9.54 ModelSignal/inc/Model_Electronics.h File Reference . . . . .	198
9.54.1 Detailed Description . . . . .	198
9.55 Model_Electronics.h . . . . .	199
9.56 ModelSignal/inc/Model_ReadOutGeometry.h File Reference . . . . .	199
9.56.1 Detailed Description . . . . .	199
9.57 Model_ReadOutGeometry.h . . . . .	200
9.58 ModelSignal/src/Model_Charge0D.cxx File Reference . . . . .	201
9.59 ModelSignal/src/Model_Charge1D.cxx File Reference . . . . .	201
9.60 ModelSignal/src/Model_Charge1.cxx File Reference . . . . .	201
9.61 ModelSignal/src/Model_Electronics.cxx File Reference . . . . .	201
9.62 ModelSignal/src/Model_ReadOutGeometry.cxx File Reference . . . . .	201
9.63 Reconstruction/inc/CombinedFit.h File Reference . . . . .	201
9.63.1 Detailed Description . . . . .	202
9.63.2 Function Documentation . . . . .	202
9.63.2.1 combinedFit() . . . . .	202
9.64 CombinedFit.h . . . . .	202
9.65 Reconstruction/inc/Control.h File Reference . . . . .	202
9.65.1 Detailed Description . . . . .	203
9.65.2 Function Documentation . . . . .	203
9.65.2.1 Control() . . . . .	203
9.66 Control.h . . . . .	203
9.67 Reconstruction/inc/dEdx.h File Reference . . . . .	203
9.68 dEdx.h . . . . .	204
9.69 Reconstruction/inc/Displays.h File Reference . . . . .	207
9.69.1 Detailed Description . . . . .	208
9.69.2 Function Documentation . . . . .	208
9.69.2.1 DrawOut_ClusterWFDisplay() [1/2] . . . . .	208
9.69.2.2 DrawOut_ClusterWFDisplay() [2/2] . . . . .	208
9.69.2.3 DrawOut_EventDisplay() [1/2] . . . . .	208
9.69.2.4 DrawOut_EventDisplay() [2/2] . . . . .	209
9.69.2.5 DrawOut_GWF() . . . . .	209
9.69.2.6 DrawOut_WaveFormDisplay() . . . . .	209
9.69.2.7 GiveMe_EvtDisplay() [1/2] . . . . .	209
9.69.2.8 GiveMe_EvtDisplay() [2/2] . . . . .	209
9.69.2.9 GiveMe_TimeDisplay() . . . . .	210
9.69.2.10 GiveMe_WaveFormDisplay() . . . . .	210
9.69.2.11 NewClusterDisplay() . . . . .	210
9.69.2.12 NewClusterDisplayMinimal() . . . . .	210
9.70 Displays.h . . . . .	210
9.71 Reconstruction/inc/DrawOuts.h File Reference . . . . .	211
9.71.1 Detailed Description . . . . .	211
9.72 DrawOuts.h . . . . .	212

---

9.73 Reconstruction/inc/DrawOuts_old.h File Reference . . . . .	213
9.73.1 Detailed Description . . . . .	214
9.73.2 Function Documentation . . . . .	214
9.73.2.1 DrawOut_Escan() . . . . .	214
9.73.2.2 DrawOut_Methods() . . . . .	214
9.73.2.3 DrawOut_Phiscan() . . . . .	214
9.73.2.4 DrawOut_Phiscan_Z() . . . . .	214
9.73.2.5 DrawOut_Scans() . . . . .	215
9.73.2.6 DrawOut_Separation() . . . . .	215
9.73.2.7 DrawOut_Separation_Reduced() . . . . .	215
9.73.2.8 DrawOut_Systematics() . . . . .	215
9.73.2.9 DrawOut_Thetascan() . . . . .	215
9.73.2.10 DrawOut_Versions() . . . . .	215
9.73.2.11 DrawOut_Yscan() . . . . .	215
9.73.2.12 DrawOut_Zscan() . . . . .	216
9.73.2.13 DrawOut_Zscan_PT() . . . . .	216
9.74 DrawOuts_old.h . . . . .	216
9.75 Reconstruction/inc/LUTs.h File Reference . . . . .	217
9.76 LUTs.h . . . . .	218
9.77 Reconstruction/inc/Monitoring.h File Reference . . . . .	220
9.77.1 Detailed Description . . . . .	221
9.78 Monitoring.h . . . . .	221
9.79 Reconstruction/inc/Recon_LinkDef.h File Reference . . . . .	222
9.79.1 Detailed Description . . . . .	222
9.80 Recon_LinkDef.h . . . . .	222
9.81 Reconstruction/inc/ReconTools.h File Reference . . . . .	222
9.81.1 Detailed Description . . . . .	223
9.81.2 Function Documentation . . . . .	223
9.81.2.1 BetheBloch() . . . . .	223
9.81.2.2 BetheBlochBhabha() . . . . .	224
9.81.2.3 BetheBlochExp() . . . . .	224
9.81.2.4 ComputeCutStage3_Cut() . . . . .	224
9.81.2.5 corr_func() . . . . .	224
9.81.2.6 DrawTH2() . . . . .	224
9.81.2.7 Fit2Gauss() [1/2] . . . . .	224
9.81.2.8 Fit2Gauss() [2/2] . . . . .	224
9.81.2.9 FourModulesInLine() . . . . .	225
9.81.2.10 GetStage3Cut_CSV() . . . . .	225
9.81.2.11 hist_to_graph() . . . . .	225
9.81.2.12 Init_selection() . . . . .	225
9.81.2.13 local_params() . . . . .	225
9.81.2.14 readCSV() . . . . .	225

9.81.2.15 SetStage3Cut_CSV()	225
9.81.2.16 Swapped_graph()	226
9.81.2.17 trk_len()	226
9.82 ReconTools.h	226
9.83 Reconstruction/inc/Selector.h File Reference	227
9.83.1 Detailed Description	227
9.83.2 Macro Definition Documentation	228
9.83.2.1 Selector_H	228
9.84 Selector.h	228
9.85 Reconstruction/inc/Variables.h File Reference	229
9.85.1 Detailed Description	231
9.86 Variables.h	231
9.87 Reconstruction/src/CombinedFit.cxx File Reference	232
9.87.1 Macro Definition Documentation	232
9.87.1.1 COMBINEDFIT_H	232
9.87.2 Variable Documentation	232
9.87.2.1 iparmuon	232
9.87.2.2 iparpion	232
9.87.2.3 iparpositron	232
9.87.2.4 iparproton	233
9.87.2.5 npar	233
9.88 Reconstruction/src/Control.cxx File Reference	233
9.88.1 Function Documentation	233
9.88.1.1 Control()	233
9.89 Reconstruction/src/dEdx.cxx File Reference	234
9.89.1 Function Documentation	234
9.89.1.1 ClassImp() [1/4]	234
9.89.1.2 ClassImp() [2/4]	234
9.89.1.3 ClassImp() [3/4]	234
9.89.1.4 ClassImp() [4/4]	235
9.90 Reconstruction/src/Displays.cxx File Reference	235
9.90.1 Function Documentation	236
9.90.1.1 DrawOut_ClusterWFDDisplay() [1/2]	236
9.90.1.2 DrawOut_ClusterWFDDisplay() [2/2]	236
9.90.1.3 DrawOut_EventDisplay() [1/2]	236
9.90.1.4 DrawOut_EventDisplay() [2/2]	236
9.90.1.5 DrawOut_GWF()	237
9.90.1.6 DrawOut_WaveFormDisplay()	237
9.90.1.7 GiveMe_EvtDisplay() [1/2]	237
9.90.1.8 GiveMe_EvtDisplay() [2/2]	237
9.90.1.9 GiveMe_TimeDisplay()	237
9.90.1.10 GiveMe_WaveFormDisplay()	237

---

9.90.1.11 NewClusterDisplay()	238
9.90.1.12 NewClusterDisplayMinimal()	238
9.91 Reconstruction/src/DrawOuts.cxx File Reference	238
9.91.1 Function Documentation	238
9.91.1.1 DrawOut_corrections()	238
9.92 Reconstruction/src/DrawOuts_old.cxx File Reference	239
9.92.1 Function Documentation	239
9.92.1.1 DrawOut_Escan()	239
9.92.1.2 DrawOut_Phiscan_Z()	239
9.92.1.3 DrawOut_Scans()	240
9.92.1.4 DrawOut_Separation()	240
9.92.1.5 DrawOut_Separation_Reduced()	240
9.92.1.6 DrawOut_Systematics()	240
9.92.1.7 DrawOut_Versions()	240
9.92.1.8 DrawOut_Zscan_PT()	240
9.93 Reconstruction/src/LUTs.cxx File Reference	240
9.93.1 Function Documentation	241
9.93.1.1 ClassImp()	241
9.94 Reconstruction/src/Monitoring.cxx File Reference	241
9.95 Reconstruction/src/ReconTools.cxx File Reference	242
9.95.1 Function Documentation	242
9.95.1.1 BetheBloch()	242
9.95.1.2 BetheBlochBhabha()	242
9.95.1.3 BetheBlochExp()	243
9.95.1.4 ComputeCutStage3_Cut()	243
9.95.1.5 corr_func()	243
9.95.1.6 DrawTH2()	243
9.95.1.7 Fit2Gauss() [1/2]	243
9.95.1.8 Fit2Gauss() [2/2]	243
9.95.1.9 FourModulesInLine()	243
9.95.1.10 GetStage3Cut_CSV()	244
9.95.1.11 hist_to_graph()	244
9.95.1.12 Init_selection()	244
9.95.1.13 local_params()	244
9.95.1.14 readCSV()	244
9.95.1.15 SetStage3Cut_CSV()	244
9.95.1.16 Swapped_graph()	245
9.95.1.17 trk_len()	245
9.96 Reconstruction/src/Selector.cxx File Reference	245
9.97 Reconstruction/src/Variables.cxx File Reference	245
9.98 Uploader/inc/GiveMe_Uploader.h File Reference	246
9.98.1 Detailed Description	246

---

9.98.2 Function Documentation . . . . .	246
9.98.2.1 GiveMe_Uploader() . . . . .	246
9.99 GiveMe_Uploader.h . . . . .	246
9.100 Uploader/inc/Uploader.h File Reference . . . . .	246
9.100.1 Detailed Description . . . . .	247
9.101 Uploader.h . . . . .	247
9.102 Uploader/inc/Uploader_ERAM01.h File Reference . . . . .	248
9.102.1 Detailed Description . . . . .	248
9.103 Uploader_ERAM01.h . . . . .	248
9.104 Uploader/inc/Uploader_MockUp_V1.h File Reference . . . . .	248
9.104.1 Detailed Description . . . . .	249
9.105 Uploader_MockUp_V1.h . . . . .	249
9.106 Uploader/inc/Uploader_MockUp_V2.h File Reference . . . . .	249
9.106.1 Detailed Description . . . . .	249
9.107 Uploader_MockUp_V2.h . . . . .	250
9.108 Uploader/inc/Uploader_Prototype.h File Reference . . . . .	251
9.108.1 Detailed Description . . . . .	251
9.109 Uploader_Prototype.h . . . . .	251
9.110 Uploader/src/GiveMe_Uploader.cxx File Reference . . . . .	252
9.110.1 Function Documentation . . . . .	252
9.110.1.1 GiveMe_Uploader() . . . . .	252
9.111 Uploader/src/Uploader.cxx File Reference . . . . .	252
9.112 Uploader/src/Uploader_ERAM01.cxx File Reference . . . . .	252
9.113 Uploader/src/Uploader_MockUp_V1.cxx File Reference . . . . .	253
9.114 Uploader/src/Uploader_MockUp_V2.cxx File Reference . . . . .	253
9.115 Uploader/src/Uploader_Prototype.cxx File Reference . . . . .	253
9.116 Utilities/inc/FuncFromTGraph.h File Reference . . . . .	253
9.116.1 Detailed Description . . . . .	253
9.117 FuncFromTGraph.h . . . . .	253
9.118 Utilities/inc/GaussFunction.h File Reference . . . . .	254
9.118.1 Detailed Description . . . . .	254
9.119 GaussFunction.h . . . . .	254
9.120 Utilities/inc/Misc.h File Reference . . . . .	254
9.120.1 Detailed Description . . . . .	255
9.121 Misc.h . . . . .	255
9.122 Utilities/inc/ParabolaFunction.h File Reference . . . . .	255
9.122.1 Detailed Description . . . . .	255
9.123 ParabolaFunction.h . . . . .	256
9.124 Utilities/inc/ParabolaFunctionNG.h File Reference . . . . .	256
9.124.1 Detailed Description . . . . .	256
9.125 ParabolaFunctionNG.h . . . . .	257
9.126 Utilities/inc/SignalTools.h File Reference . . . . .	257

---

9.126.1 Detailed Description . . . . .	257
9.126.2 Function Documentation . . . . .	258
9.126.2.1 ETF() . . . . .	258
9.127 SignalTools.h . . . . .	258
9.128 Utilities/src(FuncFromTGraph.cxx File Reference . . . . .	258
9.129 Utilities/src/GaussFunction.cxx File Reference . . . . .	258
9.130 Utilities/src/ParabolaFunction.cxx File Reference . . . . .	258
9.131 Utilities/src/ParabolaFunctionNG.cxx File Reference . . . . .	259
9.132 Utilities/src/SignalTools.cxx File Reference . . . . .	259
9.132.1 Function Documentation . . . . .	259
9.132.1.1 ETF() . . . . .	259
<b>Index</b>	<b>261</b>



# Chapter 1

## Directory Hierarchy

### 1.1 Directories

Apps . . . . .	15
Analysis.cxx . . . . .	161
DrawOut.cxx . . . . .	161
Fitters . . . . .	15
inc . . . . .	15
ClusterFitter.h . . . . .	162
PRFParameters.h . . . . .	165
TrackFitter.h . . . . .	166
TrackRecon.h . . . . .	167
src . . . . .	18
ClusterFitter.cxx . . . . .	169
PRFParameters.cxx . . . . .	171
TrackFitter.cxx . . . . .	171
TrackRecon.cxx . . . . .	171
inc . . . . .	15
ClusterFitter.h . . . . .	162
PRFParameters.h . . . . .	165
TrackFitter.h . . . . .	166
TrackRecon.h . . . . .	167
inc . . . . .	16
AmplitudeError.h . . . . .	173
Cluster.h . . . . .	173
Event.h . . . . .	176
FitOutput.h . . . . .	178
Module.h . . . . .	179
Pad.h . . . . .	180
ROBoard.h . . . . .	183
Sample.h . . . . .	185
SetOfTracks.h . . . . .	187
TimeError.h . . . . .	189
Track.h . . . . .	190
inc . . . . .	16
Model_Charge0D.h . . . . .	195
Model_Charge1D.h . . . . .	196
Model_Charge1.h . . . . .	197
Model_Electronics.h . . . . .	198
Model_ReadOutGeometry.h . . . . .	199

inc . . . . .	17
CombinedFit.h . . . . .	201
Control.h . . . . .	202
dEdx.h . . . . .	203
Displays.h . . . . .	207
DrawOuts.h . . . . .	211
DrawOuts_old.h . . . . .	213
LUTs.h . . . . .	217
Monitoring.h . . . . .	220
Recon_LinkDef.h . . . . .	222
ReconTools.h . . . . .	222
Selector.h . . . . .	227
Variables.h . . . . .	229
inc . . . . .	17
GiveMe_Uploader.h . . . . .	246
Uploader.h . . . . .	246
Uploader_ERAM01.h . . . . .	248
Uploader_MockUp_V1.h . . . . .	248
Uploader_MockUp_V2.h . . . . .	249
Uploader_Protoype.h . . . . .	251
inc . . . . .	18
FuncFromTGraph.h . . . . .	253
GaussFunction.h . . . . .	254
Misc.h . . . . .	254
ParabolaFunction.h . . . . .	255
ParabolaFunctionNG.h . . . . .	256
SignalTools.h . . . . .	257
ModelEvent . . . . .	18
inc . . . . .	16
AmplitudeError.h . . . . .	173
Cluster.h . . . . .	173
Event.h . . . . .	176
FitOutput.h . . . . .	178
Module.h . . . . .	179
Pad.h . . . . .	180
ROBoard.h . . . . .	183
Sample.h . . . . .	185
SetOfTracks.h . . . . .	187
TimeError.h . . . . .	189
Track.h . . . . .	190
src . . . . .	19
AmplitudeErrorcxx . . . . .	191
Clustercxx . . . . .	192
Eventcxx . . . . .	192
FitOutputcxx . . . . .	192
Modulecxx . . . . .	192
Padcxx . . . . .	193
ROBoardcxx . . . . .	193
Samplecxx . . . . .	193
SetOfTrackscxx . . . . .	193
TimeErrorcxx . . . . .	193
Trackcxx . . . . .	195
ModelSignal . . . . .	18
inc . . . . .	16
Model_Charge0D.h . . . . .	195
Model_Charge1D.h . . . . .	196
Model_Chargel.h . . . . .	197

Model_Electronics.h . . . . .	198
Model_ReadOutGeometry.h . . . . .	199
src . . . . .	19
Model_Charge0D.cxx . . . . .	201
Model_Charge1D.cxx . . . . .	201
Model_Chargel.cxx . . . . .	201
Model_Electronics.cxx . . . . .	201
Model_ReadOutGeometry.cxx . . . . .	201
Reconstruction . . . . .	18
inc . . . . .	17
CombinedFit.h . . . . .	201
Control.h . . . . .	202
dEdx.h . . . . .	203
Displays.h . . . . .	207
DrawOuts.h . . . . .	211
DrawOuts_old.h . . . . .	213
LUTs.h . . . . .	217
Monitoring.h . . . . .	220
Recon_LinkDef.h . . . . .	222
ReconTools.h . . . . .	222
Selector.h . . . . .	227
Variables.h . . . . .	229
src . . . . .	19
CombinedFit.cxx . . . . .	232
Control.cxx . . . . .	233
dEdx.cxx . . . . .	234
Displays.cxx . . . . .	235
DrawOuts.cxx . . . . .	238
DrawOuts_old.cxx . . . . .	239
LUTs.cxx . . . . .	240
Monitoring.cxx . . . . .	241
ReconTools.cxx . . . . .	242
Selector.cxx . . . . .	245
Variables.cxx . . . . .	245
src . . . . .	18
ClusterFitter.cxx . . . . .	169
PRFParameters.cxx . . . . .	171
TrackFitter.cxx . . . . .	171
TrackRecon.cxx . . . . .	171
src . . . . .	19
AmplitudeError.cxx . . . . .	191
Cluster.cxx . . . . .	192
Event.cxx . . . . .	192
FitOutput.cxx . . . . .	192
Module.cxx . . . . .	192
Pad.cxx . . . . .	193
ROBoard.cxx . . . . .	193
Sample.cxx . . . . .	193
SetOfTracks.cxx . . . . .	193
TimeError.cxx . . . . .	193
Track.cxx . . . . .	195
src . . . . .	19
Model_Charge0D.cxx . . . . .	201
Model_Charge1D.cxx . . . . .	201
Model_Chargel.cxx . . . . .	201
Model_Electronics.cxx . . . . .	201
Model_ReadOutGeometry.cxx . . . . .	201

src . . . . .	19
CombinedFit.cxx	232
Control.cxx	233
dEdx.cxx	234
Displays.cxx	235
DrawOuts.cxx	238
DrawOuts_old.cxx	239
LUTs.cxx	240
Monitoring.cxx	241
ReconTools.cxx	242
Selector.cxx	245
Variables.cxx	245
src . . . . .	19
GiveMe_Uploader.cxx	252
Uploader.cxx	252
Uploader_ERAM01.cxx	252
Uploader_MockUp_V1.cxx	253
Uploader_MockUp_V2.cxx	253
Uploader_Protoype.cxx	253
src . . . . .	20
FuncFromTGraph.cxx	258
GaussFunction.cxx	258
ParabolaFunction.cxx	258
ParabolaFunctionNG.cxx	259
SignalTools.cxx	259
Uploader . . . . .	20
inc . . . . .	17
GiveMe_Uploader.h	246
Uploader.h	246
Uploader_ERAM01.h	248
Uploader_MockUp_V1.h	248
Uploader_MockUp_V2.h	249
Uploader_Protoype.h	251
src . . . . .	19
GiveMe_Uploader.cxx	252
Uploader.cxx	252
Uploader_ERAM01.cxx	252
Uploader_MockUp_V1.cxx	253
Uploader_MockUp_V2.cxx	253
Uploader_Protoype.cxx	253
Utilities . . . . .	20
inc . . . . .	18
FuncFromTGraph.h	253
GaussFunction.h	254
Misc.h	254
ParabolaFunction.h	255
ParabolaFunctionNG.h	256
SignalTools.h	257
src . . . . .	20
FuncFromTGraph.cxx	258
GaussFunction.cxx	258
ParabolaFunction.cxx	258
ParabolaFunctionNG.cxx	259
SignalTools.cxx	259

# Chapter 2

## Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Reconstruction</a>	DE/dx estimation and particle identification declarations . . . . .	<a href="#">21</a>
--------------------------------	---	--------------------



# Chapter 3

## Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Cluster . . . . .	33
ClusterFitter_Diagonal . . . . .	41
ClusterFitter_Horizontal . . . . .	43
Reconstruction::dEdx . . . . .	45
Reconstruction::DrawOuts . . . . .	47
Reconstruction::ERAMMMaps . . . . .	55
Event . . . . .	56
FitOutput . . . . .	60
GaussFunction . . . . .	64
GlobalChi2_4 . . . . .	66
Reconstruction::LUT . . . . .	67
Model_Charge1 . . . . .	74
Model_Charge0D . . . . .	68
Model_Charge1D . . . . .	71
Model_Electronics . . . . .	77
Model_ReadOutGeometry . . . . .	78
Pad . . . . .	86
ParabolaFunction . . . . .	94
ParabolaFunctionNG . . . . .	96
PRFParameters . . . . .	98
ROBoard . . . . .	112
Module . . . . .	82
Sample . . . . .	115
Selector . . . . .	119
SetOfTracks . . . . .	125
StaticClusterFitter_Diagonal . . . . .	131
StaticClusterFitter_Horizontal . . . . .	132
StaticTrackFitter . . . . .	133
TObject . . . . .	
EramInfo . . . . .	52
Reconstruction::RecoCluster . . . . .	99
Reconstruction::RecoEvent . . . . .	102
Reconstruction::RecoModule . . . . .	106
Reconstruction::RecoPad . . . . .	108
Track . . . . .	134

TrackFitter . . . . .	138
Uploader . . . . .	140
Uploader_ERAM01 . . . . .	144
Uploader_MockUp_V1 . . . . .	146
Uploader_MockUp_V2 . . . . .	148
Uploader_Prototype . . . . .	158

# Chapter 4

## Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Cluster . . . . .	33
ClusterFitter_Diagonal . . . . .	41
ClusterFitter_Horizontal . . . . .	43
Reconstruction::dEdx . . . . .	45
Reconstruction::DrawOuts	
Plotting helper for reconstruction results . . . . .	47
EramInfo	
Lookup table utilities used by reconstruction algorithms . . . . .	52
Reconstruction::ERAMMaps . . . . .	55
Event . . . . .	56
FitOutput . . . . .	60
GaussFunction . . . . .	64
GlobalChi2_4 . . . . .	66
Reconstruction::LUT . . . . .	67
Model_Charge0D . . . . .	68
Model_Charge1D . . . . .	71
Model_ChargeI . . . . .	74
Model_Electronics . . . . .	77
Model_ReadOutGeometry . . . . .	78
Module . . . . .	82
Pad . . . . .	86
ParabolaFunction . . . . .	94
ParabolaFunctionNG . . . . .	96
PRFParameters . . . . .	98
Reconstruction::RecoCluster . . . . .	99
Reconstruction::RecoEvent . . . . .	102
Reconstruction::RecoModule . . . . .	106
Reconstruction::RecoPad . . . . .	108
ROBoard	
Abstraction for a read-out board grouping pads and providing mapping . . . . .	112
Sample . . . . .	115
Selector	
Encapsulates event/module/cluster selection logic . . . . .	119
SetOfTracks . . . . .	125
StaticClusterFitter_Diagonal . . . . .	131

StaticClusterFitter_Horizontal . . . . .	132
StaticTrackFitter . . . . .	133
Track . . . . .	134
TrackFitter . . . . .	138
Uploader . . . . .	140
Uploader_ERAM01 . . . . .	144
Uploader_MockUp_V1 . . . . .	146
Uploader_MockUp_V2 . . . . .	148
Uploader_Prototype . . . . .	158

# Chapter 5

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

Apps/Analysis.cxx . . . . .	161
Apps/DrawOut.cxx . . . . .	161
Fitters/inc/ClusterFitter.h	
Declarations for cluster-level fitters used to fit pad-cluster charge distributions . . . . .	162
Fitters/inc/PRFParameters.h	
Parameter container and evaluator for the Pad Response Function (PRF) . . . . .	165
Fitters/inc/TrackFitter.h	
Declaration of the TrackFitter class used to perform track fits . . . . .	166
Fitters/inc/TrackRecon.h	
Declarations for track reconstruction utilities and helpers . . . . .	167
Fitters/src/ClusterFitter.cxx . . . . .	169
Fitters/src/PRFParameters.cxx . . . . .	171
Fitters/src/TrackFitter.cxx . . . . .	171
Fitters/src/TrackRecon.cxx . . . . .	171
ModelEvent/inc/AmplitudeError.h	
Amplitude error model utilities . . . . .	173
ModelEvent/inc/Cluster.h	
Cluster container representing a group of pads forming a detected charge . . . . .	173
ModelEvent/inc/Event.h	
Event container holding modules, clusters and tracks . . . . .	176
ModelEvent/inc/FitOutput.h	
Containers to hold fit results from cluster/track fitters . . . . .	178
ModelEvent/inc/Module.h	
Module container representing a detector module of pads and clusters . . . . .	179
ModelEvent/inc/Pad.h	
Pad class representing a single readout pad and its waveform data . . . . .	180
ModelEvent/inc/ROBoard.h	
Read-out board (ROBoard) abstraction and mapping helpers . . . . .	183
ModelEvent/inc/Sample.h	
Sample container representing a dataset (collection of events) . . . . .	185
ModelEvent/inc/SetOfTracks.h	
Container for collections of Track objects . . . . .	187
ModelEvent/inc/TimeError.h	
Time error model utilities . . . . .	189
ModelEvent/inc/Track.h	
Track class representing a reconstructed particle trajectory . . . . .	190

ModelEvent/src/AmplitudeError.cxx . . . . .	191
ModelEvent/src/Cluster.cxx . . . . .	192
ModelEvent/src/Event.cxx . . . . .	192
ModelEvent/src/FitOutput.cxx . . . . .	192
ModelEvent/src/Module.cxx . . . . .	192
ModelEvent/src/Pad.cxx . . . . .	193
ModelEvent/src/ROBoard.cxx . . . . .	193
ModelEvent/src/Sample.cxx . . . . .	193
ModelEvent/src/SetOfTracks.cxx . . . . .	193
ModelEvent/src/TimeError.cxx . . . . .	193
ModelEvent/src/Track.cxx . . . . .	195
ModelSignal/inc/Model_Charge0D.h	
0D charge model declaration (inherits from Model_Charge)	195
ModelSignal/inc/Model_Charge1D.h	
1D charge model declaration (inherits from Model_Charge)	196
ModelSignal/inc/Model_Charge1.h	
Abstract base class for charge models	197
ModelSignal/inc/Model_Electronics.h	
Electronics response model declaration	198
ModelSignal/inc/Model_ReadOutGeometry.h	
Readout geometry model declaration	199
ModelSignal/src/Model_Charge0D.cxx . . . . .	201
ModelSignal/src/Model_Charge1D.cxx . . . . .	201
ModelSignal/src/Model_Charge1.h . . . . .	201
ModelSignal/src/Model_Electronics.cxx . . . . .	201
ModelSignal/src/Model_ReadOutGeometry.cxx . . . . .	201
Reconstruction/inc/CombinedFit.h	
Combined fitting utilities declaration	201
Reconstruction/inc/Control.h	
High-level orchestration function declaration	202
Reconstruction/inc/dEdx.h . . . . .	203
Reconstruction/inc/Displays.h	
High-level display utilities and plotting helpers	207
Reconstruction/inc/DrawOuts.h	
Declaration of the DrawOuts helper class for plotting and summaries	211
Reconstruction/inc/DrawOuts_old.h	
Legacy plotting helpers (kept for reference)	213
Reconstruction/inc/LUTs.h . . . . .	217
Reconstruction/inc/Monitoring.h	
Monitoring utilities and entry points for analysis workflows	220
Reconstruction/inc/Recon_LinkDef.h	
ROOT linkdef file for Reconstruction module classes	222
Reconstruction/inc/ReconTools.h	
Miscellaneous reconstruction helper functions and utilities	222
Reconstruction/inc/Selector.h	
Selection utilities to filter events, modules and clusters	227
Reconstruction/inc/Variables.h	
Central place to declare reconstruction-wide variables and helpers	229
Reconstruction/src/CombinedFit.cxx . . . . .	232
Reconstruction/src/Control.cxx . . . . .	233
Reconstruction/src/dEdx.cxx . . . . .	234
Reconstruction/src/Displays.cxx . . . . .	235
Reconstruction/src/DrawOuts.cxx . . . . .	238
Reconstruction/src/DrawOuts_old.cxx . . . . .	239
Reconstruction/src/LUTs.cxx . . . . .	240
Reconstruction/src/Monitoring.cxx . . . . .	241
Reconstruction/src/ReconTools.cxx . . . . .	242
Reconstruction/src/Selector.cxx . . . . .	245

Reconstruction/src/Variables.cxx . . . . .	245
Uploader/inc/GiveMe_Uploader.h	
Factory that constructs a concrete <a href="#">Uploader</a> instance . . . . .	246
Uploader/inc/Uploader.h	
Abstract base class for data uploaders . . . . .	246
Uploader/inc/Uploader_ERAM01.h	
<a href="#">Uploader</a> for ERAM01-format data files . . . . .	248
Uploader/inc/Uploader_MockUp_V1.h	
Mock-up uploader (version 1) declaration for testing . . . . .	248
Uploader/inc/Uploader_MockUp_V2.h	
Mock-up uploader (version 2) declaration for richer test datasets . . . . .	249
Uploader/inc/Uploader_Protoype.h	
Prototype-format uploader declaration . . . . .	251
Uploader/src/GiveMe_Uploader.cxx . . . . .	252
Uploader/src/Uploader.cxx . . . . .	252
Uploader/src/Uploader_ERAM01.cxx . . . . .	252
Uploader/src/Uploader_MockUp_V1.cxx . . . . .	253
Uploader/src/Uploader_MockUp_V2.cxx . . . . .	253
Uploader/src/Uploader_Protoype.cxx . . . . .	253
Utilities/inc/FuncFromTGraph.h	
Create function wrappers from ROOT TGraph objects . . . . .	253
Utilities/inc/GaussFunction.h	
Gaussian function wrapper declaration . . . . .	254
Utilities/inc/Misc.h	
Small general-purpose utilities and common definitions . . . . .	254
Utilities/inc/ParabolaFunction.h	
Parabola helper function declaration . . . . .	255
Utilities/inc/ParabolaFunctionNG.h	
Non-symmetric parabola helper function declaration . . . . .	256
Utilities/inc/SignalTools.h	
Signal-processing helper declarations (baseline, pulse finding, ...) . . . . .	257
Utilities/src/FuncFromTGraph.cxx . . . . .	258
Utilities/src/GaussFunction.cxx . . . . .	258
Utilities/src/ParabolaFunction.cxx . . . . .	258
Utilities/src/ParabolaFunctionNG.cxx . . . . .	259
Utilities/src/SignalTools.cxx . . . . .	259



# Chapter 6

## Directory Documentation

### 6.1 Apps Directory Reference

#### Files

- file [Analysis.cxx](#)
- file [DrawOut.cxx](#)

### 6.2 Fitters Directory Reference

#### Directories

- directory [inc](#)
- directory [src](#)

### 6.3 Fitters/inc Directory Reference

#### Files

- file [ClusterFitter.h](#)  
*Declarations for cluster-level fitters used to fit pad-cluster charge distributions.*
- file [PRFParameters.h](#)  
*Parameter container and evaluator for the Pad Response Function (PRF).*
- file [TrackFitter.h](#)  
*Declaration of the TrackFitter class used to perform track fits.*
- file [TrackRecon.h](#)  
*Declarations for track reconstruction utilities and helpers.*

## 6.4 ModelEvent/inc Directory Reference

### Files

- file [AmplitudeError.h](#)  
*Amplitude error model utilities.*
- file [Cluster.h](#)  
*Cluster container representing a group of pads forming a detected charge.*
- file [Event.h](#)  
*Event container holding modules, clusters and tracks.*
- file [FitOutput.h](#)  
*Containers to hold fit results from cluster/track fitters.*
- file [Module.h](#)  
*Module container representing a detector module of pads and clusters.*
- file [Pad.h](#)  
*Pad class representing a single readout pad and its waveform data.*
- file [ROBoard.h](#)  
*Read-out board ([ROBoard](#)) abstraction and mapping helpers.*
- file [Sample.h](#)  
*Sample container representing a dataset (collection of events).*
- file [SetOfTracks.h](#)  
*Container for collections of [Track](#) objects.*
- file [TimeError.h](#)  
*Time error model utilities.*
- file [Track.h](#)  
*Track class representing a reconstructed particle trajectory.*

## 6.5 ModelSignal/inc Directory Reference

### Files

- file [Model\\_Charge0D.h](#)  
*0D charge model declaration (inherits from [Model\\_Charge1](#)).*
- file [Model\\_Charge1D.h](#)  
*1D charge model declaration (inherits from [Model\\_Charge1](#)).*
- file [Model\\_Charge1.h](#)  
*Abstract base class for charge models.*
- file [Model\\_Electronics.h](#)  
*Electronics response model declaration.*
- file [Model\\_ReadOutGeometry.h](#)  
*Readout geometry model declaration.*

## 6.6 Reconstruction/inc Directory Reference

### Files

- file [CombinedFit.h](#)  
*Combined fitting utilities declaration.*
- file [Control.h](#)  
*High-level orchestration function declaration.*
- file [dEdx.h](#)
- file [Displays.h](#)  
*High-level display utilities and plotting helpers.*
- file [DrawOuts.h](#)  
*Declaration of the DrawOuts helper class for plotting and summaries.*
- file [DrawOuts\\_old.h](#)  
*Legacy plotting helpers (kept for reference).*
- file [LUTs.h](#)
- file [Monitoring.h](#)  
*Monitoring utilities and entry points for analysis workflows.*
- file [Recon\\_LinkDef.h](#)  
*ROOT linkdef file for [Reconstruction](#) module classes.*
- file [ReconTools.h](#)  
*Miscellaneous reconstruction helper functions and utilities.*
- file [Selector.h](#)  
*Selection utilities to filter events, modules and clusters.*
- file [Variables.h](#)  
*Central place to declare reconstruction-wide variables and helpers.*

## 6.7 Uploader/inc Directory Reference

### Files

- file [GiveMe\\_Uploader.h](#)  
*Factory that constructs a concrete [Uploader](#) instance.*
- file [Uploader.h](#)  
*Abstract base class for data uploaders.*
- file [Uploader\\_ERAM01.h](#)  
*[Uploader](#) for ERAM01-format data files.*
- file [Uploader\\_MockUp\\_V1.h](#)  
*Mock-up uploader (version 1) declaration for testing.*
- file [Uploader\\_MockUp\\_V2.h](#)  
*Mock-up uploader (version 2) declaration for richer test datasets.*
- file [Uploader\\_Prototype.h](#)  
*Prototype-format uploader declaration.*

## 6.8 Utilities/inc Directory Reference

### Files

- file [FuncFromTGraph.h](#)  
*Create function wrappers from ROOT TGraph objects.*
- file [GaussFunction.h](#)  
*Gaussian function wrapper declaration.*
- file [Misc.h](#)  
*Small general-purpose utilities and common definitions.*
- file [ParabolaFunction.h](#)  
*Parabola helper function declaration.*
- file [ParabolaFunctionNG.h](#)  
*Non-symmetric parabola helper function declaration.*
- file [SignalTools.h](#)  
*Signal-processing helper declarations (baseline, pulse finding, ...).*

## 6.9 ModelEvent Directory Reference

### Directories

- directory [inc](#)
- directory [src](#)

## 6.10 ModelSignal Directory Reference

### Directories

- directory [inc](#)
- directory [src](#)

## 6.11 Reconstruction Directory Reference

### Directories

- directory [inc](#)
- directory [src](#)

## 6.12 Fitters/src Directory Reference

### Files

- file [ClusterFitter.cxx](#)
- file [PRFParameters.cxx](#)
- file [TrackFitter.cxx](#)
- file [TrackRecon.cxx](#)

## 6.13 ModelEvent/src Directory Reference

### Files

- file [AmplitudeError.cxx](#)
- file [Cluster.cxx](#)
- file [Event.cxx](#)
- file [FitOutput.cxx](#)
- file [Module.cxx](#)
- file [Pad.cxx](#)
- file [ROBoard.cxx](#)
- file [Sample.cxx](#)
- file [SetOfTracks.cxx](#)
- file [TimeError.cxx](#)
- file [Track.cxx](#)

## 6.14 ModelSignal/src Directory Reference

### Files

- file [Model\\_Charge0D.cxx](#)
- file [Model\\_Charge1D.cxx](#)
- file [Model\\_ChargeI.cxx](#)
- file [Model\\_Electronics.cxx](#)
- file [Model\\_ReadOutGeometry.cxx](#)

## 6.15 Reconstruction/src Directory Reference

### Files

- file [CombinedFit.cxx](#)
- file [Control.cxx](#)
- file [dEdx.cxx](#)
- file [Displays.cxx](#)
- file [DrawOuts.cxx](#)
- file [DrawOuts\\_old.cxx](#)
- file [LUTs.cxx](#)
- file [Monitoring.cxx](#)
- file [ReconTools.cxx](#)
- file [Selector.cxx](#)
- file [Variables.cxx](#)

## 6.16 Uploader/src Directory Reference

### Files

- file [GiveMe\\_Uploader.cxx](#)
- file [Uploader.cxx](#)
- file [Uploader\\_ERAM01.cxx](#)
- file [Uploader\\_MockUp\\_V1.cxx](#)
- file [Uploader\\_MockUp\\_V2.cxx](#)
- file [Uploader\\_Prototype.cxx](#)

## 6.17 Utilities/src Directory Reference

### Files

- file [FuncFromTGraph.cxx](#)
- file [GaussFunction.cxx](#)
- file [ParabolaFunction.cxx](#)
- file [ParabolaFunctionNG.cxx](#)
- file [SignalTools.cxx](#)

## 6.18 Uploader Directory Reference

### Directories

- directory [inc](#)
- directory [src](#)

## 6.19 Utilities Directory Reference

### Directories

- directory [inc](#)
- directory [src](#)

# Chapter 7

# Namespace Documentation

## 7.1 Reconstruction Namespace Reference

dE/dx estimation and particle identification declarations.

### Classes

- class [RecoPad](#)
- class [RecoCluster](#)
- class [RecoModule](#)
- class [RecoEvent](#)
- class [dEdx](#)
- class [DrawOuts](#)  
*Plotting helper for reconstruction results.*
- class [ERAMMMaps](#)
- class [LUT](#)

### Functions

- void [Monitoring \(\)](#)
- void [Correction \(const int &corrRC=1, const int &corrGain=1, const int &corrWF=1, const int &corrDrift=1, const int &saveSelectOnly=1\)](#)
- void [Settings \(const std::string &testbeam, const std::string &multiScanName, const std::string &scanName, const std::string &scanspec, const std::string &runvarstr\\_name, const int &uploader, const int &modules, const int &peaking\\_time, const int &diffusion, const int &drift\\_dist, const int &timbin\)](#)
- void [ClearVectors \(\)](#)
- void [DefaultAnalysis \(\)](#)
- void [DrawSingleScan \(const int &methods=0\)](#)
- void [DrawMultipleScan \(const int &methods=0\)](#)
- void [DrawCERN22Scan \(\)](#)
- void [WFCorrection \(const std::string &OutCorr\)](#)

## Variables

- Reconstruction::dEdx \* p\_dEdx
- Reconstruction::DrawOuts \* p\_DrawOuts
- int **prototype** = 0
- int **CERN\_Escan** = 1
- int **CERN\_drift** = 0
- int **DESY\_zscan**
- int **DESY\_yscan**
- int **DESY\_phi** = 0
- int **DESY\_theta** = 0
- int **dedx** = 1
- int **DO\_Control**
- int **DO\_dEdx**
- int **DO\_Comparison**
- Uploader \* p\_uploader
- Reconstruction::LUT \* p\_lut
- const std::string **dEdxPath** = "./"
- const std::string **dataPath** = **dEdxPath** + "Data/"
- std::string **dataScanPath** = ""
- const std::string **drawoutPath** = **dEdxPath** + "OUT\_Reconstruction\_GP/"
- std::string **drawoutMultiScanPath** = ""
- std::string **drawoutScanPath** = ""
- std::string **drawoutRunPath** = ""
- std::string **drawout\_file** = ""
- std::string **rootout\_file** = ""
- std::string **log\_file** = ""
- std::string **comment**
- std::string **selectionSet**
- std::string **inputDir**
- int **drawWhichMethods** = 0
- std::string **corrFuncPath** = ""
- std::vector< std::string > **vcorrFuncPaths**
- std::vector< float > **vScanVals**
- std::vector< std::string > **vScanLabels**
- std::vector< std::string > **v\_scanspec**
- std::vector< std::string > **v\_comments**
- std::vector< std::string > **vTags**
- std::vector< std::string > **v\_datafiles**
- std::vector< std::string > **v\_rootout\_files**
- std::string **tag**
- std::string **testbeam**
- std::string **multiScanName**
- std::string **scanName**
- std::string **runvarstr**
- int **scanIndex** = 0
- int **intUploader**
- int **moduleCase**
- int **PT**
- int **Dt**
- int **DtB**
- int **DtnoB**
- int **driftDist**
- int **TB**
- int **fcorrectGain**

- int `fcorrectRC`
- int `fcorrectWF`
- int `fcorrectDrift`
- int `fsaveSelectOnly`
- int `DESY_drift` = 0
- int `DESY_row` = 0
- int `correction_wf` = 0
- int `Draw_Control` = 0
- int `Draw_dEdx` = 0
- int `Draw_Comparison` = 0
- int `Draw_Corrections` = 0
- int `Draw_DESY21SingleScan` = 0
- int `Draw_DESY21MultScan` = 0
- int `Draw_CERN22Scan` = 0
- int `DtwithBhere` = 286
- int `DtwithoutBhere` = 323
- `Uploader * pUploader`

### 7.1.1 Detailed Description

dE/dx estimation and particle identification declarations.

#### 7.1.2 \* @file dEdx.h

7.1.3 \*

7.1.4 \* Contains declarations for data structures, helper classes and functions

7.1.5 \* used to compute energy loss estimates used by the reconstruction pipeline.

7.1.6 \*

7.1.7 \*

7.1.8 \* The implementation (`dEdx.cxx`) contains the core numerical routines and

7.1.9 \* algorithms used to compute energy loss per unit length from reconstructed

7.1.10 \* cluster and pad information, together with helper utilities for particle

7.1.11 \* identification. Interfaces for computing WF and XP estimates, GigaWaveform

7.1.12 \* helpers and truncated estimators are declared here; heavy numerical work

7.1.13 \* and algorithmic details live in the source file.

#### 7.1.14 autotoc\_md13

### 7.1.15 Function Documentation

#### 7.1.15.1 ClearVectors()

```
void Reconstruction::ClearVectors ()
```

### 7.1.15.2 Correction()

```
void Reconstruction::Correction (
    const int & corrRC = 1,
    const int & corrGain = 1,
    const int & corrWF = 1,
    const int & corrDrift = 1,
    const int & saveSelectOnly = 1)
```

### 7.1.15.3 DefaultAnalysis()

```
void Reconstruction::DefaultAnalysis ()
```

### 7.1.15.4 DrawCERN22Scan()

```
void Reconstruction::DrawCERN22Scan ()
```

### 7.1.15.5 DrawMultipleScan()

```
void Reconstruction::DrawMultipleScan (
    const int & methods = 0)
```

### 7.1.15.6 DrawSingleScan()

```
void Reconstruction::DrawSingleScan (
    const int & methods = 0)
```

### 7.1.15.7 Monitoring()

```
void Reconstruction::Monitoring ()
```

### 7.1.15.8 Settings()

```
void Reconstruction::Settings (
    const std::string & testbeam,
    const std::string & multiScanName,
    const std::string & scanName,
    const std::string & scanspec,
    const std::string & runvarstr_name,
    const int & uploader,
    const int & modules,
    const int & peaking_time,
    const int & diffusion,
    const int & drift_dist,
    const int & timbin)
```

### 7.1.15.9 WFCorrection()

```
void Reconstruction::WFCorrection (
    const std::string & OutCorr)
```

## 7.1.16 Variable Documentation

### 7.1.16.1 CERN\_drift

```
int Reconstruction::CERN_drift = 0
```

### 7.1.16.2 CERN\_Escan

```
int Reconstruction::CERN_Escan = 1
```

### 7.1.16.3 comment

```
std::string Reconstruction::comment
```

### 7.1.16.4 correction\_wf

```
int Reconstruction::correction_wf = 0
```

### 7.1.16.5 corrFuncPath

```
std::string Reconstruction::corrFuncPath = ""
```

### 7.1.16.6 dataPath

```
const std::string Reconstruction::dataPath = dEdxPath + "Data/"
```

### 7.1.16.7 dataScanPath

```
std::string Reconstruction::dataScanPath = ""
```

### 7.1.16.8 dedx

```
int Reconstruction::dedx = 1
```

### 7.1.16.9 dEdxPath

```
const std::string Reconstruction::dEdxPath = ".../"
```

### 7.1.16.10 DESY\_drift

```
int Reconstruction::DESY_drift = 0
```

### 7.1.16.11 DESY\_phi

```
int Reconstruction::DESY_phi = 0
```

### 7.1.16.12 DESY\_row

```
int Reconstruction::DESY_row = 0
```

### 7.1.16.13 DESY\_theta

```
int Reconstruction::DESY_theta = 0
```

### 7.1.16.14 DESY\_yscan

```
int Reconstruction::DESY_yscan [extern]
```

### 7.1.16.15 DESY\_zscan

```
int Reconstruction::DESY_zscan [extern]
```

### 7.1.16.16 DO\_Comparison

```
int Reconstruction::DO_Comparison [extern]
```

### 7.1.16.17 DO\_Control

```
int Reconstruction::DO_Control [extern]
```

### 7.1.16.18 DO\_dEdx

```
int Reconstruction::DO_dEdx [extern]
```

### 7.1.16.19 Draw\_CERN22Scan

```
int Reconstruction::Draw_CERN22Scan = 0
```

**7.1.16.20 Draw\_Comparison**

```
int Reconstruction::Draw_Comparison = 0
```

**7.1.16.21 Draw\_Control**

```
int Reconstruction::Draw_Control = 0
```

**7.1.16.22 Draw\_Corrections**

```
int Reconstruction::Draw_Corrections = 0
```

**7.1.16.23 Draw\_dEdx**

```
int Reconstruction::Draw_dEdx = 0
```

**7.1.16.24 Draw\_DESY21MultScan**

```
int Reconstruction::Draw_DESY21MultScan = 0
```

**7.1.16.25 Draw\_DESY21SingleScan**

```
int Reconstruction::Draw_DESY21SingleScan = 0
```

**7.1.16.26 drawout\_file**

```
std::string Reconstruction::drawout_file = ""
```

**7.1.16.27 drawoutMultiScanPath**

```
std::string Reconstruction::drawoutMultiScanPath = ""
```

**7.1.16.28 drawoutPath**

```
const std::string Reconstruction::drawoutPath = dEdxPath + "OUT_Reconstruction_GP/"
```

**7.1.16.29 drawoutRunPath**

```
std::string Reconstruction::drawoutRunPath = ""
```

**7.1.16.30 drawoutScanPath**

```
std::string Reconstruction::drawoutScanPath = ""
```

**7.1.16.31 drawWhichMethods**

```
int Reconstruction::drawWhichMethods = 0
```

**7.1.16.32 driftDist**

```
int Reconstruction::driftDist
```

**7.1.16.33 Dt**

```
int Reconstruction::Dt
```

**7.1.16.34 DtB**

```
int Reconstruction::DtB
```

**7.1.16.35 DtnoB**

```
int Reconstruction::DtnoB
```

**7.1.16.36 DtwithBhere**

```
int Reconstruction::DtwithBhere = 286
```

**7.1.16.37 DtwithoutBhere**

```
int Reconstruction::DtwithoutBhere = 323
```

**7.1.16.38 fcorrectDrift**

```
int Reconstruction::fcorrectDrift
```

**7.1.16.39 fcorrectGain**

```
int Reconstruction::fcorrectGain
```

**7.1.16.40 fcorrectRC**

```
int Reconstruction::fcorrectRC
```

**7.1.16.41 fcorrectWF**

```
int Reconstruction::fcorrectWF
```

**7.1.16.42 fsaveSelectOnly**

```
int Reconstruction::fsaveSelectOnly
```

**7.1.16.43 inputDir**

```
std::string Reconstruction::inputDir
```

**7.1.16.44 intUploader**

```
int Reconstruction::intUploader
```

**7.1.16.45 log\_file**

```
std::string Reconstruction::log_file = ""
```

**7.1.16.46 moduleCase**

```
int Reconstruction::moduleCase
```

**7.1.16.47 multiScanName**

```
std::string Reconstruction::multiScanName
```

**7.1.16.48 p\_dEdx**

```
Reconstruction::dEdx * Reconstruction::p_dEdx
```

**7.1.16.49 p\_DrawOuts**

```
Reconstruction::DrawOuts * Reconstruction::p_DrawOuts
```

**7.1.16.50 p\_lut**

```
Reconstruction::LUT * Reconstruction::p_lut
```

**7.1.16.51 p\_uploader**

```
Uploader * Reconstruction::p_uploader
```

**7.1.16.52 prototype**

```
int Reconstruction::prototype = 0
```

**7.1.16.53 PT**

```
int Reconstruction::PT
```

**7.1.16.54 pUploader**

```
Uploader* Reconstruction::pUploader
```

**7.1.16.55 rootout\_file**

```
std::string Reconstruction::rootout_file = ""
```

**7.1.16.56 runvarstr**

```
std::string Reconstruction::runvarstr
```

**7.1.16.57 scanIndex**

```
int Reconstruction::scanIndex = 0
```

**7.1.16.58 scanName**

```
std::string Reconstruction::scanName
```

**7.1.16.59 selectionSet**

```
std::string Reconstruction::selectionSet
```

**7.1.16.60 tag**

```
std::string Reconstruction::tag
```

**7.1.16.61 TB**

```
int Reconstruction::TB
```

**7.1.16.62 testbeam**

```
std::string Reconstruction::testbeam
```

**7.1.16.63 v\_comments**

```
std::vector< std::string > Reconstruction::v_comments
```

**7.1.16.64 v\_datafiles**

```
std::vector< std::string > Reconstruction::v_datafiles
```

**7.1.16.65 v\_rootout\_files**

```
std::vector< std::string > Reconstruction::v_rootout_files
```

**7.1.16.66 v\_scanspec**

```
std::vector< std::string > Reconstruction::v_scanspec
```

**7.1.16.67 vcorrFuncPaths**

```
std::vector< std::string > Reconstruction::vcorrFuncPaths
```

**7.1.16.68 vScanLabels**

```
std::vector< std::string > Reconstruction::vScanLabels
```

**7.1.16.69 vScanVals**

```
std::vector< float > Reconstruction::vScanVals
```

**7.1.16.70 vTags**

```
std::vector< std::string > Reconstruction::vTags
```



# Chapter 8

## Class Documentation

### 8.1 Cluster Class Reference

```
#include <Cluster.h>
```

#### Public Member Functions

- `Cluster (const int &EventNber, const int &EntryNber, const int &ModuleNber)`  
*Construct a Cluster.*
- `virtual ~Cluster ()`  
*Virtual destructor.*
- `int Get_EventNber () const`
- `int Get_EntryNber () const`
- `int Get_ModuleNber () const`
- `void WriteOut () const`
- `int IsValid () const`
- `void Validate ()`
- `void Invalidate ()`
- `double Get_Acluster () const`
- `double Get_XWeight () const`
- `double Get_YWeight () const`
- `void Add_Pad (Pad *pPad)`
- `void DoClosure ()`
- `int Get_NberOfPads () const`
- `Pad * Get_Pad (const int &Index1D) const`
- `Pad * Get_LeadingPad () const`
- `double Get_YLeading () const`
- `double Get_TMaxLeading () const`
- `double Get_AMaxLeading () const`
- `Pad * Get_NextLeadingPad () const`
- `Pad * Get_NextNextLeadingPad () const`
- `void Set_XTrack (const double &XTrack)`
- `void Set_YTrack (const double &YTrack)`
- `void Set_eXTrack (const double &eXTrack)`
- `void Set_eYTrack (const double &eYTrack)`
- `int StatusFit ()`
- `double Get_XTrack () const`

- double `Get_eXTrack () const`
- double `Get_YTrack () const`
- double `Get_eYTrack () const`
- double `Get_Chi2Min () const`
- double `Get_XTrack_BeforeMinimisation () const`
- double `Get_YTrack_BeforeMinimisation () const`
- int `StatusFit_Horizontal ()`
- void `SetEval_Horizontal (TF1 *pTF1)`
- double `Eval_Horizontal (const double &Xin)`
- int `SetParameter_Horizontal (TVirtualFitter *pTVirtualFitter)`
- void `SetResults_Horizontal (TVirtualFitter *pTVirtualFitter)`
- void `SetResults_FailedFit_Horizontal (const int &Verbose)`
- double `Chi2_Horizontal (double par[ ])`
- int `FitRes_Horizontal_Get_NberOfTermsInChi2 () const`
- `Pad * FitRes_Horizontal_Get_Pad (const int &Index1D) const`
- double `FitRes_Horizontal_Get_Residual (const int &Index1D) const`
- double `FitRes_Horizontal_Get_Pull (const int &Index1D) const`
- int `StatusFit_Diagonal ()`
- void `SetEval_Diagonal (TF1 *pTF1)`
- double `Eval_Diagonal (const double &Xin)`
- int `SetParameter_Diagonal (TVirtualFitter *pTVirtualFitter)`
- void `SetResults_Diagonal (TVirtualFitter *pTVirtualFitter)`
- void `SetResults_FailedFit_Diagonal (const int &Verbose)`
- double `Chi2_Diagonal (double par[ ])`
- int `FitRes_Diagonal_Get_NberOfTermsInChi2 () const`
- `Pad * FitRes_Diagonal_Get_Pad (const int &Index1D) const`
- double `FitRes_Diagonal_Get_Residual (const int &Index1D) const`
- double `FitRes_Diagonal_Get_Pull (const int &Index1D) const`

## Public Attributes

- double `m_AngleRot`

## 8.1.1 Constructor & Destructor Documentation

### 8.1.1.1 Cluster()

```
Cluster::Cluster (
    const int & EventNber,
    const int & EntryNber,
    const int & ModuleNber)
```

Construct a `Cluster`.

#### Parameters

<code>EventNber</code>	event number
<code>EntryNber</code>	entry index
<code>ModuleNber</code>	module index

### 8.1.1.2 ~Cluster()

```
Cluster::~Cluster () [virtual]
```

Virtual destructor.

