EOS

Generated by Doxygen 1.6.1

Mon Jan 26 11:41:42 2015

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Class Index **3**

2.1 Class Hierarchy 3

3 Class Index **11**

3.1 Class List 11

4 File Index **17**

4.1 File List 17

5 Namespace Documentation **21**

5.1 eos Namespace Reference 21

5.1.1 Typedef Documentation 27

5.1.1.1 AnalysisPtr 27

5.1.1.2 HistoryPtr 27

5.1.1.3 LogLikelihoodBlockPtr 27

5.1.1.4 LogPriorPtr 27

5.1.1.5 MutablePtr 27

5.1.1.6 ObservablePtr 27

5.1.1.7 ProposalFunctionPtr 27

5.1.1.8 WilsonPolynomial 27

5.1.2 Enumeration Type Documentation 27

5.1.2.1 Helicity 27

5.1.2.2 LogLevel 27

5.1.3 Function Documentation 28

5.1.3.1 apply 28

5.1.3.2 apply 28

5.1.3.3 apply 28

5.1.3.4 derivative . . . 28
5.1.3.5 derivative< 1u, deriv::TwoSided > . . . 28
5.1.3.6 derivative< 2u, deriv::TwoSided > . . . 28
5.1.3.7 destringify . . . 28
5.1.3.8 dilog . . . 28
5.1.3.9 divide . . . 28
5.1.3.10 dot . . . 28
5.1.3.11 estimate_cumulative_distribution . . . 28
5.1.3.12 evolve . . . 28
5.1.3.13 integrate . . . 29
5.1.3.14 integrate . . . 29
5.1.3.15 integrate . . . 29
5.1.3.16 lambda . . . 30
5.1.3.17 make_analysis . . . 30
5.1.3.18 make_concrete_observable_factory . . . 30
5.1.3.19 make_polynomial . . . 30
5.1.3.20 make_polynomial_ht_like_ratio . . . 30
5.1.3.21 make_polynomial_observable . . . 30
5.1.3.22 make_polynomial_ratio . . . 30
5.1.3.23 memoise . . . 31
5.1.3.24 mult . . . 31
5.1.3.25 number_of_memoisations . . . 31
5.1.3.26 operator∗ . . . 31
5.1.3.27 operator∗ . . . 31
5.1.3.28 operator∗ . . . 31
5.1.3.29 operator∗ . . . 31
5.1.3.30 operator+ . . . 31
5.1.3.31 operator+ . . . 32
5.1.3.32 operator+ . . . 32
5.1.3.33 operator- . . . 32
5.1.3.34 operator- . . . 32
5.1.3.35 operator<< . . . 32
5.1.3.36 operator<< . . . 32
5.1.3.37 operator<< . . . 32
5.1.3.38 operator<< . . . 32
5.1.3.39 operator<< . . . 32

5.1.3.40 operator>> . . . 32
5.1.3.41 operator^ . . . 32
5.1.3.42 power_of . . . 32
5.1.3.43 stringify . . . 32
5.1.3.44 stringify . . . 33
5.1.3.45 stringify . . . 33
5.1.3.46 stringify_container . . . 33
5.1.3.47 trilog . . . 33
5.2 eos::btovll Namespace Reference . . . 34
5.2.1 Function Documentation . . . 34
5.2.1.1 angular_coefficients_array . . . 34
5.2.1.2 array_to_angular_coefficients . . . 34
5.2.1.3 decay_width . . . 34
5.3 eos::components Namespace Reference . . . 35
5.4 eos::deriv Namespace Reference . . . 36
5.5 eos::hdf5 Namespace Reference . . . 37
5.5.1 Typedef Documentation . . . 37
5.5.1.1 hid_t . . . 37
5.5.1.2 TypePtr . . . 37
5.5.2 Function Documentation . . . 37
5.5.2.1 operator<< . . . 37
5.5.2.2 operator>> . . . 38
5.6 eos::impl Namespace Reference . . . 39
5.7 eos::implementation Namespace Reference . . . 40
5.8 eos::InstantiationPolicy Namespace Reference . . . 41
5.9 eos::proposal_functions Namespace Reference . . . 42
5.9.1 Typedef Documentation . . . 42
5.9.1.1 MetaRecord . . . 42
5.9.1.2 MetaType . . . 42
5.9.1.3 MultivariateProposalPtr . . . 43
5.9.2 Function Documentation . . . 43
5.9.2.1 meta_record . . . 43
5.9.2.2 meta_type . . . 43
5.9.2.3 random_index . . . 43
5.9.2.4 sliding_window . . . 43
5.10 ROOT Namespace Reference . . . 44

5.11 ROOT::Minuit2 Namespace Reference . . . 45
5.12 std Namespace Reference . . . 46
5.12.1 Detailed Description . . . 49

6 Class Documentation **51**
6.1 eos::AbsoluteTestObservable Struct Reference . . . 51
6.1.1 Constructor & Destructor Documentation . . . 51
6.1.1.1 AbsoluteTestObservable . . . 51
6.1.1.2 ∼AbsoluteTestObservable . . . 51
6.1.2 Member Function Documentation . . . 51
6.1.2.1 clone . . . 51
6.1.2.2 clone . . . 51
6.1.2.3 evaluate . . . 52
6.2 eos::Accumulator Class Reference . . . 53
6.2.1 Constructor & Destructor Documentation . . . 53
6.2.1.1 Accumulator . . . 53
6.2.1.2 ∼Accumulator . . . 53
6.2.2 Member Function Documentation . . . 53
6.2.2.1 operator+= . . . 53
6.2.2.2 value . . . 53
6.3 eos::proposal_functions::AdjacencyMatrix Class Reference . . . 54
6.3.1 Detailed Description . . . 54
6.3.2 Constructor & Destructor Documentation . . . 54
6.3.2.1 AdjacencyMatrix . . . 54
6.3.3 Member Function Documentation . . . 54
6.3.3.1 add . . . 54
6.3.3.2 indices . . . 54
6.3.3.3 jump . . . 54
6.3.3.4 number_of_clusters . . . 55
6.3.3.5 reserve . . . 55
6.3.3.6 state . . . 55
6.4 eos::btovll::Amplitudes Struct Reference . . . 56
6.4.1 Member Data Documentation . . . 56
6.4.1.1 a_long_left . . . 56
6.4.1.2 a_long_right . . . 56
6.4.1.3 a_par_left . . . 56
6.4.1.4 a_par_right . . . 56

6.4.1.5 a_perp_left . . . 56
6.4.1.6 a_perp_right . . . 56
6.4.1.7 a_timelike . . . 56
6.5 eos::Analysis Class Reference . . . 57
6.5.1 Constructor & Destructor Documentation . . . 58
6.5.1.1 Analysis . . . 58
6.5.1.2 ∼Analysis . . . 58
6.5.2 Member Function Documentation . . . 58
6.5.2.1 add . . . 58
6.5.2.2 clone . . . 58
6.5.2.3 dump_descriptions . . . 58
6.5.2.4 goodness_of_fit . . . 59
6.5.2.5 log_likelihood . . . 59
6.5.2.6 log_posterior . . . 59
6.5.2.7 log_prior . . . 59
6.5.2.8 log_prior . . . 59
6.5.2.9 nuisance . . . 59
6.5.2.10 operator[] . . . 59
6.5.2.11 optimize . . . 60
6.5.2.12 optimize_minuit . . . 60
6.5.2.13 parameter_descriptions . . . 60
6.5.2.14 parameters . . . 60
6.5.2.15 read_descriptions . . . 60
6.5.2.16 restrict . . . 60
6.5.3 Friends And Related Function Documentation . . . 61
6.5.3.1 Implementation< Analysis > . . . 61
6.6 eos::AnalyticFormFactorBToPiDKMMO2008 Class Reference . . . 62
6.6.1 Constructor & Destructor Documentation . . . 62
6.6.1.1 AnalyticFormFactorBToPiDKMMO2008 . . . 62
6.6.1.2 ∼AnalyticFormFactorBToPiDKMMO2008 . . . 62
6.6.2 Member Function Documentation . . . 62
6.6.2.1 diagnostics . . . 62
6.6.2.2 f_0 . . . 62
6.6.2.3 F_lo_tw2 . . . 63
6.6.2.4 F_lo_tw3 . . . 63
6.6.2.5 F_lo_tw4 . . . 63

6.6.2.6 F_nlo_tw2 63
6.6.2.7 F_nlo_tw3 63
6.6.2.8 f_p 63
6.6.2.9 f_t 63
6.6.2.10 make 63
6.6.2.11 MB_lcsr 63
6.6.2.12 MB_svz 63
6.7 eos::btovll::AngularCoefficients Struct Reference 64
6.7.1 Member Data Documentation 64
6.7.1.1 j1c 64
6.7.1.2 j1s 64
6.7.1.3 j2c 64
6.7.1.4 j2s 64
6.7.1.5 j3 64
6.7.1.6 j4 64
6.7.1.7 j5 64
6.7.1.8 j6c 64
6.7.1.9 j6s 64
6.7.1.10 j7 64
6.7.1.11 j8 64
6.7.1.12 j9 64
6.8 eos::hdf5::Array< rank_, T_ > Class Template Reference 65
6.8.1 Member Typedef Documentation 65
6.8.1.1 Type 65
6.8.2 Constructor & Destructor Documentation 65
6.8.2.1 Array 65
6.8.3 Member Function Documentation 65
6.8.3.1 copy_from_hdf5 65
6.8.3.2 copy_to_hdf5 65
6.8.3.3 name 65
6.8.3.4 size 66
6.8.3.5 type_id 66
6.9 eos::hdf5::Attribute< T_ > Class Template Reference 67
6.9.1 Member Typedef Documentation 67
6.9.1.1 RecordType 67
6.9.2 Member Function Documentation 67

6.9.2.1 operator= . . . 67
6.9.2.2 value . . . 67
6.9.3 Friends And Related Function Documentation . . . 67
6.9.3.1 DataSet . . . 67
6.10 eos::hdf5::AttributeHandle Class Reference . . . 68
6.10.1 Constructor & Destructor Documentation . . . 68
6.10.1.1 ~AttributeHandle . . . 68
6.10.2 Member Function Documentation . . . 68
6.10.2.1 attribute_id . . . 68
6.10.2.2 read . . . 68
6.10.2.3 type_id . . . 68
6.10.2.4 write . . . 68
6.10.3 Friends And Related Function Documentation . . . 68
6.10.3.1 DataSetHandle . . . 68
6.11 eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ > Struct Template Reference . . . 69
6.11.1 Constructor & Destructor Documentation . . . 69
6.11.1.1 ~Base . . . 69
6.11.2 Member Function Documentation . . . 69
6.11.2.1 clone . . . 69
6.11.2.2 equal . . . 69
6.11.2.3 increment . . . 69
6.11.2.4 pointer . . . 70
6.11.2.5 reference . . . 70
6.11.2.6 underlying_iterator_ptr . . . 70
6.11.2.7 underlying_iterator_ptr . . . 70
6.12 eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ > Struct Template Reference . . . 71
6.12.1 Constructor & Destructor Documentation . . . 71
6.12.1.1 BaseImpl . . . 71
6.12.2 Member Function Documentation . . . 71
6.12.2.1 clone . . . 71
6.12.2.2 equal . . . 71
6.12.2.3 increment . . . 71
6.12.2.4 pointer . . . 72
6.12.2.5 reference . . . 72
6.12.2.6 underlying_iterator_ptr . . . 72

6.12.2.7 underlying_iterator_ptr 72
6.12.3 Member Data Documentation 72
6.12.3.1 i 72
6.13 eos::BCL2008FormFactors< Process_ > Class Template Reference 73
6.13.1 Constructor & Destructor Documentation 73
6.13.1.1 BCL2008FormFactors 73
6.13.2 Member Function Documentation 73
6.13.2.1 f_0 73
6.13.2.2 f_p 73
6.13.2.3 f_t 73
6.13.2.4 make 73
6.14 eos::BFW2010FormFactors< PToP > Class Template Reference 74
6.14.1 Constructor & Destructor Documentation 74
6.14.1.1 BFW2010FormFactors 74
6.14.2 Member Function Documentation 74
6.14.2.1 f_0 74
6.14.2.2 f_p 74
6.14.2.3 f_t 74
6.14.2.4 make 74
6.15 eos::Histogram< 2 >::Bin Struct Reference 75
6.15.1 Detailed Description 75
6.15.2 Constructor & Destructor Documentation 75
6.15.2.1 Bin 75
6.15.2.2 ~Bin 75
6.15.3 Member Function Documentation 76
6.15.3.1 operator< 76
6.15.4 Member Data Documentation 76
6.15.4.1 lower 76
6.15.4.2 upper 76
6.15.4.3 value 76
6.16 eos::Histogram< 1 >::Bin Struct Reference 77
6.16.1 Detailed Description 77
6.16.2 Constructor & Destructor Documentation 77
6.16.2.1 Bin 77
6.16.2.2 ~Bin 77
6.16.3 Member Function Documentation 78

6.16.3.1 operator< . . . 78
6.16.4 Member Data Documentation . . . 78
6.16.4.1 lower . . . 78
6.16.4.2 upper . . . 78
6.16.4.3 value . . . 78
6.17 eos::proposal_functions::BlockDecomposition Class Reference . . . 79
6.17.1 Member Typedef Documentation . . . 80
6.17.1.1 PriorsType . . . 80
6.17.2 Constructor & Destructor Documentation . . . 80
6.17.2.1 BlockDecomposition . . . 80
6.17.2.2 ~BlockDecomposition . . . 80
6.17.3 Member Function Documentation . . . 80
6.17.3.1 adapt . . . 80
6.17.3.2 add . . . 80
6.17.3.3 add . . . 80
6.17.3.4 clone . . . 80
6.17.3.5 dump_state . . . 80
6.17.3.6 evaluate . . . 81
6.17.3.7 priors_type . . . 81
6.17.3.8 propose . . . 81
6.17.4 Friends And Related Function Documentation . . . 81
6.17.4.1 MultivariateAccess . . . 81
6.18 eos::BMesonProperties Class Reference . . . 82
6.18.1 Constructor & Destructor Documentation . . . 82
6.18.1.1 BMesonProperties . . . 82
6.18.1.2 ~BMesonProperties . . . 82
6.18.2 Member Function Documentation . . . 82
6.18.2.1 mass_splitting_j1_j0 . . . 82
6.19 eos::Bremsstrahlung Struct Reference . . . 83
6.19.1 Member Function Documentation . . . 84
6.19.1.1 Deltai_23 . . . 84
6.19.1.2 Deltai_27 . . . 84
6.19.1.3 G_0 . . . 84
6.19.1.4 G_m1 . . . 84
6.19.1.5 itau_22 . . . 84
6.19.1.6 itau_27 . . . 84

6.19.1.7 itau_28 . . . 84
6.19.1.8 itau_29 . . . 84
6.19.1.9 tau_22 . . . 84
6.19.1.10 tau_27 . . . 84
6.19.1.11 tau_28 . . . 84
6.19.1.12 tau_29 . . . 84
6.19.1.13 tau_78 . . . 84
6.19.1.14 tau_88 . . . 84
6.19.1.15 tau_89 . . . 84
6.20 eos::BsToPhi Struct Reference . . . 85
6.20.1 Member Data Documentation . . . 85
6.20.1.1 mB . . . 85
6.20.1.2 mV . . . 85
6.21 eos::BToDilepton Class Reference . . . 86
6.21.1 Detailed Description . . . 86
6.21.2 Constructor & Destructor Documentation . . . 86
6.21.2.1 BToDilepton . . . 86
6.21.2.2 ∼BToDilepton . . . 86
6.21.3 Member Function Documentation . . . 86
6.21.3.1 branching_ratio_time_zero . . . 86
6.21.3.2 branching_ratio_untagged_integrated . . . 86
6.22 eos::BToK Struct Reference . . . 87
6.22.1 Member Typedef Documentation . . . 87
6.22.1.1 Transition . . . 87
6.22.2 Member Data Documentation . . . 87
6.22.2.1 label . . . 87
6.22.2.2 m2_Br0m . . . 87
6.22.2.3 m2_Br1m . . . 87
6.22.2.4 m_B . . . 87
6.22.2.5 m_P . . . 87
6.22.2.6 tau_m . . . 87
6.22.2.7 tau_p . . . 87
6.23 eos::BToKDilepton< LargeRecoil > Class Template Reference . . . 88
6.23.1 Constructor & Destructor Documentation . . . 89
6.23.1.1 BToKDilepton . . . 89
6.23.1.2 ∼BToKDilepton . . . 89

6.23.2 Member Function Documentation 89
6.23.2.1 a_l 89
6.23.2.2 b_l 89
6.23.2.3 c_l 89
6.23.2.4 differential_branching_ratio 89
6.23.2.5 differential_flat_term 89
6.23.2.6 differential_forward_backward_asymmetry 89
6.23.2.7 differential_ratio_muons_electrons 89
6.23.2.8 F_A 89
6.23.2.9 F_P 89
6.23.2.10 F_S 89
6.23.2.11 F_T 89
6.23.2.12 F_T5 89
6.23.2.13 F_V 89
6.23.2.14 integrated_branching_ratio 89
6.23.2.15 integrated_branching_ratio_cp_averaged 89
6.23.2.16 integrated_cp_asymmetry 89
6.23.2.17 integrated_decay_width 89
6.23.2.18 integrated_flat_term 89
6.23.2.19 integrated_flat_term_cp_averaged 89
6.23.2.20 integrated_forward_backward_asymmetry 89
6.23.2.21 integrated_forward_backward_asymmetry_cp_averaged 89
6.23.2.22 integrated_ratio_muons_electrons 89
6.23.2.23 two_differential_decay_width 89
6.24 eos::BToKDilepton< LowRecoil > Class Template Reference 91
6.24.1 Constructor & Destructor Documentation 92
6.24.1.1 BToKDilepton 92
6.24.1.2 ~BToKDilepton 92
6.24.2 Member Function Documentation 92
6.24.2.1 a_l 92
6.24.2.2 b_l 92
6.24.2.3 c_l 92
6.24.2.4 differential_branching_ratio 92
6.24.2.5 differential_flat_term 92
6.24.2.6 differential_forward_backward_asymmetry 92
6.24.2.7 differential_ratio_muons_electrons 92

6.24.2.8 imag_c7eff . . . 92
6.24.2.9 imag_c9eff . . . 92
6.24.2.10 integrated_branching_ratio . . . 92
6.24.2.11 integrated_branching_ratio_cp_averaged . . . 92
6.24.2.12 integrated_cp_asymmetry . . . 92
6.24.2.13 integrated_decay_width . . . 92
6.24.2.14 integrated_flat_term . . . 92
6.24.2.15 integrated_flat_term_cp_averaged . . . 92
6.24.2.16 integrated_forward_backward_asymmetry . . . 92
6.24.2.17 integrated_forward_backward_asymmetry_cp_averaged . . . 92
6.24.2.18 integrated_ratio_muons_electrons . . . 92
6.24.2.19 real_c7eff . . . 92
6.24.2.20 real_c9eff . . . 92
6.24.2.21 two_differential_decay_width . . . 92
6.25 eos::BToKstar Struct Reference . . . 94
6.25.1 Member Data Documentation . . . 94
6.25.1.1 mB . . . 94
6.25.1.2 mV . . . 94
6.26 eos::BToKstarDilepton< LargeRecoil > Class Template Reference . . . 95
6.26.1 Constructor & Destructor Documentation . . . 97
6.26.1.1 BToKstarDilepton . . . 97
6.26.1.2 ~BToKstarDilepton . . . 97
6.26.2 Member Function Documentation . . . 97
6.26.2.1 a_fb_zero_crossing . . . 97
6.26.2.2 a_long . . . 97
6.26.2.3 a_par . . . 97
6.26.2.4 a_perp . . . 97
6.26.2.5 differential_branching_ratio . . . 97
6.26.2.6 differential_decay_width . . . 97
6.26.2.7 differential_forward_backward_asymmetry . . . 97
6.26.2.8 differential_h_1 . . . 97
6.26.2.9 differential_h_2 . . . 97
6.26.2.10 differential_h_3 . . . 97
6.26.2.11 differential_h_4 . . . 97
6.26.2.12 differential_h_5 . . . 97
6.26.2.13 differential_isospin_asymmetry . . . 97

6.26.2.14 differential_j_1c . . . 97
6.26.2.15 differential_j_1s . . . 97
6.26.2.16 differential_j_2c . . . 97
6.26.2.17 differential_j_2s . . . 97
6.26.2.18 differential_j_3 . . . 97
6.26.2.19 differential_j_3_normalized . . . 97
6.26.2.20 differential_j_3_normalized_cp_averaged . . . 97
6.26.2.21 differential_j_4 . . . 97
6.26.2.22 differential_j_5 . . . 97
6.26.2.23 differential_j_6c . . . 97
6.26.2.24 differential_j_6s . . . 97
6.26.2.25 differential_j_7 . . . 97
6.26.2.26 differential_j_8 . . . 97
6.26.2.27 differential_j_9 . . . 97
6.26.2.28 differential_j_9_normalized . . . 97
6.26.2.29 differential_j_9_normalized_cp_averaged . . . 97
6.26.2.30 differential_longitudinal_polarisation . . . 97
6.26.2.31 differential_p_prime_4 . . . 97
6.26.2.32 differential_p_prime_5 . . . 97
6.26.2.33 differential_p_prime_6 . . . 97
6.26.2.34 differential_transversal_polarisation . . . 97
6.26.2.35 differential_transverse_asymmetry_2 . . . 97
6.26.2.36 differential_transverse_asymmetry_3 . . . 97
6.26.2.37 differential_transverse_asymmetry_4 . . . 97
6.26.2.38 differential_transverse_asymmetry_5 . . . 97
6.26.2.39 differential_transverse_asymmetry_im . . . 97
6.26.2.40 differential_transverse_asymmetry_re . . . 97
6.26.2.41 four_differential_decay_width . . . 97
6.26.2.42 integrated_a_9 . . . 97
6.26.2.43 integrated_branching_ratio . . . 97
6.26.2.44 integrated_branching_ratio_cp_averaged . . . 97
6.26.2.45 integrated_cp_asymmetry . . . 97
6.26.2.46 integrated_decay_width . . . 97
6.26.2.47 integrated_forward_backward_asymmetry . . . 97
6.26.2.48 integrated_forward_backward_asymmetry_cp_averaged . . . 97
6.26.2.49 integrated_h_1 . . . 97

6.26.2.50 integrated_h_2 97
6.26.2.51 integrated_h_3 97
6.26.2.52 integrated_h_4 97
6.26.2.53 integrated_h_5 97
6.26.2.54 integrated_isospin_asymmetry 97
6.26.2.55 integrated_j_1c 97
6.26.2.56 integrated_j_1s 97
6.26.2.57 integrated_j_2c 97
6.26.2.58 integrated_j_2s 97
6.26.2.59 integrated_j_3 97
6.26.2.60 integrated_j_3_normalized 97
6.26.2.61 integrated_j_3_normalized_cp_averaged 97
6.26.2.62 integrated_j_4 97
6.26.2.63 integrated_j_5 97
6.26.2.64 integrated_j_6c 97
6.26.2.65 integrated_j_6s 97
6.26.2.66 integrated_j_7 97
6.26.2.67 integrated_j_8 97
6.26.2.68 integrated_j_9 97
6.26.2.69 integrated_j_9_normalized 97
6.26.2.70 integrated_j_9_normalized_cp_averaged 97
6.26.2.71 integrated_longitudinal_polarisation 97
6.26.2.72 integrated_longitudinal_polarisation_cp_averaged 97
6.26.2.73 integrated_p_prime_4 97
6.26.2.74 integrated_p_prime_5 97
6.26.2.75 integrated_p_prime_6 97
6.26.2.76 integrated_transversal_polarisation 97
6.26.2.77 integrated_transversal_polarisation_cp_averaged 97
6.26.2.78 integrated_transverse_asymmetry_2 97
6.26.2.79 integrated_transverse_asymmetry_2_cp_averaged 97
6.26.2.80 integrated_transverse_asymmetry_3 97
6.26.2.81 integrated_transverse_asymmetry_4 97
6.26.2.82 integrated_transverse_asymmetry_5 97
6.26.2.83 integrated_transverse_asymmetry_im 97
6.26.2.84 integrated_transverse_asymmetry_re 97
6.26.2.85 xi_para 97

6.26.2.86 xi_perp . . . 97
6.27 eos::BToKstarDilepton< LowRecoil > Class Template Reference . . . 99
6.27.1 Constructor & Destructor Documentation . . . 102
6.27.1.1 BToKstarDilepton . . . 102
6.27.1.2 ~BToKstarDilepton . . . 102
6.27.2 Member Function Documentation . . . 102
6.27.2.1 a_long . . . 102
6.27.2.2 a_par . . . 102
6.27.2.3 a_perp . . . 102
6.27.2.4 differential_branching_ratio . . . 102
6.27.2.5 differential_cp_asymmetry_1 . . . 102
6.27.2.6 differential_cp_asymmetry_2 . . . 102
6.27.2.7 differential_cp_asymmetry_3 . . . 102
6.27.2.8 differential_cp_asymmetry_mix . . . 102
6.27.2.9 differential_decay_width . . . 102
6.27.2.10 differential_forward_backward_asymmetry . . . 102
6.27.2.11 differential_h_1 . . . 102
6.27.2.12 differential_h_2 . . . 102
6.27.2.13 differential_h_3 . . . 102
6.27.2.14 differential_h_4 . . . 102
6.27.2.15 differential_h_5 . . . 102
6.27.2.16 differential_j_1c . . . 102
6.27.2.17 differential_j_1s . . . 102
6.27.2.18 differential_j_2c . . . 102
6.27.2.19 differential_j_2s . . . 102
6.27.2.20 differential_j_3 . . . 102
6.27.2.21 differential_j_3_normalized . . . 102
6.27.2.22 differential_j_3_normalized_cp_averaged . . . 102
6.27.2.23 differential_j_4 . . . 102
6.27.2.24 differential_j_5 . . . 102
6.27.2.25 differential_j_6c . . . 102
6.27.2.26 differential_j_6s . . . 102
6.27.2.27 differential_j_7 . . . 102
6.27.2.28 differential_j_8 . . . 102
6.27.2.29 differential_j_9 . . . 102
6.27.2.30 differential_j_9_normalized . . . 102

6.27.2.31 differential_j_9_normalized_cp_averaged 102
6.27.2.32 differential_longitudinal_polarisation 102
6.27.2.33 differential_p_prime_4 102
6.27.2.34 differential_p_prime_5 102
6.27.2.35 differential_p_prime_6 102
6.27.2.36 differential_transversal_polarisation 102
6.27.2.37 differential_transverse_asymmetry_2 102
6.27.2.38 differential_transverse_asymmetry_3 102
6.27.2.39 differential_transverse_asymmetry_4 102
6.27.2.40 differential_transverse_asymmetry_5 102
6.27.2.41 differential_transverse_asymmetry_im 102
6.27.2.42 differential_transverse_asymmetry_re 102
6.27.2.43 four_differential_decay_width 102
6.27.2.44 imag_c7eff 102
6.27.2.45 imag_c9eff 102
6.27.2.46 imag_y 102
6.27.2.47 integrated_a_9 102
6.27.2.48 integrated_branching_ratio 102
6.27.2.49 integrated_branching_ratio_cp_averaged 102
6.27.2.50 integrated_cp_asymmetry 102
6.27.2.51 integrated_cp_asymmetry_1 102
6.27.2.52 integrated_cp_asymmetry_2 102
6.27.2.53 integrated_cp_asymmetry_3 102
6.27.2.54 integrated_cp_summed_decay_width 102
6.27.2.55 integrated_decay_width 102
6.27.2.56 integrated_forward_backward_asymmetry 102
6.27.2.57 integrated_forward_backward_asymmetry_cp_averaged 102
6.27.2.58 integrated_forward_backward_asymmetry_naive 102
6.27.2.59 integrated_h_1 102
6.27.2.60 integrated_h_1_naive 102
6.27.2.61 integrated_h_2 102
6.27.2.62 integrated_h_2_naive 102
6.27.2.63 integrated_h_3 102
6.27.2.64 integrated_h_3_naive 102
6.27.2.65 integrated_h_4 102
6.27.2.66 integrated_h_5 102

6.27.2.67 integrated_j_1c . . . 102
6.27.2.68 integrated_j_1s . . . 102
6.27.2.69 integrated_j_2c . . . 102
6.27.2.70 integrated_j_2s . . . 102
6.27.2.71 integrated_j_3 . . . 102
6.27.2.72 integrated_j_3_normalized . . . 102
6.27.2.73 integrated_j_3_normalized_cp_averaged . . . 102
6.27.2.74 integrated_j_4 . . . 102
6.27.2.75 integrated_j_5 . . . 102
6.27.2.76 integrated_j_6c . . . 102
6.27.2.77 integrated_j_6s . . . 102
6.27.2.78 integrated_j_7 . . . 102
6.27.2.79 integrated_j_8 . . . 102
6.27.2.80 integrated_j_9 . . . 102
6.27.2.81 integrated_j_9_normalized . . . 102
6.27.2.82 integrated_j_9_normalized_cp_averaged . . . 102
6.27.2.83 integrated_longitudinal_polarisation . . . 102
6.27.2.84 integrated_longitudinal_polarisation_cp_averaged . . . 102
6.27.2.85 integrated_longitudinal_polarisation_naive . . . 102
6.27.2.86 integrated_p_prime_4 . . . 102
6.27.2.87 integrated_p_prime_5 . . . 102
6.27.2.88 integrated_p_prime_6 . . . 102
6.27.2.89 integrated_transversal_polarisation . . . 102
6.27.2.90 integrated_transversal_polarisation_cp_averaged . . . 102
6.27.2.91 integrated_transverse_asymmetry_2 . . . 102
6.27.2.92 integrated_transverse_asymmetry_2_cp_averaged . . . 102
6.27.2.93 integrated_transverse_asymmetry_2_naive . . . 102
6.27.2.94 integrated_transverse_asymmetry_3 . . . 102
6.27.2.95 integrated_transverse_asymmetry_3_naive . . . 102
6.27.2.96 integrated_transverse_asymmetry_4 . . . 102
6.27.2.97 integrated_transverse_asymmetry_4_naive . . . 102
6.27.2.98 integrated_transverse_asymmetry_5 . . . 102
6.27.2.99 integrated_transverse_asymmetry_im . . . 102
6.27.2.100 integrated_transverse_asymmetry_re . . . 102
6.27.2.101 integrated_unnormalized_cp_asymmetry_1 . . . 102
6.27.2.102 integrated_unnormalized_forward_backward_asymmetry . . . 102

6.27.2.103 real_c7eff . . . 102
6.27.2.104 real_c9eff . . . 102
6.27.2.105 real_y . . . 102
6.27.2.106 rho_1 . . . 102
6.27.2.107 rho_2 . . . 102
6.28 eos::BToKstarGamma Class Reference . . . 104
6.28.1 Detailed Description . . . 104
6.28.2 Constructor & Destructor Documentation . . . 104
6.28.2.1 BToKstarGamma . . . 104
6.28.2.2 ~BToKstarGamma . . . 104
6.28.3 Member Function Documentation . . . 104
6.28.3.1 branching_ratio . . . 104
6.28.3.2 branching_ratio_cp_averaged . . . 104
6.28.3.3 c_kstar_gamma . . . 105
6.28.3.4 cp_asymmetry . . . 105
6.28.3.5 isospin_asymmetry . . . 105
6.28.3.6 s_kstar_gamma . . . 105
6.29 eos::BToPi Struct Reference . . . 106
6.29.1 Member Typedef Documentation . . . 106
6.29.1.1 Transition . . . 106
6.29.2 Member Data Documentation . . . 106
6.29.2.1 label . . . 106
6.29.2.2 m2_Br0m . . . 106
6.29.2.3 m2_Br1m . . . 106
6.29.2.4 m_B . . . 106
6.29.2.5 m_P . . . 106
6.29.2.6 tau_m . . . 106
6.29.2.7 tau_p . . . 106
6.30 eos::BToPiLeptonNeutrino Class Reference . . . 107
6.30.1 Constructor & Destructor Documentation . . . 107
6.30.1.1 BToPiLeptonNeutrino . . . 107
6.30.1.2 ~BToPiLeptonNeutrino . . . 107
6.30.2 Member Function Documentation . . . 107
6.30.2.1 differential_branching_ratio . . . 107
6.30.2.2 differential_zeta . . . 107
6.30.2.3 integrated_branching_ratio . . . 107

6.30.2.4 integrated_zeta . . . 107
6.31 eos::BToS Struct Reference . . . 108
6.32 eos::BToXsDilepton< HLMW2005 > Class Template Reference . . . 109
6.32.1 Constructor & Destructor Documentation . . . 109
6.32.1.1 BToXsDilepton . . . 109
6.32.1.2 ~BToXsDilepton . . . 109
6.32.2 Member Function Documentation . . . 109
6.32.2.1 diagnostics . . . 109
6.32.2.2 differential_branching_ratio . . . 109
6.32.2.3 integrated_branching_ratio . . . 109
6.33 eos::BToXsGamma< Minimal > Class Template Reference . . . 110
6.33.1 Constructor & Destructor Documentation . . . 110
6.33.1.1 BToXsGamma . . . 110
6.33.1.2 ~BToXsGamma . . . 110
6.33.2 Member Function Documentation . . . 110
6.33.2.1 integrated_branching_ratio . . . 110
6.34 eos::BToXsGamma< NLO > Class Template Reference . . . 111
6.34.1 Constructor & Destructor Documentation . . . 111
6.34.1.1 BToXsGamma . . . 111
6.34.1.2 ~BToXsGamma . . . 111
6.34.2 Member Function Documentation . . . 111
6.34.2.1 diagnostics . . . 111
6.34.2.2 integrated_branching_ratio . . . 111
6.34.2.3 photon_energy_moment_1 . . . 111
6.34.2.4 photon_energy_moment_2 . . . 111
6.35 eos::BZ2004FormFactors< Process_, PToP > Class Template Reference . . . 112
6.35.1 Constructor & Destructor Documentation . . . 112
6.35.1.1 BZ2004FormFactors . . . 112
6.35.2 Member Function Documentation . . . 112
6.35.2.1 f_0 . . . 112
6.35.2.2 f_p . . . 112
6.35.2.3 f_t . . . 112
6.35.2.4 make . . . 112
6.36 eos::BZ2004FormFactors< Process_, PToV > Class Template Reference . . . 113
6.36.1 Constructor & Destructor Documentation . . . 113
6.36.1.1 BZ2004FormFactors . . . 113