## 8.1.2 Member Function Documentation

### 8.1.2.1 Add\_Pad()

```
void Cluster::Add_Pad (
    Pad * pPad)
```

### 8.1.2.2 Chi2\_Diagonal()

```
double Cluster::Chi2_Diagonal (
    double par[ ])
```

### 8.1.2.3 Chi2\_Horizontal()

```
double Cluster::Chi2_Horizontal (
    double par[ ])
```

### 8.1.2.4 DoClosure()

```
void Cluster::DoClosure ()
```

### 8.1.2.5 Eval\_Diagonal()

```
double Cluster::Eval_Diagonal (
    const double & Xin)
```

### 8.1.2.6 Eval\_Horizontal()

```
double Cluster::Eval_Horizontal (
    const double & Xin)
```

### 8.1.2.7 FitRes\_Diagonal\_Get\_NberOfTermsInChi2()

```
int Cluster::FitRes_Diagonal_Get_NberOfTermsInChi2 () const
```

### 8.1.2.8 FitRes\_Diagonal\_Get\_Pad()

```
Pad * Cluster::FitRes_Diagonal_Get_Pad (
    const int & IndexID) const
```

### 8.1.2.9 FitRes\_Diagonal\_Get\_Pull()

```
double Cluster::FitRes_Diagonal_Get_Pull (
    const int & IndexID) const
```

### 8.1.2.10 FitRes\_Diagonal\_Get\_Residual()

```
double Cluster::FitRes_Diagonal_Get_Residual (
    const int & IndexID) const
```

### 8.1.2.11 FitRes\_Horizontal\_Get\_NberOfTermsInChi2()

```
int Cluster::FitRes_Horizontal_Get_NberOfTermsInChi2 () const
```

### 8.1.2.12 FitRes\_Horizontal\_Get\_Pad()

```
Pad * Cluster::FitRes_Horizontal_Get_Pad (
    const int & IndexID) const
```

### 8.1.2.13 FitRes\_Horizontal\_Get\_Pull()

```
double Cluster::FitRes_Horizontal_Get_Pull (
    const int & IndexID) const
```

### 8.1.2.14 FitRes\_Horizontal\_Get\_Residual()

```
double Cluster::FitRes_Horizontal_Get_Residual (
    const int & IndexID) const
```

### 8.1.2.15 Get\_Acluster()

```
double Cluster::Get_Acluster () const
```

### 8.1.2.16 Get\_AMaxLeading()

```
double Cluster::Get_AMaxLeading () const
```

**8.1.2.17 Get\_Chi2Min()**

```
double Cluster::Get_Chi2Min () const
```

**8.1.2.18 Get\_EntryNber()**

```
int Cluster::Get_EntryNber () const
```

**8.1.2.19 Get\_EventNber()**

```
int Cluster::Get_EventNber () const
```

**8.1.2.20 Get\_eXTrack()**

```
double Cluster::Get_eXTrack () const
```

**8.1.2.21 Get\_eYTrack()**

```
double Cluster::Get_eYTrack () const
```

**8.1.2.22 Get\_LeadingPad()**

```
Pad * Cluster::Get_LeadingPad () const
```

**8.1.2.23 Get\_ModuleNber()**

```
int Cluster::Get_ModuleNber () const
```

**8.1.2.24 Get\_NberOfPads()**

```
int Cluster::Get_NberOfPads () const
```

**8.1.2.25 Get\_NextLeadingPad()**

```
Pad * Cluster::Get_NextLeadingPad () const
```

**8.1.2.26 Get\_NextNextLeadingPad()**

```
Pad * Cluster::Get_NextNextLeadingPad () const
```

**8.1.2.27 Get\_Pad()**

```
Pad * Cluster::Get_Pad (
    const int & IndexID) const
```

**8.1.2.28 Get\_TMaxLeading()**

```
double Cluster::Get_TMaxLeading () const
```

**8.1.2.29 Get\_XTrack()**

```
double Cluster::Get_XTrack () const
```

**8.1.2.30 Get\_XTrack\_BeforeMinimisation()**

```
double Cluster::Get_XTrack_BeforeMinimisation () const
```

**8.1.2.31 Get\_XWeight()**

```
double Cluster::Get_XWeight () const
```

**8.1.2.32 Get\_YLeading()**

```
double Cluster::Get_YLeading () const
```

**8.1.2.33 Get\_YTrack()**

```
double Cluster::Get_YTrack () const
```

**8.1.2.34 Get\_YTrack\_BeforeMinimisation()**

```
double Cluster::Get_YTrack_BeforeMinimisation () const
```

**8.1.2.35 Get\_YWeight()**

```
double Cluster::Get_YWeight () const
```

**8.1.2.36 Invalidate()**

```
void Cluster::Invalidate ()
```

**8.1.2.37 IsValid()**

```
int Cluster::IsValid () const
```

**8.1.2.38 Set\_eXTrack()**

```
void Cluster::Set_eXTrack (
    const double & eXTrack)
```

**8.1.2.39 Set\_eYTrack()**

```
void Cluster::Set_eYTrack (
    const double & eYTrack)
```

**8.1.2.40 Set\_XTrack()**

```
void Cluster::Set_XTrack (
    const double & XTrack)
```

**8.1.2.41 Set\_YTrack()**

```
void Cluster::Set_YTrack (
    const double & YTrack)
```

**8.1.2.42 SetEval\_Diagonal()**

```
void Cluster::SetEval_Diagonal (
    TF1 * pTF1)
```

**8.1.2.43 SetEval\_Horizontal()**

```
void Cluster::SetEval_Horizontal (
    TF1 * pTF1)
```

**8.1.2.44 SetParameter\_Diagonal()**

```
int Cluster::SetParameter_Diagonal (
    TVirtualFitter * pTVirtualFitter)
```

**8.1.2.45 SetParameter\_Horizontal()**

```
int Cluster::SetParameter_Horizontal (
    TVirtualFitter * pTVirtualFitter)
```

**8.1.2.46 SetResults\_Diagonal()**

```
void Cluster::SetResults_Diagonal (
    TVirtualFitter * pTVirtualFitter)
```

**8.1.2.47 SetResults\_FailedFit\_Diagonal()**

```
void Cluster::SetResults_FailedFit_Diagonal (
    const int & Verbose)
```

**8.1.2.48 SetResults\_FailedFit\_Horizontal()**

```
void Cluster::SetResults_FailedFit_Horizontal (
    const int & Verbose)
```

**8.1.2.49 SetResults\_Horizontal()**

```
void Cluster::SetResults_Horizontal (
    TVirtualFitter * pTVirtualFitter)
```

**8.1.2.50 StatusFit()**

```
int Cluster::StatusFit ()
```

**8.1.2.51 StatusFit\_Diagonal()**

```
int Cluster::StatusFit_Diagonal ()
```

**8.1.2.52 StatusFit\_Horizontal()**

```
int Cluster::StatusFit_Horizontal ()
```

**8.1.2.53 Validate()**

```
void Cluster::Validate ()
```

**8.1.2.54 WriteOut()**

```
void Cluster::WriteOut () const
```

### 8.1.3 Member Data Documentation

#### 8.1.3.1 m\_AngleRot

```
double Cluster::m_AngleRot
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[Cluster.h](#)
- ModelEvent/src/[Cluster.cxx](#)

## 8.2 ClusterFitter\_Diagonal Class Reference

```
#include <ClusterFitter.h>
```

### Public Member Functions

- [ClusterFitter\\_Diagonal](#) (const std::string &FitterName)  
*Construct a diagonal cluster fitter.*
- virtual [~ClusterFitter\\_Diagonal](#) ()  
*Virtual destructor.*
- void [Set\\_Cluster](#) (Cluster \*pCluster)  
*Set the cluster to be fitted.*
- int [DoMinimisation](#) ()  
*Run minimisation for the current diagonal cluster.*
- double [Chi2](#) (double par[ ])  
*Compute chi2 for given parameters.*

### Protected Attributes

- std::string [m\\_FitterName](#)
- Cluster \* [p\\_Cluster](#)
- TVirtualFitter \* [p\\_TVirtualFitter](#)  
*Fitter.*

### 8.2.1 Constructor & Destructor Documentation

#### 8.2.1.1 ClusterFitter\_Diagonal()

```
ClusterFitter_Diagonal::ClusterFitter_Diagonal (
    const std::string & FitterName)
```

Construct a diagonal cluster fitter.

### Parameters

---

<i>FitterName</i>	name used to identify the fitter instance
-------------------	---

### 8.2.1.2 ~ClusterFitter\_Diagonal()

```
ClusterFitter_Diagonal::~ClusterFitter_Diagonal () [virtual]
```

Virtual destructor.

## 8.2.2 Member Function Documentation

### 8.2.2.1 Chi2()

```
double ClusterFitter_Diagonal::Chi2 (
    double par[])
```

Compute chi2 for given parameters.

#### Parameters

<i>par</i>	parameter array for chi2 evaluation
------------	-------------------------------------

#### Returns

chi2 value

### 8.2.2.2 DoMinimisation()

```
int ClusterFitter_Diagonal::DoMinimisation ()
```

Run minimisation for the current diagonal cluster.

#### Returns

0 on success, non-zero on failure

### 8.2.2.3 Set\_Cluster()

```
void ClusterFitter_Diagonal::Set_Cluster (
    Cluster * pCluster)
```

Set the cluster to be fitted.

#### Parameters

---

<code>pCluster</code>	pointer to the <a href="#">Cluster</a> (ownership not transferred)
-----------------------	--

## 8.2.3 Member Data Documentation

### 8.2.3.1 m\_FitterName

```
std::string ClusterFitter_Diagonal::m_FitterName [protected]
```

### 8.2.3.2 p\_Cluster

```
Cluster* ClusterFitter_Diagonal::p_Cluster [protected]
```

### 8.2.3.3 p\_TVirtualFitter

```
TVirtualFitter* ClusterFitter_Diagonal::p_TVirtualFitter [protected]
```

Fitter.

The documentation for this class was generated from the following files:

- Fitters/inc/[ClusterFitter.h](#)
- Fitters/src/[ClusterFitter.cxx](#)

## 8.3 ClusterFitter\_Horizontal Class Reference

```
#include <ClusterFitter.h>
```

### Public Member Functions

- [`ClusterFitter\_Horizontal`](#) (const std::string &FitterName)  
*Construct a horizontal cluster fitter.*
- virtual [`~ClusterFitter\_Horizontal`](#) ()  
*Virtual destructor.*
- void [`Set\_Cluster`](#) ([Cluster](#) \*pCluster)  
*Set the cluster to be fitted.*
- int [`DoMinimisation`](#) ()  
*Run minimisation for the current cluster.*
- double [`Chi2`](#) (double par[])  
*Compute chi2 for given parameters.*

## Protected Attributes

- std::string `m_FitterName`
  - Cluster \* `p_Cluster`
  - TVirtualFitter \* `p_TVirtualFitter`
- Fitter.*

## 8.3.1 Constructor & Destructor Documentation

### 8.3.1.1 ClusterFitter\_Horizontal()

```
ClusterFitter_Horizontal::ClusterFitter_Horizontal (
    const std::string & FitterName)
```

Construct a horizontal cluster fitter.

#### Parameters

<i>FitterName</i>	name used to identify the fitter instance
-------------------	---

### 8.3.1.2 ~ClusterFitter\_Horizontal()

```
ClusterFitter_Horizontal::~ClusterFitter_Horizontal () [virtual]
```

Virtual destructor.

## 8.3.2 Member Function Documentation

### 8.3.2.1 Chi2()

```
double ClusterFitter_Horizontal::Chi2 (
    double par[ ])
```

Compute chi2 for given parameters.

#### Parameters

<i>par</i>	parameter array for chi2 evaluation
------------	-------------------------------------

#### Returns

chi2 value

### 8.3.2.2 DoMinimisation()

```
int ClusterFitter_Horizontal::DoMinimisation ()
```

Run minimisation for the current cluster.

#### Returns

0 on success, non-zero on failure

### 8.3.2.3 Set\_Cluster()

```
void ClusterFitter_Horizontal::Set_Cluster (
    Cluster * pCluster)
```

Set the cluster to be fitted.

#### Parameters

<i>pCluster</i>	pointer to the <a href="#">Cluster</a> (ownership not transferred)
-----------------	--

## 8.3.3 Member Data Documentation

### 8.3.3.1 m\_FitterName

```
std::string ClusterFitter_Horizontal::m_FitterName [protected]
```

### 8.3.3.2 p\_Cluster

```
Cluster* ClusterFitter_Horizontal::p_Cluster [protected]
```

### 8.3.3.3 p\_TVirtualFitter

```
TVirtualFitter* ClusterFitter_Horizontal::p_TVirtualFitter [protected]
```

Fitter.

The documentation for this class was generated from the following files:

- Fitters/inc/[ClusterFitter.h](#)
- Fitters/src/[ClusterFitter.cxx](#)

## 8.4 Reconstruction::dEdx Class Reference

```
#include <dEdx.h>
```

## Public Member Functions

- [dEdx \(\)](#)
- [virtual ~dEdx \(\)](#)
- [void Reconstruction \(\)](#)
- [float ComputedEdxWF \(std::vector< float > v\\_dEdxWF, const int &nClusters, const float &alpha\)](#)
- [float ComputedEdxXP \(const std::vector< float > &v\\_dEdx, const std::vector< float > &v\\_dE, const std::vector< float > &v\\_dx, const int &nCrossedPads, const float &alpha\)](#)
- [float ComputedEdxGP1 \(const std::vector< TH1F > &vClusWF, const std::vector< float > &v\\_dEdx, const std::vector< float > &v\\_Aclus, const std::vector< float > &v\\_Lclus, const int &nClusters, const float &alpha\)](#)
- [float ComputedEdxGP \(const TH1F \\*GWF, const std::vector< TH1F > &vWF, const std::vector< float > &v\\_dEdx, const std::vector< float > &v\\_Amax, const float &eventLength, const std::vector< float > &v\\_Length, const int &nElements, const float &alpha\)](#)
- [TH1F \\* GetGigaWaveform \(const std::vector< TH1F > &vClusWF\)](#)
- [TH1F \\* GetTruncatedGigaWaveformGP1 \(const std::vector< TH1F > &vClusWF, const std::vector< float > &v\\_dEdx, const int &nClusters\)](#)
- [void DiscardedModule \(\)](#)

### 8.4.1 Constructor & Destructor Documentation

#### 8.4.1.1 dEdx()

```
Reconstruction::dEdx::dEdx ()
```

#### 8.4.1.2 ~dEdx()

```
Reconstruction::dEdx::~dEdx () [virtual]
```

### 8.4.2 Member Function Documentation

#### 8.4.2.1 ComputedEdxGP()

```
float Reconstruction::dEdx::ComputedEdxGP (
    const TH1F * GWF,
    const std::vector< TH1F > & vWF,
    const std::vector< float > & v_dEdx,
    const std::vector< float > & v_Amax,
    const float & eventLength,
    const std::vector< float > & v_Length,
    const int & nElements,
    const float & alpha)
```

#### 8.4.2.2 ComputedEdxGP1()

```
float Reconstruction::dEdx::ComputedEdxGP1 (
    const std::vector< TH1F > & vClusWF,
    const std::vector< float > & v_dEdx,
    const std::vector< float > & v_Aclus,
    const std::vector< float > & v_Lclus,
    const int & nClusters,
    const float & alpha)
```

#### 8.4.2.3 ComputedEdxWF()

```
float Reconstruction::dEdx::ComputedEdxWF (
    std::vector< float > v_dEdxWF,
    const int & NClusters,
    const float & alpha)
```

#### 8.4.2.4 ComputedEdxXP()

```
float Reconstruction::dEdx::ComputedEdxXP (
    const std::vector< float > & v_dEdx,
    const std::vector< float > & v_dE,
    const std::vector< float > & v_dx,
    const int & nCrossedPads,
    const float & alpha)
```

**dEdx** = sum(dE)/Sum(dx) and not average(dE/dx) of each pad => less sensitive to statistical fluctuations

#### 8.4.2.5 DiscardedModule()

```
void Reconstruction::dEdx::DiscardedModule ()
```

#### 8.4.2.6 GetGigaWaveform()

```
TH1F * Reconstruction::dEdx::GetGigaWaveform (
    const std::vector< TH1F > & vClusWF)
```

#### 8.4.2.7 GetTruncatedGigaWaveformGP1()

```
TH1F * Reconstruction::dEdx::GetTruncatedGigaWaveformGP1 (
    const std::vector< TH1F > & vClusWF,
    const std::vector< float > & v_dEdx,
    const int & nClusters)
```

#### 8.4.2.8 Reconstruction()

```
void Reconstruction::dEdx::Reconstruction ()
```

The documentation for this class was generated from the following files:

- Reconstruction/inc/[dEdx.h](#)
- Reconstruction/src/[dEdx.cxx](#)

## 8.5 Reconstruction::DrawOuts Class Reference

Plotting helper for reconstruction results.

```
#include <DrawOuts.h>
```

## Public Member Functions

- **DrawOuts ()**  
*Default constructor.*
- **DrawOuts (const std::string &inputFile)**  
*Construct from a single input file.*
- **DrawOuts (const std::vector< std::string > &v\_inputFiles)**  
*Construct from a list of input files.*
- **virtual ~DrawOuts ()**  
*Virtual destructor.*
- **void SetStyle ()**  
*Set plotting style (ROOT style settings).*
- **void DESY21ScanFill ()**  
*Fill DESY21 single-run scan data.*
- **void DESY21ScanDraw ()**  
*Draw DESY21 scan summaries.*
- **void CERN22ScanFill ()**  
*Fill CERN22 single-run scan data.*
- **void CERN22ScanDraw ()**  
*Draw CERN22 scan summaries.*
- **void Control ()**  
*General control/entry point for plotting.*
- **void EnergyLoss (const int &methods=0)**  
*Produce energy-loss plots.*
- **void FileComparison ()**  
*Produce file comparison plots.*
- **void DESY21SingleScan (const int &methods=0)**  
*Draw DESY21 single scan.*
- **void DESY21MultiScan (const int &methods=0)**  
*Draw DESY21 multi-scan summaries.*
- **void CERN22Scan ()**  
*Draw CERN22 scan summaries (wrapper).*
- **void AmplitudeVSLength ()**  
*Plot amplitude versus cluster length.*

### 8.5.1 Detailed Description

Plotting helper for reconstruction results.

Provides methods to create scan drawings, energy loss plots, comparisons and other visual summaries used by the analysis tools.

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 DrawOuts() [1/3]

```
Reconstruction::DrawOuts::DrawOuts ()
```

Default constructor.

### 8.5.2.2 DrawOuts() [2/3]

```
Reconstruction::DrawOuts::DrawOuts (
    const std::string & inputFile)
```

Construct from a single input file.

#### Parameters

---

<i>inputFile</i>	path to the input file
------------------	------------------------

### 8.5.2.3 DrawOuts() [3/3]

```
Reconstruction::DrawOuts::DrawOuts (
    const std::vector< std::string > & v_inputFiles)
```

Construct from a list of input files.

#### Parameters

<i>v_inputFiles</i>	vector with input file paths
---------------------	------------------------------

### 8.5.2.4 ~DrawOuts()

```
Reconstruction::DrawOuts::~DrawOuts () [virtual]
```

Virtual destructor.

## 8.5.3 Member Function Documentation

### 8.5.3.1 AmplitudeVSLength()

```
void Reconstruction::DrawOuts::AmplitudeVSLength ()
```

Plot amplitude versus cluster length.

### 8.5.3.2 CERN22Scan()

```
void Reconstruction::DrawOuts::CERN22Scan ()
```

Draw CERN22 scan summaries (wrapper).

### 8.5.3.3 CERN22ScanDraw()

```
void Reconstruction::DrawOuts::CERN22ScanDraw ()
```

Draw CERN22 scan summaries.

### 8.5.3.4 CERN22ScanFill()

```
void Reconstruction::DrawOuts::CERN22ScanFill ()
```

Fill CERN22 single-run scan data.

### 8.5.3.5 Control()

```
void Reconstruction::DrawOuts::Control ()
```

General control/entry point for plotting.

### 8.5.3.6 DESY21MultiScan()

```
void Reconstruction::DrawOuts::DESY21MultiScan (
    const int & methods = 0)
```

Draw DESY21 multi-scan summaries.

#### Parameters

<i>methods</i>	selection of methods (0: both, 1: only WF, 2: only XP)
----------------	--

### 8.5.3.7 DESY21ScanDraw()

```
void Reconstruction::DrawOuts::DESY21ScanDraw ()
```

Draw DESY21 scan summaries.

### 8.5.3.8 DESY21ScanFill()

```
void Reconstruction::DrawOuts::DESY21ScanFill ()
```

Fill DESY21 single-run scan data.

### 8.5.3.9 DESY21SingleScan()

```
void Reconstruction::DrawOuts::DESY21SingleScan (
    const int & methods = 0)
```

Draw DESY21 single scan.

#### Parameters

<i>methods</i>	selection of methods (0: both, 1: only WF, 2: only XP)
----------------	--

### 8.5.3.10 EnergyLoss()

```
void Reconstruction::DrawOuts::EnergyLoss (
    const int & methods = 0)
```

Produce energy-loss plots.

#### Parameters

---

<i>methods</i>	selection of methods (0: both, 1: only WF, 2: only XP)
----------------	--

Truncated ETF

### 8.5.3.11 FileComparison()

```
void Reconstruction::DrawOuts::FileComparison ()
```

Produce file comparison plots.

### 8.5.3.12 SetStyle()

```
void Reconstruction::DrawOuts::SetStyle ()
```

Set plotting style (ROOT style settings).

The documentation for this class was generated from the following files:

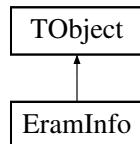
- Reconstruction/inc/[DrawOuts.h](#)
- Reconstruction/src/[DrawOuts.cxx](#)

## 8.6 EramInfo Class Reference

Lookup table utilities used by reconstruction algorithms.

```
#include <LUTs.h>
```

Inheritance diagram for EramInfo:



### Public Member Functions

- [EramInfo \(\)](#)
- virtual [~EramInfo \(\)](#)

### Public Attributes

- int [Id](#)
- int [Position](#)
- int [XX](#)
- int [YY](#)
- float [RC](#)
- float [Gain](#)
- float [Resolution](#)
- int [Endplate](#)
- bool [InbTPC](#)
- bool [IntTPC](#)

### 8.6.1 Detailed Description

Lookup table utilities used by reconstruction algorithms.

#### 8.6.2 \* @file LUTs.h

#### 8.6.3 \*

#### 8.6.4 \* Declares data structures and helpers for building and accessing LUTs

#### 8.6.5 \* (geometry, calibration, etc.) used by dE/dx and track/cluster

#### 8.6.6 \* reconstruction.

#### 8.6.7 \*

#### 8.6.8 \*

#### 8.6.9 \* The implementation file LUTs.cxx constructs ERAM maps, fills lookup tables

#### 8.6.10 \* and exposes accessors used across the reconstruction pipeline. The source

#### 8.6.11 \* also contains helpers to fill and repair maps (FillHoles), and code to

#### 8.6.12 \* compute mean gains/RC values used in calibration steps.

#### 8.6.13 autotoc\_md26

### 8.6.14 Constructor & Destructor Documentation

#### 8.6.14.1 EramInfo()

```
EramInfo::EramInfo () [inline]
```

#### 8.6.14.2 ~EramInfo()

```
virtual EramInfo::~EramInfo () [inline], [virtual]
```

### 8.6.15 Member Data Documentation

#### 8.6.15.1 Endplate

```
int EramInfo::Endplate
```

### 8.6.15.2 Gain

```
float EramInfo::Gain
```

### 8.6.15.3 Id

```
int EramInfo::Id
```

### 8.6.15.4 InbTPC

```
bool EramInfo::InbTPC
```

### 8.6.15.5 IntTPC

```
bool EramInfo::IntTPC
```

### 8.6.15.6 Position

```
int EramInfo::Position
```

### 8.6.15.7 RC

```
float EramInfo::RC
```

### 8.6.15.8 Resolution

```
float EramInfo::Resolution
```

### 8.6.15.9 XX

```
int EramInfo::XX
```

### 8.6.15.10 YY

```
int EramInfo::YY
```

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

- Reconstruction/inc/[LUTs.h](#)

## 8.7 Reconstruction::ERAMMaps Class Reference

```
#include <LUTs.h>
```

### Public Member Functions

- [ERAMMaps](#) (const std::string &file="\$HOME/Documents/Code/ERAMinfo/ERAMinfo.root")
- virtual [~ERAMMaps](#) ()
- int [ID](#) (const int &position)
- float [RC](#) (const int &position, const int &iX, const int &iY)
- float [Gain](#) (const int &position, const int &iX, const int &iY)
- float [Resolution](#) (const int &position, const int &iX, const int &iY)
- float [MeanGain](#) (const int &position)
- float [MeanRC](#) (const int &position)

### 8.7.1 Constructor & Destructor Documentation

#### 8.7.1.1 ERAMMaps()

```
Reconstruction::ERAMMaps::ERAMMaps (
    const std::string & file = "$HOME/Documents/Code/ERAMinfo/ERAMinfo.root")
```

#### 8.7.1.2 ~ERAMMaps()

```
Reconstruction::ERAMMaps::~ERAMMaps () [virtual]
```

### 8.7.2 Member Function Documentation

#### 8.7.2.1 Gain()

```
float Reconstruction::ERAMMaps::Gain (
    const int & position,
    const int & iX,
    const int & iY)
```

#### 8.7.2.2 ID()

```
int Reconstruction::ERAMMaps::ID (
    const int & position)
```

#### 8.7.2.3 MeanGain()

```
float Reconstruction::ERAMMaps::MeanGain (
    const int & position)
```

#### 8.7.2.4 MeanRC()

```
float Reconstruction::ERAMMaps::MeanRC (
    const int & position)
```

#### 8.7.2.5 RC()

```
float Reconstruction::ERAMMaps::RC (
    const int & position,
    const int & iX,
    const int & iY)
```

#### 8.7.2.6 Resolution()

```
float Reconstruction::ERAMMaps::Resolution (
    const int & position,
    const int & iX,
    const int & iY)
```

The documentation for this class was generated from the following files:

- Reconstruction/inc/LUTs.h
- Reconstruction/src/LUTs.cxx

## 8.8 Event Class Reference

```
#include <Event.h>
```

### Public Member Functions

- [Event \(const int &EventNber, const int &EntryNber, Model\\_ReadOutGeometry \\*pModel\\_ReadOutGeometry, Model\\_Electronics \\*pModel\\_Electronics, Model\\_ChargeL \\*pModel\\_ChargeL\)](#)
- virtual [~Event \(\)](#)
- int [Get\\_EventNber \(\) const](#)
- int [Get\\_EntryNber \(\) const](#)
- void [SmallDump \(\) const](#)
- void [WriteOut \(\) const](#)
- const [Model\\_ReadOutGeometry \\* Get\\_Model\\_ReadOutGeometry \(\)](#)
- const [Model\\_Electronics \\* Get\\_Model\\_Electronics \(\)](#)
- const [Model\\_ChargeL \\* Get\\_Model\\_ChargeL \(\)](#)
- int [IsValid \(\) const](#)
- void [Validate \(\)](#)
- void [Invalidate \(\)](#)
- void [Clear\\_Modules \(\)](#)
- void [Add\\_Module \(Module \\*pModule\)](#)
- [Module \\* Get\\_ThisModule \(const int &ModuleNber\)](#)
- const [Module \\* Get\\_ThisModuleConst \(const int &ModuleNber\) const](#)
- int [Get\\_NberOfModule \(\) const](#)
- [Module \\* Get\\_Module\\_InArray \(const int &Index1D\)](#)

- int `Validity_ForThisModule` (const int &ModuleNber) const
- void `Validate_ThisModule` (const int &ModuleNber)
- void `Invalidate_ThisModule` (const int &ModuleNber)
- std::vector< `Cluster` \* > `GiveMe_Clusters_ForThisModule` (const int &ModuleNber)
- double `GiveMe_CutSlopeXY_ForThisModule` (const int &ModuleNber)
- double `GiveMe_CutSlopeXZ_ForThisModule` (const int &ModuleNber)
- double `GiveMe_CutInterceptXZ_ForThisModule` (const int &ModuleNber)
- std::vector< double > `GiveMe_CutSlopeXYZ_ForThisModule` (const int &ModuleNber)
- void `Replace_Clusters_ForThisModule` (std::vector< `Cluster` \* > &VCluster, const int &ModuleNber)
- `Cluster` \* `Get_Cluster_Copy` (`Cluster` \*pCluster)
- `Track` \* `GiveMe_AnUnfittedTrack_ForThisModule` (const int &ModuleNber)
- void `Set_Track_ForThisModule` (`Track` \*pTrack, const int &ModuleNber)
- const `Track` \* `GiveMe_Track_ForThisModule` (const int &ModuleNber)

## 8.8.1 Constructor & Destructor Documentation

### 8.8.1.1 Event()

```
Event::Event (
    const int & EventNber,
    const int & EntryNber,
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_ChargeI * pModel_ChargeI)
```

Constructor

### 8.8.1.2 ~Event()

```
Event::~Event () [virtual]
```

## 8.8.2 Member Function Documentation

### 8.8.2.1 Add\_Module()

```
void Event::Add_Module (
    Module * pModule)
```

### 8.8.2.2 Clear\_Modules()

```
void Event::Clear_Modules ()
```

### 8.8.2.3 Get\_Cluster\_Copy()

```
Cluster * Event::Get_Cluster_Copy (
    Cluster * pCluster)
```

#### 8.8.2.4 Get\_EntryNber()

```
int Event::Get_EntryNber () const
```

#### 8.8.2.5 Get\_EventNber()

```
int Event::Get_EventNber () const
```

#### 8.8.2.6 Get\_Model\_ChargeI()

```
const Model_ChargeI * Event::Get_Model_ChargeI ()
```

#### 8.8.2.7 Get\_Model\_Electronics()

```
const Model_Electronics * Event::Get_Model_Electronics ()
```

#### 8.8.2.8 Get\_Model\_ReadOutGeometry()

```
const Model_ReadOutGeometry * Event::Get_Model_ReadOutGeometry ()
```

#### 8.8.2.9 Get\_Module\_InArray()

```
Module * Event::Get_Module_InArray (
    const int & IndexID)
```

#### 8.8.2.10 Get\_NberOfModule()

```
int Event::Get_NberOfModule () const
```

#### 8.8.2.11 Get\_ThisModule()

```
Module * Event::Get_ThisModule (
    const int & ModuleNber)
```

#### 8.8.2.12 Get\_ThisModuleConst()

```
const Module * Event::Get_ThisModuleConst (
    const int & ModuleNber) const
```

#### 8.8.2.13 GiveMe\_AnUnfittedTrack\_ForThisModule()

```
Track * Event::GiveMe_AnUnfittedTrack_ForThisModule (
    const int & ModuleNber)
```

**8.8.2.14 GiveMe\_Clusters\_ForThisModule()**

```
std::vector< Cluster * > Event::GiveMe_Clusters_ForThisModule (
    const int & ModuleNber)
```

**8.8.2.15 GiveMe\_CutInterCeptXZ\_ForThisModule()**

```
double Event::GiveMe_CutInterCeptXZ_ForThisModule (
    const int & ModuleNber)
```

**8.8.2.16 GiveMe\_CutSlopeXY\_ForThisModule()**

```
double Event::GiveMe_CutSlopeXY_ForThisModule (
    const int & ModuleNber)
```

**8.8.2.17 GiveMe\_CutSlopeXYZ\_ForThisModule()**

```
std::vector< double > Event::GiveMe_CutSlopeXYZ_ForThisModule (
    const int & ModuleNber)
```

**8.8.2.18 GiveMe\_CutSlopeXZ\_ForThisModule()**

```
double Event::GiveMe_CutSlopeXZ_ForThisModule (
    const int & ModuleNber)
```

**8.8.2.19 GiveMe\_Track\_ForThisModule()**

```
const Track * Event::GiveMe_Track_ForThisModule (
    const int & ModuleNber)
```

**8.8.2.20 Invalidate()**

```
void Event::Invalidate ()
```

**8.8.2.21 Invalidate\_ThisModule()**

```
void Event::Invalidate_ThisModule (
    const int & ModuleNber)
```

**8.8.2.22 IsValid()**

```
int Event::IsValid () const
```

### 8.8.2.23 Replace\_Clusters\_ForThisModule()

```
void Event::Replace_Clusters_ForThisModule (
    std::vector< Cluster * > & VCluster,
    const int & ModuleNber)
```

### 8.8.2.24 Set\_Track\_ForThisModule()

```
void Event::Set_Track_ForThisModule (
    Track * pTrack,
    const int & ModuleNber)
```

### 8.8.2.25 SmallDump()

```
void Event::SmallDump () const
```

### 8.8.2.26 Validate()

```
void Event::Validate ()
```

### 8.8.2.27 Validate\_ThisModule()

```
void Event::Validate_ThisModule (
    const int & ModuleNber)
```

### 8.8.2.28 Validity\_ForThisModule()

```
int Event::Validity_ForThisModule (
    const int & ModuleNber) const
```

### 8.8.2.29 WriteOut()

```
void Event::WriteOut () const
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[Event.h](#)
- ModelEvent/src/[Event.cxx](#)

## 8.9 FitOutput Class Reference

```
#include <FitOutput.h>
```

## Public Member Functions

- `FitOutput ()`
- `virtual ~FitOutput ()`
- `FitOutput (const FitOutput &ToBeCopied)`
- `FitOutput & operator= (const FitOutput &ToBeCopied)`
- `void Reset ()`
- `void Set (const int &NberOfModelParameters)`
- `void SetResults (TVirtualFitter *pTVirtualFitter)`
- `void SetResults (const std::vector< std::string > &V_PARname, const std::vector< double > &V_PAR, const std::vector< double > &V_ePAR)`
- `void PrintFitOutput ()`
- `void PrintFitOutputInCanvas (double Xstart, double Ystart, double Step)`

## Public Attributes

- `std::string m_NameOfTheModel`
- `int m_NberOfModelParameters`
- `int m_NberOfDataPoints`
- `double * p_par`  
*Parameters.*
- `std::string * p_parName`  
*Name.*
- `double * p_eparplus`  
*Errors.*
- `double * p_eparminus`  
*Errors.*
- `double * p_eparparab`  
*Errors.*
- `double * p_CovMatrix`  
*Cov matrix.*
- `double m_MinnLL`  
*Likelihood.*
- `std::string m_Description`

## 8.9.1 Constructor & Destructor Documentation

### 8.9.1.1 FitOutput() [1/2]

```
FitOutput::FitOutput ()
```

### 8.9.1.2 ~FitOutput()

```
FitOutput::~FitOutput () [virtual]
```

### 8.9.1.3 FitOutput() [2/2]

```
FitOutput::FitOutput (
    const FitOutput & ToBeCopied)
```

## 8.9.2 Member Function Documentation

### 8.9.2.1 operator=( )

```
FitOutput & FitOutput::operator= (
    const FitOutput & ToBeCopied)
```

### 8.9.2.2 PrintFitOutput()

```
void FitOutput::PrintFitOutput ()
```

### 8.9.2.3 PrintFitOutputInCanvas()

```
void FitOutput::PrintFitOutputInCanvas (
    double Xstart,
    double Ystart,
    double Step)
```

### 8.9.2.4 Reset()

```
void FitOutput::Reset ()
```

### 8.9.2.5 Set()

```
void FitOutput::Set (
    const int & NberOfModelParameters)
```

### 8.9.2.6 SetResults() [1/2]

```
void FitOutput::SetResults (
    const std::vector< std::string > & V_PARname,
    const std::vector< double > & V_PAR,
    const std::vector< double > & V_ePAR)
```

### 8.9.2.7 SetResults() [2/2]

```
void FitOutput::SetResults (
    TVirtualFitter * pTVirtualFitter)
```

## 8.9.3 Member Data Documentation

### 8.9.3.1 m\_Description

```
std::string FitOutput::m_Description
```

### 8.9.3.2 m\_MinnLL

```
double FitOutput::m_MinnLL
```

Likelihood.

### 8.9.3.3 m\_NameOfTheModel

```
std::string FitOutput::m_NameOfTheModel
```

### 8.9.3.4 m\_NberOfDataPoints

```
int FitOutput::m_NberOfDataPoints
```

### 8.9.3.5 m\_NberOfModelParameters

```
int FitOutput::m_NberOfModelParameters
```

### 8.9.3.6 p\_CovMatrix

```
double* FitOutput::p_CovMatrix
```

Cov matrix.

### 8.9.3.7 p\_eparminus

```
double* FitOutput::p_eparminus
```

Errors.

### 8.9.3.8 p\_eparparab

```
double* FitOutput::p_eparparab
```

Errors.

### 8.9.3.9 p\_eparplus

```
double* FitOutput::p_eparplus
```

Errors.

### 8.9.3.10 p\_par

```
double* FitOutput::p_par
```

Parameters.

### 8.9.3.11 p\_parName

```
std::string* FitOutput::p_parName
```

Name.

The documentation for this class was generated from the following files:

- ModelEvent/inc/[FitOutput.h](#)
- ModelEvent/src/[FitOutput.cxx](#)

## 8.10 GaussFunction Class Reference

```
#include <GaussFunction.h>
```

### Public Member Functions

- [GaussFunction \(\)](#)
- virtual [~GaussFunction \(\)](#)
- double [Get\\_Norm \(\)](#)
- double [Get\\_Mean \(\)](#)
- double [Get\\_Sigma \(\)](#)
- void [Set\\_Norm \(const double &Norm\)](#)
- void [Set\\_Mean \(const double &Mean\)](#)
- void [Set\\_Sigma \(const double &Sigma\)](#)
- double [operator\(\) \(double \\*x, double \\*par\)](#)

### 8.10.1 Constructor & Destructor Documentation

#### 8.10.1.1 GaussFunction()

```
GaussFunction::GaussFunction ()
```

Constructor

#### 8.10.1.2 ~GaussFunction()

```
GaussFunction::~GaussFunction () [virtual]
```

## 8.10.2 Member Function Documentation

### 8.10.2.1 Get\_Mean()

```
double GaussFunction::Get_Mean ()
```

### 8.10.2.2 Get\_Norm()

```
double GaussFunction::Get_Norm ()
```

### 8.10.2.3 Get\_Sigma()

```
double GaussFunction::Get_Sigma ()
```

### 8.10.2.4 operator()()

```
double GaussFunction::operator() (
    double * x,
    double * par)
```

### 8.10.2.5 Set\_Mean()

```
void GaussFunction::Set_Mean (
    const double & Mean)
```

### 8.10.2.6 Set\_Norm()

```
void GaussFunction::Set_Norm (
    const double & Norm)
```

### 8.10.2.7 Set\_Sigma()

```
void GaussFunction::Set_Sigma (
    const double & Sigma)
```

The documentation for this class was generated from the following files:

- Utilities/inc/[GaussFunction.h](#)
- Utilities/src/[GaussFunction.cxx](#)

## 8.11 GlobalChi2\_4 Struct Reference

### Public Member Functions

- [GlobalChi2\\_4](#) (ROOT::Math::IMultiGenFunction &f1, ROOT::Math::IMultiGenFunction &f2, ROOT::Math::IMultiGenFunction &f3, ROOT::Math::IMultiGenFunction &f4)
- double [operator\(\)](#) (const double \*par) const

### Public Attributes

- const ROOT::Math::IMultiGenFunction \* [fChi2\\_1](#)
- const ROOT::Math::IMultiGenFunction \* [fChi2\\_2](#)
- const ROOT::Math::IMultiGenFunction \* [fChi2\\_3](#)
- const ROOT::Math::IMultiGenFunction \* [fChi2\\_4](#)

### 8.11.1 Constructor & Destructor Documentation

#### 8.11.1.1 GlobalChi2\_4()

```
GlobalChi2_4::GlobalChi2_4 (
    ROOT::Math::IMultiGenFunction & f1,
    ROOT::Math::IMultiGenFunction & f2,
    ROOT::Math::IMultiGenFunction & f3,
    ROOT::Math::IMultiGenFunction & f4) [inline]
```

### 8.11.2 Member Function Documentation

#### 8.11.2.1 operator()()

```
double GlobalChi2_4::operator() (
    const double * par) const [inline]
```

### 8.11.3 Member Data Documentation

#### 8.11.3.1 fChi2\_1

```
const ROOT::Math::IMultiGenFunction* GlobalChi2_4::fChi2_1
```

#### 8.11.3.2 fChi2\_2

```
const ROOT::Math::IMultiGenFunction* GlobalChi2_4::fChi2_2
```

#### 8.11.3.3 fChi2\_3

```
const ROOT::Math::IMultiGenFunction* GlobalChi2_4::fChi2_3
```

### 8.11.3.4 fChi2\_4

```
const ROOT::Math::IMultiGenFunction* GlobalChi2_4::fChi2_4
```

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

- Reconstruction/src/CombinedFit.cxx

## 8.12 Reconstruction::LUT Class Reference

```
#include <LUTs.h>
```

### Public Member Functions

- [LUT](#) (const int &transDiffCoeffB, const int &transDiffCoeffnoB, const int &peakingTime)
- [LUT](#) (const std::string &file)
- virtual [~LUT](#) ()
- float [getRatio](#) (const int &Dt, const int &RC, const float &drift, const float &impact, const float &angle)

### 8.12.1 Constructor & Destructor Documentation

#### 8.12.1.1 LUT() [1/2]

```
Reconstruction::LUT::LUT (
    const int & transDiffCoeffB,
    const int & transDiffCoeffnoB,
    const int & peakingTime)
```

#### 8.12.1.2 LUT() [2/2]

```
Reconstruction::LUT::LUT (
    const std::string & file)
```

#### 8.12.1.3 ~LUT()

```
Reconstruction::LUT::~LUT () [virtual]
```

## 8.12.2 Member Function Documentation

### 8.12.2.1 getRatio()

```
float Reconstruction::LUT::getRatio (
    const int & Dt,
    const int & RC,
    const float & drift,
    const float & impact,
    const float & angle)
```

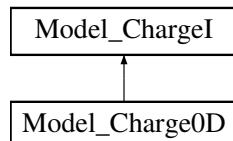
The documentation for this class was generated from the following files:

- Reconstruction/inc/LUTs.h
- Reconstruction/src/LUTs.cxx

## 8.13 Model\_Charge0D Class Reference

```
#include <Model_Charge0D.h>
```

Inheritance diagram for Model\_Charge0D:



### Public Member Functions

- [Model\\_Charge0D \(\)](#)
- virtual [~Model\\_Charge0D \(\)](#)
- double [Get\\_Qprim \(\) const](#)
- void [Set\\_Qprim \(const double &Qprim\)](#)
- double [Get\\_Qpad \(const double &Time, const double &Time0, const double &XTrue, const double &YTrue, const double &XL, const double &XH, const double &YL, const double &YH\) const](#)
- void [WriteOut \(\) const](#)
- void [Set\\_Ion \(const int &Dolons, const double &Tion, const double &RTrue, const double &RFactor\)](#)

### Public Member Functions inherited from [Model\\_ChargeI](#)

- [Model\\_ChargeI \(\)](#)
- virtual [~Model\\_ChargeI \(\)](#)
- double [Get\\_RC \(\) const](#)
- double [Get\\_Width \(\) const](#)
- double [Get\\_Gain \(\) const](#)
- void [Set\\_RC \(const double &RC\)](#)
- void [Set\\_Width \(const double &Width\)](#)
- void [Set\\_Gain \(const double &Gain\)](#)
- void [Set\\_WidthFromDriftDistance \(const double &DriftDistance\\_cm\)](#)

### Protected Member Functions

- void [SetSecondaries \(\)](#)

### Protected Member Functions inherited from [Model\\_Charge1](#)

- void [WidthFromDriftDistance \(const double &DriftDistance\\_cm\)](#)

### Protected Attributes

- double [m\\_Qprim](#)
- int [m\\_Dolons](#)
- double [m\\_Tion](#)
- double [m\\_RTrue](#)
- double [m\\_RFactor](#)

### Protected Attributes inherited from [Model\\_Charge1](#)

- double [m\\_RC](#)
- double [m\\_Width](#)
- double [m\\_Gain](#)
- double [m\\_2RCinv](#)
- double [m\\_Width2](#)
- double [m\\_NormQPad](#)

## 8.13.1 Constructor & Destructor Documentation

### 8.13.1.1 [Model\\_Charge0D\(\)](#)

```
Model_Charge0D::Model_Charge0D ()
```

Constructor

### 8.13.1.2 [~Model\\_Charge0D\(\)](#)

```
Model_Charge0D::~Model_Charge0D () [virtual]
```

## 8.13.2 Member Function Documentation

### 8.13.2.1 [Get\\_Qpad\(\)](#)

```
double Model_Charge0D::Get_Qpad (
    const double & Time,
    const double & Time0,
    const double & XTrue,
    const double & YTrue,
    const double & XL,
    const double & XH,
    const double & YL,
    const double & YH) const [virtual]
```

Implements [Model\\_Charge1](#).

### 8.13.2.2 Get\_Qprim()

```
double Model_Charge0D::Get_Qprim () const
```

### 8.13.2.3 Set\_Ion()

```
void Model_Charge0D::Set_Ion (
    const int & DoIons,
    const double & Tion,
    const double & RTrue,
    const double & RFactor)
```

### 8.13.2.4 Set\_Qprim()

```
void Model_Charge0D::Set_Qprim (
    const double & Qprim)
```

### 8.13.2.5 SetSecondaries()

```
void Model_Charge0D::SetSecondaries () [protected], [virtual]
```

Implements [Model\\_Charge1](#).

### 8.13.2.6 WriteOut()

```
void Model_Charge0D::WriteOut () const [virtual]
```

Implements [Model\\_Charge1](#).

## 8.13.3 Member Data Documentation

### 8.13.3.1 m\_Dolons

```
int Model_Charge0D::m_Dolons [protected]
```

### 8.13.3.2 m\_Qprim

```
double Model_Charge0D::m_Qprim [protected]
```

### 8.13.3.3 m\_RFactor

```
double Model_Charge0D::m_RFactor [protected]
```

### 8.13.3.4 m\_RTrue

```
double Model_Charge0D::m_RTrue [protected]
```

### 8.13.3.5 m\_Tion

```
double Model_Charge0D::m_Tion [protected]
```

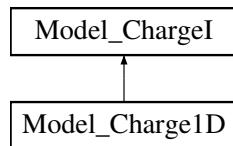
The documentation for this class was generated from the following files:

- ModelSignal/inc/[Model\\_Charge0D.h](#)
- ModelSignal/src/[Model\\_Charge0D.hxx](#)

## 8.14 Model\_Charge1D Class Reference

```
#include <Model_Charge1D.h>
```

Inheritance diagram for Model\_Charge1D:



### Public Member Functions

- [Model\\_Charge1D \(\)](#)
- virtual [~Model\\_Charge1D \(\)](#)
- double [Get\\_Lambda \(\) const](#)
- void [Set\\_Lambda \(const double &Lambda\)](#)
- double [Get\\_Qpad \(const double &Time, const double &Time0, const double &XTrue, const double &YTrue, const double &XL, const double &XH, const double &YL, const double &YH\) const](#)
- void [WriteOut \(\) const](#)

### Public Member Functions inherited from [Model\\_ChargeI](#)

- [Model\\_ChargeI \(\)](#)
- virtual [~Model\\_ChargeI \(\)](#)
- double [Get\\_RC \(\) const](#)
- double [Get\\_Width \(\) const](#)
- double [Get\\_Gain \(\) const](#)
- void [Set\\_RC \(const double &RC\)](#)
- void [Set\\_Width \(const double &Width\)](#)
- void [Set\\_Gain \(const double &Gain\)](#)
- void [Set\\_WidthFromDriftDistance \(const double &DriftDistance\\_cm\)](#)

### Protected Member Functions

- void [SetSecondaries \(\)](#)

### Protected Member Functions inherited from [Model\\_Charge1](#)

- void [WidthFromDriftDistance \(const double &DriftDistance\\_cm\)](#)

### Protected Attributes

- double [m\\_Lambda](#)

### Protected Attributes inherited from [Model\\_Charge1](#)

- double [m\\_RC](#)
- double [m\\_Width](#)
- double [m\\_Gain](#)
- double [m\\_2RCinv](#)
- double [m\\_Width2](#)
- double [m\\_NormQPad](#)

## 8.14.1 Constructor & Destructor Documentation

### 8.14.1.1 [Model\\_Charge1D\(\)](#)

```
Model_Charge1D::Model_Charge1D ()
```

Constructor

### 8.14.1.2 [~Model\\_Charge1D\(\)](#)

```
Model_Charge1D::~Model_Charge1D () [virtual]
```

## 8.14.2 Member Function Documentation

### 8.14.2.1 [Get\\_Lambda\(\)](#)

```
double Model_Charge1D::Get_Lambda () const
```

#### 8.14.2.2 Get\_Qpad()

```
double Model_Charge1D::Get_Qpad (
    const double & Time,
    const double & Time0,
    const double & XTrue,
    const double & YTrue,
    const double & XL,
    const double & XH,
    const double & YL,
    const double & YH) const [virtual]
```

Implements [Model\\_Charge1](#).

#### 8.14.2.3 Set\_Lambda()

```
void Model_Charge1D::Set_Lambda (
    const double & Lambda)
```

#### 8.14.2.4 SetSecondaries()

```
void Model_Charge1D::SetSecondaries () [protected], [virtual]
```

Implements [Model\\_Charge1](#).

#### 8.14.2.5 WriteOut()

```
void Model_Charge1D::WriteOut () const [virtual]
```

Implements [Model\\_Charge1](#).

### 8.14.3 Member Data Documentation

#### 8.14.3.1 m\_Lambda

```
double Model_Charge1D::m_Lambda [protected]
```

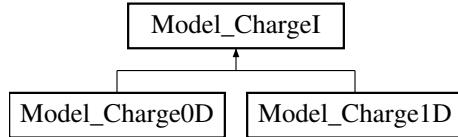
The documentation for this class was generated from the following files:

- ModelSignal/inc/[Model\\_Charge1D.h](#)
- ModelSignal/src/[Model\\_Charge1D.cxx](#)

## 8.15 Model\_ChargeI Class Reference

```
#include <Model_ChargeI.h>
```

Inheritance diagram for Model\_ChargeI:



### Public Member Functions

- [Model\\_ChargeI \(\)](#)
- [virtual ~Model\\_ChargeI \(\)](#)
- [double Get\\_RC \(\) const](#)
- [double Get\\_Width \(\) const](#)
- [double Get\\_Gain \(\) const](#)
- [void Set\\_RC \(const double &RC\)](#)
- [void Set\\_Width \(const double &Width\)](#)
- [void Set\\_Gain \(const double &Gain\)](#)
- [void Set\\_WidthFromDriftDistance \(const double &DriftDistance\\_cm\)](#)
- [virtual double Get\\_Qpad \(const double &Time, const double &Time0, const double &XTrue, const double &YTrue, const double &XL, const double &XH, const double &YL, const double &YH\) const =0](#)
- [virtual void WriteOut \(\) const =0](#)

### Protected Member Functions

- [void WidthFromDriftDistance \(const double &DriftDistance\\_cm\)](#)
- [virtual void SetSecondaries \(\)=0](#)

### Protected Attributes

- [double m\\_RC](#)
- [double m\\_Width](#)
- [double m\\_Gain](#)
- [double m\\_2RCinv](#)
- [double m\\_Width2](#)
- [double m\\_NormQPad](#)

## 8.15.1 Constructor & Destructor Documentation

### 8.15.1.1 Model\_ChargeI()

```
Model_ChargeI::Model_ChargeI ()
```

Constructor

### 8.15.1.2 ~Model\_ChargeI()

```
Model_ChargeI::~Model_ChargeI () [virtual]
```

## 8.15.2 Member Function Documentation

### 8.15.2.1 Get\_Gain()

```
double Model_ChargeI::Get_Gain () const
```

### 8.15.2.2 Get\_Qpad()

```
virtual double Model_ChargeI::Get_Qpad (
    const double & Time,
    const double & Time0,
    const double & XTrue,
    const double & YTrue,
    const double & XL,
    const double & XH,
    const double & YL,
    const double & YH) const [pure virtual]
```

Implemented in [Model\\_Charge0D](#), and [Model\\_Charge1D](#).

### 8.15.2.3 Get\_RC()

```
double Model_ChargeI::Get_RC () const
```

### 8.15.2.4 Get\_Width()

```
double Model_ChargeI::Get_Width () const
```

### 8.15.2.5 Set\_Gain()

```
void Model_ChargeI::Set_Gain (
    const double & Gain)
```

### 8.15.2.6 Set\_RC()

```
void Model_ChargeI::Set_RC (
    const double & RC)
```

### 8.15.2.7 Set\_Width()

```
void Model_ChargeI::Set_Width (
    const double & Width)
```

### 8.15.2.8 Set\_WidthFromDriftDistance()

```
void Model_ChargeI::Set_WidthFromDriftDistance (
    const double & DriftDistance_cm)
```

### 8.15.2.9 SetSecondaries()

```
virtual void Model_ChargeI::SetSecondaries () [protected], [pure virtual]
```

Implemented in [Model\\_Charge0D](#), and [Model\\_Charge1D](#).

### 8.15.2.10 WidthFromDriftDistance()

```
void Model_ChargeI::WidthFromDriftDistance (
    const double & DriftDistance_cm) [protected]
```

### 8.15.2.11 WriteOut()

```
virtual void Model_ChargeI::WriteOut () const [pure virtual]
```

Implemented in [Model\\_Charge0D](#), and [Model\\_Charge1D](#).

## 8.15.3 Member Data Documentation

### 8.15.3.1 m\_2RCinv

```
double Model_ChargeI::m_2RCinv [protected]
```

### 8.15.3.2 m\_Gain

```
double Model_ChargeI::m_Gain [protected]
```

### 8.15.3.3 m\_NormQPad

```
double Model_ChargeI::m_NormQPad [protected]
```

### 8.15.3.4 m\_RC

```
double Model_ChargeI::m_RC [protected]
```

### 8.15.3.5 m\_Width

```
double Model_ChargeI::m_Width [protected]
```

### 8.15.3.6 m\_Width2

```
double Model_ChargeI::m_Width2 [protected]
```

The documentation for this class was generated from the following files:

- ModelSignal/inc/[Model\\_ChargeI.h](#)
- ModelSignal/src/[Model\\_ChargeI.cxx](#)

## 8.16 Model\_Electronics Class Reference

```
#include <Model_Electronics.h>
```

### Public Member Functions

- [Model\\_Electronics \(\)](#)
- virtual [~Model\\_Electronics \(\)](#)
- double [Get\\_QValue \(\) const](#)
- double [Get\\_ws \(\) const](#)
- double [Get\\_Amplitude \(\) const](#)
- void [Set\\_TimeShape \(const double &QValue, const double &ws\)](#)
- void [Set\\_Amplitude \(const double &Amplitude\)](#)
- double [Response\\_Base \(const double &Time\) const](#)
- double [Derive\\_Response\\_Base \(const double &Time\) const](#)
- void [WriteOut \(\) const](#)

### 8.16.1 Constructor & Destructor Documentation

#### 8.16.1.1 Model\_Electronics()

```
Model_Electronics::Model_Electronics ()
```

Constructor

#### 8.16.1.2 ~Model\_Electronics()

```
Model_Electronics::~Model_Electronics () [virtual]
```

### 8.16.2 Member Function Documentation

#### 8.16.2.1 Derive\_Response\_Base()

```
double Model_Electronics::Derive_Response_Base (
    const double & Time) const
```

### 8.16.2.2 Get\_Amplitude()

```
double Model_Electronics::Get_Amplitude () const
```

### 8.16.2.3 Get\_QValue()

```
double Model_Electronics::Get_QValue () const
```

### 8.16.2.4 Get\_ws()

```
double Model_Electronics::Get_ws () const
```

### 8.16.2.5 Response\_Base()

```
double Model_Electronics::Response_Base (
    const double & Time) const
```

### 8.16.2.6 Set\_Amplitude()

```
void Model_Electronics::Set_Amplitude (
    const double & Amplitude)
```

### 8.16.2.7 Set\_TimeShape()

```
void Model_Electronics::Set_TimeShape (
    const double & QValue,
    const double & ws)
```

### 8.16.2.8 WriteOut()

```
void Model_Electronics::WriteOut () const
```

The documentation for this class was generated from the following files:

- ModelSignal/inc/[Model\\_Electronics.h](#)
- ModelSignal/src/[Model\\_Electronics.cxx](#)

## 8.17 Model\_ReadOutGeometry Class Reference

```
#include <Model_ReadOutGeometry.h>
```

## Public Member Functions

- `Model_ReadOutGeometry ()`
- `virtual ~Model_ReadOutGeometry ()`
- `double Get_LX () const`
- `double Get LY () const`
- `double Get_Xpad_min () const`
- `double Get_Ypad_min () const`
- `int Get_Nx () const`
- `int Get_Ny () const`
- `void SwapGeometry ()`
- `void SquareGeometry ()`
- `void Set_LX (const double &LX)`
- `void Set LY (const double &LY)`
- `void Set_Xpad_min (const double &Xpad_min)`
- `void Set_Ypad_min (const double &Ypad_min)`
- `void Set_Nx (const int &Nx)`
- `void Set_Ny (const int &Ny)`
- `double Get_XcPad (const int &iX, const int &iY, const int &ModuleNber) const`
- `double Get_YcPad (const int &iX, const int &iY, const int &ModuleNber) const`
- `double Get_XLPad (const int &iX, const int &iY, const int &ModuleNber) const`
- `double Get_YLPad (const int &iX, const int &iY, const int &ModuleNber) const`
- `double Get_XHPad (const int &iX, const int &iY, const int &ModuleNber) const`
- `double Get_YHPad (const int &iX, const int &iY, const int &ModuleNber) const`
- `void GetPadEdges (const int &iX, const int &iY, const int &ModuleNber, double &XL, double &YL, double &YH) const`
- `void WriteOut () const`

### 8.17.1 Constructor & Destructor Documentation

#### 8.17.1.1 Model\_ReadOutGeometry()

```
Model_ReadOutGeometry::Model_ReadOutGeometry ()
```

Constructor

#### 8.17.1.2 ~Model\_ReadOutGeometry()

```
Model_ReadOutGeometry::~Model_ReadOutGeometry () [virtual]
```

### 8.17.2 Member Function Documentation

#### 8.17.2.1 Get\_LX()

```
double Model_ReadOutGeometry::Get_LX () const
```

### 8.17.2.2 Get\_LY()

```
double Model_ReadOutGeometry::Get_LY () const
```

### 8.17.2.3 Get\_Nx()

```
int Model_ReadOutGeometry::Get_Nx () const
```

### 8.17.2.4 Get\_Ny()

```
int Model_ReadOutGeometry::Get_Ny () const
```

### 8.17.2.5 Get\_XcPad()

```
double Model_ReadOutGeometry::Get_XcPad (
    const int & iX,
    const int & iY,
    const int & ModuleNber) const
```

### 8.17.2.6 Get\_XHPad()

```
double Model_ReadOutGeometry::Get_XHPad (
    const int & iX,
    const int & iY,
    const int & ModuleNber) const
```

### 8.17.2.7 Get\_XLPad()

```
double Model_ReadOutGeometry::Get_XLPad (
    const int & iX,
    const int & iY,
    const int & ModuleNber) const
```

### 8.17.2.8 Get\_Xpad\_min()

```
double Model_ReadOutGeometry::Get_Xpad_min () const
```

### 8.17.2.9 Get\_YcPad()

```
double Model_ReadOutGeometry::Get_YcPad (
    const int & iX,
    const int & iY,
    const int & ModuleNber) const
```

**8.17.2.10 Get\_YHPad()**

```
double Model_ReadOutGeometry::Get_YHPad (
    const int & iX,
    const int & iY,
    const int & ModuleNber) const
```

**8.17.2.11 Get\_YLPad()**

```
double Model_ReadOutGeometry::Get_YLPad (
    const int & iX,
    const int & iY,
    const int & ModuleNber) const
```

**8.17.2.12 Get\_Ypad\_min()**

```
double Model_ReadOutGeometry::Get_Ypad_min () const
```

**8.17.2.13 GetPadEdges()**

```
void Model_ReadOutGeometry::GetPadEdges (
    const int & iX,
    const int & iY,
    const int & ModuleNber,
    double & XL,
    double & XH,
    double & YL,
    double & YH) const
```

**8.17.2.14 Set\_LX()**

```
void Model_ReadOutGeometry::Set_LX (
    const double & LX)
```

**8.17.2.15 Set LY()**

```
void Model_ReadOutGeometry::Set_LY (
    const double & LY)
```

**8.17.2.16 Set\_Nx()**

```
void Model_ReadOutGeometry::Set_Nx (
    const int & Nx)
```

### 8.17.2.17 Set\_Ny()

```
void Model_ReadOutGeometry::Set_Ny (
    const int & Ny)
```

### 8.17.2.18 Set\_Xpad\_min()

```
void Model_ReadOutGeometry::Set_Xpad_min (
    const double & Xpad_min)
```

### 8.17.2.19 Set\_Ypad\_min()

```
void Model_ReadOutGeometry::Set_Ypad_min (
    const double & Ypad_min)
```

### 8.17.2.20 SquareGeometry()

```
void Model_ReadOutGeometry::SquareGeometry ()
```

### 8.17.2.21 SwapGeometry()

```
void Model_ReadOutGeometry::SwapGeometry ()
```

### 8.17.2.22 WriteOut()

```
void Model_ReadOutGeometry::WriteOut () const
```

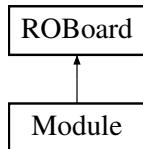
The documentation for this class was generated from the following files:

- ModelSignal/inc/[Model\\_ReadOutGeometry.h](#)
- ModelSignal/src/[Model\\_ReadOutGeometry.cxx](#)

## 8.18 Module Class Reference

```
#include <Module.h>
```

Inheritance diagram for Module:



## Public Member Functions

- `Module (const int &EventNber, const int &EntryNber, const int &ModuleNber, Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI)`
- `virtual ~Module ()`
- `int Get_EventNber () const`
- `int Get_EntryNber () const`
- `int Get_ModuleNber () const`
- `void SmallDump () const`
- `void WriteOut () const`
- `int IsValid () const`
- `void Validate ()`
- `void Invalidate ()`
- `virtual void Add_Pad (Pad *pPad)`
- `void Clear_Clusters ()`
- `void Add_Cluster (Cluster *pCluster)`
- `std::vector< Cluster * > GiveMe_Clusters_ForThisModule ()`
- `int Get_NberOfCluster () const`
- `Cluster * Get_Cluster (const int &Index1D)`
- `void ReplaceClusters (std::vector< Cluster * > &V_Cluster)`
- `Cluster * Get_Cluster_Copy (Cluster *pCluster)`
- `const Track * GiveMe_ModuleTrack ()`
- `void Set_ModuleTrack (Track *pTrack)`
- `Track * GiveMe_AnUnfittedTrack ()`

## Public Member Functions inherited from ROBoard

- `ROBoard (Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI)`
- `virtual ~ROBoard ()`
- `const Model_ReadOutGeometry * Get_Model_ReadOutGeometry ()`
- `const Model_Electronics * Get_Model_Electronics ()`
- `const Model_ChargeI * Get_Model_ChargeI ()`
- `int Get_NberOfPads () const`
- `Pad * Get_Pad (const int &Index1D)`
- `int Get_IsThisPadExisting (const int &iX, const int &iY) const`
- `Pad * Get_Pad (const int &iX, const int &iY)`
- `Pad * Get_Pad (Pad *pPad)`

## Additional Inherited Members

## Protected Member Functions inherited from ROBoard

- `int GetLinearIndex (const int &iX, const int &iY) const`
- `void Ini_Models (Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI)`

## Protected Attributes inherited from ROBoard

- short int `m_Nx`
- short int `m_Ny`
- std::vector< int > `V_ExisFlag2D`
- std::vector< `Pad` \* > `V_Pad`
- `Model_ReadOutGeometry` \* `p_Model_ReadOutGeometry`
- `Model_Electronics` \* `p_Model_Electronics`
- `Model_ChargeI` \* `p_Model_ChargeI`

## 8.18.1 Constructor & Destructor Documentation

### 8.18.1.1 Module()

```
Module::Module (
    const int & EventNber,
    const int & EntryNber,
    const int & ModuleNber,
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_ChargeI * pModel_ChargeI)
```

Constructor

### 8.18.1.2 ~Module()

```
Module::~Module () [virtual]
```

## 8.18.2 Member Function Documentation

### 8.18.2.1 Add\_Cluster()

```
void Module::Add_Cluster (
    Cluster * pCluster)
```

### 8.18.2.2 Add\_Pad()

```
void Module::Add_Pad (
    Pad * pPad) [virtual]
```

Reimplemented from [ROBoard](#).

### 8.18.2.3 Clear\_Clusters()

```
void Module::Clear_Clusters ()
```

**8.18.2.4 Get\_Cluster()**

```
Cluster * Module::Get_Cluster (
    const int & IndexID)
```

**8.18.2.5 Get\_Cluster\_Copy()**

```
Cluster * Module::Get_Cluster_Copy (
    Cluster * pCluster)
```

**8.18.2.6 Get\_EntryNber()**

```
int Module::Get_EntryNber () const
```

**8.18.2.7 Get\_EventNber()**

```
int Module::Get_EventNber () const
```

**8.18.2.8 Get\_ModuleNber()**

```
int Module::Get_ModuleNber () const
```

**8.18.2.9 Get\_NberOfCluster()**

```
int Module::Get_NberOfCluster () const
```

**8.18.2.10 GiveMe\_AnUnfittedTrack()**

```
Track * Module::GiveMe_AnUnfittedTrack ()
```

**8.18.2.11 GiveMe\_Clusters\_ForThisModule()**

```
std::vector< Cluster * > Module::GiveMe_Clusters_ForThisModule ()
```

**8.18.2.12 GiveMe\_ModuleTrack()**

```
const Track * Module::GiveMe_ModuleTrack ()
```

**8.18.2.13 Invalidate()**

```
void Module::Invalidate ()
```

**8.18.2.14 IsValid()**

```
int Module::IsValid () const
```

**8.18.2.15 ReplaceClusters()**

```
void Module::ReplaceClusters (
    std::vector< Cluster * > & V_Cluster)
```

**8.18.2.16 Set\_ModuleTrack()**

```
void Module::Set_ModuleTrack (
    Track * pTrack)
```

**8.18.2.17 SmallDump()**

```
void Module::SmallDump () const
```

**8.18.2.18 Validate()**

```
void Module::Validate ()
```

**8.18.2.19 WriteOut()**

```
void Module::WriteOut () const
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[Module.h](#)
- ModelEvent/src/[Module.cxx](#)

## 8.19 Pad Class Reference

```
#include <Pad.h>
```

## Public Member Functions

- `Pad (Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI, std::string PadName, const int &EventNber, const int &EntryNber, const int &ModuleNber, const int &iX, const int &iY, const double &XL, const double &XH, const double &YL, const double &YH)`  
*Construct a `Pad` with explicit edges.*
- `Pad (Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI, std::string PadName, const int &EventNber, const int &EntryNber, const int &ModuleNber, const int &iX, const int &iY)`  
*Construct a `Pad` using geometry lookup for edges.*
- `virtual ~Pad ()`  
*Virtual destructor.*
- `int Get_EventNber () const`
- `int Get_EntryNber () const`
- `int Get_ModuleNber () const`
- `int Get_iX () const`
- `int Get_iY () const`
- `void WriteOut () const`
- `int IsValid () const`
- `void Validate ()`
- `void Invalidate ()`
- `const Model_ReadOutGeometry * Get_Model_ReadOutGeometry () const`
- `const Model_Electronics * Get_Model_Electronics () const`
- `const Model_ChargeI * Get_Model_ChargeI () const`
- `std::string Get_PadName () const`
- `double Get_XPad () const`
- `double Get_YPad () const`
- `double Get_XL () const`
- `double Get_XH () const`
- `double Get_YL () const`
- `double Get_YH () const`
- `double Get_LX () const`
- `double Get LY () const`
- `double Get_AMax () const`
- `double Get_TMax () const`
- `void Set_AMax (const double &AMax)`
- `void Set_TMax (const double &TMax)`
- `void Set_Data_2Use (const int &iOpt)`
- `void Clear_ADC ()`
- `void Set_ADC (const int &iTimeb, const int &ADC)`
- `void WF_DoClosure ()`
- `std::vector< float > Get_vADC () const`
- `double Get_AMax_FromSet () const`
- `double Get_TMax_FromSet () const`
- `double Get_AMax_WF () const`
- `double Get_TMax_WF () const`
- `double Get_AMax_WF_01 () const`
- `double Get_TMax_WF_01 () const`
- `double Get_AMax_FIT () const`
- `double Get_TMax_FIT () const`
- `int Get_FIT_Status () const`
- `double Get_FIT_A0P () const`
- `double Get_FIT_A0M () const`
- `double Get_FIT_X0 () const`

- double Get\_FIT\_Y0 () const
- double Get\_FIT\_Xmin () const
- double Get\_FIT\_Xmax () const
- double Get\_AMax\_True () const
- double Get\_TMax\_True () const
- void Set\_AMax\_True (const double &AMax)
- void Set\_TMax\_True (const double &TMax)
- void SetSignalModel (const double &Time0, const double &XTrue, const double &YTrue)
- double Get\_Time0 () const
- double Get\_XTrue () const
- double Get\_YTrue () const
- double Get\_Qpad (const double &Time)
- double Get\_APad (const double &Time)

## 8.19.1 Constructor & Destructor Documentation

### 8.19.1.1 Pad() [1/2]

```
Pad::Pad (
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_ChargeI * pModel_ChargeI,
    std::string PadName,
    const int & EventNber,
    const int & EntryNber,
    const int & ModuleNber,
    const int & iX,
    const int & iY,
    const double & XL,
    const double & XH,
    const double & YL,
    const double & YH)
```

Construct a **Pad** with explicit edges.

### 8.19.1.2 Pad() [2/2]

```
Pad::Pad (
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_ChargeI * pModel_ChargeI,
    std::string PadName,
    const int & EventNber,
    const int & EntryNber,
    const int & ModuleNber,
    const int & iX,
    const int & iY)
```

Construct a **Pad** using geometry lookup for edges.

### 8.19.1.3 ~Pad()

```
Pad::~Pad () [virtual]
```

Virtual destructor.

## 8.19.2 Member Function Documentation

### 8.19.2.1 Clear\_ADC()

```
void Pad::Clear_ADC ()
```

### 8.19.2.2 Get\_AMax()

```
double Pad::Get_AMax () const
```

### 8.19.2.3 Get\_AMax\_FIT()

```
double Pad::Get_AMax_FIT () const
```

### 8.19.2.4 Get\_AMax\_FromSet()

```
double Pad::Get_AMax_FromSet () const
```

### 8.19.2.5 Get\_AMax\_True()

```
double Pad::Get_AMax_True () const
```

### 8.19.2.6 Get\_AMax\_WF()

```
double Pad::Get_AMax_WF () const
```

### 8.19.2.7 Get\_AMax\_WF\_01()

```
double Pad::Get_AMax_WF_01 () const
```

### 8.19.2.8 Get\_APad()

```
double Pad::Get_APad (
    const double & Time)
```

**8.19.2.9 Get\_EntryNber()**

```
int Pad::Get_EntryNber () const
```

**8.19.2.10 Get\_EventNber()**

```
int Pad::Get_EventNber () const
```

**8.19.2.11 Get\_FIT\_A0M()**

```
double Pad::Get_FIT_A0M () const
```

**8.19.2.12 Get\_FIT\_A0P()**

```
double Pad::Get_FIT_A0P () const
```

**8.19.2.13 Get\_FIT\_Status()**

```
int Pad::Get_FIT_Status () const
```

**8.19.2.14 Get\_FIT\_X0()**

```
double Pad::Get_FIT_X0 () const
```

**8.19.2.15 Get\_FIT\_Xmax()**

```
double Pad::Get_FIT_Xmax () const
```

**8.19.2.16 Get\_FIT\_Xmin()**

```
double Pad::Get_FIT_Xmin () const
```

**8.19.2.17 Get\_FIT\_Y0()**

```
double Pad::Get_FIT_Y0 () const
```

**8.19.2.18 Get\_iX()**

```
int Pad::Get_iX () const
```

**8.19.2.19 Get\_iY()**

```
int Pad::Get_iY () const
```

**8.19.2.20 Get\_LX()**

```
double Pad::Get_LX () const
```

**8.19.2.21 Get LY()**

```
double Pad::Get_LY () const
```

**8.19.2.22 Get\_Model\_ChargeI()**

```
const Model_ChargeI * Pad::Get_Model_ChargeI () const
```

**8.19.2.23 Get\_Model\_Electronics()**

```
const Model_Electronics * Pad::Get_Model_Electronics () const
```

**8.19.2.24 Get\_Model\_ReadOutGeometry()**

```
const Model_ReadOutGeometry * Pad::Get_Model_ReadOutGeometry () const
```

**8.19.2.25 Get\_ModuleNber()**

```
int Pad::Get_ModuleNber () const
```

**8.19.2.26 Get\_PadName()**

```
std::string Pad::Get_PadName () const
```

**8.19.2.27 Get\_Qpad()**

```
double Pad::Get_Qpad (
    const double & Time)
```

**8.19.2.28 Get\_Time0()**

```
double Pad::Get_Time0 () const
```

**8.19.2.29 Get\_TMax()**

```
double Pad::Get_TMax () const
```

**8.19.2.30 Get\_TMax\_FIT()**

```
double Pad::Get_TMax_FIT () const
```

**8.19.2.31 Get\_TMax\_FromSet()**

```
double Pad::Get_TMax_FromSet () const
```

**8.19.2.32 Get\_TMax\_True()**

```
double Pad::Get_TMax_True () const
```

**8.19.2.33 Get\_TMax\_WF()**

```
double Pad::Get_TMax_WF () const
```

**8.19.2.34 Get\_TMax\_WF\_01()**

```
double Pad::Get_TMax_WF_01 () const
```

**8.19.2.35 Get\_vADC()**

```
std::vector< float > Pad::Get_vADC () const
```

**8.19.2.36 Get\_XH()**

```
double Pad::Get_XH () const
```

**8.19.2.37 Get\_XL()**

```
double Pad::Get_XL () const
```

**8.19.2.38 Get\_XPad()**

```
double Pad::Get_XPad () const
```

**8.19.2.39 Get\_XTrue()**

```
double Pad::Get_XTrue () const
```

**8.19.2.40 Get\_YH()**

```
double Pad::Get_YH () const
```

**8.19.2.41 Get\_YL()**

```
double Pad::Get_YL () const
```

**8.19.2.42 Get\_YPad()**

```
double Pad::Get_YPad () const
```

**8.19.2.43 Get\_YTrue()**

```
double Pad::Get_YTrue () const
```

**8.19.2.44 Invalidate()**

```
void Pad::Invalidate ()
```

**8.19.2.45 IsValid()**

```
int Pad::IsValid () const
```

**8.19.2.46 Set\_ADC()**

```
void Pad::Set_ADC (
    const int & iTimeb,
    const int & ADC)
```

**8.19.2.47 Set\_AMax()**

```
void Pad::Set_AMax (
    const double & AMax)
```

**8.19.2.48 Set\_AMax\_True()**

```
void Pad::Set_AMax_True (
    const double & AMax)
```

**8.19.2.49 Set\_Data\_2Use()**

```
void Pad::Set_Data_2Use (
    const int & iOpt)
```

**8.19.2.50 Set\_TMax()**

```
void Pad::Set_TMax (
    const double & TMax)
```

**8.19.2.51 Set\_TMax\_True()**

```
void Pad::Set_TMax_True (
    const double & TMax)
```

**8.19.2.52 SetSignalModel()**

```
void Pad::SetSignalModel (
    const double & Time0,
    const double & XTrue,
    const double & YTrue)
```

**8.19.2.53 Validate()**

```
void Pad::Validate ()
```

**8.19.2.54 WF\_DoClosure()**

```
void Pad::WF_DoClosure ()
```

**8.19.2.55 WriteOut()**

```
void Pad::WriteOut () const
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[Pad.h](#)
- ModelEvent/src/[Pad.cxx](#)

## 8.20 ParabolaFunction Class Reference

```
#include <ParabolaFunction.h>
```

## Public Member Functions

- `ParabolaFunction ()`
- `virtual ~ParabolaFunction ()`
- `double Get_A0 ()`
- `double Get_X0 ()`
- `double Get_Y0 ()`
- `void Set_A0 (const double &A0)`
- `void Set_X0 (const double &X0)`
- `void Set_Y0 (const double &Y0)`
- `double operator() (double *x, double *par)`

## 8.20.1 Constructor & Destructor Documentation

### 8.20.1.1 ParabolaFunction()

```
ParabolaFunction::ParabolaFunction ()
```

Constructor

### 8.20.1.2 ~ParabolaFunction()

```
ParabolaFunction::~ParabolaFunction () [virtual]
```

## 8.20.2 Member Function Documentation

### 8.20.2.1 Get\_A0()

```
double ParabolaFunction::Get_A0 ()
```

### 8.20.2.2 Get\_X0()

```
double ParabolaFunction::Get_X0 ()
```

### 8.20.2.3 Get\_Y0()

```
double ParabolaFunction::Get_Y0 ()
```

### 8.20.2.4 operator()()

```
double ParabolaFunction::operator() (
    double * x,
    double * par)
```

### 8.20.2.5 Set\_A0()

```
void ParabolaFunction::Set_A0 (
    const double & A0)
```

### 8.20.2.6 Set\_X0()

```
void ParabolaFunction::Set_X0 (
    const double & X0)
```

### 8.20.2.7 Set\_Y0()

```
void ParabolaFunction::Set_Y0 (
    const double & Y0)
```

The documentation for this class was generated from the following files:

- Utilities/inc/[ParabolaFunction.h](#)
- Utilities/src/[ParabolaFunction.cxx](#)

## 8.21 ParabolaFunctionNG Class Reference

```
#include <ParabolaFunctionNG.h>
```

### Public Member Functions

- [ParabolaFunctionNG \(\)](#)
- virtual [~ParabolaFunctionNG \(\)](#)
- [double Get\\_A0P \(\)](#)
- [double Get\\_A0M \(\)](#)
- [double Get\\_X0 \(\)](#)
- [double Get\\_Y0 \(\)](#)
- [void Set\\_A0P \(const double &A0P\)](#)
- [void Set\\_A0M \(const double &A0M\)](#)
- [void Set\\_X0 \(const double &X0\)](#)
- [void Set\\_Y0 \(const double &Y0\)](#)
- [double operator\(\) \(double \\*x, double \\*par\)](#)

### 8.21.1 Constructor & Destructor Documentation

#### 8.21.1.1 ParabolaFunctionNG()

```
ParabolaFunctionNG::ParabolaFunctionNG ()
```

Constructor

### 8.21.1.2 ~ParabolaFunctionNG()

```
ParabolaFunctionNG::~ParabolaFunctionNG () [virtual]
```

## 8.21.2 Member Function Documentation

### 8.21.2.1 Get\_A0M()

```
double ParabolaFunctionNG::Get_A0M ()
```

### 8.21.2.2 Get\_A0P()

```
double ParabolaFunctionNG::Get_A0P ()
```

### 8.21.2.3 Get\_X0()

```
double ParabolaFunctionNG::Get_X0 ()
```

### 8.21.2.4 Get\_Y0()

```
double ParabolaFunctionNG::Get_Y0 ()
```

### 8.21.2.5 operator()()

```
double ParabolaFunctionNG::operator() (
    double * x,
    double * par)
```

### 8.21.2.6 Set\_A0M()

```
void ParabolaFunctionNG::Set_A0M (
    const double & A0M)
```

### 8.21.2.7 Set\_A0P()

```
void ParabolaFunctionNG::Set_A0P (
    const double & A0P)
```

### 8.21.2.8 Set\_X0()

```
void ParabolaFunctionNG::Set_X0 (
    const double & X0)
```

### 8.21.2.9 Set\_Y0()

```
void ParabolaFunctionNG::Set_Y0 (
    const double & Y0)
```

The documentation for this class was generated from the following files:

- Utilities/inc/[ParabolaFunctionNG.h](#)
- Utilities/src/[ParabolaFunctionNG.cxx](#)

## 8.22 PRFParameters Class Reference

```
#include <PRFParameters.h>
```

### Public Member Functions

- **PRFParameters ()**  
*Default constructor.*
- virtual **~PRFParameters ()**  
*Virtual destructor.*
- **double Eval (const double &Xin)**  
*Evaluate the PRF at a given position.*
- **double operator() (double \*x, double \*par)**  
*Function-call operator used by ROOT TF1-style callbacks.*
- **void SetPRF (const double &Norm, const double &A2, const double &A4, const double &B2, const double &B4)**  
*Set PRF polynomial parameters.*

### 8.22.1 Constructor & Destructor Documentation

#### 8.22.1.1 PRFParameters()

```
PRFParameters::PRFParameters ()
```

Default constructor.

#### 8.22.1.2 ~PRFParameters()

```
PRFParameters::~PRFParameters () [virtual]
```

Virtual destructor.

### 8.22.2 Member Function Documentation

#### 8.22.2.1 Eval()

```
double PRFParameters::Eval (
    const double & Xin)
```

Evaluate the PRF at a given position.

#### Parameters

---

<i>Xin</i>	input position in cm
------------	----------------------

**Returns**

evaluated PRF value

**8.22.2.2 operator()()**

```
double PRFParameters::operator() (
    double * x,
    double * par)
```

Function-call operator used by ROOT TF1-style callbacks.

**8.22.2.3 SetPRF()**

```
void PRFParameters::SetPRF (
    const double & Norm,
    const double & A2,
    const double & A4,
    const double & B2,
    const double & B4)
```

Set PRF polynomial parameters.

**Parameters**

<i>Norm</i>	normalization
<i>A2</i>	coefficient A2
<i>A4</i>	coefficient A4
<i>B2</i>	coefficient B2
<i>B4</i>	coefficient B4

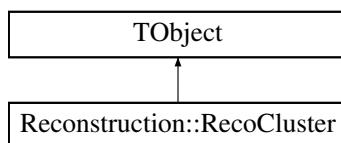
The documentation for this class was generated from the following files:

- Fitters/inc/[PRFParameters.h](#)
- Fitters/src/[PRFParameters.cxx](#)

**8.23 Reconstruction::RecoCluster Class Reference**

```
#include <dEdx.h>
```

Inheritance diagram for Reconstruction::RecoCluster:



## Public Member Functions

- virtual `~RecoCluster ()`
- `ClassDef (RecoCluster, 1)`

## Public Attributes

- `std::vector< RecoPad * > v_pads`
- float `length` = 0
- float `charge` = 0
- float `ADCmax_base` = 0
- float `ALead_base` = 0
- float `ALead_GCorr` = 0
- int `TLead` = 0
- float `dEdxWF` = 0
- int `NPads` = 0
- float `ratioCorr` = 0
- float `yCluster` = 0
- float `yWeight` = 0
- float `LUTrhoLead` = 0

### 8.23.1 Constructor & Destructor Documentation

#### 8.23.1.1 `~RecoCluster()`

```
Reconstruction::RecoCluster::~RecoCluster () [virtual]
```

### 8.23.2 Member Function Documentation

#### 8.23.2.1 `ClassDef()`

```
Reconstruction::RecoCluster::ClassDef (
    RecoCluster ,
    1 )
```

### 8.23.3 Member Data Documentation

#### 8.23.3.1 `ADCmax_base`

```
float Reconstruction::RecoCluster::ADCmax_base = 0
```

#### 8.23.3.2 `ALead_base`

```
float Reconstruction::RecoCluster::ALead_base = 0
```

**8.23.3.3 ALead\_GCorr**

```
float Reconstruction::RecoCluster::ALead_GCorr = 0
```

**8.23.3.4 charge**

```
float Reconstruction::RecoCluster::charge = 0
```

**8.23.3.5 dEdxWF**

```
float Reconstruction::RecoCluster::dEdxWF = 0
```

**8.23.3.6 length**

```
float Reconstruction::RecoCluster::length = 0
```

**8.23.3.7 LUTrhoLead**

```
float Reconstruction::RecoCluster::LUTrhoLead = 0
```

**8.23.3.8 NPads**

```
int Reconstruction::RecoCluster::NPads = 0
```

**8.23.3.9 ratioCorr**

```
float Reconstruction::RecoCluster::ratioCorr = 0
```

**8.23.3.10 TLead**

```
int Reconstruction::RecoCluster::TLead = 0
```

**8.23.3.11 v\_pads**

```
std::vector<RecoPad *> Reconstruction::RecoCluster::v_pads
```

**8.23.3.12 yCluster**

```
float Reconstruction::RecoCluster::yCluster = 0
```

### 8.23.3.13 yWeight

```
float Reconstruction::RecoCluster::yWeight = 0
```

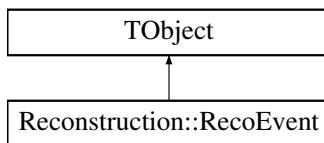
The documentation for this class was generated from the following files:

- Reconstruction/inc/dEdx.h
- Reconstruction/src/dEdx.cxx

## 8.24 Reconstruction::RecoEvent Class Reference

```
#include <dEdx.h>
```

Inheritance diagram for Reconstruction::RecoEvent:



### Public Member Functions

- [RecoEvent \(\)](#)
- virtual [~RecoEvent \(\)](#)
- void [Clear \(\)](#)
- [ClassDef \(RecoEvent, 1\)](#)

### Public Attributes

- std::vector< [RecoModule](#) \* > [v\\_modules](#)
- std::vector< int > [v\\_modules\\_position](#)
- bool [selected](#) = false
- int [eventNbr](#) = 0
- float [dEdxXP](#) = 0
- float [dEdxWF](#) = 0
- float [dEdxGP1](#) = 0
- float [dEdxGP2](#) = 0
- float [dEdxGP3](#) = 0
- float [dEdxGP4](#) = 0
- float [dEdxGP5](#) = 0
- float [dEdxXPnoTrunc](#) = 0
- float [dEdxWFnoTrunc](#) = 0
- int [NCrossedPads](#) = 0
- int [NPads](#) = 0
- int [NClusters](#) = 0
- float [lengthXP](#) = 0
- float [lengthWF](#) = 0
- int [numberOfModules](#) = 0
- float [avg\\_pad\\_mult](#) = 0
- TH1F \* [GWF](#) = nullptr
- TH1F \* [GWFtruncatedGP1](#) = nullptr
- int [peakingTime](#) = 0
- int [timeBinSize](#) = 0

### 8.24.1 Constructor & Destructor Documentation

#### 8.24.1.1 RecoEvent()

```
Reconstruction::RecoEvent::RecoEvent ()
```

#### 8.24.1.2 ~RecoEvent()

```
Reconstruction::RecoEvent::~RecoEvent () [virtual]
```

### 8.24.2 Member Function Documentation

#### 8.24.2.1 ClassDef()

```
Reconstruction::RecoEvent::ClassDef (
    RecoEvent ,
    1 )
```

#### 8.24.2.2 Clear()

```
void Reconstruction::RecoEvent::Clear ()
```

### 8.24.3 Member Data Documentation

#### 8.24.3.1 avg\_pad\_mult

```
float Reconstruction::RecoEvent::avg_pad_mult = 0
```

#### 8.24.3.2 dEdxGP1

```
float Reconstruction::RecoEvent::dEdxGP1 = 0
```

#### 8.24.3.3 dEdxGP2

```
float Reconstruction::RecoEvent::dEdxGP2 = 0
```

#### 8.24.3.4 dEdxGP3

```
float Reconstruction::RecoEvent::dEdxGP3 = 0
```

### 8.24.3.5 dEdxGP4

```
float Reconstruction::RecoEvent::dEdxGP4 = 0
```

### 8.24.3.6 dEdxGP5

```
float Reconstruction::RecoEvent::dEdxGP5 = 0
```

### 8.24.3.7 dEdxWF

```
float Reconstruction::RecoEvent::dEdxWF = 0
```

### 8.24.3.8 dEdxWFnoTrunc

```
float Reconstruction::RecoEvent::dEdxWFnoTrunc = 0
```

### 8.24.3.9 dEdxXP

```
float Reconstruction::RecoEvent::dEdxXP = 0
```

### 8.24.3.10 dEdxXPnoTrunc

```
float Reconstruction::RecoEvent::dEdxXPnoTrunc = 0
```

### 8.24.3.11 eventNbr

```
int Reconstruction::RecoEvent::eventNbr = 0
```

### 8.24.3.12 GWF

```
TH1F* Reconstruction::RecoEvent::GWF = nullptr
```

### 8.24.3.13 GWFtruncatedGP1

```
TH1F* Reconstruction::RecoEvent::GWFtruncatedGP1 = nullptr
```

### 8.24.3.14 lengthWF

```
float Reconstruction::RecoEvent::lengthWF = 0
```

**8.24.3.15 lengthXP**

```
float Reconstruction::RecoEvent::lengthXP = 0
```

**8.24.3.16 NClusters**

```
int Reconstruction::RecoEvent::NClusters = 0
```

**8.24.3.17 NCrossedPads**

```
int Reconstruction::RecoEvent::NCrossedPads = 0
```

**8.24.3.18 NPads**

```
int Reconstruction::RecoEvent::NPads = 0
```

**8.24.3.19 numberOfModules**

```
int Reconstruction::RecoEvent::numberOfModules = 0
```

**8.24.3.20 peakingTime**

```
int Reconstruction::RecoEvent::peakingTime = 0
```

**8.24.3.21 selected**

```
bool Reconstruction::RecoEvent::selected = false
```

**8.24.3.22 timeBinSize**

```
int Reconstruction::RecoEvent::timeBinSize = 0
```

**8.24.3.23 v\_modules**

```
std::vector<RecoModule *> Reconstruction::RecoEvent::v_modules
```

**8.24.3.24 v\_modules\_position**

```
std::vector<int> Reconstruction::RecoEvent::v_modules_position
```

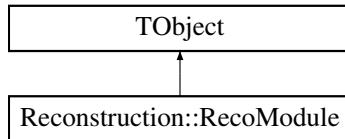
The documentation for this class was generated from the following files:

- Reconstruction/inc/[dEdx.h](#)
- Reconstruction/src/[dEdx.cxx](#)

## 8.25 Reconstruction::RecoModule Class Reference

```
#include <dEdx.h>
```

Inheritance diagram for Reconstruction::RecoModule:



### Public Member Functions

- virtual ~RecoModule ()
- [ClassDef \(RecoModule, 1\)](#)

### Public Attributes

- std::vector< [RecoCluster](#) \* > v\_clusters
- bool selected = false
- int ID = 0
- int position = 0
- float dEdxXP = 0
- float dEdxWF = 0
- float dEdxXPnoTrunc = 0
- float dEdxWFnoTrunc = 0
- int NCrossedPads = 0
- int NClusters = 0
- int NPads = 0
- float lengthXP = 0
- float lengthWF = 0
- float phi = 0
- float avg\_pad\_mult = 0
- TF1 \* Track = new TF1("pTrackFit", "[0]\*x+[1]+[2]\*x\*x", 0, 2000)

### 8.25.1 Constructor & Destructor Documentation

#### 8.25.1.1 ~RecoModule()

```
Reconstruction::RecoModule::~RecoModule () [virtual]
```

### 8.25.2 Member Function Documentation

#### 8.25.2.1 ClassDef()

```
Reconstruction::RecoModule::ClassDef (
    RecoModule ,
    1 )
```

### 8.25.3 Member Data Documentation

#### 8.25.3.1 avg\_pad\_mult

```
float Reconstruction::RecoModule::avg_pad_mult = 0
```

#### 8.25.3.2 dEdxWF

```
float Reconstruction::RecoModule::dEdxWF = 0
```

#### 8.25.3.3 dEdxWFnoTrunc

```
float Reconstruction::RecoModule::dEdxWFnoTrunc = 0
```

#### 8.25.3.4 dEdxXP

```
float Reconstruction::RecoModule::dEdxXP = 0
```

#### 8.25.3.5 dEdxXPnoTrunc

```
float Reconstruction::RecoModule::dEdxXPnoTrunc = 0
```

#### 8.25.3.6 ID

```
int Reconstruction::RecoModule::ID = 0
```

#### 8.25.3.7 lengthWF

```
float Reconstruction::RecoModule::lengthWF = 0
```

#### 8.25.3.8 lengthXP

```
float Reconstruction::RecoModule::lengthXP = 0
```

#### 8.25.3.9 NClusters

```
int Reconstruction::RecoModule::NClusters = 0
```

#### 8.25.3.10 NCrossedPads

```
int Reconstruction::RecoModule::NCrossedPads = 0
```

### 8.25.3.11 NPads

```
int Reconstruction::RecoModule::NPads = 0
```

### 8.25.3.12 phi

```
float Reconstruction::RecoModule::phi = 0
```

### 8.25.3.13 position

```
int Reconstruction::RecoModule::position = 0
```

### 8.25.3.14 selected

```
bool Reconstruction::RecoModule::selected = false
```

### 8.25.3.15 Track

```
TF1* Reconstruction::RecoModule::Track = new TF1("pTrackFit", "[0]*x+[1]+[2]*x*x", 0, 2000)
```

### 8.25.3.16 v\_clusters

```
std::vector<RecoCluster*> Reconstruction::RecoModule::v_clusters
```

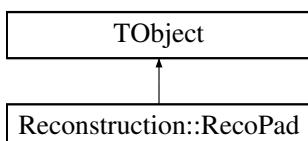
The documentation for this class was generated from the following files:

- Reconstruction/inc/[dEdx.h](#)
- Reconstruction/src/[dEdx.cxx](#)

## 8.26 Reconstruction::RecoPad Class Reference

```
#include <dEdx.h>
```

Inheritance diagram for Reconstruction::RecoPad:



### Public Member Functions

- virtual [~RecoPad \(\)](#)
- [ClassDef \(RecoPad, 1\)](#)

## Public Attributes

- bool `leading` = false
- int `ix` = 0
- int `iy` = 0
- float `xPad` = 0
- float `yPad` = 0
- float `ADCmax` = 0
- float `ADCmax_base` = 0
- float `charge` = 0
- float `dEdxXP` = 0
- float `RC` = 0
- float `gain` = 0
- float `GainCorrection` = 0
- float `phi` = 0
- float `d` = 0
- float `dd` = 0
- float `length` = 0
- int `TMax` = 0
- int `T0` = 0
- float `driftDistance` = 0
- float `ratioDrift` = 0
- float `ratioFile` = 0
- float `ratio` = 0
- float `dy` = 0

## 8.26.1 Constructor & Destructor Documentation

### 8.26.1.1 ~RecoPad()

```
Reconstruction::RecoPad::~RecoPad () [virtual]
```

## 8.26.2 Member Function Documentation

### 8.26.2.1 ClassDef()

```
Reconstruction::RecoPad::ClassDef (
    RecoPad ,
    1 )
```

## 8.26.3 Member Data Documentation

### 8.26.3.1 ADCmax

```
float Reconstruction::RecoPad::ADCmax = 0
```

**8.26.3.2 ADCmax\_base**

```
float Reconstruction::RecoPad::ADCmax_base = 0
```

**8.26.3.3 charge**

```
float Reconstruction::RecoPad::charge = 0
```

**8.26.3.4 d**

```
float Reconstruction::RecoPad::d = 0
```

**8.26.3.5 dd**

```
float Reconstruction::RecoPad::dd = 0
```

**8.26.3.6 dEdxXP**

```
float Reconstruction::RecoPad::dEdxXP = 0
```

**8.26.3.7 driftDistance**

```
float Reconstruction::RecoPad::driftDistance = 0
```

**8.26.3.8 dy**

```
float Reconstruction::RecoPad::dy = 0
```

**8.26.3.9 gain**

```
float Reconstruction::RecoPad::gain = 0
```

**8.26.3.10 GainCorrection**

```
float Reconstruction::RecoPad::GainCorrection = 0
```

**8.26.3.11 ix**

```
int Reconstruction::RecoPad::ix = 0
```

**8.26.3.12 iy**

```
int Reconstruction::RecoPad::iy = 0
```

**8.26.3.13 leading**

```
bool Reconstruction::RecoPad::leading = false
```

**8.26.3.14 length**

```
float Reconstruction::RecoPad::length = 0
```

**8.26.3.15 phi**

```
float Reconstruction::RecoPad::phi = 0
```

**8.26.3.16 ratio**

```
float Reconstruction::RecoPad::ratio = 0
```

**8.26.3.17 ratioDrift**

```
float Reconstruction::RecoPad::ratioDrift = 0
```

**8.26.3.18 ratioFile**

```
float Reconstruction::RecoPad::ratioFile = 0
```

**8.26.3.19 RC**

```
float Reconstruction::RecoPad::RC = 0
```

**8.26.3.20 T0**

```
int Reconstruction::RecoPad::T0 = 0
```

**8.26.3.21 TMax**

```
int Reconstruction::RecoPad::TMax = 0
```

### 8.26.3.22 xPad

```
float Reconstruction::RecoPad::xPad = 0
```

### 8.26.3.23 yPad

```
float Reconstruction::RecoPad::yPad = 0
```

The documentation for this class was generated from the following files:

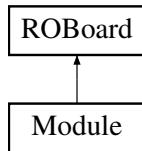
- Reconstruction/inc/[dEdx.h](#)
- Reconstruction/src/[dEdx.cxx](#)

## 8.27 ROBoard Class Reference

Abstraction for a read-out board grouping pads and providing mapping.

```
#include <ROBoard.h>
```

Inheritance diagram for ROBoard:



### Public Member Functions

- ROBoard (Model\_ReadOutGeometry \*pModel\_ReadOutGeometry, Model\_Electronics \*pModel\_Electronics, Model\_ChargeI \*pModel\_ChargeI)
- virtual ~ROBoard ()
- const Model\_ReadOutGeometry \* Get\_Model\_ReadOutGeometry ()
- const Model\_Electronics \* Get\_Model\_Electronics ()
- const Model\_ChargeI \* Get\_Model\_ChargeI ()
- virtual void Add\_Pad (Pad \*pPad)
- int Get\_NberOfPads () const
- Pad \* Get\_Pad (const int &iIndex1D)
- int Get\_IsThisPadExisting (const int &iX, const int &iY) const
- Pad \* Get\_Pad (const int &iX, const int &iY)
- Pad \* Get\_Pad (Pad \*pPad)

### Protected Member Functions

- int GetLinearIndex (const int &iX, const int &iY) const
- void Ini\_Models (Model\_ReadOutGeometry \*pModel\_ReadOutGeometry, Model\_Electronics \*pModel\_Electronics, Model\_ChargeI \*pModel\_ChargeI)

## Protected Attributes

- short int `m_Nx`
- short int `m_Ny`
- std::vector< int > `V_ExisFlag2D`
- std::vector< `Pad` \* > `V_Pad`
- `Model_ReadOutGeometry` \* `p_Model_ReadOutGeometry`
- `Model_Electronics` \* `p_Model_Electronics`
- `Model_ChargeI` \* `p_Model_ChargeI`

### 8.27.1 Detailed Description

Abstraction for a read-out board grouping pads and providing mapping.

### 8.27.2 Constructor & Destructor Documentation

#### 8.27.2.1 ROBoard()

```
ROBoard::ROBoard (
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_ChargeI * pModel_ChargeI)
```

Constructor

#### 8.27.2.2 ~ROBoard()

```
ROBoard::~ROBoard () [virtual]
```

### 8.27.3 Member Function Documentation

#### 8.27.3.1 Add\_Pad()

```
void ROBoard::Add_Pad (
    Pad * pPad) [virtual]
```

Reimplemented in [Module](#).

#### 8.27.3.2 Get\_IsThisPadExisting()

```
int ROBoard::Get_IsThisPadExisting (
    const int & iX,
    const int & iY) const
```

#### 8.27.3.3 Get\_Model\_ChargeI()

```
const Model_ChargeI * ROBoard::Get_Model_ChargeI ()
```

#### 8.27.3.4 Get\_Model\_Electronics()

```
const Model_Electronics * ROBoard::Get_Model_Electronics ()
```

#### 8.27.3.5 Get\_Model\_ReadOutGeometry()

```
const Model_ReadOutGeometry * ROBoard::Get_Model_ReadOutGeometry ()
```

#### 8.27.3.6 Get\_NberOfPads()

```
int ROBoard::Get_NberOfPads () const
```

#### 8.27.3.7 Get\_Pad() [1/3]

```
Pad * ROBoard::Get_Pad (
    const int & IndexID)
```

#### 8.27.3.8 Get\_Pad() [2/3]

```
Pad * ROBoard::Get_Pad (
    const int & iX,
    const int & iY)
```

#### 8.27.3.9 Get\_Pad() [3/3]

```
Pad * ROBoard::Get_Pad (
    Pad * pPad)
```

#### 8.27.3.10 GetLinearIndex()

```
int ROBoard::GetLinearIndex (
    const int & iX,
    const int & iY) const [protected]
```

#### 8.27.3.11 Ini\_Models()

```
void ROBoard::Ini_Models (
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_ChargeI * pModel_ChargeI) [protected]
```

## 8.27.4 Member Data Documentation

### 8.27.4.1 m\_Nx

```
short int ROBoard::m_Nx [protected]
```

### 8.27.4.2 m\_Ny

```
short int ROBoard::m_Ny [protected]
```

### 8.27.4.3 p\_Model\_ChargeI

```
Model_ChargeI* ROBoard::p_Model_ChargeI [protected]
```

### 8.27.4.4 p\_Model\_Electronics

```
Model_Electronics* ROBoard::p_Model_Electronics [protected]
```

### 8.27.4.5 p\_Model\_ReadOutGeometry

```
Model_ReadOutGeometry* ROBoard::p_Model_ReadOutGeometry [protected]
```

### 8.27.4.6 V\_ExisFlag2D

```
std::vector<int> ROBoard::V_ExisFlag2D [protected]
```

### 8.27.4.7 V\_Pad

```
std::vector<Pad *> ROBoard::V_Pad [protected]
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[ROBoard.h](#)
- ModelEvent/src/[ROBoard.cxx](#)

## 8.28 Sample Class Reference

```
#include <Sample.h>
```

## Public Member Functions

- `Sample ()`
- `Sample (Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI)`
- `virtual ~Sample ()`
- `void SmallDump () const`
- `void WriteOut () const`
- `void Set_Model_ReadOutGeometry (Model_ReadOutGeometry *pModel_ReadOutGeometry)`
- `void Set_Model_Electronics (Model_Electronics *pModel_Electronics)`
- `void Set_Model_Charge (Model_ChargeI *pModel_ChargeI)`
- `Model_ReadOutGeometry * Get_Model_ReadOutGeometry ()`
- `Model_Electronics * Get_Model_Electronics ()`
- `Model_ChargeI * Get_Model_ChargeI ()`
- `void Add_Event (Event *pEvent)`
- `int Get_NberOfEvents () const`
- `Event * Get_Event (int Index1D)`
- `SetOfTracks * Get_SetOfTracks_ForThisModule (const int &ModuleNber)`
- `void SetFilePRF (const double &Norm, const double &a2, const double &a4, const double &b2, const double &b4)`
- `void GetFilePRF (int &FilePRF_Exist, double &Norm, double &a2, double &a4, double &b2, double &b4)`
- `void Set_RC (const double &RC)`
- `void Set_DD (const double &RC)`
- `int StatusFit ()`
- `double Get_RC () const`
- `double Get_eRC () const`
- `double Get_DD () const`
- `double Get_eDD () const`
- `double Get_Chi2Min () const`
- `double Get_RC_BeforeMinimisation () const`
- `double Get_DD_BeforeMinimisation () const`

### 8.28.1 Constructor & Destructor Documentation

#### 8.28.1.1 Sample() [1/2]

```
Sample::Sample ()
```

Constructor

#### 8.28.1.2 Sample() [2/2]

```
Sample::Sample (
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_ChargeI * pModel_ChargeI)
```

#### 8.28.1.3 ~Sample()

```
Sample::~Sample () [virtual]
```

## 8.28.2 Member Function Documentation

### 8.28.2.1 Add\_Event()

```
void Sample::Add_Event (
```

`Event * pEvent)`

### 8.28.2.2 Get\_ChisqMin()

```
double Sample::Get_ChisqMin () const
```

### 8.28.2.3 Get\_DD()

```
double Sample::Get_DD () const
```

### 8.28.2.4 Get\_DD\_BeforeMinimisation()

```
double Sample::Get_DD_BeforeMinimisation () const
```

### 8.28.2.5 Get\_eDD()

```
double Sample::Get_eDD () const
```

### 8.28.2.6 Get\_eRC()

```
double Sample::Get_eRC () const
```

### 8.28.2.7 Get\_Event()

```
Event * Sample::Get_Event (
```

`int IndexID)`

### 8.28.2.8 Get\_Model\_ChargeI()

```
Model_ChargeI * Sample::Get_Model_ChargeI ()
```

### 8.28.2.9 Get\_Model\_Electronics()

```
Model_Electronics * Sample::Get_Model_Electronics ()
```

### 8.28.2.10 Get\_Model\_ReadOutGeometry()

```
Model_ReadOutGeometry * Sample::Get_Model_ReadOutGeometry ()
```

### 8.28.2.11 Get\_NberOfEvents()

```
int Sample::Get_NberOfEvents () const
```

### 8.28.2.12 Get\_RC()

```
double Sample::Get_RC () const
```

### 8.28.2.13 Get\_RC\_BeforeMinimisation()

```
double Sample::Get_RC_BeforeMinimisation () const
```

### 8.28.2.14 Get\_SetOfTracks\_ForThisModule()

```
SetOfTracks * Sample::Get_SetOfTracks_ForThisModule (
    const int & ModuleNber)
```

### 8.28.2.15 GetFilePRF()

```
void Sample::GetFilePRF (
    int & FilePRF_Exist,
    double & Norm,
    double & a2,
    double & a4,
    double & b2,
    double & b4)
```

### 8.28.2.16 Set\_DD()

```
void Sample::Set_DD (
    const double & RC)
```

### 8.28.2.17 Set\_Model\_Charge()

```
void Sample::Set_Model_Charge (
    Model_ChargeI * pModel_ChargeI)
```

### 8.28.2.18 Set\_Model\_Electronics()

```
void Sample::Set_Model_Electronics (
    Model_Electronics * pModel_Electronics)
```

**8.28.2.19 Set\_Model\_ReadOutGeometry()**

```
void Sample::Set_Model_ReadOutGeometry (
    Model_ReadOutGeometry * pModel_ReadOutGeometry)
```

**8.28.2.20 Set\_RC()**

```
void Sample::Set_RC (
    const double & RC)
```

**8.28.2.21 SetFilePRF()**

```
void Sample::SetFilePRF (
    const double & Norm,
    const double & a2,
    const double & a4,
    const double & b2,
    const double & b4)
```

**8.28.2.22 SmallDump()**

```
void Sample::SmallDump () const
```

**8.28.2.23 StatusFit()**

```
int Sample::StatusFit ()
```

**8.28.2.24 WriteOut()**

```
void Sample::WriteOut () const
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[Sample.h](#)
- ModelEvent/src/[Sample.cxx](#)

## 8.29 Selector Class Reference

Encapsulates event/module/cluster selection logic.

```
#include <Selector.h>
```

## Public Member Functions

- `Selector (const std::string DefSelection)`
- `Selector ()`
- `virtual ~Selector ()`
- `void Reset_Selection ()`  
`Reset all selections to empty.`
- `void Add_Selection (const std::string &SelectionName)`  
`Add a named selection.`
- `int NberOfSelections ()`  
`Get number of selections.`
- `std::string Get_SelectionName (const int &iTem)`  
`Retrieve a selection name by index.`
- `void Tell_Selection ()`  
`Print selection information to stdout.`
- `void ApplySelection (Sample &aSample, const int &ModuleNber)`  
`Apply selection to a Sample.`
- `void Apply_ASelection (Sample &aSample, const int &ModuleNber, const int &iTem)`  
`Apply a specific selection index to a Sample.`
- `void ApplySelection (Event *pEvent)`  
`Apply selection to an Event.`
- `void Apply_ASelection (Event *pEvent, const int &ModuleNber, const int &iTem)`  
`Apply a specific selection index to an Event.`
- `void Reset_StatCounters ()`  
`Reset statistical counters used for selection monitoring.`
- `void PrintStat ()`
- `double Get_Cut_StageFinal_NCluster_Low ()`
- `void Set_Cut_StageFinal_NCluster_Low (double Cut_StageFinal_NCluster_Low)`
- `double Get_Cut_Stage5_Npads_Hig ()`
- `void Set_Cut_Stage5_Npads_Hig (double Cut_Stage5_Npads_Hig)`
- `double Get_Cut_Stage2_EventBased ()`
- `void Set_Cut_Stage2_EventBased (double Cut_Stage2_EventBased)`
- `double Get_Cut_Stage3_TLow ()`
- `double Get_Cut_Stage3_THigh ()`
- `void Set_Cut_Stage3_TLow (double Cut_Stage3_TLow)`
- `void Set_Cut_Stage3_THigh (double Cut_Stage3_THigh)`
- `double Get_Cut_Stage6_Amax_Low ()`
- `void Set_Cut_Stage6_Amax_Low (double Cut_Stage6_Amax_Low)`
- `double Get_Cut_Stage6_Amax_Hig ()`
- `void Set_Cut_Stage6_Amax_Hig (double Cut_Stage6_Amax_Hig)`
- `double Get_Cut_Stage4_APM_Low ()`
- `void Set_Cut_Stage4_APM_Low (double Cut_Stage4_APM_Low)`
- `double Get_Cut_Stage4_APM_High ()`
- `void Set_Cut_Stage4_APM_High (double Cut_Stage4_APM_High)`

### 8.29.1 Detailed Description

Encapsulates event/module/cluster selection logic.

## 8.29.2 Constructor & Destructor Documentation

### 8.29.2.1 Selector() [1/2]

```
Selector::Selector (
    const std::string DefSelection)
```

Constructor with default selection name

### 8.29.2.2 Selector() [2/2]

```
Selector::Selector ()
```

Default constructor

### 8.29.2.3 ~Selector()

```
Selector::~Selector () [virtual]
```

Virtual destructor

## 8.29.3 Member Function Documentation

### 8.29.3.1 Add\_Selection()

```
void Selector::Add_Selection (
    const std::string & SelectionName)
```

Add a named selection.

### 8.29.3.2 Apply\_ASelection() [1/2]

```
void Selector::Apply_ASelection (
    Event * pEvent,
    const int & ModuleNber,
    const int & iItem)
```

Apply a specific selection index to an [Event](#).

### 8.29.3.3 Apply\_ASelection() [2/2]

```
void Selector::Apply_ASelection (
    Sample & aSample,
    const int & ModuleNber,
    const int & iItem)
```

Apply a specific selection index to a [Sample](#).

### 8.29.3.4 ApplySelection() [1/2]

```
void Selector::ApplySelection (
    Event * pEvent)
```

Apply selection to an [Event](#).

### 8.29.3.5 ApplySelection() [2/2]

```
void Selector::ApplySelection (
    Sample & aSample,
    const int & ModuleNber)
```

Apply selection to a [Sample](#).

#### Parameters

<i>aSample</i>	reference to the <a href="#">Sample</a>
<i>ModuleNber</i>	module index

### 8.29.3.6 Get\_Cut\_Stage2\_EventBased()

```
double Selector::Get_Cut_Stage2_EventBased ()
```

### 8.29.3.7 Get\_Cut\_Stage3\_THigh()

```
double Selector::Get_Cut_Stage3_THigh ()
```

### 8.29.3.8 Get\_Cut\_Stage3\_TLow()

```
double Selector::Get_Cut_Stage3_TLow ()
```

### 8.29.3.9 Get\_Cut\_Stage4\_APM\_High()

```
double Selector::Get_Cut_Stage4_APM_High ()
```

### 8.29.3.10 Get\_Cut\_Stage4\_APM\_Low()

```
double Selector::Get_Cut_Stage4_APM_Low ()
```

### 8.29.3.11 Get\_Cut\_Stage5\_Npads\_Hig()

```
double Selector::Get_Cut_Stage5_Npads_Hig ()
```

**8.29.3.12 Get\_Cut\_Stage6\_Amax\_Hig()**

```
double Selector::Get_Cut_Stage6_Amax_Hig ()
```

**8.29.3.13 Get\_Cut\_Stage6\_Amax\_Low()**

```
double Selector::Get_Cut_Stage6_Amax_Low ()
```

**8.29.3.14 Get\_Cut\_StageFinal\_NCluster\_Low()**

```
double Selector::Get_Cut_StageFinal_NCluster_Low ()
```

**8.29.3.15 Get\_SelectionName()**

```
std::string Selector::Get_SelectionName (
    const int & iItem)
```

Retrieve a selection name by index.

**8.29.3.16 NberOfSelections()**

```
int Selector::NberOfSelections ()
```

Get number of selections.

**8.29.3.17 PrintStat()**

```
void Selector::PrintStat ()
```

**8.29.3.18 Reset\_Selection()**

```
void Selector::Reset_Selection ()
```

Reset all selections to empty.

**8.29.3.19 Reset\_StatCounters()**

```
void Selector::Reset_StatCounters ()
```

Reset statistical counters used for selection monitoring.

**8.29.3.20 Set\_Cut\_Stage2\_EventBased()**

```
void Selector::Set_Cut_Stage2_EventBased (
    double Cut_Stage2_EventBased)
```

**8.29.3.21 Set\_Cut\_Stage3\_THigh()**

```
void Selector::Set_Cut_Stage3_THigh (
    double Cut_Stage3_THigh)
```

**8.29.3.22 Set\_Cut\_Stage3\_TLow()**

```
void Selector::Set_Cut_Stage3_TLow (
    double Cut_Stage3_TLow)
```

**8.29.3.23 Set\_Cut\_Stage4\_APM\_High()**

```
void Selector::Set_Cut_Stage4_APM_High (
    double Cut_Stage4_APM_High)
```

**8.29.3.24 Set\_Cut\_Stage4\_APM\_Low()**

```
void Selector::Set_Cut_Stage4_APM_Low (
    double Cut_Stage4_APM_Low)
```

**8.29.3.25 Set\_Cut\_Stage5\_Npads\_Hig()**

```
void Selector::Set_Cut_Stage5_Npads_Hig (
    double Cut_Stage5_Npads_Hig)
```

**8.29.3.26 Set\_Cut\_Stage6\_Amax\_Hig()**

```
void Selector::Set_Cut_Stage6_Amax_Hig (
    double Cut_Stage6_Amax_Hig)
```

**8.29.3.27 Set\_Cut\_Stage6\_Amax\_Low()**

```
void Selector::Set_Cut_Stage6_Amax_Low (
    double Cut_Stage6_Amax_Low)
```

**8.29.3.28 Set\_Cut\_StageFinal\_NCluster\_Low()**

```
void Selector::Set_Cut_StageFinal_NCluster_Low (
    double Cut_StageFinal_NCluster_Low)
```

### 8.29.3.29 Tell\_Selection()

```
void Selector::Tell_Selection ()
```

Print selection information to stdout.

The documentation for this class was generated from the following files:

- Reconstruction/inc/[Selector.h](#)
- Reconstruction/src/[Selector.cxx](#)

## 8.30 SetOfTracks Class Reference

```
#include <SetOfTracks.h>
```

### Public Member Functions

- [SetOfTracks \(\)](#)
- virtual [~SetOfTracks \(\)](#)
- void [Add\\_Track \(const Track \\*pTrack\)](#)
- int [Get\\_NberOfTrack \(\) const](#)
- const Track \* [Get\\_Track \(int Index1D\) const](#)
- void [DumpRec \(\) const](#)
- int [Get\\_MinimalNberOfEntries \(\)](#)
- void [Set\\_MinimalNberOfEntries \(const int &MinimalNberOfEntries\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackInTracks \(const std::string &TAG, const int &iYBeam\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackInTracks \(const std::string &TAG, const int &Nbins, const double &Ymin, const double &Ymax\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackInTracks \(const std::string &TAG, const int &iYBeam, const int &iCol\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackInTracks \(const std::string &TAG, const int &Nbins, const double &Ymin, const double &Ymax, const int &iCol\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YFitCol \(const std::string &TAG, const int &iYBeam\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YFitCol \(const std::string &TAG, const int &Nbins, const double &Ymin, const double &Ymax\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YFitCol \(const std::string &TAG, const int &iYBeam, const int &iCol\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YFitCol \(const std::string &TAG, const int &Nbins, const double &Ymin, const double &Ymax, const int &iCol\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackYPadInTracks \(const std::string &TAG, const int &iCol\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackYPadInTracks \(const std::string &TAG, const int &iCol, const int &Nbins, const double &Xmin, const double &Xmax\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackYPadLeadingInTracks \(const std::string &TAG, const int &iCol\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_YTrackYPadLeadingInTracks \(const std::string &TAG, const int &iCol, const int &Nbins, const double &Xmin, const double &Xmax\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_TrackDeltaT \(const std::string &TAG, const int &iCol\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_Ch2MinPerNODF \(const std::string &TAGFinal, const double &Xmax\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_Ch2Min \(const std::string &TAGFinal, const double &Xmax\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_Residual \(const std::string &TAGFinal, double &Mean, double &eMean, double &Sigma, double &eSigma\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_Residual \(const std::string &TAGFinal, const int &iX, double &Mean, double &eMean, double &Sigma, double &eSigma\)](#)
- TH1F \* [GiveMe\\_pTH1F\\_Residual \(const std::string &TAGFinal, const int &iX, double YL, double YH, double YL\\_Rescaled, double YH\\_Rescaled, double &Mean, double &eMean, double &Sigma, double &eSigma\)](#)

- TH1F \* [GiveMe\\_pTH1F\\_Pull](#) (const std::string &TAGFinal)
- TH1F \* [GiveMe\\_pTH1F\\_Pull](#) (const std::string &TAGFinal, double &Mean, double &eMean, double &Sigma, double &eSigma)
- TH1F \* [GiveMe\\_pTH1F\\_Pull](#) (const std::string &TAGFinal, double &Mean, double &eMean, double &Sigma, double &eSigma, const double &XminH, const double &XmaxH)
- TH1F \* [GiveMe\\_pTH1F\\_Pull](#) (const std::string &TAGFinal, const int &iX, double &Mean, double &eMean, double &Sigma, double &eSigma)
- TH1F \* [GiveMe\\_pTH1F\\_Pull](#) (const std::string &TAGFinal, const int &iX, double &Mean, double &eMean, double &Sigma, double &eSigma, const double &XminH, const double &XmaxH)

## 8.30.1 Constructor & Destructor Documentation

### 8.30.1.1 SetOfTracks()

```
SetOfTracks::SetOfTracks ()
```

### 8.30.1.2 ~SetOfTracks()

```
SetOfTracks::~SetOfTracks () [virtual]
```

## 8.30.2 Member Function Documentation

### 8.30.2.1 Add\_Track()

```
void SetOfTracks::Add_Track (
    const Track * pTrack)
```

### 8.30.2.2 DumpRec()

```
void SetOfTracks::DumpRec () const
```

### 8.30.2.3 Get\_MinimalNberOfEntries()

```
int SetOfTracks::Get_MinimalNberOfEntries ()
```

### 8.30.2.4 Get\_NberOfTrack()

```
int SetOfTracks::Get_NberOfTrack () const
```

### 8.30.2.5 Get\_Track()

```
const Track * SetOfTracks::Get_Track (
    int IndexID) const
```

**8.30.2.6 GiveMe\_pTH1F\_Ch2Min()**

```
TH1F * SetOfTracks::GiveMe_pTH1F_Ch2Min (
    const std::string & TAGFinal,
    const double & Xmax)
```

**8.30.2.7 GiveMe\_pTH1F\_Ch2MinPerNODF()**

```
TH1F * SetOfTracks::GiveMe_pTH1F_Ch2MinPerNODF (
    const std::string & TAGFinal,
    const double & Xmax)
```

**8.30.2.8 GiveMe\_pTH1F\_Pull() [1/5]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_Pull (
    const std::string & TAGFinal)
```

**8.30.2.9 GiveMe\_pTH1F\_Pull() [2/5]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_Pull (
    const std::string & TAGFinal,
    const int & iX,
    double & Mean,
    double & eMean,
    double & Sigma,
    double & eSigma)
```

**8.30.2.10 GiveMe\_pTH1F\_Pull() [3/5]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_Pull (
    const std::string & TAGFinal,
    const int & iX,
    double & Mean,
    double & eMean,
    double & Sigma,
    double & eSigma,
    const double & XminH,
    const double & XmaxH)
```

**8.30.2.11 GiveMe\_pTH1F\_Pull() [4/5]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_Pull (
    const std::string & TAGFinal,
    double & Mean,
    double & eMean,
    double & Sigma,
    double & eSigma)
```

### 8.30.2.12 GiveMe\_pTH1F\_Pull() [5/5]

```
TH1F * SetOfTracks::GiveMe_pTH1F_Pull (
    const std::string & TAGFinal,
    double & Mean,
    double & eMean,
    double & Sigma,
    double & eSigma,
    const double & XminH,
    const double & XmaxH)
```

### 8.30.2.13 GiveMe\_pTH1F\_Residual() [1/3]

```
TH1F * SetOfTracks::GiveMe_pTH1F_Residual (
    const std::string & TAGFinal,
    const int & iX,
    double & Mean,
    double & eMean,
    double & Sigma,
    double & eSigma)
```

### 8.30.2.14 GiveMe\_pTH1F\_Residual() [2/3]

```
TH1F * SetOfTracks::GiveMe_pTH1F_Residual (
    const std::string & TAGFinal,
    const int & iX,
    double YL,
    double YH,
    double YL_Rescaled,
    double YH_Rescaled,
    double & Mean,
    double & eMean,
    double & Sigma,
    double & eSigma)
```

### 8.30.2.15 GiveMe\_pTH1F\_Residual() [3/3]

```
TH1F * SetOfTracks::GiveMe_pTH1F_Residual (
    const std::string & TAGFinal,
    double & Mean,
    double & eMean,
    double & Sigma,
    double & eSigma)
```

### 8.30.2.16 GiveMe\_pTH1F\_TrackDeltaT()

```
TH1F * SetOfTracks::GiveMe_pTH1F_TrackDeltaT (
    const std::string & TAG,
    const int & iCol)
```

**8.30.2.17 GiveMe\_pTH1F\_YFitCol() [1/4]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_YFitCol (
    const std::string & TAG,
    const int & iYBeam)
```

**8.30.2.18 GiveMe\_pTH1F\_YFitCol() [2/4]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_YFitCol (
    const std::string & TAG,
    const int & iYBeam,
    const int & iCol)
```

**8.30.2.19 GiveMe\_pTH1F\_YFitCol() [3/4]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_YFitCol (
    const std::string & TAG,
    const int & Nbins,
    const double & Ymin,
    const double & Ymax)
```

**8.30.2.20 GiveMe\_pTH1F\_YFitCol() [4/4]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_YFitCol (
    const std::string & TAG,
    const int & Nbins,
    const double & Ymin,
    const double & Ymax,
    const int & iCol)
```

**8.30.2.21 GiveMe\_pTH1F\_YTrackInTracks() [1/4]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackInTracks (
    const std::string & TAG,
    const int & iYBeam)
```

**8.30.2.22 GiveMe\_pTH1F\_YTrackInTracks() [2/4]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackInTracks (
    const std::string & TAG,
    const int & iYBeam,
    const int & iCol)
```

**8.30.2.23 GiveMe\_pTH1F\_YTrackInTracks() [3/4]**

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackInTracks (
    const std::string & TAG,
    const int & Nbins,
    const double & Ymin,
    const double & Ymax)
```

### 8.30.2.24 GiveMe\_pTH1F\_YTrackInTracks() [4/4]

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackInTracks (
    const std::string & TAG,
    const int & Nbins,
    const double & Ymin,
    const double & Ymax,
    const int & iCol)
```

### 8.30.2.25 GiveMe\_pTH1F\_YTrackYPadInTracks() [1/2]

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackYPadInTracks (
    const std::string & TAG,
    const int & iCol)
```

### 8.30.2.26 GiveMe\_pTH1F\_YTrackYPadInTracks() [2/2]

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackYPadInTracks (
    const std::string & TAG,
    const int & iCol,
    const int & Nbins,
    const double & Xmin,
    const double & Xmax)
```

### 8.30.2.27 GiveMe\_pTH1F\_YTrackYPadLeadingInTracks() [1/2]

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackYPadLeadingInTracks (
    const std::string & TAG,
    const int & iCol)
```

### 8.30.2.28 GiveMe\_pTH1F\_YTrackYPadLeadingInTracks() [2/2]

```
TH1F * SetOfTracks::GiveMe_pTH1F_YTrackYPadLeadingInTracks (
    const std::string & TAG,
    const int & iCol,
    const int & Nbins,
    const double & Xmin,
    const double & Xmax)
```

### 8.30.2.29 Set\_MinimalNberOfEntries()

```
void SetOfTracks::Set_MinimalNberOfEntries (
    const int & MinimalNberOfEntries)
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[SetOfTracks.h](#)
- ModelEvent/src/[SetOfTracks.cxx](#)

## 8.31 StaticClusterFitter\_Diagonal Class Reference

### Public Member Functions

- [StaticClusterFitter\\_Diagonal \(\)](#)
- virtual [~StaticClusterFitter\\_Diagonal \(\)](#)

### Static Public Member Functions

- static void [Set \(ClusterFitter\\_Diagonal \\*pClusterFitter\\_Diagonal\)](#)

### Static Public Attributes

- static [ClusterFitter\\_Diagonal \\* p\\_ClusterFitter\\_Diagonal](#)

### 8.31.1 Constructor & Destructor Documentation

#### 8.31.1.1 StaticClusterFitter\_Diagonal()

```
StaticClusterFitter_Diagonal::StaticClusterFitter_Diagonal ()
```

#### 8.31.1.2 ~StaticClusterFitter\_Diagonal()

```
StaticClusterFitter_Diagonal::~StaticClusterFitter_Diagonal () [virtual]
```

### 8.31.2 Member Function Documentation

#### 8.31.2.1 Set()

```
void StaticClusterFitter_Diagonal::Set (
    ClusterFitter_Diagonal * pClusterFitter_Diagonal) [static]
```

### 8.31.3 Member Data Documentation

#### 8.31.3.1 p\_ClusterFitter\_Diagonal

```
ClusterFitter_Diagonal * StaticClusterFitter_Diagonal::p_ClusterFitter_Diagonal [static]
```

##### Initial value:

```
= NULL
```

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

- Fitters/src/[ClusterFitter.cxx](#)

## 8.32 StaticClusterFitter\_Horizontal Class Reference

### Public Member Functions

- [StaticClusterFitter\\_Horizontal \(\)](#)
- [virtual ~StaticClusterFitter\\_Horizontal \(\)](#)

### Static Public Member Functions

- [static void Set \(\[ClusterFitter\\\_Horizontal \\\*pClusterFitter\\\_Horizontal\]\(#\)\)](#)

### Static Public Attributes

- [static \[ClusterFitter\\\_Horizontal \\\* p\\\_ClusterFitter\\\_Horizontal\]\(#\) = NULL](#)

### 8.32.1 Constructor & Destructor Documentation

#### 8.32.1.1 StaticClusterFitter\_Horizontal()

```
StaticClusterFitter_Horizontal::StaticClusterFitter_Horizontal ()
```

#### 8.32.1.2 ~StaticClusterFitter\_Horizontal()

```
StaticClusterFitter_Horizontal::~StaticClusterFitter_Horizontal () [virtual]
```

### 8.32.2 Member Function Documentation

#### 8.32.2.1 Set()

```
void StaticClusterFitter_Horizontal::Set (
    ClusterFitter\_Horizontal \* pClusterFitter\_Horizontal) [static]
```

### 8.32.3 Member Data Documentation

#### 8.32.3.1 p\_ClusterFitter\_Horizontal

```
ClusterFitter\_Horizontal \* StaticClusterFitter\_Horizontal::p\_ClusterFitter\_Horizontal = NULL
[static]
```

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

- Fitters/src/[ClusterFitter.cxx](#)

## 8.33 StaticTrackFitter Class Reference

### Public Member Functions

- `StaticTrackFitter ()`
- `virtual ~StaticTrackFitter ()`

### Static Public Member Functions

- `static void Set (TrackFitter *pTrackFitter)`

### Static Public Attributes

- `static TrackFitter * p_TrackFitter = NULL`

### 8.33.1 Constructor & Destructor Documentation

#### 8.33.1.1 StaticTrackFitter()

```
StaticTrackFitter::StaticTrackFitter ()
```

#### 8.33.1.2 ~StaticTrackFitter()

```
StaticTrackFitter::~StaticTrackFitter () [virtual]
```

### 8.33.2 Member Function Documentation

#### 8.33.2.1 Set()

```
void StaticTrackFitter::Set (
    TrackFitter * pTrackFitter) [static]
```

### 8.33.3 Member Data Documentation

#### 8.33.3.1 p\_TrackFitter

```
TrackFitter * StaticTrackFitter::p_TrackFitter = NULL [static]
```

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

- Fitters/src/[TrackFitter.cxx](#)

## 8.34 Track Class Reference

```
#include <Track.h>
```

### Public Member Functions

- `Track` (const int &*EventNber*, const int &*EntryNber*, const int &*ModuleNber*)  
*Construct a Track.*
- virtual `~Track` ()  
*Virtual destructor.*
- int `Get_EventNber` () const  
*Get Event number.*
- int `Get_EntryNber` () const  
*Get Entry number.*
- int `Get_ModuleNber` () const  
*Get Module number.*
- void `SetNberOfParameters` (const int &*NberOfParam*)
- int `GetNberOfParameters` () const
- std::string `Get_ParameterName` (const int &*iPar*) const
- double `Get_ParameterValue` (const int &*iPar*) const
- double `Get_ParameterError` (const int &*iPar*) const
- double `Get_ParameterValue_BeforeMinimisation` (const int &*iPar*) const
- void `Add_Cluster` (`Cluster` \**pCluster*)
- int `Get_NberOfCluster` () const
- const `Cluster` \* `Get_Cluster` (int *Index1D*) const
- double `Get_Residual` (int *Index1D*) const
- double `Get_Pull` (int *Index1D*) const
- void `DoClosure` ()
- void `DumpRec` () const
- double `Y_Position` (const double &*X*) const
- double `Get_Chi2Min` () const
- int `SetParameter` (`TVirtualFitter` \**pTVirtualFitter*)
- void `SetResults` (`TVirtualFitter` \**pTVirtualFitter*)
- double `Chi2` (double *par* [])
- void `SetParameters_Internal` (double *par* [])
- `TMatrixD` `Get_CovMatrix` () const

### Public Attributes

- int `m_NberOfClusters`
- std::vector< `Cluster` \* > `V_Cluster`
- std::vector< double > `V_Residual`
- std::vector< double > `V_Pull`

### 8.34.1 Constructor & Destructor Documentation

#### 8.34.1.1 `Track()`

```
Track::Track (
    const int & EventNber,
    const int & EntryNber,
    const int & ModuleNber)
```

*Construct a Track.*

#### Parameters

---

<i>EventNber</i>	event number
<i>EntryNber</i>	entry index
<i>ModuleNber</i>	module index

### 8.34.1.2 ~Track()

```
Track::~Track () [virtual]
```

Virtual destructor.

## 8.34.2 Member Function Documentation

### 8.34.2.1 Add\_Cluster()

```
void Track::Add_Cluster (
    Cluster * pCluster)
```

### 8.34.2.2 Chi2()

```
double Track::Chi2 (
    double par[ ])
```

### 8.34.2.3 DoClosure()

```
void Track::DoClosure ()
```

### 8.34.2.4 DumpRec()

```
void Track::DumpRec () const
```

### 8.34.2.5 Get\_ChisqMin()

```
double Track::Get_ChisqMin () const
```

### 8.34.2.6 Get\_Cluster()

```
const Cluster * Track::Get_Cluster (
    int IndexID) const
```

### 8.34.2.7 Get\_CovMatrix()

```
TMatrixD Track::Get_CovMatrix () const
```

**8.34.2.8 Get\_EntryNber()**

```
int Track::Get_EntryNber () const
```

Get Entry number.

**8.34.2.9 Get\_EventNber()**

```
int Track::Get_EventNber () const
```

Get Event number.

**8.34.2.10 Get\_ModuleNber()**

```
int Track::Get_ModuleNber () const
```

Get Module number.