6.36.2 Member Function Documentation . . . 113
6.36.2.1 a_0 . . . 113
6.36.2.2 a_1 . . . 113
6.36.2.3 a_12 . . . 113
6.36.2.4 a_2 . . . 113
6.36.2.5 make . . . 114
6.36.2.6 v . . . 114
6.37 eos::BZ2004FormFactorsSplit< Process_ > Class Template Reference . . . 115
6.37.1 Constructor & Destructor Documentation . . . 115
6.37.1.1 BZ2004FormFactorsSplit . . . 115
6.37.2 Member Function Documentation . . . 115
6.37.2.1 f_0 . . . 115
6.37.2.2 f_p . . . 115
6.37.2.3 f_t . . . 115
6.37.2.4 make . . . 115
6.38 eos::CartesianProduct< T_ > Class Template Reference . . . 116
6.38.1 Detailed Description . . . 116
6.38.2 Member Typedef Documentation . . . 116
6.38.2.1 Iterator . . . 116
6.38.3 Constructor & Destructor Documentation . . . 116
6.38.3.1 CartesianProduct . . . 116
6.38.4 Member Function Documentation . . . 117
6.38.4.1 begin . . . 117
6.38.4.2 end . . . 117
6.38.4.3 over . . . 117
6.38.4.4 size . . . 117
6.39 eos::CharmLoops Struct Reference . . . 118
6.39.1 Member Function Documentation . . . 120
6.39.1.1 A . . . 120
6.39.1.2 B . . . 120
6.39.1.3 B0 . . . 120
6.39.1.4 C . . . 120
6.39.1.5 C0 . . . 120
6.39.1.6 F17_massive . . . 120
6.39.1.7 F17_massless . . . 120
6.39.1.8 F19_massive . . . 120

6.39.1.9 F19_massless . . . 120
6.39.1.10 F27_massive . . . 120
6.39.1.11 F27_massless . . . 120
6.39.1.12 F29_massive . . . 120
6.39.1.13 F29_massless . . . 120
6.39.1.14 F87_massless . . . 120
6.39.1.15 F89_massless . . . 120
6.39.1.16 h . . . 120
6.39.1.17 h . . . 120
6.40 eos::ChiSquared Struct Reference . . . 122
6.40.1 Member Typedef Documentation . . . 122
6.40.1.1 Function . . . 122
6.40.2 Member Function Documentation . . . 122
6.40.2.1 make . . . 122
6.40.2.2 with_combined_uncertainties . . . 122
6.40.2.3 with_theory_offset . . . 122
6.41 eos::CKMScanComponent Class Reference . . . 123
6.41.1 Constructor & Destructor Documentation . . . 123
6.41.1.1 CKMScanComponent . . . 123
6.41.2 Member Function Documentation . . . 123
6.41.2.1 ckm_cb . . . 123
6.41.2.2 ckm_cd . . . 123
6.41.2.3 ckm_cs . . . 123
6.41.2.4 ckm_tb . . . 123
6.41.2.5 ckm_td . . . 123
6.41.2.6 ckm_ts . . . 124
6.41.2.7 ckm_ub . . . 124
6.41.2.8 ckm_ud . . . 124
6.41.2.9 ckm_us . . . 124
6.42 eos::CKMScanModel Class Reference . . . 125
6.42.1 Constructor & Destructor Documentation . . . 125
6.42.1.1 CKMScanModel . . . 125
6.42.1.2 ∼CKMScanModel . . . 125
6.42.2 Member Function Documentation . . . 125
6.42.2.1 make . . . 125
6.43 eos::Cluster Class Reference . . . 126

6.43.1 Member Typedef Documentation 126
6.43.1.1 IndexIterator 126
6.43.1.2 Iterator 126
6.43.1.3 RValueFunction 126
6.43.2 Constructor & Destructor Documentation 126
6.43.2.1 Cluster 126
6.43.2.2 ∼Cluster 127
6.43.3 Member Function Documentation 127
6.43.3.1 add 127
6.43.3.2 begin 127
6.43.3.3 begin_indices 127
6.43.3.4 end 127
6.43.3.5 end_indices 127
6.43.3.6 mean 127
6.43.3.7 means 127
6.43.3.8 overlaps 127
6.43.3.9 parameter_indices 127
6.43.3.10 variances 128
6.44 eos::HierarchicalClustering::Component Class Reference 129
6.44.1 Detailed Description 129
6.44.2 Constructor & Destructor Documentation 129
6.44.2.1 Component 129
6.44.2.2 Component 129
6.44.2.3 ∼Component 129
6.44.3 Member Function Documentation 129
6.44.3.1 covariance 129
6.44.3.2 determinant 129
6.44.3.3 inverse_covariance 129
6.44.3.4 mean 129
6.44.3.5 weight 129
6.45 eos::hdf5::Composite< T_ > Class Template Reference 130
6.45.1 Member Typedef Documentation 130
6.45.1.1 Type 130
6.45.2 Constructor & Destructor Documentation 130
6.45.2.1 Composite 130
6.45.2.2 ∼Composite 130

6.45.3 Member Function Documentation 130
6.45.3.1 copy_from_hdf5 130
6.45.3.2 copy_to_hdf5 130
6.45.3.3 name 130
6.45.3.4 size 131
6.45.3.5 type_id 131
6.46 eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ > Struct Template Reference . 132
6.46.1 Member Function Documentation 132
6.46.1.1 copy_from_hdf5 132
6.46.1.2 copy_to_hdf5 132
6.47 eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ > Struct Template Reference 133
6.47.1 Member Function Documentation 133
6.47.1.1 copy_from_hdf5 133
6.47.1.2 copy_to_hdf5 133
6.48 eos::ConcreteObservable< Decay_, Args_ > Class Template Reference 134
6.48.1 Constructor & Destructor Documentation 134
6.48.1.1 ConcreteObservable 134
6.48.2 Member Function Documentation 134
6.48.2.1 clone 134
6.48.2.2 clone 134
6.48.2.3 evaluate 134
6.48.2.4 kinematics 135
6.48.2.5 name 135
6.48.2.6 options 135
6.48.2.7 parameters 135
6.49 eos::ConcreteObservableFactory< Decay_, Args_ > Class Template Reference 136
6.49.1 Constructor & Destructor Documentation 136
6.49.1.1 ConcreteObservableFactory 136
6.49.1.2 ~ConcreteObservableFactory 136
6.49.2 Member Function Documentation 136
6.49.2.1 make 136
6.50 eos::ConditionVariable Class Reference 137
6.50.1 Constructor & Destructor Documentation 137
6.50.1.1 ConditionVariable 137
6.50.1.2 ~ConditionVariable 137
6.50.2 Member Function Documentation 137

6.50.2.1 acquire_then_signal . . . 137
6.50.2.2 broadcast . . . 137
6.50.2.3 signal . . . 137
6.50.2.4 wait . . . 138
6.51 eos::MarkovChainSampler::Config Class Reference . . . 139
6.51.1 Detailed Description . . . 140
6.51.2 Member Function Documentation . . . 140
6.51.2.1 Default . . . 140
6.51.2.2 Quick . . . 140
6.51.3 Member Data Documentation . . . 141
6.51.3.1 adapt_iterations . . . 141
6.51.3.2 block_proposal_parameters . . . 141
6.51.3.3 chunk_size . . . 141
6.51.3.4 chunks . . . 141
6.51.3.5 find_modes . . . 141
6.51.3.6 global_local_config . . . 141
6.51.3.7 max_efficiency . . . 141
6.51.3.8 min_efficiency . . . 141
6.51.3.9 need_main_run . . . 141
6.51.3.10 need_prerun . . . 141
6.51.3.11 number_of_chains . . . 142
6.51.3.12 output_file . . . 142
6.51.3.13 parallelize . . . 142
6.51.3.14 partitions . . . 142
6.51.3.15 prerun_chains_per_partition . . . 142
6.51.3.16 prerun_iterations_max . . . 142
6.51.3.17 prerun_iterations_min . . . 142
6.51.3.18 prerun_iterations_update . . . 142
6.51.3.19 proposal . . . 142
6.51.3.20 rvalue_criterion_param . . . 142
6.51.3.21 rvalue_criterion_posterior . . . 143
6.51.3.22 scale_automatic . . . 143
6.51.3.23 scale_nuisance . . . 143
6.51.3.24 scale_reduction . . . 143
6.51.3.25 seed . . . 143
6.51.3.26 skip_initial . . . 143

6.51.3.27 store . . . 143
6.51.3.28 store_observables_and_proposals . . . 143
6.51.3.29 store_prerun . . . 143
6.51.3.30 student_t_degrees_of_freedom . . . 143
6.51.3.31 use_posterior_rvalue . . . 144
6.51.3.32 use_strict_rvalue_definition . . . 144
6.52 eos::EquationSolver::Config Struct Reference . . . 145
6.52.1 Member Function Documentation . . . 145
6.52.1.1 Default . . . 145
6.52.2 Member Data Documentation . . . 145
6.52.2.1 maximum_steps . . . 145
6.52.2.2 strategy . . . 145
6.52.2.3 tolerance . . . 145
6.53 eos::PopulationMonteCarloSampler::Config Struct Reference . . . 146
6.53.1 Detailed Description . . . 148
6.53.2 Member Function Documentation . . . 148
6.53.2.1 Default . . . 148
6.53.2.2 Quick . . . 148
6.53.3 Member Data Documentation . . . 148
6.53.3.1 adjust_sample_size . . . 148
6.53.3.2 block_decomposition . . . 148
6.53.3.3 chunk_size . . . 148
6.53.3.4 chunks . . . 148
6.53.3.5 component_means . . . 149
6.53.3.6 component_offset . . . 149
6.53.3.7 component_variances . . . 149
6.53.3.8 component_weights . . . 149
6.53.3.9 components_per_cluster . . . 149
6.53.3.10 convergence_eff_sample_size . . . 149
6.53.3.11 convergence_perplexity . . . 149
6.53.3.12 crop_highest_weights . . . 149
6.53.3.13 degrees_of_freedom . . . 149
6.53.3.14 final_chunk_size . . . 149
6.53.3.15 group_by_r_value . . . 150
6.53.3.16 ignore_eff_sample_size . . . 150
6.53.3.17 ignore_groups . . . 150

6.53.3.18 maximum_relative_std_deviation . . . 150
6.53.3.19 minimum_eff_sample_size . . . 150
6.53.3.20 minimum_overlap . . . 150
6.53.3.21 minimum_perplexity . . . 150
6.53.3.22 minimum_steps . . . 150
6.53.3.23 mode_distance . . . 151
6.53.3.24 need_prerun . . . 151
6.53.3.25 number_of_workers . . . 151
6.53.3.26 output_file . . . 151
6.53.3.27 override_global_local_proposal . . . 151
6.53.3.28 parallelize . . . 151
6.53.3.29 patch_around_local_mode . . . 151
6.53.3.30 print_steps . . . 151
6.53.3.31 r_value_no_nuisance . . . 151
6.53.3.32 random_start . . . 151
6.53.3.33 seed . . . 152
6.53.3.34 single_cluster . . . 152
6.53.3.35 skip_initial . . . 152
6.53.3.36 sliding_window . . . 152
6.53.3.37 starting_points . . . 152
6.53.3.38 std_dev_reduction . . . 152
6.53.3.39 store . . . 152
6.53.3.40 store_hc_initial . . . 152
6.53.3.41 store_input_components . . . 152
6.53.3.42 store_prerun . . . 152
6.53.3.43 super_clusters . . . 153
6.54 eos::PriorSampler::Config Struct Reference . . . 154
6.54.1 Detailed Description . . . 154
6.54.2 Member Function Documentation . . . 154
6.54.2.1 Default . . . 154
6.54.3 Member Data Documentation . . . 154
6.54.3.1 n_samples . . . 154
6.54.3.2 n_workers . . . 154
6.54.3.3 output_file . . . 155
6.54.3.4 parallelize . . . 155
6.54.3.5 seed . . . 155

6.54.3.6 store_parameters . . . 155
6.55 eos::proposal_functions::GlobalLocal::Config Struct Reference . . . 156
6.55.1 Member Enumeration Documentation . . . 156
6.55.1.1 HistoryPointWeighting . . . 156
6.55.2 Member Function Documentation . . . 157
6.55.2.1 Default . . . 157
6.55.3 Member Data Documentation . . . 157
6.55.3.1 clustering_maximum_r_value . . . 157
6.55.3.2 clustering_strict_r_value . . . 157
6.55.3.3 cooling_power . . . 157
6.55.3.4 equal_weight_components . . . 157
6.55.3.5 history_point_weighting . . . 157
6.55.3.6 history_points . . . 157
6.55.3.7 history_points_local_covariance_size . . . 158
6.55.3.8 history_points_ordered . . . 158
6.55.3.9 join_chains_symmetrically . . . 158
6.55.3.10 local_jump_probability . . . 158
6.55.3.11 long_jump_indices . . . 158
6.55.3.12 minimum_relative_cluster_weight . . . 158
6.55.3.13 perform_clustering . . . 158
6.55.3.14 rescale_local_covariance . . . 158
6.55.3.15 skip_initial . . . 159
6.56 eos::HierarchicalClustering::Config Class Reference . . . 160
6.56.1 Detailed Description . . . 160
6.56.2 Member Function Documentation . . . 160
6.56.2.1 Default . . . 160
6.56.2.2 Quick . . . 160
6.56.3 Member Data Documentation . . . 161
6.56.3.1 equal_weights . . . 161
6.56.3.2 kill_clusters . . . 161
6.56.3.3 maximum_steps . . . 161
6.56.3.4 precision . . . 161
6.57 eos::Constraint Class Reference . . . 162
6.57.1 Detailed Description . . . 162
6.57.2 Member Typedef Documentation . . . 163
6.57.2.1 BlockIterator . . . 163

6.57.2.2 ObservableIterator . . . 163
6.57.3 Constructor & Destructor Documentation . . . 163
6.57.3.1 Constraint . . . 163
6.57.3.2 ~Constraint . . . 163
6.57.4 Member Function Documentation . . . 163
6.57.4.1 begin_blocks . . . 163
6.57.4.2 begin_observables . . . 163
6.57.4.3 end_blocks . . . 163
6.57.4.4 end_observables . . . 163
6.57.4.5 make . . . 163
6.57.4.6 name . . . 163
6.58 eos::hdf5::DataSet< T_ > Class Template Reference . . . 164
6.58.1 Detailed Description . . . 165
6.58.2 Member Typedef Documentation . . . 165
6.58.2.1 RecordType . . . 165
6.58.3 Constructor & Destructor Documentation . . . 165
6.58.3.1 ~DataSet . . . 165
6.58.4 Member Function Documentation . . . 165
6.58.4.1 create_attribute . . . 165
6.58.4.2 create_or_open_attribute . . . 165
6.58.4.3 end . . . 165
6.58.4.4 name . . . 165
6.58.4.5 open_attribute . . . 165
6.58.4.6 operator[] . . . 165
6.58.4.7 records . . . 165
6.58.4.8 set_index . . . 166
6.58.5 Friends And Related Function Documentation . . . 166
6.58.5.1 hdf5::File . . . 166
6.58.5.2 operator<< . . . 166
6.58.5.3 operator>> . . . 166
6.59 eos::hdf5::DataSetHandle Class Reference . . . 167
6.59.1 Constructor & Destructor Documentation . . . 168
6.59.1.1 ~DataSetHandle . . . 168
6.59.2 Member Function Documentation . . . 168
6.59.2.1 create_attribute . . . 168
6.59.2.2 open_attribute . . . 168

6.59.2.3 read_one . . . 168
6.59.2.4 select . . . 168
6.59.2.5 set_id . . . 168
6.59.2.6 size . . . 168
6.59.2.7 space_id . . . 168
6.59.2.8 type_id . . . 168
6.59.2.9 write_one . . . 168
6.59.3 Friends And Related Function Documentation . . . 168
6.59.3.1 File . . . 168
6.60 eos::hdf5::DataType< T_ > Struct Template Reference . . . 169
6.60.1 Member Function Documentation . . . 169
6.60.1.1 type_id . . . 169
6.61 eos::hdf5::DataType< const char * > Struct Template Reference . . . 170
6.61.1 Member Function Documentation . . . 170
6.61.1.1 type_id . . . 170
6.62 eos::hdf5::DataType< double > Struct Template Reference . . . 171
6.62.1 Member Function Documentation . . . 171
6.62.1.1 type_id . . . 171
6.63 eos::hdf5::DataType< int > Struct Template Reference . . . 172
6.63.1 Member Function Documentation . . . 172
6.63.1.1 type_id . . . 172
6.64 eos::hdf5::DataType< short > Struct Template Reference . . . 173
6.64.1 Member Function Documentation . . . 173
6.64.1.1 type_id . . . 173
6.65 eos::hdf5::DataType< unsigned > Struct Template Reference . . . 174
6.65.1 Member Function Documentation . . . 174
6.65.1.1 type_id . . . 174
6.66 eos::InstantiationPolicy::DeleteOnDestruction< T_ > Class Template Reference . . . 175
6.67 eos::Diagnostics Class Reference . . . 176
6.67.1 Detailed Description . . . 176
6.67.2 Member Typedef Documentation . . . 177
6.67.2.1 Iterator . . . 177
6.67.3 Constructor & Destructor Documentation . . . 177
6.67.3.1 Diagnostics . . . 177
6.67.3.2 ~Diagnostics . . . 177
6.67.4 Member Function Documentation . . . 177

6.67.4.1 add . . . 177
6.67.4.2 begin . . . 177
6.67.4.3 end . . . 177
6.67.4.4 size . . . 177
6.68 eos::implementation::DoStringify< T_ > Struct Template Reference . . . 178
6.68.1 Member Function Documentation . . . 178
6.68.1.1 stringify . . . 178
6.69 eos::EMContributions Struct Reference . . . 179
6.69.1 Member Function Documentation . . . 179
6.69.1.1 omegaem_1010 . . . 179
6.69.1.2 omegaem_22 . . . 179
6.69.1.3 omegaem_27 . . . 179
6.69.1.4 omegaem_29 . . . 179
6.69.1.5 omegaem_77 . . . 179
6.69.1.6 omegaem_79 . . . 179
6.69.1.7 omegaem_99 . . . 179
6.70 eos::Diagnostics::Entry Struct Reference . . . 180
6.70.1 Detailed Description . . . 180
6.70.2 Member Data Documentation . . . 180
6.70.2.1 description . . . 180
6.70.2.2 value . . . 180
6.71 eos::EquationSolver Class Reference . . . 181
6.71.1 Member Typedef Documentation . . . 181
6.71.1.1 Equation . . . 181
6.71.2 Constructor & Destructor Documentation . . . 181
6.71.2.1 EquationSolver . . . 181
6.71.2.2 ∼EquationSolver . . . 181
6.71.3 Member Function Documentation . . . 181
6.71.3.1 add . . . 181
6.71.3.2 add . . . 181
6.71.3.3 add . . . 181
6.71.3.4 solve . . . 181
6.72 eos::Exception Class Reference . . . 183
6.72.1 Constructor & Destructor Documentation . . . 183
6.72.1.1 Exception . . . 183
6.72.1.2 ∼Exception . . . 183

6.72.2 Member Function Documentation . . . 183
6.72.2.1 what . . . 183
6.73 eos::proposal_functions::Factory Struct Reference . . . 184
6.73.1 Detailed Description . . . 184
6.73.2 Member Function Documentation . . . 184
6.73.2.1 make . . . 184
6.74 eos::hdf5::File Class Reference . . . 185
6.74.1 Constructor & Destructor Documentation . . . 185
6.74.1.1 ~File . . . 185
6.74.2 Member Function Documentation . . . 186
6.74.2.1 copy . . . 186
6.74.2.2 Create . . . 186
6.74.2.3 create_data_set . . . 186
6.74.2.4 create_group . . . 186
6.74.2.5 create_or_open_data_set . . . 186
6.74.2.6 Exists . . . 187
6.74.2.7 group_exists . . . 187
6.74.2.8 name . . . 187
6.74.2.9 number_of_objects . . . 187
6.74.2.10 Open . . . 187
6.74.2.11 open_data_set . . . 187
6.75 eos::hdf5::FileHandle Class Reference . . . 188
6.75.1 Constructor & Destructor Documentation . . . 188
6.75.1.1 ~FileHandle . . . 188
6.75.2 Member Function Documentation . . . 188
6.75.2.1 id . . . 188
6.75.2.2 read_only . . . 188
6.75.3 Friends And Related Function Documentation . . . 188
6.75.3.1 File . . . 188
6.76 eos::FormFactorAdapter< Transition_ > Class Template Reference . . . 189
6.76.1 Constructor & Destructor Documentation . . . 189
6.76.1.1 FormFactorAdapter . . . 189
6.76.2 Member Function Documentation . . . 189
6.76.2.1 clone . . . 189
6.76.2.2 clone . . . 189
6.76.2.3 evaluate . . . 189

6.76.2.4 kinematics . . . 190
6.76.2.5 name . . . 190
6.76.2.6 options . . . 190
6.76.2.7 parameters . . . 190
6.77 eos::FormFactorAdapterFactory< Transition_ > Class Template Reference . . . 191
6.77.1 Constructor & Destructor Documentation . . . 191
6.77.1.1 FormFactorAdapterFactory . . . 191
6.77.1.2 ~FormFactorAdapterFactory . . . 191
6.77.2 Member Function Documentation . . . 191
6.77.2.1 make . . . 191
6.78 eos::FormFactorFactory< OneHalfPlusToOneHalfPlus > Class Template Reference . . . 192
6.78.1 Member Function Documentation . . . 192
6.78.1.1 create . . . 192
6.79 eos::FormFactorFactory< PToP > Class Template Reference . . . 193
6.79.1 Member Function Documentation . . . 193
6.79.1.1 create . . . 193
6.80 eos::FormFactorFactory< PToV > Class Template Reference . . . 194
6.80.1 Member Function Documentation . . . 194
6.80.1.1 create . . . 194
6.81 eos::FormFactorRatioAdapter< Transition_ > Class Template Reference . . . 195
6.81.1 Constructor & Destructor Documentation . . . 195
6.81.1.1 FormFactorRatioAdapter . . . 195
6.81.2 Member Function Documentation . . . 195
6.81.2.1 clone . . . 195
6.81.2.2 clone . . . 195
6.81.2.3 evaluate . . . 195
6.81.2.4 kinematics . . . 196
6.81.2.5 name . . . 196
6.81.2.6 options . . . 196
6.81.2.7 parameters . . . 196
6.82 eos::FormFactorRatioAdapterFactory< Transition_ > Class Template Reference . . . 197
6.82.1 Constructor & Destructor Documentation . . . 197
6.82.1.1 FormFactorRatioAdapterFactory . . . 197
6.82.1.2 ~FormFactorRatioAdapterFactory . . . 197
6.82.2 Member Function Documentation . . . 197
6.82.2.1 make . . . 197

6.83 eos::FormFactors< OneHalfPlusToOneHalfPlus > Class Template Reference 198
6.83.1 Constructor & Destructor Documentation 198
6.83.1.1 ~FormFactors 198
6.83.2 Member Function Documentation 198
6.83.2.1 f_long_a 198
6.83.2.2 f_long_v 198
6.83.2.3 f_perp_a 198
6.83.2.4 f_perp_v 198
6.84 eos::FormFactors< PToP > Class Template Reference 199
6.84.1 Constructor & Destructor Documentation 199
6.84.1.1 ~FormFactors 199
6.84.2 Member Function Documentation 199
6.84.2.1 f_0 199
6.84.2.2 f_p 199
6.84.2.3 f_p_d1 199
6.84.2.4 f_p_d2 199
6.84.2.5 f_t 199
6.85 eos::FormFactors< PToV > Class Template Reference 201
6.85.1 Constructor & Destructor Documentation 201
6.85.1.1 ~FormFactors 201
6.85.2 Member Function Documentation 201
6.85.2.1 a_0 201
6.85.2.2 a_1 201
6.85.2.3 a_12 201
6.85.2.4 a_2 201
6.85.2.5 v 202
6.86 eos::proposal_functions::GlobalLocal Class Reference 203
6.86.1 Detailed Description 204
6.86.2 Member Typedef Documentation 204
6.86.2.1 ComponentType 204
6.86.2.2 HistoryType 204
6.86.2.3 JumpType 204
6.86.2.4 LocalCovarianceType 204
6.86.3 Constructor & Destructor Documentation 204
6.86.3.1 GlobalLocal 204
6.86.3.2 GlobalLocal 205

6.86.3.3 ~GlobalLocal . . . 205
6.86.4 Member Function Documentation . . . 205
6.86.4.1 adapt . . . 205
6.86.4.2 clone . . . 205
6.86.4.3 component_probabilities . . . 206
6.86.4.4 component_type . . . 206
6.86.4.5 config . . . 206
6.86.4.6 dump_state . . . 206
6.86.4.7 evaluate . . . 206
6.86.4.8 history_states . . . 206
6.86.4.9 history_type . . . 206
6.86.4.10 jump_type . . . 206
6.86.4.11 local_covariance_type . . . 206
6.86.4.12 local_covariances . . . 206
6.86.4.13 mode . . . 206
6.86.4.14 propose . . . 206
6.87 eos::HardScattering Struct Reference . . . 208
6.87.1 Member Function Documentation . . . 208
6.87.1.1 I1 . . . 208
6.88 eos::HDF5Error Class Reference . . . 209
6.88.1 Detailed Description . . . 209
6.88.2 Constructor & Destructor Documentation . . . 209
6.88.2.1 HDF5Error . . . 209
6.89 eos::HierarchicalClustering Class Reference . . . 210
6.89.1 Detailed Description . . . 210
6.89.2 Member Typedef Documentation . . . 211
6.89.2.1 ClusterIterator . . . 211
6.89.2.2 ComponentIterator . . . 211
6.89.2.3 MapIterator . . . 211
6.89.2.4 MixtureDensity . . . 211
6.89.3 Constructor & Destructor Documentation . . . 211
6.89.3.1 HierarchicalClustering . . . 211
6.89.3.2 ~HierarchicalClustering . . . 211
6.89.4 Member Function Documentation . . . 211
6.89.4.1 add . . . 211
6.89.4.2 begin_clusters . . . 212

6.89.4.3 begin_components . . . 212
6.89.4.4 begin_map . . . 212
6.89.4.5 end_clusters . . . 212
6.89.4.6 end_components . . . 212
6.89.4.7 end_map . . . 212
6.89.4.8 initial_guess . . . 212
6.89.4.9 run . . . 212
6.90 eos::Histogram< 1 > Class Template Reference . . . 213
6.90.1 Member Typedef Documentation . . . 214
6.90.1.1 ConstIterator . . . 214
6.90.2 Constructor & Destructor Documentation . . . 214
6.90.2.1 Histogram . . . 214
6.90.2.2 ~Histogram . . . 214
6.90.3 Member Function Documentation . . . 214
6.90.3.1 begin . . . 214
6.90.3.2 end . . . 214
6.90.3.3 entries . . . 214
6.90.3.4 insert . . . 214
6.90.3.5 insert . . . 214
6.90.3.6 WithEqualBinning . . . 214
6.91 eos::Histogram< 2 > Class Template Reference . . . 216
6.91.1 Member Typedef Documentation . . . 217
6.91.1.1 ConstIterator . . . 217
6.91.1.2 Iterator . . . 217
6.91.2 Constructor & Destructor Documentation . . . 217
6.91.2.1 Histogram . . . 217
6.91.2.2 ~Histogram . . . 217
6.91.3 Member Function Documentation . . . 217
6.91.3.1 begin . . . 217
6.91.3.2 cbegin . . . 217
6.91.3.3 cend . . . 217
6.91.3.4 end . . . 217
6.91.3.5 entries . . . 217
6.91.3.6 find . . . 217
6.91.3.7 insert . . . 217
6.91.3.8 WithEqualBinning . . . 217

6.92 eos::MarkovChain::History Struct Reference . . . 219
6.92.1 Detailed Description . . . 219
6.92.2 Member Function Documentation . . . 219
6.92.2.1 local_mode . . . 219
6.92.2.2 mean_and_covariance . . . 219
6.92.2.3 mean_and_variance . . . 220
6.92.3 Member Data Documentation . . . 220
6.92.3.1 keep . . . 220
6.92.3.2 states . . . 220
6.93 eos::HLMW2005 Struct Reference . . . 221
6.94 eos::MarkovChain::HyperParameter Struct Reference . . . 222
6.94.1 Member Data Documentation . . . 222
6.94.1.1 component . . . 222
6.95 eos::InstantiationPolicy< T_, NonCopyable > Class Template Reference . . . 223
6.95.1 Constructor & Destructor Documentation . . . 223
6.95.1.1 InstantiationPolicy . . . 223
6.96 eos::InstantiationPolicy< T_, NonInstantiable > Class Template Reference . . . 224
6.97 eos::InstantiationPolicy< T_, Singleton > Class Template Reference . . . 225
6.97.1 Constructor & Destructor Documentation . . . 225
6.97.1.1 InstantiationPolicy . . . 225
6.97.2 Member Function Documentation . . . 225
6.97.2.1 instance . . . 225
6.97.3 Friends And Related Function Documentation . . . 225
6.97.3.1 DeleteOnDestruction . . . 225
6.98 eos::InternalError Struct Reference . . . 226
6.98.1 Constructor & Destructor Documentation . . . 226
6.98.1.1 InternalError . . . 226
6.99 eos::Kinematics Class Reference . . . 227
6.99.1 Detailed Description . . . 227
6.99.2 Constructor & Destructor Documentation . . . 227
6.99.2.1 Kinematics . . . 227
6.99.2.2 Kinematics . . . 228
6.99.2.3 ~Kinematics . . . 228
6.99.3 Member Function Documentation . . . 228
6.99.3.1 as_string . . . 228
6.99.3.2 clone . . . 228

6.99.3.3 declare . . . 228
6.99.3.4 operator!= . . . 228
6.99.3.5 operator== . . . 228
6.99.3.6 operator[] . . . 228
6.99.3.7 set . . . 229
6.100 eos::KinematicVariable Class Reference . . . 230
6.100.1 Detailed Description . . . 230
6.100.2 Constructor & Destructor Documentation . . . 230
6.100.2.1 ~KinematicVariable . . . 230
6.100.3 Member Function Documentation . . . 230
6.100.3.1 operator double . . . 230
6.100.3.2 operator() . . . 230
6.100.3.3 operator= . . . 231
6.100.4 Friends And Related Function Documentation . . . 231
6.100.4.1 Implementation< Kinematics > . . . 231
6.100.4.2 Kinematics . . . 231
6.101 eos::KMPW2010FormFactors< PToP > Class Template Reference . . . 232
6.101.1 Constructor & Destructor Documentation . . . 232
6.101.1.1 KMPW2010FormFactors . . . 232
6.101.2 Member Function Documentation . . . 232
6.101.2.1 f_0 . . . 232
6.101.2.2 f_p . . . 232
6.101.2.3 f_t . . . 232
6.101.2.4 make . . . 232
6.102 eos::KMPW2010FormFactors< PToV > Class Template Reference . . . 233
6.102.1 Constructor & Destructor Documentation . . . 233
6.102.1.1 KMPW2010FormFactors . . . 233
6.102.2 Member Function Documentation . . . 233
6.102.2.1 a_0 . . . 233
6.102.2.2 a_1 . . . 233
6.102.2.3 a_12 . . . 233
6.102.2.4 a_2 . . . 233
6.102.2.5 make . . . 234
6.102.2.6 v . . . 234
6.103 eos::LambdaBToLambdaDilepton< LargeRecoil > Class Template Reference . . . 235
6.103.1 Constructor & Destructor Documentation . . . 236

6.103.1.1 LambdaBToLambdaDilepton 236
6.103.1.2 ~LambdaBToLambdaDilepton 236
6.103.2 Member Function Documentation 236
6.103.2.1 differential_a_fb_combined 236
6.103.2.2 differential_a_fb_hadronic 236
6.103.2.3 differential_a_fb_leptonic 236
6.103.2.4 differential_branching_ratio 236
6.103.2.5 differential_fzero 236
6.103.2.6 integrated_a_fb_combined 236
6.103.2.7 integrated_a_fb_hadronic 236
6.103.2.8 integrated_a_fb_leptonic 236
6.103.2.9 integrated_branching_ratio 236
6.103.2.10 integrated_fzero 236
6.104 eos::LambdaBToLambdaDilepton< LowRecoil > Class Template Reference 237
6.104.1 Constructor & Destructor Documentation 238
6.104.1.1 LambdaBToLambdaDilepton 238
6.104.1.2 ~LambdaBToLambdaDilepton 238
6.104.2 Member Function Documentation 238
6.104.2.1 differential_a_fb_combined 238
6.104.2.2 differential_a_fb_hadronic 238
6.104.2.3 differential_a_fb_leptonic 238
6.104.2.4 differential_branching_ratio 238
6.104.2.5 differential_fzero 238
6.104.2.6 integrated_a_fb_combined 238
6.104.2.7 integrated_a_fb_hadronic 238
6.104.2.8 integrated_a_fb_leptonic 238
6.104.2.9 integrated_branching_ratio 238
6.104.2.10 integrated_fzero 238
6.105 eos::LargeRecoil Struct Reference 239
6.106 eos::LCDA Struct Reference 240
6.106.1 Detailed Description 240
6.106.2 Member Function Documentation 240
6.106.2.1 evolve_gegenbauer_moment 240
6.107 eos::Lock Class Reference 241
6.107.1 Constructor & Destructor Documentation 241
6.107.1.1 Lock 241

6.107.1.2 ~Lock . . . 241
6.108 eos::Log Class Reference . . . 242
6.108.1 Detailed Description . . . 242
6.108.2 Constructor & Destructor Documentation . . . 242
6.108.2.1 ~Log . . . 242
6.108.3 Member Function Documentation . . . 242
6.108.3.1 get_log_level . . . 242
6.108.3.2 message . . . 242
6.108.3.3 set_log_level . . . 243
6.108.3.4 set_log_stream . . . 243
6.108.3.5 set_program_name . . . 243
6.108.4 Friends And Related Function Documentation . . . 243
6.108.4.1 InstantiationPolicy< Log, Singleton > . . . 243
6.108.4.2 LogMessageHandler . . . 243
6.109 eos::LogLikelihood Class Reference . . . 244
6.109.1 Detailed Description . . . 244
6.109.2 Member Typedef Documentation . . . 245
6.109.2.1 ConstraintIterator . . . 245
6.109.3 Constructor & Destructor Documentation . . . 245
6.109.3.1 LogLikelihood . . . 245
6.109.3.2 ~LogLikelihood . . . 245
6.109.4 Member Function Documentation . . . 245
6.109.4.1 add . . . 245
6.109.4.2 add . . . 245
6.109.4.3 begin . . . 245
6.109.4.4 bootstrap_p_value . . . 246
6.109.4.5 clone . . . 246
6.109.4.6 end . . . 246
6.109.4.7 number_of_observations . . . 246
6.109.4.8 observable_cache . . . 246
6.109.4.9 operator() . . . 246
6.109.4.10 operator() . . . 247
6.109.4.11 parameters . . . 247
6.109.4.12 reset . . . 247
6.110 eos::LogLikelihoodBlock Class Reference . . . 248
6.110.1 Detailed Description . . . 249