**8.34.2.11 Get\_NberOfCluster()**

```
int Track::Get_NberOfCluster () const
```

**8.34.2.12 Get\_ParameterError()**

```
double Track::Get_ParameterError (
    const int & iPar) const
```

**8.34.2.13 Get\_ParameterName()**

```
std::string Track::Get_ParameterName (
    const int & iPar) const
```

**8.34.2.14 Get\_ParameterValue()**

```
double Track::Get_ParameterValue (
    const int & iPar) const
```

**8.34.2.15 Get\_ParameterValue\_BeforeMinimisation()**

```
double Track::Get_ParameterValue_BeforeMinimisation (
    const int & iPar) const
```

**8.34.2.16 Get\_Pull()**

```
double Track::Get_Pull (
    int IndexID) const
```

**8.34.2.17 Get\_Residual()**

```
double Track::Get_Residual (
    int IndexID) const
```

**8.34.2.18 GetNberOfParameters()**

```
int Track::GetNberOfParameters () const
```

**8.34.2.19 SetNberOfParameters()**

```
void Track::SetNberOfParameters (
    const int & NberOfParam)
```

**8.34.2.20 SetParameter()**

```
int Track::SetParameter (
    TVirtualFitter * pTVirtualFitter)
```

**8.34.2.21 SetParameters\_Internal()**

```
void Track::SetParameters_Internal (
    double par[ ])
```

**8.34.2.22 SetResults()**

```
void Track::SetResults (
    TVirtualFitter * pTVirtualFitter)
```

**8.34.2.23 Y\_Position()**

```
double Track::Y_Position (
    const double & X) const
```

**8.34.3 Member Data Documentation****8.34.3.1 m\_NberOfClusters**

```
int Track::m_NberOfClusters
```

### 8.34.3.2 V\_Cluster

```
std::vector<Cluster *> Track::V_Cluster
```

### 8.34.3.3 V\_Pull

```
std::vector<double> Track::V_Pull
```

### 8.34.3.4 V\_Residual

```
std::vector<double> Track::V_Residual
```

The documentation for this class was generated from the following files:

- ModelEvent/inc/[Track.h](#)
- ModelEvent/src/[Track.cxx](#)

## 8.35 TrackFitter Class Reference

```
#include <TrackFitter.h>
```

### Public Member Functions

- [TrackFitter](#) (const std::string &FitterName, const int &NberOfParameters=2)  
*Construct a TrackFitter.*
- virtual [~TrackFitter](#) ()  
*Virtual destructor.*
- void [Set\\_Track](#) (Track \*pTrack)  
*Set the track to be fitted.*
- int [DoMinimisation](#) ()  
*Run the minimisation routine.*
- double [Chi2](#) (double par[])  
*Compute chi2 for the current parameter vector.*

### Protected Attributes

- std::string [m\\_FitterName](#)
- int [m\\_NberOfParameters](#)
- Track \* [p\\_Track](#)
- TVirtualFitter \* [p\\_TVVirtualFitter](#)  
*Fitter.*

### 8.35.1 Constructor & Destructor Documentation

#### 8.35.1.1 TrackFitter()

```
TrackFitter::TrackFitter (
    const std::string & FitterName,
    const int & NberOfParameters = 2)
```

Construct a [TrackFitter](#).

#### Parameters

---

<i>FitterName</i>	name used to identify the fitter instance
<i>NberOfParameters</i>	number of fit parameters (default: 2)

### 8.35.1.2 ~TrackFitter()

```
TrackFitter::~TrackFitter () [virtual]
```

Virtual destructor.

## 8.35.2 Member Function Documentation

### 8.35.2.1 Chi2()

```
double TrackFitter::Chi2 (
    double par[])
```

Compute chi2 for the current parameter vector.

#### Parameters

<i>par</i>	parameter array used for the chi2 evaluation
------------	--

#### Returns

computed chi2 value

### 8.35.2.2 DoMinimisation()

```
int TrackFitter::DoMinimisation ()
```

Run the minimisation routine.

#### Returns

status code (0 on success, non-zero on failure)

### 8.35.2.3 Set\_Track()

```
void TrackFitter::Set_Track (
    Track * pTrack)
```

Set the track to be fitted.

#### Parameters

<code>pTrack</code>	pointer to a <a href="#">Track</a> object (ownership not transferred)
---------------------	---

### 8.35.3 Member Data Documentation

#### 8.35.3.1 `m_FitterName`

```
std::string TrackFitter::m_FitterName [protected]
```

#### 8.35.3.2 `m_NberOfParameters`

```
int TrackFitter::m_NberOfParameters [protected]
```

#### 8.35.3.3 `p_Track`

```
Track* TrackFitter::p_Track [protected]
```

#### 8.35.3.4 `p_TVirtuFitter`

```
TVirtualFitter* TrackFitter::p_TVirtuFitter [protected]
```

Fitter.

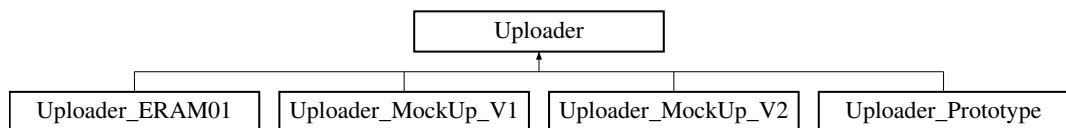
The documentation for this class was generated from the following files:

- Fitters/inc/[TrackFitter.h](#)
- Fitters/src/[TrackFitter.cxx](#)

## 8.36 Uploader Class Reference

```
#include <Uploader.h>
```

Inheritance diagram for Uploader:



## Public Member Functions

- `Uploader (const std::string &SampleFile, Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_Charge1D *pModel_Charge1D)`
- `virtual ~Uploader ()`
- `std::string Get_SampleFile ()`
- `int Get_NberOfEvent ()`
- `virtual Event * GiveMe_Event (const int &iEvent, const int &ModuleNber_Input, const int &Data_to_Use, const int &CloseWF=1)=0`
- `int Get_PRF_exist ()`
- `double Get_Norm ()`
- `double Get_a2 ()`
- `double Get_a4 ()`
- `double Get_b2 ()`
- `double Get_b4 ()`
- `Model_ReadOutGeometry * Get_Model_ReadOutGeometry ()`
- `Model_Electronics * Get_Model_Electronics ()`
- `Model_Charge1D * Get_Model_Charge1D ()`

## Protected Attributes

- `std::string m_SampleFile`
- `int m_NberOfEvent`
- `TFile * p_TFile`
- `TTree * p_TTree`
- `Model_ReadOutGeometry * p_Model_ReadOutGeometry`
- `Model_Electronics * p_Model_Electronics`
- `Model_Charge1D * p_Model_Charge1D`
- `short int m_PRF_exist`
- `double m_Norm`
- `double m_a2`
- `double m_a4`
- `double m_b2`
- `double m_b4`

## 8.36.1 Constructor & Destructor Documentation

### 8.36.1.1 Uploader()

```
Uploader::Uploader (
    const std::string & SampleFile,
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_Charge1D * pModel_Charge1D)
```

Constructor

### 8.36.1.2 ~Uploader()

```
Uploader::~Uploader () [virtual]
```

## 8.36.2 Member Function Documentation

### 8.36.2.1 Get\_a2()

```
double Uploader::Get_a2 ()
```

### 8.36.2.2 Get\_a4()

```
double Uploader::Get_a4 ()
```

### 8.36.2.3 Get\_b2()

```
double Uploader::Get_b2 ()
```

### 8.36.2.4 Get\_b4()

```
double Uploader::Get_b4 ()
```

### 8.36.2.5 Get\_Model\_Charge1D()

```
Model_Charge1D * Uploader::Get_Model_Charge1D ()
```

### 8.36.2.6 Get\_Model\_Electronics()

```
Model_Electronics * Uploader::Get_Model_Electronics ()
```

### 8.36.2.7 Get\_Model\_ReadOutGeometry()

```
Model_ReadOutGeometry * Uploader::Get_Model_ReadOutGeometry ()
```

### 8.36.2.8 Get\_NberOfEvent()

```
int Uploader::Get_NberOfEvent ()
```

### 8.36.2.9 Get\_Norm()

```
double Uploader::Get_Norm ()
```

### 8.36.2.10 Get\_PRF\_exist()

```
int Uploader::Get_PRF_exist ()
```

### 8.36.2.11 Get\_SampleFile()

```
std::string Uploader::Get_SampleFile ()
```

### 8.36.2.12 GiveMe\_Event()

```
virtual Event * Uploader::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use,
    const int & CloseWF = 1) [pure virtual]
```

Implemented in [Uploader\\_ERAM01](#), [Uploader\\_MockUp\\_V1](#), [Uploader\\_MockUp\\_V2](#), and [Uploader\\_Protoype](#).

## 8.36.3 Member Data Documentation

### 8.36.3.1 m\_a2

```
double Uploader::m_a2 [protected]
```

### 8.36.3.2 m\_a4

```
double Uploader::m_a4 [protected]
```

### 8.36.3.3 m\_b2

```
double Uploader::m_b2 [protected]
```

### 8.36.3.4 m\_b4

```
double Uploader::m_b4 [protected]
```

### 8.36.3.5 m\_NberOfEvent

```
int Uploader::m_NberOfEvent [protected]
```

### 8.36.3.6 m\_Norm

```
double Uploader::m_Norm [protected]
```

### 8.36.3.7 m\_PRF\_exist

```
short int Uploader::m_PRF_exist [protected]
```

### 8.36.3.8 m\_SampleFile

```
std::string Uploader::m_SampleFile [protected]
```

### 8.36.3.9 p\_Model\_Charge1D

```
Model_Charge1D* Uploader::p_Model_Charge1D [protected]
```

### 8.36.3.10 p\_Model\_Electronics

```
Model_Electronics* Uploader::p_Model_Electronics [protected]
```

### 8.36.3.11 p\_Model\_ReadOutGeometry

```
Model_ReadOutGeometry* Uploader::p_Model_ReadOutGeometry [protected]
```

### 8.36.3.12 p\_TFile

```
TFile* Uploader::p_TFile [protected]
```

### 8.36.3.13 p\_TTree

```
TTree* Uploader::p_TTree [protected]
```

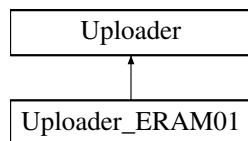
The documentation for this class was generated from the following files:

- Uploader/inc/[Uploader.h](#)
- Uploader/src/[Uploader.cxx](#)

## 8.37 Uploader\_ERAM01 Class Reference

```
#include <Uploader_ERAM01.h>
```

Inheritance diagram for Uploader\_ERAM01:



## Public Member Functions

- `Uploader_ERAM01 (const std::string &SampleFile, Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_Charge1D *pModel_Charge1D)`
- `virtual ~Uploader_ERAM01 ()`
- `virtual Event * GiveMe_Event (const int &iEvent, const int &ModuleNber_Input, const int &Data_to_Use, const int &CloseWF)`
- `virtual Event * GiveMe_Event (const int &iEvent, const int &ModuleNber_Input, const int &Data_to_Use)`

## Public Member Functions inherited from Uploader

- `Uploader (const std::string &SampleFile, Model_ReadOutGeometry *pModel_ReadOutGeometry, Model_Electronics *pModel_Electronics, Model_Charge1D *pModel_Charge1D)`
- `virtual ~Uploader ()`
- `std::string Get_SampleFile ()`
- `int Get_NberOfEvent ()`
- `int Get_PRF_exist ()`
- `double Get_Norm ()`
- `double Get_a2 ()`
- `double Get_a4 ()`
- `double Get_b2 ()`
- `double Get_b4 ()`
- `Model_ReadOutGeometry * Get_Model_ReadOutGeometry ()`
- `Model_Electronics * Get_Model_Electronics ()`
- `Model_Charge1D * Get_Model_Charge1D ()`

## Additional Inherited Members

### Protected Attributes inherited from Uploader

- `std::string m_SampleFile`
- `int m_NberOfEvent`
- `TFile * p_TFile`
- `TTree * p_TTree`
- `Model_ReadOutGeometry * p_Model_ReadOutGeometry`
- `Model_Electronics * p_Model_Electronics`
- `Model_Charge1D * p_Model_Charge1D`
- `short int m_PRF_exist`
- `double m_Norm`
- `double m_a2`
- `double m_a4`
- `double m_b2`
- `double m_b4`

## 8.37.1 Constructor & Destructor Documentation

### 8.37.1.1 Uploader\_ERAM01()

```
Uploader_ERAM01::Uploader_ERAM01 (
    const std::string & SampleFile,
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_Charge1D * pModel_Charge1D)
```

Constructor

### 8.37.1.2 ~Uploader\_ERAM01()

```
Uploader_ERAM01::~Uploader_ERAM01 () [virtual]
```

## 8.37.2 Member Function Documentation

### 8.37.2.1 GiveMe\_Event() [1/2]

```
Event * Uploader_ERAM01::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use) [virtual]
```

### 8.37.2.2 GiveMe\_Event() [2/2]

```
Event * Uploader_ERAM01::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use,
    const int & CloseWF) [virtual]
```

Implements [Uploader](#).

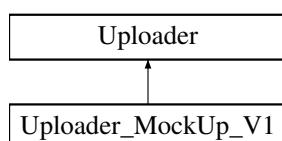
The documentation for this class was generated from the following files:

- Uploader/inc/[Uploader\\_ERAM01.h](#)
- Uploader/src/[Uploader\\_ERAM01.cxx](#)

## 8.38 Uploader\_MockUp\_V1 Class Reference

```
#include <Uploader_MockUp_V1.h>
```

Inheritance diagram for Uploader\_MockUp\_V1:



### Public Member Functions

- [Uploader\\_MockUp\\_V1](#) (const std::string &SampleFile, [Model\\_ReadOutGeometry](#) \*pModel\_ReadOutGeometry, [Model\\_Electronics](#) \*pModel\_Electronics, [Model\\_Charge1D](#) \*pModel\_Charge1D)
- virtual [~Uploader\\_MockUp\\_V1](#) ()
- virtual [Event \\* GiveMe\\_Event](#) (const int &iEvent, const int &ModuleNber\_Input, const int &Data\_to\_Use, const int &CloseWF)
- virtual [Event \\* GiveMe\\_Event](#) (const int &iEvent, const int &ModuleNber\_Input, const int &Data\_to\_Use)

## Public Member Functions inherited from Uploader

- Uploader (const std::string &SampleFile, Model\_ReadOutGeometry \*pModel\_ReadOutGeometry, Model\_Electronics \*pModel\_Electronics, Model\_Charge1D \*pModel\_Charge1D)
- virtual ~Uploader ()
- std::string Get\_SampleFile ()
- int Get\_NberOfEvent ()
- int Get\_PRF\_exist ()
- double Get\_Norm ()
- double Get\_a2 ()
- double Get\_a4 ()
- double Get\_b2 ()
- double Get\_b4 ()
- Model\_ReadOutGeometry \* Get\_Model\_ReadOutGeometry ()
- Model\_Electronics \* Get\_Model\_Electronics ()
- Model\_Charge1D \* Get\_Model\_Charge1D ()

## Additional Inherited Members

### Protected Attributes inherited from Uploader

- std::string m\_SampleFile
- int m\_NberOfEvent
- TFile \* p\_TFile
- TTree \* p\_TTree
- Model\_ReadOutGeometry \* p\_Model\_ReadOutGeometry
- Model\_Electronics \* p\_Model\_Electronics
- Model\_Charge1D \* p\_Model\_Charge1D
- short int m\_PRF\_exist
- double m\_Norm
- double m\_a2
- double m\_a4
- double m\_b2
- double m\_b4

## 8.38.1 Constructor & Destructor Documentation

### 8.38.1.1 Uploader\_MockUp\_V1()

```
Uploader_MockUp_V1::Uploader_MockUp_V1 (
    const std::string & SampleFile,
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_Charge1D * pModel_Charge1D)
```

Constructor

### 8.38.1.2 ~Uploader\_MockUp\_V1()

```
Uploader_MockUp_V1::~Uploader_MockUp_V1 () [virtual]
```

## 8.38.2 Member Function Documentation

### 8.38.2.1 GiveMe\_Event() [1/2]

```
Event * Uploader_MockUp_V1::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use) [virtual]
```

### 8.38.2.2 GiveMe\_Event() [2/2]

```
Event * Uploader_MockUp_V1::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use,
    const int & CloseWF) [virtual]
```

Implements [Uploader](#).

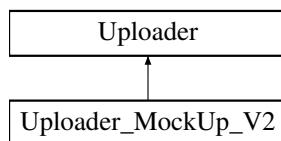
The documentation for this class was generated from the following files:

- Uploader/inc/[Uploader\\_MockUp\\_V1.h](#)
- Uploader/src/[Uploader\\_MockUp\\_V1.cxx](#)

## 8.39 Uploader\_MockUp\_V2 Class Reference

```
#include <Uploader_MockUp_V2.h>
```

Inheritance diagram for Uploader\_MockUp\_V2:



### Public Member Functions

- [Uploader\\_MockUp\\_V2](#) (const std::string &SampleFile, [Model\\_ReadOutGeometry](#) \*pModel\_ReadOutGeometry, [Model\\_Electronics](#) \*pModel\_Electronics, [Model\\_Charge1D](#) \*pModel\_Charge1D)
- virtual ~[Uploader\\_MockUp\\_V2](#) ()
- virtual [Event](#) \* [GiveMe\\_Event](#) (const int &iEvent, const int &ModuleNber\_Input, const int &Data\_to\_Use, const int &CloseWF)
- virtual [Event](#) \* [GiveMe\\_Event](#) (const int &iEvent, const int &ModuleNber\_Input, const int &Data\_to\_Use)
- void [SetBeforeMerging](#) (const int &i\_SetBeforeMerging)
- void [Setwap\\_XY](#) (const int &i\_Swap\_XY)
- void [Init](#) ()

## Public Member Functions inherited from Uploader

- Uploader (const std::string &SampleFile, Model\_ReadOutGeometry \*pModel\_ReadOutGeometry, Model\_Electronics \*pModel\_Electronics, Model\_Charge1D \*pModel\_Charge1D)
- virtual ~Uploader ()
- std::string Get\_SampleFile ()
- int Get\_NberOfEvent ()
- int Get\_PRF\_exist ()
- double Get\_Norm ()
- double Get\_a2 ()
- double Get\_a4 ()
- double Get\_b2 ()
- double Get\_b4 ()
- Model\_ReadOutGeometry \* Get\_Model\_ReadOutGeometry ()
- Model\_Electronics \* Get\_Model\_Electronics ()
- Model\_Charge1D \* Get\_Model\_Charge1D ()

## Public Attributes

- Int\_t fCurrent
- Int\_t ev
- Int\_t track
- ULong64\_t date
- Bool\_t beforeMerging
- Double\_t dEdx
- Double\_t angle\_yz
- Double\_t angle\_xy
- Int\_t rob\_clusters
- Double\_t quality
- Double\_t mom
- Double\_t sina
- Double\_t offset
- Int\_t max\_mult
- Double\_t mean\_mult
- std::vector< int > \* multiplicity
- std::vector< int > \* charge
- std::vector< double > \* residual
- std::vector< double > \* residual\_corr
- std::vector< std::vector< double > > \* dx
- std::vector< std::vector< double > > \* qfrac
- std::vector< std::vector< int > > \* time
- std::vector< double > \* clust\_pos
- std::vector< double > \* clust\_pos\_err
- std::vector< double > \* track\_pos
- std::vector< int > \* module
- std::vector< std::vector< int > > \* pad\_charge
- std::vector< std::vector< int > > \* pad\_time
- std::vector< std::vector< int > > \* wf\_width
- std::vector< std::vector< int > > \* wf\_fwhm
- std::vector< std::vector< int > > \* pad\_x
- std::vector< std::vector< int > > \* pad\_y
- std::vector< std::vector< std::vector< int > > > \* pad\_wf\_q
- TBranch \* b\_ev
- TBranch \* b\_track

- TBranch \* [b\\_date](#)
- TBranch \* [b\\_beforeMerging](#)
- TBranch \* [b\\_dEdx](#)
- TBranch \* [b\\_angle\\_yz](#)
- TBranch \* [b\\_angle\\_xy](#)
- TBranch \* [b\\_rob\\_clusters](#)
- TBranch \* [b\\_quality](#)
- TBranch \* [b\\_mom](#)
- TBranch \* [b\\_sina](#)
- TBranch \* [b\\_offset](#)
- TBranch \* [b\\_max\\_mult](#)
- TBranch \* [b\\_mean\\_mult](#)
- TBranch \* [b\\_multiplicity](#)
- TBranch \* [b\\_charge](#)
- TBranch \* [b\\_residual](#)
- TBranch \* [b\\_residual\\_corr](#)
- TBranch \* [b\\_dx](#)
- TBranch \* [b\\_qfrac](#)
- TBranch \* [b\\_time](#)
- TBranch \* [b\\_clust\\_pos](#)
- TBranch \* [b\\_clust\\_pos\\_err](#)
- TBranch \* [b\\_track\\_pos](#)
- TBranch \* [b\\_module](#)
- TBranch \* [b\\_pad\\_charge](#)
- TBranch \* [b\\_pad\\_time](#)
- TBranch \* [b\\_wf\\_width](#)
- TBranch \* [b\\_wf\\_fwhm](#)
- TBranch \* [b\\_pad\\_x](#)
- TBranch \* [b\\_pad\\_y](#)
- TBranch \* [b\\_pad\\_wf\\_q](#)

#### Additional Inherited Members

#### Protected Attributes inherited from Uploader

- std::string [m\\_SampleFile](#)
- int [m\\_NberOfEvent](#)
- TFile \* [p\\_TFile](#)
- TTree \* [p\\_TTree](#)
- Model\_ReadOutGeometry \* [p\\_Model\\_ReadOutGeometry](#)
- Model\_Electronics \* [p\\_Model\\_Electronics](#)
- Model\_Charge1D \* [p\\_Model\\_Charge1D](#)
- short int [m\\_PRF\\_exist](#)
- double [m\\_Norm](#)
- double [m\\_a2](#)
- double [m\\_a4](#)
- double [m\\_b2](#)
- double [m\\_b4](#)

### 8.39.1 Constructor & Destructor Documentation

#### 8.39.1.1 Uploader\_MockUp\_V2()

```
Uploader_MockUp_V2::Uploader_MockUp_V2 (
    const std::string & SampleFile,
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_Charge1D * pModel_Charge1D)
```

Constructor

#### 8.39.1.2 ~Uploader\_MockUp\_V2()

```
Uploader_MockUp_V2::~Uploader_MockUp_V2 () [virtual]
```

### 8.39.2 Member Function Documentation

#### 8.39.2.1 GiveMe\_Event() [1/2]

```
Event * Uploader_MockUp_V2::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use) [virtual]
```

#### 8.39.2.2 GiveMe\_Event() [2/2]

```
Event * Uploader_MockUp_V2::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use,
    const int & CloseWF) [virtual]
```

Implements [Uploader](#).

#### 8.39.2.3 Init()

```
void Uploader_MockUp_V2::Init ()
```

#### 8.39.2.4 SetBeforeMerging()

```
void Uploader_MockUp_V2::SetBeforeMerging (
    const int & i_SetBeforeMerging)
```

#### 8.39.2.5 Setwap\_XY()

```
void Uploader_MockUp_V2::Setwap_XY (
    const int & i_Swap_XY)
```

### 8.39.3 Member Data Documentation

#### 8.39.3.1 **angle\_xy**

```
Double_t Uploader_MockUp_V2::angle_xy
```

#### 8.39.3.2 **angle\_yz**

```
Double_t Uploader_MockUp_V2::angle_yz
```

#### 8.39.3.3 **b\_angle\_xy**

```
TBranch* Uploader_MockUp_V2::b_angle_xy
```

#### 8.39.3.4 **b\_angle\_yz**

```
TBranch* Uploader_MockUp_V2::b_angle_yz
```

#### 8.39.3.5 **b\_beforeMerging**

```
TBranch* Uploader_MockUp_V2::b_beforeMerging
```

#### 8.39.3.6 **b\_charge**

```
TBranch* Uploader_MockUp_V2::b_charge
```

#### 8.39.3.7 **b\_clust\_pos**

```
TBranch* Uploader_MockUp_V2::b_clust_pos
```

#### 8.39.3.8 **b\_clust\_pos\_err**

```
TBranch* Uploader_MockUp_V2::b_clust_pos_err
```

#### 8.39.3.9 **b\_date**

```
TBranch* Uploader_MockUp_V2::b_date
```

#### 8.39.3.10 **b\_dEdx**

```
TBranch* Uploader_MockUp_V2::b_dEdx
```

**8.39.3.11 b\_dx**

```
TBranch* Uploader_MockUp_V2::b_dx
```

**8.39.3.12 b\_ev**

```
TBranch* Uploader_MockUp_V2::b_ev
```

**8.39.3.13 b\_max\_mult**

```
TBranch* Uploader_MockUp_V2::b_max_mult
```

**8.39.3.14 b\_mean\_mult**

```
TBranch* Uploader_MockUp_V2::b_mean_mult
```

**8.39.3.15 b\_module**

```
TBranch* Uploader_MockUp_V2::b_module
```

**8.39.3.16 b\_mom**

```
TBranch* Uploader_MockUp_V2::b_mom
```

**8.39.3.17 b\_multiplicity**

```
TBranch* Uploader_MockUp_V2::b_multiplicity
```

**8.39.3.18 b\_offset**

```
TBranch* Uploader_MockUp_V2::b_offset
```

**8.39.3.19 b\_pad\_charge**

```
TBranch* Uploader_MockUp_V2::b_pad_charge
```

**8.39.3.20 b\_pad\_time**

```
TBranch* Uploader_MockUp_V2::b_pad_time
```

**8.39.3.21 b\_pad\_wf\_q**

```
TBranch* Uploader_MockUp_V2::b_pad_wf_q
```

**8.39.3.22 b\_pad\_x**

```
TBranch* Uploader_MockUp_V2::b_pad_x
```

**8.39.3.23 b\_pad\_y**

```
TBranch* Uploader_MockUp_V2::b_pad_y
```

**8.39.3.24 b\_qfrac**

```
TBranch* Uploader_MockUp_V2::b_qfrac
```

**8.39.3.25 b\_quality**

```
TBranch* Uploader_MockUp_V2::b_quality
```

**8.39.3.26 b\_residual**

```
TBranch* Uploader_MockUp_V2::b_residual
```

**8.39.3.27 b\_residual\_corr**

```
TBranch* Uploader_MockUp_V2::b_residual_corr
```

**8.39.3.28 b\_rob\_clusters**

```
TBranch* Uploader_MockUp_V2::b_rob_clusters
```

**8.39.3.29 b\_sina**

```
TBranch* Uploader_MockUp_V2::b_sina
```

**8.39.3.30 b\_time**

```
TBranch* Uploader_MockUp_V2::b_time
```

**8.39.3.31 b\_track**

```
TBranch* Uploader_MockUp_V2::b_track
```

**8.39.3.32 b\_track\_pos**

```
TBranch* Uploader_MockUp_V2::b_track_pos
```

**8.39.3.33 b\_wf\_fwhm**

```
TBranch* Uploader_MockUp_V2::b_wf_fwhm
```

**8.39.3.34 b\_wf\_width**

```
TBranch* Uploader_MockUp_V2::b_wf_width
```

**8.39.3.35 beforeMerging**

```
Bool_t Uploader_MockUp_V2::beforeMerging
```

**8.39.3.36 charge**

```
std::vector<int>* Uploader_MockUp_V2::charge
```

**8.39.3.37 clust\_pos**

```
std::vector<double>* Uploader_MockUp_V2::clust_pos
```

**8.39.3.38 clust\_pos\_err**

```
std::vector<double>* Uploader_MockUp_V2::clust_pos_err
```

**8.39.3.39 date**

```
ULong64_t Uploader_MockUp_V2::date
```

**8.39.3.40 dEdx**

```
Double_t Uploader_MockUp_V2::dEdx
```

**8.39.3.41 dx**

```
std::vector<std::vector<double>>* Uploader_MockUp_V2::dx
```

**8.39.3.42 ev**

```
Int_t Uploader_MockUp_V2::ev
```

**8.39.3.43 fCurrent**

```
Int_t Uploader_MockUp_V2::fCurrent
```

**8.39.3.44 max\_mult**

```
Int_t Uploader_MockUp_V2::max_mult
```

**8.39.3.45 mean\_mult**

```
Double_t Uploader_MockUp_V2::mean_mult
```

**8.39.3.46 module**

```
std::vector<int>* Uploader_MockUp_V2::module
```

**8.39.3.47 mom**

```
Double_t Uploader_MockUp_V2::mom
```

**8.39.3.48 multiplicity**

```
std::vector<int>* Uploader_MockUp_V2::multiplicity
```

**8.39.3.49 offset**

```
Double_t Uploader_MockUp_V2::offset
```

**8.39.3.50 pad\_charge**

```
std::vector<std::vector<int>>* Uploader_MockUp_V2::pad_charge
```

**8.39.3.51 pad\_time**

```
std::vector<std::vector<int>>* Uploader_MockUp_V2::pad_time
```

**8.39.3.52 pad\_wf\_q**

```
std::vector<std::vector<std::vector<int>>>* Uploader_MockUp_V2::pad_wf_q
```

**8.39.3.53 pad\_x**

```
std::vector<std::vector<int>>* Uploader_MockUp_V2::pad_x
```

**8.39.3.54 pad\_y**

```
std::vector<std::vector<int>>* Uploader_MockUp_V2::pad_y
```

**8.39.3.55 qfrac**

```
std::vector<std::vector<double>>* Uploader_MockUp_V2::qfrac
```

**8.39.3.56 quality**

```
Double_t Uploader_MockUp_V2::quality
```

**8.39.3.57 residual**

```
std::vector<double>* Uploader_MockUp_V2::residual
```

**8.39.3.58 residual\_corr**

```
std::vector<double>* Uploader_MockUp_V2::residual_corr
```

**8.39.3.59 rob\_clusters**

```
Int_t Uploader_MockUp_V2::rob_clusters
```

**8.39.3.60 sina**

```
Double_t Uploader_MockUp_V2::sina
```

### 8.39.3.61 time

```
std::vector<std::vector<int> *>* Uploader_MockUp_V2::time
```

### 8.39.3.62 track

```
Int_t Uploader_MockUp_V2::track
```

### 8.39.3.63 track\_pos

```
std::vector<double>*>* Uploader_MockUp_V2::track_pos
```

### 8.39.3.64 wf\_fwhm

```
std::vector<std::vector<int> *>* Uploader_MockUp_V2::wf_fwhm
```

### 8.39.3.65 wf\_width

```
std::vector<std::vector<int> *>* Uploader_MockUp_V2::wf_width
```

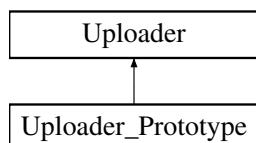
The documentation for this class was generated from the following files:

- Uploader/inc/[Uploader\\_MockUp\\_V2.h](#)
- Uploader/src/[Uploader\\_MockUp\\_V2.cxx](#)

## 8.40 Uploader\_Prototype Class Reference

```
#include <Uploader_Prototype.h>
```

Inheritance diagram for Uploader\_Prototype:



### Public Member Functions

- [`Uploader\_Prototype`](#) (const std::string &SampleFile, [Model\\_ReadOutGeometry](#) \*pModel\_ReadOutGeometry, [Model\\_Electronics](#) \*pModel\_Electronics, [Model\\_Charge1D](#) \*pModel\_Charge1D)
- virtual [`~Uploader\_Prototype`](#) ()
- virtual [`Event \* GiveMe\_Event`](#) (const int &iEvent, const int &ModuleNber\_Input, const int &Data\_to\_Use, const int &CloseWF)
- virtual [`Event \* GiveMe\_Event`](#) (const int &iEvent, const int &ModuleNber\_Input, const int &Data\_to\_Use)

## Public Member Functions inherited from Uploader

- Uploader (const std::string &SampleFile, Model\_ReadOutGeometry \*pModel\_ReadOutGeometry, Model\_Electronics \*pModel\_Electronics, Model\_Charge1D \*pModel\_Charge1D)
- virtual ~Uploader ()
- std::string Get\_SampleFile ()
- int Get\_NberOfEvent ()
- int Get\_PRF\_exist ()
- double Get\_Norm ()
- double Get\_a2 ()
- double Get\_a4 ()
- double Get\_b2 ()
- double Get\_b4 ()
- Model\_ReadOutGeometry \* Get\_Model\_ReadOutGeometry ()
- Model\_Electronics \* Get\_Model\_Electronics ()
- Model\_Charge1D \* Get\_Model\_Charge1D ()

## Additional Inherited Members

## Protected Attributes inherited from Uploader

- std::string m\_SampleFile
- int m\_NberOfEvent
- TFile \* p\_TFile
- TTree \* p\_TTree
- Model\_ReadOutGeometry \* p\_Model\_ReadOutGeometry
- Model\_Electronics \* p\_Model\_Electronics
- Model\_Charge1D \* p\_Model\_Charge1D
- short int m\_PRF\_exist
- double m\_Norm
- double m\_a2
- double m\_a4
- double m\_b2
- double m\_b4

## 8.40.1 Constructor & Destructor Documentation

### 8.40.1.1 Uploader\_Protoype()

```
Uploader_Protoype::Uploader_Protoype (
    const std::string & SampleFile,
    Model_ReadOutGeometry * pModel_ReadOutGeometry,
    Model_Electronics * pModel_Electronics,
    Model_Charge1D * pModel_Charge1D)
```

Constructor

### 8.40.1.2 ~Uploader\_Protoype()

```
Uploader_Protoype::~Uploader_Protoype () [virtual]
```

## 8.40.2 Member Function Documentation

### 8.40.2.1 GiveMe\_Event() [1/2]

```
Event * Uploader_Prototype::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use)  [virtual]
```

### 8.40.2.2 GiveMe\_Event() [2/2]

```
Event * Uploader_Prototype::GiveMe_Event (
    const int & iEvent,
    const int & ModuleNber_Input,
    const int & Data_to_Use,
    const int & CloseWF)  [virtual]
```

Implements [Uploader](#).

The documentation for this class was generated from the following files:

- Uploader/inc/[Uploader\\_Prototype.h](#)
- Uploader/src/[Uploader\\_Prototype.cxx](#)

# Chapter 9

## File Documentation

### 9.1 Apps/Analysis.cxx File Reference

```
#include <iomanip>
#include <iostream>
#include "Monitoring.h"
```

#### Functions

- int `main` (int argc, char \*argv[ ])

#### 9.1.1 Function Documentation

##### 9.1.1.1 `main()`

```
int main (
    int argc,
    char * argv[])
```

### 9.2 Apps/DrawOutcxx File Reference

```
#include "DrawOuts.h"
```

#### Functions

- int `main` (int argc, char \*argv[ ])

## 9.2.1 Function Documentation

### 9.2.1.1 main()

```
int main (
    int argc,
    char * argv[ ])
```

## 9.3 Fitters/inc/ClusterFitter.h File Reference

Declarations for cluster-level fitters used to fit pad-cluster charge distributions.

```
#include "Misc.h"
#include "Cluster.h"
#include "Event.h"
#include "Sample.h"
#include "TMath.h"
#include "TMatrixD.h"
#include "TMinuit.h"
#include "TVirtualFitter.h"
#include "TF1.h"
```

### Classes

- class [ClusterFitter\\_Horizontal](#)
- class [ClusterFitter\\_Diagonal](#)

### Functions

- void [ClusterFit\\_Horizontal](#) ([Sample](#) &aSample, const int &ModuleNber, [TF1](#) \*pTF1\_ToBeUsed)
- void [ClusterFit\\_Horizontal\\_Event](#) ([Event](#) \*pEvent, const int &ModuleNber, [TF1](#) \*pTF1\_ToBeUsed, [ClusterFitter\\_Horizontal](#) &aClusterFitter\_Horizontal)
- void [ClusterFit\\_Horizontal\\_Cluster](#) ([Cluster](#) \*pCluster, const int &ModuleNber, [TF1](#) \*pTF1\_ToBeUsed, [ClusterFitter\\_Horizontal](#) &aClusterFitter\_Horizontal)
- void [ClusterFit\\_Diagonal](#) (const double &AngleRot, [Sample](#) &aSample, const int &ModuleNber, [TF1](#) \*pTF1\_ToBeUsed)
- void [ClusterFit\\_Diagonal\\_Event](#) (const double &AngleRot, [Event](#) \*pEvent, const int &ModuleNber, [TF1](#) \*pTF1\_ToBeUsed, int &Kounter\_Fit, int &Kounter\_Failure, [ClusterFitter\\_Diagonal](#) &aClusterFitter\_Diagonal)
- void [ClusterFit\\_Diagonal\\_Cluster](#) (const double &AngleRot, [Cluster](#) \*pCluster, const int &ModuleNber, [TF1](#) \*pTF1\_ToBeUsed, int &Kounter\_Fit, int &Kounter\_Failure, [ClusterFitter\\_Diagonal](#) &aClusterFitter\_Diagonal)

### 9.3.1 Detailed Description

Declarations for cluster-level fitters used to fit pad-cluster charge distributions.

This header defines both horizontal and diagonal cluster fitter classes and related helper functions to run fits on Samples, Events and Clusters. Implementations live in [ClusterFitter.cxx](#).

The corresponding implementation uses ROOT's TVirtualFitter/TMinuit and registers file-local static bridge classes that forward Minuit callbacks to the instance methods (Chi2 evaluation). The source contains both the horizontal and diagonal fitter implementations and several helper functions to apply filters to Samples, Events and individual Clusters.

### 9.3.2 Function Documentation

#### 9.3.2.1 ClusterFit\_Diagonal()

```
void ClusterFit_Diagonal (
    const double & AngleRot,
    Sample & aSample,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed)
```

#### 9.3.2.2 ClusterFit\_Diagonal\_Cluster()

```
void ClusterFit_Diagonal_Cluster (
    const double & AngleRot,
    Cluster * pCluster,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    int & Kounter_Fit,
    int & Kounter_Failure,
    ClusterFitter_Diagonal & aClusterFitter_Diagonal)
```

#### 9.3.2.3 ClusterFit\_Diagonal\_Event()

```
void ClusterFit_Diagonal_Event (
    const double & AngleRot,
    Event * pEvent,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    int & Kounter_Fit,
    int & Kounter_Failure,
    ClusterFitter_Diagonal & aClusterFitter_Diagonal)
```

#### 9.3.2.4 ClusterFit\_Horizontal()

```
void ClusterFit_Horizontal (
    Sample & aSample,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed)
```

#### 9.3.2.5 ClusterFit\_Horizontal\_Cluster()

```
void ClusterFit_Horizontal_Cluster (
    Cluster * pCluster,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    ClusterFitter_Horizontal & aClusterFitter_Horizontal)
```

### 9.3.2.6 ClusterFit\_Horizontal\_Event()

```
void ClusterFit_Horizontal_Event (
    Event * pEvent,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    ClusterFitter_Horizontal & aClusterFitter_Horizontal)
```

## 9.4 ClusterFitter.h

[Go to the documentation of this file.](#)

```
00001
00017 #ifndef CLUSTER_FIT_H
00018 #define CLUSTER_FIT_H
00019
00020 #include "Misc.h"
00021
00022 #include "Cluster.h"
00023 #include "Event.h"
00024 #include "Sample.h"
00025
00026 #include "TMath.h"
00027 #include "TMatrixD.h"
00028 #include "TMinuit.h"
00029 #include "TVirtualFitter.h"
00030
00031 #include "TF1.h"
00032
00033 // HORIZONTAL FITTER
00034 // Fit Basic: MIGRAD/MINOS
00035 //
00036 // Fit Rescues:
00037 // 1: MIGRAD->MINIMIZE/MIGRAD
00038 // 2: MIGRAD->SIMPLEX/MIGRAD
00039 //
00040 // Fit failure -> Failure procedure called
00041 //
00042 class ClusterFitter_Horizontal {
00043 public:
00044     ClusterFitter_Horizontal(const std::string &FitterName);
00045
00046     virtual ~ClusterFitter_Horizontal();
00047
00048     public:
00049         void Set_Cluster(Cluster *pCluster);
00050
00051         int DoMinimisation();
00052
00053         double Chi2(double par[]);
00054
00055     protected:
00056         std::string m_FitterName;
00057         Cluster *p_Cluster;
00058         TVirtualFitter *p_TVirtualFitter;
00059     };
00060
00061
00062 void ClusterFit_Horizontal(Sample &aSample, const int &ModuleNber,
00063                             TF1 *pTF1_ToBeUsed);
00064 void ClusterFit_Horizontal_Event(
00065     Event *pEvent, const int &ModuleNber, TF1 *pTF1_ToBeUsed,
00066     ClusterFitter_Horizontal &aClusterFitter_Horizontal);
00067 void ClusterFit_Horizontal_Cluster(
00068     Cluster *pCluster, const int &ModuleNber, TF1 *pTF1_ToBeUsed,
00069     ClusterFitter_Horizontal &aClusterFitter_Horizontal);
00070
00071
00072 // DIAGONAL FITTER
00073 // Fit Basic: MIGRAD/MINOS
00074 //
00075 // Fit Rescues:
00076 // 1: MIGRAD->MINIMIZE/MIGRAD
00077 // 2: MIGRAD->SIMPLEX/MIGRAD
00078 //
00079 // Fit failure -> Failure procedure called
00080 //
00081 class ClusterFitter_Diagonal {
00082 public:
```

```

00107     ClusterFitter_Diagonal(const std::string &FitterName);
00108
00112     virtual ~ClusterFitter_Diagonal();
00113
00114 public:
00119     void Set_Cluster(Cluster *pCluster);
00120
00125     int DoMinimisation();
00126
00132     double Chi2(double par[]);
00133
00134 protected:
00135     std::string m_FitterName;
00136     Cluster *p_Cluster;
00137     TVirtualFitter *p_TVirtualFitter;
00138 };
00139
00140 //-----Cluster
00141 // Fits-----
00142 // Cluster Fits
00143
00144 void ClusterFit_Diagonal(const double &AngleRot, Sample &aSample,
00145                           const int &ModuleNber, TF1 *pTF1_ToBeUsed);
00146 void ClusterFit_Diagonal_Event(const double &AngleRot, Event *pEvent,
00147                                 const int &ModuleNber, TF1 *pTF1_ToBeUsed,
00148                                 int &Kounter_Fit, int &Kounter_Failure,
00149                                 ClusterFitter_Diagonal &aClusterFitter_Diagonal);
00150 void ClusterFit_Diagonal_Cluster(
00151     const double &AngleRot, Cluster *pCluster, const int &ModuleNber,
00152     TF1 *pTF1_ToBeUsed, int &Kounter_Fit, int &Kounter_Failure,
00153     ClusterFitter_Diagonal &aClusterFitter_Diagonal);
00154
00155 #endif

```

## 9.5 Fitters/inc/PRFParameters.h File Reference

Parameter container and evaluator for the [Pad](#) Response Function (PRF).

```
#include "Misc.h"
```

### Classes

- class [PRFParameters](#)

#### 9.5.1 Detailed Description

Parameter container and evaluator for the [Pad](#) Response Function (PRF).

Declares the [PRFParameters](#) class which stores polynomial coefficients used to evaluate a PRF model used by cluster/track fitting. Implementation is in [PRFParameters.cxx](#).

The implementation provides Eval, SetPRF and operator() so the object can be used directly as a functor with ROOT's TF1. It stores the polynomial coefficients and provides a small, stable interface for fitters and waveform utilities.

## 9.6 PRFParameters.h

[Go to the documentation of this file.](#)

```

00001 #
00015 #ifndef PRFParameters_H
00016 #define PRFParameters_H
00017
00018 #include "Misc.h"
00019
00021 class PRFParameters {
00022 public:
00026     PRFParameters();
00027
00031     virtual ~PRFParameters();
00032
00038     double Eval(const double &Xin);
00039
00043     double operator()(double *x, double *par);
00044
00053     void SetPRF(const double &Norm, const double &A2, const double &A4,
00054                 const double &B2, const double &B4);
00055
00056 private:
00057     double m_Norm;
00058     double m_A2;
00059     double m_A4;
00060     double m_B2;
00061     double m_B4;
00062 };
00063
00064 #endif

```

## 9.7 Fitters/inc/TrackFitter.h File Reference

Declaration of the [TrackFitter](#) class used to perform track fits.

```

#include "Misc.h"
#include "Track.h"
#include "TMath.h"
#include "TMatrixD.h"
#include "TMinuit.h"
#include "TVirtualFitter.h"

```

### Classes

- class [TrackFitter](#)

### 9.7.1 Detailed Description

Declaration of the [TrackFitter](#) class used to perform track fits.

This header defines the [TrackFitter](#) interface which wraps a minimiser (ROOT's TVirtualFitter/TMinuit) and exposes methods to set the track, run minimisation and compute the chi2. The implementation lives in [TrackFitter.cxx](#).

Implementation notes: the corresponding source file, [TrackFitter.cxx](#), contains the concrete interfacing code to ROOT's minimisers and the static bridge used by Minuit (a file-local helper class that forwards the minimiser callbacks to a [TrackFitter](#) instance). The Chi2 evaluation routine and Minuit callback registration are performed in the source implementation. Care must be taken not to change the global/static bridge semantics as Minuit requires a static callback wrapper.

## 9.8 TrackFitter.h

[Go to the documentation of this file.](#)

```

00001
00019
00020 #ifndef TrackFitter_H
00021 #define TrackFitter_H
00022
00023 #include "Misc.h"
00024
00025 #include "Track.h"
00026
00027 #include "TMath.h"
00028 #include "TMatrixD.h"
00029 #include "TMinuit.h"
00030 #include "TVirtualFitter.h"
00031
00032 //
00033 // Parameters of fit should be unconstrained
00034 //
00035 // Fit Basic: MIGRAD/MINOS
00036 //
00037 // Fit Rescues:
00038 // 1: MIGRAD->MINIMIZE/MIGRAD
00039 // 2: MIGRAD->SIMPLEX/MIGRAD
00040 //
00041 // Fit failure -> abort
00042 //
00043 class TrackFitter {
00044 public:
00050     TrackFitter(const std::string &FitterName, const int &NberOfParameters = 2);
00051     virtual ~TrackFitter();
00056
00057 protected:
00059     void Set_Track(Track *pTrack);
00065     int DoMinimisation();
00071     double Chi2(double par[]);
00077
00079
00081     std::string m_FitterName;
00083
00084     int m_NberOfParameters;
00085
00086     Track *p_Track;
00087
00088     TVirtualFitter *p_TVirtualFitter;
00089 };
00090
00091 #endif

```

## 9.9 Fitters/inc/TrackRecon.h File Reference

Declarations for track reconstruction utilities and helpers.

```
#include "Misc.h"
#include "Sample.h"
#include "TrackFitter.h"
```

### Functions

- void [TrackRecon](#) ([Sample](#) &aSample, const int &ModuleNber, const int &NberOfParameters=3)  
*Run track reconstruction over a sample.*
- int [TrackRecon\\_Event](#) ([TrackFitter](#) &aTrackFitter, [Event](#) \*pEvent, const int &ModuleNber, const int &NberOfParameters=3)  
*Run track reconstruction for a single event using a [TrackFitter](#).*

### 9.9.1 Detailed Description

Declarations for track reconstruction utilities and helpers.

This header exposes the TrackRecon top-level functions that run reconstruction over a [Sample](#) or an [Event](#) using a [TrackFitter](#). Implementation is in [TrackRecon.hxx](#).

The implementation orchestrates fitting over Samples and Events and uses the [TrackFitter](#) API to run per-track minimisations. It contains the top-level control loops that increment fit/failure counters and report simple summaries after processing.

### 9.9.2 Function Documentation

#### 9.9.2.1 TrackRecon()

```
void TrackRecon (
    Sample & aSample,
    const int & ModuleNber,
    const int & NberOfParameters = 3)
```

Run track reconstruction over a sample.

##### Parameters

<i>aSample</i>	reference to the input <a href="#">Sample</a>
<i>ModuleNber</i>	module index
<i>NberOfParameters</i>	number of fit parameters (default: 3)

#### 9.9.2.2 TrackRecon\_Event()

```
int TrackRecon_Event (
    TrackFitter & aTrackFitter,
    Event * pEvent,
    const int & ModuleNber,
    const int & NberOfParameters = 3)
```

Run track reconstruction for a single event using a [TrackFitter](#).

##### Parameters

<i>aTrackFitter</i>	reference to an existing <a href="#">TrackFitter</a> instance
<i>pEvent</i>	pointer to the <a href="#">Event</a> to process
<i>ModuleNber</i>	module index
<i>NberOfParameters</i>	number of fit parameters (default: 3)

##### Returns

integer status (0 on success)

## 9.10 TrackRecon.h

[Go to the documentation of this file.](#)

```
00001 #
00015 #include "Misc.h"
00016
00017 #include "Sample.h"
00018
00019 #include "TrackFitter.h"
00020
00027 void TrackRecon(Sample &aSample, const int &ModuleNber,
00028           const int &NberOfParameters = 3);
00029
00038 int TrackRecon_Event(TrackFitter &aTrackFitter, Event *pEvent,
00039           const int &ModuleNber, const int &NberOfParameters = 3);
```

## 9.11 Fitters/src/ClusterFitter.cxx File Reference

```
#include "ClusterFitter.h"
#include "TFitter.h"
#include "Util.h"
```

### Classes

- class [StaticClusterFitter\\_Horizontal](#)
- class [StaticClusterFitter\\_Diagonal](#)

### Functions

- void [ClusterFit\\_Horizontal](#) (Sample &aSample, const int &ModuleNber, TF1 \*pTF1\_ToBeUsed)
- void [ClusterFit\\_Horizontal\\_Event](#) (Event \*pEvent, const int &ModuleNber, TF1 \*pTF1\_ToBeUsed, ClusterFitter\_Horizontal &aClusterFitter\_Horizontal)
- void [ClusterFit\\_Horizontal\\_Cluster](#) (Cluster \*pCluster, const int &ModuleNber, TF1 \*pTF1\_ToBeUsed, ClusterFitter\_Horizontal &aClusterFitter\_Horizontal)
- void [ClusterFitter\\_HorizontalFunction](#) (int &nDim, double \*gout, double &result, double par[], int flg)
- void [ClusterFit\\_Diagonal](#) (const double &AngleRot, Sample &aSample, const int &ModuleNber, TF1 \*pTF1\_ToBeUsed)
- void [ClusterFit\\_Diagonal\\_Event](#) (const double &AngleRot, Event \*pEvent, const int &ModuleNber, TF1 \*pTF1\_ToBeUsed, int &Kounter\_Fit, int &Kounter\_Failure, ClusterFitter\_Diagonal &aClusterFitter\_Diagonal)
- void [ClusterFit\\_Diagonal\\_Cluster](#) (const double &AngleRot, Cluster \*pCluster, const int &ModuleNber, TF1 \*pTF1\_ToBeUsed, int &Kounter\_Fit, int &Kounter\_Failure, ClusterFitter\_Diagonal &aClusterFitter\_Diagonal)
- void [ClusterFitter\\_DiagonalFunction](#) (int &nDim, double \*gout, double &result, double par[], int flg)

### 9.11.1 Function Documentation

#### 9.11.1.1 ClusterFit\_Diagonal()

```
void ClusterFit_Diagonal (
    const double & AngleRot,
    Sample & aSample,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed)
```

### 9.11.1.2 ClusterFit\_Diagonal\_Cluster()

```
void ClusterFit_Diagonal_Cluster (
    const double & AngleRot,
    Cluster * pCluster,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    int & Kounter_Fit,
    int & Kounter_Failure,
    ClusterFitter_Diagonal & aClusterFitter_Diagonal)
```

### 9.11.1.3 ClusterFit\_Diagonal\_Event()

```
void ClusterFit_Diagonal_Event (
    const double & AngleRot,
    Event * pEvent,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    int & Kounter_Fit,
    int & Kounter_Failure,
    ClusterFitter_Diagonal & aClusterFitter_Diagonal)
```

### 9.11.1.4 ClusterFit\_Horizontal()

```
void ClusterFit_Horizontal (
    Sample & aSample,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed)
```

### 9.11.1.5 ClusterFit\_Horizontal\_Cluster()

```
void ClusterFit_Horizontal_Cluster (
    Cluster * pCluster,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    ClusterFitter_Horizontal & aClusterFitter_Horizontal)
```

### 9.11.1.6 ClusterFit\_Horizontal\_Event()

```
void ClusterFit_Horizontal_Event (
    Event * pEvent,
    const int & ModuleNber,
    TF1 * pTF1_ToBeUsed,
    ClusterFitter_Horizontal & aClusterFitter_Horizontal)
```

### 9.11.1.7 ClusterFitter\_DiagonalFunction()

```
void ClusterFitter_DiagonalFunction (
    int & nDim,
    double * gout,
    double & result,
    double par[],
    int flg)
```

### 9.11.1.8 ClusterFitter\_HorizontalFunction()

```
void ClusterFitter_HorizontalFunction (
    int & nDim,
    double * gout,
    double & result,
    double par[],
    int flg)
```

## 9.12 Fitters/src/PRFParameters.cxx File Reference

```
#include "PRFParameters.h"
```

## 9.13 Fitters/src/TrackFittercxx File Reference

```
#include "TrackFitter.h"
#include "TFitter.h"
```

### Classes

- class [StaticTrackFitter](#)

### Functions

- void [TrackFitterFunction](#) (int &nDim, double \*gout, double &result, double par[], int flg)

### 9.13.1 Function Documentation

#### 9.13.1.1 TrackFitterFunction()

```
void TrackFitterFunction (
    int & nDim,
    double * gout,
    double & result,
    double par[],
    int flg)
```

## 9.14 Fitters/src/TrackReconcxx File Reference

```
#include "TrackRecon.h"
#include "Cluster.h"
#include "Event.h"
#include "Track.h"
```

## Functions

- void [TrackRecon](#) ([Sample](#) &aSample, const int &ModuleNber, const int &NberOfParameters)  
*Run track reconstruction over a sample.*
- int [TrackRecon\\_Event](#) ([TrackFitter](#) &aTrackFitter, [Event](#) \*pEvent, const int &ModuleNber, const int &NberOfParameters)  
*Run track reconstruction for a single event using a [TrackFitter](#).*

### 9.14.1 Function Documentation

#### 9.14.1.1 [TrackRecon\(\)](#)

```
void TrackRecon (
    Sample & aSample,
    const int & ModuleNber,
    const int & NberOfParameters = 3)
```

Run track reconstruction over a sample.

##### Parameters

<i>aSample</i>	reference to the input <a href="#">Sample</a>
<i>ModuleNber</i>	module index
<i>NberOfParameters</i>	number of fit parameters (default: 3)

#### 9.14.1.2 [TrackRecon\\_Event\(\)](#)

```
int TrackRecon_Event (
    TrackFitter & aTrackFitter,
    Event * pEvent,
    const int & ModuleNber,
    const int & NberOfParameters = 3)
```

Run track reconstruction for a single event using a [TrackFitter](#).

##### Parameters

<i>aTrackFitter</i>	reference to an existing <a href="#">TrackFitter</a> instance
<i>pEvent</i>	pointer to the <a href="#">Event</a> to process
<i>ModuleNber</i>	module index
<i>NberOfParameters</i>	number of fit parameters (default: 3)

##### Returns

integer status (0 on success)

## 9.15 ModelEvent/inc/AmplitudeError.h File Reference

Amplitude error model utilities.

```
#include "Misc.h"
#include "Pad.h"
```

### Functions

- double AmplitudeError (Pad \*pPad, Pad \*pPad\_Leading, Pad \*pPad\_NextLeading, Pad \*pPad\_NextNextLeading)

### 9.15.1 Detailed Description

Amplitude error model utilities.

Declares helpers to compute amplitude uncertainties for pads and clusters used during fitting and dE/dx estimation.

### 9.15.2 Function Documentation

#### 9.15.2.1 AmplitudeError()

```
double AmplitudeError (
    Pad * pPad,
    Pad * pPad_Leading,
    Pad * pPad_NextLeading,
    Pad * pPad_NextNextLeading)
```

## 9.16 AmplitudeError.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #ifndef AmplitudeError_H
00009 #define AmplitudeError_H
00010
00011 #include "Misc.h"
00012
00013 #include "Pad.h"
00014
00015 double AmplitudeError(Pad *pPad, Pad *pPad_Leading, Pad *pPad_NextLeading,
00016                         Pad *pPad_NextNextLeading);
00017
00018 #endif
```

## 9.17 ModelEvent/inc/Cluster.h File Reference

**Cluster** container representing a group of pads forming a detected charge.

```
#include "Misc.h"
#include "Pad.h"
#include "FitOutput.h"
#include "TF1.h"
#include "TMatrixD.h"
#include "TVirtualFitter.h"
```

## Classes

- class [Cluster](#)

### 9.17.1 Detailed Description

[Cluster](#) container representing a group of pads forming a detected charge.

The [Cluster](#) class provides accessors and utilities used by fitting and reconstruction code.

## 9.18 Cluster.h

[Go to the documentation of this file.](#)

```

00001 #
00008 #ifndef Cluster_H
00009 #define Cluster_H
00010
00011 #include "Misc.h"
00012 #include "Pad.h"
00013
00014 #include "FitOutput.h"
00015
00016 #include "TF1.h"
00017 #include "TMatrixD.h"
00018 #include "TVirtualFitter.h"
00019
00021 class Cluster {
00022 public:
00029   Cluster(const int &EventNber, const int &EntryNber, const int &ModuleNber);
00030
00032   virtual ~Cluster();
00033
00034 //-----IDs-----
00035   int Get_EventNber() const; // Get Event nber
00036   int Get_EntryNber() const; // Get Entry nber
00037   int Get_ModuleNber() const; // Get Module nber
00038
00039 //-----Prints-----
00040   void WriteOut() const; // Big dump
00041
00042 //-----Cluster Validity-----
00043   int IsValid() const; // Get validation status
00044   void Validate(); // Validate
00045   void Invalidate(); // Invalidate
00046
00047 //-----Clusters Data-----
00048   double Get_Acluster() const; // Sum of Pads amplitude
00049   double Get_XWeight() const; // Averaged cluster position
00050   double Get_YWeight() const; // Averaged cluster position
00051
00052 //-----Pads-----
00053   void Add_Pad(Pad *pPad); // Add Pad
00054   void DoClosure(); // Closure: Order pads, compute Leading pad stuff
00055
00056   int Get_NberOfPads() const; // Size of Pad set
00057   Pad *Get_Pad(const int &Index1D) const; // Get Pad
00058
00059 // Leading pad
00060   Pad *Get_LeadingPad() const; // Get Leading pad
00061   double Get_YLeading() const; // Get Leading pad Y position (m)
00062   double Get_TMaxLeading() const; // Get Leading pad Time at max (Time bin)
00063   double Get_AMaxLeading() const; // Get Leading pad Max amplitude (ADC)
00064
00065   Pad *Get_NextLeadingPad() const; // Get Next Leading pad, the hottest of the
00066                                // nearest pads of the leading pad
00067   Pad *Get_NextNextLeadingPad() const; // Get NextNext Leading pad, the coldest
00068                                // of the nearest pads of the leading pad
00069
00070 //-----Data Members-----
00071 private:
00072   short int m_NberOfPads;
00073   std::vector<Pad *> V_Pad;
00074   Pad *Pad_Leading;
00075   Pad *Pad_NextLeading;

```

```

00076     Pad *Pad_NextNextLeading;
00077
00078     double m_Acluster;
00079     double m_XWeight;
00080     double m_YWeight;
00081
00082     int m_EventNber;
00083     int m_EntryNber;
00084     short int m_ModuleNber;
00085
00086 //-----Cluster Fit Stuff-----//
00087 private:
00088     double m_XTrack;
00089     double m_YTrack;
00090     double m_eXTrack;
00091     double m_eYTrack;
00092     bool m_StatusFit; // True: fit failed; False: Fit OK
00093
00094     double m_Chi2Min;
00095
00096     double m_XTrack_BeforeMinimisation;
00097     double m_YTrack_BeforeMinimisation;
00098
00099     short int m_isValid = 1;
00100
00101 public:
00102     void Set_XTrack(const double &XTrack);
00103     void Set_YTrack(const double &YTrack);
00104
00105     void Set_eXTrack(const double &eXTrack);
00106     void Set_eYTrack(const double &eYTrack);
00107
00108     int StatusFit(); // 0: OK; 1: Failed
00109
00110     double Get_XTrack() const;
00111     double Get_eXTrack() const;
00112     double Get_YTrack() const;
00113     double Get_eYTrack() const;
00114
00115     double Get_Chi2Min() const;
00116
00117     double Get_XTrack_BeforeMinimisation() const;
00118     double Get_YTrack_BeforeMinimisation() const;
00119
00120 //-----Horizontal-----//
00121 public:
00122     int StatusFit_Horizontal(); // 0: OK; 1: Failed
00123     void SetEval_Horizontal(TF1 *pTF1);
00124     double Eval_Horizontal(const double &xin);
00125     int SetParameter_Horizontal(TVirtualFitter *pTVirtualFitter);
00126     void SetResults_Horizontal(TVirtualFitter *pTVirtualFitter);
00127     void SetResults_FailedFit_Horizontal(const int &Verbose);
00128     double Chi2_Horizontal(double par[]);
00129
00130     int FitRes_Horizontal_Get_NberOfTermsInChi2() const;
00131     Pad *FitRes_Horizontal_Get_Pad(const int &IndexID) const;
00132     double FitRes_Horizontal_Get_Residual(const int &IndexID) const;
00133     double FitRes_Horizontal_Get_Pull(const int &IndexID) const;
00134
00135 private:
00136     void Beg_Horizontal();
00137     void End_Horizontal();
00138     void SetParameters_Internal_Horizontal(double par[]);
00139
00140     TF1 *p_TF1_Horizontal;
00141     bool m_StatusFit_Horizontal; // True: fit failed; False: Fit OK
00142     short int m_NberOf_V_FitRes_Horizontal_Pad;
00143     std::vector<Pad *> V_FitRes_Horizontal_Pad;
00144     std::vector<double> V_FitRes_Horizontal_Residual;
00145     std::vector<double> V_FitRes_Horizontal_Pull;
00146
00147     FitOutput *p_FitOutput_Horizontal;
00148
00149 //-----Diagonal-----//
00150 public:
00151     int StatusFit_Diagonal(); // 0: OK; 1: Failed
00152     void SetEval_Diagonal(TF1 *pTF1);
00153     double Eval_Diagonal(const double &xin);
00154     int SetParameter_Diagonal(TVirtualFitter *pTVirtualFitter);
00155     void SetResults_Diagonal(TVirtualFitter *pTVirtualFitter);
00156     void SetResults_FailedFit_Diagonal(const int &Verbose);
00157     double Chi2_Diagonal(double par[]);
00158 ;
00159     double m_AngleRot;
00160
00161     int FitRes_Diagonal_Get_NberOfTermsInChi2() const;
00162     Pad *FitRes_Diagonal_Get_Pad(const int &IndexID) const;

```

```

00163     double FitRes_Diagonal_Get_Residual(const int &Index1D) const;
00164     double FitRes_Diagonal_Get_Pull(const int &Index1D) const;
00165
00166 private:
00167     double m_VTrack;
00168
00169     void Beg_Diagonal();
00170     void End_Diagonal();
00171     void SetParameters_Internal_Diagonal(double par[]);
00172
00173     TF1 *p_TF1_Diagonal;
00174     bool m_StatusFit_Diagonal; // True: fit failed; False: Fit OK
00175     short int m_NberOf_V_FitRes_Diagonal_Pad;
00176     std::vector<Pad *> V_FitRes_Diagonal_Pad;
00177     std::vector<double> V_FitRes_Diagonal_Residual;
00178     std::vector<double> V_FitRes_Diagonal_Pull;
00179
00180     FitOutput *p_FitOutput_Diagonal;
00181 };
00182
00183 #endif

```

## 9.19 ModelEvent/inc/Event.h File Reference

[Event](#) container holding modules, clusters and tracks.