6.110.2 Constructor & Destructor Documentation . . . 249
6.110.2.1 ~LogLikelihoodBlock . . . 249
6.110.3 Member Function Documentation . . . 249
6.110.3.1 Amoroso . . . 249
6.110.3.2 Amoroso . . . 250
6.110.3.3 AmorosoLimit . . . 250
6.110.3.4 AmorosoMode . . . 251
6.110.3.5 as_string . . . 251
6.110.3.6 clone . . . 251
6.110.3.7 evaluate . . . 252
6.110.3.8 Gaussian . . . 252
6.110.3.9 LogGamma . . . 252
6.110.3.10 LogGamma . . . 252
6.110.3.11 Mixture . . . 253
6.110.3.12 MultivariateGaussian . . . 253
6.110.3.13 MultivariateGaussian . . . 253
6.110.3.14 number_of_observations . . . 254
6.110.3.15 prepare_sampling . . . 254
6.110.3.16 sample . . . 254
6.110.3.17 significance . . . 254
6.111 eos::LogMessageHandler Class Reference . . . 255
6.111.1 Detailed Description . . . 255
6.111.2 Constructor & Destructor Documentation . . . 255
6.111.2.1 ~LogMessageHandler . . . 255
6.111.3 Member Function Documentation . . . 255
6.111.3.1 operator<< . . . 255
6.111.4 Friends And Related Function Documentation . . . 255
6.111.4.1 Log::message . . . 255
6.112 eos::LogPrior Class Reference . . . 256
6.112.1 Detailed Description . . . 257
6.112.2 Member Typedef Documentation . . . 257
6.112.2.1 Iterator . . . 257
6.112.3 Constructor & Destructor Documentation . . . 257
6.112.3.1 LogPrior . . . 257
6.112.3.2 ~LogPrior . . . 257
6.112.4 Member Function Documentation . . . 257

6.112.4.1 as_string . . . 257
6.112.4.2 begin . . . 257
6.112.4.3 clone . . . 257
6.112.4.4 Discrete . . . 258
6.112.4.5 end . . . 258
6.112.4.6 Flat . . . 258
6.112.4.7 Gauss . . . 259
6.112.4.8 LogGamma . . . 260
6.112.4.9 Make . . . 260
6.112.4.10 mean . . . 260
6.112.4.11 operator() . . . 261
6.112.4.12 sample . . . 261
6.112.4.13 variance . . . 261
6.112.5 Member Data Documentation . . . 261
6.112.5.1 _parameter_descriptions . . . 261
6.112.5.2 _parameters . . . 261
6.113 eos::LongDistance Struct Reference . . . 262
6.113.1 Member Function Documentation . . . 262
6.113.1.1 g_had_ccbar . . . 262
6.114 eos::LowRecoil Struct Reference . . . 263
6.115 eos::MarkovChain Class Reference . . . 264
6.115.1 Constructor & Destructor Documentation . . . 265
6.115.1.1 MarkovChain . . . 265
6.115.1.2 ~MarkovChain . . . 265
6.115.2 Member Function Documentation . . . 266
6.115.2.1 clear . . . 266
6.115.2.2 current_state . . . 266
6.115.2.3 dump_description . . . 266
6.115.2.4 dump_history . . . 266
6.115.2.5 dump_proposal . . . 266
6.115.2.6 history . . . 266
6.115.2.7 hyper_parameter . . . 266
6.115.2.8 iterations_last_run . . . 266
6.115.2.9 keep_history . . . 266
6.115.2.10 normalized_density . . . 267
6.115.2.11 parameter_descriptions . . . 267

6.115.2.12 proposal_accepted . . . 267
6.115.2.13 proposal_function . . . 267
6.115.2.14 proposal_function . . . 267
6.115.2.15 proposed_state . . . 267
6.115.2.16 read_data . . . 267
6.115.2.17 read_descriptions . . . 268
6.115.2.18 reset . . . 268
6.115.2.19 run . . . 268
6.115.2.20 set_analysis . . . 268
6.115.2.21 set_mode . . . 269
6.115.2.22 set_point . . . 269
6.115.2.23 statistics . . . 269
6.116 eos::MarkovChainSampler Class Reference . . . 270
6.116.1 Constructor & Destructor Documentation . . . 270
6.116.1.1 MarkovChainSampler . . . 270
6.116.1.2 ∼MarkovChainSampler . . . 271
6.116.2 Member Function Documentation . . . 271
6.116.2.1 build_global_local . . . 271
6.116.2.2 config . . . 271
6.116.2.3 massive_mode_finding . . . 271
6.116.2.4 pre_run_info . . . 271
6.116.2.5 resume . . . 271
6.116.2.6 run . . . 272
6.117 eos::MemoisationControl Class Reference . . . 273
6.117.1 Constructor & Destructor Documentation . . . 273
6.117.1.1 MemoisationControl . . . 273
6.117.1.2 ∼MemoisationControl . . . 273
6.117.2 Member Function Documentation . . . 273
6.117.2.1 clear . . . 273
6.117.2.2 register_clear_function . . . 273
6.118 eos::Memoiser< Result_, Params_ > Class Template Reference . . . 274
6.118.1 Member Typedef Documentation . . . 274
6.118.1.1 FunctionType . . . 274
6.118.1.2 KeyType . . . 274
6.118.2 Constructor & Destructor Documentation . . . 274
6.118.2.1 Memoiser . . . 274

6.118.2.2 ∼Memoiser . . . 274
6.118.3 Member Function Documentation . . . 274
6.118.3.1 clear . . . 274
6.118.3.2 number_of_memoisations . . . 274
6.118.3.3 operator() . . . 274
6.119 eos::Minimal Struct Reference . . . 275
6.120 eos::Model Class Reference . . . 276
6.120.1 Detailed Description . . . 276
6.120.2 Constructor & Destructor Documentation . . . 276
6.120.2.1 ∼Model . . . 276
6.120.3 Member Function Documentation . . . 276
6.120.3.1 make . . . 276
6.121 eos::ModelComponent< components::CKM > Class Template Reference . . . 277
6.121.1 Detailed Description . . . 277
6.121.2 Member Function Documentation . . . 277
6.121.2.1 ckm_cb . . . 277
6.121.2.2 ckm_cd . . . 277
6.121.2.3 ckm_cs . . . 277
6.121.2.4 ckm_tb . . . 277
6.121.2.5 ckm_td . . . 278
6.121.2.6 ckm_ts . . . 278
6.121.2.7 ckm_ub . . . 278
6.121.2.8 ckm_ud . . . 278
6.121.2.9 ckm_us . . . 278
6.122 eos::ModelComponent< components::DeltaB1 > Class Template Reference . . . 279
6.122.1 Detailed Description . . . 279
6.122.2 Member Function Documentation . . . 279
6.122.2.1 wilson_coefficients_b_to_s . . . 279
6.123 eos::ModelComponent< components::QCD > Class Template Reference . . . 280
6.123.1 Detailed Description . . . 280
6.123.2 Member Function Documentation . . . 280
6.123.2.1 alpha_s . . . 280
6.123.2.2 m_b_kin . . . 280
6.123.2.3 m_b_msbar . . . 280
6.123.2.4 m_b_pole . . . 280
6.123.2.5 m_b_ps . . . 281

6.123.2.6 m_c_kin . . . 281
6.123.2.7 m_c_msbar . . . 281
6.123.2.8 m_c_pole . . . 281
6.123.2.9 m_s_msbar . . . 281
6.123.2.10 m_t_msbar . . . 281
6.123.2.11 m_t_pole . . . 281
6.123.2.12 m_ud_msbar . . . 281
6.124 eos::proposal_functions::Multivariate Class Reference . . . 282
6.124.1 Member Typedef Documentation . . . 283
6.124.1.1 CovarianceType . . . 283
6.124.2 Constructor & Destructor Documentation . . . 283
6.124.2.1 Multivariate . . . 283
6.124.2.2 ~Multivariate . . . 284
6.124.3 Member Function Documentation . . . 284
6.124.3.1 _compute_cholesky_and_inverse . . . 284
6.124.3.2 _compute_norm . . . 284
6.124.3.3 _copy . . . 284
6.124.3.4 _dump_covariance . . . 284
6.124.3.5 adapt . . . 284
6.124.3.6 covariance . . . 284
6.124.3.7 covariance_type . . . 284
6.124.3.8 dimension . . . 284
6.124.3.9 rescale . . . 284
6.124.3.10 reset . . . 285
6.124.3.11 set_indices . . . 285
6.124.4 Member Data Documentation . . . 285
6.124.4.1 _covariance . . . 285
6.124.4.2 _covariance_chol . . . 285
6.124.4.3 _covariance_inverse . . . 285
6.124.4.4 _dimension . . . 285
6.124.4.5 _index_list . . . 285
6.124.4.6 _tmp_left . . . 285
6.124.4.7 _tmp_right . . . 285
6.124.4.8 _tmp_sample_covariance_current . . . 285
6.124.4.9 adaptations . . . 285
6.124.4.10 cooling_power . . . 285

6.124.4.11 covariance_scale . . . 286
6.124.4.12 covariance_scale_max . . . 286
6.124.4.13 covariance_scale_min . . . 286
6.124.4.14 covariance_scale_update_factor . . . 286
6.124.4.15 norm . . . 286
6.125 eos::proposal_functions::MultivariateAccess Struct Reference . . . 287
6.125.1 Member Function Documentation . . . 287
6.125.1.1 access . . . 287
6.126 eos::proposal_functions::MultivariateGaussian Class Reference . . . 288
6.126.1 Member Typedef Documentation . . . 289
6.126.1.1 ScalarsType . . . 289
6.126.2 Constructor & Destructor Documentation . . . 289
6.126.2.1 MultivariateGaussian . . . 289
6.126.2.2 ~MultivariateGaussian . . . 289
6.126.3 Member Function Documentation . . . 289
6.126.3.1 _compute_norm . . . 289
6.126.3.2 clone . . . 289
6.126.3.3 dump_state . . . 289
6.126.3.4 evaluate . . . 289
6.126.3.5 propose . . . 289
6.126.3.6 scalars_type . . . 290
6.127 eos::proposal_functions::MultivariateStudentT Class Reference . . . 291
6.127.1 Member Typedef Documentation . . . 292
6.127.1.1 ScalarsType . . . 292
6.127.2 Constructor & Destructor Documentation . . . 292
6.127.2.1 MultivariateStudentT . . . 292
6.127.2.2 ~MultivariateStudentT . . . 292
6.127.3 Member Function Documentation . . . 292
6.127.3.1 _compute_norm . . . 292
6.127.3.2 clone . . . 292
6.127.3.3 dump_state . . . 292
6.127.3.4 evaluate . . . 292
6.127.3.5 propose . . . 292
6.127.3.6 scalars_type . . . 293
6.127.4 Member Data Documentation . . . 293
6.127.4.1 dof . . . 293

6.128eos::Mutable Class Reference . . . 294
6.128.1 Detailed Description . . . 294
6.128.2 Constructor & Destructor Documentation . . . 294
6.128.2.1 ∼Mutable . . . 294
6.128.3 Member Function Documentation . . . 295
6.128.3.1 clone . . . 295
6.128.3.2 name . . . 295
6.128.3.3 operator double . . . 295
6.128.3.4 operator() . . . 295
6.128.3.5 operator= . . . 295
6.129eos::Mutex Class Reference . . . 296
6.129.1 Constructor & Destructor Documentation . . . 296
6.129.1.1 Mutex . . . 296
6.129.1.2 ∼Mutex . . . 296
6.129.2 Member Function Documentation . . . 296
6.129.2.1 mutex . . . 296
6.130eos::NLO Struct Reference . . . 297
6.131eos::NoSuchChiSquaredError Struct Reference . . . 298
6.131.1 Constructor & Destructor Documentation . . . 298
6.131.1.1 NoSuchChiSquaredError . . . 298
6.132eos::NoSuchModelError Struct Reference . . . 299
6.132.1 Constructor & Destructor Documentation . . . 299
6.132.1.1 NoSuchModelError . . . 299
6.133eos::Observable Class Reference . . . 300
6.133.1 Member Function Documentation . . . 300
6.133.1.1 clone . . . 300
6.133.1.2 clone . . . 300
6.133.1.3 evaluate . . . 300
6.133.1.4 kinematics . . . 300
6.133.1.5 make . . . 301
6.133.1.6 name . . . 301
6.133.1.7 options . . . 301
6.133.1.8 parameters . . . 301
6.134eos::ObservableCache Class Reference . . . 302
6.134.1 Member Typedef Documentation . . . 302
6.134.1.1 Id . . . 302

6.134.2 Constructor & Destructor Documentation . . . 303
6.134.2.1 ObservableCache . . . 303
6.134.2.2 ∼ObservableCache . . . 303
6.134.3 Member Function Documentation . . . 303
6.134.3.1 add . . . 303
6.134.3.2 clone . . . 303
6.134.3.3 observable . . . 303
6.134.3.4 operator[] . . . 303
6.134.3.5 parameters . . . 303
6.134.3.6 reset . . . 304
6.134.3.7 size . . . 304
6.134.3.8 update . . . 304
6.134.3.9 update . . . 304
6.135eos::ObservableFactory Class Reference . . . 305
6.135.1 Constructor & Destructor Documentation . . . 305
6.135.1.1 ObservableFactory . . . 305
6.135.1.2 ∼ObservableFactory . . . 305
6.135.2 Member Function Documentation . . . 305
6.135.2.1 make . . . 305
6.136eos::ObservableNameError Struct Reference . . . 306
6.136.1 Detailed Description . . . 306
6.136.2 Constructor & Destructor Documentation . . . 306
6.136.2.1 ObservableNameError . . . 306
6.137eos::ObservableSet Class Reference . . . 307
6.137.1 Detailed Description . . . 307
6.137.2 Member Typedef Documentation . . . 307
6.137.2.1 Iterator . . . 307
6.137.3 Constructor & Destructor Documentation . . . 308
6.137.3.1 ObservableSet . . . 308
6.137.3.2 ∼ObservableSet . . . 308
6.137.4 Member Function Documentation . . . 308
6.137.4.1 add . . . 308
6.137.4.2 begin . . . 308
6.137.4.3 end . . . 308
6.137.4.4 operator[] . . . 308
6.137.4.5 parameters . . . 308

6.137.4.6 size . . . 309
6.138 eos::OneHalfPlusToOneHalfPlus Struct Reference . . . 310
6.139 eos::OneOf< Types_ > Class Template Reference . . . 311
6.139.1 Constructor & Destructor Documentation . . . 311
6.139.1.1 OneOf . . . 311
6.139.1.2 OneOf . . . 311
6.139.2 Member Function Documentation . . . 311
6.139.2.1 accept_returning . . . 311
6.140 eos::OneOfValue< Type_, Types_ > Struct Template Reference . . . 312
6.140.1 Constructor & Destructor Documentation . . . 312
6.140.1.1 OneOfValue . . . 312
6.140.2 Member Function Documentation . . . 312
6.140.2.1 accept . . . 312
6.140.3 Member Data Documentation . . . 312
6.140.3.1 value . . . 312
6.141 eos::OneOfValueBase< Types_ > Struct Template Reference . . . 313
6.141.1 Constructor & Destructor Documentation . . . 313
6.141.1.1 ~OneOfValueBase . . . 313
6.141.2 Member Function Documentation . . . 313
6.141.2.1 accept . . . 313
6.142 eos::OneOfVisitor< Types_ > Struct Template Reference . . . 314
6.143 eos::OneOfVisitorVisit< Type_ > Struct Template Reference . . . 315
6.143.1 Member Function Documentation . . . 315
6.143.1.1 visit . . . 315
6.144 eos::OneOfVisitorWrapper< Underlying_, Result_, Types_ > Struct Template Reference . . . 316
6.144.1 Constructor & Destructor Documentation . . . 316
6.144.1.1 OneOfVisitorWrapper . . . 316
6.145 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ > Struct Template Reference . . . 317
6.145.1 Constructor & Destructor Documentation . . . 317
6.145.1.1 OneOfVisitorWrapperVisit . . . 317
6.145.2 Member Data Documentation . . . 317
6.145.2.1 result . . . 317
6.145.2.2 underlying . . . 317
6.146 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...> Struct Template Reference . . . 318
6.146.1 Constructor & Destructor Documentation . . . 318

6.146.1.1 OneOfVisitorWrapperVisit . . . 318
6.146.2 Member Function Documentation . . . 318
6.146.2.1 visit . . . 318
6.147 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void > Struct Template Reference 319
6.147.1 Constructor & Destructor Documentation . . . 319
6.147.1.1 OneOfVisitorWrapperVisit . . . 319
6.147.2 Member Data Documentation . . . 319
6.147.2.1 underlying . . . 319
6.148 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...> Struct Template Reference . . . 320
6.148.1 Constructor & Destructor Documentation . . . 320
6.148.1.1 OneOfVisitorWrapperVisit . . . 320
6.148.2 Member Function Documentation . . . 320
6.148.2.1 visit . . . 320
6.149 eos::Analysis::OptimizationOptions Struct Reference . . . 321
6.149.1 Member Function Documentation . . . 321
6.149.1.1 Defaults . . . 321
6.149.2 Member Data Documentation . . . 321
6.149.2.1 algorithm . . . 321
6.149.2.2 fix_flat_nuisance . . . 321
6.149.2.3 initial_step_size . . . 321
6.149.2.4 maximum_iterations . . . 321
6.149.2.5 mcmc_pre_run . . . 322
6.149.2.6 splitting_tolerance . . . 322
6.149.2.7 strategy_level . . . 322
6.149.2.8 tolerance . . . 322
6.150 eos::Options Class Reference . . . 323
6.150.1 Detailed Description . . . 323
6.150.2 Constructor & Destructor Documentation . . . 323
6.150.2.1 Options . . . 323
6.150.2.2 Options . . . 323
6.150.2.3 ~Options . . . 324
6.150.3 Member Function Documentation . . . 324
6.150.3.1 as_string . . . 324
6.150.3.2 get . . . 324
6.150.3.3 has . . . 324
6.150.3.4 operator!= . . . 324

6.150.3.5 operator== . . . 324
6.150.3.6 operator[] . . . 324
6.150.3.7 set . . . 324
6.150.4 Friends And Related Function Documentation . . . 324
6.150.4.1 operator+ . . . 324
6.151 eos::PopulationMonteCarloSampler::Output Struct Reference . . . 325
6.151.1 Detailed Description . . . 325
6.151.2 Member Typedef Documentation . . . 326
6.151.2.1 ComponentType . . . 326
6.151.2.2 IgnoreType . . . 326
6.151.2.3 SampleType . . . 326
6.151.2.4 StatisticsType . . . 326
6.151.2.5 WeightType . . . 326
6.151.3 Member Function Documentation . . . 326
6.151.3.1 component_record . . . 326
6.151.3.2 component_type . . . 326
6.151.3.3 ignore_record . . . 326
6.151.3.4 ignore_type . . . 326
6.151.3.5 sample_record . . . 326
6.151.3.6 sample_type . . . 326
6.151.3.7 statistics_record . . . 326
6.151.3.8 statistics_type . . . 326
6.151.3.9 weight_record . . . 326
6.151.3.10 weight_type . . . 326
6.152 eos::Analysis::Output Struct Reference . . . 328
6.152.1 Member Typedef Documentation . . . 328
6.152.1.1 DescriptionType . . . 328
6.152.2 Member Function Documentation . . . 328
6.152.2.1 description_record . . . 328
6.152.2.2 description_type . . . 328
6.153 eos::Parameter Class Reference . . . 329
6.153.1 Detailed Description . . . 330
6.153.2 Member Typedef Documentation . . . 330
6.153.2.1 Id . . . 330
6.153.3 Constructor & Destructor Documentation . . . 330
6.153.3.1 Parameter . . . 330

6.153.3.2 ∼Parameter . . . 330
6.153.4 Member Function Documentation . . . 330
6.153.4.1 central . . . 330
6.153.4.2 clone . . . 330
6.153.4.3 id . . . 331
6.153.4.4 max . . . 331
6.153.4.5 min . . . 331
6.153.4.6 name . . . 331
6.153.4.7 operator double . . . 331
6.153.4.8 operator() . . . 331
6.153.4.9 operator= . . . 331
6.153.5 Friends And Related Function Documentation . . . 331
6.153.5.1 Implementation< Parameters > . . . 331
6.153.5.2 Parameters . . . 331
6.154eos::ParameterDescription Struct Reference . . . 332
6.154.1 Member Data Documentation . . . 332
6.154.1.1 discrete . . . 332
6.154.1.2 max . . . 332
6.154.1.3 min . . . 332
6.154.1.4 nuisance . . . 332
6.154.1.5 parameter . . . 332
6.155eos::ParameterRange Struct Reference . . . 333
6.155.1 Member Data Documentation . . . 333
6.155.1.1 max . . . 333
6.155.1.2 min . . . 333
6.156eos::Parameters Class Reference . . . 334
6.156.1 Detailed Description . . . 334
6.156.2 Member Typedef Documentation . . . 335
6.156.2.1 Iterator . . . 335
6.156.3 Constructor & Destructor Documentation . . . 335
6.156.3.1 ∼Parameters . . . 335
6.156.4 Member Function Documentation . . . 335
6.156.4.1 begin . . . 335
6.156.4.2 clone . . . 335
6.156.4.3 declare . . . 335
6.156.4.4 Defaults . . . 335

6.156.4.5 end . . . 335
6.156.4.6 operator!= . . . 335
6.156.4.7 operator[] . . . 336
6.156.4.8 operator[] . . . 336
6.156.4.9 set . . . 336
6.156.5 Friends And Related Function Documentation . . . 336
6.156.5.1 Implementation< Parameter > . . . 336
6.156.5.2 Implementation< Parameters > . . . 336
6.156.5.3 Parameter . . . 336
6.157 eos::ParameterUser Class Reference . . . 337
6.157.1 Detailed Description . . . 337
6.157.2 Member Typedef Documentation . . . 337
6.157.2.1 ConstIterator . . . 337
6.157.3 Member Function Documentation . . . 337
6.157.3.1 begin . . . 337
6.157.3.2 end . . . 337
6.157.3.3 uses . . . 337
6.157.3.4 uses . . . 338
6.157.4 Member Data Documentation . . . 338
6.157.4.1 _ids . . . 338
6.158 eos::PionLCDAs Class Reference . . . 339
6.158.1 Constructor & Destructor Documentation . . . 340
6.158.1.1 PionLCDAs . . . 340
6.158.1.2 ~PionLCDAs . . . 340
6.158.2 Member Function Documentation . . . 340
6.158.2.1 a2pi . . . 340
6.158.2.2 a4pi . . . 340
6.158.2.3 deltapipi . . . 340
6.158.2.4 diagnostics . . . 340
6.158.2.5 eta3pi . . . 340
6.158.2.6 f3pi . . . 340
6.158.2.7 mupi . . . 340
6.158.2.8 omega3pi . . . 340
6.158.2.9 omega4pi . . . 340
6.158.2.10 phi . . . 340
6.158.2.11 phi3p . . . 340

6.158.2.12 phi3s . . . 340
6.158.2.13 phi3s_d1 . . . 340
6.158.2.14 phi4 . . . 340
6.158.2.15 phi4_d1 . . . 340
6.158.2.16 phi4_d2 . . . 340
6.158.2.17 psi4 . . . 340
6.158.2.18 psi4_i . . . 340
6.159 eos::PopulationMonteCarloSampler Class Reference . . . 341
6.159.1 Constructor & Destructor Documentation . . . 341
6.159.1.1 PopulationMonteCarloSampler . . . 341
6.159.1.2 PopulationMonteCarloSampler . . . 342
6.159.1.3 ~PopulationMonteCarloSampler . . . 342
6.159.2 Member Function Documentation . . . 342
6.159.2.1 calculate_weights . . . 342
6.159.2.2 config . . . 342
6.159.2.3 draw_samples . . . 342
6.159.2.4 read_samples . . . 342
6.159.2.5 run . . . 343
6.159.2.6 status . . . 343
6.159.2.7 status . . . 343
6.160 eos::MarkovChainSampler::PreRunInfo Struct Reference . . . 344
6.160.1 Detailed Description . . . 344
6.160.2 Member Data Documentation . . . 344
6.160.2.1 converged . . . 344
6.160.2.2 iterations . . . 344
6.160.2.3 iterations_at_convergence . . . 344
6.160.2.4 rvalue_parameters . . . 344
6.160.2.5 rvalue_posterior . . . 345
6.161 eos::PriorSampler Class Reference . . . 346
6.161.1 Detailed Description . . . 346
6.161.2 Member Typedef Documentation . . . 346
6.161.2.1 ObservablesType . . . 346
6.161.2.2 SamplesList . . . 346
6.161.3 Constructor & Destructor Documentation . . . 346
6.161.3.1 PriorSampler . . . 346
6.161.3.2 ~PriorSampler . . . 347

6.161.4 Member Function Documentation . . . 347
6.161.4.1 add . . . 347
6.161.4.2 add . . . 347
6.161.4.3 observables_type . . . 347
6.161.4.4 run . . . 347
6.161.4.5 run . . . 347
6.162eos::PrivateImplementationPattern< T_ > Class Template Reference . . . 348
6.162.1 Constructor & Destructor Documentation . . . 348
6.162.1.1 PrivateImplementationPattern . . . 348
6.162.1.2 ~PrivateImplementationPattern . . . 348
6.162.2 Member Data Documentation . . . 348
6.162.2.1 _imp . . . 348
6.163eos::MarkovChain::ProposalFunction Struct Reference . . . 349
6.163.1 Detailed Description . . . 349
6.163.2 Constructor & Destructor Documentation . . . 349
6.163.2.1 ~ProposalFunction . . . 349
6.163.3 Member Function Documentation . . . 349
6.163.3.1 adapt . . . 349
6.163.3.2 clone . . . 350
6.163.3.3 dump_state . . . 350
6.163.3.4 evaluate . . . 350
6.163.3.5 propose . . . 350
6.164eos::PToP Struct Reference . . . 351
6.165eos::PToV Struct Reference . . . 352
6.166eos::QCD Class Reference . . . 353
6.166.1 Member Typedef Documentation . . . 353
6.166.1.1 AnomalousMassDimension . . . 353
6.166.1.2 BetaFunction . . . 353
6.166.2 Member Function Documentation . . . 354
6.166.2.1 alpha_s . . . 354
6.166.2.2 m_q_kin . . . 354
6.166.2.3 m_q_msbar . . . 354
6.166.2.4 m_q_msbar . . . 354
6.166.2.5 m_q_pole . . . 355
6.166.2.6 m_q_ps . . . 355
6.166.3 Member Data Documentation . . . 355

6.166.3.1 beta_function_nf_3 . . . 355
6.166.3.2 beta_function_nf_4 . . . 355
6.166.3.3 beta_function_nf_5 . . . 355
6.166.3.4 beta_function_nf_6 . . . 356
6.166.3.5 casimir_a . . . 356
6.166.3.6 casimir_f . . . 356
6.166.3.7 gamma_m_nf_3 . . . 356
6.166.3.8 gamma_m_nf_4 . . . 356
6.166.3.9 gamma_m_nf_5 . . . 356
6.167 eos::QCDFIntegrals Class Reference . . . 357
6.167.1 Member Function Documentation . . . 357
6.167.1.1 dilepton_bottom_case . . . 357
6.167.1.2 dilepton_charm_case . . . 358
6.167.1.3 dilepton_massless_case . . . 358
6.167.1.4 photon_bottom_case . . . 358
6.167.1.5 photon_charm_case . . . 359
6.167.1.6 photon_massless_case . . . 359
6.168 eos::RandomNumberGenerator Class Reference . . . 360
6.168.1 Detailed Description . . . 360
6.168.2 Constructor & Destructor Documentation . . . 360
6.168.2.1 RandomNumberGenerator . . . 360
6.168.2.2 ~RandomNumberGenerator . . . 360
6.168.3 Member Function Documentation . . . 360
6.168.3.1 max . . . 360
6.168.3.2 min . . . 361
6.168.3.3 operator() . . . 361
6.169 eos::implementation::ResultOf< Result_(*)(Args_...)> Struct Template Reference . . . 362
6.169.1 Member Typedef Documentation . . . 362
6.169.1.1 Type . . . 362
6.170 eos::implementation::ResultOf< Result_(Class_::*)(Args_...)> Struct Template Reference 363
6.170.1 Member Typedef Documentation . . . 363
6.170.1.1 Type . . . 363
6.171 eos::QCDFIntegrals::Results Struct Reference . . . 364
6.171.1 Detailed Description . . . 364
6.171.2 Member Data Documentation . . . 365
6.171.2.1 j0_parallel . . . 365

6.171.2.2 j0_perp . . . 365
6.171.2.3 j0bar_perp . . . 365
6.171.2.4 j1_parallel . . . 365
6.171.2.5 j1_perp . . . 365
6.171.2.6 j2_perp . . . 365
6.171.2.7 j3_parallel . . . 365
6.171.2.8 j4_parallel . . . 365
6.171.2.9 j4_perp . . . 365
6.171.2.10 j5_perp . . . 365
6.171.2.11 j6_perp . . . 365
6.171.2.12 j7_perp . . . 365
6.171.2.13 jtilde1_perp . . . 365
6.171.2.14 jtilde2_parallel . . . 365
6.172 eos::RValue Struct Reference . . . 366
6.172.1 Member Function Documentation . . . 366
6.172.1.1 approximation . . . 366
6.172.1.2 gelman_rubin . . . 366
6.173 eos::Save< V_, R_ > Class Template Reference . . . 367
6.173.1 Constructor & Destructor Documentation . . . 367
6.173.1.1 Save . . . 367
6.173.1.2 ~Save . . . 367
6.174 eos::hdf5::Scalar< T_ > Class Template Reference . . . 368
6.174.1 Detailed Description . . . 368
6.174.2 Member Typedef Documentation . . . 368
6.174.2.1 Type . . . 368
6.174.3 Constructor & Destructor Documentation . . . 368
6.174.3.1 Scalar . . . 368
6.174.3.2 ~Scalar . . . 368
6.174.4 Member Function Documentation . . . 368
6.174.4.1 copy_from_hdf5 . . . 368
6.174.4.2 copy_to_hdf5 . . . 368
6.174.4.3 name . . . 369
6.174.4.4 size . . . 369
6.174.4.5 type_id . . . 369
6.175 eos::SelectOneOfType< Want_ > Struct Template Reference . . . 370
6.175.1 Member Typedef Documentation . . . 370

6.175.1.1 Type 370
6.176eos::SelectOneOfType< Want_, Try_, Rest_...> Struct Template Reference 371
6.176.1 Member Typedef Documentation 371
6.176.1.1 Type 371
6.177eos::ShortDistanceLowRecoil Struct Reference 372
6.177.1 Member Function Documentation 372
6.177.1.1 c7eff 372
6.177.1.2 c9eff 372
6.178eos::SMComponent< components::CKM > Class Template Reference 374
6.178.1 Constructor & Destructor Documentation 374
6.178.1.1 SMComponent 374
6.178.2 Member Function Documentation 374
6.178.2.1 ckm_cb 374
6.178.2.2 ckm_cd 374
6.178.2.3 ckm_cs 374
6.178.2.4 ckm_tb 374
6.178.2.5 ckm_td 375
6.178.2.6 ckm_ts 375
6.178.2.7 ckm_ub 375
6.178.2.8 ckm_ud 375
6.178.2.9 ckm_us 375
6.179eos::SMComponent< components::DeltaB1 > Class Template Reference 376
6.179.1 Constructor & Destructor Documentation 376
6.179.1.1 SMComponent 376
6.179.2 Member Function Documentation 376
6.179.2.1 wilson_coefficients_b_to_s 376
6.180eos::SMComponent< components::QCD > Class Template Reference 377
6.180.1 Constructor & Destructor Documentation 377
6.180.1.1 SMComponent 377
6.180.2 Member Function Documentation 377
6.180.2.1 alpha_s 377
6.180.2.2 m_b_kin 377
6.180.2.3 m_b_msbar 377
6.180.2.4 m_b_pole 378
6.180.2.5 m_b_ps 378
6.180.2.6 m_c_kin 378

6.180.2.7 m_c_msbar . . . 378
6.180.2.8 m_c_pole . . . 378
6.180.2.9 m_s_msbar . . . 378
6.180.2.10 m_t_msbar . . . 378
6.180.2.11 m_t_pole . . . 378
6.180.2.12 m_ud_msbar . . . 378
6.181 eos::EquationSolver::Solution Struct Reference . . . 379
6.181.1 Member Data Documentation . . . 379
6.181.1.1 parameters . . . 379
6.181.1.2 valid . . . 379
6.181.1.3 value . . . 379
6.182 eos::StandardModel Class Reference . . . 380
6.182.1 Constructor & Destructor Documentation . . . 380
6.182.1.1 StandardModel . . . 380
6.182.1.2 ~StandardModel . . . 380
6.182.2 Member Function Documentation . . . 380
6.182.2.1 make . . . 380
6.183 eos::MarkovChain::State Struct Reference . . . 381
6.183.1 Detailed Description . . . 381
6.183.2 Member Typedef Documentation . . . 382
6.183.2.1 Iterator . . . 382
6.183.3 Constructor & Destructor Documentation . . . 382
6.183.3.1 State . . . 382
6.183.3.2 State . . . 382
6.183.4 Member Data Documentation . . . 382
6.183.4.1 hyper_parameter . . . 382
6.183.4.2 log_likelihood . . . 382
6.183.4.3 log_posterior . . . 382
6.183.4.4 log_prior . . . 382
6.183.4.5 point . . . 382
6.184 eos::MarkovChain::Stats Struct Reference . . . 383
6.184.1 Detailed Description . . . 383
6.184.2 Member Data Documentation . . . 383
6.184.2.1 iterations_accepted . . . 383
6.184.2.2 iterations_invalid . . . 383
6.184.2.3 iterations_rejected . . . 383

6.184.2.4 iterations_total . . . 384
6.184.2.5 mean_of_parameters . . . 384
6.184.2.6 mean_of_posterior . . . 384
6.184.2.7 mode_of_posterior . . . 384
6.184.2.8 parameters_at_mode . . . 384
6.184.2.9 variance_of_parameters . . . 384
6.184.2.10 variance_of_posterior . . . 384
6.185 eos::PopulationMonteCarloSampler::Status Struct Reference . . . 385
6.185.1 Detailed Description . . . 385
6.185.2 Constructor & Destructor Documentation . . . 385
6.185.2.1 Status . . . 385
6.185.3 Member Data Documentation . . . 385
6.185.3.1 chunk_size . . . 385
6.185.3.2 converged . . . 385
6.185.3.3 eff_sample_size . . . 385
6.185.3.4 evidence . . . 386
6.185.3.5 iterations_at_convergence . . . 386
6.185.3.6 perplexity . . . 386
6.186 eos::TestObservable Struct Reference . . . 387
6.186.1 Constructor & Destructor Documentation . . . 387
6.186.1.1 TestObservable . . . 387
6.186.1.2 ∼TestObservable . . . 387
6.186.2 Member Function Documentation . . . 387
6.186.2.1 clone . . . 387
6.186.2.2 clone . . . 387
6.186.2.3 evaluate . . . 388
6.186.2.4 kinematics . . . 388
6.186.2.5 name . . . 388
6.186.2.6 options . . . 388
6.186.2.7 parameters . . . 388
6.186.2.8 set_option . . . 388
6.186.3 Member Data Documentation . . . 388
6.186.3.1 k . . . 388
6.186.3.2 mass . . . 388
6.186.3.3 mass_name . . . 388
6.186.3.4 n . . . 388

6.186.3.5 o . . . 388
6.186.3.6 p . . . 388
6.187 eos::Thread Class Reference . . . 389
6.187.1 Detailed Description . . . 389
6.187.2 Member Typedef Documentation . . . 389
6.187.2.1 Function . . . 389
6.187.3 Constructor & Destructor Documentation . . . 389
6.187.3.1 Thread . . . 389
6.187.3.2 ∼Thread . . . 390
6.187.4 Member Function Documentation . . . 390
6.187.4.1 completed . . . 390
6.188 eos::ThreadPool Class Reference . . . 391
6.188.1 Constructor & Destructor Documentation . . . 391
6.188.1.1 ThreadPool . . . 391
6.188.1.2 ∼ThreadPool . . . 391
6.188.2 Member Function Documentation . . . 391
6.188.2.1 enqueue . . . 391
6.188.2.2 instance . . . 391
6.188.2.3 number_of_threads . . . 391
6.188.2.4 wait_for_free_capacity . . . 391
6.189 eos::Ticket Class Reference . . . 392
6.189.1 Detailed Description . . . 392
6.189.2 Constructor & Destructor Documentation . . . 392
6.189.2.1 Ticket . . . 392
6.189.2.2 ∼Ticket . . . 392
6.189.3 Member Function Documentation . . . 392
6.189.3.1 mark . . . 392
6.189.3.2 wait . . . 393
6.189.4 Friends And Related Function Documentation . . . 393
6.189.4.1 TicketList . . . 393
6.190 eos::TicketList Class Reference . . . 394
6.190.1 Detailed Description . . . 394
6.190.2 Constructor & Destructor Documentation . . . 394
6.190.2.1 TicketList . . . 394
6.190.2.2 ∼TicketList . . . 394
6.190.3 Member Function Documentation . . . 394

6.190.3.1 push_back . . . 394
6.190.3.2 wait . . . 394
6.191 eos::TopLoops Struct Reference . . . 396
6.191.1 Detailed Description . . . 396
6.191.2 Member Function Documentation . . . 396
6.191.2.1 A0 . . . 396
6.191.2.2 A1 . . . 396
6.191.2.3 B0 . . . 396
6.191.2.4 B1 . . . 397
6.191.2.5 C0 . . . 397
6.191.2.6 C1 . . . 397
6.191.2.7 D0 . . . 397
6.191.2.8 D1 . . . 397
6.191.2.9 E0 . . . 397
6.191.2.10 E1 . . . 398
6.191.2.11 F0 . . . 398
6.191.2.12 F1 . . . 398
6.191.2.13 G1 . . . 398
6.192 eos::TryLock Class Reference . . . 399
6.192.1 Constructor & Destructor Documentation . . . 399
6.192.1.1 TryLock . . . 399
6.192.1.2 ∼TryLock . . . 399
6.192.2 Member Function Documentation . . . 399
6.192.2.1 operator() . . . 399
6.193 eos::deriv::TwoSided Class Reference . . . 400
6.194 eos::hdf5::Type Struct Reference . . . 401
6.194.1 Constructor & Destructor Documentation . . . 401
6.194.1.1 ∼Type . . . 401
6.194.2 Member Function Documentation . . . 401
6.194.2.1 copy_from_hdf5 . . . 401
6.194.2.2 copy_to_hdf5 . . . 401
6.194.2.3 name . . . 401
6.194.2.4 size . . . 401
6.194.2.5 type_id . . . 401
6.195 eos::UnknownConstraintError Struct Reference . . . 403
6.195.1 Detailed Description . . . 403

6.195.2 Constructor & Destructor Documentation 403
6.195.2.1 UnknownConstraintError 403
6.196eos::UnknownKinematicVariableError Struct Reference 404
6.196.1 Detailed Description 404
6.196.2 Constructor & Destructor Documentation 404
6.196.2.1 UnknownKinematicVariableError 404
6.197eos::UnknownOptionError Struct Reference 405
6.197.1 Detailed Description 405
6.197.2 Constructor & Destructor Documentation 405
6.197.2.1 UnknownOptionError 405
6.198eos::UnknownParameterError Struct Reference 406
6.198.1 Detailed Description 406
6.198.2 Constructor & Destructor Documentation 406
6.198.2.1 UnknownParameterError 406
6.199eos::proposal_functions::UnknownProposalError Struct Reference 407
6.199.1 Detailed Description 407
6.199.2 Constructor & Destructor Documentation 407
6.199.2.1 UnknownProposalError 407
6.200eos::UsedParameter Class Reference 408
6.200.1 Detailed Description 408
6.200.2 Constructor & Destructor Documentation 408
6.200.2.1 UsedParameter 408
6.201eos::VerifiedRange< T_ > Class Template Reference 409
6.201.1 Detailed Description 409
6.201.2 Constructor & Destructor Documentation 409
6.201.2.1 VerifiedRange 409
6.201.3 Member Function Documentation 409
6.201.3.1 max 409
6.201.3.2 min 410
6.201.3.3 operator T_ 410
6.201.3.4 operator= 410
6.202eos::VerifiedRangeError Class Reference 411
6.202.1 Detailed Description 411
6.202.2 Constructor & Destructor Documentation 411
6.202.2.1 VerifiedRangeError 411
6.203eos::VerifiedRangeOverflow Struct Reference 412

6.203.1 Detailed Description 412
6.203.2 Constructor & Destructor Documentation 412
6.203.2.1 VerifiedRangeOverflow 412
6.204 eos::VerifiedRangeUnderflow Struct Reference 413
6.204.1 Detailed Description 413
6.204.2 Constructor & Destructor Documentation 413
6.204.2.1 VerifiedRangeUnderflow 413
6.205 eos::Welford Struct Reference 414
6.205.1 Detailed Description 414
6.205.2 Constructor & Destructor Documentation 414
6.205.2.1 Welford 414
6.205.3 Member Function Documentation 414
6.205.3.1 add 414
6.205.3.2 mean 414
6.205.3.3 number_of_elements 414
6.205.3.4 std_deviation 414
6.205.3.5 variance 414
6.206 eos::WilsonCoefficients< BToS > Struct Template Reference 415
6.206.1 Constructor & Destructor Documentation 415
6.206.1.1 WilsonCoefficients 415
6.206.2 Member Function Documentation 417
6.206.2.1 c1 417
6.206.2.2 c10 417
6.206.2.3 c10prime 417
6.206.2.4 c2 417
6.206.2.5 c2b 417
6.206.2.6 c3 417
6.206.2.7 c4 417
6.206.2.8 c5 417
6.206.2.9 c6 417
6.206.2.10 c7 417
6.206.2.11 c7prime 417
6.206.2.12 c8 417
6.206.2.13 c8prime 417
6.206.2.14 c9 417
6.206.2.15 c9prime 417

6.206.2.16cP . . . 417
6.206.2.17cPprime . . . 417
6.206.2.18cq3 . . . 417
6.206.2.19cq4 . . . 417
6.206.2.20cq5 . . . 417
6.206.2.21cq6 . . . 417
6.206.2.22cS . . . 417
6.206.2.23cSprime . . . 417
6.206.2.24cT . . . 417
6.206.2.25cT5 . . . 417
6.206.3 Member Data Documentation . . . 417
6.206.3.1 _alpha_s . . . 417
6.206.3.2 _primed_coefficients . . . 417
6.206.3.3 _scalar_tensor_coefficients . . . 417
6.206.3.4 _sm_like_coefficients . . . 417
6.207eos::WilsonPolynomialCloner Class Reference . . . 419
6.207.1 Constructor & Destructor Documentation . . . 419
6.207.1.1 WilsonPolynomialCloner . . . 419
6.207.2 Member Function Documentation . . . 419
6.207.2.1 visit . . . 419
6.207.2.2 visit . . . 419
6.207.2.3 visit . . . 419
6.207.2.4 visit . . . 419
6.207.2.5 visit . . . 419
6.207.2.6 visit . . . 419
6.208eos::WilsonPolynomialEvaluator Class Reference . . . 420
6.208.1 Member Function Documentation . . . 420
6.208.1.1 visit . . . 420
6.208.1.2 visit . . . 420
6.208.1.3 visit . . . 420
6.208.1.4 visit . . . 420
6.208.1.5 visit . . . 420
6.208.1.6 visit . . . 420
6.209eos::WilsonPolynomialPrinter Class Reference . . . 421
6.209.1 Constructor & Destructor Documentation . . . 421
6.209.1.1 WilsonPolynomialPrinter . . . 421

6.209.2 Member Function Documentation 421
6.209.2.1 visit 421
6.209.2.2 visit 421
6.209.2.3 visit 421
6.209.2.4 visit 421
6.209.2.5 visit 421
6.209.2.6 visit 421
6.210eos::WilsonScanComponent Class Reference 422
6.210.1 Constructor & Destructor Documentation 422
6.210.1.1 WilsonScanComponent 422
6.210.2 Member Function Documentation 422
6.210.2.1 wilson_coefficients_b_to_s 422
6.211eos::WilsonScanModel Class Reference 423
6.211.1 Constructor & Destructor Documentation 423
6.211.1.1 WilsonScanModel 423
6.211.1.2 ∼WilsonScanModel 423
6.211.2 Member Function Documentation 423
6.211.2.1 make 423
6.212eos::WrappedForwardIterator< Tag_, Value_ > Class Template Reference 424
6.212.1 Detailed Description 425
6.212.2 Member Typedef Documentation 427
6.212.2.1 difference_type 427
6.212.2.2 iterator_category 427
6.212.2.3 pointer 427
6.212.2.4 reference 427
6.212.2.5 value_type 427
6.212.3 Constructor & Destructor Documentation 427
6.212.3.1 WrappedForwardIterator 427
6.212.3.2 ∼WrappedForwardIterator 427
6.212.3.3 WrappedForwardIterator 427
6.212.3.4 WrappedForwardIterator 427
6.212.4 Member Function Documentation 427
6.212.4.1 operator!= 427
6.212.4.2 operator∗ 427
6.212.4.3 operator++ 427
6.212.4.4 operator++ 427

6.212.4.5 operator-> . . . 427
6.212.4.6 operator= . . . 427
6.212.4.7 operator== . . . 427
6.212.4.8 underlying_iterator . . . 427
6.212.4.9 underlying_iterator . . . 427

7 File Documentation **429**

7.1 /home/alexshires/EOS/eos-next/eos/b-decays/b-to-pi-l-nu.hh File Reference . . . 429
7.1.1 Define Documentation . . . 429
7.1.1.1 EOS_GUARD_EOS_B_DECAYS_EXCLUSIVE_B_TO_U_L_NU_HH 429
7.2 /home/alexshires/EOS/eos-next/eos/b-decays/properties.hh File Reference . . . 430
7.2.1 Define Documentation . . . 430
7.2.1.1 EOS_GUARD_EOS_B_DECAYS_PROPERTIES_HH . . . 430
7.3 /home/alexshires/EOS/eos-next/eos/constraint.hh File Reference . . . 431
7.3.1 Define Documentation . . . 431
7.3.1.1 EOS_GUARD_SRC_CONSTRAINT_HH . . . 431
7.4 /home/alexshires/EOS/eos-next/eos/form-factors/analytic-b-to-pi.hh File Reference . . . 432
7.4.1 Define Documentation . . . 432
7.4.1.1 EOS_GUARD_EOS_FORM_FACTORS_ANALYTIC_B_TO_PI_HH . 432
7.5 /home/alexshires/EOS/eos-next/eos/form-factors/baryonic-impl.hh File Reference . . . 433
7.5.1 Define Documentation . . . 433
7.5.1.1 EOS_GUARD_EOS_FORM_FACTORS_BARYONIC_IMPL_HH . . . 433
7.6 /home/alexshires/EOS/eos-next/eos/form-factors/baryonic.hh File Reference . . . 434
7.6.1 Define Documentation . . . 434
7.6.1.1 EOS_GUARD_EOS_FORM_FACTORS_BARYONIC_HH . . . 434
7.7 /home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh File Reference . . . 435
7.7.1 Define Documentation . . . 435
7.7.1.1 EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTOR_-ADAPTER_HH . . . 435
7.8 /home/alexshires/EOS/eos-next/eos/form-factors/form-factors-fwd.hh File Reference . . . 436
7.8.1 Define Documentation . . . 436
7.8.1.1 EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTORS_FWD_HH 436
7.9 /home/alexshires/EOS/eos-next/eos/form-factors/form-factors.hh File Reference . . . 437
7.9.1 Define Documentation . . . 437
7.9.1.1 EOS_GUARD_SRC_FORM_FACTORS_FORM_FACTORS_HH . . . 437
7.10 /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh File Reference . . . 438
7.10.1 Define Documentation . . . 438

7.10.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_FORM_FACTORS_-IMPL_HH . . . 438
7.11 /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh File Reference . . . 439
7.11.1 Define Documentation . . . 439
7.11.1.1 EOS_GUARD_SRC_FORM_FACTORS_MESONIC_HH . . . 439
7.12 /home/alexshires/EOS/eos-next/eos/form-factors/pi-lcdas.hh File Reference . . . 440
7.12.1 Define Documentation . . . 440
7.12.1.1 EOS_GUARD_EOS_FORM_FACTORS_PI_LCDAS_HH . . . 440
7.13 /home/alexshires/EOS/eos-next/eos/observable.hh File Reference . . . 441
7.13.1 Define Documentation . . . 441
7.13.1.1 EOS_GUARD_SRC_UTILS_OBSERVABLE_HH . . . 441
7.14 /home/alexshires/EOS/eos-next/eos/rare-b-decays/bremsstrahlung.hh File Reference . . . 442
7.14.1 Define Documentation . . . 442
7.14.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_BREMSSTRAHLUNG_HH 442
7.15 /home/alexshires/EOS/eos-next/eos/rare-b-decays/charm-loops.hh File Reference . . . 443
7.15.1 Define Documentation . . . 443
7.15.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_CHARM_LOOPS_HH . . . 443
7.16 /home/alexshires/EOS/eos-next/eos/rare-b-decays/decays.hh File Reference . . . 444
7.16.1 Define Documentation . . . 444
7.16.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_DECAYS_HH . . . 444
7.17 /home/alexshires/EOS/eos-next/eos/rare-b-decays/em-contributions.hh File Reference . . . 445
7.17.1 Define Documentation . . . 445
7.17.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_EM_CONTRIBUTIONS_HH 445
7.18 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-dilepton.hh File Reference 446
7.18.1 Define Documentation . . . 446
7.18.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_-DILEPTON_HH . . . 446
7.19 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-large-recoil.hh File Reference . . . 447
7.19.1 Define Documentation . . . 447
7.19.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_-DILEPTON_LARGE_RECOIL_HH . . . 447
7.20 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-low-recoil.hh File Reference . . . 448
7.20.1 Define Documentation . . . 448
7.20.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_-DILEPTON_LOW_RECOIL_HH . . . 448
7.21 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton.hh File Reference . . . 449

7.21.1 Define Documentation 449
7.21.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_-DILEPTON_HH 449
7.22 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-gamma.hh File Reference 450
7.22.1 Define Documentation 450
7.22.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_-GAMMA_HH 450
7.23 /home/alexshires/EOS/eos-next/eos/rare-b-decays/hard-scattering.hh File Reference 451
7.23.1 Define Documentation 451
7.23.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_HARD_SCATTERING_HH 451
7.24 /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-dilepton.hh File Reference 452
7.24.1 Define Documentation 452
7.24.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_INCLUSIVE_B_TO_S_-DILEPTON_HH 452
7.25 /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh File Reference 453
7.25.1 Define Documentation 453
7.25.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_INCLUSIVE_B_TO_S_-GAMMA_HH 453
7.26 /home/alexshires/EOS/eos-next/eos/rare-b-decays/lambda-b-to-lambda-dilepton.hh File Reference 454
7.26.1 Define Documentation 454
7.26.1.1 EOS_GUARD_EOS_RARE_B_DECAYS_BARYONIC_B_TO_S_-DILEPTON_HH 454
7.27 /home/alexshires/EOS/eos-next/eos/rare-b-decays/long-distance.hh File Reference 455
7.27.1 Define Documentation 455
7.27.1.1 EOS_GUARD_SRC_RARE_B_DECAYS_LONG_DISTANCE_HH . . 455
7.28 /home/alexshires/EOS/eos-next/eos/rare-b-decays/qcdf_integrals.hh File Reference 456
7.28.1 Define Documentation 456
7.28.1.1 EOS_GUARD_EOS_RARE_B_DECAYS_QCDF_INTEGRALS_HH . 456
7.29 /home/alexshires/EOS/eos-next/eos/utils/accumulator.hh File Reference 457
7.29.1 Define Documentation 457
7.29.1.1 EOS_GUARD_SRC_UTILS_ACCUMULATOR_HH 457
7.30 /home/alexshires/EOS/eos-next/eos/utils/analysis-fwd.hh File Reference 458
7.30.1 Define Documentation 458
7.30.1.1 EOS_GUARD_SRC_UTILS_ANALYSIS_FWD_HH 458
7.31 /home/alexshires/EOS/eos-next/eos/utils/analysis.hh File Reference 459
7.31.1 Define Documentation 459
7.31.1.1 EOS_GUARD_SRC_UTILS_ANALYSIS_HH 459

7.32 /home/alexshires/EOS/eos-next/eos/utils/analysis_TEST.hh File Reference 460
7.32.1 Define Documentation 460
7.32.1.1 EOS_GUARD_SRC_UTILS_ANALYSIS_TEST_HH 460
7.33 /home/alexshires/EOS/eos-next/eos/utils/apply.hh File Reference 461
7.33.1 Define Documentation 461
7.33.1.1 EOS_GUARD_SRC_UTILS_APPLY_HH 461
7.34 /home/alexshires/EOS/eos-next/eos/utils/cartesian-product.hh File Reference 462
7.34.1 Define Documentation 462
7.34.1.1 EOS_GUARD_SRC_UTILS_CARTESIAN_PRODUCT_HH 462
7.35 /home/alexshires/EOS/eos-next/eos/utils/chi-squared.hh File Reference 463
7.35.1 Define Documentation 463
7.35.1.1 EOS_GUARD_SRC_UTILS_CHI_SQUARED_HH 463
7.36 /home/alexshires/EOS/eos-next/eos/utils/ckm_scan_model.hh File Reference 464
7.36.1 Define Documentation 464
7.36.1.1 EOS_GUARD_SRC_UTILS_CKM_SCAN_MODEL_HH 464
7.37 /home/alexshires/EOS/eos-next/eos/utils/cluster.hh File Reference 465
7.37.1 Define Documentation 465
7.37.1.1 EOS_GUARD_EOS_UTILS_CLUSTER_HH 465
7.38 /home/alexshires/EOS/eos-next/eos/utils/complex.hh File Reference 466
7.38.1 Define Documentation 466
7.38.1.1 WILSON_FITTER_GUARD_SRC_UTILS_COMPLEX_HH 466
7.39 /home/alexshires/EOS/eos-next/eos/utils/concrete_observable.hh File Reference 467
7.39.1 Define Documentation 467
7.39.1.1 EOS_GUARD_SRC_UTILS_CONCRETE_OBSERVABLE_HH 467
7.40 /home/alexshires/EOS/eos-next/eos/utils/condition_variable.hh File Reference 468
7.40.1 Define Documentation 468
7.40.1.1 EOS_GUARD_SRC_UTILS_CONDITION_VARIABLE_HH 468
7.41 /home/alexshires/EOS/eos-next/eos/utils/derivative.hh File Reference 469
7.41.1 Define Documentation 469
7.41.1.1 BTOPI_FF_GUARD_EOS_UTILS_DERIVATIVE_HH 469
7.42 /home/alexshires/EOS/eos-next/eos/utils/destringify.hh File Reference 470
7.42.1 Define Documentation 470
7.42.1.1 EOS_GUARD_SRC_UTILS_DESTRINGIFY_HH 470
7.43 /home/alexshires/EOS/eos-next/eos/utils/diagnostics.hh File Reference 471
7.43.1 Define Documentation 471
7.43.1.1 EOS_GUARD_EOS_UTILS_DIAGNOSTICS_HH 471

7.44 /home/alexshires/EOS/eos-next/eos/utils/equation_solver.hh File Reference 472
7.44.1 Define Documentation 472
7.44.1.1 EOS_GUARD_EOS_UTILS_EQUATION_SOLVER_HH 472
7.45 /home/alexshires/EOS/eos-next/eos/utils/exception.hh File Reference 473
7.45.1 Define Documentation 473
7.45.1.1 EOS_GUARD_SRC_UTILS_EXCEPTION_HH 473
7.46 /home/alexshires/EOS/eos-next/eos/utils/hdf5-fwd.hh File Reference 474
7.46.1 Define Documentation 474
7.46.1.1 EOS_GUARD_SRC_UTILS_HDF5_FWD_HH 474
7.47 /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh File Reference 475
7.47.1 Define Documentation 476
7.47.1.1 EOS_GUARD_SRC_UTILS_HDF5_HH 476
7.48 /home/alexshires/EOS/eos-next/eos/utils/hierarchical-clustering.hh File Reference 477
7.48.1 Define Documentation 477
7.48.1.1 EOS_GUARD_SRC_UTILS_HIERARCHICAL_CLUSTERING_HH . 477
7.49 /home/alexshires/EOS/eos-next/eos/utils/histogram.hh File Reference 478
7.49.1 Define Documentation 478
7.49.1.1 EOS_GUARD_SRC_UTILS_HISTROGRAM_HH 478
7.50 /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy-impl.hh File Reference . . . 479
7.50.1 Define Documentation 479
7.50.1.1 EOS_GUARD_SRC_UTILS_INSTANTIATION_POLICY_IMPL_HH . 479
7.50.2 Variable Documentation 479
7.50.2.1 _ptr 479
7.51 /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy.hh File Reference 480
7.51.1 Define Documentation 480
7.51.1.1 EOS_GUARD_SRC_UTILS_INSTANTIATION_POLICY_HH 480
7.52 /home/alexshires/EOS/eos-next/eos/utils/integrate-impl.hh File Reference 481
7.52.1 Define Documentation 481
7.52.1.1 EOS_GUARD_EOS_UTILS_INTEGRATE_IMPL_HH 481
7.53 /home/alexshires/EOS/eos-next/eos/utils/integrate.hh File Reference 482
7.53.1 Define Documentation 482
7.53.1.1 EOS_GUARD_SRC_UTILS_INTEGRATE_HH 482
7.54 /home/alexshires/EOS/eos-next/eos/utils/kinematic.hh File Reference 483
7.54.1 Define Documentation 483
7.54.1.1 EOS_GUARD_SRC_UTILS_KINEMATIC_HH 483
7.55 /home/alexshires/EOS/eos-next/eos/utils/lcda.hh File Reference 484

7.55.1 Define Documentation 484
7.55.1.1 EOS_GUARD_SRC_UTILS_LCDA_HH 484
7.56 /home/alexshires/EOS/eos-next/eos/utils/lock.hh File Reference 485
7.56.1 Define Documentation 485
7.56.1.1 EOS_GUARD_SRC_UTILS_LOCK_HH 485
7.57 /home/alexshires/EOS/eos-next/eos/utils/log.hh File Reference 486
7.57.1 Define Documentation 486
7.57.1.1 EOS_GUARD_SRC_UTILS_LOG_HH 486
7.58 /home/alexshires/EOS/eos-next/eos/utils/log_likelihood-fwd.hh File Reference 487
7.58.1 Define Documentation 487
7.58.1.1 EOS_GUARD_SRC_UTILS_LOG_LIKELIHOOD_FWD_HH 487
7.59 /home/alexshires/EOS/eos-next/eos/utils/log_likelihood.hh File Reference 488
7.59.1 Define Documentation 488
7.59.1.1 EOS_GUARD_SRC_UTILS_LIKELIHOOD_HH 488
7.60 /home/alexshires/EOS/eos-next/eos/utils/log_prior-fwd.hh File Reference 489
7.60.1 Define Documentation 489
7.60.1.1 EOS_GUARD_SRC_UTILS_PRIOR_FWD_HH 489
7.61 /home/alexshires/EOS/eos-next/eos/utils/log_prior.hh File Reference 490
7.61.1 Define Documentation 490
7.61.1.1 EOS_GUARD_SRC_UTILS_PRIOR_HH 490
7.62 /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh File Reference 491
7.62.1 Define Documentation 491
7.62.1.1 EOS_GUARD_SRC_UTILS_MARKOV_CHAIN_HH 491
7.63 /home/alexshires/EOS/eos-next/eos/utils/markov_chain_sampler.hh File Reference 492
7.63.1 Define Documentation 492
7.63.1.1 EOS_GUARD_SRC_UTILS_MC_SAMPLER_HH 492
7.64 /home/alexshires/EOS/eos-next/eos/utils/matrix.hh File Reference 493
7.64.1 Define Documentation 494
7.64.1.1 EOS_GUARD_SRC_UTILS_MATRIX_HH 494
7.65 /home/alexshires/EOS/eos-next/eos/utils/memoise.hh File Reference 495
7.65.1 Define Documentation 495
7.65.1.1 EOS_GUARD_SRC_UTILS_MEMOISE_HH 495
7.66 /home/alexshires/EOS/eos-next/eos/utils/model.hh File Reference 496
7.66.1 Define Documentation 496
7.66.1.1 EOS_GUARD_SRC_UTILS_MODEL_HH 496
7.67 /home/alexshires/EOS/eos-next/eos/utils/mutable-fwd.hh File Reference 497

7.67.1 Define Documentation 497
7.67.1.1 EOS_GUARD_EOS_UTILS_MUTABLE_FWD_HH 497
7.68 /home/alexshires/EOS/eos-next/eos/utils/mutable.hh File Reference 498
7.68.1 Define Documentation 498
7.68.1.1 EOS_GUARD_EOS_UTILS_MUTABLE_HH 498
7.69 /home/alexshires/EOS/eos-next/eos/utils/mutex.hh File Reference 499
7.69.1 Define Documentation 499
7.69.1.1 EOS_GUARD_SRC_UTILS_MUTEX_HH 499
7.70 /home/alexshires/EOS/eos-next/eos/utils/observable_cache.hh File Reference 500
7.70.1 Define Documentation 500
7.70.1.1 EOS_GUARD_SRC_UTILS_OBSERVABLE_CACHE_HH 500
7.71 /home/alexshires/EOS/eos-next/eos/utils/observable_set.hh File Reference 501
7.71.1 Define Documentation 501
7.71.1.1 EOS_GUARD_SRC_UTILS_OBSERVABLE_SET_HH 501
7.72 /home/alexshires/EOS/eos-next/eos/utils/one-of.hh File Reference 502
7.72.1 Define Documentation 502
7.72.1.1 EOS_GUARD_SRC_UTILS_ONE_OF_HH 502
7.73 /home/alexshires/EOS/eos-next/eos/utils/options.hh File Reference 503
7.73.1 Define Documentation 503
7.73.1.1 EOS_GUARD_SRC_UTILS_OPTIONS_HH 503
7.74 /home/alexshires/EOS/eos-next/eos/utils/parameters-fwd.hh File Reference 504
7.74.1 Define Documentation 504
7.74.1.1 EOS_GUARD_EOS_UTILS_PARAMETERS_FWD_HH 504
7.75 /home/alexshires/EOS/eos-next/eos/utils/parameters.hh File Reference 505
7.75.1 Define Documentation 505
7.75.1.1 EOS_GUARD_SRC_UTILS_PARAMETERS_HH 505
7.76 /home/alexshires/EOS/eos-next/eos/utils/polylog.hh File Reference 506
7.76.1 Define Documentation 506
7.76.1.1 EOS_GUARD_EOS_UTILS_POLYLOG_HH 506
7.77 /home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_sampler.hh File Reference 507
7.77.1 Define Documentation 507
7.77.1.1 EOS_GUARD_SRC_UTILS_PMC_SAMPLER_HH 507
7.78 /home/alexshires/EOS/eos-next/eos/utils/power_of.hh File Reference 508
7.78.1 Define Documentation 508
7.78.1.1 EOS_GUARD_SRC_UTILS_POWER_OF_HH 508
7.79 /home/alexshires/EOS/eos-next/eos/utils/prior_sampler.hh File Reference 509

7.79.1 Define Documentation 509
7.79.1.1 EOS_GUARD_SRC_UTILS_PRIOR_SAMPLER_HH 509
7.80 /home/alexshires/EOS/eos-next/eos/utils/private_implementation_pattern-impl.hh File Reference 510
7.80.1 Define Documentation 510
7.80.1.1 WILSON_FITTER_GUARD_UTILS_PRIVATE_-IMPLEMENTATION_PATTERN_IMPL_HH 510
7.81 /home/alexshires/EOS/eos-next/eos/utils/private_implementation_pattern.hh File Reference 511
7.81.1 Define Documentation 511
7.81.1.1 WILSON_FITTER_GUARD_UTILS_PRIVATE_-IMPLEMENTATION_PATTERN_HH 511
7.82 /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh File Reference 512
7.82.1 Define Documentation 513
7.82.1.1 EOS_GUARD_EOS_UTILS_PROPROSAL_FUNCTIONS_HH 513
7.83 /home/alexshires/EOS/eos-next/eos/utils/qcd.hh File Reference 514
7.83.1 Define Documentation 514
7.83.1.1 EOS_GUARD_SRC_UTILS_QCD_HH 514
7.84 /home/alexshires/EOS/eos-next/eos/utils/random_number_generator.hh File Reference . . 515
7.84.1 Define Documentation 515
7.84.1.1 EOS_GUARD_SRC_UTILS_RANDOM_NUMBER_GENERATOR_HH 515
7.85 /home/alexshires/EOS/eos-next/eos/utils/rvalue.hh File Reference 516
7.85.1 Define Documentation 516
7.85.1.1 EOS_GUARD_SRC_UTILS_RVALUE_HH 516
7.86 /home/alexshires/EOS/eos-next/eos/utils/save.hh File Reference 517
7.86.1 Define Documentation 517
7.86.1.1 EOS_GUARD_SRC_UTILS_SAVE_HH 517
7.87 /home/alexshires/EOS/eos-next/eos/utils/standard-model.hh File Reference 518
7.87.1 Define Documentation 518
7.87.1.1 EOS_GUARD_SRC_UTILS_STANDARD_MODEL_HH 518
7.88 /home/alexshires/EOS/eos-next/eos/utils/stringify.hh File Reference 519
7.88.1 Define Documentation 519
7.88.1.1 EOS_GUARD_SRC_UTILS_STRINGIFY_HH 519
7.89 /home/alexshires/EOS/eos-next/eos/utils/thread.hh File Reference 520
7.89.1 Define Documentation 520
7.89.1.1 EOS_GUARD_SRC_UTILS_THREAD_HH 520
7.90 /home/alexshires/EOS/eos-next/eos/utils/thread_pool.hh File Reference 521
7.90.1 Define Documentation 521

7.90.1.1 EOS_GUARD_SRC_UTILS_THREAD_POOL_HH 521
7.91 /home/alexshires/EOS/eos-next/eos/utils/ticket.hh File Reference 522
7.91.1 Define Documentation 522
7.91.1.1 EOS_GUARD_SRC_UTILS_TICKET_HH 522
7.92 /home/alexshires/EOS/eos-next/eos/utils/top-loops.hh File Reference 523
7.92.1 Define Documentation 523
7.92.1.1 EOS_GUARD_SRC_UTILS_TOP_LOOPS_HH 523
7.93 /home/alexshires/EOS/eos-next/eos/utils/verify.hh File Reference 524
7.93.1 Define Documentation 524
7.93.1.1 EOS_GUARD_SRC_UTILS_VERIFY_HH 524
7.94 /home/alexshires/EOS/eos-next/eos/utils/welford.hh File Reference 525
7.94.1 Define Documentation 525
7.94.1.1 EOS_GUARD_EOS_UTILS_WELFORD_HH 525
7.95 /home/alexshires/EOS/eos-next/eos/utils/wilson-polynomial.hh File Reference 526
7.95.1 Define Documentation 526
7.95.1.1 EOS_GUARD_SRC_UTILS_WILSON_POLYNOMIAL_HH 526
7.96 /home/alexshires/EOS/eos-next/eos/utils/wilson_coefficients.hh File Reference 527
7.96.1 Define Documentation 527
7.96.1.1 EOS_GUARD_SRC_UTILS_WILSON_COEFFICIENTS_HH 527
7.97 /home/alexshires/EOS/eos-next/eos/utils/wilson_scan_model.hh File Reference 528
7.97.1 Define Documentation 528
7.97.1.1 EOS_GUARD_SRC_UTILS_WILSON_SCAN_MODEL_HH 528
7.98 /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-fwd.hh File Reference 529
7.98.1 Define Documentation 529
7.98.1.1 EOS_GUARD_UTIL_WRAPPED_FORWARD_ITERATOR_FWD_HH 529
7.99 /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-impl.hh File Reference 530
7.99.1 Define Documentation 530
7.99.1.1 EOS_GUARD_UTIL_WRAPPED_FORWARD_ITERATOR_IMPL_HH 530
7.100/home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator.hh File Reference . . . 531
7.100.1 Define Documentation 531
7.100.1.1 EOS_GUARD_UTILS_WRAPPED_FORWARD_ITERATOR_HH . . . 531

8 Example Documentation 533
8.1 For 533
8.2 if 534
8.3 index_list 535
8.4 print_steps 536

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

eos 21
eos::btovll 34
eos::components 35
eos::deriv 36
eos::hdf5 37
eos::impl 39
eos::implementation 40
eos::InstantiationPolicy 41
eos::proposal_functions 42
ROOT 44
ROOT::Minuit2 45
std (STL namespace) 46