```

#include "Misc.h"
#include "Cluster.h"
#include "Module.h"
#include "Pad.h"
#include "Track.h"
#include "Model_ChargeI.h"
#include "Model_Electronics.h"
#include "Model_ReadOutGeometry.h"

```

### Classes

- class [Event](#)

#### 9.19.1 Detailed Description

[Event](#) container holding modules, clusters and tracks.

The [Event](#) class aggregates detector modules, clusters and track candidates for a single event. It also holds pointers to model objects (geometry, electronics, charge) used by simulation and reconstruction.

## 9.20 Event.h

[Go to the documentation of this file.](#)

```

00001 #
00009 #ifndef Event_H
00010 #define Event_H
00011
00012 #include "Misc.h"
00013
00014 #include "Cluster.h"
00015 #include "Module.h"
00016 #include "Pad.h"
00017 #include "Track.h"

```

```

00018 #include "Model_ChargeI.h"
00019 #include "Model_Electronics.h"
00020 #include "Model_ReadOutGeometry.h"
00021
00022
00023 class Event {
00024 public:
00025     Event(const int &EventNber, const int &EntryNber,
00026           Model_ReadOutGeometry *pModel_ReadOutGeometry,
00027           Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI);
00028     virtual ~Event();
00029
00030 //-----IDs-----
00031     int Get_EventNber() const; // Get Event nber
00032     int Get_EntryNber() const; // Get Entry nber
00033
00034 //-----Print-----
00035     void SmallDump() const; // Small dump
00036     void WriteOut() const; // Big dump
00037
00038 //-----Models-----
00039     const Model_ReadOutGeometry *Get_Model_ReadOutGeometry();
00040     const Model_Electronics *Get_Model_Electronics();
00041     const Model_ChargeI *Get_Model_ChargeI();
00042
00043 //-----Event Validity-----
00044     int IsValid() const; // Get validation status of the Event
00045     void Validate(); // Validate the Event
00046     void Invalidate(); // Invalidate the Event
00047
00048 //-----Module Addition-----
00049     void Clear_Modules(); // Clear existing Module set
00050     void Add_Module(Module *pModule); // Add a Module
00051
00052 //-----Module Access-----
00053     Module *Get_ThisModule(
00054         const int &ModuleNber); // Get Module by the Module Nber if exists
00055     const Module *Get_ThisModuleConst(const int &ModuleNber) const;
00056
00057     int Get_NberOfModule() const; // Size of Module set
00058     Module *Get_Module_InArray(const int &Index1D); // Get Module
00059
00060 //-----Module Validity-----
00061     int Validity_ForThisModule(
00062         const int &ModuleNber) const; // Get validation status of the Module
00063     void Validate_ThisModule(const int &ModuleNber); // Validate the Module
00064     void Invalidate_ThisModule(const int &ModuleNber); // Invalidate the Module
00065
00066 //-----Clusters Access-----
00067     std::vector<Cluster *> GiveMe_Clusters_ForThisModule(const int &ModuleNber);
00068
00069     double GiveMe_CutSlopeXY_ForThisModule(const int &ModuleNber);
00070
00071     double GiveMe_CutSlopeXZ_ForThisModule(const int &ModuleNber);
00072     double GiveMe_CutInterCeptXZ_ForThisModule(const int &ModuleNber);
00073
00074     std::vector<double> GiveMe_CutSlopeXYZ_ForThisModule(const int &ModuleNber);
00075
00076     void Replace_Clusters_ForThisModule(
00077         std::vector<Cluster *> &VCluster,
00078         const int &ModuleNber); // Replace existing clusters set
00079
00080     Cluster *Get_Cluster_Copy(Cluster *pCluster); // Get copy of the input cluster
00081
00082 //-----Track Access-----
00083     Track *GiveMe_AnUnfittedTrack_ForThisModule(
00084         const int &ModuleNber); // Ownership is passed to the user
00085     void Set_Track_ForThisModule(
00086         Track *pTrack,
00087         const int &ModuleNber); // Ownership is passed to the Module
00088     const Track *GiveMe_Track_ForThisModule(const int &ModuleNber);
00089
00090 //-----Data Members-----
00091 private:
00092     short int m_IsValid = 1;
00093
00094     int m_EventNber;
00095     int m_EntryNber;
00096
00097     short int m_NberOfModules;
00098     std::vector<Module *> V_Module;
00099
00100    void Ini_Models(Model_ReadOutGeometry *pModel_ReadOutGeometry,
00101                    Model_Electronics *pModel_Electronics,
00102                    Model_ChargeI *pModel_ChargeI);
00103
00104    Model_ReadOutGeometry *p_Model_ReadOutGeometry;

```

```

00107 Model_Electronics *p_Model_Electronics;
00108 Model_ChargeI *p_Model_ChargeI;
00109 };
00110
00111 #endif

```

## 9.21 ModelEvent/inc/FitOutput.h File Reference

Containers to hold fit results from cluster/track fitters.

```

#include "Misc.h"
#include "TCanvas.h"
#include "TMatrixD.h"
#include "TVirtualFitter.h"

```

### Classes

- class [FitOutput](#)

#### 9.21.1 Detailed Description

Containers to hold fit results from cluster/track fitters.

Declares [FitOutput](#) which stores parameter values, uncertainties and auxiliary quality flags produced by fitting routines.

## 9.22 FitOutput.h

[Go to the documentation of this file.](#)

```

00001 #
00008 #ifndef FitOutput_H
00009 #define FitOutput_H
00010
00011 #include "Misc.h"
00012
00013 #include "TCanvas.h"
00014 #include "TMatrixD.h"
00015 #include "TVirtualFitter.h"
00016
00018 class FitOutput {
00019 public:
00020     FitOutput();
00021     virtual ~FitOutput();
00022
00023     FitOutput(const FitOutput &ToBeCopied);
00024     FitOutput &operator=(const FitOutput &ToBeCopied);
00025
00026 public:
00028     void Reset();
00029     void Set(const int &NberOfModelParameters);
00031
00032     void SetResults(TVirtualFitter *pTVirtualFitter);
00033     void SetResults(const std::vector<std::string> &V_PARname,
00034                     const std::vector<double> &V_PAR,
00035                     const std::vector<double> &V_ePAR);
00036
00037     void PrintFitOutput();
00038     void PrintFitOutputInCanvas(double Xstart, double Ystart, double Step);
00039
00040 public:

```

```

00042     std::string m_NameOfTheModel;
00044     int m_NberOfModelParameters;
00046     int m_NberOfDataPoints;
00048
00049     double *p_par;
00050     std::string *p_parName;
00051     double *p_eparplus;
00052     double *p_eparminus;
00053     double *p_eparparab;
00054     double *p_CovMatrix;
00055     double m_MinnLL;
00056
00057     std::string m_Description;
00058 };
00059 #endif

```

## 9.23 ModelEvent/inc/Module.h File Reference

[Module](#) container representing a detector module of pads and clusters.

```

#include "Misc.h"
#include "Cluster.h"
#include "Pad.h"
#include "Track.h"
#include "ROBoard.h"
#include "Model_ChargeI.h"
#include "Model_Electronics.h"
#include "Model_ReadOutGeometry.h"

```

### Classes

- class [Module](#)

#### 9.23.1 Detailed Description

[Module](#) container representing a detector module of pads and clusters.

[Module](#) extends [ROBoard](#) and provides helpers to access and manipulate per-module collections used by the reconstruction pipeline.

## 9.24 Module.h

[Go to the documentation of this file.](#)

```

00001 #
00008 #ifndef Module_H
00009 #define Module_H
00010
00011 #include "Misc.h"
00012
00013 #include "Cluster.h"
00014 #include "Pad.h"
00015 #include "Track.h"
00016
00017 #include "ROBoard.h"
00018

```

```

00019 #include "Model_ChargeI.h"
00020 #include "Model_Electronics.h"
00021 #include "Model_ReadOutGeometry.h"
00022
00023 class Module : public ROBoard {
00024     public:
00025         Module(const int &EventNber, const int &EntryNber, const int &ModuleNber,
00026                 Model_ReadOutGeometry *pModel_ReadOutGeometry,
00027                 Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI);
00028         virtual ~Module();
00029
00030     //-----IDs-----
00031     int Get_EventNber() const; // Get Event nber
00032     int Get_EntryNber() const; // Get Entry nber
00033     int Get_ModuleNber() const; // Get Module nber
00034
00035     //-----Prints-----
00036     void SmallDump() const; // Small dump
00037     void WriteOut() const; // Big dump
00038
00039     //-----Module Validity-----
00040     int IsValid() const; // Get validation status of the Module
00041     void Validate(); // Validate the Module
00042     void Invalidate(); // Invalidate the Module
00043
00044     //-----Pad addition-----
00045     virtual void Add_Pad(Pad *pPad); // Add a pad
00046
00047     //-----Clusters Addition-----
00048     void Clear_Clusters(); // Clear existing clusters set
00049     void Add_Cluster(Cluster *pCluster); // Add a cluster
00050
00051     //-----Clusters Access-----
00052     std::vector<Cluster *> GiveMe_Clusters_ForThisModule();
00053     int Get_NberOfCluster() const; // Size of Cluster set
00054     Cluster *Get_Cluster(const int &IndexID); // Get Cluster
00055
00056     void ReplaceClusters(
00057         std::vector<Cluster *> &V_Cluster); // Replace existing clusters set
00058
00059     Cluster *Get_Cluster_Copy(Cluster *pCluster); // Get copy of the input cluster
00060
00061     //-----Track-----
00062     const Track *GiveMe_ModuleTrack();
00063     void Set_ModuleTrack(Track *pTrack); // Ownership is passed to the Module
00064     Track *GiveMe_AnUnfittedTrack(); // Ownership is passed to the user
00065
00066     //-----Data Members-----
00067
00068 private:
00069     short int m_IsValid = 1;
00070
00071     int m_EventNber;
00072     int m_EntryNber;
00073     short int m_ModuleNber;
00074
00075     short int m_NberOfClusters;
00076     std::vector<Cluster *> V_Cluster;
00077
00078     Track *p_TrackOfTheModule;
00079
00080 };
00081
00082 #endif

```

## 9.25 ModelEvent/inc/Pad.h File Reference

[Pad](#) class representing a single readout pad and its waveform data.

```
#include "Misc.h"
#include "Model_ChargeI.h"
#include "Model_Electronics.h"
#include "Model_ReadOutGeometry.h"
```

### Classes

- class [Pad](#)

### 9.25.1 Detailed Description

[Pad](#) class representing a single readout pad and its waveform data.

Contains identifiers, waveform buffers and helpers used by clustering and fitting algorithms. Implementations live in [Pad.cxx](#).

## 9.26 Pad.h

[Go to the documentation of this file.](#)

```

00001 #
00008 #ifndef Pad_H
00009 #define Pad_H
00010
00011 #include "Misc.h"
00012
00013 #include "Model_ChargeI.h"
00014 #include "Model_Electronics.h"
00015 #include "Model_ReadOutGeometry.h"
00016
00018 class Pad {
00019 public:
00023     Pad(Model_ReadOutGeometry *pModel_ReadOutGeometry,
00024         Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI,
00025         std::string PadName, const int &EventNber, const int &EntryNber,
00026         const int &ModuleNber, const int &iX, const int &iY, const double &XL,
00027         const double &XH, const double &YL, const double &YH);
00028
00032     Pad(Model_ReadOutGeometry *pModel_ReadOutGeometry,
00033         Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI,
00034         std::string PadName, const int &EventNber, const int &EntryNber,
00035         const int &ModuleNber, const int &iX, const int &iY);
00036
00038     virtual ~Pad();
00039
00040 //-----IDs-----
00041     int Get_EventNber() const; // Get Event nber
00042     int Get_EntryNber() const; // Get Entry nber
00043     int Get_ModuleNber() const; // Get Module nber
00044     int Get_iX() const; // Get X index
00045     int Get_iY() const; // Get Y index
00046
00047 //-----Print-----
00048     void WriteOut() const; // Big dump
00049
00050 //-----Pad Validity-----
00051     int IsValid() const; // Get validation status
00052     void Validate(); // Validate
00053     void Invalidate(); // Invalidate
00054
00055 //-----Models-----
00056     const Model_ReadOutGeometry *Get_Model_ReadOutGeometry() const;
00057     const Model_Electronics *Get_Model_Electronics() const;
00058     const Model_ChargeI *Get_Model_ChargeI() const;
00059
00060 //-----Pad Data-----
00061     std::string Get_PadName() const; // Get Pad name
00062
00063     double Get_XPad() const; // Get X position of Pad center (m)
00064     double Get_YPad() const; // Get Y position of Pad center (m)
00065
00066     double Get_XL() const; // Get X position of Left Pad edge (m)
00067     double Get_XH() const; // Get X position of Right Pad edge (m)
00068     double Get_YL() const; // Get Y position of Bottom Pad edge (m)
00069     double Get_YH() const; // Get Y position of top Pad edge (m)
00070
00071     double Get_LX() const; // Get X Pad size (m)
00072     double Get LY() const; // Get Y Pad size (m)
00073
00074     double Get_AMax() const; // Get max amplitude (ADC)
00075     double Get_TMax() const; // Get Time at max (Time bin)
00076
00077     void Set_AMax(const double &AMax); // Set max amplitude (ADC)
00078     void Set_TMax(const double &TMax); // Set Time at max (Time bin)
00079
00080     void
00081     Set_Data_2Use(const int &iOpt); // Data: switch for data type to use

```

```

00082                                     // iOpt = 0; Amax and Tmax from input ntuple
00083                                     //          = 1; Amax and Tmax from 1st max
00084                                     //of WF      = 2; Amax and Tmax from 1st clean max
00085                                     //of WF      = 3; Amax and Tmax from
00086                                     // fit of WF peak (Default)
00087
00088 //-----WFS-----//
00089 // Waveform stuff
00090 void Clear_ADC(); // Reset ADC set
00091 void Set_ADC(const int &iTimeb,
00092               const int &ADC); // Set ADC set
00093 void WF_DoClosure(); // Closure: compute max amplitude and time
00094 std::vector<float> Get_vADC() const; // Get ADC as vector
00095
00096 // A and T max
00097 // A and T from last call to Set functions
00098 double Get_AMax_FromSet() const; // Get max amplitude (ADC)
00099 double Get_TMax_FromSet() const; // Get Time at max (Time bin)
00100
00101 // 1st maximum in WF
00102 double Get_AMax_WF() const; // Get max amplitude (ADC)
00103 double Get_TMax_WF() const; // Get Time at max (Time bin)
00104
00105 // Improved 1st maximum in WF
00106 double Get_AMax_WF_01() const; // Get max amplitude (ADC)
00107 double Get_TMax_WF_01() const; // Get Time at max (Time bin)
00108
00109 // Fit of peak
00110 double Get_AMax_FIT() const; // Get max amplitude (ADC)
00111 double Get_TMax_FIT() const; // Get Time at max (Time bin)
00112 int Get_FIT_Status() const; // 1: fit failed; 0: Fit OK
00113 double Get_FIT_AOP() const; //
00114 double Get_FIT_AOM() const; //
00115 double Get_FIT_X0() const; //
00116 double Get_FIT_Y0() const; //
00117 double Get_FIT_Xmin() const; //
00118 double Get_FIT_Xmax() const; //
00119
00120 // A and T from thruth
00121 double Get_AMax_True() const; // Get max amplitude (ADC)
00122 double Get_TMax_True() const; // Get Time at max (Time bin)
00123 void Set_AMax_True(const double &AMax); // Set max amplitude (ADC)
00124 void Set_TMax_True(const double &TMax); // Set Time at max (Time bin)
00125
00126
00127 private:
00128 void Get_A_T_Max_WF(double &Amax, double &Tmax);
00129 // Get Amax and Tmax from WF in the time window [iTimeBin_min;iTimeBin_max]
00130 void Get_A_T_Max_WF_01(double &Amax, double &Tmax);
00131 // Get Amax and Tmax from WF in the time window [iTimeBin_min;iTimeBin_max]
00132 void Get_A_T_Max_FIT(double &Amax, double &Tmax);
00133 // Get Amax and Tmax from WF Fit in the time window
00134 // [iTimeBin_min;iTimeBin_max]
00135
00136 //-----Signal Simulation-----//
00137 public:
00138 // Set parameters of the simulation
00139 void SetSignalModel(const double &Time0, const double &XTrue,
00140                      const double &YTrue);
00141
00142 double Get_Time0() const;
00143 double Get_XTrue() const;
00144 double Get_YTrue() const;
00145
00146 double Get_Qpad(const double &Time); // charge
00147 double Get_APad(const double &Time); // amplitude
00148
00149 //-----Data Members-----//
00150 private:
00151 short int m_IsValid = 1;
00152
00153 void SetEdges(const double &XL, const double &XH, const double &YL,
00154                 const double &YH);
00155
00156 void Ini_Models(Model_ReadOutGeometry *pModel_ReadOutGeometry,
00157                   Model_Electronics *pModel_Electronics,
00158                   Model_ChargeI *pModel_ChargeI);
00159
00160 private:
00161 std::string m_PadName;
00162
00163 int m_EventNber;
00164 int m_EntryNber;
00165 short int m_ModuleNber;
00166 short int m_iX;
00167 short int m_iY;
00168

```

```

00169 //      double m_XPad;
00170 //      double m_YPad;
00171 //      double m_LX;
00172 //      double m LY;
00173
00174     double m_XL;
00175     double m_XH;
00176     double m_YL;
00177     double m_YH;
00178
00179     double m_Time0;
00180     double m_XTrue;
00181     double m_YTrue;
00182
00183     double m_TimeConvoMin;
00184     double m_TimeConvoMax;
00185     int m_NberOfTimeConvoPoints;
00186     double m_TimeConvoStep;
00187
00188     double m_AMax;
00189     double m_TMax;
00190
00191     Model_ReadOutGeometry *p_Model_ReadOutGeometry;
00192     Model_Electronics *p_Model_Electronics;
00193     Model_ChargeI *p_Model_ChargeI;
00194
00195     std::vector<short int> v_ADC;
00196     std::vector<short int> v_iTimeb;
00197
00198     double m_AMax_FromSet;
00199     double m_TMax_FromSet;
00200
00201     double m_AMax_WF;
00202     double m_TMax_WF;
00203
00204     double m_AMax_WF_01;
00205     double m_TMax_WF_01;
00206
00207     double m_AMax_FIT;
00208     double m_TMax_FIT;
00209     bool m_FIT_Status; // True: fit failed; False: Fit OK
00210     double m_FIT_A0P;
00211     double m_FIT_A0M;
00212     double m_FIT_X0;
00213     double m_FIT_Y0;
00214     double m_FIT_Xmin;
00215     double m_FIT_Xmax;
00216
00217     double m_AMax_True;
00218     double m_TMax_True;
00219 };
00220
00221 #endif

```

## 9.27 ModelEvent/inc/ROBoard.h File Reference

Read-out board ([ROBoard](#)) abstraction and mapping helpers.

```

#include "Misc.h"
#include "Pad.h"
#include "Model_ChargeI.h"
#include "Model_Electronics.h"
#include "Model_ReadOutGeometry.h"

```

### Classes

- class [ROBoard](#)

*Abstraction for a read-out board grouping pads and providing mapping.*

## Macros

- #define ROBoard\_H

### 9.27.1 Detailed Description

Read-out board ([ROBoard](#)) abstraction and mapping helpers.

Models hardware-level groupings of pads and supports mapping between electronics channels and pad indices.

### 9.27.2 Macro Definition Documentation

#### 9.27.2.1 ROBoard\_H

```
#define ROBoard_H
```

## 9.28 ROBoard.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #ifndef ROBOARD_H
00009 #define ROBOARD_H
00010 #ifndef ROBoard_H
00011 #define ROBoard_H
00012
00013 #include "Misc.h"
00014
00015 #include "Pad.h"
00016
00017 #include "Model_ChargeI.h"
00018 #include "Model_Electronics.h"
00019 #include "Model_ReadOutGeometry.h"
00020
00022
00026 class ROBoard {
00027 public:
00029     ROBoard(Model_ReadOutGeometry *pModel_ReadOutGeometry,
00030             Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI);
00031     virtual ~ROBoard();
00032
00033     // Models
00034     const Model_ReadOutGeometry *Get_Model_ReadOutGeometry();
00035     const Model_Electronics *Get_Model_Electronics();
00036     const Model_ChargeI *Get_Model_ChargeI();
00037
00038     // Pads addition
00039     virtual void Add_Pad(Pad *pPad); // Add a pad
00040
00041     // Pads Access
00042     int Get_NberOfPads() const; // Size of Pad set
00043     Pad *Get_Pad(const int &Index1D); // Get Pad
00044
00045     int Get_IsThisPadExisting(
00046         const int &iX,
00047         const int &iY) const; // return Pad (iX,iY) existence state
00048     Pad *Get_Pad(const int &iX, const int &iY); // return Pad (iX,iY) if it exists
00049
00050     Pad *Get_Pad(Pad *pPad); // return Pad (iX,iY) if it exists
00051
00052 protected:
00053     int GetLinearIndex(const int &iX, const int &iY) const;
00054
00055     void Ini_Models(Model_ReadOutGeometry *pModel_ReadOutGeometry,
00056                      Model_Electronics *pModel_Electronics,
00057                      Model_ChargeI *pModel_ChargeI);
00058
00059 protected:
```

```

00060     short int m_Nx;
00061     short int m_Ny;
00062
00063     std::vector<int> V_ExisFlag2D;
00064
00065     std::vector<Pad *> V_Pad;
00066
00067     Model_ReadOutGeometry *p_Model_ReadOutGeometry;
00068     Model_Electronics *p_Model_Electronics;
00069     Model_ChargeI *p_Model_ChargeI;
00070 };
00071
00072 #endif

```

## 9.29 ModelEvent/inc/Sample.h File Reference

[Sample](#) container representing a dataset (collection of events).

```

#include "Misc.h"
#include "Model_ChargeI.h"
#include "Model_Electronics.h"
#include "Model_ReadOutGeometry.h"
#include "Cluster.h"
#include "Event.h"
#include "Pad.h"
#include "FitOutput.h"
#include "SetOfTracks.h"
#include "Track.h"
#include "TF1.h"
#include "TMatrixD.h"
#include "TVirtualFitter.h"

```

### Classes

- class [Sample](#)

#### 9.29.1 Detailed Description

[Sample](#) container representing a dataset (collection of events).

The [Sample](#) class stores a set of [Event](#) objects and provides convenience methods for iteration, model association and PRF handling.

## 9.30 Sample.h

[Go to the documentation of this file.](#)

```

00001 #
00008 #ifndef Sample_H
00009 #define Sample_H
00010
00011 #include "Misc.h"
00012
00013 #include "Model_ChargeI.h"
00014 #include "Model_Electronics.h"
00015 #include "Model_ReadOutGeometry.h"
00016

```

```

00017 #include "Cluster.h"
00018 #include "Event.h"
00019 #include "Pad.h"
00020
00021 #include "FitOutput.h"
00022
00023 #include "SetOfTracks.h"
00024 #include "Track.h"
00025
00026 #include "TF1.h"
00027 #include "TMatrixD.h"
00028 #include "TVirtualFitter.h"
00029
00031 class Sample {
00032 public:
00033     Sample();
00034     Sample(Model_ReadOutGeometry *pModel_ReadOutGeometry,
00035            Model_Electronics *pModel_Electronics, Model_ChargeI *pModel_ChargeI);
00036     virtual ~Sample();
00038
00039 //-----Prints-----
00040 void SmallDump() const; // Small dump
00041 void WriteOut() const; // Big dump
00042
00043 //-----Models-----
00044 // Set Models
00045 void Set_Model_ReadOutGeometry(Model_ReadOutGeometry *pModel_ReadOutGeometry);
00046 void Set_Model_Electronics(Model_Electronics *pModel_Electronics);
00047 void Set_Model_Charge(Model_ChargeI *pModel_ChargeI);
00048
00049 // Get Models
00050 Model_ReadOutGeometry *Get_Model_ReadOutGeometry();
00051 Model_Electronics *Get_Model_Electronics();
00052 Model_ChargeI *Get_Model_ChargeI();
00053
00054 //-----Event Addition-----
00055 void Add_Event(Event *pEvent); // Add Event
00056
00057 //-----Event Access-----
00058 int Get_NberOfEvents() const; // Size of Event set
00059 Event *Get_Event(int Index1D); // Get Event
00060
00061 //-----Track Access-----
00062 SetOfTracks *Get_SetOfTracks_ForThisModule(
00063     const int &ModuleNber); // Ownership is passed to user
00064
00065 //-----PRF from input file-----
00066 void SetFilePRF(const double &Norm, const double &a2, const double &a4,
00067                  const double &b2, const double &b4);
00068
00069 void GetFilePRF(int &FilePRF_Exist, double &Norm, double &a2, double &a4,
00070                  double &b2, double &b4);
00071
00072 //-----Data Members-----
00073 private:
00074     // Copy constructor
00075     Sample(const Sample &obj) {}
00076
00077     // Copy assignment operator
00078     Sample &operator=(const Sample &tmp_obj) { return *this; }
00079
00080 private:
00081     Model_ReadOutGeometry *p_Model_ReadOutGeometry;
00082     Model_Electronics *p_Model_Electronics;
00083     Model_ChargeI *p_Model_ChargeI;
00084
00085     int m_NberOfEvents;
00086     std::vector<Event *> V_Event;
00087
00088     short int m_FilePRF_Exist;
00089     double m_FilePRF_Norm;
00090     double m_FilePRF_a2;
00091     double m_FilePRF_a4;
00092     double m_FilePRF_b2;
00093     double m_FilePRF_b4;
00094
00095 //-----RC and DD Fit Stuff-----
00096 private:
00097     double m_RC;
00098     double m_eRC;
00099     double m_DD;
00100    double m_eDD;
00101    bool m_StatusFit; // True: fit failed; False: Fit OK
00102
00103    double m_Chi2Min;
00104    double m_RC_BeforeMinimisation;

```

```

00106     double m_DD_BeforeMinimisation;
00107
00108 public:
00109     void Set_RC(const double &RC);
00110
00111     void Set_DD(const double &RC);
00112
00113     int StatusFit(); // 0: OK; 1: Failed
00114
00115     double Get_RC() const;
00116     double Get_eRC() const;
00117
00118     double Get_DD() const;
00119     double Get_eDD() const;
00120
00121     double Get_Chi2Min() const;
00122
00123     double Get_RC_BeforeMinimisation() const;
00124     double Get_DD_BeforeMinimisation() const;
00125 };
00126
00127 #endif

```

## 9.31 ModelEvent/inc/SetOfTracks.h File Reference

Container for collections of [Track](#) objects.

```

#include "Misc.h"
#include "Track.h"
#include "TH1F.h"

```

### Classes

- class [SetOfTracks](#)

#### 9.31.1 Detailed Description

Container for collections of [Track](#) objects.

Provides storage and plotting helpers for sets of tracks used in reconstruction and analysis.

## 9.32 SetOfTracks.h

[Go to the documentation of this file.](#)

```

00001 #
00008 #ifndef SetOfTracks_H
00009 #define SetOfTracks_H
0010
0011 #include "Misc.h"
0012 #include "Track.h"
0013
0014 #include "TH1F.h"
0015
0017 class SetOfTracks {
0018 public:
0019     SetOfTracks();
0020     virtual ~SetOfTracks();
0021
0022     -----Track Addition-----//
0023     void Add_Track(const Track *pTrack);
0024
0025     -----Track Acess-----//

```

```

00026     int Get_NberOfTrack() const;
00027     const Track *Get_Track(int IndexID) const;
00028
00029 //-----Prints-----
00030 void DumpRec() const;
00031
00032 //-----Plots-----
00033 int Get_MinimalNberOfEntries();
00034 void Set_MinimalNberOfEntries(const int &MinimalNberOfEntries);
00035
00036 TH1F *GiveMe_pTH1F_YTrackInTracks(const std::string &TAG, const int &iYBeam);
00037 TH1F *GiveMe_pTH1F_YTrackInTracks(const std::string &TAG, const int &Nbins,
00038                                         const double &Ymin, const double &Ymax);
00039 TH1F *GiveMe_pTH1F_YTrackInTracks(const std::string &TAG, const int &iYBeam,
00040                                         const int &iCol);
00041 TH1F *GiveMe_pTH1F_YTrackInTracks(const std::string &TAG, const int &Nbins,
00042                                         const double &Ymin, const double &Ymax,
00043                                         const int &iCol);
00044
00045 //
00046 TH1F *GiveMe_pTH1F_YFitCol(const std::string &TAG, const int &iYBeam);
00047 TH1F *GiveMe_pTH1F_YFitCol(const std::string &TAG, const int &Nbins,
00048                                         const double &Ymin, const double &Ymax);
00049 TH1F *GiveMe_pTH1F_YFitCol(const std::string &TAG, const int &iYBeam,
00050                                         const int &iCol);
00051 TH1F *GiveMe_pTH1F_YFitCol(const std::string &TAG, const int &Nbins,
00052                                         const double &Ymin, const double &Ymax,
00053                                         const int &iCol);
00054 //
00055 TH1F *GiveMe_pTH1F_YTrackYPadInTracks(const std::string &TAG,
00056                                         const int &iCol);
00057 TH1F *GiveMe_pTH1F_YTrackYPadInTracks(const std::string &TAG, const int &iCol,
00058                                         const int &Nbins, const double &Xmin,
00059                                         const double &Xmax);
00060 TH1F *GiveMe_pTH1F_YTrackYPadLeadingInTracks(const std::string &TAG,
00061                                         const int &iCol);
00062 TH1F *GiveMe_pTH1F_YTrackYPadLeadingInTracks(const std::string &TAG,
00063                                         const int &iCol,
00064                                         const int &Nbins,
00065                                         const double &Xmin,
00066                                         const double &Xmax);
00067 //
00068 TH1F *GiveMe_pTH1F_TrackDeltaT(const std::string &TAG, const int &iCol);
00069 //
00070 TH1F *GiveMe_pTH1F_Ch2MinPerNODF(const std::string &TAGFinal,
00071                                         const double &Xmax);
00072 TH1F *GiveMe_pTH1F_Ch2Min(const std::string &TAGFinal, const double &Xmax);
00073 TH1F *GiveMe_pTH1F_Residual(const std::string &TAGFinal, double &Mean,
00074                                         double &eMean, double &Sigma, double &eSigma);
00075 TH1F *GiveMe_pTH1F_Residual(const std::string &TAGFinal, const int &iX,
00076                                         double &Mean, double &eMean, double &Sigma,
00077                                         double &eSigma);
00078
00079 TH1F *GiveMe_pTH1F_Residual(const std::string &TAGFinal, const int &iX,
00080                                         double YL, double YH, double YL_Rescaled,
00081                                         double YH_Rescaled, double &Mean, double &eMean,
00082                                         double &Sigma, double &eSigma);
00083
00084 TH1F *GiveMe_pTH1F_Pull(const std::string &TAGFinal);
00085 TH1F *GiveMe_pTH1F_Pull(const std::string &TAGFinal, double &Mean,
00086                                         double &eMean, double &Sigma, double &eSigma);
00087 TH1F *GiveMe_pTH1F_Pull(const std::string &TAGFinal, double &Mean,
00088                                         double &eMean, double &Sigma, double &eSigma,
00089                                         const double &XminH, const double &XmaxH);
00090 TH1F *GiveMe_pTH1F_Pull(const std::string &TAGFinal, const int &iX,
00091                                         double &Mean, double &eMean, double &Sigma,
00092                                         double &eSigma);
00093 TH1F *GiveMe_pTH1F_Pull(const std::string &TAGFinal, const int &iX,
00094                                         double &Mean, double &eMean, double &Sigma,
00095                                         double &eSigma, const double &XminH,
00096                                         const double &XmaxH);
00097
00098 private:
00100     int m_MinimalNberOfEntries;
00101
00102     int m_NberOfTracks;
00103     std::vector<const Track *> V_Track;
00104 };
00105
00106 #endif

```

## 9.33 ModelEvent/inc/TimeError.h File Reference

Time error model utilities.

```
#include "Misc.h"
#include "Pad.h"
```

### Functions

- double **TimeError** (const double &DeltaT\_in)
- double **TimeError** (**Pad** \*pPad, **Pad** \*pPad\_Leading, **Pad** \*pPad\_NextLeading, **Pad** \*pPad\_NextNextLeading)

### 9.33.1 Detailed Description

Time error model utilities.

Declares helpers to compute temporal uncertainties used during signal processing and fitting.

### 9.33.2 Function Documentation

#### 9.33.2.1 TimeError() [1/2]

```
double TimeError (
    const double & DeltaT_in)
```

#### 9.33.2.2 TimeError() [2/2]

```
double TimeError (
    Pad * pPad,
    Pad * pPad_Leading,
    Pad * pPad_NextLeading,
    Pad * pPad_NextNextLeading)
```

## 9.34 TimeError.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #ifndef TimeError_H
00009 #define TimeError_H
00010
00011 #include "Misc.h"
00012
00013 #include "Pad.h"
00014
00015 double TimeError(const double &DeltaT_in);
00016
00017 double TimeError(Pad *pPad, Pad *pPad_Leading, Pad *pPad_NextLeading,
00018             Pad *pPad_NextNextLeading);
00019
00020 #endif
```

## 9.35 ModelEvent/inc/Track.h File Reference

[Track](#) class representing a reconstructed particle trajectory.

```
#include "Cluster.h"
#include "Misc.h"
#include "FitOutput.h"
```

### Classes

- class [Track](#)

#### 9.35.1 Detailed Description

[Track](#) class representing a reconstructed particle trajectory.

The [Track](#) class stores fit parameters, associated clusters and provides utilities for fitting and evaluating the reconstructed trajectory.

## 9.36 Track.h

[Go to the documentation of this file.](#)

```
00001 #
00002 #ifndef Track_H
00003 #define Track_H
00010
00011 #include "Cluster.h"
00012 #include "Misc.h"
00013
00014 #include "FitOutput.h"
00015
00016 // Track equation : Y = Sum_(i=0,Npar-1) par_i * X^i
00017
00019 class Track {
00020 public:
00027   Track(const int &EventNber, const int &EntryNber, const int &ModuleNber);
00028
00030   virtual ~Track();
00031
00032   // IDs
00033   int Get_EventNber() const;
00034   int Get_EntryNber() const;
00035   int Get_ModuleNber() const;
00036
00037   //-----Track Data-----
00038   void SetNberOfParameters(const int &NberOfParam);
00039   int GetNberOfParameters() const;
00040
00041   std::string Get_ParameterName(const int &iPar) const;
00042   double Get_ParameterValue(const int &iPar) const;
00043   double Get_ParameterError(const int &iPar) const;
00044
00045   double Get_ParameterValue_BeforeMinimisation(const int &iPar) const;
00046
00047   //-----Cluster Addition-----
00048   void Add_Cluster(Cluster *pCluster);
00049
00050   //-----Cluster Access-----
00051   int Get_NberOfCluster() const;
00052   const Cluster *Get_Cluster(int Index1D) const;
00053   double Get_Residual(int Index1D) const;
00054   double Get_Pull(int Index1D) const;
00055
00056   void DoClosure();
00057   void DumpRec() const;
```

```

00058 //-----According to Fit Y position for given
00059 // X-----
00060 double Y_Position(const double &X) const;
00061
00062 //-----Data Members-----
00063
00064 private:
00065     int m_EventNber;
00066     int m_EntryNber;
00067     short int m_ModuleNber;
00068
00069     int m_NberOfParam;
00070
00071     std::vector<double> m_Parameter;
00072     std::vector<double> m_eParameter;
00073     std::vector<std::string> m_ParameterName;
00074
00075     std::vector<double> m_Parameter_BeforeMinimisation;
00076
00077 public:
00078
00079     int m_NberOfClusters;
00080     std::vector<Cluster *> V_Cluster;
00081     std::vector<double> V_Residual;
00082     std::vector<double> V_Pull;
00083
00084 //-----Track Fit Stuff-----
00085
00086 private:
00087     double m_Chi2Min;
00088     TMatrixD m_covmatrix;
00089
00090 public:
00091     double Get_Chi2Min() const;
00092     int SetParameter(TVirtualFitter *pTVirtualFitter);
00093     void SetResults(TVirtualFitter *pTVirtualFitter);
00094     double Chi2(double par[]);
00095     void SetParameters_Internal(double par[]);
00096     TMatrixD Get_CovMatrix() const;
00097
00098 private:
00099     FitOutput *p_FitOutput;
00100 };
00101
00102 #endif

```

## 9.37 ModelEvent/src/AmplitudeError.cxx File Reference

```
#include "AmplitudeError.h"
```

### Functions

- double AmplitudeError\_Leading ()
- double AmplitudeError\_NextLeading ()
- double AmplitudeError\_NextNextLeading ()
- double AmplitudeError (Pad \*pPad, Pad \*pPad\_Leading, Pad \*pPad\_NextLeading, Pad \*pPad\_NextNextLeading)

### 9.37.1 Function Documentation

#### 9.37.1.1 AmplitudeError()

```
double AmplitudeError (
    Pad * pPad,
    Pad * pPad_Leading,
    Pad * pPad_NextLeading,
    Pad * pPad_NextNextLeading)
```

### 9.37.1.2 AmplitudeError\_Leading()

```
double AmplitudeError_Leading ()
```

### 9.37.1.3 AmplitudeError\_NextLeading()

```
double AmplitudeError_NextLeading ()
```

### 9.37.1.4 AmplitudeError\_NextNextLeading()

```
double AmplitudeError_NextNextLeading ()
```

## 9.38 ModelEvent/src/Cluster.cxx File Reference

```
#include "Cluster.h"
#include "AmplitudeError.h"
#include "TimeError.h"
#include "Util.h"
```

## 9.39 ModelEvent/src/Event.cxx File Reference

```
#include "Event.h"
#include "TGraph.h"
```

## 9.40 ModelEvent/src/FitOutput.cxx File Reference

```
#include "FitOutput.h"
#include "TLatex.h"
#include "TMatrixD.h"
```

## 9.41 ModelEvent/src/Module.cxx File Reference

```
#include "Module.h"
```

## 9.42 ModelEvent/src/Pad.cxx File Reference

```
#include "Pad.h"
#include "ParabolaFunction.h"
#include "ParabolaFunctionNG.h"
#include "TF1.h"
#include "TH1F.h"
```

## 9.43 ModelEvent/src/ROBoardcxx File Reference

```
#include "ROBoard.h"
```

## 9.44 ModelEvent/src/Samplecxx File Reference

```
#include "Sample.h"
#include "AmplitudeError.h"
#include "TimeError.h"
```

## 9.45 ModelEvent/src/SetOfTrackscxx File Reference

```
#include "SetOfTracks.h"
#include "GaussFunction.h"
#include "TAxis.h"
#include "TCanvas.h"
#include "TF1.h"
#include "TGraph.h"
#include "TH1F.h"
#include "TLegend.h"
#include "TLine.h"
#include "TMultiGraph.h"
#include "TPaveStats.h"
#include "TProfile.h"
#include "TStyle.h"
```

## 9.46 ModelEvent/src/TimeErrorcxx File Reference

```
#include "TimeError.h"
```

## Functions

- double TimeError\_Leading ()
- double TimeError\_NextLeading ()
- double TimeError\_NextNextLeading ()
- double TimeError\_BASE (const double &DeltaT\_in)
- double TimeError (*Pad* \*pPad, *Pad* \*pPad\_Leading, *Pad* \*pPad\_NextLeading, *Pad* \*pPad\_NextNextLeading)
- double TimeError (const double &DeltaT\_in)

### 9.46.1 Function Documentation

#### 9.46.1.1 TimeError() [1/2]

```
double TimeError (
    const double & DeltaT_in)
```

#### 9.46.1.2 TimeError() [2/2]

```
double TimeError (
    Pad * pPad,
    Pad * pPad_Leading,
    Pad * pPad_NextLeading,
    Pad * pPad_NextNextLeading)
```

#### 9.46.1.3 TimeError\_BASE()

```
double TimeError_BASE (
    const double & DeltaT_in)
```

#### 9.46.1.4 TimeError\_Leading()

```
double TimeError_Leading ()
```

#### 9.46.1.5 TimeError\_NextLeading()

```
double TimeError_NextLeading ()
```

#### 9.46.1.6 TimeError\_NextNextLeading()

```
double TimeError_NextNextLeading ()
```

## 9.47 ModelEvent/src/Track.cxx File Reference

```
#include "Track.h"
#include "TAxix.h"
#include "TCanvas.h"
#include "TF1.h"
#include "TFile.h"
#include "TGraph.h"
#include "TGraphErrors.h"
#include "TH1F.h"
#include "TH2F.h"
#include "TLegend.h"
#include "TLine.h"
#include "TMutiGraph.h"
#include "TPaveStats.h"
#include "TProfile.h"
#include "TStyle.h"
```

## 9.48 ModelSignal/inc/Model\_Charge0D.h File Reference

0D charge model declaration (inherits from [Model\\_ChargeI](#)).

```
#include "Misc.h"
#include "Model_ChageI.h"
```

### Classes

- class [Model\\_Chage0D](#)

### 9.48.1 Detailed Description

0D charge model declaration (inherits from [Model\\_ChageI](#)).

Declares [Model\\_Chage0D](#) which estimates integrated charge produced by an ionising particle in a pad or collection region. Implementation is in [Model\\_Chage0D.cxx](#).

The 0D model returns integrated pad charges and can be used for quick analytic studies or as a baseline reference in tests. It provides configuration for ion contributions and normalization used by fitter utilities.

## 9.49 Model\_Charge0D.h

[Go to the documentation of this file.](#)

```

00001 #
00015 #ifndef Model_Charge0D_H
00016 #define Model_Charge0D_H
00017
00018 #include "Misc.h"
00019 #include "Model_ChargeI.h"
00020
00022 class Model_Charge0D : public Model_ChargeI {
00023 public:
00025     Model_Charge0D();
00026     virtual ~Model_Charge0D();
00027
00028     // Get charge spreading parameters
00029     double Get_Qprim() const;
00030
00031     // Set charge spreading parameters
00032     void Set_Qprim(const double &Qprim);
00033
00034     // Get Charge
00035     double Get_Qpad(const double &Time, const double &Time0, const double &XTrue,
00036                      const double &YTrue, const double &XL, const double &XH,
00037                      const double &YL, const double &YH) const;
00038
00039     // Big Dump
00040     void WriteOut() const;
00041
00042     // Ion Effect: Set charge spreading parameters
00043     void Set_Ion(const int &DoIons, const double &Tion, const double &RTrue,
00044                  const double &RFactor);
00045
00046 protected:
00047     void SetSecondaries();
00048
00049     double m_Qprim;
00050
00051     int m_DoIons;
00052     double m_Tion;
00053     double m_RTrue;
00054     double m_RFactor;
00055 };
00056
00057 #endif

```

## 9.50 ModelSignal/inc/Model\_Charge1D.h File Reference

1D charge model declaration (inherits from [Model\\_ChargeI](#)).

```
#include "Misc.h"
#include "Model_ChargeI.h"
```

### Classes

- class [Model\\_Charge1D](#)

### 9.50.1 Detailed Description

1D charge model declaration (inherits from [Model\\_ChargeI](#)).

Declares [Model\\_Charge1D](#) which models charge spreading along a single axis (e.g. time/drift). Concrete `Get_Qpad` and related helpers are provided in [Model\\_Charge1D.cxx](#).

The implementation computes pad-integrated charge for longitudinal spreading-dominated scenarios and exposes helpers to configure lambda and normalization used in simulations and fits.

## 9.51 Model\_Charge1D.h

[Go to the documentation of this file.](#)

```

00001 #
00014 #ifndef Model_Charge1D_H
00015 #define Model_Charge1D_H
00016
00017 #include "Misc.h"
00018 #include "Model_ChargeI.h"
00019
00021 class Model_Charge1D : public Model_ChargeI {
00022 public:
00024     Model_Charge1D();
00025     virtual ~Model_Charge1D();
00026
00027     // Get charge spreading parameters
00028     double Get_Lambda() const;
00029
00030     // Set charge spreading parameters
00031     void Set_Lambda(const double &Lambda);
00032
00033     // Get Charge
00034     double Get_Qpad(const double &Time, const double &Time0, const double &XTrue,
00035                     const double &YTrue, const double &XL, const double &XH,
00036                     const double &YL, const double &YH) const;
00037
00038     // Big Dump
00039     void WriteOut() const;
00040
00041 protected:
00042     void SetSecondaries();
00043
00044     double m_Lambda;
00045 };
00046
00047 #endif

```

## 9.52 ModelSignal/inc/Model\_ChargeI.h File Reference

Abstract base class for charge models.

```
#include "Misc.h"
```

### Classes

- class [Model\\_ChargeI](#)

### 9.52.1 Detailed Description

Abstract base class for charge models.

Defines the interface used by concrete charge model implementations (0D, 1D, etc.). Provides common parameter accessors and utility helpers.

Implementations (Model\_Charge0D/1D) provide concrete `Get_Qpad` and `SetSecondaries` logic; this base class implements shared helpers used by concrete models, such as width-from-drift calculations and normalization helpers.

## 9.53 Model\_ChargeI.h

[Go to the documentation of this file.](#)

```

00001 #
00014 #ifndef Model_ChargeI_H
00015 #define Model_ChargeI_H
00016
00017 #include "Misc.h"
00018
00020 class Model_ChargeI {
00021 public:
00023     Model_ChargeI();
00024     virtual ~Model_ChargeI();
00025
00026     // Get charge spreading parameters
00027     double Get_RC() const;
00028     double Get_Width() const;
00029     double Get_Gain() const;
00030
00031     // Set charge spreading parameters
00032     void Set_RC(const double &RC);
00033     void Set_Width(const double &Width);
00034     void Set_Gain(const double &Gain);
00035
00036     // Get width from Drift Distance
00037     void Set_WidthFromDriftDistance(const double &DriftDistance_cm);
00038
00039     // Get Charge
00040     virtual double Get_Qpad(const double &Time, const double &Time0,
00041                             const double &XTrue, const double &YTrue,
00042                             const double &XL, const double &XH, const double &YL,
00043                             const double &YH) const = 0;
00044
00045     // Big Dump
00046     virtual void WriteOut() const = 0;
00047
00048 protected:
00049     void WidthFromDriftDistance(const double &DriftDistance_cm);
00050     virtual void SetSecondaries() = 0;
00051
00052     double m_RC;
00053     double m_Width;
00054     double m_Gain;
00055
00056     double m_2RCinv;
00057     double m_Width2;
00058     double m_NormQPad;
00059 };
00060
00061 #endif

```

## 9.54 ModelSignal/inc/Model\_Electronics.h File Reference

Electronics response model declaration.

```
#include "Misc.h"
```

### Classes

- class [Model\\_Electronics](#)

### 9.54.1 Detailed Description

Electronics response model declaration.

Declares [Model\\_Electronics](#) which provides methods to configure impulse response parameters and evaluate the electronics time-shape and derivative. Implementation is in [Model\\_Electronics.cxx](#).

The implementation models the readout shaping and provides both the base impulse response and its derivative. These helpers are used by waveform simulation and fitting code and should match the digitiser settings used by the experiment.

## 9.55 Model\_Electronics.h

[Go to the documentation of this file.](#)

```

00001 #
00015 #ifndef Model_Electronics_H
00016 #define Model_Electronics_H
00017
00018 #include "Misc.h"
00019
00021 class Model_Electronics {
00022 public:
00024     Model_Electronics();
00025     virtual ~Model_Electronics();
00026
00027     // Get electronics parameters
00028     double Get_QValue() const;
00029     double Get_ws() const;
00030     double Get_Amplitude() const;
00031
00032     // Set electronics parameters
00033     void Set_TimeShape(const double &QValue, const double &ws);
00034     void Set_Amplitude(const double &Amplitude);
00035
00036     // Impulse response
00037     double Response_Base(const double &Time) const;
00038
00039     // Derivative of Impulse response
00040     double Derive_Response_Base(const double &Time) const;
00041
00042     // Big Dump
00043     void WriteOut() const;
00044
00045 private:
00046 private:
00047     void SetSecondaries();
00048     void GetExtremum();
00049
00050     double m_QValue;
00051     double m_ws;
00052     double m_Amplitude;
00053
00054     double m_Qfactor1;
00055     double m_Qfactor2;
00056     double m_Prefactor;
00057
00058     double m_AcurMax;
00059     double m_TcurMax;
00060 };
00061
00062 #endif

```

## 9.56 ModelSignal/inc/Model\_ReadOutGeometry.h File Reference

Readout geometry model declaration.

```
#include "Misc.h"
```

### Classes

- class [Model\\_ReadOutGeometry](#)

### 9.56.1 Detailed Description

Readout geometry model declaration.

Declares [Model\\_ReadOutGeometry](#) which describes pad layout, sizes and indexing. Provides utilities to query pad centre coordinates and edges.

The implementation contains utility functions to query pad centres and edges and supports swapping/squaring the geometry for different board layouts. Geometry parameters are used by simulation and reconstruction modules.

## 9.57 Model\_ReadOutGeometry.h

[Go to the documentation of this file.](#)

```

00001 #
00014 #ifndef Model_ReadOutGeometry_H
00015 #define Model_ReadOutGeometry_H
00016
00017 #include "Misc.h"
00018
00019 class Model_ReadOutGeometry {
00020 public:
00021     Model_ReadOutGeometry();
00022     virtual ~Model_ReadOutGeometry();
00023
00024     // Get Pad board data
00025     double Get_LX() const; // Get X Pad size (m)
00026     double Get LY() const; // Get Y Pad size (m)
00027
00028     double Get_Xpad_min() const; // Get Lower left corner X position (m)
00029     double Get_Ypad_min() const; // Get Lower left corner Y position (m)
00030
00031     int Get_Nx() const; // Get Nber of pads along X
00032     int Get_Ny() const; // Get Nber of pads along Y
00033
00034     // Swap Geometry
00035     void SwapGeometry(); // Swap Nx/Ny and Lx/Ly
00036
00037     // Square Geometry
00038     void SquareGeometry(); // Set Nx=Ny to the highest of the originals
00039
00040     // Set Pad board data
00041     void Set_LX(const double &LX); // Set X Pad size (m)
00042     void Set LY(const double &LY); // Set Y Pad size (m)
00043     void
00044     Set_Xpad_min(const double &Xpad_min); // Set Lower left corner X position (m)
00045     void
00046     Set_Ypad_min(const double &Ypad_min); // Set Lower left corner Y position (m)
00047
00048     void Set_Nx(const int &Nx); // Set Nber of pads along X
00049     void Set_Ny(const int &Ny); // Set Nber of pads along Y
00050
00051     // Pad Parameters
00052     double Get_XcPad(const int &iX, const int &iY,
00053                     const int &ModuleNber)
00054     const; // Get Pad(iX,iY) X position (m) of the center
00055     double Get_YcPad(const int &iX, const int &iY,
00056                     const int &ModuleNber)
00057     const; // Get Pad(iX,iY) Y position (m) of the center
00058
00059     double Get_XLPad(const int &iX, const int &iY,
00060                     const int &ModuleNber)
00061     const; // Get Pad(iX,iY) X position (m) of the Left edge
00062     double Get_YLPad(const int &iX, const int &iY,
00063                     const int &ModuleNber)
00064     const; // Get Pad(iX,iY) Y position (m) of the Bottom edge
00065
00066     double Get_XHPad(const int &iX, const int &iY,
00067                     const int &ModuleNber)
00068     const; // Get Pad(iX,iY) X position (m) of the Right edge
00069     double Get_YHPad(const int &iX, const int &iY,
00070                     const int &ModuleNber)
00071     const; // Get Pad(iX,iY) Y position (m) of the Top edge
00072
00073     void GetPadEdges(const int &iX, const int &iY, const int &ModuleNber,
00074                      double &XL, double &XH, double &YL,
00075                      double &YH) const; // Get Pad(iX,iY) X sides position (m)
00076
00077     // Big Dump
00078     void WriteOut() const;
00079
00080 private:
00081     private:
00082     double m_LX;
00083     double m LY;
00084     double m_Xpad_min;
00085     double m_Ypad_min;
00086     double m_Nx;
00087     double m_Ny;
00088
00089 };
00090
00091
00092 #endif

```

## 9.58 ModelSignal/src/Model\_Charge0D.cxx File Reference

```
#include "Model_Charge0D.h"
```

## 9.59 ModelSignal/src/Model\_Charge1D.cxx File Reference

```
#include "Model_Charge1D.h"
```

## 9.60 ModelSignal/src/Model\_ChargeI.cxx File Reference

```
#include "Model_ChargeI.h"
```

## 9.61 ModelSignal/src/Model\_Electronics.cxx File Reference

```
#include "Model_Electronics.h"
```

## 9.62 ModelSignal/src/Model\_ReadOutGeometrycxx File Reference

```
#include "Model_ReadOutGeometry.h"
```

## 9.63 Reconstruction/inc/CombinedFit.h File Reference

Combined fitting utilities declaration.

```
#include <Fit/BinData.h>
#include <Fit/Chi2FCN.h>
#include <Fit/Fitter.h>
#include <HFitInterface.h>
#include <Math/WrappedMultiTF1.h>
#include <TCanvas.h>
#include <TH1.h>
#include <TStyle.h>
```

### Functions

- void [combinedFit](#) (std::vector< TGraphErrors \* > &v\_tge, std::vector< TF1 \* > &v\_tf1)

### 9.63.1 Detailed Description

Combined fitting utilities declaration.

Declares the combinedFit helper used to fit multiple TGraphErrors/TF1 series together (e.g., high-level fits like Bethe-Bloch curve fits).

Implementation coordinates multiple TF1 instances and ROOT fit helpers to perform simultaneous fits across different data series (for example, fitting Bethe-Bloch parameters across particle species). It uses ROOT's fitting framework including WrappedMultiTF1 and the ROOT::Fit::Fitter interface and exposes a single helper function `combinedFit` which takes vectors of TGraphErrors and TF1 pointers and runs a global chi2 fit.

### 9.63.2 Function Documentation

#### 9.63.2.1 combinedFit()

```
void combinedFit (
    std::vector< TGraphErrors * > & v_tge,
    std::vector< TF1 * > & v_tf1)
```

## 9.64 CombinedFit.h

[Go to the documentation of this file.](#)

```
00001 #
00016 #ifndef COMBINEDFIT_H
00017 #define COMBINEDFIT_H
00018
00019 #include <Fit/BinData.h>
00020 #include <Fit/Chi2FCN.h>
00021 #include <Fit/Fitter.h>
00022 #include <HFitInterface.h>
00023 #include <Math/WrappedMultiTF1.h>
00024 #include <TCanvas.h>
00025 #include <TH1.h>
00026 #include <TStyle.h>
00027
00028 void combinedFit(std::vector<TGraphErrors *> &v_tge, std::vector<TF1 *> &v_tf1);
00029
00030 #endif // COMBINEDFIT_H
```

## 9.65 Reconstruction/inc/Control.h File Reference

High-level orchestration function declaration.

```
#include "Misc.h"
#include "GiveMe_Uploader.h"
#include "Uploader.h"
```

### Functions

- void `Control` (const std::string &OutDir, std::string const &Tag, std::string const &Comment, std::string const &EventFile, std::string const &SelectionSet, `Uploader` \*pUploader, int const &NbrOfMod, int const &Data\_to\_Use, int const &PT, int const &TB, const std::string &particle)

### 9.65.1 Detailed Description

High-level orchestration function declaration.

Declares the `Control` function used by top-level applications to set up runs and invoke reconstruction/analysis steps.

### 9.65.2 Function Documentation

#### 9.65.2.1 Control()

```
void Control (
    const std::string & OutDir,
    std::string const & Tag,
    std::string const & Comment,
    std::string const & EventFile,
    std::string const & SelectionSet,
    Uploader * pUploader,
    int const & NbrOfMod,
    int const & Data_to_Use,
    int const & PT,
    int const & TB,
    const std::string & particle)
```

## 9.66 Control.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #ifndef CONTROL_H
00009 #define CONTROL_H
00010
00011 #include "Misc.h"
00012
00013 #include "GiveMe_Uploader.h"
00014 #include "Uploader.h"
00015
00016 void Control(const std::string &OutDir, std::string const &Tag,
00017                 std::string const &Comment, std::string const &EventFile,
00018                 std::string const &SelectionSet, Uploader *pUploader,
00019                 int const &NbrOfMod, int const &Data_to_Use, int const &PT,
00020                 int const &TB, const std::string &particle);
00021
00022 #endif // CONTROL_H
```

## 9.67 Reconstruction/inc/dEdx.h File Reference

```
#include "Misc.h"
#include "LUTs.h"
#include "Variables.h"
#include <TBranch.h>
#include <TFile.h>
#include <TH1F.h>
#include <TObject.h>
#include <TROOT.h>
#include <TTree.h>
#include "Uploader.h"
```

## Classes

- class Reconstruction::RecoPad
- class Reconstruction::RecoCluster
- class Reconstruction::RecoModule
- class Reconstruction::RecoEvent
- class Reconstruction::dEdx

## Namespaces

- namespace Reconstruction
- dE/dx estimation and particle identification declarations.*

## 9.68 dEdx.h

Go to the documentation of this file.

```

00001 #
00016 #ifndef DEDX_H
00017 #define DEDX_H
00018
00019 #include "Misc.h"
00020
00021 #include "LUTs.h"
00022 #include "Variables.h"
00023
00024 #include < TBranch.h>
00025 #include <TFile.h>
00026 #include <TH1F.h>
00027 #include < TObject.h>
00028 #include < TROOT.h>
00029 #include < TTree.h>
00030
00031 #include "Uploader.h"
00032
00033 namespace Reconstruction {
00034
00035 class RecoPad : public TObject {
00036 public:
00037     virtual ~RecoPad();
00038
00039     bool leading = false;
00040     int ix = 0;
00041     int iy = 0;
00042     float xPad = 0; // in cm
00043     float yPad = 0; // in cm
00044     float ADCmax = 0;
00045     float ADCmax_base = 0;
00046     float charge = 0;
00047     float dEdxXP = 0;
00048     float RC = 0;
00049     float gain = 0;
00050     float GainCorrection = 0;
00051     float phi = 0;
00052     float d = 0;
00053     float dd = 0;
00054     float length = 0;
00055     int TMax = 0;
00056     int T0 = 0;
00057     float driftDistance = 0;
00058     float ratioDrift = 0;
00059     float ratioFile = 0;
00060     float ratio = 0;
00061     float dy = 0; // distance to the track in y
00062
00063     ClassDef(RecoPad, 1);
00064 };
00065
00066 class RecoCluster : public TObject {
00067 public:
00068     virtual ~RecoCluster();
00069
00070     std::vector<RecoPad *> v_pads;
00071     float length = 0;

```

```
00072     float charge = 0;
00073     float ADCmax_base = 0;
00074     float ALead_base = 0;
00075     float ALead_GCorr = 0;
00076     int TLead = 0;
00077     float dEdxWF = 0;
00078     int NPads = 0;
00079     float ratioCorr = 0;
00080     float yCluster = 0;
00081     float yWeight = 0;
00082     float LUTrhoLead = 0;
00083
00084     ClassDef(RecoCluster, 1);
00085 };
00086
00087 class RecoModule : public TObject {
00088 public:
00089     virtual ~RecoModule();
00090
00091     std::vector<RecoCluster *> v_clusters;
00092     bool selected = false;
00093     int ID = 0;
00094     int position = 0;
00095     float dEdxXP = 0;
00096     float dEdxWF = 0;
00097     float dEdxXPnoTrunc = 0;
00098     float dEdxWFnoTrunc = 0;
00099     int NCrossedPads = 0;
00100    int NClusters = 0;
00101    int NPads = 0;
00102    float lengthXP = 0;
00103    float lengthWF = 0;
00104    float phi = 0;
00105    float avg_pad_mult = 0;
00106    TF1 *Track = new TF1("pTrackFit", "[0]*x+[1]+[2]*x*x", 0, 2000);
00107
00108    ClassDef(RecoModule, 1);
00109 };
00110
00111 class RecoEvent : public TObject {
00112 public:
00113     RecoEvent();
00114     virtual ~RecoEvent();
00115     void Clear();
00116
00117     std::vector<RecoModule *> v_modules;
00118     std::vector<int> v_modules_position;
00119     bool selected = false;
00120     int eventNbr = 0;
00121     float dEdxXP = 0;
00122     float dEdxWF = 0;
00123     float dEdxGP1 = 0;
00124     float dEdxGP2 = 0;
00125     float dEdxGP3 = 0;
00126     float dEdxGP4 = 0;
00127     float dEdxGP5 = 0;
00128     float dEdxXPnoTrunc = 0;
00129     float dEdxWFnoTrunc = 0;
00130     int NCrossedPads = 0;
00131     int NPads = 0;
00132     int NClusters = 0;
00133     float lengthXP = 0;
00134     float lengthWF = 0;
00135     int numberOfModules = 0;
00136     float avg_pad_mult = 0;
00137
00138     // Giga Pad variables
00139     TH1F *GWF = nullptr;
00140     TH1F *GWFtruncatedGP1 = nullptr;
00141     int peakingTime = 0; // in ns
00142     int timeBinSize = 0; // in ns
00143
00144     ClassDef(RecoEvent, 1);
00145 };
00146
00147 class dEdx {
00148 public:
00149     dEdx();
00150     virtual ~dEdx();
00151     void Reconstruction();
00152     float ComputedEdxWF(std::vector<float> v_dEdxWF, const int &NClusters,
00153                           const float &alpha);
00154     float ComputedEdxXP(const std::vector<float> &v_dEdx,
00155                           const std::vector<float> &v_dE,
00156                           const std::vector<float> &v_dx, const int &nCrossedPads,
00157                           const float &alpha);
00158 }
```

```

00159 // GP1: sum waveforms of bottom 70% clusters
00160 float ComputedEdxGP1(const std::vector<TH1F> &vClusWF,
00161           const std::vector<float> &v_dEdx,
00162           const std::vector<float> &v_Aclus,
00163           const std::vector<float> &v_Lclus, const int &nClusters,
00164           const float &alpha);
00165
00166 // GP2: Remove from total GWF the ETF of the top 30% clusters
00167 // GP3: Remove from total GWF the ETF of the top 30% clusters computed as
00168 // Alead*ratio(y_barycenter)
00169 // GP4: Remove from total GWF the ETF of the top 30% clusters computed with
00170 // LUT GP5: Remove from total GWF the ETF of the top 30% crossed pads computed
00171 // with LUT
00172 float ComputedEdxGP(const TH1F *GWF, const std::vector<TH1F> &vWF,
00173           const std::vector<float> &v_dEdx,
00174           const std::vector<float> &v_Amax,
00175           const float &eventLength,
00176           const std::vector<float> &v_Length, const int &nElements,
00177           const float &alpha);
00178
00179 TH1F *GetGigaWaveform(const std::vector<TH1F> &vClusWF);
00180 TH1F *GetTruncatedGigaWaveformGP1(const std::vector<TH1F> &vClusWF,
00181           const std::vector<float> &v_dEdx,
00182           const int &nClusters);
00183
00184 void DiscardedModule();
00185
00186 private:
00187   // Output file variables
00188   TFile *fpFile_dEdx;
00189   TTree *fpTree_dEdx;
00190
00191   // Setup variables
00192   static constexpr float AVG_GAIN =
00193     1947.72; // average gain of the 32 mounted ERAMs
00194   static constexpr float PHIMAX = 42.10; // std::atan(10.19/11.28)*180/M_PI;
00195   static constexpr float falpha = 0.7;
00196   static constexpr float fnParamsTrack = 3;
00197   static constexpr float fminLength = 0.002;
00198   static constexpr float XPADLENGTH = 11.28; // mm
00199   static constexpr float YPADLENGTH = 10.19; // mm
00200
00201   // Input classes
00202   Event *pEvent;
00203   Module *pModule;
00204   const Track *pTrack;
00205   Cluster *pCluster;
00206   Pad *pPad;
00207
00208   // Reconstruction classes
00209   Reconstruction::RecoEvent *p_recoevent;
00210   Reconstruction::RecoModule *p_recomodule;
00211   Reconstruction::RecoCluster *p_recocluster;
00212   Reconstruction::RecoPad *p_recopad;
00213
00214   // ERAM mapping
00215   Reconstruction::ERAMMaps *pERAMMaps;
00216   std::vector<int> fERAMs_ID;
00217   std::vector<int> fERAMs_pos;
00218
00219   // Diagonal variables
00220   bool diag;
00221   float costheta;
00222   std::vector<int> v_theta;
00223
00224   // WF correction variables
00225   TF1 *pcorrFunctionWF;
00226   float fAref;
00227
00228   // Fitting variables
00229   TF1 *ptf1PRF;
00230   int fcounterFit;
00231   int fcounterFail;
00232
00233   // Analysis variables //
00234   // Iterators
00235   int NEvents;
00236   int iEvent;
00237   float fmodID;
00238   int NMod;
00239   int iMod;
00240   int NClusters;
00241   int iC;
00242   int NPads;
00243   int iP;
00244   // Modules
00245   std::vector<float> v_mod_dx;

```

```

00246     std::vector<float> v_mod_dE;
00247     std::vector<float> v_mod_dEdxXP;
00248     std::vector<float> v_mod_dEdxWF;
00249 // Events
00250     std::vector<float> v_evt_dx;
00251     std::vector<float> v_evt_dE;
00252     std::vector<float> v_evt_dEdxXP;
00253     std::vector<float> v_evt_dEdxWF;
00254 // Waveforms
00255     std::vector<float> waveform_cluster;
00256     std::vector<float> waveform_pad;
00257 // histograms
00258     TH1F *phlf_WF;
00259     TH1F *phlf_XP;
00260     TH1F *phlf_GP1;
00261     TH1F *phlf_GP2;
00262     TH1F *phlf_GP3;
00263     TH1F *phlf_GP4;
00264     TH1F *phlf_GP5;
00265 };
00266 } // namespace Reconstruction
00267
00268 #endif

```

## 9.69 Reconstruction/inc/Displays.h File Reference

High-level display utilities and plotting helpers.

```

#include "Misc.h"
#include "Event.h"
#include "Module.h"
#include "Pad.h"
#include "Sample.h"
#include "TH1F.h"
#include "TH2D.h"

```

### Functions

- void **DrawOut\_EventDisplay** (*Event* \*pEvent, const int &ModuleNber, const std::string &OUTDIR, const std::string &TAG)
 

*Output an event display for a given event.*
- void **DrawOut\_EventDisplay** (*Module* \*pModule, const std::string &OUTDIR, const std::string &TAG, const std::string &type, const double &parabola, const double &slope, const double &intercept)
 

*Output an event display for a module.*
- *TH2D* \* **GiveMe\_EvtDisplay** (*Event* \*pEvent, const int &ModuleNber, const std::string &TAG)
 

*Get an event display TH2D for an event.*
- *TH2D* \* **GiveMe\_EvtDisplay** (*Module* \*pModule, const std::string &TAG)
 

*Get an event display TH2D for a module.*
- *TH2D* \* **GiveMe\_TimeDisplay** (*Module* \*pModule, const std::string &TAG)
- void **DrawOut\_WaveFormDisplay** (*Pad* \*pPad, const std::string &OUTDIR, const std::string &TAG)
- *TH1F* \* **GiveMe\_WaveFormDisplay** (*Pad* \*pPad, const std::string &TAG)
- void **DrawOut\_ClusterWFDisplay** (*Cluster* \*pCluster, const std::string &OUTDIR, const std::string &TAG)
- void **DrawOut\_ClusterWFDisplay** (*Cluster* \*pCluster, const std::string &OUTDIR, const std::string &TAG, const int &Option, const int &PT, const int &TB)
- void **DrawOut\_GWF** (*Event* \*pEvent, const int &ModuleNber, const std::string &OUTDIR, const std::string &TAG, const int &PT, const int &TB, const float &phi\_rad)
- void **NewClusterDisplay** (*Event* \*pEvent, const std::string &OUTDIR, const std::string &TAG, const int &PT, const int &TB)
- void **NewClusterDisplayMinimal** (*Event* \*pEvent, const std::string &OUTDIR, const std::string &TAG)

## 9.69.1 Detailed Description

High-level display utilities and plotting helpers.