Chapter 2

Class Index

2.1 Class Hierarchy

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

eos::proposal_functions::AdjacencyMatrix 54
eos::btovll::Amplitudes 56
eos::btovll::AngularCoefficients 64
eos::hdf5::Attribute< T_ > 67
eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ > 69
  eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ > 71
eos::BCL2008FormFactors< Process_ > 73
eos::Histogram< 2 >::Bin 75
eos::Histogram< 1 >::Bin 77
eos::Bremsstrahlung 83
eos::BsToPhi 85
eos::BToK 87
eos::BToKstar 94
eos::BToPi 106
eos::BToS 108
eos::CartesianProduct< T_ > 116
eos::CharmLoops 118
eos::ChiSquared 122
eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ > 132
eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ > 133
eos::ConditionVariable 137
eos::MarkovChainSampler::Config 139
eos::EquationSolver::Config 145
eos::PopulationMonteCarloSampler::Config 146
eos::PriorSampler::Config 154
eos::proposal_functions::GlobalLocal::Config 156
eos::HierarchicalClustering::Config 160
eos::hdf5::DataSet< T_ > 164
eos::hdf5::DataType< T_ > 169
eos::hdf5::DataType< const char * > 170
eos::hdf5::DataType< double > 171
eos::hdf5::DataType< int > 172
eos::hdf5::DataType< short > 173

eos::hdf5::DataType< unsigned > . . . 174
eos::InstantiationPolicy::DeleteOnDestruction< T_ > . . . 175
eos::implementation::DoStringify< T_ > . . . 178
eos::EMContributions . . . 179
eos::Diagnostics::Entry . . . 180
eos::Exception . . . 183
  eos::HDF5Error . . . 209
  eos::InternalError . . . 226
  eos::NoSuchChiSquaredError . . . 298
  eos::NoSuchModelError . . . 299
  eos::ObservableNameError . . . 306
  eos::proposal_functions::UnknownProposalError . . . 407
  eos::UnknownConstraintError . . . 403
  eos::UnknownKinematicVariableError . . . 404
  eos::UnknownOptionError . . . 405
  eos::UnknownParameterError . . . 406
  eos::VerifiedRangeError . . . 411
    eos::VerifiedRangeOverflow . . . 412
    eos::VerifiedRangeUnderflow . . . 413
eos::proposal_functions::Factory . . . 184
eos::hdf5::File . . . 185
eos::FormFactorFactory< OneHalfPlusToOneHalfPlus > . . . 192
eos::FormFactorFactory< PToP > . . . 193
eos::FormFactorFactory< PToV > . . . 194
eos::HardScattering . . . 208
eos::MarkovChain::History . . . 219
eos::HLMW2005 . . . 221
eos::MarkovChain::HyperParameter . . . 222
eos::InstantiationPolicy< T_, NonCopyable > . . . 223
eos::InstantiationPolicy< T_, NonInstantiable > . . . 224
eos::InstantiationPolicy< T_, Singleton > . . . 225
eos::KinematicVariable . . . 230
eos::LargeRecoil . . . 239
eos::LCDA . . . 240
eos::Lock . . . 241
eos::LogLikelihoodBlock . . . 248
eos::LogMessageHandler . . . 255
eos::LogPrior . . . 256
eos::LongDistance . . . 262
eos::LowRecoil . . . 263
eos::MemoisationControl . . . 273
eos::Memoiser< Result_, Params_ > . . . 274
eos::Minimal . . . 275
eos::ModelComponent< components::CKM > . . . 277
  eos::CKMScanComponent . . . 123
    eos::CKMScanModel . . . 125
  eos::Model . . . 276
    eos::CKMScanModel . . . 125
    eos::StandardModel . . . 380
    eos::WilsonScanModel . . . 423
  eos::SMComponent< components::CKM > . . . 374
    eos::StandardModel . . . 380
    eos::WilsonScanModel . . . 423

eos::ModelComponent< components::DeltaB1 > . . . 279
eos::Model . . . 276
eos::SMComponent< components::DeltaB1 > . . . 376
eos::CKMScanModel . . . 125
eos::StandardModel . . . 380
eos::WilsonScanComponent . . . 422
eos::WilsonScanModel . . . 423
eos::ModelComponent< components::QCD > . . . 280
eos::Model . . . 276
eos::SMComponent< components::QCD > . . . 377
eos::CKMScanModel . . . 125
eos::StandardModel . . . 380
eos::WilsonScanModel . . . 423
eos::proposal_functions::MultivariateAccess . . . 287
eos::Mutable . . . 294
eos::Parameter . . . 329
eos::UsedParameter . . . 408
eos::Mutex . . . 296
eos::NLO . . . 297
eos::ObservableFactory . . . 305
eos::ConcreteObservableFactory< Decay_, Args_ > . . . 136
eos::FormFactorAdapterFactory< Transition_ > . . . 191
eos::FormFactorRatioAdapterFactory< Transition_ > . . . 197
eos::OneHalfPlusToOneHalfPlus . . . 310
eos::OneOf< Types_ > . . . 311
eos::OneOfValueBase< Types_ > . . . 313
eos::OneOfValueBase< Types_...> . . . 313
eos::OneOfValue< Type_, Types_ > . . . 312
eos::OneOfVisitorVisit< Type_ > . . . 315
eos::OneOfVisitorVisit< Types_ > . . . 315
eos::OneOfVisitor< Types_ > . . . 314
eos::OneOfVisitorWrapper< Underlying_, Result_, Types_ > . . . 316
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ > . . . 317
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...> . . . 318
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void > . . . 319
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...> . . . 320
eos::Analysis::OptimizationOptions . . . 321
eos::PopulationMonteCarloSampler::Output . . . 325
eos::Analysis::Output . . . 328
eos::ParameterDescription . . . 332
eos::ParameterRange . . . 333
eos::ParameterUser . . . 337
eos::BMesonProperties . . . 82
eos::BToDilepton . . . 86
eos::BToKDilepton< LargeRecoil > . . . 88
eos::BToKDilepton< LowRecoil > . . . 91
eos::BToKstarDilepton< LargeRecoil > . . . 95
eos::BToKstarDilepton< LowRecoil > . . . 99
eos::BToKstarGamma . . . 104
eos::BToPiLeptonNeutrino . . . 107
eos::BToXsDilepton< HLMW2005 > . . . 109

eos::BToXsGamma< Minimal > . . . 110
eos::BToXsGamma< NLO > . . . 111
eos::FormFactors< OneHalfPlusToOneHalfPlus > . . . 198
eos::FormFactors< PToP > . . . 199
eos::AnalyticFormFactorBToPiDKMMO2008 . . . 62
eos::BFW2010FormFactors< PToP > . . . 74
eos::BZ2004FormFactors< Process_, PToP > . . . 112
eos::BZ2004FormFactorsSplit< Process_ > . . . 115
eos::KMPW2010FormFactors< PToP > . . . 232
eos::FormFactors< PToV > . . . 201
eos::BZ2004FormFactors< Process_, PToV > . . . 113
eos::KMPW2010FormFactors< PToV > . . . 233
eos::LambdaBToLambdaDilepton< LargeRecoil > . . . 235
eos::LambdaBToLambdaDilepton< LowRecoil > . . . 237
eos::Model . . . 276
eos::Observable . . . 300
eos::ConcreteObservable< Decay_, Args_ > . . . 134
eos::FormFactorAdapter< Transition_ > . . . 189
eos::FormFactorRatioAdapter< Transition_ > . . . 195
eos::TestObservable . . . 387
eos::AbsoluteTestObservable . . . 51
eos::PionLCDAs . . . 339
eos::MarkovChainSampler::PreRunInfo . . . 344
eos::PrivateImplementationPattern< T_ > . . . 348
eos::PrivateImplementationPattern< Accumulator > . . . 348
eos::Accumulator . . . 53
eos::PrivateImplementationPattern< Analysis > . . . 348
eos::Analysis . . . 57
eos::PrivateImplementationPattern< AnalyticFormFactorBToPiDKMMO2008 > . . . 348
eos::AnalyticFormFactorBToPiDKMMO2008 . . . 62
eos::PrivateImplementationPattern< AttributeHandle > . . . 348
eos::hdf5::AttributeHandle . . . 68
eos::PrivateImplementationPattern< BMesonProperties > . . . 348
eos::BMesonProperties . . . 82
eos::PrivateImplementationPattern< BToDilepton > . . . 348
eos::BToDilepton . . . 86
eos::PrivateImplementationPattern< BToKDilepton< LargeRecoil > > . . . 348
eos::BToKDilepton< LargeRecoil > . . . 88
eos::PrivateImplementationPattern< BToKDilepton< LowRecoil > > . . . 348
eos::BToKDilepton< LowRecoil > . . . 91
eos::PrivateImplementationPattern< BToKstarDilepton< LargeRecoil > > . . . 348
eos::BToKstarDilepton< LargeRecoil > . . . 95
eos::PrivateImplementationPattern< BToKstarDilepton< LowRecoil > > . . . 348
eos::BToKstarDilepton< LowRecoil > . . . 99
eos::PrivateImplementationPattern< BToKstarGamma > . . . 348
eos::BToKstarGamma . . . 104
eos::PrivateImplementationPattern< BToPiLeptonNeutrino > . . . 348
eos::BToPiLeptonNeutrino . . . 107
eos::PrivateImplementationPattern< BToXsDilepton< HLMW2005 > > . . . 348

eos::BToXsDilepton< HLMW2005 > . . . 109
eos::PrivateImplementationPattern< BToXsGamma< Minimal > > . . . 348
eos::BToXsGamma< Minimal > . . . 110
eos::PrivateImplementationPattern< BToXsGamma< NLO > > . . . 348
eos::BToXsGamma< NLO > . . . 111
eos::PrivateImplementationPattern< Cluster > . . . 348
eos::Cluster . . . 126
eos::PrivateImplementationPattern< Constraint > . . . 348
eos::Constraint . . . 162
eos::PrivateImplementationPattern< Diagnostics > . . . 348
eos::Diagnostics . . . 176
eos::PrivateImplementationPattern< EquationSolver > . . . 348
eos::EquationSolver . . . 181
eos::PrivateImplementationPattern< hdf5::DataSetHandle > . . . 348
eos::hdf5::DataSetHandle . . . 167
eos::PrivateImplementationPattern< hdf5::FileHandle > . . . 348
eos::hdf5::FileHandle . . . 188
eos::PrivateImplementationPattern< HierarchicalClustering > . . . 348
eos::HierarchicalClustering . . . 210
eos::PrivateImplementationPattern< HierarchicalClustering::Component > . . . 348
eos::HierarchicalClustering::Component . . . 129
eos::PrivateImplementationPattern< Histogram< 1 > > . . . 348
eos::Histogram< 1 > . . . 213
eos::PrivateImplementationPattern< Histogram< 2 > > . . . 348
eos::Histogram< 2 > . . . 216
eos::PrivateImplementationPattern< Kinematics > . . . 348
eos::Kinematics . . . 227
eos::PrivateImplementationPattern< LambdaBToLambdaDilepton< LargeRecoil > > . . . 348
eos::LambdaBToLambdaDilepton< LargeRecoil > . . . 235
eos::PrivateImplementationPattern< LambdaBToLambdaDilepton< LowRecoil > > . . . 348
eos::LambdaBToLambdaDilepton< LowRecoil > . . . 237
eos::PrivateImplementationPattern< Log > . . . 348
eos::Log . . . 242
eos::PrivateImplementationPattern< LogLikelihood > . . . 348
eos::LogLikelihood . . . 244
eos::PrivateImplementationPattern< MarkovChain > . . . 348
eos::MarkovChain . . . 264
eos::PrivateImplementationPattern< MarkovChainSampler > . . . 348
eos::MarkovChainSampler . . . 270
eos::PrivateImplementationPattern< ObservableCache > . . . 348
eos::ObservableCache . . . 302
eos::PrivateImplementationPattern< ObservableSet > . . . 348
eos::ObservableSet . . . 307
eos::PrivateImplementationPattern< Options > . . . 348
eos::Options . . . 323
eos::PrivateImplementationPattern< Parameters > . . . 348

eos::Parameters . . . 334
eos::PrivateImplementationPattern< PionLCDAs > . . . 348
eos::PionLCDAs . . . 339
eos::PrivateImplementationPattern< PopulationMonteCarloSampler > . . . 348
eos::PopulationMonteCarloSampler . . . 341
eos::PrivateImplementationPattern< PriorSampler > . . . 348
eos::PriorSampler . . . 346
eos::PrivateImplementationPattern< RandomNumberGenerator > . . . 348
eos::RandomNumberGenerator . . . 360
eos::PrivateImplementationPattern< Thread > . . . 348
eos::Thread . . . 389
eos::PrivateImplementationPattern< ThreadPool > . . . 348
eos::ThreadPool . . . 391
eos::PrivateImplementationPattern< Ticket > . . . 348
eos::Ticket . . . 392
eos::PrivateImplementationPattern< TicketList > . . . 348
eos::TicketList . . . 394
eos::MarkovChain::ProposalFunction . . . 349
eos::proposal_functions::BlockDecomposition . . . 79
eos::proposal_functions::GlobalLocal . . . 203
eos::proposal_functions::Multivariate . . . 282
eos::proposal_functions::MultivariateGaussian . . . 288
eos::proposal_functions::MultivariateStudentT . . . 291
eos::PToP . . . 351
eos::PToV . . . 352
eos::QCD . . . 353
eos::QCDFIntegrals . . . 357
eos::implementation::ResultOf< Result_(∗)(Args_...)> . . . 362
eos::implementation::ResultOf< Result_(Class_::∗)(Args_...)> . . . 363
eos::QCDFIntegrals::Results . . . 364
eos::RValue . . . 366
eos::Save< V_, R_ > . . . 367
eos::SelectOneOfType< Want_ > . . . 370
eos::SelectOneOfType< Want_, Try_, Rest_...> . . . 371
eos::ShortDistanceLowRecoil . . . 372
eos::EquationSolver::Solution . . . 379
eos::MarkovChain::State . . . 381
eos::MarkovChain::Stats . . . 383
eos::PopulationMonteCarloSampler::Status . . . 385
eos::TopLoops . . . 396
eos::TryLock . . . 399
eos::deriv::TwoSided . . . 400
eos::hdf5::Type . . . 401
eos::hdf5::Array< rank_, T_ > . . . 65
eos::hdf5::Composite< T_ > . . . 130
eos::hdf5::Scalar< T_ > . . . 368
eos::VerifiedRange< T_ > . . . 409
eos::Welford . . . 414
eos::WilsonCoefficients< BToS > . . . 415
eos::WilsonPolynomialCloner . . . 419

eos::WilsonPolynomialEvaluator 420
eos::WilsonPolynomialPrinter 421
eos::WrappedForwardIterator< Tag_, Value_ > 424

Chapter 3

Class Index

3.1 Class List

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

eos::AbsoluteTestObservable 51
eos::Accumulator 53
eos::proposal_functions::AdjacencyMatrix 54
eos::btovll::Amplitudes 56
eos::Analysis 57
eos::AnalyticFormFactorBToPiDKMMO2008 62
eos::btovll::AngularCoefficients 64
eos::hdf5::Array< rank_, T_ > 65
eos::hdf5::Attribute< T_ > 67
eos::hdf5::AttributeHandle 68
eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ > 69
eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ > 71
eos::BCL2008FormFactors< Process_ > 73
eos::BFW2010FormFactors< PToP > 74
eos::Histogram< 2 >::Bin 75
eos::Histogram< 1 >::Bin 77
eos::proposal_functions::BlockDecomposition 79
eos::BMesonProperties 82
eos::Bremsstrahlung 83
eos::BsToPhi 85
eos::BToDilepton 86
eos::BToK 87
eos::BToKDilepton< LargeRecoil > 88
eos::BToKDilepton< LowRecoil > 91
eos::BToKstar 94
eos::BToKstarDilepton< LargeRecoil > 95
eos::BToKstarDilepton< LowRecoil > 99
eos::BToKstarGamma 104
eos::BToPi 106
eos::BToPiLeptonNeutrino 107
eos::BToS 108
eos::BToXsDilepton< HLMW2005 > 109
eos::BToXsGamma< Minimal > 110

eos::BToXsGamma< NLO > . . . 111
eos::BZ2004FormFactors< Process_, PToP > . . . 112
eos::BZ2004FormFactors< Process_, PToV > . . . 113
eos::BZ2004FormFactorsSplit< Process_ > . . . 115
eos::CartesianProduct< T_ > (CartesianProduct is a template class to represent a Cartesian Product of an arbitrary number of T_. T_ should be a container providing a random access iterator) . . . 116
eos::CharmLoops . . . 118
eos::ChiSquared . . . 122
eos::CKMScanComponent . . . 123
eos::CKMScanModel . . . 125
eos::Cluster . . . 126
eos::HierarchicalClustering::Component . . . 129
eos::hdf5::Composite< T_ > . . . 130
eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ > . . . 132
eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ > . . . 133
eos::ConcreteObservable< Decay_, Args_ > . . . 134
eos::ConcreteObservableFactory< Decay_, Args_ > . . . 136
eos::ConditionVariable . . . 137
eos::MarkovChainSampler::Config . . . 139
eos::EquationSolver::Config . . . 145
eos::PopulationMonteCarloSampler::Config . . . 146
eos::PriorSampler::Config . . . 154
eos::proposal_functions::GlobalLocal::Config . . . 156
eos::HierarchicalClustering::Config . . . 160
eos::Constraint . . . 162
eos::hdf5::DataSet< T_ > . . . 164
eos::hdf5::DataSetHandle . . . 167
eos::hdf5::DataType< T_ > . . . 169
eos::hdf5::DataType< const char * > . . . 170
eos::hdf5::DataType< double > . . . 171
eos::hdf5::DataType< int > . . . 172
eos::hdf5::DataType< short > . . . 173
eos::hdf5::DataType< unsigned > . . . 174
eos::InstantiationPolicy::DeleteOnDestruction< T_ > . . . 175
eos::Diagnostics . . . 176
eos::implementation::DoStringify< T_ > . . . 178
eos::EMContributions . . . 179
eos::Diagnostics::Entry . . . 180
eos::EquationSolver . . . 181
eos::Exception . . . 183
eos::proposal_functions::Factory (Abstract factory to read in a proposal density from file) . . . 184
eos::hdf5::File . . . 185
eos::hdf5::FileHandle . . . 188
eos::FormFactorAdapter< Transition_ > . . . 189
eos::FormFactorAdapterFactory< Transition_ > . . . 191
eos::FormFactorFactory< OneHalfPlusToOneHalfPlus > . . . 192
eos::FormFactorFactory< PToP > . . . 193
eos::FormFactorFactory< PToV > . . . 194
eos::FormFactorRatioAdapter< Transition_ > . . . 195
eos::FormFactorRatioAdapterFactory< Transition_ > . . . 197
eos::FormFactors< OneHalfPlusToOneHalfPlus > . . . 198
eos::FormFactors< PToP > . . . 199
eos::FormFactors< PToV > . . . 201

eos::proposal_functions::GlobalLocal . . . 203
eos::HardScattering . . . 208
eos::HDF5Error . . . 209
eos::HierarchicalClustering . . . 210
eos::Histogram< 1 > . . . 213
eos::Histogram< 2 > . . . 216
eos::MarkovChain::History . . . 219
eos::HLMW2005 . . . 221
eos::MarkovChain::HyperParameter . . . 222
eos::InstantiationPolicy< T_, NonCopyable > . . . 223
eos::InstantiationPolicy< T_, NonInstantiable > . . . 224
eos::InstantiationPolicy< T_, Singleton > . . . 225
eos::InternalError . . . 226
eos::Kinematics . . . 227
eos::KinematicVariable . . . 230
eos::KMPW2010FormFactors< PToP > . . . 232
eos::KMPW2010FormFactors< PToV > . . . 233
eos::LambdaBToLambdaDilepton< LargeRecoil > . . . 235
eos::LambdaBToLambdaDilepton< LowRecoil > . . . 237
eos::LargeRecoil . . . 239
eos::LCDA . . . 240
eos::Lock . . . 241
eos::Log . . . 242
eos::LogLikelihood . . . 244
eos::LogLikelihoodBlock . . . 248
eos::LogMessageHandler . . . 255
eos::LogPrior . . . 256
eos::LongDistance . . . 262
eos::LowRecoil . . . 263
eos::MarkovChain . . . 264
eos::MarkovChainSampler . . . 270
eos::MemoisationControl . . . 273
eos::Memoiser< Result_, Params_ > . . . 274
eos::Minimal . . . 275
eos::Model . . . 276
eos::ModelComponent< components::CKM > . . . 277
eos::ModelComponent< components::DeltaB1 > . . . 279
eos::ModelComponent< components::QCD > . . . 280
eos::proposal_functions::Multivariate . . . 282
eos::proposal_functions::MultivariateAccess . . . 287
eos::proposal_functions::MultivariateGaussian . . . 288
eos::proposal_functions::MultivariateStudentT . . . 291
eos::Mutable (Base class for all mutable entities, e.g. a Parameter) . . . 294
eos::Mutex . . . 296
eos::NLO . . . 297
eos::NoSuchChiSquaredError . . . 298
eos::NoSuchModelError . . . 299
eos::Observable . . . 300
eos::ObservableCache . . . 302
eos::ObservableFactory . . . 305
eos::ObservableNameError . . . 306
eos::ObservableSet . . . 307
eos::OneHalfPlusToOneHalfPlus . . . 310
eos::OneOf< Types_ > . . . 311

eos::OneOfValue< Type_, Types_ > . . . 312
eos::OneOfValueBase< Types_ > . . . 313
eos::OneOfVisitor< Types_ > . . . 314
eos::OneOfVisitorVisit< Type_ > . . . 315
eos::OneOfVisitorWrapper< Underlying_, Result_, Types_ > . . . 316
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ > . . . 317
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...> . . . 318
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void > . . . 319
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...> . . . 320
eos::Analysis::OptimizationOptions . . . 321
eos::Options . . . 323
eos::PopulationMonteCarloSampler::Output . . . 325
eos::Analysis::Output . . . 328
eos::Parameter . . . 329
eos::ParameterDescription . . . 332
eos::ParameterRange . . . 333
eos::Parameters . . . 334
eos::ParameterUser . . . 337
eos::PionLCDAs . . . 339
eos::PopulationMonteCarloSampler . . . 341
eos::MarkovChainSampler::PreRunInfo . . . 344
eos::PriorSampler . . . 346
eos::PrivateImplementationPattern< T_ > . . . 348
eos::MarkovChain::ProposalFunction . . . 349
eos::PToP . . . 351
eos::PToV . . . 352
eos::QCD . . . 353
eos::QCDFIntegrals . . . 357
eos::RandomNumberGenerator . . . 360
eos::implementation::ResultOf< Result_(∗)(Args_...)> . . . 362
eos::implementation::ResultOf< Result_(Class_::∗)(Args_...)> . . . 363
eos::QCDFIntegrals::Results . . . 364
eos::RValue . . . 366
eos::Save< V_, R_ > . . . 367
eos::hdf5::Scalar< T_ > . . . 368
eos::SelectOneOfType< Want_ > . . . 370
eos::SelectOneOfType< Want_, Try_, Rest_...> . . . 371
eos::ShortDistanceLowRecoil . . . 372
eos::SMComponent< components::CKM > . . . 374
eos::SMComponent< components::DeltaB1 > . . . 376
eos::SMComponent< components::QCD > . . . 377
eos::EquationSolver::Solution . . . 379
eos::StandardModel . . . 380
eos::MarkovChain::State . . . 381
eos::MarkovChain::Stats . . . 383
eos::PopulationMonteCarloSampler::Status . . . 385
eos::TestObservable . . . 387
eos::Thread . . . 389
eos::ThreadPool . . . 391
eos::Ticket . . . 392
eos::TicketList . . . 394
eos::TopLoops . . . 396
eos::TryLock . . . 399
eos::deriv::TwoSided . . . 400

eos::hdf5::Type 401
eos::UnknownConstraintError 403
eos::UnknownKinematicVariableError 404
eos::UnknownOptionError 405
eos::UnknownParameterError 406
eos::proposal_functions::UnknownProposalError 407
eos::UsedParameter 408
eos::VerifiedRange< T_ > 409
eos::VerifiedRangeError 411
eos::VerifiedRangeOverflow 412
eos::VerifiedRangeUnderflow 413
eos::Welford 414
eos::WilsonCoefficients< BToS > 415
eos::WilsonPolynomialCloner 419
eos::WilsonPolynomialEvaluator 420
eos::WilsonPolynomialPrinter 421
eos::WilsonScanComponent 422
eos::WilsonScanModel 423
eos::WrappedForwardIterator< Tag_, Value_ > 424

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/alexshires/EOS/eos-next/eos/constraint.hh . . . 431
/home/alexshires/EOS/eos-next/eos/observable.hh . . . 441
/home/alexshires/EOS/eos-next/eos/b-decays/b-to-pi-l-nu.hh . . . 429
/home/alexshires/EOS/eos-next/eos/b-decays/properties.hh . . . 430
/home/alexshires/EOS/eos-next/eos/form-factors/analytic-b-to-pi.hh . . . 432
/home/alexshires/EOS/eos-next/eos/form-factors/baryonic-impl.hh . . . 433
/home/alexshires/EOS/eos-next/eos/form-factors/baryonic.hh . . . 434
/home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh . . . 435
/home/alexshires/EOS/eos-next/eos/form-factors/form-factors-fwd.hh . . . 436
/home/alexshires/EOS/eos-next/eos/form-factors/form-factors.hh . . . 437
/home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh . . . 438
/home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh . . . 439
/home/alexshires/EOS/eos-next/eos/form-factors/pi-lcdas.hh . . . 440
/home/alexshires/EOS/eos-next/eos/rare-b-decays/bremsstrahlung.hh . . . 442
/home/alexshires/EOS/eos-next/eos/rare-b-decays/charm-loops.hh . . . 443
/home/alexshires/EOS/eos-next/eos/rare-b-decays/decays.hh . . . 444
/home/alexshires/EOS/eos-next/eos/rare-b-decays/em-contributions.hh . . . 445
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-dilepton.hh . . . 446
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-large-recoil.hh . . . 447
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-low-recoil.hh . . . 448
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton.hh . . . 449
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-gamma.hh . . . 450
/home/alexshires/EOS/eos-next/eos/rare-b-decays/hard-scattering.hh . . . 451
/home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-dilepton.hh . . . 452
/home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh . . . 453
/home/alexshires/EOS/eos-next/eos/rare-b-decays/lambda-b-to-lambda-dilepton.hh . . . 454
/home/alexshires/EOS/eos-next/eos/rare-b-decays/long-distance.hh . . . 455
/home/alexshires/EOS/eos-next/eos/rare-b-decays/qcdf_integrals.hh . . . 456
/home/alexshires/EOS/eos-next/eos/utils/accumulator.hh . . . 457
/home/alexshires/EOS/eos-next/eos/utils/analysis-fwd.hh . . . 458
/home/alexshires/EOS/eos-next/eos/utils/analysis.hh . . . 459
/home/alexshires/EOS/eos-next/eos/utils/analysis_TEST.hh . . . 460
/home/alexshires/EOS/eos-next/eos/utils/apply.hh . . . 461

/home/alexshires/EOS/eos-next/eos/utils/cartesian-product.hh . . . 462
/home/alexshires/EOS/eos-next/eos/utils/chi-squared.hh . . . 463
/home/alexshires/EOS/eos-next/eos/utils/ckm_scan_model.hh . . . 464
/home/alexshires/EOS/eos-next/eos/utils/cluster.hh . . . 465
/home/alexshires/EOS/eos-next/eos/utils/complex.hh . . . 466
/home/alexshires/EOS/eos-next/eos/utils/concrete_observable.hh . . . 467
/home/alexshires/EOS/eos-next/eos/utils/condition_variable.hh . . . 468
/home/alexshires/EOS/eos-next/eos/utils/derivative.hh . . . 469
/home/alexshires/EOS/eos-next/eos/utils/destringify.hh . . . 470
/home/alexshires/EOS/eos-next/eos/utils/diagnostics.hh . . . 471
/home/alexshires/EOS/eos-next/eos/utils/equation_solver.hh . . . 472
/home/alexshires/EOS/eos-next/eos/utils/exception.hh . . . 473
/home/alexshires/EOS/eos-next/eos/utils/hdf5-fwd.hh . . . 474
/home/alexshires/EOS/eos-next/eos/utils/hdf5.hh . . . 475
/home/alexshires/EOS/eos-next/eos/utils/hierarchical-clustering.hh . . . 477
/home/alexshires/EOS/eos-next/eos/utils/histogram.hh . . . 478
/home/alexshires/EOS/eos-next/eos/utils/instantiation_policy-impl.hh . . . 479
/home/alexshires/EOS/eos-next/eos/utils/instantiation_policy.hh . . . 480
/home/alexshires/EOS/eos-next/eos/utils/integrate-impl.hh . . . 481
/home/alexshires/EOS/eos-next/eos/utils/integrate.hh . . . 482
/home/alexshires/EOS/eos-next/eos/utils/kinematic.hh . . . 483
/home/alexshires/EOS/eos-next/eos/utils/lcda.hh . . . 484
/home/alexshires/EOS/eos-next/eos/utils/lock.hh . . . 485
/home/alexshires/EOS/eos-next/eos/utils/log.hh . . . 486
/home/alexshires/EOS/eos-next/eos/utils/log_likelihood-fwd.hh . . . 487
/home/alexshires/EOS/eos-next/eos/utils/log_likelihood.hh . . . 488
/home/alexshires/EOS/eos-next/eos/utils/log_prior-fwd.hh . . . 489
/home/alexshires/EOS/eos-next/eos/utils/log_prior.hh . . . 490
/home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh . . . 491
/home/alexshires/EOS/eos-next/eos/utils/markov_chain_sampler.hh . . . 492
/home/alexshires/EOS/eos-next/eos/utils/matrix.hh . . . 493
/home/alexshires/EOS/eos-next/eos/utils/memoise.hh . . . 495
/home/alexshires/EOS/eos-next/eos/utils/model.hh . . . 496
/home/alexshires/EOS/eos-next/eos/utils/mutable-fwd.hh . . . 497
/home/alexshires/EOS/eos-next/eos/utils/mutable.hh . . . 498
/home/alexshires/EOS/eos-next/eos/utils/mutex.hh . . . 499
/home/alexshires/EOS/eos-next/eos/utils/observable_cache.hh . . . 500
/home/alexshires/EOS/eos-next/eos/utils/observable_set.hh . . . 501
/home/alexshires/EOS/eos-next/eos/utils/one-of.hh . . . 502
/home/alexshires/EOS/eos-next/eos/utils/options.hh . . . 503
/home/alexshires/EOS/eos-next/eos/utils/parameters-fwd.hh . . . 504
/home/alexshires/EOS/eos-next/eos/utils/parameters.hh . . . 505
/home/alexshires/EOS/eos-next/eos/utils/polylog.hh . . . 506
/home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_sampler.hh . . . 507
/home/alexshires/EOS/eos-next/eos/utils/power_of.hh . . . 508
/home/alexshires/EOS/eos-next/eos/utils/prior_sampler.hh . . . 509
/home/alexshires/EOS/eos-next/eos/utils/private_implementation_pattern-impl.hh . . . 510
/home/alexshires/EOS/eos-next/eos/utils/private_implementation_pattern.hh . . . 511
/home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh . . . 512
/home/alexshires/EOS/eos-next/eos/utils/qcd.hh . . . 514
/home/alexshires/EOS/eos-next/eos/utils/random_number_generator.hh . . . 515
/home/alexshires/EOS/eos-next/eos/utils/rvalue.hh . . . 516
/home/alexshires/EOS/eos-next/eos/utils/save.hh . . . 517
/home/alexshires/EOS/eos-next/eos/utils/standard-model.hh . . . 518

/home/alexshires/EOS/eos-next/eos/utils/stringify.hh 519
/home/alexshires/EOS/eos-next/eos/utils/thread.hh 520
/home/alexshires/EOS/eos-next/eos/utils/thread_pool.hh 521
/home/alexshires/EOS/eos-next/eos/utils/ticket.hh 522
/home/alexshires/EOS/eos-next/eos/utils/top-loops.hh 523
/home/alexshires/EOS/eos-next/eos/utils/verify.hh 524
/home/alexshires/EOS/eos-next/eos/utils/welford.hh 525
/home/alexshires/EOS/eos-next/eos/utils/wilson-polynomial.hh 526
/home/alexshires/EOS/eos-next/eos/utils/wilson_coefficients.hh 527
/home/alexshires/EOS/eos-next/eos/utils/wilson_scan_model.hh 528
/home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-fwd.hh 529
/home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-impl.hh 530
/home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator.hh 531

Chapter 5

Namespace Documentation

5.1 eos Namespace Reference

Namespaces

- namespace btovll
- namespace components
- namespace deriv
- namespace hdf5
- namespace impl
- namespace implementation
- namespace InstantiationPolicy
- namespace proposal_functions

Classes

- class BToPiLeptonNeutrino
- class BMesonProperties
- class Constraint
- struct UnknownConstraintError
- class AnalyticFormFactorBToPiDKMMO2008
- struct OneHalfPlusToOneHalfPlus
- class FormFactors< OneHalfPlusToOneHalfPlus >
- class FormFactorFactory< OneHalfPlusToOneHalfPlus >
- class FormFactorAdapter
- class FormFactorAdapterFactory
- class FormFactorRatioAdapter
- class FormFactorRatioAdapterFactory
- struct BToKstar
- struct BsToPhi
- class BZ2004FormFactors< Process_, PToV >
- class KMPW2010FormFactors< PToV >
- struct BToK
- struct BToPi
- class BCL2008FormFactors

- class BZ2004FormFactors< Process_, PToP >
- class BZ2004FormFactorsSplit
- class KMPW2010FormFactors< PToP >
- class BFW2010FormFactors< PToP >
- struct PToV
- struct PToP
- class FormFactors< PToV >
- class FormFactorFactory< PToV >
- class FormFactors< PToP >
- class FormFactorFactory< PToP >
- class PionLCDAs
- class Observable
- class ObservableFactory
- struct ObservableNameError
- struct Bremsstrahlung
- struct CharmLoops
- struct ShortDistanceLowRecoil
- struct LargeRecoil
- struct LowRecoil
- struct EMContributions
- class BToDilepton
- class BToKstarDilepton< LargeRecoil >
- class BToKDilepton< LargeRecoil >
- class BToKstarDilepton< LowRecoil >
- class BToKDilepton< LowRecoil >
- class BToKstarGamma
- struct HardScattering
- struct HLMW2005
- class BToXsDilepton< HLMW2005 >
- struct Minimal
- class BToXsGamma< Minimal >
- struct NLO
- class BToXsGamma< NLO >
- class LambdaBToLambdaDilepton< LargeRecoil >
- class LambdaBToLambdaDilepton< LowRecoil >
- struct LongDistance
- class QCDFIntegrals
- class Accumulator
- class Analysis
- struct TestObservable
- struct AbsoluteTestObservable
- class CartesianProduct

  CartesianProduct is a template class to represent a Cartesian Product of an arbitrary number of T_. T_ should be a container providing a random access iterator.