Declares functions and helpers used by the GUI and plotting components of the reconstruction pipeline (event displays, waveform displays, cluster visualisations, etc.).

The implementation in `Displays.cxx` contains high-level presentation code that constructs ROOT-based canvases, histograms and graphical summaries. This is primarily display code that depends on ROOT and should remain separate from core reconstruction algorithms.

## 9.69.2 Function Documentation

### 9.69.2.1 DrawOut\_ClusterWFDisplay() [1/2]

```
void DrawOut_ClusterWFDisplay (
    Cluster * pCluster,
    const std::string & OUTDIR,
    const std::string & TAG)
```

### 9.69.2.2 DrawOut\_ClusterWFDisplay() [2/2]

```
void DrawOut_ClusterWFDisplay (
    Cluster * pCluster,
    const std::string & OUTDIR,
    const std::string & TAG,
    const int & Option,
    const int & PT,
    const int & TB)
```

### 9.69.2.3 DrawOut\_EventDisplay() [1/2]

```
void DrawOut_EventDisplay (
    Event * pEvent,
    const int & ModuleNber,
    const std::string & OUTDIR,
    const std::string & TAG)
```

Output an event display for a given event.

#### Parameters

<i>pEvent</i>	pointer to the <code>Event</code>
<i>ModuleNber</i>	module index
<i>OUTDIR</i>	output directory path
<i>TAG</i>	tagging string used in output filenames

### 9.69.2.4 DrawOut\_EventDisplay() [2/2]

```
void DrawOut_EventDisplay (
    Module * pModule,
    const std::string & OUTDIR,
    const std::string & TAG,
    const std::string & type,
    const double & parabola,
    const double & slope,
    const double & intercept)
```

Output an event display for a module.

### 9.69.2.5 DrawOut\_GWF()

```
void DrawOut_GWF (
    Event * pEvent,
    const int & ModuleNber,
    const std::string & OUTDIR,
    const std::string & TAG,
    const int & PT,
    const int & TB,
    const float & phi_rad)
```

### 9.69.2.6 DrawOut\_WaveFormDisplay()

```
void DrawOut_WaveFormDisplay (
    Pad * pPad,
    const std::string & OUTDIR,
    const std::string & TAG)
```

### 9.69.2.7 GiveMe\_EvtDisplay() [1/2]

```
TH2D * GiveMe_EvtDisplay (
    Event * pEvent,
    const int & ModuleNber,
    const std::string & TAG)
```

Get an event display TH2D for an event.

### 9.69.2.8 GiveMe\_EvtDisplay() [2/2]

```
TH2D * GiveMe_EvtDisplay (
    Module * pModule,
    const std::string & TAG)
```

Get an event display TH2D for a module.

### 9.69.2.9 GiveMe\_TimeDisplay()

```
TH2D * GiveMe_TimeDisplay (
    Module * pModule,
    const std::string & TAG)
```

### 9.69.2.10 GiveMe\_WaveFormDisplay()

```
TH1F * GiveMe_WaveFormDisplay (
    Pad * pPad,
    const std::string & TAG)
```

### 9.69.2.11 NewClusterDisplay()

```
void NewClusterDisplay (
    Event * pEvent,
    const std::string & OUTDIR,
    const std::string & TAG,
    const int & PT,
    const int & TB)
```

### 9.69.2.12 NewClusterDisplayMinimal()

```
void NewClusterDisplayMinimal (
    Event * pEvent,
    const std::string & OUTDIR,
    const std::string & TAG)
```

## 9.70 Displays.h

[Go to the documentation of this file.](#)

```
00001
00015
00016 #ifndef DISPLAYS_H
00017 #define DISPLAYS_H
00018 #include "Misc.h"
00019
00020 #include "Event.h"
00021 #include "Module.h"
00022 #include "Pad.h"
00023 #include "Sample.h"
00024
00025 #include "TH1F.h"
00026 #include "TH2D.h"
00027
00028 // Event display
00029 void DrawOut_EventDisplay(Event *pEvent, const int &ModuleNber,
00030                           const std::string &OUTDIR, const std::string &TAG);
00031
00032 void DrawOut_EventDisplay(Module *pModule, const std::string &OUTDIR,
00033                           const std::string &TAG, const std::string &type,
00034                           const double &parabola, const double &slope,
00035                           const double &intercept);
00036
00037 TH2D *GiveMe_EvtDisplay(Event *pEvent, const int &ModuleNber,
00038                           const std::string &TAG);
00039
00040 TH2D *GiveMe_EvtDisplay(Module *pModule, const std::string &TAG);
```

```
00041 // Get Timing Event display
```

```

00054 TH2D *GiveMe_TimeDisplay(Module *pModule, const std::string &TAG);
00055
00056 //-----WF pad Display
00057 // Output waveform of a pad with tagging string TAG, placed in OUTDIR dir
00058 void DrawOut_WaveFormDisplay(Pad *pPad, const std::string &OUTDIR,
00059                                     const std::string &TAG);
00060
00061 // Get waveform histo of pad with tagging string TAG
00062 TH1F *GiveMe_WaveFormDisplay(Pad *pPad, const std::string &TAG);
00063
00064 //-----WFs cluster Display
00065 // TD: Draw Waveforms of all Pads in a Cluster
00066 void DrawOut_ClusterWFDisplay(Cluster *pCluster, const std::string &OUTDIR,
00067                                     const std::string &TAG);
00068 void DrawOut_ClusterWFDisplay(Cluster *pCluster, const std::string &OUTDIR,
00069                                     const std::string &TAG, const int &Option,
00070                                     const int &PT, const int &TB);
00071
00072 // Draw Event waveform
00073 void DrawOut_GWF(Event *pEvent, const int &ModuleNber,
00074                      const std::string &OUTDIR, const std::string &TAG,
00075                      const int &PT, const int &TB, const float &phi_rad);
00076
00077 // Redo Cluster display
00078 void NewClusterDisplay(Event *pEvent, const std::string &OUTDIR,
00079                         const std::string &TAG, const int &PT, const int &TB);
00080
00081 void NewClusterDisplayMinimal(Event *pEvent, const std::string &OUTDIR,
00082                                const std::string &TAG);

```

## 9.71 Reconstruction/inc/DrawOuts.h File Reference

Declaration of the DrawOuts helper class for plotting and summaries.

```

#include "Misc.h"
#include "SetStyle.h"
#include "TLegend.h"
#include "dEdx.h"

```

### Classes

- class **Reconstruction::DrawOuts**  
*Plotting helper for reconstruction results.*

### Namespaces

- namespace **Reconstruction**  
*dE/dx estimation and particle identification declarations.*

### 9.71.1 Detailed Description

Declaration of the DrawOuts helper class for plotting and summaries.

The DrawOuts class provides routines for creating plots and visual summaries of reconstruction results (scan drawing, energy loss plots, comparisons, etc.). Implementation and plotting logic live in [DrawOuts.cxx](#).

A small top-level executable **DrawOut** exists ([Apps/DrawOut.cxx](#)) which calls into **Reconstruction::DrawOuts** as the runner for generating visual outputs. The class itself provides fine-grained control functions used by the executable and by any other consumer of plotting helpers.

## 9.72 DrawOuts.h

[Go to the documentation of this file.](#)

```

00001
00016
00017 #ifndef DRAWOUTS_H
00018 #define DRAWOUTS_H
00019
00020 #include "Misc.h"
00021 #include "SetStyle.h"
00022 #include "TLegend.h"
00023 #include "dEdx.h"
00024 namespace Reconstruction {
00025
00033 class DrawOuts {
00034 public:
00036     DrawOuts();
00037
00042     DrawOuts(const std::string &inputFile);
00043
00048     DrawOuts(const std::vector<std::string> &v_inputFiles);
00049
00051     virtual ~DrawOuts();
00052
00054     void SetStyle();
00055
00057     void DESY21ScanFill();
00058
00060     void DESY21ScanDraw();
00061
00063     void CERN22ScanFill();
00064
00066     void CERN22ScanDraw();
00067
00069     void Control();
00070
00075     void EnergyLoss(const int &methods = 0);
00076
00078     void FileComparison();
00079
00084     void DESY21SingleScan(const int &methods = 0);
00085
00090     void DESY21MultiScan(const int &methods = 0);
00091
00093     void CERN22Scan();
00094
00096     void AmplitudeVSLength();
00097
00098 private:
00099 // Files
00100 std::string finputFile;
00101 std::string foutputDir;
00102 std::string foutputFile;
00103 std::string foutputFileWF;
00104 std::string foutputFileXP;
00105
00106 // Data file
00107 std::vector<TFile *> v_fFiles;
00108 std::vector<TTree *> v_fTrees;
00109 std::vector<RecoEvent *> v_fEvents;
00110 std::vector<int> v_fnentries;
00111
00112 // Reconstruction classes
00113 Reconstruction::RecoEvent *p_recoevent;
00114 Reconstruction::RecoModule *p_recomodule;
00115 Reconstruction::RecoCluster *p_recocluster;
00116 Reconstruction::RecoPad *p_recopad;
00117
00118 // Settings
00119 std::string fparticleType;
00120 TCanvas *fpCanvas;
00121 int drawMultiScans = 0; // 0: normal scan | 1: multi scan
00122 int fwhichMethods = 0; // 0: both methods | 1: only WF | 2: only XP
00123 int fnMethods = 2; // Number of methods: 2 (WF and XP) or 1 (WF or XP)
00124
00125 // Shared variables
00126 int NMod = 0;
00127 int NClusters = 0;
00128 int NPads = 0;
00129 int ix = 0;
00130 int iy = 0;
00131 int position = 0;
00132
00133 constexpr static float YRESOMAX = 11;
00134 constexpr static float YRESOMIN = 6.5;

```

```

00135  constexpr static float YMEANMAX = 1000;
00136  constexpr static float YMEANMIN = 550;
00137  constexpr static float YSTDMAX = 90;
00138  constexpr static float YSTDMIN = 40;
00139
00140  constexpr static float YRESOMAXCERN = 7;
00141  constexpr static float YRESOMINCERN = 3;
00142  constexpr static float YMEANMAXCERN = 3;
00143  constexpr static float YMEANMINCERN = 0.5;
00144  constexpr static float YSTDMAXCERN = 0.15;
00145  constexpr static float YSTDMINCERN = 0;
00146  float keV = 5.9 / (224 * 1703.74 /
00147          183); // 5.9 Fe peak energy | 1703 mean MockUp gain | 224
00148          // e- created with 5.9keV | 183 e- for 1 ADC
00149
00150  // Single scan pointers
00151  TF1 *fptf1_WF;
00152  TF1 *fptf1_XP;
00153  TGraphErrors *fpTGE_reso_WF;
00154  TGraphErrors *fpTGE_reso_XP;
00155  TGraphErrors *fpTGE_mean_WF;
00156  TGraphErrors *fpTGE_mean_XP;
00157  TGraphErrors *fpTGE_std_WF;
00158  TGraphErrors *fpTGE_std_XP;
00159
00160  // Multiple scan pointers
00161  int nScans = 0;
00162  int nRuns = 0;
00163  std::vector<int> markers = {22, 34, 23, 47, 33, 43};
00164  std::vector<int> colors = {kCyan - 6, kMagenta - 6, kCyan + 2,
00165          kMagenta + 2, kCyan + 3, kMagenta + 3};
00166  std::vector<int> markersCERN = {20, 47, 34, 21};
00167  std::vector<int> colorsCERN = {kOrange + 7, kAzure - 6, kSpring - 6,
00168          kRed + 1};
00169  std::vector<TF1 *> v_fptf1_WF;
00170  std::vector<TF1 *> v_fptf1_XP;
00171  std::vector<TGraphErrors *> v_fpTGE_mean_WF;
00172  std::vector<TGraphErrors *> v_fpTGE_mean_XP;
00173  std::vector<TGraphErrors *> v_fpTGE_std_WF;
00174  std::vector<TGraphErrors *> v_fpTGE_std_XP;
00175  std::vector<TGraphErrors *> v_fpTGE_reso_WF;
00176  std::vector<TGraphErrors *> v_fpTGE_reso_XP;
00177
00178  // Bethe-Bloch fitting
00179  std::vector<TF1 *> v_fptf1_BB;
00180 };
00181 } // namespace Reconstruction
00182
00183 #endif

```

## 9.73 Reconstruction/inc/DrawOuts\_old.h File Reference

Legacy plotting helpers (kept for reference).

```
#include "Misc.h"
#include "SetStyle.h"
#include "dEdx.h"
```

### Functions

- void [DrawOut\\_Methods](#) (const std::string &OutDir, const std::string &Tag, const std::string &Comment, const int &nMod)
- void [DrawOut\\_Versions](#) (const std::string &inputDir, const std::string &Method, const std::string &Comment1, const std::string &Comment2)
- void [DrawOut\\_Separation](#) (const std::string &inputDir, const std::string &Comment)
- void [DrawOut\\_Separation\\_Reduced](#) (const std::string &inputDir, const std::string &Comment, std::string Energy)
- void [DrawOut\\_Scans](#) (const std::string &inputDir, const std::string &Comment, const std::string &WFversion)
- void [DrawOut\\_Zscan](#) (const std::string &inputDir, const std::string &Comment, const int &PT)

- void `DrawOut_Yscan` (const std::string &inputDir, const std::string &Comment)
- void `DrawOut_Phiscan` (const std::string &inputDir, const std::string &Comment, const std::string &WFversion, const std::string &zdrift)
- void `DrawOut_Thetascan` (const std::string &inputDir, const std::string &Comment)
- void `DrawOut_Zscan_PT` (const std::string &inputDir, const std::string &Comment)
- void `DrawOut_Phiscan_Z` (const std::string &inputDir, const std::string &Comment, const std::string &WFversion)
- void `DrawOut_Escan` (const std::string &inputDir, const std::string &Comment)
- void `DrawOut_Systematics` (const std::string &inputDir, const std::string &Comment, const std::string &scan)

### 9.73.1 Detailed Description

Legacy plotting helpers (kept for reference).

Contains an older variant of the DrawOuts plotting helper kept for compatibility and reference. Prefer `DrawOuts` for new code.

### 9.73.2 Function Documentation

#### 9.73.2.1 `DrawOut_Escan()`

```
void DrawOut_Escan (
    const std::string & inputDir,
    const std::string & Comment)
```

#### 9.73.2.2 `DrawOut_Methods()`

```
void DrawOut_Methods (
    const std::string & OutDir,
    const std::string & Tag,
    const std::string & Comment,
    const int & nMod)
```

#### 9.73.2.3 `DrawOut_Phiscan()`

```
void DrawOut_Phiscan (
    const std::string & inputDir,
    const std::string & Comment,
    const std::string & WFversion,
    const std::string & zdrift)
```

#### 9.73.2.4 `DrawOut_Phiscan_Z()`

```
void DrawOut_Phiscan_Z (
    const std::string & inputDir,
    const std::string & Comment,
    const std::string & WFversion)
```

### 9.73.2.5 DrawOut\_Scans()

```
void DrawOut_Scans (
    const std::string & inputDir,
    const std::string & Comment,
    const std::string & WFversion)
```

### 9.73.2.6 DrawOut\_Separation()

```
void DrawOut_Separation (
    const std::string & inputDir,
    const std::string & Comment)
```

### 9.73.2.7 DrawOut\_Separation\_Reduced()

```
void DrawOut_Separation_Reduced (
    const std::string & inputDir,
    const std::string & Comment,
    std::string Energy)
```

### 9.73.2.8 DrawOut\_Systematics()

```
void DrawOut_Systematics (
    const std::string & inputDir,
    const std::string & Comment,
    const std::string & scan)
```

### 9.73.2.9 DrawOut\_Thetascan()

```
void DrawOut_Thetascan (
    const std::string & inputDir,
    const std::string & Comment)
```

### 9.73.2.10 DrawOut\_Versions()

```
void DrawOut_Versions (
    const std::string & inputDir,
    const std::string & Method,
    const std::string & Comment1,
    const std::string & Comment2)
```

### 9.73.2.11 DrawOut\_Yscan()

```
void DrawOut_Yscan (
    const std::string & inputDir,
    const std::string & Comment)
```

### 9.73.2.12 DrawOut\_Zscan()

```
void DrawOut_Zscan (
    const std::string & inputDir,
    const std::string & Comment,
    const int & PT)
```

### 9.73.2.13 DrawOut\_Zscan\_PT()

```
void DrawOut_Zscan_PT (
    const std::string & inputDir,
    const std::string & Comment)
```

## 9.74 DrawOuts\_old.h

[Go to the documentation of this file.](#)

```
00001
00008
00009 #ifndef DRAWOUTSOLD_H
00010 #define DRAWOUTSOLD_H
00011
00012 #include "Misc.h"
00013 #include "SetStyle.h"
00014 #include "dEdx.h"
00015
00016 // TH2 comparisons of the different methods
00017 void DrawOut_Methods(const std::string &OutDir, const std::string &Tag,
00018                      const std::string &Comment, const int &nMod);
00019
00020 // Draw resolution of 2 different procedures
00021 void DrawOut_Versions(const std::string &inputDir, const std::string &Method,
00022                        const std::string &Comment1, const std::string &Comment2);
00023
00024 // Draw separation power histograms for all methods and particles
00025 void DrawOut_Separation(const std::string &inputDir,
00026                           const std::string &Comment);
00027
00028 // Draw separation power histograms for all methods and particles only for WF &
00029 // XP
00030 void DrawOut_Separation_Reduced(const std::string &inputDir,
00031                                   const std::string &Comment, std::string Energy);
00032
00033 // Draw all scans together
00034 void DrawOut_Scans(const std::string &inputDir, const std::string &Comment,
00035                      const std::string &WFversion);
00036
00037 // Draw resolution as function of Z scan
00038 void DrawOut_Zscan(const std::string &inputDir, const std::string &Comment,
00039                      const int &PT);
00040
00041 // Draw resolution as function of Y scan
00042 void DrawOut_Yscan(const std::string &inputDir, const std::string &Comment);
00043
00044 // Draw resolution as function of phi scan
00045 void DrawOut_Phiscan(const std::string &inputDir, const std::string &Comment,
00046                        const std::string &WFversion, const std::string &zdrift);
00047
00048 // Draw resolution as function of theta scan
00049 void DrawOut_Thetascan(const std::string &inputDir, const std::string &Comment);
00050
00051 // Transverse diffusion coefficient effect on Z scan
00052 void DrawOut_Zscan_PT(const std::string &inputDir, const std::string &Comment);
00053
00054 // Draw resolution as function of phi scan for different drift distances
00055 void DrawOut_Phiscan_Z(const std::string &inputDir, const std::string &Comment,
00056                         const std::string &WFversion);
00057
00058 // Draw Energy scan
00059 void DrawOut_Escan(const std::string &inputDir, const std::string &Comment);
00060
00061 // DrawOut dE/dx systematics with Z scan
00062 void DrawOut_Systematics(const std::string &inputDir,
```

```

00063             const std::string &Comment, const std::string &scan);
00064
00065 #endif
00066 //*****
00067 * File: DrawOuts_old.h
00068 * Project: dEdxRecon
00069 *
00070 * Brief: Legacy/older drawout interfaces kept for backward compatibility.
00071 * Declares older drawing utilities that may be used by legacy
00072 * workflows or for regression comparisons.
00073 *
00074 * Contents: legacy drawing class/function declarations.
00075 *
00076 * Notes: Prefer the newer DrawOuts API; this header is retained for
00077 * compatibility.
00078 *****/
00079
00080 #ifndef DRAWOUTSOLD_H
00081 #define DRAWOUTSOLD_H
00082
00083 #include "Misc.h"
00084 #include "SetStyle.h"
00085 #include "dEdx.h"
00086
00087 // TH2 comparisons of the different methods
00088 void DrawOut_Methods(const std::string &OutDir, const std::string &Tag,
00089                      const std::string &Comment, const int &nMod);
00090
00091 // Draw resolution of 2 different procedures
00092 void DrawOut_Versions(const std::string &inputDir, const std::string &Method,
00093                        const std::string &Comment1, const std::string &Comment2);
00094
00095 // Draw separation power histograms for all methods and particles
00096 void DrawOut_Separation(const std::string &inputDir,
00097                          const std::string &Comment);
00098
00099 // Draw separation power histograms for all methods and particles only for WF &
00100 // XP
00101 void DrawOut_Separation_Reduced(const std::string &inputDir,
00102                                   const std::string &Comment, std::string Energy);
00103
00104 // Draw all scans together
00105 void DrawOut_Scans(const std::string &inputDir, const std::string &Comment,
00106                      const std::string &WFversion);
00107
00108 // Draw resolution as function of Z scan
00109 void DrawOut_Zscan(const std::string &inputDir, const std::string &Comment,
00110                      const int &PT);
00111
00112 // Draw resolution as function of Y scan
00113 void DrawOut_Yscan(const std::string &inputDir, const std::string &Comment);
00114
00115 // Draw resolution as function of phi scan
00116 void DrawOut_Phiscan(const std::string &inputDir, const std::string &Comment,
00117                        const std::string &WFversion, const std::string &zdrift);
00118
00119 // Draw resolution as function of theta scan
00120 void DrawOut_Thetascan(const std::string &inputDir, const std::string &Comment);
00121
00122 // Transverse diffusion coefficient effect on Z scan
00123 void DrawOut_Zscan_PT(const std::string &inputDir, const std::string &Comment);
00124
00125 // Draw resolution as function of phi scan for different drift distances
00126 void DrawOut_Phiscan_Z(const std::string &inputDir, const std::string &Comment,
00127                         const std::string &WFversion);
00128
00129 // Draw Energy scan
00130 void DrawOut_Escan(const std::string &inputDir, const std::string &Comment);
00131
00132 // DrawOut dE/dx systematics with Z scan
00133 void DrawOut_Systematics(const std::string &inputDir,
00134                           const std::string &Comment, const std::string &scan);
00135
00136#endif

```

## 9.75 Reconstruction/inc/LUTs.h File Reference

```
#include <TBranch.h>
#include <TFile.h>
#include <TTree.h>
```

```
#include <string>
```

## Classes

- class [EramInfo](#)  
*Lookup table utilities used by reconstruction algorithms.*
- class [Reconstruction::ERAMMaps](#)
- class [Reconstruction::LUT](#)

## Namespaces

- namespace [Reconstruction](#)  
*dE/dx estimation and particle identification declarations.*

## 9.76 LUTs.h

Go to the documentation of this file.

```
00001 #
00015 #ifndef LUT_H
00016 #define LUT_H
00017
00018 #include < TBranch.h>
00019 #include <TFile.h>
00020 #include <TTree.h>
00021 #include <string>
00022
00023 class EramInfo : public TObject {
00024 public:
00025     int Id;
00026     int Position;
00027     int XX;
00028     int YY;
00029     float RC;
00030     float Gain;
00031     float Resolution;
00032
00033     int Endplate;
00034     bool InbTPC;
00035     bool IntTPC;
00036
00037     EramInfo() {}
00038     virtual ~EramInfo() {}
00039
00040     ClassDef(EramInfo, 2)
00041 };
00042
00043 namespace Reconstruction {
00044
00045 /* ERAM MAPS READING
00046 */
00047 */
00048 class ERAMMaps {
00049 public:
00050     /* Constructor */
00051     ERAMMaps(
00052         const std::string &file = "$HOME/Documents/Code/ERAMinfo/ERAMinfo.root");
00053     virtual ~ERAMMaps();
00054
00055     int ID(const int &position);
00056     float RC(const int &position, const int &iX, const int &iY);
00057     float Gain(const int &position, const int &iX, const int &iY);
00058     float Resolution(const int &position, const int &iX, const int &iY);
00059     float MeanGain(const int &position);
00060     float MeanRC(const int &position);
00061
00062 private:
00063     bool verbose = false;
00064     std::string fFile;
```

```

00065     std::vector<int> fID;
00066     std::vector<float> v_sides;
00067
00068     float fGain[34][36][32]; // [position][iX][iY]
00069     float fRC[34][36][32];
00070     float fResolution[34][36][32];
00071     std::vector<float> fmean_gain;
00072     std::vector<float> fmean_RC;
00073
00074     void Load();
00075     void setGain(const int &position, const int &iX, const int &iY,
00076                  const float &gain);
00077     void setRC(const int &position, const int &iX, const int &iY,
00078                const float &RC);
00079     void setResolution(const int &position, const int &iX, const int &iY,
00080                         const float &resolution);
00081     void setMeanGain(const int &position, const float &meanGain);
00082     void setMeanRC(const int &position, const float &meanRC);
00083     void FillHoles();
00084
00085     std::vector<int> channel2iD = {24, 30, 28, 19, 21, 13, 9, 2, 26,
00086                                    17, 23, 29, 1, 10, 11, 3, /*bottom HATPC*/
00087                                    47, 16, 14, 15, 42, 45, 37, 36, 20,
00088                                    38, 7, 44, 43, 39, 41, 46, /*top HATPC*/
00089                                    12, 18}; // CERN22 MockUp and prototype
00090
00091 //-----Data Members-----//
00092 TFile *pFile;
00093 TTree *pTree;
00094 TBranch *pBranch;
00095
00096 // Leaves
00097 int fid;
00098 int fpos;
00099 int fx;
00100 int fy;
00101 float frc;
00102 float fgain;
00103 float fres;
00104 };
00105
00106 /* Look Up Tables for XP method
00107 */
00108 */
00109 class LUT {
00110 public:
00111     /* Constructor */
00112     LUT(const int &transDiffCoeffB, const int &transDiffCoeffnoB,
00113          const int &peakingTime);
00114     LUT(const std::string &file);
00115
00116     virtual ~LUT();
00117
00118     float getRatio(const int &Dt, const int &RC, const float &drift,
00119                     const float &impact, const float &angle);
00120
00121 private:
00122     std::string fFile_LUT;
00123
00124     static constexpr float PAD_DIAG = 15.20; // sqrt(pow(11.28,2) + pow(10.19, 2))
00125
00126     static constexpr int RCmin = 80;
00127     static constexpr int RCmax = 160;
00128
00129     // Number of discrete steps in each dimension of the Look Up Table
00130     static const int SNSTEPS_TRANS = 2;
00131     static const int SNSTEPS_RC = 17;
00132     static const int SNSTEPS_DRIFT = 21;
00133     static const int SNSTEPS_IMPACT = 250;
00134     static const int SNSTEPS_PHI = 250;
00135
00136     static float LUTValues[SNSTEPS_TRANS][SNSTEPS_RC][SNSTEPS_DRIFT]
00137                 [SNSTEPS_IMPACT][SNSTEPS_PHI];
00138
00139     float stepSizeTrans = 37; // 286 -> 323 or 310->350
00140     static constexpr float sSTEP_RC = 5;
00141     static constexpr float sSTEP_PHI = 90. / (SNSTEPS_PHI - 1);
00142     static constexpr float sSTEP_IMPACT = (PAD_DIAG / 2) / (SNSTEPS_IMPACT - 1);
00143     static constexpr float sSTEP_DRIFT = 1000. / (SNSTEPS_DRIFT - 1);
00144
00145     void Load();
00146
00147 //-----Data Members-----//
00148
00149 TFile *pFile_LUT;
00150 TTree *pTree_LUT;

```

```

00151 // Branches
00152 float fweight;
00153 float fdt;
00154 float fRC;
00155 float fdrift;
00156 double fd;
00157 double fphi;
00158
00159 int DtwithB;
00160 int DtwithoutB;
00161
00162 };
00163
00164 } // namespace Reconstruction
00165
00166 #endif

```

## 9.77 Reconstruction/inc/Monitoring.h File Reference

Monitoring utilities and entry points for analysis workflows.

```

#include "GiveMe_Uploader.h"
#include "Uploader.h"
#include "DrawOuts.h"
#include "dEdx.h"

```

### Namespaces

- namespace **Reconstruction**  
*dE/dx estimation and particle identification declarations.*

### Functions

- void **Reconstruction::Monitoring ()**
- void **Reconstruction::Correction (const int &corrRC=1, const int &corrGain=1, const int &corrWF=1, const int &corrDrift=1, const int &saveSelectOnly=1)**
- void **Reconstruction::Settings (const std::string &testbeam, const std::string &multiScanName, const std::string &scanName, const std::string &scanspec, const std::string &runvarstr\_name, const int &uploader, const int &modules, const int &peaking\_time, const int &diffusion, const int &drift\_dist, const int &timbin)**
- void **Reconstruction::ClearVectors ()**
- void **Reconstruction::DefaultAnalysis ()**
- void **Reconstruction::DrawSingleScan (const int &methods=0)**
- void **Reconstruction::DrawMultipleScan (const int &methods=0)**
- void **Reconstruction::DrawCERN22Scan ()**

### Variables

- **Reconstruction::dEdx \* Reconstruction::p\_dEdx**
- **Reconstruction::DrawOuts \* Reconstruction::p\_DrawOuts**
- int **Reconstruction::prototype = 0**
- int **Reconstruction::CERN\_Escan = 1**
- int **Reconstruction::CERN\_drift = 0**
- int **Reconstruction::DESY\_zscan**
- int **Reconstruction::DESY\_yscan**
- int **Reconstruction::DESY\_phi = 0**
- int **Reconstruction::DESY\_theta = 0**
- int **Reconstruction::dedx = 1**
- int **Reconstruction::DO\_Control**
- int **Reconstruction::DO\_dEdx**
- int **Reconstruction::DO\_Comparison**

### 9.77.1 Detailed Description

Monitoring utilities and entry points for analysis workflows.

Declares functions used to run monitoring, produce diagnostic summaries (histograms, performance numbers) and control analysis settings.

A small executable `Apps/Analysis.cxx` exists which calls `Reconstruction::Monitoring()` as a top-level entry point to run analysis workflows; the implementation of Monitoring lives in `Reconstruction/src/Monitoring.cxx`.

The source file contains routines that assemble diagnostic histograms, run monitoring workflows and produce summaries for inspection. It is the primary entry point used by the `Analysis` executable to run automated monitoring and analysis sequences; configuration and selection setup live in the header-level declarations and other helper modules.

## 9.78 Monitoring.h

[Go to the documentation of this file.](#)

```

00001 #
00021 #ifndef MONITORING_H
00022 #define MONITORING_H
00023
00024 #include "GiveMe_Uploader.h"
00025 #include "Uploader.h"
00026
00027 #include "DrawOuts.h"
00028 #include "dEdx.h"
00029
00030 namespace Reconstruction {
00031
00032 extern Reconstruction::dEdx *p_dEdx;
00033 extern Reconstruction::DrawOuts *p_DrawOuts;
00034
00035 void Monitoring();
00036 void Correction(const int &corrRC = 1, const int &corrGain = 1,
00037                     const int &corrWF = 1, const int &corrDrift = 1,
00038                     const int &saveSelectOnly = 1);
00039 void Settings(const std::string &testbeam, const std::string &multiScanName,
00040                  const std::string &scanName, const std::string &scanspec,
00041                  const std::string &runvarstr_name, const int &uploader,
00042                  const int &modules, const int &peakning_time, const int &diffusion,
00043                  const int &drift_dist, const int &timbin);
00044 void ClearVectors();
00045 void DefaultAnalysis();
00046 void DrawSingleScan(const int &methods = 0);
00047 void DrawMultipleScan(const int &methods = 0);
00048 void DrawCERN22Scan();
00049
00050 // Run selection
00051 // Files to use
00052 extern int prototype;
00053 extern int CERN_Escan;
00054 extern int CERN_drift;
00055
00056 extern int DESY_zscan;
00057 extern int DESY_yscan;
00058 extern int DESY_phi;
00059 extern int DESY_theta;
00060
00061 // Computations
00062 extern int dedx;
00063
00064 // DrawOuts
00065 extern int DO_Control;
00066 extern int DO_dEdx;
00067 extern int DO_Comparison;
00068 } // namespace Reconstruction
00069
00070 #endif

```

## 9.79 Reconstruction/inc/Recon\_LinkDef.h File Reference

ROOT linkdef file for [Reconstruction](#) module classes.

```
#include "dEdx.h"
#include <vector>
```

### 9.79.1 Detailed Description

ROOT linkdef file for [Reconstruction](#) module classes.

Contains #pragma link directives used by rootcling to generate dictionaries for I/O and interactive ROOT usage.

## 9.80 Recon\_LinkDef.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #include "dEdx.h"
00009 #include <vector>
00010 #ifdef __ROOTCLING__
00011
00012 #pragma link off all globals;
00013 #pragma link off all classes;
00014 #pragma link off all functions;
00015
00016 // For dEdx output
00017 #pragma link C++ namespace Reconstruction;
00018
00019 #pragma link C++ namespace Reconstruction;
00020 #pragma link C++ class Reconstruction::RecoPad + ;
00021 #pragma link C++ class std::vector < Reconstruction::RecoPad *> + ;
00022 #pragma link C++ class Reconstruction::RecoCluster + ;
00023 #pragma link C++ class std::vector < Reconstruction::RecoCluster *> + ;
00024 #pragma link C++ class Reconstruction::RecoModule + ;
00025 #pragma link C++ class std::vector < Reconstruction::RecoModule *> + ;
00026 #pragma link C++ class Reconstruction::RecoEvent + ;
00027
00028 // For ERAM maps reading
00029 #pragma link C++ class EramInfo + ;
00030 #endif
```

## 9.81 Reconstruction/inc/ReconTools.h File Reference

Miscellaneous reconstruction helper functions and utilities.

```
#include <vector>
#include "TF1.h"
#include "TH1F.h"
#include "Misc.h"
#include "Util.h"
#include "Module.h"
#include "Pad.h"
#include "Track.h"
#include "GiveMe_Uploader.h"
#include "Selector.h"
#include "Uploader.h"
```

## Namespaces

- namespace **Reconstruction**  
*dE/dx estimation and particle identification declarations.*

## Functions

- bool **FourModulesInLine** (const std::vector< int > &vec)
- void **Reconstruction::WFCorrection** (const std::string &OutCorr)
- TF1 \* **corr\_func** (const std::string &EventFile, const std::string &Tag, const int &correctWF)
- void **Init\_selection** (const std::string &SelectionSet, **Selector** &aSelector, const std::string &Tag, **Uploader** \*pUploader, const int &NbrOfMod, const int &Data\_to\_Use)
- std::vector< std::vector< float > > **readCSV** (std::string filename)
- std::vector< int > **ComputeCutStage3\_Cut** (**Uploader** \*pUploader, const int &NbrOfMod, const int &Data\_to\_Use)
- bool **GetStage3Cut\_CSV** (const std::string &filename, const std::string &targetWord, int &value1, int &value2)
- void **SetStage3Cut\_CSV** (const std::string &filename, const std::string &targetWord, int value1, int value2)
- TF1 \* **BetheBloch** (const float &Emin, const float &Emax, const double &m, const std::string &particle)
- TF1 \* **BetheBlochBhabha** (const float &Pmin, const float &Pmax, const double &m, const std::string &particle)
- TF1 \* **BetheBlochExp** (const float &Pmin, const float &Pmax, const double &M, const std::string &particle)
- TF1 \* **Fit2Gauss** (TH1F \*h1F)
- TF1 \* **Fit2Gauss** (TH1F \*h1F, const float &x1min, const float &x1max, const float &x2min, const float &x2max)
- void **local\_params** (**Pad** \*pPad, const **Track** \*pTrack, float &d, float &dd, float &phi, float &trk\_len\_pad)
- float **trk\_len** (**Module** \*pModule, const **Track** \*pTrack)
- TGraph \* **hist\_to\_graph** (TH1 \*h1)
- TGraph \* **Swapped\_graph** (TH1 \*h1)
- void **DrawTH2** (const std::string &OutDir, TH2 \*h2)

### 9.81.1 Detailed Description

Miscellaneous reconstruction helper functions and utilities.

Small helpers and adapters used across reconstruction, plotting and fitting code (linspace equivalents, I/O helpers, Bethe-Bloch wrappers, etc.).

The implementation file **ReconTools.cxx** contains a collection of lightweight utilities relied upon by higher-level modules (correction functions, selection initialization, CSV helpers and physics wrappers).

### 9.81.2 Function Documentation

#### 9.81.2.1 **BetheBloch()**

```
TF1 * BetheBloch (
    const float & Emin,
    const float & Emax,
    const double & m,
    const std::string & particle)
```

### 9.81.2.2 BetheBlochBhabha()

```
TF1 * BetheBlochBhabha (
    const float & Pmin,
    const float & Pmax,
    const double & m,
    const std::string & particle)
```

### 9.81.2.3 BetheBlochExp()

```
TF1 * BetheBlochExp (
    const float & Pmin,
    const float & Pmax,
    const double & M,
    const std::string & particle)
```

### 9.81.2.4 ComputeCutStage3\_Cut()

```
std::vector< int > ComputeCutStage3_Cut (
    Uploader * pUploader,
    const int & NbrOfMod,
    const int & Data_to_Use)
```

### 9.81.2.5 corr\_func()

```
TF1 * corr_func (
    const std::string & EventFile,
    const std::string & Tag,
    const int & correctWF)
```

### 9.81.2.6 DrawTH2()

```
void DrawTH2 (
    const std::string & OutDir,
    TH2 * h2)
```

### 9.81.2.7 Fit2Gauss() [1/2]

```
TF1 * Fit2Gauss (
    TH1F * h1F)
```

### 9.81.2.8 Fit2Gauss() [2/2]

```
TF1 * Fit2Gauss (
    TH1F * h1F,
    const float & x1min,
    const float & x1max,
    const float & x2min,
    const float & x2max)
```

**9.81.2.9 FourModulesInLine()**

```
bool FourModulesInLine (
    const std::vector< int > & vec)
```

**9.81.2.10 GetStage3Cut\_CSV()**

```
bool GetStage3Cut_CSV (
    const std::string & filename,
    const std::string & targetWord,
    int & value1,
    int & value2)
```

**9.81.2.11 hist\_to\_graph()**

```
TGraph * hist_to_graph (
    TH1 * h1)
```

**9.81.2.12 Init\_selection()**

```
void Init_selection (
    const std::string & SelectionSet,
    Selector & aSelector,
    const std::string & Tag,
    Uploader * pUploader,
    const int & NbrOfMod,
    const int & Data_to_Use)
```

**9.81.2.13 local\_params()**

```
void local_params (
    Pad * pPad,
    const Track * pTrack,
    float & d,
    float & dd,
    float & phi,
    float & trk_len_pad)
```

**9.81.2.14 readCSV()**

```
std::vector< std::vector< float > > readCSV (
    std::string filename)
```

**9.81.2.15 SetStage3Cut\_CSV()**

```
void SetStage3Cut_CSV (
    const std::string & filename,
    const std::string & targetWord,
    int value1,
    int value2)
```

### 9.81.2.16 Swapped\_graph()

```
TGraph * Swapped_graph (
    TH1 * h1)
```

### 9.81.2.17 trk\_len()

```
float trk_len (
    Module * pModule,
    const Track * pTrack)
```

## 9.82 ReconTools.h

[Go to the documentation of this file.](#)

```
00001 #
00013 #ifndef Reconstruction_TOOLS_H
00014 #define Reconstruction_TOOLS_H
00015
00016 #include <vector>
00017
00018 #include "TF1.h"
00019 #include "TH1F.h"
00020
00021 #include "Misc.h"
00022 #include "Util.h"
00023
00024 #include "Module.h"
00025 #include "Pad.h"
00026 #include "Track.h"
00027
00028 #include "GiveMe_Uploader.h"
00029 #include "Selector.h"
00030 #include "Uploader.h"
00031
00032 /* Python's numpy equivalent of linspace
00033 * -----
00034 */
00035
00036 bool FourModulesInLine(const std::vector<int> &vec);
00037
00038 namespace Reconstruction {
00039 void WFCorrection(const std::string &OutCorr);
00040 }
00041
00042 /* Call the correction function of WF method
00043 * -----
00044 TF1 *corr_func(const std::string &EventFile, const std::string &Tag,
00045                 const int &correctWF);
00046
00047 /* Initialize selection stages
00048 * -----
00049 void Init_selection(const std::string &SelectionSet, Selector &aSelector,
00050                      const std::string &Tag, Uploader *pUploader,
00051                      const int &NbrOfMod, const int &Data_to_Use);
00052
00053 /* FILE HANDLING */
00054 // Read CSV file
00055 std::vector<std::vector<float>> readCSV(std::string filename);
00056
00057 // Determine the time cut
00058 std::vector<int> ComputeCutStage3_Cut(Uploader *pUploader, const int &NbrOfMod,
00059                                         const int &Data_to_Use);
00060
00061 // Function to search for a word in a CSV file
00062 bool GetStage3Cut_CSV(const std::string &filename,
00063                        const std::string &targetWord, int &value1, int &value2);
00064
00065 // Function to update a CSV file
00066 void SetStage3Cut_CSV(const std::string &filename,
00067                        const std::string &targetWord, int value1, int value2);
00068
00069 /* GENERAL PHYSICS */
00070 // Bethe-Bloch function
```

```

00071 TF1 *BetheBloch(const float &Emin, const float &Emax, const double &m,
00072           const std::string &particle);
00073
00074 // Bethe-Bloch relativistic for positrons with Bhabha scattering
00075 TF1 *BetheBlochBhabha(const float &Pmin, const float &Pmax, const double &m,
00076           const std::string &particle);
00077
00078 // Bethe-Bloch with experimental parametrisation
00079 TF1 *BetheBlochExp(const float &Pmin, const float &Pmax, const double &M,
00080           const std::string &particle);
00081
00082 /* GENERAL MATH*/
00083
00084 // Fit with 2 gaussians
00085 TF1 *Fit2Gauss(TH1F *h1F);
00086 TF1 *Fit2Gauss(TH1F *h1F, const float &x1min, const float &x1max,
00087           const float &x2min, const float &x2max);
00088
00089 /* SPECIFIC MATH*/
00090 // impact parameter d (in m) & track angle phi (in degrees) computed locally at
00091 // the level of the pad
00092 void local_params(Pad *pPad, const Track *pTrack, float &d, float &dd,
00093           float &phi, float &trk_len_pad);
00094
00095 // Track length
00096 float trk_len(Module *pModule, const Track *pTrack);
00097
00098 // From TH1 get TGraph
00099 TGraph *hist_to_graph(TH1 *h1);
00100
00101 // From TH1 get swapped TGraph (transpose x and y)
00102 TGraph *Swapped_graph(TH1 *h1);
00103
00104 /* ROOT */
00105
00106 // Draw TH2
00107 void DrawTH2(const std::string &OutDir, TH2 *h2);
00108 #endif

```

## 9.83 Reconstruction/inc/Selector.h File Reference

Selection utilities to filter events, modules and clusters.

```
#include "Misc.h"
#include "Sample.h"
```

### Classes

- class **Selector**

*Encapsulates event/module/cluster selection logic.*

### Macros

- #define **Selector\_H**

#### 9.83.1 Detailed Description

Selection utilities to filter events, modules and clusters.

The **Selector** class encapsulates selection stages and criteria and provides methods to apply selections to samples and events.

Implementation in **Selector.cxx** contains the selection-stage definitions and concrete logic for known selection presets (DESY21, CERN22, etc.). Changing selection logic can affect downstream analyses; keep tests updated when criteria change.

## 9.83.2 Macro Definition Documentation

### 9.83.2.1 Selector\_H

```
#define Selector_H
```

## 9.84 Selector.h

[Go to the documentation of this file.](#)

```
00001
00014 #ifndef SELECTOR_H
00015 #define SELECTOR_H
00016 #ifndef Selector_H
00017 #define Selector_H
00018
00019 #include "Misc.h"
00020
00021 #include "Sample.h"
00022
00024
00028 class Selector {
00029 public:
00031     Selector(const std::string DefSelection);
00033     Selector();
00035     virtual ~Selector();
00036
00038     void Reset_Selection();
00039
00041     void Add_Selection(const std::string &SelectionName);
00042
00044     int NberOfSelections();
00045
00047     std::string Get_SelectionName(const int &iItem);
00048
00050     void Tell_Selection();
00051
00052     void ApplySelection(Sample &aSample, const int &ModuleNber);
00053
00055     void Apply_ASelection(Sample &aSample, const int &ModuleNber,
00056                             const int &iItem);
00057
00059     void Reset_StatCounters();
00060     void PrintStat();
00061
00062     double Get_Cut_StageFinal_NCluster_Low();
00063     void Set_Cut_StageFinal_NCluster_Low(double Cut_StageFinal_NCluster_Low);
00064
00065     double Get_Cut_Stage5_Npads_Hig();
00066     void Set_Cut_Stage5_Npads_Hig(double Cut_Stage5_Npads_Hig);
00067
00068     double Get_Cut_Stage2_EventBased();
00069     void Set_Cut_Stage2_EventBased(double Cut_Stage2_EventBased);
00070
00071     double Get_Cut_Stage3_TLow();
00072     double Get_Cut_Stage3_THigh();
00073     void Set_Cut_Stage3_TLow(double Cut_Stage3_TLow);
00074     void Set_Cut_Stage3_THigh(double Cut_Stage3_THigh);
00075
00076     double Get_Cut_Stage6_Amax_Low();
00077     void Set_Cut_Stage6_Amax_Low(double Cut_Stage6_Amax_Low);
00078     double Get_Cut_Stage6_Amax_Hig();
00079     void Set_Cut_Stage6_Amax_Hig(double Cut_Stage6_Amax_Hig);
00080
00081     double Get_Cut_Stage4_APM_Low();
00082     void Set_Cut_Stage4_APM_Low(double Cut_Stage4_APM_Low);
00083     double Get_Cut_Stage4_APM_High();
00084     void Set_Cut_Stage4_APM_High(double Cut_Stage4_APM_High);
00085
00086     private:
00087     // Stage 1: Remove clusters in first and last columns
00088     void Stage1_Def();
00089     void Stage1(Sample &aSample, const int &ModuleNber);
00090     void Stage1(Event *pEvent, const int &ModuleNber);
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
```

```

00102 // Stage 2: Remove clusters out of time (Event Based)
00103 void Stage2_Def();
00104 void Stage2(Sample &aSample, const int &ModuleNber);
00105 void Stage2(Event *pEvent, const int &ModuleNber);
00106
00107 // Stage 3: Remove clusters out of time
00108 void Stage3_Def();
00109 void Stage3(Sample &aSample, const int &ModuleNber);
00110 void Stage3(Event *pEvent, const int &ModuleNber);
00111
00112 // Stage 4: Remove Events with an average pad multiplicity too high or too low
00113 void Stage4_Def();
00114 void Stage4(Sample &aSample, const int &ModuleNber);
00115 void Stage4(Event *pEvent, const int &ModuleNber);
00116
00117 // Stage 5: Remove clusters with too many pads
00118 void Stage5_Def();
00119 void Stage5(Sample &aSample, const int &ModuleNber);
00120 void Stage5(Event *pEvent, const int &ModuleNber);
00121
00122 // Stage 6: Remove Clusters with a too high APad
00123 void Stage6_Def();
00124 void Stage6(Sample &aSample, const int &ModuleNber);
00125 void Stage6(Event *pEvent, const int &ModuleNber);
00126
00127 // Stage Final: Reject Events with too few clusters
00128 void StageFinal_Def();
00129 void StageFinal(Sample &aSample, const int &ModuleNber);
00130 void StageFinal(Event *pEvent, const int &ModuleNber);
00131
00132 // Defaults
00133 void Set_DESY21_Event(); // 1+5+11
00134 void Set_DESY21theta_Event();
00135 void Set_T2_CERN22_Event(); // 1+11+120+15+5+20
00136 void Set_TMC_CERN22_Event(); // 1+5
00137
00138 //
00139 void SetStat_Before(Sample &aSample, const int &ModuleNber, const int &iItem);
00140 void SetStat_After(Sample &aSample, const int &ModuleNber, const int &iItem);
00141
00142 void SetStat_Before(Event *pEvent, const int &iItem);
00143 void SetStat_After(Event *pEvent, const int &iItem);
00144
00145 std::vector<std::string> ListOfSelectionName;
00146 std::vector<int> ListOfNberOfEvents_Before;
00147 std::vector<int> ListOfNberOfEvents_After;
00148 std::vector<int> ListOfNberOfModules_Before;
00149 std::vector<int> ListOfNberOfModules_After;
00150 std::vector<int> ListOfNberOfClusters_Before;
00151 std::vector<int> ListOfNberOfClusters_After;
00152 std::vector<int> ListOfNberOfPads_Before;
00153 std::vector<int> ListOfNberOfPads_After;
00154
00155 //
00156 void Set_Cuts();
00157
00158 double m_Cut_StageFinal_NCluster_Low;
00159
00160 double m_Cut_Stage5_Npads_Hig;
00161
00162 double m_Cut_Stage2_EventBased;
00163
00164 double m_Cut_Stage3_TLow;
00165 double m_Cut_Stage3_THigh;
00166
00167 double m_Cut_Stage6_Amax_Low;
00168 double m_Cut_Stage6_Amax_Hig;
00169
00170 double m_Cut_Stage4_APM_Low;
00171 double m_Cut_Stage4_APM_High;
00172 };
00173
00174 #endif

```

## 9.85 Reconstruction/inc/Variables.h File Reference

Central place to declare reconstruction-wide variables and helpers.

```
#include "LUTs.h"
#include "Misc.h"
```

```
#include "Uploader.h"
```

## Namespaces

- namespace [Reconstruction](#)  
*dE/dx estimation and particle identification declarations.*

## Variables

- [Uploader \\* Reconstruction::p\\_uploader](#)
- [Reconstruction::LUT \\* Reconstruction::p\\_lut](#)
- const std::string [Reconstruction::dEdxPath](#) = "../"
- const std::string [Reconstruction::dataPath](#) = [dEdxPath](#) + "Data/"
- std::string [Reconstruction::dataScanPath](#) = ""
- const std::string [Reconstruction::drawoutPath](#) = [dEdxPath](#) + "OUT\_Reconstruction\_GP/"
- std::string [Reconstruction::drawoutMultiScanPath](#) = ""
- std::string [Reconstruction::drawoutScanPath](#) = ""
- std::string [Reconstruction::drawoutRunPath](#) = ""
- std::string [Reconstruction::drawout\\_file](#) = ""
- std::string [Reconstruction::rootout\\_file](#) = ""
- std::string [Reconstruction::log\\_file](#) = ""
- std::string [Reconstruction::comment](#)
- std::string [Reconstruction::selectionSet](#)
- std::string [Reconstruction::inputDir](#)
- int [Reconstruction::drawWhichMethods](#) = 0
- std::string [Reconstruction::corrFuncPath](#) = ""
- std::vector< std::string > [Reconstruction::vcorrFuncPaths](#)
- std::vector< float > [Reconstruction::vScanVals](#)
- std::vector< std::string > [Reconstruction::vScanLabels](#)
- std::vector< std::string > [Reconstruction::v\\_scanspec](#)
- std::vector< std::string > [Reconstruction::v\\_comments](#)
- std::vector< std::string > [Reconstruction::vTags](#)
- std::vector< std::string > [Reconstruction::v\\_datafiles](#)
- std::vector< std::string > [Reconstruction::v\\_rootout\\_files](#)
- std::string [Reconstruction::tag](#)
- std::string [Reconstruction::testbeam](#)
- std::string [Reconstruction::multiScanName](#)
- std::string [Reconstruction::scanName](#)
- std::string [Reconstruction::runvarstr](#)
- int [Reconstruction::scanIndex](#) = 0
- int [Reconstruction::intUploader](#)
- int [Reconstruction::moduleCase](#)
- int [Reconstruction::PT](#)
- int [Reconstruction::Dt](#)
- int [Reconstruction::Dtb](#)
- int [Reconstruction::DtnoB](#)
- int [Reconstruction::driftDist](#)
- int [Reconstruction::TB](#)
- int [Reconstruction::fcorrectGain](#)
- int [Reconstruction::fcorrectRC](#)
- int [Reconstruction::fcorrectWF](#)
- int [Reconstruction::fcorrectDrift](#)
- int [Reconstruction::fsaveSelectOnly](#)

### 9.85.1 Detailed Description

Central place to declare reconstruction-wide variables and helpers.

This header exposes shared data structures, constants and light helper utilities used by drawing, monitoring and fitting code. Definitions and initialisation (if any) should live in [Variables.cxx](#).

The implementation file [Variables.cxx](#) defines module-global configuration strings, pointers and run parameters used across reconstruction and plotting code. Keep definitions minimal: only shared state should be declared here and initialised in the source file.

## 9.86 Variables.h

[Go to the documentation of this file.](#)

```

00001 #ifndef Reconstruction_VARIABLES_H
00002 #define Reconstruction_VARIABLES_H
00003
00004 #include "LUTs.h"
00005 #include "Misc.h"
00006 #include "Uploader.h"
00007
00008 namespace Reconstruction {
00009
00010 extern Uploader *p_uploader;
00011
00026
00027 extern Reconstruction::LUT *p_lut;
00028
00029 // Output file variables
00030 extern const std::string dEdxPath;
00031 extern const std::string dataPath;
00032 extern std::string dataScanPath;
00033 extern const std::string drawoutPath;
00034 extern std::string drawoutMultiScanPath;
00035 extern std::string drawoutScanPath;
00036 extern std::string drawoutRunPath;
00037 extern std::string drawout_file;
00038 extern std::string rootout_file;
00039 extern std::string log_file;
00040 extern std::string comment;
00041 extern std::string selectionSet;
00042 extern std::string inputDir;
00043 extern int drawWhichMethods; // 0: both methods | 1: only WF | 2: only XP
00044 extern std::string corrFuncPath;
00045 extern std::vector<std::string> vcorrFuncPaths;
00046
00047 // Run variables
00048 extern std::vector<float> vScanVals;
00049 extern std::vector<std::string> vScanLabels;
00050 extern std::vector<std::string> v_scanspec;
00051 extern std::vector<std::string> v_comments;
00052 extern std::vector<std::string> vTags;
00053 extern std::vector<std::string> v_datafiles;
00054 extern std::vector<std::string> v_rootout_files;
00055 extern std::string tag;
00056 extern std::string testbeam;
00057 extern std::string multiScanName;
00058 extern std::string scanName;
00059 extern std::string runvarstr;
00060 extern int scanIndex;
00061 extern int intUploader;
00062 extern int moduleCase; // DESY21:0 | CERN22:-1
00063 extern int PT; // Peaking time (ns)
00064 extern int Dt; // Transverse diffusion coefficient ( $\mu\text{m}/\sqrt{\text{cm}})$ )
00065 extern int DtB; // Transverse diffusion coefficient with B( $\mu\text{m}/\sqrt{\text{cm}})$ )
00066 extern int DtnoB; // Transverse diffusion coefficient without B( $\mu\text{m}/\sqrt{\text{cm}})$ )
00067 extern int driftDist; // drift distance (mm)
00068 extern int TB; // timebin length (ns)
00069 extern int fcorrectGain; // 0: no correction | 1: correct pad per pad | 2:
00070 // correct ERAM per ERAM
00071 extern int fcorrectRC;
00072 extern int fcorrectWF;
00073 extern int fcorrectDrift;
00074 extern int fsaveSelectOnly;
00075 } // namespace Reconstruction
00076
00077 #endif

```

## 9.87 Reconstruction/src/CombinedFit.cxx File Reference

```
#include "CombinedFit.h"
#include "ReconTools.h"
#include "SetStyle.h"
```

### Classes

- struct [GlobalChi2\\_4](#)

### Macros

- `#define COMBINEDFIT_H`

### Variables

- int [npar](#) = 6
- int [iparpositron](#) [] = {0, 1, 2, 3, 4, 5}
- int [iparmuon](#) [] = {0, 1, 2, 3, 4, 6}
- int [iparpion](#) [] = {0, 1, 2, 3, 4, 7}
- int [iparproton](#) [] = {0, 1, 2, 3, 4, 8}

### 9.87.1 Macro Definition Documentation

#### 9.87.1.1 COMBINEDFIT\_H

```
#define COMBINEDFIT_H
```

### 9.87.2 Variable Documentation

#### 9.87.2.1 iparmuon

```
int iparmuon[] = {0, 1, 2, 3, 4, 6}
```

#### 9.87.2.2 iparpion

```
int iparpion[] = {0, 1, 2, 3, 4, 7}
```

#### 9.87.2.3 iparpositron

```
int iparpositron[] = {0, 1, 2, 3, 4, 5}
```

### 9.87.2.4 iparproton

```
int iparproton[ ] = {0, 1, 2, 3, 4, 8}
```

### 9.87.2.5 npar

```
int npar = 6
```

## 9.88 Reconstruction/src/Control.hxx File Reference

```
#include "Control.h"
#include "ReconTools.h"
#include "Util.h"
#include "Displays.h"
#include "Selector.h"
```

### Functions

- void **Control** (const std::string &OutDir, std::string const &Tag, std::string const &Comment, std::string const &EventFile, std::string const &SelectionSet, **Uploader** \*pUploader, int const &NbrOfMod, int const &Data\_to\_Use, int const &PT, int const &TB, const std::string &particle)

### 9.88.1 Function Documentation

#### 9.88.1.1 Control()

```
void Control (
    const std::string & OutDir,
    std::string const & Tag,
    std::string const & Comment,
    std::string const & EventFile,
    std::string const & SelectionSet,
    Uploader * pUploader,
    int const & NbrOfMod,
    int const & Data_to_Use,
    int const & PT,
    int const & TB,
    const std::string & particle)
```

## 9.89 Reconstruction/src/dEdx.cxx File Reference

```
#include "dEdx.h"
#include "LUTs.h"
#include "Misc_Functions.h"
#include "ReconTools.h"
#include "SignalTools.h"
#include "Variables.h"
#include "Util.h"
#include <cmath>
#include <fstream>
#include <numeric>
#include <random>
#include <string>
#include "ClusterFitter.h"
#include "Displays.h"
#include "GiveMe_Uploader.h"
#include "PRFParameters.h"
#include "Selector.h"
#include "TrackFitter.h"
#include "TrackRecon.h"
#include <chrono>
```

### Functions

- [ClassImp \(Reconstruction::RecoPad\)](#)
- [ClassImp \(Reconstruction::RecoCluster\)](#)
- [ClassImp \(Reconstruction::RecoModule\)](#)
- [ClassImp \(Reconstruction::RecoEvent\)](#)

### 9.89.1 Function Documentation

#### 9.89.1.1 ClassImp() [1/4]

```
ClassImp (
    Reconstruction::RecoCluster )
```

#### 9.89.1.2 ClassImp() [2/4]

```
ClassImp (
    Reconstruction::RecoEvent )
```

#### 9.89.1.3 ClassImp() [3/4]

```
ClassImp (
    Reconstruction::RecoModule )
```

### 9.89.1.4 ClassImp() [4/4]

```
ClassImp (
    Reconstruction::RecoPad )
```

## 9.90 Reconstruction/src/Displays.cxx File Reference

```
#include "Displays.h"
#include "ParabolaFunction.h"
#include "ParabolaFunctionNG.h"
#include "SetStyle.h"
#include "SignalTools.h"
#include "Util.h"
#include <typeinfo>
#include "TAxis.h"
#include "TCanvas.h"
#include "TF1.h"
#include "TFile.h"
#include "TGraph.h"
#include "TGraphErrors.h"
#include "TLatex.h"
#include "TLegend.h"
#include "TLegendEntry.h"
#include "TLine.h"
#include "TMultiGraph.h"
#include "TPaveStats.h"
#include "TProfile.h"
#include "TROOT.h"
#include "TStyle.h"
```

### Functions

- void [DrawOut\\_EventDisplay](#) ([Event](#) \*pEvent, const int &ModuleNber, const std::string &OUTDIR, const std::string &TAG)
   
*Output an event display for a given event.*
- void [DrawOut\\_EventDisplay](#) ([Module](#) \*pModule, const std::string &OUTDIR, const std::string &TAG, const std::string &type, const double &parabola, const double &slope, const double &intercept)
   
*Output an event display for a module.*
- [TH2D \\* GiveMe\\_EvtDisplay](#) ([Event](#) \*pEvent, const int &ModuleNber, const std::string &TAG)
   
*Get an event display TH2D for an event.*
- [TH2D \\* GiveMe\\_EvtDisplay](#) ([Module](#) \*pModule, const std::string &TAG)
   
*Get an event display TH2D for a module.*
- [TH2D \\* GiveMe\\_TimeDisplay](#) ([Module](#) \*pModule, const std::string &TAG)
- void [DrawOut\\_WaveFormDisplay](#) ([Pad](#) \*pPad, const std::string &OUTDIR, const std::string &TAG)
- [TH1F \\* GiveMe\\_WaveFormDisplay](#) ([Pad](#) \*pPad, const std::string &TAG)
- void [DrawOut\\_ClusterWFDisplay](#) ([Cluster](#) \*pCluster, const std::string &OUTDIR, const std::string &TAG)
- void [DrawOut\\_ClusterWFDisplay](#) ([Cluster](#) \*pCluster, const std::string &OUTDIR, const std::string &TAG, const int &Option, const int &PT, const int &TB)
- void [DrawOut\\_GWF](#) ([Event](#) \*pEvent, const int &ModuleNber, const std::string &OUTDIR, const std::string &TAG, const int &PT, const int &TB, const float &phi\_rad)
- void [NewClusterDisplay](#) ([Event](#) \*pEvent, const std::string &OUTDIR, const std::string &TAG, const int &PT, const int &TB)
- void [NewClusterDisplayMinimal](#) ([Event](#) \*pEvent, const std::string &OUTDIR, const std::string &TAG)

## 9.90.1 Function Documentation

### 9.90.1.1 DrawOut\_ClusterWFDisplay() [1/2]

```
void DrawOut_ClusterWFDisplay (
    Cluster * pCluster,
    const std::string & OUTDIR,
    const std::string & TAG)
```

### 9.90.1.2 DrawOut\_ClusterWFDisplay() [2/2]

```
void DrawOut_ClusterWFDisplay (
    Cluster * pCluster,
    const std::string & OUTDIR,
    const std::string & TAG,
    const int & Option,
    const int & PT,
    const int & TB)
```

### 9.90.1.3 DrawOut\_EventDisplay() [1/2]

```
void DrawOut_EventDisplay (
    Event * pEvent,
    const int & ModuleNber,
    const std::string & OUTDIR,
    const std::string & TAG)
```

Output an event display for a given event.

#### Parameters

<i>pEvent</i>	pointer to the <a href="#">Event</a>
<i>ModuleNber</i>	module index
<i>OUTDIR</i>	output directory path
<i>TAG</i>	tagging string used in output filenames

### 9.90.1.4 DrawOut\_EventDisplay() [2/2]

```
void DrawOut_EventDisplay (
    Module * pModule,
    const std::string & OUTDIR,
    const std::string & TAG,
    const std::string & type,
    const double & parabola,
    const double & slope,
    const double & intercept)
```

Output an event display for a module.

**9.90.1.5 DrawOut\_GWF()**

```
void DrawOut_GWF (
    Event * pEvent,
    const int & ModuleNber,
    const std::string & OUTDIR,
    const std::string & TAG,
    const int & PT,
    const int & TB,
    const float & phi_rad)
```

**9.90.1.6 DrawOut\_WaveFormDisplay()**

```
void DrawOut_WaveFormDisplay (
    Pad * pPad,
    const std::string & OUTDIR,
    const std::string & TAG)
```

**9.90.1.7 GiveMe\_EvtDisplay() [1/2]**

```
TH2D * GiveMe_EvtDisplay (
    Event * pEvent,
    const int & ModuleNber,
    const std::string & TAG)
```

Get an event display TH2D for an event.

**9.90.1.8 GiveMe\_EvtDisplay() [2/2]**

```
TH2D * GiveMe_EvtDisplay (
    Module * pModule,
    const std::string & TAG)
```

Get an event display TH2D for a module.

**9.90.1.9 GiveMe\_TimeDisplay()**

```
TH2D * GiveMe_TimeDisplay (
    Module * pModule,
    const std::string & TAG)
```

**9.90.1.10 GiveMe\_WaveFormDisplay()**

```
TH1F * GiveMe_WaveFormDisplay (
    Pad * pPad,
    const std::string & TAG)
```

### 9.90.1.11 NewClusterDisplay()

```
void NewClusterDisplay (
    Event * pEvent,
    const std::string & OUTDIR,
    const std::string & TAG,
    const int & PT,
    const int & TB)
```

### 9.90.1.12 NewClusterDisplayMinimal()

```
void NewClusterDisplayMinimal (
    Event * pEvent,
    const std::string & OUTDIR,
    const std::string & TAG)
```

## 9.91 Reconstruction/src/DrawOuts.cxx File Reference

```
#include "DrawOuts.h"
#include "CombinedFit.h"
#include "Misc_Functions.h"
#include "ReconTools.h"
#include "SignalTools.h"
#include "dEdx.h"
#include "Util.h"
#include <cmath>
#include <numeric>
#include "TF1.h"
#include "TFile.h"
#include "TLatex.h"
#include "TLegend.h"
#include "TLegendEntry.h"
#include "TLine.h"
#include "TPaveStats.h"
#include "TROOT.h"
```

### Functions

- void [DrawOut\\_corrections \(\)](#)

## 9.91.1 Function Documentation

### 9.91.1.1 DrawOut\_corrections()

```
void DrawOut_corrections ()
```

## 9.92 Reconstruction/src/DrawOuts\_old.cxx File Reference

```
#include "DrawOuts_old.h"
#include "CombinedFit.h"
#include "Misc_Functions.h"
#include "ReconTools.h"
#include "dEdx.h"
#include "Util.h"
#include <cmath>
#include <numeric>
#include "TF1.h"
#include "TFile.h"
#include "TLatex.h"
#include "TLegend.h"
#include "TLegendEntry.h"
#include "TLine.h"
#include "TPaveStats.h"
#include "TROOT.h"
```

### Functions

- void [DrawOut\\_Versions](#) (const std::string &inputDir, const std::string &Method, const std::string &Comment1, const std::string &Comment2)
- void [DrawOut\\_Separation](#) (const std::string &inputDir, const std::string &Comment)
- void [DrawOut\\_Separation\\_Reduced](#) (const std::string &inputDir, const std::string &Comment, std::string Energy)
- void [DrawOut\\_Scans](#) (const std::string &inputDir, const std::string &Comment, const std::string &WFversion)
- void [DrawOut\\_Zscan\\_PT](#) (const std::string &inputDir, const std::string &Comment)
- void [DrawOut\\_Phiscan\\_Z](#) (const std::string &inputDir, const std::string &Comment, const std::string &WFversion)
- void [DrawOut\\_Escan](#) (const std::string &inputDir, const std::string &Comment)
- void [DrawOut\\_Systematics](#) (const std::string &inputDir, const std::string &Comment, const std::string &scan)

### 9.92.1 Function Documentation

#### 9.92.1.1 DrawOut\_Escan()

```
void DrawOut_Escan (
    const std::string & inputDir,
    const std::string & Comment)
```

#### 9.92.1.2 DrawOut\_Phiscan\_Z()

```
void DrawOut_Phiscan_Z (
    const std::string & inputDir,
    const std::string & Comment,
    const std::string & WFversion)
```

### 9.92.1.3 DrawOut\_Scans()

```
void DrawOut_Scans (
    const std::string & inputDir,
    const std::string & Comment,
    const std::string & WFversion)
```

### 9.92.1.4 DrawOut\_Separation()

```
void DrawOut_Separation (
    const std::string & inputDir,
    const std::string & Comment)
```

### 9.92.1.5 DrawOut\_Separation\_Reduced()

```
void DrawOut_Separation_Reduced (
    const std::string & inputDir,
    const std::string & Comment,
    std::string Energy)
```

### 9.92.1.6 DrawOut\_Systematics()

```
void DrawOut_Systematics (
    const std::string & inputDir,
    const std::string & Comment,
    const std::string & scan)
```

### 9.92.1.7 DrawOut\_Versions()

```
void DrawOut_Versions (
    const std::string & inputDir,
    const std::string & Method,
    const std::string & Comment1,
    const std::string & Comment2)
```

### 9.92.1.8 DrawOut\_Zscan\_PT()

```
void DrawOut_Zscan_PT (
    const std::string & inputDir,
    const std::string & Comment)
```

## 9.93 Reconstruction/src/LUTs.cxx File Reference

```
#include "LUTs.h"
#include <algorithm>
#include <iomanip>
#include <iostream>
#include <iterator>
#include <sstream>
```

## Functions

- ClassImp (EramInfo) Reconstruction

### 9.93.1 Function Documentation

#### 9.93.1.1 ClassImp()

```
ClassImp (  
          EramInfo )
```

## 9.94 Reconstruction/src/Monitoring.cxx File Reference

```
#include "Misc.h"  
#include "Util.h"  
#include "Monitoring.h"  
#include "ReconTools.h"  
#include "Variables.h"  
#include "Control.h"  
#include "Selector.h"
```

## Namespaces

- namespace Reconstruction  
*dE/dx estimation and particle identification declarations.*

## Variables

- int Reconstruction::DESY\_drift = 0
- int Reconstruction::DESY\_row = 0
- int Reconstruction::correction\_wf = 0
- int Reconstruction::Draw\_Control = 0
- int Reconstruction::Draw\_dEdx = 0
- int Reconstruction::Draw\_Comparison = 0
- int Reconstruction::Draw\_Corrections = 0
- int Reconstruction::Draw\_DESY21SingleScan = 0
- int Reconstruction::Draw\_DESY21MultScan = 0
- int Reconstruction::Draw\_CERN22Scan = 0
- int Reconstruction::DtwithBhere = 286
- int Reconstruction::DtwithoutBhere = 323

## 9.95 Reconstruction/src/ReconTools.cxx File Reference

```
#include "ReconTools.h"
#include "Displays.h"
#include "LUTs.h"
#include "ClusterFitter.h"
#include "PRFParameters.h"
#include "TrackFitter.h"
#include "TrackRecon.h"
#include "TPaveText.h"
#include "Variables.h"
```

### Functions

- bool [FourModulesInLine](#) (const std::vector< int > &vec)
- TF1 \* [corr\\_func](#) (const std::string &EventFile, const std::string &Tag, const int &correctWF)
- void [Init\\_selection](#) (const std::string &SelectionSet, [Selector](#) &aSelector, const std::string &Tag, [Uploader](#) \*pUploader, const int &NbrOfMod, const int &Data\_to\_Use)
- std::vector< std::vector< float > > [readCSV](#) (std::string filename)
- std::vector< int > [ComputeCutStage3\\_Cut](#) ([Uploader](#) \*pUploader, const int &NbrOfMod, const int &Data\_to\_Use)
- bool [GetStage3Cut\\_CSV](#) (const std::string &filename, const std::string &targetWord, int &value1, int &value2)
- void [SetStage3Cut\\_CSV](#) (const std::string &filename, const std::string &targetWord, int value1, int value2)
- TF1 \* [Fit2Gauss](#) (TH1F \*h1F)
- TF1 \* [Fit2Gauss](#) (TH1F \*h1F, const float &x1min, const float &x1max, const float &x2min, const float &x2max)
- TF1 \* [BetheBloch](#) (const float &Pmin, const float &Pmax, const double &M, const std::string &particle)
- TF1 \* [BetheBlochBhabha](#) (const float &Pmin, const float &Pmax, const double &m, const std::string &particle)
- TF1 \* [BetheBlochExp](#) (const float &Pmin, const float &Pmax, const double &M, const std::string &particle)
- void [local\\_params](#) (Pad \*pPad, const Track \*pTrack, float &d, float &dd, float &phi, float &trk\_len\_pad)
- float [trk\\_len](#) (Module \*pModule, const Track \*pTrack)
- void [DrawTH2](#) (const std::string &OutDir, TH2 \*h2)
- TGraph \* [hist\\_to\\_graph](#) (TH1 \*h1)
- TGraph \* [Swapped\\_graph](#) (TH1 \*h1)

### 9.95.1 Function Documentation

#### 9.95.1.1 BetheBloch()

```
TF1 * BetheBloch (
    const float & Pmin,
    const float & Pmax,
    const double & M,
    const std::string & particle)
```

#### 9.95.1.2 BetheBlochBhabha()

```
TF1 * BetheBlochBhabha (
    const float & Pmin,
    const float & Pmax,
    const double & m,
    const std::string & particle)
```

**9.95.1.3 BetheBlochExp()**

```
TF1 * BetheBlochExp (
    const float & Pmin,
    const float & Pmax,
    const double & M,
    const std::string & particle)
```

**9.95.1.4 ComputeCutStage3\_Cut()**

```
std::vector< int > ComputeCutStage3_Cut (
    Uploader * pUploader,
    const int & NbrOfMod,
    const int & Data_to_Use)
```

**9.95.1.5 corr\_func()**

```
TF1 * corr_func (
    const std::string & EventFile,
    const std::string & Tag,
    const int & correctWF)
```

**9.95.1.6 DrawTH2()**

```
void DrawTH2 (
    const std::string & OutDir,
    TH2 * h2)
```

**9.95.1.7 Fit2Gauss() [1/2]**

```
TF1 * Fit2Gauss (
    TH1F * h1F)
```

**9.95.1.8 Fit2Gauss() [2/2]**

```
TF1 * Fit2Gauss (
    TH1F * h1F,
    const float & x1min,
    const float & x1max,
    const float & x2min,
    const float & x2max)
```

**9.95.1.9 FourModulesInLine()**

```
bool FourModulesInLine (
    const std::vector< int > & vec)
```

### 9.95.1.10 GetStage3Cut\_CSV()

```
bool GetStage3Cut_CSV (
    const std::string & filename,
    const std::string & targetWord,
    int & value1,
    int & value2)
```

### 9.95.1.11 hist\_to\_graph()

```
TGraph * hist_to_graph (
    TH1 * h1)
```

### 9.95.1.12 Init\_selection()

```
void Init_selection (
    const std::string & SelectionSet,
    Selector & aSelector,
    const std::string & Tag,
    Uploader * pUploader,
    const int & NbrOfMod,
    const int & Data_to_Use)
```

### 9.95.1.13 local\_params()

```
void local_params (
    Pad * pPad,
    const Track * pTrack,
    float & d,
    float & dd,
    float & phi,
    float & trk_len_pad)
```

### 9.95.1.14 readCSV()

```
std::vector< std::vector< float > > readCSV (
    std::string filename)
```

### 9.95.1.15 SetStage3Cut\_CSV()

```
void SetStage3Cut_CSV (
    const std::string & filename,
    const std::string & targetWord,
    int value1,
    int value2)
```

**9.95.1.16 Swapped\_graph()**

```
TGraph * Swapped_graph (
    TH1 * h1)
```

**9.95.1.17 trk\_len()**

```
float trk_len (
    Module * pModule,
    const Track * pTrack)
```

## 9.96 Reconstruction/src/Selector.cxx File Reference

```
#include "Selector.h"
#include "Event.h"
#include "Pad.h"
#include "TAxis.h"
#include "TCanvas.h"
#include "TF1.h"
#include "TGraph.h"
#include "TGraphErrors.h"
#include "TLegend.h"
#include "TLine.h"
#include "TMultiGraph.h"
#include "TPaveStats.h"
#include "TProfile.h"
#include "TStyle.h"
```

## 9.97 Reconstruction/src/Variables.cxx File Reference

```
#include "Variables.h"
```

### Namespaces

- namespace [Reconstruction](#)  
*dE/dx estimation and particle identification declarations.*

### Variables

- [Uploader \\* Reconstruction::pUploader](#)

## 9.98 Uploader/inc/GiveMe\_Uploader.h File Reference

Factory that constructs a concrete [Uploader](#) instance.

```
#include "Misc.h"
#include "Uploader.h"
```

### Functions

- [Uploader \\* GiveMe\\_Uploader](#) (const int &intUploader, const std::string &SampleFile)

#### 9.98.1 Detailed Description

Factory that constructs a concrete [Uploader](#) instance.

Helper that returns a configured [Uploader](#) pointer based on an integer selector. The caller receives ownership of the returned pointer.

#### 9.98.2 Function Documentation

##### 9.98.2.1 GiveMe\_Uploader()

```
Uploader * GiveMe_Uploader (
    const int & intUploader,
    const std::string & SampleFile)
```

## 9.99 GiveMe\_Uploader.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #include "Misc.h"
00009 #include "Uploader.h"
00010
00011 Uploader *GiveMe_Uploader(const int &intUploader,
00012                               const std::string &SampleFile);
```

## 9.100 Uploader/inc/Uploader.h File Reference

Abstract base class for data uploaders.

```
#include "Event.h"
#include "Misc.h"
#include "Model_Charge1D.h"
#include "Model_Electronics.h"
#include "Model_ReadOutGeometry.h"
#include "TFile.h"
#include "TTree.h"
```

## Classes

- class [Uploader](#)

### 9.100.1 Detailed Description

Abstract base class for data uploaders.

Declares the [Uploader](#) interface used to read input samples and produce [Event](#) objects for the reconstruction pipeline. Concrete uploaders implement format-specific logic.

## 9.101 Uploader.h

[Go to the documentation of this file.](#)

```

00001 #
00009 #ifndef Uploader_H
00010 #define Uploader_H
00011
00012 #include "Event.h"
00013 #include "Misc.h"
00014
00015 #include "Model_Charge1D.h"
00016 #include "Model_Electronics.h"
00017 #include "Model_ReadOutGeometry.h"
00018
00019 #include "TFile.h"
00020 #include "TTree.h"
00021
00023 class Uploader {
00024 public:
00026     Uploader(const std::string &SampleFile,
00027             Model_ReadOutGeometry *pModel_ReadOutGeometry,
00028             Model_Electronics *pModel_Electronics,
00029             Model_Charge1D *pModel_Charge1D);
00030     virtual ~Uploader();
00031
00032     std::string Get_SampleFile();
00033
00034     int Get_NberOfEvent();
00035
00036     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00037                                     const int &Data_to_Use,
00038                                     const int &CloseWF = 1) = 0;
00039     int Get_PRF_exist();
00040     double Get_Norm();
00041     double Get_a2();
00042     double Get_a4();
00043     double Get_b2();
00044     double Get_b4();
00045
00046     Model_ReadOutGeometry *Get_Model_ReadOutGeometry();
00047     Model_Electronics *Get_Model_Electronics();
00048     Model_Charge1D *Get_Model_Charge1D();
00049
00050 //-----Data Members-----//
00051 protected:
00052     std::string m_SampleFile;
00053     int m_NberOfEvent;
00054
00055     TFile *p_TFile;
00056
00057     TTree *p_TTree;
00058
00059     Model_ReadOutGeometry *p_Model_ReadOutGeometry;
00060     Model_Electronics *p_Model_Electronics;
00061     Model_Charge1D *p_Model_Charge1D;
00062
00063     short int m_PRF_exist;
00064     double m_Norm;
00065     double m_a2;
00066     double m_a4;
00067     double m_b2;
00068     double m_b4;
00069 };
00070
00071 #endif

```

## 9.102 Uploader/inc/Uploader\_ERAM01.h File Reference

[Uploader](#) for ERAM01-format data files.

```
#include "Misc.h"
#include "Uploader.h"
```

### Classes

- class [Uploader\\_ERAM01](#)

#### 9.102.1 Detailed Description

[Uploader](#) for ERAM01-format data files.

Declares [Uploader\\_ERAM01](#) which converts ERAM01-format data into [Event](#) objects for use by the reconstruction pipeline.

## 9.103 Uploader\_ERAM01.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #ifndef Uploader_ERAM01_H
00009 #define Uploader_ERAM01_H
00010
00011 #include "Misc.h"
00012 #include "Uploader.h"
00013
00015 class Uploader_ERAM01 : public Uploader {
00016 public:
00018     Uploader_ERAM01(const std::string &SampleFile,
00019                     Model_ReadOutGeometry *pModel_ReadOutGeometry,
00020                     Model_Electronics *pModel_Electronics,
00021                     Model_ChargeID *pModel_ChargeID);
00022     virtual ~Uploader_ERAM01();
00023
00024     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00025                                 const int &Data_to_Use, const int &CloseWF);
00026
00027     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00028                                 const int &Data_to_Use);
00029
00030 //-----Data Members-----//
00031 private:
00032 };
00033
00034 #endif
```

## 9.104 Uploader/inc/Uploader\_MockUp\_V1.h File Reference

Mock-up uploader (version 1) declaration for testing.

```
#include "Misc.h"
#include "Uploader.h"
```

**Classes**

- class [Uploader\\_MockUp\\_V1](#)

**9.104.1 Detailed Description**

Mock-up uploader (version 1) declaration for testing.

Reads simplified datasets and fills [Event](#) structures for development and testing of reconstruction code.

**9.105 Uploader\_MockUp\_V1.h**

[Go to the documentation of this file.](#)

```
00001 #
00008 #ifndef Uploader_MOCKUP_V1_H
00009 #define Uploader_MOCKUP_V1_H
00010
00011 #include "Misc.h"
00012 #include "Uploader.h"
00013
00015 class Uploader_MockUp_V1 : public Uploader {
00016 public:
00018     Uploader_MockUp_V1(const std::string &SampleFile,
00019                         Model_ReadOutGeometry *pModel_ReadOutGeometry,
00020                         Model_Electronics *pModel_Electronics,
00021                         Model_ChargeID *pModel_ChargeID);
00022     virtual ~Uploader_MockUp_V1();
00023
00024     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00025                                 const int &Data_to_Use, const int &CloseWF);
00026
00027     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00028                                 const int &Data_to_Use);
00029
00030 //-----Data Members-----
00031 private:
00032 };
00033
00034 #endif
```

**9.106 Uploader/inc/Uploader\_MockUp\_V2.h File Reference**

Mock-up uploader (version 2) declaration for richer test datasets.

```
#include "Misc.h"
#include "Uploader.h"
```

**Classes**

- class [Uploader\\_MockUp\\_V2](#)

**9.106.1 Detailed Description**

Mock-up uploader (version 2) declaration for richer test datasets.

Version 2 exposes additional branches and fields commonly found in newer test datasets and provides branch pointers used to populate vectors and waveform arrays.

## 9.107 Uploader\_MockUp\_V2.h

[Go to the documentation of this file.](#)

```

00001 #
00009 #ifndef Uploader__V2_H
00010 #define Uploader__V2_H
00011
00012 #include "Misc.h"
00013 #include "Uploader.h"
00014
00016 class Uploader_MockUp_V2 : public Uploader {
00017 public:
00019     Uploader_MockUp_V2(const std::string &SampleFile,
00020                         Model_ReadOutGeometry *pModel_ReadOutGeometry,
00021                         Model_Electronics *pModel_Electronics,
00022                         Model_Charge1D *pModel_Charge1D);
00023     virtual ~Uploader_MockUp_V2();
00024
00025     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00026                                 const int &Data_to_Use, const int &CloseWF);
00027
00028     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00029                                 const int &Data_to_Use);
00030
00031     void SetBeforeMerging(const int &i_SetBeforeMerging);
00032     void SetSwap_XY(const int &i_Swap_XY);
00033
00034 //-----Data Members-----/
00035 public:
00036     Int_t fCurrent;
00037
00038     Int_t ev;
00039     Int_t track;
00040     ULong64_t date; // Added since code update contains the date of the events !
00041     Bool_t beforeMerging; // Added since code update tells if this track has been
00042                           // saved before or after merging algorithm
00043     Double_t dEdx;
00044     Double_t angle_yz;
00045     Double_t angle_xy;
00046     Int_t rob_clusters;
00047     Double_t quality;
00048     Double_t mom;
00049     Double_t sina;
00050     Double_t offset;
00051     Int_t max_mult;
00052     Double_t mean_mult;
00053     std::vector<int> *multiplicity;
00054     std::vector<int> *charge;
00055     std::vector<double> *residual;
00056     std::vector<double> *residual_corr;
00057     std::vector<std::vector<double>> *dx;
00058     std::vector<std::vector<double>> *qfrac;
00059     std::vector<std::vector<int>> *time;
00060     std::vector<double> *clust_pos;
00061     std::vector<double> *clust_pos_err;
00062     std::vector<double> *track_pos;
00063     std::vector<int> *module;
00064     std::vector<std::vector<int>> *pad_charge;
00065     std::vector<std::vector<int>> *pad_time;
00066     std::vector<std::vector<int>> *wf_width;
00067     std::vector<std::vector<int>> *wf_fwhm;
00068     std::vector<std::vector<int>> *pad_x;
00069     std::vector<std::vector<int>> *pad_y;
00070     std::vector<std::vector<std::vector<int>>> *pad_wf_q;
00071
00072 // List of branches
00073     TBranch *b_ev;
00074     TBranch *b_track;
00075     TBranch *b_date;
00076     TBranch *b_beforeMerging;
00077     TBranch *b_dEdx;
00078     TBranch *b_angle_yz;
00079     TBranch *b_angle_xy;
00080     TBranch *b_rob_clusters;
00081     TBranch *b_quality;
00082     TBranch *b_mom;
00083     TBranch *b_sina;
00084     TBranch *b_offset;
00085     TBranch *b_max_mult;
00086     TBranch *b_mean_mult;
00087     TBranch *b_multiplicity;
00088     TBranch *b_charge;
00089     TBranch *b_residual;
00090     TBranch *b_residual_corr;
00091     TBranch *b_dx;

```

```

00092 TBranch *b_qfrac;
00093 TBranch *b_time;
00094 TBranch *b_clust_pos;
00095 TBranch *b_clust_pos_err;
00096 TBranch *b_track_pos;
00097 TBranch *b_module;
00098 TBranch *b_pad_charge;
00099 TBranch *b_pad_time;
00100 TBranch *b_wf_width;
00101 TBranch *b_wf_fwhm;
00102 TBranch *b_pad_x;
00103 TBranch *b_pad_y;
00104 TBranch *b_pad_wf_q;
00105
00106 void Init();
00107
00108 private:
00109 int m_BeforeMerging;
00110 int m_Swap_XY;
00111 int m_Nx;
00112 int m_Ny;
00113 };
00114
00115 #endif

```

## 9.108 Uploader/inc/Uploader\_Protoype.h File Reference

Prototype-format uploader declaration.

```
#include "Misc.h"
#include "Uploader.h"
```

### Classes

- class [Uploader\\_Protoype](#)

#### 9.108.1 Detailed Description

Prototype-format uploader declaration.

Declares [Uploader\\_Protoype](#) which reads prototype-format datasets and converts them into [Event](#) objects for reconstruction.

## 9.109 Uploader\_Protoype.h

[Go to the documentation of this file.](#)

```

00001 #
00008 #ifndef Uploader_Protoype_H
00009 #define Uploader_Protoype_H
00010
00011 #include "Misc.h"
00012 #include "Uploader.h"
00013
00015 class Uploader_Protoype : public Uploader {
00016 public:
00018     Uploader_Protoype(const std::string &SampleFile,
00019                         Model_ReadOutGeometry *pModel_ReadOutGeometry,
00020                         Model_Electronics *pModel_Electronics,
00021                         Model_ChargeID *pModel_ChargeID);
00022     virtual ~Uploader_Protoype();
00023
00024     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,

```

```

00025     const int &Data_to_Use, const int &CloseWF);
00026
00027     virtual Event *GiveMe_Event(const int &iEvent, const int &ModuleNber_Input,
00028                                     const int &Data_to_Use);
00029
00030     //-----Data Members-----//
00031 private:
00032 };
00033
00034 #endif

```

## 9.110 Uploader/src/GiveMe\_Uploader.cxx File Reference

```

#include "GiveMe_Uploader.h"
#include "Model_Charge1D.h"
#include "Model_Electronics.h"
#include "Model_ReadOutGeometry.h"
#include "Uploader_ERAM01.h"
#include "Uploader_MockUp_V1.h"
#include "Uploader_MockUp_V2.h"
#include "Uploader_ProtoType.h"

```

### Functions

- [Uploader \\* GiveMe\\_Uploader \(const int &intUploader, const std::string &SampleFile\)](#)

### 9.110.1 Function Documentation

#### 9.110.1.1 GiveMe\_Uploader()

```

Uploader * GiveMe_Uploader (
    const int & intUploader,
    const std::string & SampleFile)

```

## 9.111 Uploader/src/Uploader.cxx File Reference

```

#include "Uploader.h"
#include "TF1.h"

```

## 9.112 Uploader/src/Uploader\_ERAM01.cxx File Reference

```
#include "Uploader_ERAM01.h"
```

## 9.113 Uploader/src/Uploader\_MockUp\_V1.cxx File Reference

```
#include "Uploader_MockUp_V1.h"
```

## 9.114 Uploader/src/Uploader\_MockUp\_V2.cxx File Reference

```
#include "Uploader_MockUp_V2.h"
```

## 9.115 Uploader/src/Uploader\_Prototype.cxx File Reference

```
#include "Uploader_Prototype.h"
```

## 9.116 Utilities/inc/FuncFromTGraph.h File Reference

Create function wrappers from ROOT TGraph objects.

### 9.116.1 Detailed Description

Create function wrappers from ROOT TGraph objects.

Utility to produce analytic-like function wrappers from TGraph/TGraphErrors objects. Useful for interpolation and for creating TF1-compatible callables from discrete points.

## 9.117 FuncFromTGraph.h

[Go to the documentation of this file.](#)

```
00001 #ifndef FuncFromTGraph_H
00002 #define FuncFromTGraph_H
00003
00004 #
00012 #ifndef FuncFromTGraph_H
00013 #define FuncFromTGraph_H
00014 #include "Misc.h"
00015
00016 #include "TGraph.h"
00017
00018 //
00020 class FuncFromTGraph {
00021 public:
00023     FuncFromTGraph(TGraph *pTGraph);
00024     virtual ~FuncFromTGraph();
00025
00026     double Get_Y(const double &vX);
00027
00028     void Dump();
00029
00030     double operator()(double *x, double *par);
00031
00032 private:
00033     TGraph *p_TGraph;
00034     double m_StepX;
00035     double m_X_Lowest;
00036     double m_X_Highest;
00037 };
00038
00039 #endif
```

## 9.118 Utilities/inc/GaussFunction.h File Reference

Gaussian function wrapper declaration.

```
#include "Misc.h"
```

### Classes

- class [GaussFunction](#)

### 9.118.1 Detailed Description

Gaussian function wrapper declaration.

Declares [GaussFunction](#) which provides getters/setters and a TF1-compatible operator() to evaluate a Gaussian with configurable normalization, mean and sigma.

## 9.119 GaussFunction.h

[Go to the documentation of this file.](#)

```
00001 #
00009 #ifndef GaussFunction_H
00010 #define GaussFunction_H
00011
00012 #include "Misc.h"
00013
00015 class GaussFunction {
00016 public:
00018     GaussFunction();
00019     virtual ~GaussFunction();
00020
00021     double Get_Norm();
00022     double Get_Mean();
00023     double Get_Sigma();
00024     void Set_Norm(const double &Norm);
00025     void Set_Mean(const double &Mean);
00026     void Set_Sigma(const double &Sigma);
00027
00028     double operator()(double *x, double *par);
00029
00030 private:
00031     double m_Norm;
00032     double m_Mean;
00033     double m_Sigma;
00034 };
00035
00036 #endif
```

## 9.120 Utilities/inc/Misc.h File Reference

Small general-purpose utilities and common definitions.

```
#include <algorithm>
#include <cctype>
#include <cmath>
#include <cstdlib>
#include <cstring>
```

```
#include <fstream>
#include <iomanip>
#include <iostream>
#include <numeric>
#include <sstream>
#include <stdio.h>
#include <string>
#include <vector>
```

### 9.120.1 Detailed Description

Small general-purpose utilities and common definitions.

Contains lightweight helpers, constants and convenience utilities used across the project. Keep this header minimal to reduce coupling.

## 9.121 Misc.h

[Go to the documentation of this file.](#)

```
00001 #
00008 #ifndef Misc_H
00009 #define Misc_H
00010
00011 #include <algorithm>
00012 #include <cctype>
00013 #include <cmath>
00014 #include <cstdlib>
00015 #include <cstring>
00016 #include <fstream>
00017 #include <iomanip>
00018 #include <iostream>
00019 #include <numeric>
00020 #include <sstream>
00021 #include <stdio.h>
00022 #include <string>
00023 #include <vector>
00024
00025 #endif
```

## 9.122 Utilities/inc/ParabolaFunction.h File Reference

Parabola helper function declaration.

```
#include "Misc.h"
```

### Classes

- class [ParabolaFunction](#)

### 9.122.1 Detailed Description

Parabola helper function declaration.

Declares [ParabolaFunction](#) which implements a simple parabola wrapper used for fitting and curve approximation. A TF1-callable operator() is provided for integration with ROOT if needed.

## 9.123 ParabolaFunction.h

[Go to the documentation of this file.](#)

```
00001 #
00009 #ifndef ParabolaFunction_H
0010 #define ParabolaFunction_H
0011
0012 #include "Misc.h"
0013
0014 //
0015 // Fit parabola -std::fabs(A0) * std::pow(X0-vX,2) + Y0 );
0016
0018 class ParabolaFunction {
0019 public:
0021     ParabolaFunction();
0022     virtual ~ParabolaFunction();
0023
0024     double Get_A0();
0025     double Get_X0();
0026     double Get_Y0();
0027     void Set_A0(const double &A0);
0028     void Set_X0(const double &X0);
0029     void Set_Y0(const double &Y0);
0030
0031     double operator()(double *x, double *par);
0032
0033 private:
0034     double m_A0;
0035     double m_X0;
0036     double m_Y0;
0037 };
0038
0039 #endif
```

## 9.124 Utilities/inc/ParabolaFunctionNG.h File Reference

Non-symmetric parabola helper function declaration.

```
#include "Misc.h"
```

### Classes

- class [ParabolaFunctionNG](#)

### 9.124.1 Detailed Description

Non-symmetric parabola helper function declaration.

Declares [ParabolaFunctionNG](#) which implements a parabola with different curvature on either side of the vertex. Useful for asymmetric fits and TF1-compatible callbacks.

## 9.125 ParabolaFunctionNG.h

[Go to the documentation of this file.](#)

```

00001 #
00009 #ifndef ParabolaFunctionNG_H
0010 #define ParabolaFunctionNG_H
0011
0012 #include "Misc.h"
0013
0014 //
0015 // Fit parabola -std::fabs(A0) * std::pow(X0-vX,2) + Y0 ;
0016
0018 class ParabolaFunctionNG {
0019 public:
0021     ParabolaFunctionNG();
0022     virtual ~ParabolaFunctionNG();
0023
0024     double Get_A0P();
0025     double Get_A0M();
0026     double Get_X0();
0027     double Get_Y0();
0028     void Set_A0P(const double &A0P);
0029     void Set_A0M(const double &A0M);
0030     void Set_X0(const double &X0);
0031     void Set_Y0(const double &Y0);
0032
0033     double operator()(double *x, double *par);
0034
0035 private:
0036     double m_A0P;
0037     double m_A0M;
0038     double m_X0;
0039     double m_Y0;
0040 };
0041
0042 #endif

```

## 9.126 Utilities/inc/SignalTools.h File Reference

Signal-processing helper declarations (baseline, pulse finding, ...).

```
#include "TH1F.h"
```

### Functions

- TH1F \* **ETF** (std::string name, const float &tmin, float const &tmax, const float &t\_0, const int &nbins, const int &iC, const float &PT, const float &TB)

### 9.126.1 Detailed Description

Signal-processing helper declarations (baseline, pulse finding, ...).

Provides utility declarations used by reconstruction and fitting code for basic signal processing tasks such as building ETF pulse shapes.

## 9.126.2 Function Documentation

### 9.126.2.1 ETF()

```
TH1F * ETF (
    std::string name,
    const float & tmin,
    float const & tmax,
    const float & t_0,
    const int & nbins,
    const int & iC,
    const float & PT,
    const float & TB)
```

The corresponding source file [SignalTools.cxx](#) implements ETF and other electronics-response helpers used to simulate and normalise pulse shapes. These utilities are used by reconstruction and testing code that needs a reproducible electronics impulse response (ETF) and related helpers.

## 9.127 SignalTools.h

[Go to the documentation of this file.](#)

```
00001
00009 #
00010 #
00017
00018 #ifndef MISC_SIGNALTOOLS_H
00019 #define MISC_SIGNALTOOLS_H
00020
00021 #include "TH1F.h"
00022
00023 // Electronics pulse response function
00024 TH1F *ETF(std::string name, const float &tmin, float const &tmax,
00025           const float &t_0, const int &nbins, const int &iC, const float &PT,
00026           const float &TB);
00027
00028 #endif
```

## 9.128 Utilities/src/FuncFromTGraph.cxx File Reference

```
#include "FuncFromTGraph.h"
```

## 9.129 Utilities/src/GaussFunction.cxx File Reference

```
#include "GaussFunction.h"
```

## 9.130 Utilities/src/ParabolaFunction.cxx File Reference

```
#include "ParabolaFunction.h"
```

## 9.131 Utilities/src/ParabolaFunctionNG.cxx File Reference

```
#include "ParabolaFunctionNG.h"
```

## 9.132 Utilities/src/SignalTools.cxx File Reference

```
#include "SignalTools.h"
#include "Misc_Functions.h"
```

### Functions

- TH1F \* **ETF** (std::string name, const float &tmin, float const &tmax, const float &t\_0, const int &nbins, const int &iC, const float &PT, const float &TB)

### 9.132.1 Function Documentation

#### 9.132.1.1 ETF()