- struct ChiSquared
- struct NoSuchChiSquaredError
- class CKMScanComponent
- class CKMScanModel
- class Cluster

- class ConcreteObservable
- class ConcreteObservableFactory
- class ConditionVariable
- class Diagnostics
- class EquationSolver
- class Exception
- struct InternalError
- class HDF5Error
- class HierarchicalClustering
- class Histogram< 1 >
- class Histogram< 2 >
- class InstantiationPolicy< T_, NonCopyable >
- class InstantiationPolicy< T_, NonInstantiable >
- class InstantiationPolicy< T_, Singleton >
- struct UnknownKinematicVariableError
- class Kinematics
- class KinematicVariable
- struct LCDA
- class Lock
- class TryLock
- class Log
- class LogMessageHandler
- class LogLikelihoodBlock
- class LogLikelihood
- class LogPrior
- class MarkovChain
- class MarkovChainSampler
- class MemoisationControl
- class Memoiser
- class ModelComponent< components::CKM >
- class ModelComponent< components::QCD >
- class ModelComponent< components::DeltaB1 >
- class Model
- struct NoSuchModelError
- class Mutable

  Base class for all mutable entities, e.g. a Parameter.

- class Mutex
- class ObservableCache
- class ObservableSet
- struct SelectOneOfType< Want_ >
- struct SelectOneOfType< Want_, Try_, Rest_...>
- struct OneOfVisitorVisit
- struct OneOfVisitor
- struct OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >
- struct OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >
- struct OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>
- struct OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>
- struct OneOfVisitorWrapper
- struct OneOfValueBase

- struct OneOfValue
- class OneOf
- struct UnknownOptionError
- class Options
- struct UnknownParameterError
- class Parameters
- class Parameter
- class ParameterUser
- class UsedParameter
- struct ParameterDescription
- struct ParameterRange
- class PopulationMonteCarloSampler
- class PriorSampler
- class PrivateImplementationPattern
- class QCD
- class RandomNumberGenerator
- struct RValue
- class Save
- class SMComponent< components::CKM >
- class SMComponent< components::QCD >
- class SMComponent< components::DeltaB1 >
- class StandardModel
- class Thread
- class ThreadPool
- class Ticket
- class TicketList
- struct TopLoops
- class VerifiedRangeError
- struct VerifiedRangeOverflow
- struct VerifiedRangeUnderflow
- class VerifiedRange
- struct Welford
- class WilsonPolynomialCloner
- class WilsonPolynomialPrinter
- class WilsonPolynomialEvaluator
- struct BToS
- struct WilsonCoefficients< BToS >
- class WilsonScanComponent
- class WilsonScanModel
- class WrappedForwardIterator

Typedefs

- typedef std::shared_ptr< Observable > ObservablePtr
- typedef std::shared_ptr< Analysis > AnalysisPtr
- typedef std::shared_ptr< LogLikelihoodBlock > LogLikelihoodBlockPtr
- typedef std::shared_ptr< LogPrior > LogPriorPtr
- typedef std::shared_ptr< MarkovChain::History > HistoryPtr
- typedef std::shared_ptr< MarkovChain::ProposalFunction > ProposalFunctionPtr
- typedef std::shared_ptr< Mutable > MutablePtr
- typedef OneOf< Constant, Sum, Product, Sine, Cosine, Parameter > WilsonPolynomial

Enumerations

- enum Helicity { left_handed = -1, right_handed = +1 }
- enum LogLevel {
  ll_silent, ll_error, ll_warning, ll_informational,
  ll_debug, ll_last }

Functions

- Analysis make_analysis (bool flat)
- template<typename Result_ , typename... FunctionArgs_, typename... TupleElements_>
  Result_ apply (Result_(∗f)(FunctionArgs_...), const std::tuple< TupleElements_...> &t)
- template<typename Result_ , typename... FunctionArgs_, typename... TupleElements_>
  Result_ apply (const std::function< Result_(FunctionArgs_...)> &f, const std::tuple< TupleElements_...> &t)
- template<typename Result_ , typename Class_ , typename... FunctionArgs_, typename... TupleElements_>
  Result_ apply (Result_(Class_::∗f)(FunctionArgs_...), const std::tuple< Class_ ∗, TupleElements_-...> &t)
- template<typename Decay_ , typename Tuple_ , typename... Args_>
  ObservableFactory ∗ make_concrete_observable_factory (const std::string &name, double(Decay_-::∗function)(const Args_ &...) const, const Tuple_ &kinematics_names=std::make_tuple())
- template<unsigned order_, typename Method_ >
  double derivative (const std::function< double(const double &)> &f, const double &x0)
- template<>
  double derivative< 1u, deriv::TwoSided > (const std::function< double(const double &)> &f, const double &x0)
- template<>
  double derivative< 2u, deriv::TwoSided > (const std::function< double(const double &)> &f, const double &x0)
- template<typename T_ >
  T_ destringify (const std::string &input)
- std::ostream & operator<< (std::ostream &lhs, const HierarchicalClustering::Component &rhs)
- Histogram< 1 > estimate_cumulative_distribution (const Histogram< 1 > &distribution)
- double lambda (const double &a, const double &b, const double &c)
- std::ostream & operator<< (std::ostream &lhs, const MarkovChain::State &rhs)
- std::ostream & operator<< (std::ostream &, const MarkovChainSampler::Config &config)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< std::array< T_, n_ >, m_ > operator+ (const std::array< std::array< T_, n_ >, m_ > &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< std::array< T_, n_ >, m_ > operator- (const std::array< std::array< T_, n_ >, m_ > &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t m_>
  std::array< T_, m_ > operator+ (const std::array< T_, m_ > &x, const std::array< T_, m_ > &y)
- template<typename T_ , std::size_t m_>
  std::array< T_, m_ > operator- (const std::array< T_, m_ > &x, const std::array< T_, m_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_, std::size_t o_>
  std::array< std::array< T_, n_ >, m_ > operator∗ (const std::array< std::array< T_, o_ >, m_ > &x, const std::array< std::array< T_, n_ >, o_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< T_, m_ > operator∗ (const std::array< std::array< T_, n_ >, m_ > &x, const std::array< T_, n_ > &y)

- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< T_, n_ > operator^ (const std::array< T_, m_ > &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< std::array< T_, n_ >, m_ > operator∗ (const T_ &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t n_>
  std::array< T_, n_ > operator∗ (const T_ &x, const std::array< T_, n_ > &y)
- template<typename T_ , std::size_t n_>
  T_ dot (const std::array< T_, n_ > &x, const std::array< T_, n_ > &y)
- template<typename T_ , std::size_t n_>
  std::array< T_, n_ > mult (const std::array< T_, n_ > &x, const std::array< T_, n_ > &y)
- template<typename T_ , std::size_t n_>
  std::array< T_, n_ > divide (const std::array< T_, n_ > &x, const std::array< T_, n_ > &y)
- template<typename FunctionType_ , typename... Params>
  implementation::ResultOf< FunctionType_ >::Type memoise (FunctionType_ f, const Params &...p)
- template<typename FunctionType_ , typename... Params>
  unsigned number_of_memoisations (FunctionType_, const Params &...)
- Options operator+ (const Options &lhs, const Options &rhs)

  Merge operator.

- complex< double > dilog (const complex< double > &z) __attribute__((pure))
- complex< double > trilog (const complex< double > &z) __attribute__((pure))
- std::ostream & operator<< (std::ostream &, const PopulationMonteCarloSampler::Config &)
- template<unsigned n_, typename T_ >
  T_ power_of (const T_ &x)
- template<typename T_ >
  std::string stringify (const T_ &x, const unsigned &precision=10)
- template<typename Iterator_ >
  std::string stringify (const Iterator_ &begin, const Iterator_ &end, const unsigned &precision=10)
- template<typename T_ >
  std::string stringify (const T_ ∗m, const unsigned &dim, const unsigned &precision=10)
- template<typename Container_ >
  std::string stringify_container (const Container_ &container, unsigned precision=10)
- WilsonPolynomial make_polynomial (const ObservablePtr &, const std::list< std::string > &)
- ObservablePtr make_polynomial_observable (const WilsonPolynomial &polynomial, const Parameters &parameters)
- ObservablePtr make_polynomial_ratio (const WilsonPolynomial &numerator, const WilsonPolynomial &denominator, const Parameters &parameters)
- ObservablePtr make_polynomial_ht_like_ratio (const WilsonPolynomial &numerator, const WilsonPolynomial &denominator, const WilsonPolynomial &denominator2, const Parameters &parameters)
- WilsonCoefficients< BToS > evolve (const std::array< complex< double >, 15 > &wc_qcd_0, const std::array< complex< double >, 15 > &wc_qcd_1, const std::array< complex< double >, 15 > &wc_qcd_2, const double &alpha_s_0, const double &alpha_s, const double &nf, const QCD::BetaFunction &beta)

  - template<std::size_t k>
    std::array< double, k > integrate (const std::function< std::array< double, k >(const double &)> &f, unsigned n, const double &a, const double &b)

- double integrate (const std::function< double(const double &)> &f, unsigned n, const double &a, const double &b)
- complex< double > integrate (const std::function< complex< double >(const double &)> &f, unsigned n, const double &a, const double &b)
- std::ostream & operator<< (std::ostream &lhs, const LogLevel &rhs)
- std::istream & operator>> (std::istream &lhs, LogLevel &rhs)

5.1.1 Typedef Documentation

5.1.1.1 typedef std::shared_ptr<Analysis> eos::AnalysisPtr

5.1.1.2 typedef std::shared_ptr<MarkovChain::History> eos::HistoryPtr

5.1.1.3 typedef std::shared_ptr<LogLikelihoodBlock> eos::LogLikelihoodBlockPtr

5.1.1.4 typedef std::shared_ptr<LogPrior> eos::LogPriorPtr

5.1.1.5 typedef std::shared_ptr<Mutable> eos::MutablePtr

5.1.1.6 typedef std::shared_ptr< Observable > eos::ObservablePtr

5.1.1.7 typedef std::shared_ptr<MarkovChain::ProposalFunction> eos::ProposalFunctionPtr

5.1.1.8 typedef OneOf<Constant, Sum, Product, Sine, Cosine, Parameter> eos::WilsonPolynomial

5.1.2 Enumeration Type Documentation

5.1.2.1 enum eos::Helicity

Enumerator:

left_handed

right_handed

5.1.2.2 enum eos::LogLevel

Categories for the severity of log messages

Enumerator:

ll_silent do not print any error message

ll_error only print error messages

ll_warning also print warning messages

ll_informational also print informational messages

ll_debug also print debug messages

ll_last

5.1.3 Function Documentation

5.1.3.1 template<typename Result_ , typename Class_ , typename... FunctionArgs_, typename... TupleElements_> Result_ eos::apply (Result_(Class_::∗)(FunctionArgs_...) *f*, const std::tuple< Class_ ∗, TupleElements_...> & *t*) `[inline]`

5.1.3.2 template<typename Result_ , typename... FunctionArgs_, typename... TupleElements_> Result_ eos::apply (const std::function< Result_(FunctionArgs_...)> & *f*, const std::tuple< TupleElements_...> & *t*) `[inline]`

5.1.3.3 template<typename Result_ , typename... FunctionArgs_, typename... TupleElements_> Result_ eos::apply (Result_(∗)(FunctionArgs_...) *f*, const std::tuple< TupleElements_...> & *t*) `[inline]`

5.1.3.4 template<unsigned order_, typename Method_ > double eos::derivative (const std::function< double(const double &)> & *f*, const double & *x0*) `[inline]`

5.1.3.5 template<> double eos::derivative< 1u, deriv::TwoSided > (const std::function< double(const double &)> & *f*, const double & *x0*) `[inline]`

5.1.3.6 template<> double eos::derivative< 2u, deriv::TwoSided > (const std::function< double(const double &)> & *f*, const double & *x0*) `[inline]`

5.1.3.7 template<typename T_ > T_ eos::destringify (const std::string & *input*) `[inline]`

5.1.3.8 complex<double> eos::dilog (const complex< double > & *z*)

5.1.3.9 template<typename T_ , std::size_t n_> std::array<T_, n_> eos::divide (const std::array< T_, n_ > & *x*, const std::array< T_, n_ > & *y*) `[inline]`

5.1.3.10 template<typename T_ , std::size_t n_> T_ eos::dot (const std::array< T_, n_ > & *x*, const std::array< T_, n_ > & *y*) `[inline]`

5.1.3.11 Histogram<1> eos::estimate_cumulative_distribution (const Histogram< 1 > & *distribution*)

Compute the estimated cumulative distribution function for a given 1D distribution.

Parameters:

distribution The histogram of a distribution function.

5.1.3.12 WilsonCoefficients<BToS> eos::evolve (const std::array< complex< double >, 15 > & *wc_qcd_0*, const std::array< complex< double >, 15 > & *wc_qcd_1*, const std::array< complex< double >, 15 > & *wc_qcd_2*, const double & *alpha_s_0*, const double & *alpha_s*, const double & *nf*, const QCD::BetaFunction & *beta*)

Evolution of b -> s Wilson coefficients

Calculation according to [BMU1999], Eq. (25). The helicity-flipped, scalar, pseudoscalar, and tensor components are implicitly kept at zero.

Parameters:

wc_qcd_0 The initial scale Wilson coefficients at O(alpha_s^0)
wc_qcd_1 The initial scale Wilson coefficients at O(alpha_s^1)
wc_qcd_2 The initial scale Wilson coefficients at O(alpha_s^2)
alpha_s_0 The strong coupling constant at the initial scale
alpha_s The strong coupling constant at the low scale
nf The number of active flavors
beta Coefficients of the beta function of QCD for nf active flavors.

5.1.3.13 complex<double> eos::integrate (const std::function< complex< double >(const double &)> & *f*, unsigned *n*, const double & *a*, const double & *b*)

Numerically integrate functions of one real-valued parameter.

Uses the Delta^2-Rule by Aitkin to refine the result.

Parameters:

f Integrand.
n Number of evaluations, must be a power of 2.
a Lower limit of the domain of integration.
b Upper limit of the domain of integration.

5.1.3.14 double eos::integrate (const std::function< double(const double &)> & *f*, unsigned *n*, const double & *a*, const double & *b*)

Numerically integrate functions of one real-valued parameter.

Uses the Delta^2-Rule by Aitkin to refine the result.

Parameters:

f Integrand.
n Number of evaluations, must be a power of 2.
a Lower limit of the domain of integration.
b Upper limit of the domain of integration.

5.1.3.15 template<std::size_t k> std::array< double, k > eos::integrate (const std::function< std::array< double, k >(const double &)> & *f*, unsigned *n*, const double & *a*, const double & *b*) `[inline]`

Numerically integrate functions of one real-valued parameter.

Uses the Delta^2-Rule by Aitkin to refine the result.

Parameters:

f Integrand.
n Number of evaluations, must be a power of 2.
a Lower limit of the domain of integration.
b Upper limit of the domain of integration.

5.1.3.16 double eos::lambda (const double & *a*, const double & *b*, const double & *c*) `[inline]`

5.1.3.17 Analysis eos::make_analysis (bool *flat*)

5.1.3.18 template<typename Decay_ , typename Tuple_ , typename... Args_> ObservableFactory* eos::make_concrete_observable_factory (const std::string & *name*, double(Decay_::*)(const Args_ &...) const *function*, const Tuple_ & *kinematics_names* = `std::make_tuple()`) `[inline]`

5.1.3.19 WilsonPolynomial eos::make_polynomial (const ObservablePtr &, const std::list< std::string > &)

5.1.3.20 ObservablePtr eos::make_polynomial_ht_like_ratio (const WilsonPolynomial & *numerator*, const WilsonPolynomial & *denominator*, const WilsonPolynomial & *denominator2*, const Parameters & *parameters*)

Return an Observable that is a ratio similar to H_T^(i) of three WilsonPolynomial objects N,D1,D2:

$$N/\sqrt{D1 \cdot D2}$$

Parameters:

numerator The numerator of the ratio.

denominator1 The first denominator component of the ratio.

denominator2 The first denominator component of the ratio.

parameters The common Parameters object of both numerator and denominator.

5.1.3.21 ObservablePtr eos::make_polynomial_observable (const WilsonPolynomial & *polynomial*, const Parameters & *parameters*)

Return an Observable that wraps a WilsonPolynomial object.

Parameters:

polynomial The polynomial that shall be wrapped.

parameters The Parameters object of the polynomial.

5.1.3.22 ObservablePtr eos::make_polynomial_ratio (const WilsonPolynomial & *numerator*, const WilsonPolynomial & *denominator*, const Parameters & *parameters*)

Return an Observable that is a ratio of two WilsonPolynomial objects.

Parameters:

numerator The numerator of the ratio.

denominator The denominator of the ratio.

parameters The common Parameters object of both numerator and denominator.

5.1.3.23 template<typename FunctionType_ , typename... Params> implementation::ResultOf<FunctionType_>::Type eos::memoise (FunctionType_ *f*, const Params &... *p*) [inline]

5.1.3.24 template<typename T_ , std::size_t n_> std::array<T_, n_> eos::mult (const std::array< T_, n_ > & *x*, const std::array< T_, n_ > & *y*) [inline]

5.1.3.25 template<typename FunctionType_ , typename... Params> unsigned eos::number_of_memoisations (FunctionType_, const Params & ...) [inline]

5.1.3.26 template<typename T_ , std::size_t n_> std::array<T_, n_> eos::operator∗ (const T_ & *x*, const std::array< T_, n_ > & *y*) [inline]

5.1.3.27 template<typename T_ , std::size_t m_, std::size_t n_> std::array<std::array<T_, n_>, m_> eos::operator∗ (const T_ & *x*, const std::array< std::array< T_, n_ >, m_ > & *y*) [inline]

5.1.3.28 template<typename T_ , std::size_t m_, std::size_t n_> std::array<T_, m_> eos::operator∗ (const std::array< std::array< T_, n_ >, m_ > & *x*, const std::array< T_, n_ > & *y*) [inline]

5.1.3.29 template<typename T_ , std::size_t m_, std::size_t n_, std::size_t o_> std::array<std::array<T_, n_>, m_> eos::operator∗ (const std::array< std::array< T_, o_ >, m_ > & *x*, const std::array< std::array< T_, n_ >, o_ > & *y*) [inline]

5.1.3.30 Options eos::operator+ (const Options & *lhs*, const Options & *rhs*)

Merge operator.

5.1.3.31 **template<typename T_ , std::size_t m_> std::array<T_, m_> eos::operator+ (const std::array< T_, m_ > & *x*, const std::array< T_, m_ > & *y*)** `[inline]`

5.1.3.32 **template<typename T_ , std::size_t m_, std::size_t n_> std::array<std::array<T_, n_>, m_> eos::operator+ (const std::array< std::array< T_, n_ >, m_ > & *x*, const std::array< std::array< T_, n_ >, m_ > & *y*)** `[inline]`

5.1.3.33 **template<typename T_ , std::size_t m_> std::array<T_, m_> eos::operator- (const std::array< T_, m_ > & *x*, const std::array< T_, m_ > & *y*)** `[inline]`

5.1.3.34 **template<typename T_ , std::size_t m_, std::size_t n_> std::array<std::array<T_, n_>, m_> eos::operator- (const std::array< std::array< T_, n_ >, m_ > & *x*, const std::array< std::array< T_, n_ >, m_ > & *y*)** `[inline]`

5.1.3.35 **std::ostream& eos::operator<< (std::ostream &, const PopulationMonteCarloSampler::Config &)**

5.1.3.36 **std::ostream& eos::operator<< (std::ostream &, const MarkovChainSampler::Config & *config*)**

5.1.3.37 **std::ostream& eos::operator<< (std::ostream & *lhs*, const MarkovChain::State & *rhs*)**

5.1.3.38 **std::ostream& eos::operator<< (std::ostream & *lhs*, const LogLevel & *rhs*)**

(De)stringification of LogLevel

5.1.3.39 **std::ostream& eos::operator<< (std::ostream & *lhs*, const HierarchicalClustering::Component & *rhs*)**

5.1.3.40 **std::istream& eos::operator>> (std::istream & *lhs*, LogLevel & *rhs*)**

(De)stringification of LogLevel

5.1.3.41 **template<typename T_ , std::size_t m_, std::size_t n_> std::array<T_, n_> eos::operator^ (const std::array< T_, m_ > & *x*, const std::array< std::array< T_, n_ >, m_ > & *y*)** `[inline]`

5.1.3.42 **template<unsigned n_, typename T_ > T_ eos::power_of (const T_ & *x*)** `[inline]`

5.1.3.43 **template<typename T_ > std::string eos::stringify (const T_ * *m*, const unsigned & *dim*, const unsigned & *precision* = 10)** `[inline]`

Stringify a C-style square matrix

Parameters:

m Pointer to the first matrix element.

dim Dimension of the matrix.

precision (Optional) floating point precision for the stringification.

5.1.3.44 template<typename Iterator_ > std::string eos::stringify (const Iterator_ & *begin*, const Iterator_ & *end*, const unsigned & *precision* = 10) [inline]

Stringify a range of iterators.

Parameters:

begin Iterator pointing to the first element of the range.

end Iterator pointing beyond the last element of the range.

precision (Optional) floating point precision for the stringification.

5.1.3.45 template<typename T_ > std::string eos::stringify (const T_ & *x*, const unsigned & *precision* = 10) [inline]

Stringify an arbitrary (scalar) data type.

Parameters:

x Object that shall be stringified.

precision (Optional) floating point precision for the stringification.

5.1.3.46 template<typename Container_ > std::string eos::stringify_container (const Container_ & *container*, unsigned *precision* = 10) [inline]

5.1.3.47 complex<double> eos::trilog (const complex< double > & *z*)

5.2 eos::btovll Namespace Reference

Classes

- struct Amplitudes
- struct AngularCoefficients

Functions

- AngularCoefficients array_to_angular_coefficients (const std::array< double, 12 > &arr)
- double decay_width (const AngularCoefficients &a_c)
- std::array< double, 12 > angular_coefficients_array (const Amplitudes &amplitudes, const double &s, const double &m_l)

5.2.1 Function Documentation

5.2.1.1 std::array<double, 12> eos::btovll::angular_coefficients_array (const Amplitudes & *amplitudes*, const double & *s*, const double & *m_l*) [inline]

5.2.1.2 AngularCoefficients eos::btovll::array_to_angular_coefficients (const std::array< double, 12 > & *arr*) [inline]

5.2.1.3 double eos::btovll::decay_width (const AngularCoefficients & *a_c*) [inline]

5.3 eos::components Namespace Reference

5.4 eos::deriv Namespace Reference

Classes

- class TwoSided

5.5 eos::hdf5 Namespace Reference

Classes

- struct DataType
- struct DataType< double >
- struct DataType< unsigned >
- struct DataType< int >
- struct DataType< short >
- struct DataType< const char * >
- struct Type
- class Scalar
- class Array
- struct CompositeCopier
- struct CompositeCopier< 0, TypeTuple_, Tuple_ >
- class Composite
- class FileHandle
- class DataSetHandle
- class AttributeHandle
- class File
- class DataSet
- class Attribute

Typedefs

- typedef int hid_t
- typedef std::shared_ptr< Type > TypePtr

Functions

- template<typename T_ >
  DataSet< T_ > & operator<< (DataSet< T_ > &, const typename T_::Type &)
- template<typename T_ >
  DataSet< T_ > & operator>> (DataSet< T_ > &, typename T_::Type &)

5.5.1 Typedef Documentation

5.5.1.1 typedef int eos::hdf5::hid_t

5.5.1.2 typedef std::shared_ptr<Type> eos::hdf5::TypePtr

5.5.2 Function Documentation

5.5.2.1 template<typename T_ > DataSet< T_ > & eos::hdf5::operator<< (DataSet< T_ > & *lhs*, const typename T_::Type & *rhs*) `[inline]`

Append a record to the end of an HDF5 DataSet<> .

Parameters:

lhs The DataSet<> to which shall be written.

rhs The record which shall be appended.

5.5.2.2 template<typename T_ > DataSet< T_ > & eos::hdf5::operator>> (DataSet< T_ > & *lhs*, typename T_::Type & *rhs*) `[inline]`

Extract a record from an HDF5 DataSet.

Parameters:

lhs The DataSet<> from which shall be read.

rhs The record to which the data shall be copied.

5.6 eos::impl Namespace Reference

Classes

- struct **Applicator**
- struct **Applicator**< **Result_, 0** >
- struct **Applicator**< **void, 0** >
- struct **ConvertTo**
- struct **TupleMaker**
- struct **TupleMaker**< **0** >
- struct **TupleSize**< **std::tuple**< **TupleElements_...**> >
- struct **SimpleDestringify**
- struct **DoDestringify**< **double** >
- struct **DoDestringify**< **unsigned** >
- struct **DoDestringify**< **long** >
- struct **DoDestringify**< **unsigned long** >
- struct **DoDestringify**< **bool** >
- struct **PowerOf**
- struct **PowerOf**< **0, T_** >

5.7 eos::implementation Namespace Reference

Classes

- struct ResultOf< Result_(Class_::*)(Args_...)>
- struct ResultOf< Result_(*)(Args_...)>
- struct DoStringify
- struct **DoStringify**< **std::string** >

5.8 eos::InstantiationPolicy Namespace Reference

Classes

- class DeleteOnDestruction

5.9 eos::proposal_functions Namespace Reference

Classes

- struct UnknownProposalError
- struct Factory

  Abstract factory to read in a proposal density from file.

- class AdjacencyMatrix
- class GlobalLocal
- class Multivariate
- class MultivariateGaussian
- class MultivariateStudentT
- struct MultivariateAccess
- class BlockDecomposition

Typedefs

- typedef hdf5::Composite< hdf5::Scalar< const char * >, hdf5::Scalar< unsigned > > MetaType
- typedef std::tuple< const char *, unsigned > MetaRecord
- typedef std::shared_ptr< Multivariate > MultivariateProposalPtr

Functions

- unsigned random_index (const std::vector< double > &cumulative, gsl_rng *rng)
- void sliding_window (const unsigned &K, const unsigned &size, const unsigned &j, unsigned &j_min, unsigned &j_max)
- MetaType meta_type ()
- MetaRecord meta_record ()

5.9.1 Typedef Documentation

5.9.1.1 typedef std::tuple<const char *, unsigned> eos::proposal_functions::MetaRecord

5.9.1.2 typedef hdf5::Composite<hdf5::Scalar<const char *>, hdf5::Scalar<unsigned> > eos::proposal_functions::MetaType

This data type descriptor is needed to identify the proposal function type

5.9.1.3 typedef std::shared_ptr<Multivariate> eos::proposal_functions::MultivariateProposalPtr

5.9.2 Function Documentation

5.9.2.1 MetaRecord eos::proposal_functions::meta_record ()

5.9.2.2 MetaType eos::proposal_functions::meta_type ()

5.9.2.3 unsigned eos::proposal_functions::random_index (const std::vector< double > & *cumulative*, gsl_rng * *rng*)

For a discrete probability distribution defined by its cumulative, draw a random state given by its index in the cumulative.

5.9.2.4 void eos::proposal_functions::sliding_window (const unsigned & *K*, const unsigned & *size*, const unsigned & *j*, unsigned & *j_min*, unsigned & *j_max*)

Find the indices [j_min, j_max[such that they cover range of size within [0, K] around j

5.10 ROOT Namespace Reference

Namespaces

- namespace Minuit2

5.11 ROOT::Minuit2 Namespace Reference

5.12 std Namespace Reference

STL namespace.

Classes

- struct **__TupleHasher**
- struct **__TupleHasher**< **0, T_...**>
- struct **hash**< **std::tuple**< **T_...**> >
- class **allocator**

  STL class.

- class **auto_ptr**

  STL class.

- class **ios_base**

  STL class.

- class **basic_ios**

  STL class.

- class **basic_istream**

  STL class.

- class **basic_ostream**

  STL class.

- class **basic_iostream**

  STL class.

- class **basic_ifstream**

  STL class.

- class **basic_ofstream**

  STL class.

- class **basic_fstream**

  STL class.

- class **basic_istringstream**

  STL class.

- class **basic_ostringstream**

  STL class.

- class **basic_stringstream**

  STL class.

- class **ios**

STL class.

- class **wios**

  STL class.

- class **istream**

  STL class.

- class **wistream**

  STL class.

- class **ostream**

  STL class.

- class **wostream**

  STL class.

- class **ifstream**

  STL class.

- class **wifstream**

  STL class.

- class **ofstream**

  STL class.

- class **wofstream**

  STL class.

- class **fstream**

  STL class.

- class **wfstream**

  STL class.

- class **istringstream**

  STL class.

- class **wistringstream**

  STL class.

- class **ostringstream**

  STL class.

- class **wostringstream**

  STL class.

- class **stringstream**

  STL class.

- class **wstringstream**
  STL class.
- class **basic_string**
  STL class.
- class **string**
  STL class.
- class **wstring**
  STL class.
- class **complex**
  STL class.
- class **bitset**
  STL class.
- class **deque**
  STL class.
- class **list**
  STL class.
- class **map**
  STL class.
- class **multimap**
  STL class.
- class **set**
  STL class.
- class **multiset**
  STL class.
- class **vector**
  STL class.
- class **queue**
  STL class.
- class **priority_queue**
  STL class.
- class **stack**
  STL class.
- class **valarray**
  STL class.

- class **exception**
  STL class.
- class **bad_alloc**
  STL class.
- class **bad_cast**
  STL class.
- class **bad_typeid**
  STL class.
- class **logic_error**
  STL class.
- class **runtime_error**
  STL class.
- class **bad_exception**
  STL class.
- class **domain_error**
  STL class.
- class **invalid_argument**
  STL class.
- class **length_error**
  STL class.
- class **out_of_range**
  STL class.
- class **range_error**
  STL class.
- class **overflow_error**
  STL class.
- class **underflow_error**
  STL class.

5.12.1 Detailed Description

STL namespace.

Chapter 6

Class Documentation

6.1 eos::AbsoluteTestObservable Struct Reference

`#include <analysis_TEST.hh>`

Inherits eos::TestObservable.

Public Member Functions

- AbsoluteTestObservable (const Parameters &p, const Kinematics &k, const std::string &mass_name)
- virtual ~AbsoluteTestObservable ()
- virtual double evaluate () const
- virtual ObservablePtr clone () const
- virtual ObservablePtr clone (const Parameters &parameters) const

6.1.1 Constructor & Destructor Documentation

6.1.1.1 eos::AbsoluteTestObservable::AbsoluteTestObservable (const Parameters & *p*, const Kinematics & *k*, const std::string & *mass_name*) `[inline]`

6.1.1.2 virtual eos::AbsoluteTestObservable::~AbsoluteTestObservable () `[inline, virtual]`

6.1.2 Member Function Documentation

6.1.2.1 virtual ObservablePtr eos::AbsoluteTestObservable::clone (const Parameters & *parameters*) const `[inline, virtual]`

Reimplemented from eos::TestObservable.

6.1.2.2 virtual ObservablePtr eos::AbsoluteTestObservable::clone () const `[inline, virtual]`

Reimplemented from eos::TestObservable.

6.1.2.3 virtual double eos::AbsoluteTestObservable::evaluate () const [inline, virtual]

Reimplemented from eos::TestObservable.

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

- /home/alexshires/EOS/eos-next/eos/utils/analysis_TEST.hh

6.2 eos::Accumulator Class Reference

```
#include <accumulator.hh>
```

Inherits PrivateImplementationPattern< Accumulator >.

Public Member Functions

- Accumulator (const double &value)
- ~Accumulator ()
- double operator+= (const double &addend)
- double value () const

6.2.1 Constructor & Destructor Documentation

6.2.1.1 eos::Accumulator::Accumulator (const double & *value*)

6.2.1.2 eos::Accumulator::~Accumulator ()

6.2.2 Member Function Documentation

6.2.2.1 double eos::Accumulator::operator+= (const double & *addend*)

6.2.2.2 double eos::Accumulator::value () const

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

- /home/alexshires/EOS/eos-next/eos/utils/accumulator.hh

6.3 eos::proposal_functions::AdjacencyMatrix Class Reference

```
#include <proposal_functions.hh>
```

Public Member Functions

- AdjacencyMatrix ()

  Empty and useless.

- void reserve (const unsigned &number_of_clusters)

  Reserve storage for fixed number of clusters.

- void add (const MarkovChain::State &)
- void indices (const std::vector< unsigned > index_list)
- const std::vector< double > & jump (const unsigned &h_x, const unsigned &h_y) const
- const unsigned & number_of_clusters () const
- const MarkovChain::State & state (const unsigned &i) const

6.3.1 Detailed Description

Store long jump vectors and retrieve them efficiently.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 eos::proposal_functions::AdjacencyMatrix::AdjacencyMatrix ()

Empty and useless.

6.3.3 Member Function Documentation

6.3.3.1 void eos::proposal_functions::AdjacencyMatrix::add (const MarkovChain::State &)

Add a state to the list, and compute the vector difference between the new and all existing ones.

Note:

Fails with an exception if there are enough states already.

6.3.3.2 void eos::proposal_functions::AdjacencyMatrix::indices (const std::vector< unsigned > *index_list*)

6.3.3.3 const std::vector<double>& eos::proposal_functions::AdjacencyMatrix::jump (const unsigned & *h_x*, const unsigned & *h_y*) const

Undirected jump excluding the sign.

Note:

The caller needs to decide if the sign of the components is to be flipped or not.

Parameters:

h_x Index of the component into which a jump is desired.

h_y Index of the component out out which a jump is desired.

Returns:

The vector of the difference in coordinates.

6.3.3.4 const unsigned& eos::proposal_functions::AdjacencyMatrix::number_of_clusters () const `[inline]`

6.3.3.5 void eos::proposal_functions::AdjacencyMatrix::reserve (const unsigned & *number_of_clusters*)

Reserve storage for fixed number of clusters.

6.3.3.6 const MarkovChain::State& eos::proposal_functions::AdjacencyMatrix::state (const unsigned & *i*) const

Retrieve the fixed state used for cluster i.

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.4 eos::btovll::Amplitudes Struct Reference

```
#include <exclusive-b-to-s-dilepton.hh>
```

Public Attributes

- complex< double > a_long_right
- complex< double > a_long_left
- complex< double > a_perp_right
- complex< double > a_perp_left
- complex< double > a_par_right
- complex< double > a_par_left
- complex< double > a_timelike

6.4.1 Member Data Documentation

6.4.1.1 complex<double> eos::btovll::Amplitudes::a_long_left

6.4.1.2 complex<double> eos::btovll::Amplitudes::a_long_right

6.4.1.3 complex<double> eos::btovll::Amplitudes::a_par_left

6.4.1.4 complex<double> eos::btovll::Amplitudes::a_par_right

6.4.1.5 complex<double> eos::btovll::Amplitudes::a_perp_left

6.4.1.6 complex<double> eos::btovll::Amplitudes::a_perp_right

6.4.1.7 complex<double> eos::btovll::Amplitudes::a_timelike

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton.hh

6.5 eos::Analysis Class Reference

```
#include <analysis.hh>
```

Inherits PrivateImplementationPattern< Analysis >.

Classes

- struct OptimizationOptions
- struct Output

Public Member Functions

- std::pair< std::vector< double >, double > optimize (const std::vector< double > &initial_guess, const OptimizationOptions &options)
- const ROOT::Minuit2::FunctionMinimum & optimize_minuit (const std::vector< double > &initial_guess, const OptimizationOptions &options)
- void restrict (const std::string &name, const double &min, const double &max)

Basic Functions

- Analysis (const LogLikelihood &log_likelihood)
- ~Analysis ()

  Destructor.

- AnalysisPtr clone () const

  Clone this Analysis.

Friends

- struct Implementation< Analysis >

Accessors

- const std::vector< ParameterDescription > & parameter_descriptions () const

  Retrieve a set of all parameters, including ranges.

- Parameter operator[ ] (const unsigned &index)
- Parameters parameters () const

  Retrieve our associates Parameters object.

- bool add (const LogPriorPtr &prior, bool nuisance=false)
- void dump_descriptions (hdf5::File &file, std::string data_set_base="/descriptions") const
- std::pair< double, double > goodness_of_fit (const std::vector< double > &parameter_values, const unsigned &simulated_datasets, std::string output_file="")
- LogLikelihood log_likelihood ()

  Retrieve the overall Log(likelihood) for this analysis.

- double log_prior ()

Retrieve the overall Log(prior) for this analysis.

- LogPriorPtr log_prior (const std::string &name) const
- double log_posterior ()
- bool nuisance (const std::string &name) const
- static std::vector< ParameterDescription > read_descriptions (const hdf5::File &file, std::string data_set_base="/descriptions")

6.5.1 Constructor & Destructor Documentation

6.5.1.1 eos::Analysis::Analysis (const LogLikelihood & *log_likelihood*)

Constructor.

Extracts parameters, observables from LogLikelihood. The default prior (flat) is assumed for all parameters.

Parameters:

log_likelihood The LogLikelihood functor which shall be analysed.

Note:

Analysis assumes ownership of log_likelihood

6.5.1.2 eos::Analysis::~Analysis ()

Destructor.

6.5.2 Member Function Documentation

6.5.2.1 bool eos::Analysis::add (const LogPriorPtr & *prior*, bool *nuisance* = `false`)

Add one or more parameters and associated prior density

Parameters:

prior The logarithmic prior density.

nuisance False for a parameter of interest

6.5.2.2 AnalysisPtr eos::Analysis::clone () const

Clone this Analysis.

6.5.2.3 void eos::Analysis::dump_descriptions (hdf5::File & *file*, std::string *data_set_base* = `"/descriptions"`) const

Write parameter descriptions, constraints, observables into the hdf5 file under the given group name.

6.5.2.4 std::pair<double, double> eos::Analysis::goodness_of_fit (const std::vector< double > & *parameter_values*, const unsigned & *simulated_datasets*, std::string *output_file* = "")

Calculate the p-value based on the χ^2 test statistic for fixed parameter_values

Parameters:

parameter_values

simulate if true, simulate data sets to estimate distribution of test statistic, else use the cumulative of the χ^2-distribution with (N-k) degrees-of-freedom, where N is the number of observations and k is the number of fitted parameters

Returns:

$< \chi^2$, p>

6.5.2.5 LogLikelihood eos::Analysis::log_likelihood ()

Retrieve the overall Log(likelihood) for this analysis.

6.5.2.6 double eos::Analysis::log_posterior ()

Retrieve the overall Log(posterior) for this analysis. Incorporate normalization constant, the evidence here in getter if available.

6.5.2.7 LogPriorPtr eos::Analysis::log_prior (const std::string & *name*) const

Find the prior for a given parameter

6.5.2.8 double eos::Analysis::log_prior ()

Retrieve the overall Log(prior) for this analysis.

6.5.2.9 bool eos::Analysis::nuisance (const std::string & *name*) const

Check if a given parameter is a nuisance parameter for this Analysis.

Parameters:

name The name of the parameter we are interested in.

6.5.2.10 Parameter eos::Analysis::operator[ ] (const unsigned & *index*)

Retrieve a parameter by index.

Parameters:

index The index of the parameter.

6.5.2.11 std::pair<std::vector<double>, double> eos::Analysis::optimize (const std::vector< double > & *initial_guess*, const OptimizationOptions & *options*)

Optimize the posterior using the Nelder-Mead simplex algorithm.

Parameters:

initial_guess Starting point for simplex construction

options If no tuning desired, use Analysis::OptimizationOptions::Defaults()

Returns:

<parameter values at mode, posterior value at mode>

6.5.2.12 const ROOT::Minuit2::FunctionMinimum& eos::Analysis::optimize_minuit (const std::vector< double > & *initial_guess*, const OptimizationOptions & *options*)

6.5.2.13 const std::vector<ParameterDescription>& eos::Analysis::parameter_descriptions () const

Retrieve a set of all parameters, including ranges.

6.5.2.14 Parameters eos::Analysis::parameters () const

Retrieve our associates Parameters object.

6.5.2.15 static std::vector<ParameterDescription> eos::Analysis::read_descriptions (const hdf5::File & *file*, std::string *data_set_base* = `"/descriptions"`) `[static]`

Read in parameter descriptions from a previous dump.

Parameters:

sample_file The HDF5 file with the information.

base The base directory where to look for the data set.

Returns:

The descriptions, one per parameter

6.5.2.16 void eos::Analysis::restrict (const std::string & *name*, const double & *min*, const double & *max*)

Restrict to a subrange of a given parameter.

Parameters:

name

min

max

6.5.3 Friends And Related Function Documentation

6.5.3.1 friend struct Implementation< Analysis > [friend]

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

- /home/alexshires/EOS/eos-next/eos/utils/analysis.hh

6.6 eos::AnalyticFormFactorBToPiDKMMO2008 Class Reference

```
#include <analytic-b-to-pi.hh>
```

Inherits eos::FormFactors< PToP >, and PrivateImplementationPattern< AnalyticFormFactorBToPiDKMMO2008 >.

Public Member Functions

- AnalyticFormFactorBToPiDKMMO2008 (const Parameters &, const Options &)
- ~AnalyticFormFactorBToPiDKMMO2008 ()
- double F_lo_tw2 (const double &q2) const
- double F_lo_tw3 (const double &q2) const
- double F_lo_tw4 (const double &q2) const
- double F_nlo_tw2 (const double &q2) const
- double F_nlo_tw3 (const double &q2) const
- virtual double f_p (const double &q2) const
- virtual double f_0 (const double &q2) const
- virtual double f_t (const double &q2) const
- double MB_lcsr (const double &q2) const
- double MB_svz () const
- Diagnostics diagnostics () const

Static Public Member Functions

- static FormFactors< PToP > * make (const Parameters &, unsigned)

6.6.1 Constructor & Destructor Documentation

6.6.1.1 eos::AnalyticFormFactorBToPiDKMMO2008::AnalyticFormFactorBToPiDKMMO2008 (const Parameters &, const Options &)

6.6.1.2 eos::AnalyticFormFactorBToPiDKMMO2008::~AnalyticFormFactorBToPiDKMMO2008 ()

6.6.2 Member Function Documentation

6.6.2.1 Diagnostics eos::AnalyticFormFactorBToPiDKMMO2008::diagnostics () const

6.6.2.2 virtual double eos::AnalyticFormFactorBToPiDKMMO2008::f_0 (const double & *q2*) const [virtual]

Implements eos::FormFactors< PToP >.

6.6.2.3 double eos::AnalyticFormFactorBToPiDKMMO2008::F_lo_tw2 (const double & *q2*) const

6.6.2.4 double eos::AnalyticFormFactorBToPiDKMMO2008::F_lo_tw3 (const double & *q2*) const

6.6.2.5 double eos::AnalyticFormFactorBToPiDKMMO2008::F_lo_tw4 (const double & *q2*) const

6.6.2.6 double eos::AnalyticFormFactorBToPiDKMMO2008::F_nlo_tw2 (const double & *q2*) const

6.6.2.7 double eos::AnalyticFormFactorBToPiDKMMO2008::F_nlo_tw3 (const double & *q2*) const

6.6.2.8 virtual double eos::AnalyticFormFactorBToPiDKMMO2008::f_p (const double & *q2*) const `[virtual]`

Implements eos::FormFactors< PToP >.

6.6.2.9 virtual double eos::AnalyticFormFactorBToPiDKMMO2008::f_t (const double & *q2*) const `[virtual]`

Implements eos::FormFactors< PToP >.

6.6.2.10 static FormFactors<PToP>* eos::AnalyticFormFactorBToPiDKMMO2008::make (const Parameters &, unsigned) `[static]`

6.6.2.11 double eos::AnalyticFormFactorBToPiDKMMO2008::MB_lcsr (const double & *q2*) const

6.6.2.12 double eos::AnalyticFormFactorBToPiDKMMO2008::MB_svz () const

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

- /home/alexshires/EOS/eos-next/eos/form-factors/analytic-b-to-pi.hh

6.7 eos::btovll::AngularCoefficients Struct Reference

```
#include <exclusive-b-to-s-dilepton.hh>
```

Public Attributes

- double j1s
- double j1c
- double j2s
- double j2c
- double j3
- double j4
- double j5
- double j6s
- double j6c
- double j7
- double j8
- double j9

6.7.1 Member Data Documentation

6.7.1.1 double eos::btovll::AngularCoefficients::j1c

6.7.1.2 double eos::btovll::AngularCoefficients::j1s

6.7.1.3 double eos::btovll::AngularCoefficients::j2c

6.7.1.4 double eos::btovll::AngularCoefficients::j2s

6.7.1.5 double eos::btovll::AngularCoefficients::j3

6.7.1.6 double eos::btovll::AngularCoefficients::j4

6.7.1.7 double eos::btovll::AngularCoefficients::j5

6.7.1.8 double eos::btovll::AngularCoefficients::j6c

6.7.1.9 double eos::btovll::AngularCoefficients::j6s

6.7.1.10 double eos::btovll::AngularCoefficients::j7

6.7.1.11 double eos::btovll::AngularCoefficients::j8

6.7.1.12 double eos::btovll::AngularCoefficients::j9

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton.hh

6.8 eos::hdf5::Array< rank_, T_ > Class Template Reference

`#include <hdf5.hh>`

Inherits eos::hdf5::Type.

Public Types

- typedef std::vector< T_ > Type

Public Member Functions

- Array (const std::string &name, const std::initializer_list< hsize_t > &dimensions)
- virtual const std::string & name () const
- virtual hid_t type_id () const
- virtual hsize_t size () const
- virtual void copy_to_hdf5 (const void *src, void *dest) const
- virtual void copy_from_hdf5 (const void *src, void *dest) const

template<unsigned rank_, typename T_> class eos::hdf5::Array< rank_, T_ >

6.8.1 Member Typedef Documentation

6.8.1.1 template<unsigned rank_, typename T_ > typedef std::vector<T_> eos::hdf5::Array< rank_, T_ >::Type

6.8.2 Constructor & Destructor Documentation

6.8.2.1 template<unsigned rank_, typename T_ > eos::hdf5::Array< rank_, T_ >::Array (const std::string & *name*, const std::initializer_list< hsize_t > & *dimensions*) `[inline]`

6.8.3 Member Function Documentation

6.8.3.1 template<unsigned rank_, typename T_ > virtual void eos::hdf5::Array< rank_, T_ >::copy_from_hdf5 (const void * *src*, void * *dest*) const `[inline, virtual]`

Implements eos::hdf5::Type.

6.8.3.2 template<unsigned rank_, typename T_ > virtual void eos::hdf5::Array< rank_, T_ >::copy_to_hdf5 (const void * *src*, void * *dest*) const `[inline, virtual]`

Implements eos::hdf5::Type.

6.8.3.3 template<unsigned rank_, typename T_ > virtual const std::string& eos::hdf5::Array< rank_, T_ >::name () const `[inline, virtual]`

Implements eos::hdf5::Type.

6.8.3.4 template<unsigned rank_, typename T_ > virtual hsize_t eos::hdf5::Array< rank_, T_ >::size () const [inline, virtual]

Implements eos::hdf5::Type.

6.8.3.5 template<unsigned rank_, typename T_ > virtual hid_t eos::hdf5::Array< rank_, T_ >::type_id () const [inline, virtual]

Implements eos::hdf5::Type.

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.9 eos::hdf5::Attribute< T_ > Class Template Reference

```
#include <hdf5.hh>
```

Public Types

- typedef T_::Type RecordType

Public Member Functions

- RecordType value () const
- Attribute & operator= (const RecordType &rhs)

Friends

- class DataSet

template<typename T_> class eos::hdf5::Attribute< T_ >

6.9.1 Member Typedef Documentation

6.9.1.1 template<typename T_> typedef T_::Type eos::hdf5::Attribute< T_ >::RecordType

6.9.2 Member Function Documentation

6.9.2.1 template<typename T_> Attribute& eos::hdf5::Attribute< T_ >::operator= (const RecordType & *rhs*) `[inline]`

6.9.2.2 template<typename T_> RecordType eos::hdf5::Attribute< T_ >::value () const `[inline]`

6.9.3 Friends And Related Function Documentation

6.9.3.1 template<typename T_> friend class DataSet `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.10 eos::hdf5::AttributeHandle Class Reference

`#include <hdf5.hh>`

Inherits PrivateImplementationPattern< AttributeHandle >.

Public Member Functions

- ~AttributeHandle ()
- hid_t attribute_id () const
- hid_t type_id () const
- void write (const void *buffer)
- void read (void *buffer) const

Friends

- class DataSetHandle

6.10.1 Constructor & Destructor Documentation

6.10.1.1 eos::hdf5::AttributeHandle::~AttributeHandle ()

6.10.2 Member Function Documentation

6.10.2.1 hid_t eos::hdf5::AttributeHandle::attribute_id () const

6.10.2.2 void eos::hdf5::AttributeHandle::read (void * *buffer*) const

6.10.2.3 hid_t eos::hdf5::AttributeHandle::type_id () const

6.10.2.4 void eos::hdf5::AttributeHandle::write (const void * *buffer*)

6.10.3 Friends And Related Function Documentation

6.10.3.1 friend class DataSetHandle `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.11 eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ > Struct Template Reference

```
#include <wrapped_forward_iterator-impl.hh>
```

Inherited by eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >.

Public Member Functions

- virtual Base ∗ clone () const =0
- virtual void increment ()=0
- virtual WrappedForwardIterator< Tag_, Value_ >::pointer pointer () const =0
- virtual WrappedForwardIterator< Tag_, Value_ >::reference reference () const =0
- virtual bool equal (const Base ∗) const =0
- virtual void ∗ underlying_iterator_ptr ()=0
- virtual const void ∗ underlying_iterator_ptr () const =0
- virtual ∼Base ()

template<typename Tag_, typename Value_>template<typename Tag_, typename Value_> struct eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >

6.11.1 Constructor & Destructor Documentation

6.11.1.1 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::∼Base () [inline, virtual]

6.11.2 Member Function Documentation

6.11.2.1 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual Base∗ eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::clone () const [pure virtual]

Implemented in eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >.

6.11.2.2 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual bool eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::equal (const Base< Tag_, Value_ > ∗) const [pure virtual]

6.11.2.3 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual void eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::increment () [pure virtual]

Implemented in eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >.

6.11.2.4 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual WrappedForwardIterator<Tag_, Value_>::pointer eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::pointer () const [pure virtual]

Implemented in eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >.

6.11.2.5 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual WrappedForwardIterator<Tag_, Value_>::reference eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::reference () const [pure virtual]

Implemented in eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >.

6.11.2.6 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual const void* eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::underlying_iterator_ptr () const [pure virtual]

Implemented in eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >.

6.11.2.7 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > virtual void* eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >::underlying_iterator_ptr () [pure virtual]

Implemented in eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >.

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

- /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-impl.hh

6.12 eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ > Struct Template Reference

```
#include <wrapped_forward_iterator-impl.hh>
```

Inherits eos::WrappedForwardIterator::Base.

Public Member Functions

- BaseImpl (const Iter_ &ii)
- Base ∗ clone () const
- void increment ()
- WrappedForwardIterator< Tag_, Value_ >::reference reference () const
- WrappedForwardIterator< Tag_, Value_ >::pointer pointer () const
- bool equal (const Base ∗other) const
- void ∗ underlying_iterator_ptr ()
- const void ∗ underlying_iterator_ptr () const

Public Attributes

- Iter_ i

template<typename Tag_, typename Value_>template<typename Tag_, typename Value_> struct eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >

6.12.1 Constructor & Destructor Documentation

6.12.1.1 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::BaseImpl (const Iter_ & *ii*) `[inline]`

6.12.2 Member Function Documentation

6.12.2.1 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > Base∗ eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::clone () const `[inline, virtual]`

Implements eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >.

6.12.2.2 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > bool eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::equal (const Base ∗ *other*) const `[inline]`

6.12.2.3 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > void eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::increment () `[inline, virtual]`

Implements eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >.

6.12.2.4 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > WrappedForwardIterator<Tag_, Value_>::pointer eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::pointer () const [inline, virtual]

Implements eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >.

6.12.2.5 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > WrappedForwardIterator<Tag_, Value_>::reference eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::reference () const [inline, virtual]

Implements eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >.

6.12.2.6 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > const void* eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::underlying_iterator_ptr () const [inline, virtual]

Implements eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >.

6.12.2.7 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > void* eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::underlying_iterator_ptr () [inline, virtual]

Implements eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >.

6.12.3 Member Data Documentation

6.12.3.1 template<typename Tag_, typename Value_> template<typename Tag_ , typename Value_ > Iter_ eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >::i

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

- /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-impl.hh

6.13 eos::BCL2008FormFactors< Process_ > Class Template Reference

```
#include <mesonic-impl.hh>
```

Public Member Functions

- BCL2008FormFactors (const Parameters &p, const Options &)
- virtual double f_p (const double &s) const
- virtual double f_0 (const double &s) const
- virtual double f_t (const double &s) const

Static Public Member Functions

- static FormFactors< PToP > ∗ make (const Parameters &parameters, unsigned)

template<typename Process_> class eos::BCL2008FormFactors< Process_ >

6.13.1 Constructor & Destructor Documentation

6.13.1.1 template<typename Process_ > eos::BCL2008FormFactors< Process_ >::BCL2008FormFactors (const Parameters & *p*, const Options &) `[inline]`

6.13.2 Member Function Documentation

6.13.2.1 template<typename Process_ > virtual double eos::BCL2008FormFactors< Process_ >::f_0 (const double & *s*) const `[inline, virtual]`

6.13.2.2 template<typename Process_ > virtual double eos::BCL2008FormFactors< Process_ >::f_p (const double & *s*) const `[inline, virtual]`

6.13.2.3 template<typename Process_ > virtual double eos::BCL2008FormFactors< Process_ >::f_t (const double & *s*) const `[inline, virtual]`

6.13.2.4 template<typename Process_ > static FormFactors<PToP>∗ eos::BCL2008FormFactors< Process_ >::make (const Parameters & *parameters*, unsigned) `[inline, static]`

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.14 eos::BFW2010FormFactors< PToP > Class Template Reference

`#include <mesonic-impl.hh>`

Inherits eos::FormFactors< PToP >.

Public Member Functions

- BFW2010FormFactors (const Parameters &p, const Options &)
- virtual double f_p (const double &s) const
- virtual double f_0 (const double &s) const
- virtual double f_t (const double &s) const

Static Public Member Functions

- static FormFactors< PToP > * make (const Parameters &parameters, unsigned)

template<> class eos::BFW2010FormFactors< PToP >

6.14.1 Constructor & Destructor Documentation

6.14.1.1 eos::BFW2010FormFactors< PToP >::BFW2010FormFactors (const Parameters & *p*, const Options &) `[inline]`

6.14.2 Member Function Documentation

6.14.2.1 virtual double eos::BFW2010FormFactors< PToP >::f_0 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.14.2.2 virtual double eos::BFW2010FormFactors< PToP >::f_p (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.14.2.3 virtual double eos::BFW2010FormFactors< PToP >::f_t (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.14.2.4 static FormFactors<PToP>* eos::BFW2010FormFactors< PToP >::make (const Parameters & *parameters*, unsigned) `[inline, static]`

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.15 eos::Histogram< 2 >::Bin Struct Reference

```
#include <histogram.hh>
```

Public Member Functions

- bool operator< (const Bin &other) const

Basic Functions

- Bin (const std::array< double, 2 > &lower, const std::array< double, 2 > &upper, const double &value=0.0)

  Destructor.

- ~Bin ()

  Destructor.

Public Attributes

- std::array< double, 2 > lower

  Lower limit of the Bin's interval for each dimension.

- std::array< double, 2 > upper

  Upper limit of the Bin's interval for each dimension.

- double value

  Content of the Bin.

6.15.1 Detailed Description

template<> struct eos::Histogram< 2 >::Bin

Holds information on any one of Histogram<2>'s bins.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 eos::Histogram< 2 >::Bin::Bin (const std::array< double, 2 > & *lower*, const std::array< double, 2 > & *upper*, const double & *value* = 0.0) [inline]

Destructor.

6.15.2.2 eos::Histogram< 2 >::Bin::~Bin () [inline]

Destructor.

6.15.3 Member Function Documentation

6.15.3.1 bool eos::Histogram< 2 >::Bin::operator< (const Bin & *other*) const [inline]

Compare if this bin is of lower order than another bin.

Parameters:

other The other Bin.

6.15.4 Member Data Documentation

6.15.4.1 std::array<double, 2> eos::Histogram< 2 >::Bin::lower

Lower limit of the Bin's interval for each dimension.

6.15.4.2 std::array<double, 2> eos::Histogram< 2 >::Bin::upper

Upper limit of the Bin's interval for each dimension.

6.15.4.3 double eos::Histogram< 2 >::Bin::value

Content of the Bin.

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

- /home/alexshires/EOS/eos-next/eos/utils/histogram.hh

6.16 eos::Histogram< 1 >::Bin Struct Reference

```
#include <histogram.hh>
```

Public Member Functions

- bool operator< (const Bin &other) const

Basic Functions

- Bin (const double &lower, const double &upper, const double &value=0.0)
  Destructor.
- ~Bin ()
  Destructor.

Public Attributes

- double lower
  Lower limit of the Bin's interval.
- double upper
  Width of the Bin's interval.
- double value
  Content of the Bin.

6.16.1 Detailed Description

template<> struct eos::Histogram< 1 >::Bin

Holds information on any one of Histogram<1>'s bins.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 eos::Histogram< 1 >::Bin::Bin (const double & *lower*, const double & *upper*, const double & *value* = 0.0) [inline]

Destructor.

6.16.2.2 eos::Histogram< 1 >::Bin::~Bin () [inline]

Destructor.

6.16.3 Member Function Documentation

6.16.3.1 bool eos::Histogram< 1 >::Bin::operator< (const Bin & *other*) const [inline]

Compare if this bin is left of another bin.

Parameters:

other The other Bin.

6.16.4 Member Data Documentation

6.16.4.1 double eos::Histogram< 1 >::Bin::lower

Lower limit of the Bin's interval.

6.16.4.2 double eos::Histogram< 1 >::Bin::upper

Width of the Bin's interval.

6.16.4.3 double eos::Histogram< 1 >::Bin::value

Content of the Bin.

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

- /home/alexshires/EOS/eos-next/eos/utils/histogram.hh

6.17 eos::proposal_functions::BlockDecomposition Class Reference

`#include <proposal_functions.hh>`

Inherits eos::MarkovChain::ProposalFunction.

Public Types

- typedef hdf5::Composite< hdf5::Scalar< const char * > > PriorsType

Public Member Functions

- BlockDecomposition ()
- virtual ~BlockDecomposition ()
- virtual void adapt (const MarkovChain::State::Iterator &begin, const MarkovChain::State::Iterator &end, const double &efficiency, const double &efficiency_min, const double &efficiency_max)
- void add (const MultivariateProposalPtr &)
- void add (const LogPriorPtr &)
- virtual ProposalFunctionPtr clone () const

  Create independent copy.

- virtual void dump_state (hdf5::File &file, const std::string &data_set_base_name) const

  Store state in the file under the given base name.

- virtual double evaluate (const MarkovChain::State &x, const MarkovChain::State &y) const

  Evaluate the density to propose x given y.

- virtual void propose (MarkovChain::State &x, const MarkovChain::State &y, gsl_rng *rng) const

  Obtain from the density a proposal x given y.

Static Public Member Functions

- static PriorsType priors_type ()

Friends

- class MultivariateAccess

6.17.1 Member Typedef Documentation

6.17.1.1 typedef hdf5::Composite<hdf5::Scalar<const char *> > eos::proposal_functions::BlockDecomposition::PriorsType

6.17.2 Constructor & Destructor Documentation

6.17.2.1 eos::proposal_functions::BlockDecomposition::BlockDecomposition ()

6.17.2.2 virtual eos::proposal_functions::BlockDecomposition::~BlockDecomposition () [virtual]

6.17.3 Member Function Documentation

6.17.3.1 virtual void eos::proposal_functions::BlockDecomposition::adapt (const MarkovChain::State::Iterator & *begin*, const MarkovChain::State::Iterator & *end*, const double & *efficiency*, const double & *efficiency_min*, const double & *efficiency_max*) [virtual]

Adapt the proposal function to the chain's current state and history.

Note:

adapt() always uses the full history passed as an argument. If only a subset of an existing history is to be used, the caller is responsible for removing the unneeded parts from history before calling adapt().

Parameters:

history

Implements eos::MarkovChain::ProposalFunction.

6.17.3.2 void eos::proposal_functions::BlockDecomposition::add (const LogPriorPtr &)

6.17.3.3 void eos::proposal_functions::BlockDecomposition::add (const MultivariateProposalPtr &)

6.17.3.4 virtual ProposalFunctionPtr eos::proposal_functions::BlockDecomposition::clone () const [virtual]

Create independent copy.

Implements eos::MarkovChain::ProposalFunction.

6.17.3.5 virtual void eos::proposal_functions::BlockDecomposition::dump_state (hdf5::File & *file*, const std::string & *data_set_base_name*) const [virtual]

Store state in the file under the given base name.

Implements eos::MarkovChain::ProposalFunction.

6.17.3.6 virtual double eos::proposal_functions::BlockDecomposition::evaluate (const MarkovChain::State & *x*, const MarkovChain::State & *y*) const [virtual]

Evaluate the density to propose x given y.

Implements eos::MarkovChain::ProposalFunction.

6.17.3.7 static PriorsType eos::proposal_functions::BlockDecomposition::priors_type () [static]

6.17.3.8 virtual void eos::proposal_functions::BlockDecomposition::propose (MarkovChain::State & *x*, const MarkovChain::State & *y*, gsl_rng * *rng*) const [virtual]

Obtain from the density a proposal x given y.

Implements eos::MarkovChain::ProposalFunction.

6.17.4 Friends And Related Function Documentation

6.17.4.1 friend class MultivariateAccess [friend]

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.18 eos::BMesonProperties Class Reference

```
#include <properties.hh>
```

Inherits eos::ParameterUser, and PrivateImplementationPattern< BMesonProperties >.

Public Member Functions

- BMesonProperties (const Parameters &parameters, const Options &options)
- ~BMesonProperties ()
- double mass_splitting_j1_j0 () const

6.18.1 Constructor & Destructor Documentation

6.18.1.1 eos::BMesonProperties::BMesonProperties (const Parameters & *parameters*, const Options & *options*)

6.18.1.2 eos::BMesonProperties::~BMesonProperties ()

6.18.2 Member Function Documentation

6.18.2.1 double eos::BMesonProperties::mass_splitting_j1_j0 () const

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

- /home/alexshires/EOS/eos-next/eos/b-decays/properties.hh

6.19 eos::Bremsstrahlung Struct Reference

```
#include <bremsstrahlung.hh>
```

Static Public Member Functions

- static complex< double > G_m1 (const double &t)
- static complex< double > G_0 (const double &t)
- static complex< double > Deltai_23 (const double &s_hat, const double &w, const double &z)
- static complex< double > Deltai_27 (const double &s_hat, const double &w, const double &z)
- static complex< double > tau_22 (const double &s_hat, const double &w, const double &z)
- static complex< double > tau_27 (const double &s_hat, const double &w, const double &z)
- static complex< double > tau_28 (const double &s_hat, const double &w, const double &z)
- static complex< double > tau_29 (const double &s_hat, const double &w, const double &z)
- static double tau_78 (const double &s_hat)
- static double tau_88 (const double &s_hat)
- static double tau_89 (const double &s_hat)
- static complex< double > itau_22 (const double &s_hat, const double &z)
- static complex< double > itau_27 (const double &s_hat, const double &z)
- static complex< double > itau_28 (const double &s_hat, const double &z)
- static complex< double > itau_29 (const double &s_hat, const double &z)

6.19.1 Member Function Documentation

6.19.1.1 static complex<double> eos::Bremsstrahlung::Deltai_23 (const double & *s_hat*, const double & *w*, const double & *z*) [static]

6.19.1.2 static complex<double> eos::Bremsstrahlung::Deltai_27 (const double & *s_hat*, const double & *w*, const double & *z*) [static]

6.19.1.3 static complex<double> eos::Bremsstrahlung::G_0 (const double & *t*) [static]

6.19.1.4 static complex<double> eos::Bremsstrahlung::G_m1 (const double & *t*) [static]

6.19.1.5 static complex<double> eos::Bremsstrahlung::itau_22 (const double & *s_hat*, const double & *z*) [static]

6.19.1.6 static complex<double> eos::Bremsstrahlung::itau_27 (const double & *s_hat*, const double & *z*) [static]

6.19.1.7 static complex<double> eos::Bremsstrahlung::itau_28 (const double & *s_hat*, const double & *z*) [static]

6.19.1.8 static complex<double> eos::Bremsstrahlung::itau_29 (const double & *s_hat*, const double & *z*) [static]

6.19.1.9 static complex<double> eos::Bremsstrahlung::tau_22 (const double & *s_hat*, const double & *w*, const double & *z*) [static]

6.19.1.10 static complex<double> eos::Bremsstrahlung::tau_27 (const double & *s_hat*, const double & *w*, const double & *z*) [static]

6.19.1.11 static complex<double> eos::Bremsstrahlung::tau_28 (const double & *s_hat*, const double & *w*, const double & *z*) [static]

6.19.1.12 static complex<double> eos::Bremsstrahlung::tau_29 (const double & *s_hat*, const double & *w*, const double & *z*) [static]

6.19.1.13 static double eos::Bremsstrahlung::tau_78 (const double & *s_hat*) [static]

6.19.1.14 static double eos::Bremsstrahlung::tau_88 (const double & *s_hat*) [static]

6.19.1.15 static double eos::Bremsstrahlung::tau_89 (const double & *s_hat*) [static]

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/bremsstrahlung.hh

6.20 eos::BsToPhi Struct Reference

```
#include <mesonic-impl.hh>
```

Static Public Attributes

- static constexpr double mB = 5.366
- static constexpr double mV = 1.020

6.20.1 Member Data Documentation

6.20.1.1 constexpr double eos::BsToPhi::mB = 5.366 [static]

6.20.1.2 constexpr double eos::BsToPhi::mV = 1.020 [static]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.21 eos::BToDilepton Class Reference

```
#include <exclusive-b-to-dilepton.hh>
```

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToDilepton >.

Public Member Functions

Basic Functions

Constructor.

- BToDilepton (const Parameters &parameters, const Options &options)
  Destructor.
- ~BToDilepton ()
  Destructor.

Observables

- double branching_ratio_time_zero () const
- double branching_ratio_untagged_integrated () const

6.21.1 Detailed Description

Calculation according to [BEKU2002].

6.21.2 Constructor & Destructor Documentation

6.21.2.1 eos::BToDilepton::BToDilepton (const Parameters & *parameters*, const Options & *options*)

Destructor.

6.21.2.2 eos::BToDilepton::~BToDilepton ()

Destructor.

6.21.3 Member Function Documentation

6.21.3.1 double eos::BToDilepton::branching_ratio_time_zero () const

6.21.3.2 double eos::BToDilepton::branching_ratio_untagged_integrated () const

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-dilepton.hh

6.22 eos::BToK Struct Reference

```
#include <mesonic-impl.hh>
```

Public Types

- typedef PToP Transition

Static Public Attributes

- static constexpr const char ∗ label = "B->K"
- static constexpr const double m_B = 5.279
- static constexpr const double m_P = 0.492
- static constexpr const double m2_Br0m = 5.366 ∗ 5.366
- static constexpr const double m2_Br1m = 5.415 ∗ 5.415
- static constexpr const double tau_p = (m_B + m_P) ∗ (m_B + m_P)
- static constexpr const double tau_m = (m_B - m_P) ∗ (m_B - m_P)

6.22.1 Member Typedef Documentation

6.22.1.1 typedef PToP eos::BToK::Transition

6.22.2 Member Data Documentation

6.22.2.1 constexpr const char∗ eos::BToK::label = "B->K" [static]

6.22.2.2 constexpr const double eos::BToK::m2_Br0m = 5.366 ∗ 5.366 [static]

6.22.2.3 constexpr const double eos::BToK::m2_Br1m = 5.415 ∗ 5.415 [static]

6.22.2.4 constexpr const double eos::BToK::m_B = 5.279 [static]

6.22.2.5 constexpr const double eos::BToK::m_P = 0.492 [static]

6.22.2.6 constexpr const double eos::BToK::tau_m = (m_B - m_P) ∗ (m_B - m_P) [static]

6.22.2.7 constexpr const double eos::BToK::tau_p = (m_B + m_P) ∗ (m_B + m_P) [static]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.23 eos::BToKDilepton< LargeRecoil > Class Template Reference

`#include <exclusive-b-to-s-dilepton-large-recoil.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToKDilepton< LargeRecoil > >.