```
TH1F * ETF (
    std::string name,
    const float & tmin,
    float const & tmax,
    const float & t_0,
    const int & nbins,
    const int & iC,
    const float & PT,
    const float & TB)
```

The corresponding source file [SignalTools.cxx](#) implements ETF and other electronics-response helpers used to simulate and normalise pulse shapes. These utilities are used by reconstruction and testing code that needs a reproducible electronics impulse response (ETF) and related helpers.



# Index

~Cluster  
    Cluster, 34  
~ClusterFitter\_Diagonal  
    ClusterFitter\_Diagonal, 42  
~ClusterFitter\_Horizontal  
    ClusterFitter\_Horizontal, 44  
~DrawOuts  
    Reconstruction::DrawOuts, 50  
~ERAMMaps  
    Reconstruction::ERAMMaps, 55  
~EramInfo  
    EramInfo, 53  
~Event  
    Event, 57  
~FitOutput  
    FitOutput, 61  
~GaussFunction  
    GaussFunction, 64  
~LUT  
    Reconstruction::LUT, 67  
~Model\_Charge0D  
    Model\_Charge0D, 69  
~Model\_Charge1D  
    Model\_Charge1D, 72  
~Model\_ChargeI  
    Model\_ChargeI, 74  
~Model\_Electronics  
    Model\_Electronics, 77  
~Model\_ReadOutGeometry  
    Model\_ReadOutGeometry, 79  
~Module  
    Module, 84  
~PRFParameters  
    PRFParameters, 98  
~Pad  
    Pad, 88  
~ParabolaFunction  
    ParabolaFunction, 95  
~ParabolaFunctionNG  
    ParabolaFunctionNG, 96  
~ROBoard  
    ROBoard, 113  
~RecoCluster  
    Reconstruction::RecoCluster, 100  
~RecoEvent  
    Reconstruction::RecoEvent, 103  
~RecoModule  
    Reconstruction::RecoModule, 106  
~RecoPad  
    Reconstruction::RecoPad, 109  
~Sample  
    Sample, 116  
~Selector  
    Selector, 121  
~SetOfTracks  
    SetOfTracks, 126  
~StaticClusterFitter\_Diagonal  
    StaticClusterFitter\_Diagonal, 131  
~StaticClusterFitter\_Horizontal  
    StaticClusterFitter\_Horizontal, 132  
~StaticTrackFitter  
    StaticTrackFitter, 133  
~Track  
    Track, 135  
~TrackFitter  
    TrackFitter, 139  
~Uploader  
    Uploader, 141  
~Uploader\_ERAM01  
    Uploader\_ERAM01, 145  
~Uploader\_MockUp\_V1  
    Uploader\_MockUp\_V1, 147  
~Uploader\_MockUp\_V2  
    Uploader\_MockUp\_V2, 151  
~Uploader\_Prototype  
    Uploader\_Prototype, 159  
~dEdx  
    Reconstruction::dEdx, 46  
ADCmax  
    Reconstruction::RecoPad, 109  
ADCmax\_base  
    Reconstruction::RecoCluster, 100  
    Reconstruction::RecoPad, 109  
Add\_Cluster  
    Module, 84  
    Track, 135  
Add\_Event  
    Sample, 117  
Add\_Module  
    Event, 57  
Add\_Pad  
    Cluster, 35  
    Module, 84  
    ROBoard, 113  
Add\_Selection  
    Selector, 121  
Add\_Track  
    SetOfTracks, 126

**ALead\_base**  
 Reconstruction::RecoCluster, 100  
**ALead\_GCorr**  
 Reconstruction::RecoCluster, 100  
**AmplitudeError**  
 AmplitudeError.cxx, 191  
 AmplitudeError.h, 173  
**AmplitudeError.cxx**  
 AmplitudeError, 191  
 AmplitudeError\_Leading, 191  
 AmplitudeError\_NextLeading, 192  
 AmplitudeError\_NextNextLeading, 192  
**AmplitudeError.h**  
 AmplitudeError, 173  
**AmplitudeError\_Leading**  
 AmplitudeError.cxx, 191  
**AmplitudeError\_NextLeading**  
 AmplitudeError.cxx, 192  
**AmplitudeError\_NextNextLeading**  
 AmplitudeError.cxx, 192  
**AmplitudeVSLength**  
 Reconstruction::DrawOuts, 50  
**Analysis.cxx**  
 main, 161  
**angle\_xy**  
 Uploader\_MockUp\_V2, 152  
**angle\_yz**  
 Uploader\_MockUp\_V2, 152  
**Apply\_ASelection**  
 Selector, 121  
**ApplySelection**  
 Selector, 121, 122  
**Apps Directory Reference**, 15  
**Apps/Analysis.cxx**, 161  
**Apps/DrawOut.cxx**, 161  
**avg\_pad\_mult**  
 Reconstruction::RecoEvent, 103  
 Reconstruction::RecoModule, 107  
  
**b\_angle\_xy**  
 Uploader\_MockUp\_V2, 152  
**b\_angle\_yz**  
 Uploader\_MockUp\_V2, 152  
**b\_beforeMerging**  
 Uploader\_MockUp\_V2, 152  
**b\_charge**  
 Uploader\_MockUp\_V2, 152  
**b\_clust\_pos**  
 Uploader\_MockUp\_V2, 152  
**b\_clust\_pos\_err**  
 Uploader\_MockUp\_V2, 152  
**b\_date**  
 Uploader\_MockUp\_V2, 152  
**b\_dEdx**  
 Uploader\_MockUp\_V2, 152  
**b\_dx**  
 Uploader\_MockUp\_V2, 152  
**b\_ev**  
 Uploader\_MockUp\_V2, 153  
  
**b\_max\_mult**  
 Uploader\_MockUp\_V2, 153  
**b\_mean\_mult**  
 Uploader\_MockUp\_V2, 153  
**b\_module**  
 Uploader\_MockUp\_V2, 153  
**b\_mom**  
 Uploader\_MockUp\_V2, 153  
**b\_multiplicity**  
 Uploader\_MockUp\_V2, 153  
**b\_offset**  
 Uploader\_MockUp\_V2, 153  
**b\_pad\_charge**  
 Uploader\_MockUp\_V2, 153  
**b\_pad\_time**  
 Uploader\_MockUp\_V2, 153  
**b\_pad\_wf\_q**  
 Uploader\_MockUp\_V2, 153  
**b\_pad\_x**  
 Uploader\_MockUp\_V2, 154  
**b\_pad\_y**  
 Uploader\_MockUp\_V2, 154  
**b\_qfrac**  
 Uploader\_MockUp\_V2, 154  
**b\_quality**  
 Uploader\_MockUp\_V2, 154  
**b\_residual**  
 Uploader\_MockUp\_V2, 154  
**b\_residual\_corr**  
 Uploader\_MockUp\_V2, 154  
**b\_rob\_clusters**  
 Uploader\_MockUp\_V2, 154  
**b\_sina**  
 Uploader\_MockUp\_V2, 154  
**b\_time**  
 Uploader\_MockUp\_V2, 154  
**b\_track**  
 Uploader\_MockUp\_V2, 154  
**b\_track\_pos**  
 Uploader\_MockUp\_V2, 155  
**b\_wf\_fwhm**  
 Uploader\_MockUp\_V2, 155  
**b\_wf\_width**  
 Uploader\_MockUp\_V2, 155  
**beforeMerging**  
 Uploader\_MockUp\_V2, 155  
**BetheBloch**  
 ReconTools.cxx, 242  
 ReconTools.h, 223  
**BetheBlochBhabha**  
 ReconTools.cxx, 242  
 ReconTools.h, 223  
**BetheBlochExp**  
 ReconTools.cxx, 242  
 ReconTools.h, 224  
  
**CERN22Scan**  
 Reconstruction::DrawOuts, 50  
**CERN22ScanDraw**

Reconstruction::DrawOuts, 50  
CERN22ScanFill  
    Reconstruction::DrawOuts, 50  
CERN\_drift  
    Reconstruction, 25  
CERN\_Escan  
    Reconstruction, 25  
charge  
    Reconstruction::RecoCluster, 101  
    Reconstruction::RecoPad, 110  
    Uploader\_MockUp\_V2, 155  
Chi2  
    ClusterFitter\_Diagonal, 42  
    ClusterFitter\_Horizontal, 44  
    Track, 135  
    TrackFitter, 139  
Chi2\_Diagonal  
    Cluster, 35  
Chi2\_Horizontal  
    Cluster, 35  
ClassDef  
    Reconstruction::RecoCluster, 100  
    Reconstruction::RecoEvent, 103  
    Reconstruction::RecoModule, 106  
    Reconstruction::RecoPad, 109  
ClassImp  
    dEdx.cxx, 234  
    LUTs.cxx, 241  
Clear  
    Reconstruction::RecoEvent, 103  
Clear\_ADC  
    Pad, 89  
Clear\_Clusters  
    Module, 84  
Clear\_Modules  
    Event, 57  
ClearVectors  
    Reconstruction, 23  
clust\_pos  
    Uploader\_MockUp\_V2, 155  
clust\_pos\_err  
    Uploader\_MockUp\_V2, 155  
Cluster, 33  
    ~Cluster, 34  
    Add\_Pad, 35  
    Chi2\_Diagonal, 35  
    Chi2\_Horizontal, 35  
    Cluster, 34  
    DoClosure, 35  
    Eval\_Diagonal, 35  
    Eval\_Horizontal, 35  
    FitRes\_Diagonal\_Get\_NberOfTermsInChi2, 35  
    FitRes\_Diagonal\_Get\_Pad, 35  
    FitRes\_Diagonal\_Get\_Pull, 36  
    FitRes\_Diagonal\_Get\_Residual, 36  
    FitRes\_Horizontal\_Get\_NberOfTermsInChi2, 36  
    FitRes\_Horizontal\_Get\_Pad, 36  
    FitRes\_Horizontal\_Get\_Pull, 36  
    FitRes\_Horizontal\_Get\_Residual, 36  
Get\_Acluster, 36  
Get\_AMaxLeading, 36  
Get\_Chi2Min, 36  
Get\_EntryNber, 37  
Get\_EventNber, 37  
Get\_eXTrack, 37  
Get\_eYTrack, 37  
Get\_LeadingPad, 37  
Get\_ModuleNber, 37  
Get\_NberOfPads, 37  
Get\_NextLeadingPad, 37  
Get\_NextNextLeadingPad, 37  
Get\_Pad, 37  
Get\_TMaxLeading, 38  
Get\_XTrack, 38  
Get\_XTrack\_BeforeMinimisation, 38  
Get\_XWeight, 38  
Get\_YLeading, 38  
Get\_YTrack, 38  
Get\_YTrack\_BeforeMinimisation, 38  
Get\_YWeight, 38  
Invalidate, 38  
IsValid, 38  
m\_AngleRot, 41  
Set\_eXTrack, 39  
Set\_eYTrack, 39  
Set\_XTrack, 39  
Set\_YTrack, 39  
SetEval\_Diagonal, 39  
SetEval\_Horizontal, 39  
SetParameter\_Diagonal, 39  
SetParameter\_Horizontal, 39  
SetResults\_Diagonal, 39  
SetResults\_FailedFit\_Diagonal, 40  
SetResults\_FailedFit\_Horizontal, 40  
SetResults\_Horizontal, 40  
StatusFit, 40  
StatusFit\_Diagonal, 40  
StatusFit\_Horizontal, 40  
Validate, 40  
WriteOut, 40  
ClusterFit\_Diagonal  
    ClusterFitter.cxx, 169  
    ClusterFitter.h, 163  
ClusterFit\_Diagonal\_Cluster  
    ClusterFitter.cxx, 169  
    ClusterFitter.h, 163  
ClusterFit\_Diagonal\_Event  
    ClusterFitter.cxx, 170  
    ClusterFitter.h, 163  
ClusterFit\_Horizontal  
    ClusterFitter.cxx, 170  
    ClusterFitter.h, 163  
ClusterFit\_Horizontal\_Cluster  
    ClusterFitter.cxx, 170  
    ClusterFitter.h, 163  
ClusterFit\_Horizontal\_Event

ClusterFitter.cxx, 170  
 ClusterFitter.h, 163  
 ClusterFitter.hxx  
     ClusterFit\_Diagonal, 169  
     ClusterFit\_Diagonal\_Cluster, 169  
     ClusterFit\_Diagonal\_Event, 170  
     ClusterFit\_Horizontal, 170  
     ClusterFit\_Horizontal\_Cluster, 170  
     ClusterFit\_Horizontal\_Event, 170  
     ClusterFitter\_DiagonalFunction, 170  
     ClusterFitter\_HorizontalFunction, 170  
 ClusterFitter.h  
     ClusterFit\_Diagonal, 163  
     ClusterFit\_Diagonal\_Cluster, 163  
     ClusterFit\_Diagonal\_Event, 163  
     ClusterFit\_Horizontal, 163  
     ClusterFit\_Horizontal\_Cluster, 163  
     ClusterFit\_Horizontal\_Event, 163  
 ClusterFitter\_Diagonal, 41  
     ~ClusterFitter\_Diagonal, 42  
     Chi2, 42  
     ClusterFitter\_Diagonal, 41  
     DoMinimisation, 42  
     m\_FitterName, 43  
     p\_Cluster, 43  
     p\_TVirtualFitter, 43  
     Set\_Cluster, 42  
 ClusterFitter\_DiagonalFunction  
     ClusterFitter.cxx, 170  
 ClusterFitter\_Horizontal, 43  
     ~ClusterFitter\_Horizontal, 44  
     Chi2, 44  
     ClusterFitter\_Horizontal, 44  
     DoMinimisation, 44  
     m\_FitterName, 45  
     p\_Cluster, 45  
     p\_TVirtualFitter, 45  
     Set\_Cluster, 45  
 ClusterFitter\_HorizontalFunction  
     ClusterFitter.cxx, 170  
 combinedFit  
     CombinedFit.h, 202  
 CombinedFit.hxx  
     COMBINEDFIT\_H, 232  
     iparmuon, 232  
     iparpion, 232  
     iparpositron, 232  
     iparproton, 232  
     npar, 233  
 CombinedFit.h  
     combinedFit, 202  
 COMBINEDFIT\_H  
     CombinedFit.cxx, 232  
 comment  
     Reconstruction, 25  
 ComputeCutStage3\_Cut  
     ReconTools.cxx, 243  
     ReconTools.h, 224  
 ComputedEdxGP  
     Reconstruction::dEdx, 46  
 ComputedEdxGP1  
     Reconstruction::dEdx, 46  
 ComputedEdxWF  
     Reconstruction::dEdx, 46  
 ComputedEdxXP  
     Reconstruction::dEdx, 47  
 Control  
     Control.hxx, 233  
     Control.h, 203  
     Reconstruction::DrawOuts, 50  
 Control.hxx  
     Control, 233  
 Control.h  
     Control, 203  
 corr\_func  
     ReconTools.cxx, 243  
     ReconTools.h, 224  
 Correction  
     Reconstruction, 23  
 correction\_wf  
     Reconstruction, 25  
 corrFuncPath  
     Reconstruction, 25  
 d  
     Reconstruction::RecoPad, 110  
 dataPath  
     Reconstruction, 25  
 dataScanPath  
     Reconstruction, 25  
 date  
     Uploader\_MockUp\_V2, 155  
 dd  
     Reconstruction::RecoPad, 110  
 dEdx  
     Reconstruction::dEdx, 46  
     Uploader\_MockUp\_V2, 155  
 dedx  
     Reconstruction, 25  
 dEdx.hxx  
     ClassImp, 234  
 dEdxGP1  
     Reconstruction::RecoEvent, 103  
 dEdxGP2  
     Reconstruction::RecoEvent, 103  
 dEdxGP3  
     Reconstruction::RecoEvent, 103  
 dEdxGP4  
     Reconstruction::RecoEvent, 103  
 dEdxGP5  
     Reconstruction::RecoEvent, 104  
 dEdxPath  
     Reconstruction, 25  
 dEdxWF  
     Reconstruction::RecoCluster, 101  
     Reconstruction::RecoEvent, 104  
     Reconstruction::RecoModule, 107

dEdxWFnoTrunc  
    Reconstruction::RecoEvent, 104  
    Reconstruction::RecoModule, 107

dEdxXP  
    Reconstruction::RecoEvent, 104  
    Reconstruction::RecoModule, 107  
    Reconstruction::RecoPad, 110

dEdxXPnoTrunc  
    Reconstruction::RecoEvent, 104  
    Reconstruction::RecoModule, 107

DefaultAnalysis  
    Reconstruction, 24

Derive\_Response\_Base  
    Model\_Electronics, 77

DESY21MultiScan  
    Reconstruction::DrawOuts, 51

DESY21ScanDraw  
    Reconstruction::DrawOuts, 51

DESY21ScanFill  
    Reconstruction::DrawOuts, 51

DESY21SingleScan  
    Reconstruction::DrawOuts, 51

DESY\_drift  
    Reconstruction, 25

DESY\_phi  
    Reconstruction, 26

DESY\_row  
    Reconstruction, 26

DESY\_theta  
    Reconstruction, 26

DESY\_yscan  
    Reconstruction, 26

DESY\_zscan  
    Reconstruction, 26

DiscardedModule  
    Reconstruction::dEdx, 47

Displays.cxx  
    DrawOut\_ClusterWFDisplay, 236  
    DrawOut\_EventDisplay, 236  
    DrawOut\_GWF, 236  
    DrawOut\_WaveFormDisplay, 237  
    GiveMe\_EvtDisplay, 237  
    GiveMe\_TimeDisplay, 237  
    GiveMe\_WaveFormDisplay, 237  
    NewClusterDisplay, 237  
    NewClusterDisplayMinimal, 238

Displays.h  
    DrawOut\_ClusterWFDisplay, 208  
    DrawOut\_EventDisplay, 208  
    DrawOut\_GWF, 209  
    DrawOut\_WaveFormDisplay, 209  
    GiveMe\_EvtDisplay, 209  
    GiveMe\_TimeDisplay, 209  
    GiveMe\_WaveFormDisplay, 210  
    NewClusterDisplay, 210  
    NewClusterDisplayMinimal, 210

DO\_Comparison  
    Reconstruction, 26

DO\_Control  
    Reconstruction, 26

DO\_dEdx  
    Reconstruction, 26

DoClosure  
    Cluster, 35  
    Track, 135

DoMinimisation  
    ClusterFitter\_Diagonal, 42  
    ClusterFitter\_Horizontal, 44  
    TrackFitter, 139

Draw\_CERN22Scan  
    Reconstruction, 26

Draw\_Comparison  
    Reconstruction, 26

Draw\_Control  
    Reconstruction, 27

Draw\_Corrections  
    Reconstruction, 27

Draw\_dEdx  
    Reconstruction, 27

Draw\_DESY21MultScan  
    Reconstruction, 27

Draw\_DESY21SingleScan  
    Reconstruction, 27

DrawCERN22Scan  
    Reconstruction, 24

DrawMultipleScan  
    Reconstruction, 24

DrawOut.cxx  
    main, 162

DrawOut\_ClusterWFDisplay  
    Displays.cxx, 236  
    Displays.h, 208

DrawOut\_corrections  
    DrawOuts.cxx, 238

DrawOut\_Escan  
    DrawOuts\_old.cxx, 239  
    DrawOuts\_old.h, 214

DrawOut\_EventDisplay  
    Displays.cxx, 236  
    Displays.h, 208

drawout\_file  
    Reconstruction, 27

DrawOut\_GWF  
    Displays.cxx, 236  
    Displays.h, 209

DrawOut\_Methods  
    DrawOuts\_old.h, 214

DrawOut\_Phiscan  
    DrawOuts\_old.h, 214

DrawOut\_Phiscan\_Z  
    DrawOuts\_old.cxx, 239  
    DrawOuts\_old.h, 214

DrawOut\_Scans  
    DrawOuts\_old.cxx, 239  
    DrawOuts\_old.h, 214

DrawOut\_Separation

DrawOuts\_old.cxx, 240  
 DrawOuts\_old.h, 215  
**DrawOut\_Separation\_Reduced**  
 DrawOuts\_old.cxx, 240  
 DrawOuts\_old.h, 215  
**DrawOut\_Systematics**  
 DrawOuts\_old.cxx, 240  
 DrawOuts\_old.h, 215  
**DrawOut\_Thetascan**  
 DrawOuts\_old.h, 215  
**DrawOut\_Versions**  
 DrawOuts\_old.cxx, 240  
 DrawOuts\_old.h, 215  
**DrawOut\_WaveFormDisplay**  
 Displays.cxx, 237  
 Displays.h, 209  
**DrawOut\_Yscan**  
 DrawOuts\_old.h, 215  
**DrawOut\_Zscan**  
 DrawOuts\_old.h, 215  
**DrawOut\_Zscan\_PT**  
 DrawOuts\_old.cxx, 240  
 DrawOuts\_old.h, 216  
**drawoutMultiScanPath**  
 Reconstruction, 27  
**drawoutPath**  
 Reconstruction, 27  
**drawoutRunPath**  
 Reconstruction, 27  
**DrawOuts**  
 Reconstruction::DrawOuts, 48, 50  
**DrawOuts.cxx**  
 DrawOut\_corrections, 238  
**DrawOuts\_old.cxx**  
 DrawOut\_Escan, 239  
 DrawOut\_Phiscan\_Z, 239  
 DrawOut\_Scans, 239  
 DrawOut\_Separation, 240  
 DrawOut\_Separation\_Reduced, 240  
 DrawOut\_Systematics, 240  
 DrawOut\_Versions, 240  
 DrawOut\_Zscan\_PT, 240  
**DrawOuts\_old.h**  
 DrawOut\_Escan, 214  
 DrawOut\_Methods, 214  
 DrawOut\_Phiscan, 214  
 DrawOut\_Phiscan\_Z, 214  
 DrawOut\_Scans, 214  
 DrawOut\_Separation, 215  
 DrawOut\_Separation\_Reduced, 215  
 DrawOut\_Systematics, 215  
 DrawOut\_Thetascan, 215  
 DrawOut\_Versions, 215  
 DrawOut\_Yscan, 215  
 DrawOut\_Zscan, 215  
 DrawOut\_Zscan\_PT, 216  
**drawoutScanPath**  
 Reconstruction, 27

DrawSingleScan  
 Reconstruction, 24  
**DrawTH2**  
 ReconTools.cxx, 243  
 ReconTools.h, 224  
**drawWhichMethods**  
 Reconstruction, 28  
**driftDist**  
 Reconstruction, 28  
**driftDistance**  
 Reconstruction::RecoPad, 110  
**Dt**  
 Reconstruction, 28  
**DtB**  
 Reconstruction, 28  
**DtnoB**  
 Reconstruction, 28  
**DtwithBhere**  
 Reconstruction, 28  
**DtwithoutBhere**  
 Reconstruction, 28  
**DumpRec**  
 SetOfTracks, 126  
 Track, 135  
**dx**  
 Uploader\_MockUp\_V2, 155  
**dy**  
 Reconstruction::RecoPad, 110

**Endplate**  
 EramInfo, 53  
**EnergyLoss**  
 Reconstruction::DrawOuts, 51  
**EramInfo**, 52  
 ~EramInfo, 53  
 Endplate, 53  
 EramInfo, 53  
 Gain, 53  
 Id, 54  
 InbTPC, 54  
 IntTPC, 54  
 Position, 54  
 RC, 54  
 Resolution, 54  
 XX, 54  
 YY, 54  
**ERAMMaps**  
 Reconstruction::ERAMMaps, 55

**ETF**  
 SignalTools.cxx, 259  
 SignalTools.h, 258  
**ev**  
 Uploader\_MockUp\_V2, 156  
**Eval**  
 PRFParameters, 98  
**Eval\_Diagonal**  
 Cluster, 35  
**Eval\_Horizontal**  
 Cluster, 35

Event, 56  
~Event, 57  
Add\_Module, 57  
Clear\_Modules, 57  
Event, 57  
Get\_Cluster\_Copy, 57  
Get\_EntryNber, 57  
Get\_EventNber, 58  
Get\_Model\_Charge, 58  
Get\_Model\_Electronics, 58  
Get\_Model\_ReadOutGeometry, 58  
Get\_Module\_InArray, 58  
Get\_NberOfModule, 58  
Get\_ThisModule, 58  
Get\_ThisModuleConst, 58  
GiveMe\_AnUnfittedTrack\_ForThisModule, 58  
GiveMe\_Clusters\_ForThisModule, 58  
GiveMe\_CutInterceptXZ\_ForThisModule, 59  
GiveMe\_CutSlopeXY\_ForThisModule, 59  
GiveMe\_CutSlopeXYZ\_ForThisModule, 59  
GiveMe\_CutSlopeXZ\_ForThisModule, 59  
GiveMe\_Track\_ForThisModule, 59  
Invalidate, 59  
Invalidate\_ThisModule, 59  
IsValid, 59  
Replace\_Clusters\_ForThisModule, 59  
Set\_Track\_ForThisModule, 60  
SmallDump, 60  
Validate, 60  
Validate\_ThisModule, 60  
Validity\_ForThisModule, 60  
WriteOut, 60  
eventNbr  
    Reconstruction::RecoEvent, 104  
fChi2\_1  
    GlobalChi2\_4, 66  
fChi2\_2  
    GlobalChi2\_4, 66  
fChi2\_3  
    GlobalChi2\_4, 66  
fChi2\_4  
    GlobalChi2\_4, 66  
fcorrectDrift  
    Reconstruction, 28  
fcorrectGain  
    Reconstruction, 28  
fcorrectRC  
    Reconstruction, 28  
fcorrectWF  
    Reconstruction, 29  
fCurrent  
    Uploader\_MockUp\_V2, 156  
FileComparison  
    Reconstruction::DrawOuts, 52  
Fit2Gauss  
    ReconTools.cxx, 243  
    ReconTools.h, 224  
FitOutput, 60  
    ~FitOutput, 61  
    FitOutput, 61  
    m\_Description, 62  
    m\_MinnLL, 62  
    m\_NameOfTheModel, 63  
    m\_NberOfDataPoints, 63  
    m\_NberOfModelParameters, 63  
    operator=, 62  
    p\_CovMatrix, 63  
    p\_eparminus, 63  
    p\_eparparab, 63  
    p\_eparplus, 63  
    p\_par, 63  
    p\_parName, 64  
    PrintFitOutput, 62  
    PrintFitOutputInCanvas, 62  
    Reset, 62  
    Set, 62  
    SetResults, 62  
    FitRes\_Diagonal\_Get\_NberOfTermsInChi2  
        Cluster, 35  
    FitRes\_Diagonal\_Get\_Pad  
        Cluster, 35  
    FitRes\_Diagonal\_Get\_Pull  
        Cluster, 36  
    FitRes\_Diagonal\_Get\_Residual  
        Cluster, 36  
    FitRes\_Horizontal\_Get\_NberOfTermsInChi2  
        Cluster, 36  
    FitRes\_Horizontal\_Get\_Pad  
        Cluster, 36  
    FitRes\_Horizontal\_Get\_Pull  
        Cluster, 36  
    FitRes\_Horizontal\_Get\_Residual  
        Cluster, 36  
    Fitters Directory Reference, 15  
    Fitters/inc Directory Reference, 15  
    Fitters/inc/ClusterFitter.h, 162, 164  
    Fitters/inc/PRFParameters.h, 165, 166  
    Fitters/inc/TrackFitter.h, 166, 167  
    Fitters/inc/TrackRecon.h, 167, 169  
    Fitters/src Directory Reference, 18  
    Fitters/src/ClusterFitter.cxx, 169  
    Fitters/src/PRFParameters.cxx, 171  
    Fitters/src/TrackFitter.cxx, 171  
    Fitters/src/TrackRecon.cxx, 171  
FourModulesInLine  
    ReconTools.cxx, 243  
    ReconTools.h, 224  
fsaveSelectOnly  
    Reconstruction, 29  
Gain  
    EramInfo, 53  
    Reconstruction::ERAMMaps, 55  
gain  
    Reconstruction::RecoPad, 110  
GainCorrection  
    Reconstruction::RecoPad, 110

GaussFunction, 64  
 ~GaussFunction, 64  
 GaussFunction, 64  
 Get\_Mean, 65  
 Get\_Norm, 65  
 Get\_Sigma, 65  
 operator(), 65  
 Set\_Mean, 65  
 Set\_Norm, 65  
 Set\_Sigma, 65  
 Get\_A0  
     ParabolaFunction, 95  
 Get\_A0M  
     ParabolaFunctionNG, 97  
 Get\_A0P  
     ParabolaFunctionNG, 97  
 Get\_a2  
     Uploader, 142  
 Get\_a4  
     Uploader, 142  
 Get\_Acluster  
     Cluster, 36  
 Get\_AMax  
     Pad, 89  
 Get\_AMax\_FIT  
     Pad, 89  
 Get\_AMax\_FromSet  
     Pad, 89  
 Get\_AMax\_True  
     Pad, 89  
 Get\_AMax\_WF  
     Pad, 89  
 Get\_AMax\_WF\_01  
     Pad, 89  
 Get\_AMaxLeading  
     Cluster, 36  
 Get\_Amplitude  
     Model\_Electronics, 77  
 Get\_APad  
     Pad, 89  
 Get\_b2  
     Uploader, 142  
 Get\_b4  
     Uploader, 142  
 Get\_Chi2Min  
     Cluster, 36  
     Sample, 117  
     Track, 135  
 Get\_Cluster  
     Module, 84  
     Track, 135  
 Get\_Cluster\_Copy  
     Event, 57  
     Module, 85  
 Get\_CovMatrix  
     Track, 135  
 Get\_Cut\_Stage2\_EventBased  
     Selector, 122  
     Get\_Cut\_Stage3\_THigh  
         Selector, 122  
     Get\_Cut\_Stage3\_TLow  
         Selector, 122  
     Get\_Cut\_Stage4\_APM\_High  
         Selector, 122  
     Get\_Cut\_Stage4\_APM\_Low  
         Selector, 122  
     Get\_Cut\_Stage5\_Npads\_Hig  
         Selector, 122  
     Get\_Cut\_Stage6\_Amax\_Hig  
         Selector, 122  
     Get\_Cut\_Stage6\_Amax\_Low  
         Selector, 123  
     Get\_Cut\_StageFinal\_NCluster\_Low  
         Selector, 123  
     Get\_DD  
         Sample, 117  
     Get\_DD\_BeforeMinimisation  
         Sample, 117  
     Get\_eDD  
         Sample, 117  
     Get\_EntryNber  
         Cluster, 37  
         Event, 57  
         Module, 85  
         Pad, 89  
         Track, 135  
     Get\_eRC  
         Sample, 117  
     Get\_Event  
         Sample, 117  
     Get\_EventNber  
         Cluster, 37  
         Event, 58  
         Module, 85  
         Pad, 90  
         Track, 136  
     Get\_eXTrack  
         Cluster, 37  
     Get\_eYTrack  
         Cluster, 37  
     Get\_FIT\_A0M  
         Pad, 90  
     Get\_FIT\_A0P  
         Pad, 90  
     Get\_FIT\_Status  
         Pad, 90  
     Get\_FIT\_X0  
         Pad, 90  
     Get\_FIT\_Xmax  
         Pad, 90  
     Get\_FIT\_Xmin  
         Pad, 90  
     Get\_FIT\_Y0  
         Pad, 90  
     Get\_Gain  
         Model\_ChargeI, 75

Get\_IsThisPadExisting  
    ROBoard, 113  
Get\_iX  
    Pad, 90  
Get\_iY  
    Pad, 90  
Get\_Lambda  
    Model\_Charge1D, 72  
Get\_LeadingPad  
    Cluster, 37  
Get\_LX  
    Model\_ReadOutGeometry, 79  
    Pad, 91  
Get LY  
    Model\_ReadOutGeometry, 79  
    Pad, 91  
Get\_Mean  
    GaussFunction, 65  
Get\_MinimalNberOfEntries  
    SetOfTracks, 126  
Get\_Model\_Charge1D  
    Uploader, 142  
Get\_Model\_Chargel  
    Event, 58  
    Pad, 91  
    ROBoard, 113  
    Sample, 117  
Get\_Model\_Electronics  
    Event, 58  
    Pad, 91  
    ROBoard, 113  
    Sample, 117  
    Uploader, 142  
Get\_Model\_ReadOutGeometry  
    Event, 58  
    Pad, 91  
    ROBoard, 114  
    Sample, 117  
    Uploader, 142  
Get\_Module\_InArray  
    Event, 58  
Get\_ModuleNber  
    Cluster, 37  
    Module, 85  
    Pad, 91  
    Track, 136  
Get\_NberOfCluster  
    Module, 85  
    Track, 136  
Get\_NberOfEvent  
    Uploader, 142  
Get\_NberOfEvents  
    Sample, 118  
Get\_NberOfModule  
    Event, 58  
Get\_NberOfPads  
    Cluster, 37  
    ROBoard, 114  
Get\_NberOfTrack  
    SetOfTracks, 126  
Get\_NextLeadingPad  
    Cluster, 37  
Get\_NextNextLeadingPad  
    Cluster, 37  
Get\_Norm  
    GaussFunction, 65  
    Uploader, 142  
Get\_Nx  
    Model\_ReadOutGeometry, 80  
Get\_Ny  
    Model\_ReadOutGeometry, 80  
Get\_Pad  
    Cluster, 37  
    ROBoard, 114  
Get\_PadName  
    Pad, 91  
Get\_ParameterError  
    Track, 136  
Get\_ParameterName  
    Track, 136  
Get\_ParameterValue  
    Track, 136  
Get\_ParameterValue\_BeforeMinimisation  
    Track, 136  
Get\_PRF\_exist  
    Uploader, 142  
Get\_Pull  
    Track, 136  
Get\_Qpad  
    Model\_Charge0D, 69  
    Model\_Charge1D, 72  
    Model\_Chargel, 75  
    Pad, 91  
Get\_Qprim  
    Model\_Charge0D, 69  
Get\_QValue  
    Model\_Electronics, 78  
Get\_RC  
    Model\_Chargel, 75  
    Sample, 118  
Get\_RC\_BeforeMinimisation  
    Sample, 118  
Get\_Residual  
    Track, 137  
Get\_SampleFile  
    Uploader, 142  
Get\_SelectionName  
    Selector, 123  
Get\_SetOfTracks\_ForThisModule  
    Sample, 118  
Get\_Sigma  
    GaussFunction, 65  
Get\_ThisModule  
    Event, 58  
Get\_ThisModuleConst  
    Event, 58

Get\_Time0  
     Pad, 91  
 Get\_TMax  
     Pad, 91  
 Get\_TMax\_FIT  
     Pad, 92  
 Get\_TMax\_FromSet  
     Pad, 92  
 Get\_TMax\_True  
     Pad, 92  
 Get\_TMax\_WF  
     Pad, 92  
 Get\_TMax\_WF\_01  
     Pad, 92  
 Get\_TMaxLeading  
     Cluster, 38  
 Get\_Track  
     SetOfTracks, 126  
 Get\_vADC  
     Pad, 92  
 Get\_Width  
     Model\_ChargeI, 75  
 Get\_ws  
     Model\_Electronics, 78  
 Get\_X0  
     ParabolaFunction, 95  
     ParabolaFunctionNG, 97  
 Get\_XcPad  
     Model\_ReadOutGeometry, 80  
 Get\_XH  
     Pad, 92  
 Get\_XHPad  
     Model\_ReadOutGeometry, 80  
 Get\_XL  
     Pad, 92  
 Get\_XLPad  
     Model\_ReadOutGeometry, 80  
 Get\_XPad  
     Pad, 92  
 Get\_Xpad\_min  
     Model\_ReadOutGeometry, 80  
 Get\_XTrack  
     Cluster, 38  
 Get\_XTrack\_BeforeMinimisation  
     Cluster, 38  
 Get\_XTrue  
     Pad, 92  
 Get\_XWeight  
     Cluster, 38  
 Get\_Y0  
     ParabolaFunction, 95  
     ParabolaFunctionNG, 97  
 Get\_YcPad  
     Model\_ReadOutGeometry, 80  
 Get\_YH  
     Pad, 93  
 Get\_YHPad  
     Model\_ReadOutGeometry, 80  
 Get\_YL  
     Pad, 93  
 Get\_YLeading  
     Cluster, 38  
 Get\_YLPad  
     Model\_ReadOutGeometry, 81  
 Get\_YPad  
     Pad, 93  
 Get\_Ypad\_min  
     Model\_ReadOutGeometry, 81  
 Get\_YTrack  
     Cluster, 38  
 Get\_YTrack\_BeforeMinimisation  
     Cluster, 38  
 Get\_YTrue  
     Pad, 93  
 Get\_YWeight  
     Cluster, 38  
 GetFilePRF  
     Sample, 118  
 GetGigaWaveform  
     Reconstruction::dEdx, 47  
 GetLinearIndex  
     ROBoard, 114  
 GetNberOfParameters  
     Track, 137  
 GetPadEdges  
     Model\_ReadOutGeometry, 81  
 getRatio  
     Reconstruction::LUT, 68  
 GetStage3Cut\_CSV  
     ReconTools.cxx, 243  
     ReconTools.h, 225  
 GetTruncatedGigaWaveformGP1  
     Reconstruction::dEdx, 47  
 GiveMe\_AnUnfittedTrack  
     Module, 85  
 GiveMe\_AnUnfittedTrack\_ForThisModule  
     Event, 58  
 GiveMe\_Clusters\_ForThisModule  
     Event, 58  
     Module, 85  
 GiveMe\_CutInterCeptXZ\_ForThisModule  
     Event, 59  
 GiveMe\_CutSlopeXY\_ForThisModule  
     Event, 59  
 GiveMe\_CutSlopeXYZ\_ForThisModule  
     Event, 59  
 GiveMe\_CutSlopeXZ\_ForThisModule  
     Event, 59  
 GiveMe\_Event  
     Uploader, 143  
     Uploader\_ERAM01, 146  
     Uploader\_MockUp\_V1, 148  
     Uploader\_MockUp\_V2, 151  
     Uploader\_Prototype, 160  
 GiveMe\_EvtDisplay  
     Displays.cxx, 237

Displays.h, 209  
GiveMe\_ModuleTrack  
Module, 85  
GiveMe\_pTH1F\_Ch2Min  
SetOfTracks, 126  
GiveMe\_pTH1F\_Ch2MinPerNODF  
SetOfTracks, 127  
GiveMe\_pTH1F\_Pull  
SetOfTracks, 127  
GiveMe\_pTH1F\_Residual  
SetOfTracks, 128  
GiveMe\_pTH1F\_TrackDeltaT  
SetOfTracks, 128  
GiveMe\_pTH1F\_YFitCol  
SetOfTracks, 128, 129  
GiveMe\_pTH1F\_YTrackInTracks  
SetOfTracks, 129  
GiveMe\_pTH1F\_YTrackYPadInTracks  
SetOfTracks, 130  
GiveMe\_pTH1F\_YTrackYPadLeadingInTracks  
SetOfTracks, 130  
GiveMe\_TimeDisplay  
Displays.cxx, 237  
Displays.h, 209  
GiveMe\_Track\_ForThisModule  
Event, 59  
GiveMe\_Uploader  
GiveMe\_Uploader.cxx, 252  
GiveMe\_Uploader.h, 246  
GiveMe\_Uploader.cxx  
GiveMe\_Uploader, 252  
GiveMe\_Uploader.h  
GiveMe\_Uploader, 246  
GiveMe\_WaveFormDisplay  
Displays.cxx, 237  
Displays.h, 210  
GlobalChi2\_4, 66  
fChi2\_1, 66  
fChi2\_2, 66  
fChi2\_3, 66  
fChi2\_4, 66  
GlobalChi2\_4, 66  
operator(), 66  
GWF  
Reconstruction::RecoEvent, 104  
GWFtruncatedGP1  
Reconstruction::RecoEvent, 104  
hist\_to\_graph  
ReconTools.cxx, 244  
ReconTools.h, 225  
ID  
Reconstruction::ERAMMaps, 55  
Reconstruction::RecoModule, 107  
Id  
EramInfo, 54  
InbTPC  
EramInfo, 54  
Ini\_Models  
ROBoard, 114  
Init  
Uploader\_MockUp\_V2, 151  
Init\_selection  
ReconTools.hxx, 244  
ReconTools.h, 225  
inputDir  
Reconstruction, 29  
IntTPC  
EramInfo, 54  
intUploader  
Reconstruction, 29  
Invalidate  
Cluster, 38  
Event, 59  
Module, 85  
Pad, 93  
Invalidate\_ThisModule  
Event, 59  
iparmuon  
CombinedFit.hxx, 232  
iparpion  
CombinedFit.hxx, 232  
iparpositron  
CombinedFit.hxx, 232  
iparproton  
CombinedFit.hxx, 232  
isValid  
Cluster, 38  
Event, 59  
Module, 85  
Pad, 93  
ix  
Reconstruction::RecoPad, 110  
iy  
Reconstruction::RecoPad, 110  
leading  
Reconstruction::RecoPad, 111  
length  
Reconstruction::RecoCluster, 101  
Reconstruction::RecoPad, 111  
lengthWF  
Reconstruction::RecoEvent, 104  
Reconstruction::RecoModule, 107  
lengthXP  
Reconstruction::RecoEvent, 104  
Reconstruction::RecoModule, 107  
local\_params  
ReconTools.cxx, 244  
ReconTools.h, 225  
log\_file  
Reconstruction, 29  
LUT  
Reconstruction::LUT, 67  
LUTRhoLead  
Reconstruction::RecoCluster, 101  
LUTs.hxx

**ClassImp**, 241  
**m\_2RCinv**  
  Model\_ChargeI, 76  
**m\_a2**  
  Uploader, 143  
**m\_a4**  
  Uploader, 143  
**m\_AngleRot**  
  Cluster, 41  
**m\_b2**  
  Uploader, 143  
**m\_b4**  
  Uploader, 143  
**m\_Description**  
  FitOutput, 62  
**m\_Dolons**  
  Model\_Charge0D, 70  
**m\_FitterName**  
  ClusterFitter\_Diagonal, 43  
  ClusterFitter\_Horizontal, 45  
  TrackFitter, 140  
**m\_Gain**  
  Model\_ChargeI, 76  
**m\_Lambda**  
  Model\_Charge1D, 73  
**m\_MinnLL**  
  FitOutput, 62  
**m\_NameOfTheModel**  
  FitOutput, 63  
**m\_NberOfClusters**  
  Track, 137  
**m\_NberOfDataPoints**  
  FitOutput, 63  
**m\_NberOfEvent**  
  Uploader, 143  
**m\_NberOfModelParameters**  
  FitOutput, 63  
**m\_NberOfParameters**  
  TrackFitter, 140  
**m\_Norm**  
  Uploader, 143  
**m\_NormQPad**  
  Model\_ChageI, 76  
**m\_Nx**  
  ROBoard, 115  
**m\_Ny**  
  ROBoard, 115  
**m\_PRF\_exist**  
  Uploader, 143  
**m\_Qprim**  
  Model\_Chage0D, 70  
**m\_RC**  
  Model\_ChageI, 76  
**m\_RFactor**  
  Model\_Chage0D, 70  
**m\_RTrue**  
  Model\_Chage0D, 70  
**m\_SampleFile**  
    Uploader, 143  
    m\_Tion  
      Model\_Chage0D, 71  
    m\_Width  
      Model\_ChageI, 76  
    m\_Width2  
      Model\_ChageI, 76  
    main  
      Analysis.hxx, 161  
      DrawOut.hxx, 162  
    max\_mult  
      Uploader\_MockUp\_V2, 156  
    mean\_mult  
      Uploader\_MockUp\_V2, 156  
    MeanGain  
      Reconstruction::ERAMMaps, 55  
    MeanRC  
      Reconstruction::ERAMMaps, 55  
    Model\_Chage0D, 68  
      ~Model\_Chage0D, 69  
      Get\_Qpad, 69  
      Get\_Qprim, 69  
      m\_Dolons, 70  
      m\_Qprim, 70  
      m\_RFactor, 70  
      m\_RTrue, 70  
      m\_Tion, 71  
      Model\_Chage0D, 69  
      Set\_Ion, 70  
      Set\_Qprim, 70  
      SetSecondaries, 70  
      WriteOut, 70  
    Model\_Chage1D, 71  
      ~Model\_Chage1D, 72  
      Get\_Lambda, 72  
      Get\_Qpad, 72  
      m\_Lambda, 73  
      Model\_Chage1D, 72  
      Set\_Lambda, 73  
      SetSecondaries, 73  
      WriteOut, 73  
    Model\_ChageI, 74  
      ~Model\_ChageI, 74  
      Get\_Gain, 75  
      Get\_Qpad, 75  
      Get\_RC, 75  
      Get\_Width, 75  
      m\_2RCinv, 76  
      m\_Gain, 76  
      m\_NormQPad, 76  
      m\_RC, 76  
      m\_Width, 76  
      m\_Width2, 76  
      Model\_ChageI, 74  
      Set\_Gain, 75  
      Set\_RC, 75  
      Set\_Width, 75  
      Set\_WidthFromDriftDistance, 75

SetSecondaries, 76  
WidthFromDriftDistance, 76  
WriteOut, 76  
Model\_Electronics, 77  
~Model\_Electronics, 77  
Derive\_Response\_Base, 77  
Get\_Amplitude, 77  
Get\_QValue, 78  
Get\_ws, 78  
Model\_Electronics, 77  
Response\_Base, 78  
Set\_Amplitude, 78  
Set\_TimeShape, 78  
WriteOut, 78  
Model\_ReadOutGeometry, 78  
~Model\_ReadOutGeometry, 79  
Get\_LX, 79  
Get LY, 79  
Get\_Nx, 80  
Get Ny, 80  
Get\_XcPad, 80  
Get\_XHPad, 80  
Get\_XLPad, 80  
Get\_Xpad\_min, 80  
Get\_YcPad, 80  
Get\_YHPad, 80  
Get\_YLPad, 81  
Get\_Ypad\_min, 81  
GetPadEdges, 81  
Model\_ReadOutGeometry, 79  
Set\_LX, 81  
Set LY, 81  
Set Nx, 81  
Set Ny, 81  
Set\_Xpad\_min, 82  
Set\_Ypad\_min, 82  
SquareGeometry, 82  
SwapGeometry, 82  
WriteOut, 82  
ModelEvent Directory Reference, 18  
ModelEvent/inc Directory Reference, 16  
ModelEvent/inc/AmplitudeError.h, 173  
ModelEvent/inc/Cluster.h, 173, 174  
ModelEvent/inc/Event.h, 176  
ModelEvent/inc/FitOutput.h, 178  
ModelEvent/inc/Module.h, 179  
ModelEvent/inc/Pad.h, 180, 181  
ModelEvent/inc/ROBoard.h, 183, 184  
ModelEvent/inc/Sample.h, 185  
ModelEvent/inc/SetOfTracks.h, 187  
ModelEvent/inc/TimeError.h, 189  
ModelEvent/inc/Track.h, 190  
ModelEvent/src Directory Reference, 19  
ModelEvent/src/AmplitudeError.cxx, 191  
ModelEvent/src/Cluster.hxx, 192  
ModelEvent/src/Event.hxx, 192  
ModelEvent/src/FitOutput.hxx, 192  
ModelEvent/src/Module.hxx, 192  
ModelEvent/src/Pad.hxx, 193  
ModelEvent/src/ROBoard.hxx, 193  
ModelEvent/src/Sample.hxx, 193  
ModelEvent/src/SetOfTracks.hxx, 193  
ModelEvent/src/TimeError.hxx, 193  
ModelEvent/src/Track.hxx, 195  
ModelSignal Directory Reference, 18  
ModelSignal/inc Directory Reference, 16  
ModelSignal/inc/Model\_Charge0D.h, 195, 196  
ModelSignal/inc/Model\_Charge1D.h, 196, 197  
ModelSignal/inc/Model\_Charge1L.h, 197, 198  
ModelSignal/inc/Model\_Electronics.h, 198, 199  
ModelSignal/inc/Model\_ReadOutGeometry.h, 199, 200  
ModelSignal/src Directory Reference, 19  
ModelSignal/src/Model\_Charge0D.hxx, 201  
ModelSignal/src/Model\_Charge1D.hxx, 201  
ModelSignal/src/Model\_Charge1L.hxx, 201  
ModelSignal/src/Model\_Electronics.hxx, 201  
ModelSignal/src/Model\_ReadOutGeometry.hxx, 201  
Module, 82  
~Module, 84  
Add\_Cluster, 84  
Add\_Pad, 84  
Clear\_Clusters, 84  
Get\_Cluster, 84  
Get\_Cluster\_Copy, 85  
Get\_EntryNber, 85  
Get\_EventNber, 85  
Get\_ModuleNber, 85  
Get\_NberOfCluster, 85  
GiveMe\_AnUnfittedTrack, 85  
GiveMe\_Clusters\_ForThisModule, 85  
GiveMe\_ModuleTrack, 85  
Invalidate, 85  
IsValid, 85  
Module, 84  
ReplaceClusters, 86  
Set\_ModuleTrack, 86  
SmallDump, 86  
Validate, 86  
WriteOut, 86  
module  
  Uploader\_MockUp\_V2, 156  
moduleCase  
  Reconstruction, 29  
mom  
  Uploader\_MockUp\_V2, 156  
Monitoring  
  Reconstruction, 24  
multiplicity  
  Uploader\_MockUp\_V2, 156  
multiScanName  
  Reconstruction, 29  
NberOfSelections  
  Selector, 123  
NClusters  
  Reconstruction::RecoEvent, 105  
  Reconstruction::RecoModule, 107

NCrossedPads  
 Reconstruction::RecoEvent, 105  
 Reconstruction::RecoModule, 107

NewClusterDisplay  
 Displays.cxx, 237  
 Displays.h, 210

NewClusterDisplayMinimal  
 Displays.cxx, 238  
 Displays.h, 210

NPad  
 Reconstruction::RecoCluster, 101  
 Reconstruction::RecoEvent, 105  
 Reconstruction::RecoModule, 107

npar  
 CombinedFitcxx, 233

numberOfModules  
 Reconstruction::RecoEvent, 105

offset  
 Uploader\_MockUp\_V2, 156

operator()  
 GaussFunction, 65  
 GlobalChi2\_4, 66  
 ParabolaFunction, 95  
 ParabolaFunctionNG, 97  
 PRFParameters, 99

operator=  
 FitOutput, 62

p\_Cluster  
 ClusterFitter\_Diagonal, 43  
 ClusterFitter\_Horizontal, 45

p\_ClusterFitter\_Diagonal  
 StaticClusterFitter\_Diagonal, 131

p\_ClusterFitter\_Horizontal  
 StaticClusterFitter\_Horizontal, 132

p\_CovMatrix  
 FitOutput, 63

p\_dEdx  
 Reconstruction, 29

p\_DrawOuts  
 Reconstruction, 29

p\_eparminus  
 FitOutput, 63

p\_eparparab  
 FitOutput, 63

p\_eparplus  
 FitOutput, 63

p\_lut  
 Reconstruction, 29

p\_Model\_Charge1D  
 Uploader, 144

p\_Model\_Chargel  
 ROBoard, 115

p\_Model\_Electronics  
 ROBoard, 115  
 Uploader, 144

p\_Model\_ReadOutGeometry  
 ROBoard, 115

Uploader, 144

p\_par  
 FitOutput, 63

p\_parName  
 FitOutput, 64

p\_TFile  
 Uploader, 144

p\_Track  
 TrackFitter, 140

p\_TrackFitter  
 StaticTrackFitter, 133

p\_TTree  
 Uploader, 144

p\_TVirtuallFitter  
 ClusterFitter\_Diagonal, 43  
 ClusterFitter\_Horizontal, 45  
 TrackFitter, 140

p\_uploader  
 Reconstruction, 30

Pad, 86  
 ~Pad, 88  
 Clear\_ADC, 89  
 Get\_AMax, 89  
 Get\_AMax\_FIT, 89  
 Get\_AMax\_FromSet, 89  
 Get\_AMax\_True, 89  
 Get\_AMax\_WF, 89  
 Get\_AMax\_WF\_01, 89  
 Get\_APad, 89  
 Get\_EntryNber, 89  
 Get\_EventNber, 90  
 Get\_FIT\_A0M, 90  
 Get\_FIT\_A0P, 90  
 Get\_FIT\_Status, 90  
 Get\_FIT\_X0, 90  
 Get\_FIT\_Xmax, 90  
 Get\_FIT\_Xmin, 90  
 Get\_FIT\_Y0, 90  
 Get\_iX, 90  
 Get\_iY, 90  
 Get\_LX, 91  
 Get LY, 91  
 Get Model\_Chargel, 91  
 Get Model\_Electronics, 91  
 Get Model\_ReadOutGeometry, 91  
 Get ModuleNber, 91  
 Get PadName, 91  
 Get Qpad, 91  
 Get Time0, 91  
 Get TMax, 91  
 Get TMax\_FIT, 92  
 Get TMax\_FromSet, 92  
 Get TMax\_True, 92  
 Get TMax\_WF, 92  
 Get TMax\_WF\_01, 92  
 Get vADC, 92  
 Get XH, 92  
 Get XL, 92

Get\_XPad, 92  
Get\_XTrue, 92  
Get\_YH, 93  
Get\_YL, 93  
Get\_YPad, 93  
Get\_YTrue, 93  
Invalidate, 93  
IsValid, 93  
Pad, 88  
Set\_ADC, 93  
Set\_AMax, 93  
Set\_AMax\_True, 93  
Set\_Data\_2Use, 93  
Set\_TMax, 94  
Set\_TMax\_True, 94  
SetSignalModel, 94  
Validate, 94  
WF\_DoClosure, 94  
WriteOut, 94  
pad\_charge  
    Uploader\_MockUp\_V2, 156  
pad\_time  
    Uploader\_MockUp\_V2, 156  
pad\_wf\_q  
    Uploader\_MockUp\_V2, 157  
pad\_x  
    Uploader\_MockUp\_V2, 157  
pad\_y  
    Uploader\_MockUp\_V2, 157  
ParabolaFunction, 94  
    ~ParabolaFunction, 95  
    Get\_A0, 95  
    Get\_X0, 95  
    Get\_Y0, 95  
    operator(), 95  
    ParabolaFunction, 95  
    Set\_A0, 95  
    Set\_X0, 96  
    Set\_Y0, 96  
ParabolaFunctionNG, 96  
    ~ParabolaFunctionNG, 96  
    Get\_A0M, 97  
    Get\_A0P, 97  
    Get\_X0, 97  
    Get\_Y0, 97  
    operator(), 97  
    ParabolaFunctionNG, 96  
    Set\_A0M, 97  
    Set\_A0P, 97  
    Set\_X0, 97  
    Set\_Y0, 97  
peakingTime  
    Reconstruction::RecoEvent, 105  
phi  
    Reconstruction::RecoModule, 108  
    Reconstruction::RecoPad, 111  
Position  
    EramInfo, 54  
position  
    Reconstruction::RecoModule, 108  
PRFParameters, 98  
    ~PRFParameters, 98  
    Eval, 98  
    operator(), 99  
    PRFParameters, 98  
    SetPRF, 99  
PrintFitOutput  
    FitOutput, 62  
PrintFitOutputInCanvas  
    FitOutput, 62  
PrintStat  
    Selector, 123  
prototype  
    Reconstruction, 30  
PT  
    Reconstruction, 30  
pUploader  
    Reconstruction, 30  
qfrac  
    Uploader\_MockUp\_V2, 157  
quality  
    Uploader\_MockUp\_V2, 157  
ratio  
    Reconstruction::RecoPad, 111  
ratioCorr  
    Reconstruction::RecoCluster, 101  
ratioDrift  
    Reconstruction::RecoPad, 111  
ratioFile  
    Reconstruction::RecoPad, 111  
RC  
    EramInfo, 54  
    Reconstruction::ERAMMaps, 56  
    Reconstruction::RecoPad, 111  
readCSV  
    ReconTools.cxx, 244  
    ReconTools.h, 225  
RecoEvent  
    Reconstruction::RecoEvent, 103  
Reconstruction, 21  
    CERN\_drift, 25  
    CERN\_Escan, 25  
    ClearVectors, 23  
    comment, 25  
    Correction, 23  
    correction\_wf, 25  
    corrFuncPath, 25  
    dataPath, 25  
    dataScanPath, 25  
    dedx, 25  
    dEdxPath, 25  
    DefaultAnalysis, 24  
    DESY\_drift, 25  
    DESY\_phi, 26  
    DESY\_row, 26

DESY\_theta, 26  
 DESY\_yscan, 26  
 DESY\_zscan, 26  
 DO\_Comparison, 26  
 DO\_Control, 26  
 DO\_dEdx, 26  
 Draw\_CERN22Scan, 26  
 Draw\_Comparison, 26  
 Draw\_Control, 27  
 Draw\_Corrections, 27  
 Draw\_dEdx, 27  
 Draw\_DESY21MultScan, 27  
 Draw\_DESY21SingleScan, 27  
 DrawCERN22Scan, 24  
 DrawMultipleScan, 24  
 drawout\_file, 27  
 drawoutMultiScanPath, 27  
 drawoutPath, 27  
 drawoutRunPath, 27  
 drawoutScanPath, 27  
 DrawSingleScan, 24  
 drawWhichMethods, 28  
 driftDist, 28  
 Dt, 28  
 DtB, 28  
 DtnoB, 28  
 DtwithBhere, 28  
 DtwithoutBhere, 28  
 fcorrectDrift, 28  
 fcorrectGain, 28  
 fcorrectRC, 28  
 fcorrectWF, 29  
 fsaveSelectOnly, 29  
 inputDir, 29  
 intUploader, 29  
 log\_file, 29  
 moduleCase, 29  
 Monitoring, 24  
 multiScanName, 29  
 p\_dEdx, 29  
 p\_DrawOuts, 29  
 p\_lut, 29  
 p\_uploader, 30  
 prototype, 30  
 PT, 30  
 pUploader, 30  
 Reconstruction::dEdx, 47  
 rootout\_file, 30  
 runvarstr, 30  
 scanIndex, 30  
 scanName, 30  
 selectionSet, 30  
 Settings, 24  
 tag, 30  
 TB, 31  
 testbeam, 31  
 v\_comments, 31  
 v\_datafiles, 31  
 v\_rootout\_files, 31  
 v\_scanspec, 31  
 vcorrFuncPaths, 31  
 vScanLabels, 31  
 vScanVals, 31  
 vTags, 31  
 WFCorrection, 24  
 Reconstruction Directory Reference, 18  
 Reconstruction/inc Directory Reference, 17  
 Reconstruction/inc/CombinedFit.h, 201, 202  
 Reconstruction/inc/Control.h, 202, 203  
 Reconstruction/inc/dEdx.h, 203, 204  
 Reconstruction/inc/Displays.h, 207, 210  
 Reconstruction/inc/DrawOuts.h, 211, 212  
 Reconstruction/inc/DrawOuts\_old.h, 213, 216  
 Reconstruction/inc/LUTs.h, 217, 218  
 Reconstruction/inc/Monitoring.h, 220, 221  
 Reconstruction/inc/Recon\_LinkDef.h, 222  
 Reconstruction/inc/ReconTools.h, 222, 226  
 Reconstruction/inc/Selector.h, 227, 228  
 Reconstruction/inc/Variables.h, 229, 231  
 Reconstruction/src Directory Reference, 19  
 Reconstruction/src/CombinedFit.cxx, 232  
 Reconstruction/src/Control.cxx, 233  
 Reconstruction/src/dEdx.cxx, 234  
 Reconstruction/src/Displays.cxx, 235  
 Reconstruction/src/DrawOuts.cxx, 238  
 Reconstruction/src/DrawOuts\_old.cxx, 239  
 Reconstruction/src/LUTs.cxx, 240  
 Reconstruction/src/Monitoring.cxx, 241  
 Reconstruction/src/ReconTools.cxx, 242  
 Reconstruction/src/Selector.cxx, 245  
 Reconstruction/src/Variables.cxx, 245  
 Reconstruction::dEdx, 45  
     ~dEdx, 46  
     ComputedEdxGP, 46  
     ComputedEdxGP1, 46  
     ComputedEdxWF, 46  
     ComputedEdxXP, 47  
     dEdx, 46  
     DiscardedModule, 47  
     GetGigaWaveform, 47  
     GetTruncatedGigaWaveformGP1, 47  
     Reconstruction, 47  
 Reconstruction::DrawOuts, 47  
     ~DrawOuts, 50  
     AmplitudeVSLength, 50  
     CERN22Scan, 50  
     CERN22ScanDraw, 50  
     CERN22ScanFill, 50  
     Control, 50  
     DESY21MultiScan, 51  
     DESY21ScanDraw, 51  
     DESY21ScanFill, 51  
     DESY21SingleScan, 51  
     DrawOuts, 48, 50  
     EnergyLoss, 51  
     FileComparison, 52

SetStyle, 52  
Reconstruction::ERAMMaps, 55  
  ~ERAMMaps, 55  
  ERAMMaps, 55  
  Gain, 55  
  ID, 55  
  MeanGain, 55  
  MeanRC, 55  
  RC, 56  
  Resolution, 56  
Reconstruction::LUT, 67  
  ~LUT, 67  
  getRatio, 68  
  LUT, 67  
Reconstruction::RecoCluster, 99  
  ~RecoCluster, 100  
  ADCmax\_base, 100  
  ALead\_base, 100  
  ALead\_GCorr, 100  
  charge, 101  
  ClassDef, 100  
  dEdxWF, 101  
  length, 101  
  LUTrhoLead, 101  
  NPads, 101  
  ratioCorr, 101  
  TLead, 101  
  v\_pads, 101  
  yCluster, 101  
  yWeight, 101  
Reconstruction::RecoEvent, 102  
  ~RecoEvent, 103  
  avg\_pad\_mult, 103  
  ClassDef, 103  
  Clear, 103  
  dEdxGP1, 103  
  dEdxGP2, 103  
  dEdxGP3, 103  
  dEdxGP4, 103  
  dEdxGP5, 104  
  dEdxWF, 104  
  dEdxWFnoTrunc, 104  
  dEdxXP, 104  
  dEdxXPnoTrunc, 104  
  eventNbr, 104  
  GWF, 104  
  GWFtruncatedGP1, 104  
  lengthWF, 104  
  lengthXP, 104  
  NClusters, 105  
  NCrossedPads, 105  
  NPads, 105  
  numberOfModules, 105  
  peakingTime, 105  
  RecoEvent, 103  
  selected, 105  
  timeBinSize, 105  
  v\_modules, 105  
          v\_modules\_position, 105  
Reconstruction::RecoModule, 106  
  ~RecoModule, 106  
  avg\_pad\_mult, 107  
  ClassDef, 106  
  dEdxWF, 107  
  dEdxWFnoTrunc, 107  
  dEdxXP, 107  
  dEdxXPnoTrunc, 107  
  ID, 107  
  lengthWF, 107  
  lengthXP, 107  
  NClusters, 107  
  NCrossedPads, 107  
  NPads, 107  
  phi, 108  
  position, 108  
  selected, 108  
  Track, 108  
  v\_clusters, 108  
Reconstruction::RecoPad, 108  
  ~RecoPad, 109  
  ADCmax, 109  
  ADCmax\_base, 109  
  charge, 110  
  ClassDef, 109  
  d, 110  
  dd, 110  
  dEdxXP, 110  
  driftDistance, 110  
  dy, 110  
  gain, 110  
  GainCorrection, 110  
  ix, 110  
  iy, 110  
  leading, 111  
  length, 111  
  phi, 111  
  ratio, 111  
  ratioDrift, 111  
  ratioFile, 111  
  RC, 111  
  T0, 111  
  TMax, 111  
  xPad, 111  
  yPad, 112  
ReconTools.cxx  
  BetheBloch, 242  
  BetheBlochBhabha, 242  
  BetheBlochExp, 242  
  ComputeCutStage3\_Cut, 243  
  corr\_func, 243  
  DrawTH2, 243  
  Fit2Gauss, 243  
  FourModulesInLine, 243  
  GetStage3Cut\_CSV, 243  
  hist\_to\_graph, 244  
  Init\_selection, 244

local\_params, 244  
 readCSV, 244  
 SetStage3Cut\_CSV, 244  
 Swapped\_graph, 244  
 trk\_len, 245  
**ReconTools.h**  
     BetheBloch, 223  
     BetheBlochBhabha, 223  
     BetheBlochExp, 224  
     ComputeCutStage3\_Cut, 224  
     corr\_func, 224  
     DrawTH2, 224  
     Fit2Gauss, 224  
     FourModulesInLine, 224  
     GetStage3Cut\_CSV, 225  
     hist\_to\_graph, 225  
     Init\_selection, 225  
     local\_params, 225  
     readCSV, 225  
     SetStage3Cut\_CSV, 225  
     Swapped\_graph, 225  
     trk\_len, 226  
**Replace\_Clusters\_ForThisModule**  
     Event, 59  
**ReplaceClusters**  
     Module, 86  
**Reset**  
     FitOutput, 62  
**Reset\_Selection**  
     Selector, 123  
**Reset\_StatCounters**  
     Selector, 123  
**residual**  
     Uploader\_MockUp\_V2, 157  
**residual\_corr**  
     Uploader\_MockUp\_V2, 157  
**Resolution**  
     EramInfo, 54  
     Reconstruction::ERAMMaps, 56  
**Response\_Base**  
     Model\_Electronics, 78  
**rob\_clusters**  
     Uploader\_MockUp\_V2, 157  
**ROBoard**, 112  
     ~ROBoard, 113  
     Add\_Pad, 113  
     Get\_IsThisPadExisting, 113  
     Get\_Model\_ChargeI, 113  
     Get\_Model\_Electronics, 113  
     Get\_Model\_ReadOutGeometry, 114  
     Get\_NberOfPads, 114  
     Get\_Pad, 114  
     GetLinearIndex, 114  
     Ini\_Models, 114  
     m\_Nx, 115  
     m\_Ny, 115  
     p\_Model\_ChargeI, 115  
     p\_Model\_Electronics, 115  
     p\_Model\_ReadOutGeometry, 115  
     ROBoard, 113  
     V\_ExisFlag2D, 115  
     V\_Pad, 115  
**ROBoard.h**  
     ROBoard\_H, 184  
**ROBoard\_H**  
     ROBoard.h, 184  
**rootout\_file**  
     Reconstruction, 30  
**runvarstr**  
     Reconstruction, 30  
**Sample**, 115  
     ~Sample, 116  
     Add\_Event, 117  
     Get\_Chi2Min, 117  
     Get\_DD, 117  
     Get\_DD\_BeforeMinimisation, 117  
     Get\_eDD, 117  
     Get\_eRC, 117  
     Get\_Event, 117  
     Get\_Model\_ChargeI, 117  
     Get\_Model\_Electronics, 117  
     Get\_Model\_ReadOutGeometry, 117  
     Get\_NberOfEvents, 118  
     Get\_RC, 118  
     Get\_RC\_BeforeMinimisation, 118  
     Get\_SetOfTracks\_ForThisModule, 118  
     GetFilePRF, 118  
     Sample, 116  
     Set\_DD, 118  
     Set\_Model\_Charge, 118  
     Set\_Model\_Electronics, 118  
     Set\_Model\_ReadOutGeometry, 118  
     Set\_RC, 119  
     SetFilePRF, 119  
     SmallDump, 119  
     StatusFit, 119  
     WriteOut, 119  
**scanIndex**  
     Reconstruction, 30  
**scanName**  
     Reconstruction, 30  
**selected**  
     Reconstruction::RecoEvent, 105  
     Reconstruction::RecoModule, 108  
**selectionSet**  
     Reconstruction, 30  
**Selector**, 119  
     ~Selector, 121  
     Add\_Selection, 121  
     Apply\_ASelection, 121  
     ApplySelection, 121, 122  
     Get\_Cut\_Stage2\_EventBased, 122  
     Get\_Cut\_Stage3\_THigh, 122  
     Get\_Cut\_Stage3\_TLow, 122  
     Get\_Cut\_Stage4\_APM\_High, 122  
     Get\_Cut\_Stage4\_APM\_Low, 122

Get\_Cut\_Stage5\_Npads\_Hig, 122  
Get\_Cut\_Stage6\_Amax\_Hig, 122  
Get\_Cut\_Stage6\_Amax\_Low, 123  
Get\_Cut\_StageFinal\_NCluster\_Low, 123  
Get\_SelectionName, 123  
NberOfSelections, 123  
PrintStat, 123  
Reset\_Selection, 123  
Reset\_StatCounters, 123  
Selector, 121  
Set\_Cut\_Stage2\_EventBased, 123  
Set\_Cut\_Stage3\_THigh, 123  
Set\_Cut\_Stage3\_TLow, 124  
Set\_Cut\_Stage4\_APM\_High, 124  
Set\_Cut\_Stage4\_APM\_Low, 124  
Set\_Cut\_Stage5\_Npads\_Hig, 124  
Set\_Cut\_Stage6\_Amax\_Hig, 124  
Set\_Cut\_Stage6\_Amax\_Low, 124  
Set\_Cut\_StageFinal\_NCluster\_Low, 124  
Tell\_Selection, 124  
Selector.h  
    Selector\_H, 228  
Selector\_H  
    Selector.h, 228  
Set  
    FitOutput, 62  
    StaticClusterFitter\_Diagonal, 131  
    StaticClusterFitter\_Horizontal, 132  
    StaticTrackFitter, 133  
Set\_A0  
    ParabolaFunction, 95  
Set\_A0M  
    ParabolaFunctionNG, 97  
Set\_A0P  
    ParabolaFunctionNG, 97  
Set\_ADC  
    Pad, 93  
Set\_AMax  
    Pad, 93  
Set\_AMax\_True  
    Pad, 93  
Set\_Amplitude  
    Model\_Electronics, 78  
Set\_Cluster  
    ClusterFitter\_Diagonal, 42  
    ClusterFitter\_Horizontal, 45  
Set\_Cut\_Stage2\_EventBased  
    Selector, 123  
Set\_Cut\_Stage3\_THigh  
    Selector, 123  
Set\_Cut\_Stage3\_TLow  
    Selector, 124  
Set\_Cut\_Stage4\_APM\_High  
    Selector, 124  
Set\_Cut\_Stage4\_APM\_Low  
    Selector, 124  
Set\_Cut\_Stage5\_Npads\_Hig  
    Selector, 124  
Set\_Cut\_Stage6\_Amax\_Hig  
    Selector, 124  
Set\_Cut\_Stage6\_Amax\_Low  
    Selector, 124  
Set\_Cut\_StageFinal\_NCluster\_Low  
    Selector, 124  
Set\_Data\_2Use  
    Pad, 93  
Set\_DD  
    Sample, 118  
Set\_eXTrack  
    Cluster, 39  
Set\_eYTrack  
    Cluster, 39  
Set\_Gain  
    Model\_Charge1, 75  
Set\_Ion  
    Model\_Charge0D, 70  
Set\_Lambda  
    Model\_Charge1D, 73  
Set\_LX  
    Model\_ReadOutGeometry, 81  
Set LY  
    Model\_ReadOutGeometry, 81  
Set\_Mean  
    GaussFunction, 65  
Set\_MinimalNberOfEntries  
    SetOfTracks, 130  
Set\_Model\_Charge  
    Sample, 118  
Set\_Model\_Electronics  
    Sample, 118  
Set\_Model\_ReadOutGeometry  
    Sample, 118  
Set\_ModuleTrack  
    Module, 86  
Set\_Norm  
    GaussFunction, 65  
Set\_Nx  
    Model\_ReadOutGeometry, 81  
Set\_Ny  
    Model\_ReadOutGeometry, 81  
Set\_Qprim  
    Model\_Charge0D, 70  
Set\_RC  
    Model\_Charge1, 75  
    Sample, 119  
Set\_Sigma  
    GaussFunction, 65  
Set\_TimeShape  
    Model\_Electronics, 78  
Set\_TMax  
    Pad, 94  
Set\_TMax\_True  
    Pad, 94  
Set\_Track  
    TrackFitter, 139  
Set\_Track\_ForThisModule

Event, 60  
**Set\_Width**  
 Model\_ChargeI, 75  
**Set\_WidthFromDriftDistance**  
 Model\_ChargeI, 75  
**Set\_X0**  
 ParabolaFunction, 96  
 ParabolaFunctionNG, 97  
**Set\_Xpad\_min**  
 Model\_ReadOutGeometry, 82  
**Set\_XTrack**  
 Cluster, 39  
**Set\_Y0**  
 ParabolaFunction, 96  
 ParabolaFunctionNG, 97  
**Set\_Ypad\_min**  
 Model\_ReadOutGeometry, 82  
**Set\_YTrack**  
 Cluster, 39  
**SetBeforeMerging**  
 Uploader\_MockUp\_V2, 151  
**SetEval\_Diagonal**  
 Cluster, 39  
**SetEval\_Horizontal**  
 Cluster, 39  
**SetFilePRF**  
 Sample, 119  
**SetNberOfParameters**  
 Track, 137  
**SetOfTracks**, 125  
 ~SetOfTracks, 126  
 Add\_Track, 126  
 DumpRec, 126  
 Get\_MinimalNberOfEntries, 126  
 Get\_NberOfTrack, 126  
 Get\_Track, 126  
 GiveMe\_pTH1F\_Ch2Min, 126  
 GiveMe\_pTH1F\_Ch2MinPerNODF, 127  
 GiveMe\_pTH1F\_Pull, 127  
 GiveMe\_pTH1F\_Residual, 128  
 GiveMe\_pTH1F\_TrackDeltaT, 128  
 GiveMe\_pTH1F\_YFitCol, 128, 129  
 GiveMe\_pTH1F\_YTrackInTracks, 129  
 GiveMe\_pTH1F\_YTrackYPadInTracks, 130  
 GiveMe\_pTH1F\_YTrackYPadLeadingInTracks, 130  
 Set\_MinimalNberOfEntries, 130  
 SetOfTracks, 126  
**SetParameter**  
 Track, 137  
**SetParameter\_Diagonal**  
 Cluster, 39  
**SetParameter\_Horizontal**  
 Cluster, 39  
**SetParameters\_Internal**  
 Track, 137  
**SetPRF**  
 PRFParameters, 99  
**SetResults**  
 FitOutput, 62  
 Track, 137  
**SetResults\_Diagonal**  
 Cluster, 39  
**SetResults\_FailedFit\_Diagonal**  
 Cluster, 40  
**SetResults\_FailedFit\_Horizontal**  
 Cluster, 40  
**SetResults\_Horizontal**  
 Cluster, 40  
**SetSecondaries**  
 Model\_Charge0D, 70  
 Model\_Charge1D, 73  
 Model\_ChargeI, 76  
**SetSignalModel**  
 Pad, 94  
**SetStage3Cut\_CSV**  
 ReconTools.hxx, 244  
 ReconTools.h, 225  
**SetStyle**  
 Reconstruction::DrawOuts, 52  
**Settings**  
 Reconstruction, 24  
**Setwap\_XY**  
 Uploader\_MockUp\_V2, 151  
**SignalTools.hxx**  
 ETF, 259  
**SignalTools.h**  
 ETF, 258  
**sina**  
 Uploader\_MockUp\_V2, 157  
**SmallDump**  
 Event, 60  
 Module, 86  
 Sample, 119  
**SquareGeometry**  
 Model\_ReadOutGeometry, 82  
**StaticClusterFitter\_Diagonal**, 131  
 ~StaticClusterFitter\_Diagonal, 131  
 p\_ClusterFitter\_Diagonal, 131  
 Set, 131  
 StaticClusterFitter\_Diagonal, 131  
**StaticClusterFitter\_Horizontal**, 132  
 ~StaticClusterFitter\_Horizontal, 132  
 p\_ClusterFitter\_Horizontal, 132  
 Set, 132  
 StaticClusterFitter\_Horizontal, 132  
**StaticTrackFitter**, 133  
 ~StaticTrackFitter, 133  
 p\_TrackFitter, 133  
 Set, 133  
 StaticTrackFitter, 133  
**StatusFit**  
 Cluster, 40  
 Sample, 119  
**StatusFit\_Diagonal**  
 Cluster, 40  
**StatusFit\_Horizontal**

Cluster, 40  
SwapGeometry  
  Model\_ReadOutGeometry, 82  
Swapped\_graph  
  ReconTools.cxx, 244  
  ReconTools.h, 225

T0  
  Reconstruction::RecoPad, 111

tag  
  Reconstruction, 30

TB  
  Reconstruction, 31

Tell\_Selection  
  Selector, 124

testbeam  
  Reconstruction, 31

time  
  Uploader\_MockUp\_V2, 157

timeBinSize  
  Reconstruction::RecoEvent, 105

TimeError  
  TimeError.cxx, 194  
  TimeError.h, 189

TimeError.cxx  
  TimeError, 194  
  TimeError\_BASE, 194  
  TimeError\_Leading, 194  
  TimeError\_NextLeading, 194  
  TimeError\_NextNextLeading, 194

TimeError.h  
  TimeError, 189

TimeError\_BASE  
  TimeError.cxx, 194

TimeError\_Leading  
  TimeError.cxx, 194

TimeError\_NextLeading  
  TimeError.cxx, 194

TimeError\_NextNextLeading  
  TimeError.cxx, 194

TLead  
  Reconstruction::RecoCluster, 101

TMax  
  Reconstruction::RecoPad, 111

Track, 134  
  ~Track, 135  
  Add\_Cluster, 135  
  Chi2, 135  
  DoClosure, 135  
  DumpRec, 135  
  Get\_Chi2Min, 135  
  Get\_Cluster, 135  
  Get\_CovMatrix, 135  
  Get\_EntryNber, 135  
  Get\_EventNber, 136  
  Get\_ModuleNber, 136  
  Get\_NberOfCluster, 136  
  Get\_ParameterError, 136  
  Get\_ParameterName, 136  
  Get\_ParameterValue, 136  
  Get\_ParameterValue\_BeforeMinimisation, 136  
  Get\_Pull, 136  
  Get\_Residual, 137  
  GetNberOfParameters, 137  
  m\_NberOfClusters, 137  
  Reconstruction::RecoModule, 108  
  SetNberOfParameters, 137  
  SetParameter, 137  
  SetParameters\_Internal, 137  
  SetResults, 137  
  Track, 134  
  V\_Cluster, 137  
  V\_Pull, 138  
  V\_Residual, 138  
  Y\_Position, 137

track  
  Uploader\_MockUp\_V2, 158

track\_pos  
  Uploader\_MockUp\_V2, 158

TrackFitter, 138  
  ~TrackFitter, 139  
  Chi2, 139  
  DoMinimisation, 139  
  m\_FitterName, 140  
  m\_NberOfParameters, 140  
  p\_Track, 140  
  p\_TVirtuallFitter, 140  
  Set\_Track, 139  
  TrackFitter, 138

TrackFitter.cxx  
  TrackFitterFunction, 171

TrackFitterFunction  
  TrackFitter.cxx, 171

TrackRecon  
  TrackRecon.cxx, 172  
  TrackRecon.h, 168

TrackRecon.cxx  
  TrackRecon, 172  
  TrackRecon\_Event, 172

TrackRecon.h  
  TrackRecon, 168  
  TrackRecon\_Event, 168

TrackRecon\_Event  
  TrackRecon.cxx, 172  
  TrackRecon.h, 168

trk\_len  
  ReconTools.cxx, 245  
  ReconTools.h, 226

Uploader, 140  
  ~Uploader, 141  
  Get\_a2, 142  
  Get\_a4, 142  
  Get\_b2, 142  
  Get\_b4, 142  
  Get\_Model\_Charge1D, 142  
  Get\_Model\_Electronics, 142  
  Get\_Model\_ReadOutGeometry, 142

Get\_NberOfEvent, 142  
 Get\_Norm, 142  
 Get\_PRF\_exist, 142  
 Get\_SampleFile, 142  
 GiveMe\_Event, 143  
 m\_a2, 143  
 m\_a4, 143  
 m\_b2, 143  
 m\_b4, 143  
 m\_NberOfEvent, 143  
 m\_Norm, 143  
 m\_PRF\_exist, 143  
 m\_SampleFile, 143  
 p\_Model\_Charge1D, 144  
 p\_Model\_Electronics, 144  
 p\_Model\_ReadOutGeometry, 144  
 p\_TFile, 144  
 p\_TTree, 144  
 Uploader, 141  
 Uploader Directory Reference, 20  
 Uploader/inc Directory Reference, 17  
 Uploader/inc/GiveMe\_Uploader.h, 246  
 Uploader/inc/Uploader.h, 246, 247  
 Uploader/inc/Uploader\_ERAM01.h, 248  
 Uploader/inc/Uploader\_MockUp\_V1.h, 248, 249  
 Uploader/inc/Uploader\_MockUp\_V2.h, 249, 250  
 Uploader/inc/Uploader\_Protoype.h, 251  
 Uploader/src Directory Reference, 19  
 Uploader/src/GiveMe\_Uploader.cxx, 252  
 Uploader/src/Uploader.cxx, 252  
 Uploader/src/Uploader\_ERAM01.cxx, 252  
 Uploader/src/Uploader\_MockUp\_V1.cxx, 253  
 Uploader/src/Uploader\_MockUp\_V2.cxx, 253  
 Uploader/src/Uploader\_Protoype.cxx, 253  
 Uploader\_ERAM01, 144  
   ~Uploader\_ERAM01, 145  
     GiveMe\_Event, 146  
     Uploader\_ERAM01, 145  
 Uploader\_MockUp\_V1, 146  
   ~Uploader\_MockUp\_V1, 147  
     GiveMe\_Event, 148  
     Uploader\_MockUp\_V1, 147  
 Uploader\_MockUp\_V2, 148  
   ~Uploader\_MockUp\_V2, 151  
     angle\_xy, 152  
     angle\_yz, 152  
     b\_angle\_xy, 152  
     b\_angle\_yz, 152  
     b\_beforeMerging, 152  
     b\_charge, 152  
     b\_clust\_pos, 152  
     b\_clust\_pos\_err, 152  
     b\_date, 152  
     b\_dEdx, 152  
     b\_dx, 152  
     b\_ev, 153  
     b\_max\_mult, 153  
     b\_mean\_mult, 153  
     b\_module, 153  
     b\_mom, 153  
     b\_multiplicity, 153  
     b\_offset, 153  
     b\_pad\_charge, 153  
     b\_pad\_time, 153  
     b\_pad\_wf\_q, 153  
     b\_pad\_x, 154  
     b\_pad\_y, 154  
     b\_qfrac, 154  
     b\_quality, 154  
     b\_residual, 154  
     b\_residual\_corr, 154  
     b\_rob\_clusters, 154  
     b\_sina, 154  
     b\_time, 154  
     b\_track, 154  
     b\_track\_pos, 155  
     b\_wf\_fwhm, 155  
     b\_wf\_width, 155  
     beforeMerging, 155  
     charge, 155  
     clust\_pos, 155  
     clust\_pos\_err, 155  
     date, 155  
     dEdx, 155  
     dx, 155  
     ev, 156  
     fCurrent, 156  
     GiveMe\_Event, 151  
     Init, 151  
     max\_mult, 156  
     mean\_mult, 156  
     module, 156  
     mom, 156  
     multiplicity, 156  
     offset, 156  
     pad\_charge, 156  
     pad\_time, 156  
     pad\_wf\_q, 157  
     pad\_x, 157  
     pad\_y, 157  
     qfrac, 157  
     quality, 157  
     residual, 157  
     residual\_corr, 157  
     rob\_clusters, 157  
     SetBeforeMerging, 151  
     Setwap\_XY, 151  
     sina, 157  
     time, 157  
     track, 158  
     track\_pos, 158  
     Uploader\_MockUp\_V2, 151  
     wf\_fwhm, 158  
     wf\_width, 158  
     Uploader\_Protoype, 158  
     ~Uploader\_Protoype, 159

GiveMe\_Event, 160  
    Uploader\_Prototype, 159

Utilities Directory Reference, 20

Utilities/inc Directory Reference, 18

Utilities/inc/FuncFromTGraph.h, 253

Utilities/inc/GaussFunction.h, 254

Utilities/inc/Misc.h, 254, 255

Utilities/inc/ParabolaFunction.h, 255, 256

Utilities/inc/ParabolaFunctionNG.h, 256, 257

Utilities/inc/SignalTools.h, 257, 258

Utilities/src Directory Reference, 20

Utilities/src/FuncFromTGraph.hxx, 258

Utilities/src/GaussFunction.hxx, 258

Utilities/src/ParabolaFunction.hxx, 258

Utilities/src/ParabolaFunctionNG.hxx, 259

Utilities/src/SignalTools.hxx, 259

V\_Cluster  
    Track, 137

v\_clusters  
    Reconstruction::RecoModule, 108

v\_comments  
    Reconstruction, 31

v\_datafiles  
    Reconstruction, 31

V\_ExisFlag2D  
    ROBoard, 115

v\_modules  
    Reconstruction::RecoEvent, 105

v\_modules\_position  
    Reconstruction::RecoEvent, 105

V\_Pad  
    ROBoard, 115

v\_pads  
    Reconstruction::RecoCluster, 101

V\_Pull  
    Track, 138

V\_Residual  
    Track, 138

v\_rootout\_files  
    Reconstruction, 31

v\_scanspec  
    Reconstruction, 31

Validate  
    Cluster, 40  
    Event, 60  
    Module, 86  
    Pad, 94

Validate\_ThisModule  
    Event, 60

Validity\_ForThisModule  
    Event, 60

vcorrFuncPaths  
    Reconstruction, 31

vScanLabels  
    Reconstruction, 31

vScanVals  
    Reconstruction, 31

vTags

    Reconstruction, 31

WF\_DoClosure  
    Pad, 94

wf\_fwhm  
    Uploader\_MockUp\_V2, 158

wf\_width  
    Uploader\_MockUp\_V2, 158

WFCorrection  
    Reconstruction, 24

WidthFromDriftDistance  
    Model\_Charge1, 76

WriteOut  
    Cluster, 40  
    Event, 60  
    Model\_Charge0D, 70  
    Model\_Charge1D, 73  
    Model\_Charge1, 76  
    Model\_Electronics, 78  
    Model\_ReadOutGeometry, 82  
    Module, 86  
    Pad, 94  
    Sample, 119

xPad  
    Reconstruction::RecoPad, 111

XX  
    EramInfo, 54

Y\_Position  
    Track, 137

yCluster  
    Reconstruction::RecoCluster, 101

yPad  
    Reconstruction::RecoPad, 112

yWeight  
    Reconstruction::RecoCluster, 101

YY  
    EramInfo, 54