Public Member Functions

- BToKDilepton (const Parameters &parameters, const Options &options)
- ~BToKDilepton ()
- std::complex< double > F_A (const double &s) const
- std::complex< double > F_V (const double &s) const
- std::complex< double > F_S (const double &s) const
- std::complex< double > F_P (const double &s) const
- std::complex< double > F_T (const double &s) const
- std::complex< double > F_T5 (const double &s) const
- double a_l (const double &s) const
- double b_l (const double &s) const
- double c_l (const double &s) const
- double two_differential_decay_width (const double &s, const double &c_theta_l) const
- double differential_branching_ratio (const double &s) const
- double differential_flat_term (const double &s) const
- double differential_forward_backward_asymmetry (const double &s) const
- double differential_ratio_muons_electrons (const double &s) const
- double integrated_decay_width (const double &s_min, const double &s_max) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- double integrated_branching_ratio_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_cp_asymmetry (const double &s_min, const double &s_max) const
- double integrated_flat_term (const double &s_min, const double &s_max) const
- double integrated_flat_term_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_ratio_muons_electrons (const double &s_min, const double &s_max) const

template<> class eos::BToKDilepton< LargeRecoil >

6.23.1 Constructor & Destructor Documentation

6.23.1.1 eos::BToKDilepton< LargeRecoil >::BToKDilepton (const Parameters & *parameters*, const Options & *options*)

6.23.1.2 eos::BToKDilepton< LargeRecoil >::~BToKDilepton ()

6.23.2 Member Function Documentation

6.23.2.1 double eos::BToKDilepton< LargeRecoil >::a_l (const double & *s*) const

6.23.2.2 double eos::BToKDilepton< LargeRecoil >::b_l (const double & *s*) const

6.23.2.3 double eos::BToKDilepton< LargeRecoil >::c_l (const double & *s*) const

6.23.2.4 double eos::BToKDilepton< LargeRecoil >::differential_branching_ratio (const double & *s*) const

6.23.2.5 double eos::BToKDilepton< LargeRecoil >::differential_flat_term (const double & *s*) const

6.23.2.6 double eos::BToKDilepton< LargeRecoil >::differential_forward_backward_asymmetry (const double & *s*) const

6.23.2.7 double eos::BToKDilepton< LargeRecoil >::differential_ratio_muons_electrons (const double & *s*) const

6.23.2.8 std::complex<double> eos::BToKDilepton< LargeRecoil >::F_A (const double & *s*) const

6.23.2.9 std::complex<double> eos::BToKDilepton< LargeRecoil >::F_P (const double & *s*) const

6.23.2.10 std::complex<double> eos::BToKDilepton< LargeRecoil >::F_S (const double & *s*) const

6.23.2.11 std::complex<double> eos::BToKDilepton< LargeRecoil >::F_T (const double & *s*) const

6.23.2.12 std::complex<double> eos::BToKDilepton< LargeRecoil >::F_T5 (const double & *s*) const

6.23.2.13 std::complex<double> eos::BToKDilepton< LargeRecoil >::F_V (const double & *s*) const

6.23.2.14 double eos::BToKDilepton< LargeRecoil >::integrated_branching_ratio (const double & *s_min*, const double & *s_max*) const

6.23.2.15 double eos::BToKDilepton< LargeRecoil >::integrated_branching_ratio_cp_averaged (const double & *s_min*, const double & *s_max*) const

6.23.2.16 double eos::BToKDilepton< LargeRecoil >::integrated_cp_asymmetry (const double & *s_min*, const double & *s_max*) const

6.23.2.17 double eos::BToKDilepton< LargeRecoil >::integrated_decay_width (const double & *s_min*, const double & *s_max*) const

6.23.2.18 double eos::BToKDilepton< LargeRecoil >::integrated_flat_term (const double & *s_min*, const double & *s_max*) const

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-large-recoil.hh

6.24 eos::BToKDilepton< LowRecoil > Class Template Reference

`#include <exclusive-b-to-s-dilepton-low-recoil.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToKDilepton< LowRecoil > >.

Public Member Functions

- BToKDilepton (const Parameters &parameters, const Options &options)
- ~BToKDilepton ()
- double real_c9eff (const double &s) const
- double imag_c9eff (const double &s) const
- double real_c7eff (const double &s) const
- double imag_c7eff (const double &s) const
- double a_l (const double &s) const
- double b_l (const double &s) const
- double c_l (const double &s) const
- double two_differential_decay_width (const double &s, const double &c_theta_l) const
- double differential_branching_ratio (const double &s) const
- double differential_flat_term (const double &s) const
- double differential_forward_backward_asymmetry (const double &s) const
- double differential_ratio_muons_electrons (const double &s) const
- double integrated_decay_width (const double &s_min, const double &s_max) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- double integrated_branching_ratio_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_cp_asymmetry (const double &s_min, const double &s_max) const
- double integrated_flat_term (const double &s_min, const double &s_max) const
- double integrated_flat_term_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_ratio_muons_electrons (const double &s_min, const double &s_max) const

template<> class eos::BToKDilepton< LowRecoil >

6.24.1 Constructor & Destructor Documentation

6.24.1.1 eos::BToKDilepton< LowRecoil >::BToKDilepton (const Parameters & *parameters*, const Options & *options*)

6.24.1.2 eos::BToKDilepton< LowRecoil >::~BToKDilepton ()

6.24.2 Member Function Documentation

6.24.2.1 double eos::BToKDilepton< LowRecoil >::a_l (const double & *s*) const

6.24.2.2 double eos::BToKDilepton< LowRecoil >::b_l (const double & *s*) const

6.24.2.3 double eos::BToKDilepton< LowRecoil >::c_l (const double & *s*) const

6.24.2.4 double eos::BToKDilepton< LowRecoil >::differential_branching_ratio (const double & *s*) const

6.24.2.5 double eos::BToKDilepton< LowRecoil >::differential_flat_term (const double & *s*) const

6.24.2.6 double eos::BToKDilepton< LowRecoil >::differential_forward_backward_asymmetry (const double & *s*) const

6.24.2.7 double eos::BToKDilepton< LowRecoil >::differential_ratio_muons_electrons (const double & *s*) const

6.24.2.8 double eos::BToKDilepton< LowRecoil >::imag_c7eff (const double & *s*) const

6.24.2.9 double eos::BToKDilepton< LowRecoil >::imag_c9eff (const double & *s*) const

6.24.2.10 double eos::BToKDilepton< LowRecoil >::integrated_branching_ratio (const double & *s_min*, const double & *s_max*) const

6.24.2.11 double eos::BToKDilepton< LowRecoil >::integrated_branching_ratio_cp_averaged (const double & *s_min*, const double & *s_max*) const

6.24.2.12 double eos::BToKDilepton< LowRecoil >::integrated_cp_asymmetry (const double & *s_min*, const double & *s_max*) const

6.24.2.13 double eos::BToKDilepton< LowRecoil >::integrated_decay_width (const double & *s_min*, const double & *s_max*) const

6.24.2.14 double eos::BToKDilepton< LowRecoil >::integrated_flat_term (const double & *s_min*, const double & *s_max*) const

6.24.2.15 double eos::BToKDilepton< LowRecoil >::integrated_flat_term_cp_averaged (const double & *s_min*, const double & *s_max*) const

6.24.2.16 double eos::BToKDilepton< LowRecoil >::integrated_forward_backward_asymmetry (const double & *s_min*, const double & *s_max*) const

6.24.2.17 double eos::BToKDilepton< LowRecoil >::integrated_forward_backward_-asymmetry_cp_averaged (const double & *s_min*, const double & *s_max*) const

6.24.2.18 double eos::BToKDilepton< LowRecoil >::integrated_ratio_muons_electrons (const double & *s_min*, const double & *s_max*) const

6.24.2.19 double eos::BToKDilepton< LowRecoil >::real_c7eff (const double & *s*) const

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-low-recoil.hh

6.25 eos::BToKstar Struct Reference

```
#include <mesonic-impl.hh>
```

Static Public Attributes

- static constexpr double mB = 5.279
- static constexpr double mV = 0.896

6.25.1 Member Data Documentation

6.25.1.1 constexpr double eos::BToKstar::mB = 5.279 [static]

6.25.1.2 constexpr double eos::BToKstar::mV = 0.896 [static]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.26 eos::BToKstarDilepton< LargeRecoil > Class Template Reference

```
#include <exclusive-b-to-s-dilepton-large-recoil.hh>
```

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToKstarDilepton< LargeRecoil > >.

Public Member Functions

- BToKstarDilepton (const Parameters &parameters, const Options &options)
- ~BToKstarDilepton ()
- double xi_perp (const double &s) const
- double xi_para (const double &s) const
- complex< double > a_long (const Helicity &h, const double &s) const
- complex< double > a_perp (const Helicity &h, const double &s) const
- complex< double > a_par (const Helicity &h, const double &s) const
- double a_fb_zero_crossing () const
- double four_differential_decay_width (const double &s, const double &c_theta_l, const double &c_theta_k, const double &phi) const
- double differential_branching_ratio (const double &s) const
- double differential_decay_width (const double &s) const
- double differential_isospin_asymmetry (const double &s) const
- double differential_forward_backward_asymmetry (const double &s) const
- double differential_longitudinal_polarisation (const double &s) const
- double differential_transversal_polarisation (const double &s) const
- double differential_transverse_asymmetry_2 (const double &s) const
- double differential_transverse_asymmetry_3 (const double &s) const
- double differential_transverse_asymmetry_4 (const double &s) const
- double differential_transverse_asymmetry_5 (const double &s) const
- double differential_transverse_asymmetry_re (const double &s) const
- double differential_transverse_asymmetry_im (const double &s) const
- double differential_p_prime_4 (const double &s) const
- double differential_p_prime_5 (const double &s) const
- double differential_p_prime_6 (const double &s) const
- double differential_h_1 (const double &s) const
- double differential_h_2 (const double &s) const
- double differential_h_3 (const double &s) const
- double differential_h_4 (const double &s) const
- double differential_h_5 (const double &s) const
- double differential_j_1s (const double &s) const
- double differential_j_1c (const double &s) const
- double differential_j_2s (const double &s) const
- double differential_j_2c (const double &s) const
- double differential_j_3 (const double &s) const
- double differential_j_3_normalized (const double &s) const
- double differential_j_3_normalized_cp_averaged (const double &s) const
- double differential_j_4 (const double &s) const
- double differential_j_5 (const double &s) const
- double differential_j_6s (const double &s) const
- double differential_j_6c (const double &s) const

- double differential_j_7 (const double &s) const
- double differential_j_8 (const double &s) const
- double differential_j_9 (const double &s) const
- double differential_j_9_normalized (const double &s) const
- double differential_j_9_normalized_cp_averaged (const double &s) const
- double integrated_decay_width (const double &s_min, const double &s_max) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- double integrated_branching_ratio_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_cp_asymmetry (const double &s_min, const double &s_max) const
- double integrated_isospin_asymmetry (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_longitudinal_polarisation (const double &s_min, const double &s_max) const
- double integrated_longitudinal_polarisation_cp_averaged (const double &s_min, const double &s_-max) const
- double integrated_transversal_polarisation (const double &s_min, const double &s_max) const
- double integrated_transversal_polarisation_cp_averaged (const double &s_min, const double &s_-max) const
- double integrated_transverse_asymmetry_2 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_2_cp_averaged (const double &s_min, const double &s_-max) const
- double integrated_transverse_asymmetry_3 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_4 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_5 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_re (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_im (const double &s_min, const double &s_max) const
- double integrated_p_prime_4 (const double &s_min, const double &s_max) const
- double integrated_p_prime_5 (const double &s_min, const double &s_max) const
- double integrated_p_prime_6 (const double &s_min, const double &s_max) const
- double integrated_h_1 (const double &s_min, const double &s_max) const
- double integrated_h_2 (const double &s_min, const double &s_max) const
- double integrated_h_3 (const double &s_min, const double &s_max) const
- double integrated_h_4 (const double &s_min, const double &s_max) const
- double integrated_h_5 (const double &s_min, const double &s_max) const
- double integrated_j_1s (const double &s_min, const double &s_max) const
- double integrated_j_1c (const double &s_min, const double &s_max) const
- double integrated_j_2s (const double &s_min, const double &s_max) const
- double integrated_j_2c (const double &s_min, const double &s_max) const
- double integrated_j_3 (const double &s_min, const double &s_max) const
- double integrated_j_3_normalized (const double &s_min, const double &s_max) const
- double integrated_j_3_normalized_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_j_4 (const double &s_min, const double &s_max) const
- double integrated_j_5 (const double &s_min, const double &s_max) const
- double integrated_j_6s (const double &s_min, const double &s_max) const
- double integrated_j_6c (const double &s_min, const double &s_max) const
- double integrated_j_7 (const double &s_min, const double &s_max) const
- double integrated_j_8 (const double &s_min, const double &s_max) const
- double integrated_j_9 (const double &s_min, const double &s_max) const
- double integrated_j_9_normalized (const double &s_min, const double &s_max) const
- double integrated_j_9_normalized_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_a_9 (const double &s_min, const double &s_max) const

template<> class eos::BToKstarDilepton< LargeRecoil >

6.26.1 Constructor & Destructor Documentation

6.26.1.1 eos::BToKstarDilepton< LargeRecoil >::BToKstarDilepton (const Parameters & *parameters*, const Options & *options*)

6.26.1.2 eos::BToKstarDilepton< LargeRecoil >::~BToKstarDilepton ()

6.26.2 Member Function Documentation

6.26.2.1 double eos::BToKstarDilepton< LargeRecoil >::a_fb_zero_crossing () const

6.26.2.2 complex<double> eos::BToKstarDilepton< LargeRecoil >::a_long (const Helicity & *h*, const double & *s*) const

6.26.2.3 complex<double> eos::BToKstarDilepton< LargeRecoil >::a_par (const Helicity & *h*, const double & *s*) const

6.26.2.4 complex<double> eos::BToKstarDilepton< LargeRecoil >::a_perp (const Helicity & *h*, const double & *s*) const

6.26.2.5 double eos::BToKstarDilepton< LargeRecoil >::differential_branching_ratio (const double & *s*) const

6.26.2.6 double eos::BToKstarDilepton< LargeRecoil >::differential_decay_width (const double & *s*) const

6.26.2.7 double eos::BToKstarDilepton< LargeRecoil >::differential_forward_backward_-asymmetry (const double & *s*) const

6.26.2.8 double eos::BToKstarDilepton< LargeRecoil >::differential_h_1 (const double & *s*) const

6.26.2.9 double eos::BToKstarDilepton< LargeRecoil >::differential_h_2 (const double & *s*) const

6.26.2.10 double eos::BToKstarDilepton< LargeRecoil >::differential_h_3 (const double & *s*) const

6.26.2.11 double eos::BToKstarDilepton< LargeRecoil >::differential_h_4 (const double & *s*) const

6.26.2.12 double eos::BToKstarDilepton< LargeRecoil >::differential_h_5 (const double & *s*) const

6.26.2.13 double eos::BToKstarDilepton< LargeRecoil >::differential_isospin_asymmetry (const double & *s*) const

6.26.2.14 double eos::BToKstarDilepton< LargeRecoil >::differential_j_1c (const double & *s*) const

6.26.2.15 double eos::BToKstarDilepton< LargeRecoil >::differential_j_1s (const double & *s*) const

6.26.2.16 double eos::BToKstarDilepton< LargeRecoil >::differential_j_2c (const double & *s*) const

6.26.2.17 double eos::BToKstarDilepton< LargeRecoil >::differential_j_2s (const double & *s*) const

6.26.2.18 double eos::BToKstarDilepton< LargeRecoil >::differential_j_3 (const double & *s*) const

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-large-recoil.hh

6.27 eos::BToKstarDilepton< LowRecoil > Class Template Reference

`#include <exclusive-b-to-s-dilepton-low-recoil.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToKstarDilepton< LowRecoil > >.

Public Member Functions

- BToKstarDilepton (const Parameters &parameters, const Options &options)
- ~BToKstarDilepton ()
- complex< double > a_long (const Helicity &h, const double &s) const
- complex< double > a_perp (const Helicity &h, const double &s) const
- complex< double > a_par (const Helicity &h, const double &s) const
- double real_y (const double &s) const
- double imag_y (const double &s) const
- double real_c9eff (const double &s) const
- double imag_c9eff (const double &s) const
- double real_c7eff (const double &s) const
- double imag_c7eff (const double &s) const
- double rho_1 (const double &s) const
- double rho_2 (const double &s) const
- double four_differential_decay_width (const double &s, const double &c_theta_l, const double &c_theta_k, const double &phi) const
- double differential_branching_ratio (const double &s) const
- double differential_decay_width (const double &s) const
- double differential_forward_backward_asymmetry (const double &s) const
- double differential_longitudinal_polarisation (const double &s) const
- double differential_transversal_polarisation (const double &s) const
- double differential_transverse_asymmetry_2 (const double &s) const
- double differential_transverse_asymmetry_3 (const double &s) const
- double differential_transverse_asymmetry_4 (const double &s) const
- double differential_transverse_asymmetry_5 (const double &s) const
- double differential_transverse_asymmetry_re (const double &s) const
- double differential_transverse_asymmetry_im (const double &s) const
- double differential_p_prime_4 (const double &s) const
- double differential_p_prime_5 (const double &s) const
- double differential_p_prime_6 (const double &s) const
- double differential_h_1 (const double &s) const
- double differential_h_2 (const double &s) const
- double differential_h_3 (const double &s) const
- double differential_h_4 (const double &s) const
- double differential_h_5 (const double &s) const
- double differential_cp_asymmetry_1 (const double &s) const
- double differential_cp_asymmetry_2 (const double &s) const
- double differential_cp_asymmetry_3 (const double &s) const
- double differential_cp_asymmetry_mix (const double &s) const
- double differential_j_1s (const double &s) const
- double differential_j_1c (const double &s) const
- double differential_j_2s (const double &s) const

- double differential_j_2c (const double &s) const
- double differential_j_3 (const double &s) const
- double differential_j_3_normalized (const double &s) const
- double differential_j_3_normalized_cp_averaged (const double &s) const
- double differential_j_4 (const double &s) const
- double differential_j_5 (const double &s) const
- double differential_j_6s (const double &s) const
- double differential_j_6c (const double &s) const
- double differential_j_7 (const double &s) const
- double differential_j_8 (const double &s) const
- double differential_j_9 (const double &s) const
- double differential_j_9_normalized (const double &s) const
- double differential_j_9_normalized_cp_averaged (const double &s) const
- double integrated_decay_width (const double &s_min, const double &s_max) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- double integrated_branching_ratio_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry_naive (const double &s_min, const double &s_-max) const
- double integrated_forward_backward_asymmetry (const double &s_min, const double &s_max) const
- double integrated_unnormalized_forward_backward_asymmetry (const double &s_min, const double &s_max) const
- double integrated_forward_backward_asymmetry_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_longitudinal_polarisation_naive (const double &s_min, const double &s_max) const
- double integrated_longitudinal_polarisation (const double &s_min, const double &s_max) const
- double integrated_longitudinal_polarisation_cp_averaged (const double &s_min, const double &s_-max) const
- double integrated_transversal_polarisation (const double &s_min, const double &s_max) const
- double integrated_transversal_polarisation_cp_averaged (const double &s_min, const double &s_-max) const
- double integrated_transverse_asymmetry_2 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_2_cp_averaged (const double &s_min, const double &s_-max) const
- double integrated_transverse_asymmetry_2_naive (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_3 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_3_naive (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_4 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_4_naive (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_5 (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_re (const double &s_min, const double &s_max) const
- double integrated_transverse_asymmetry_im (const double &s_min, const double &s_max) const
- double integrated_p_prime_4 (const double &s_min, const double &s_max) const
- double integrated_p_prime_5 (const double &s_min, const double &s_max) const
- double integrated_p_prime_6 (const double &s_min, const double &s_max) const
- double integrated_h_1 (const double &s_min, const double &s_max) const
- double integrated_h_1_naive (const double &s_min, const double &s_max) const

- double integrated_h_2 (const double &s_min, const double &s_max) const
- double integrated_h_2_naive (const double &s_min, const double &s_max) const
- double integrated_h_3 (const double &s_min, const double &s_max) const
- double integrated_h_3_naive (const double &s_min, const double &s_max) const
- double integrated_h_4 (const double &s_min, const double &s_max) const
- double integrated_h_5 (const double &s_min, const double &s_max) const
- double integrated_cp_asymmetry (const double &s_min, const double &s_max) const
- double integrated_cp_asymmetry_1 (const double &s_min, const double &s_max) const
- double integrated_cp_asymmetry_2 (const double &s_min, const double &s_max) const
- double integrated_cp_asymmetry_3 (const double &s_min, const double &s_max) const
- double integrated_cp_summed_decay_width (const double &s_min, const double &s_max) const
- double integrated_unnormalized_cp_asymmetry_1 (const double &s_min, const double &s_max) const
- double integrated_j_1s (const double &s_min, const double &s_max) const
- double integrated_j_1c (const double &s_min, const double &s_max) const
- double integrated_j_2s (const double &s_min, const double &s_max) const
- double integrated_j_2c (const double &s_min, const double &s_max) const
- double integrated_j_3 (const double &s_min, const double &s_max) const
- double integrated_j_3_normalized (const double &s_min, const double &s_max) const
- double integrated_j_3_normalized_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_j_4 (const double &s_min, const double &s_max) const
- double integrated_j_5 (const double &s_min, const double &s_max) const
- double integrated_j_6s (const double &s_min, const double &s_max) const
- double integrated_j_6c (const double &s_min, const double &s_max) const
- double integrated_j_7 (const double &s_min, const double &s_max) const
- double integrated_j_8 (const double &s_min, const double &s_max) const
- double integrated_j_9 (const double &s_min, const double &s_max) const
- double integrated_j_9_normalized (const double &s_min, const double &s_max) const
- double integrated_j_9_normalized_cp_averaged (const double &s_min, const double &s_max) const
- double integrated_a_9 (const double &s_min, const double &s_max) const

template<> class eos::BToKstarDilepton< LowRecoil >

6.27.1 Constructor & Destructor Documentation

6.27.1.1 eos::BToKstarDilepton< LowRecoil >::BToKstarDilepton (const Parameters & *parameters*, const Options & *options*)

6.27.1.2 eos::BToKstarDilepton< LowRecoil >::~BToKstarDilepton ()

6.27.2 Member Function Documentation

6.27.2.1 complex<double> eos::BToKstarDilepton< LowRecoil >::a_long (const Helicity & *h*, const double & *s*) const

6.27.2.2 complex<double> eos::BToKstarDilepton< LowRecoil >::a_par (const Helicity & *h*, const double & *s*) const

6.27.2.3 complex<double> eos::BToKstarDilepton< LowRecoil >::a_perp (const Helicity & *h*, const double & *s*) const

6.27.2.4 double eos::BToKstarDilepton< LowRecoil >::differential_branching_ratio (const double & *s*) const

6.27.2.5 double eos::BToKstarDilepton< LowRecoil >::differential_cp_asymmetry_1 (const double & *s*) const

6.27.2.6 double eos::BToKstarDilepton< LowRecoil >::differential_cp_asymmetry_2 (const double & *s*) const

6.27.2.7 double eos::BToKstarDilepton< LowRecoil >::differential_cp_asymmetry_3 (const double & *s*) const

6.27.2.8 double eos::BToKstarDilepton< LowRecoil >::differential_cp_asymmetry_mix (const double & *s*) const

6.27.2.9 double eos::BToKstarDilepton< LowRecoil >::differential_decay_width (const double & *s*) const

6.27.2.10 double eos::BToKstarDilepton< LowRecoil >::differential_forward_backward_-asymmetry (const double & *s*) const

6.27.2.11 double eos::BToKstarDilepton< LowRecoil >::differential_h_1 (const double & *s*) const

6.27.2.12 double eos::BToKstarDilepton< LowRecoil >::differential_h_2 (const double & *s*) const

6.27.2.13 double eos::BToKstarDilepton< LowRecoil >::differential_h_3 (const double & *s*) const

6.27.2.14 double eos::BToKstarDilepton< LowRecoil >::differential_h_4 (const double & *s*) const

6.27.2.15 double eos::BToKstarDilepton< LowRecoil >::differential_h_5 (const double & *s*) const

6.27.2.16 double eos::BToKstarDilepton< LowRecoil >::differential_j_1c (const double & *s*) const

6.27.2.17 double eos::BToKstarDilepton< LowRecoil >::differential_j_1s (const double & *s*) const

6.27.2.18 double eos::BToKstarDilepton< LowRecoil >::differential_j_2c (const double & *s*) const

6.27.2.19 double eos::BToKstarDilepton< LowRecoil >::differential_j_2s (const double & *s*) const

6.27.2.20 double eos::BToKstarDilepton< LowRecoil >::differential_j_3 (const double & *s*) const

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-low-recoil.hh

6.28 eos::BToKstarGamma Class Reference

```
#include <exclusive-b-to-s-gamma.hh>
```

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToKstarGamma >.

Public Member Functions

- BToKstarGamma (const Parameters &parameters, const Options &options)
- ~BToKstarGamma ()
- double branching_ratio () const

  Branching Ratio.

- double branching_ratio_cp_averaged () const

  Branching Ratio (CP averaged).

- double cp_asymmetry () const

  Direct CP asymmetry A_CP.

- double s_kstar_gamma () const

  Time dependent CP asymmetry S_K^∗gamma.

- double c_kstar_gamma () const

  Time dependent CP asymmetry C_K^∗gamma.

- double isospin_asymmetry () const

  Isospin asymmetry.

6.28.1 Detailed Description

Calculates observables in B->K^∗gamma decays

6.28.2 Constructor & Destructor Documentation

6.28.2.1 eos::BToKstarGamma::BToKstarGamma (const Parameters & *parameters*, const Options & *options*)

6.28.2.2 eos::BToKstarGamma::~BToKstarGamma ()

6.28.3 Member Function Documentation

6.28.3.1 double eos::BToKstarGamma::branching_ratio () const

Branching Ratio.

6.28.3.2 double eos::BToKstarGamma::branching_ratio_cp_averaged () const

Branching Ratio (CP averaged).

6.28.3.3 double eos::BToKstarGamma::c_kstar_gamma () const

Time dependent CP asymmetry C_K^∗gamma.

6.28.3.4 double eos::BToKstarGamma::cp_asymmetry () const

Direct CP asymmetry A_CP.

6.28.3.5 double eos::BToKstarGamma::isospin_asymmetry () const

Isospin asymmetry.

6.28.3.6 double eos::BToKstarGamma::s_kstar_gamma () const

Time dependent CP asymmetry S_K^∗gamma.

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-gamma.hh

6.29 eos::BToPi Struct Reference

```
#include <mesonic-impl.hh>
```

Public Types

- typedef PToP Transition

Static Public Attributes

- static constexpr const char ∗ label = "B->pi"
- static constexpr const double m_B = 5.279
- static constexpr const double m_P = 0.135
- static constexpr const double m2_Br0m = 5.279 ∗ 5.279
- static constexpr const double m2_Br1m = 5.325 ∗ 5.325
- static constexpr const double tau_p = (m_B + m_P) ∗ (m_B + m_P)
- static constexpr const double tau_m = (m_B - m_P) ∗ (m_B - m_P)

6.29.1 Member Typedef Documentation

6.29.1.1 typedef PToP eos::BToPi::Transition

6.29.2 Member Data Documentation

6.29.2.1 constexpr const char∗ eos::BToPi::label = "B->pi" [static]

6.29.2.2 constexpr const double eos::BToPi::m2_Br0m = 5.279 ∗ 5.279 [static]

6.29.2.3 constexpr const double eos::BToPi::m2_Br1m = 5.325 ∗ 5.325 [static]

6.29.2.4 constexpr const double eos::BToPi::m_B = 5.279 [static]

6.29.2.5 constexpr const double eos::BToPi::m_P = 0.135 [static]

6.29.2.6 constexpr const double eos::BToPi::tau_m = (m_B - m_P) ∗ (m_B - m_P) [static]

6.29.2.7 constexpr const double eos::BToPi::tau_p = (m_B + m_P) ∗ (m_B + m_P) [static]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.30 eos::BToPiLeptonNeutrino Class Reference

`#include <b-to-pi-l-nu.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToPiLeptonNeutrino >.

Public Member Functions

- BToPiLeptonNeutrino (const Parameters &parameters, const Options &options)
- ~BToPiLeptonNeutrino ()
- double differential_branching_ratio (const double &s) const
- double differential_zeta (const double &s) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- double integrated_zeta (const double &s_min, const double &s_max) const

6.30.1 Constructor & Destructor Documentation

6.30.1.1 eos::BToPiLeptonNeutrino::BToPiLeptonNeutrino (const Parameters & *parameters*, const Options & *options*)

6.30.1.2 eos::BToPiLeptonNeutrino::~BToPiLeptonNeutrino ()

6.30.2 Member Function Documentation

6.30.2.1 double eos::BToPiLeptonNeutrino::differential_branching_ratio (const double & *s*) const

6.30.2.2 double eos::BToPiLeptonNeutrino::differential_zeta (const double & *s*) const

6.30.2.3 double eos::BToPiLeptonNeutrino::integrated_branching_ratio (const double & *s_min*, const double & *s_max*) const

6.30.2.4 double eos::BToPiLeptonNeutrino::integrated_zeta (const double & *s_min*, const double & *s_max*) const

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

- /home/alexshires/EOS/eos-next/eos/b-decays/b-to-pi-l-nu.hh

6.31 eos::BToS Struct Reference

```
#include <wilson_coefficients.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/utils/wilson_coefficients.hh

6.32 eos::BToXsDilepton< HLMW2005 > Class Template Reference

`#include <inclusive-b-to-s-dilepton.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToXsDilepton< HLMW2005 > >.

Public Member Functions

- BToXsDilepton (const Parameters &parameters, const Options &options)
- ~BToXsDilepton ()
- double differential_branching_ratio (const double &s) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- Diagnostics diagnostics () const

template<> class eos::BToXsDilepton< HLMW2005 >

6.32.1 Constructor & Destructor Documentation

6.32.1.1 eos::BToXsDilepton< HLMW2005 >::BToXsDilepton (const Parameters & *parameters*, const Options & *options*)

6.32.1.2 eos::BToXsDilepton< HLMW2005 >::~BToXsDilepton ()

6.32.2 Member Function Documentation

6.32.2.1 Diagnostics eos::BToXsDilepton< HLMW2005 >::diagnostics () const

6.32.2.2 double eos::BToXsDilepton< HLMW2005 >::differential_branching_ratio (const double & *s*) const

6.32.2.3 double eos::BToXsDilepton< HLMW2005 >::integrated_branching_ratio (const double & *s_min*, const double & *s_max*) const

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-dilepton.hh

6.33 eos::BToXsGamma< Minimal > Class Template Reference

`#include <inclusive-b-to-s-gamma.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToXsGamma< Minimal > >.

Public Member Functions

- BToXsGamma (const Parameters &parameters, const Options &options)
- ~BToXsGamma ()
- double integrated_branching_ratio () const

template<> class eos::BToXsGamma< Minimal >

6.33.1 Constructor & Destructor Documentation

6.33.1.1 eos::BToXsGamma< Minimal >::BToXsGamma (const Parameters & *parameters*, const Options & *options*)

6.33.1.2 eos::BToXsGamma< Minimal >::~BToXsGamma ()

6.33.2 Member Function Documentation

6.33.2.1 double eos::BToXsGamma< Minimal >::integrated_branching_ratio () const

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh

6.34 eos::BToXsGamma< NLO > Class Template Reference

`#include <inclusive-b-to-s-gamma.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< BToXsGamma< NLO > >.

Public Member Functions

- BToXsGamma (const Parameters &parameters, const Options &options)
- ~BToXsGamma ()
- double integrated_branching_ratio (const double &emin) const
- double photon_energy_moment_1 (const double &emin) const
- double photon_energy_moment_2 (const double &emin) const
- Diagnostics diagnostics () const

template<> class eos::BToXsGamma< NLO >

6.34.1 Constructor & Destructor Documentation

6.34.1.1 eos::BToXsGamma< NLO >::BToXsGamma (const Parameters & *parameters*, const Options & *options*)

6.34.1.2 eos::BToXsGamma< NLO >::~BToXsGamma ()

6.34.2 Member Function Documentation

6.34.2.1 Diagnostics eos::BToXsGamma< NLO >::diagnostics () const

6.34.2.2 double eos::BToXsGamma< NLO >::integrated_branching_ratio (const double & *emin*) const

6.34.2.3 double eos::BToXsGamma< NLO >::photon_energy_moment_1 (const double & *emin*) const

6.34.2.4 double eos::BToXsGamma< NLO >::photon_energy_moment_2 (const double & *emin*) const

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh

6.35 eos::BZ2004FormFactors< Process_, PToP > Class Template Reference

`#include <mesonic-impl.hh>`

Inherits eos::FormFactors< PToP >.

Public Member Functions

- BZ2004FormFactors (const Parameters &p, const Options &)
- virtual double f_p (const double &s) const
- virtual double f_0 (const double &s) const
- virtual double f_t (const double &s) const

Static Public Member Functions

- static FormFactors< PToP > * make (const Parameters &parameters, unsigned)

template<typename Process_> class eos::BZ2004FormFactors< Process_, PToP >

6.35.1 Constructor & Destructor Documentation

6.35.1.1 template<typename Process_ > eos::BZ2004FormFactors< Process_, PToP >::BZ2004FormFactors (const Parameters & *p*, const Options &) `[inline]`

6.35.2 Member Function Documentation

6.35.2.1 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToP >::f_0 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.35.2.2 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToP >::f_p (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.35.2.3 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToP >::f_t (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.35.2.4 template<typename Process_ > static FormFactors<PToP>* eos::BZ2004FormFactors< Process_, PToP >::make (const Parameters & *parameters*, unsigned) `[inline, static]`

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.36 eos::BZ2004FormFactors< Process_, PToV > Class Template Reference

`#include <mesonic-impl.hh>`

Inherits eos::FormFactors< PToV >.

Public Member Functions

- BZ2004FormFactors (const Parameters &p, const Options &)
- virtual double v (const double &s) const
- virtual double a_0 (const double &s) const
- virtual double a_1 (const double &s) const
- virtual double a_2 (const double &s) const
- virtual double a_12 (const double &s) const

Static Public Member Functions

- static FormFactors< PToV > * make (const Parameters &parameters, unsigned)

template<typename Process_> class eos::BZ2004FormFactors< Process_, PToV >

6.36.1 Constructor & Destructor Documentation

6.36.1.1 template<typename Process_ > eos::BZ2004FormFactors< Process_, PToV >::BZ2004FormFactors (const Parameters & *p*, const Options &) `[inline]`

6.36.2 Member Function Documentation

6.36.2.1 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToV >::a_0 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.36.2.2 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToV >::a_1 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.36.2.3 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToV >::a_12 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.36.2.4 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToV >::a_2 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.36.2.5 template<typename Process_ > static FormFactors<PToV>* eos::BZ2004FormFactors< Process_, PToV >::make (const Parameters & *parameters*, unsigned) [inline, static]

6.36.2.6 template<typename Process_ > virtual double eos::BZ2004FormFactors< Process_, PToV >::v (const double & *s*) const [inline, virtual]

Implements eos::FormFactors< PToV >.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.37 eos::BZ2004FormFactorsSplit< Process_ > Class Template Reference

`#include <mesonic-impl.hh>`

Inherits eos::FormFactors< PToP >.

Public Member Functions

- BZ2004FormFactorsSplit (const Parameters &p, const Options &)
- virtual double f_p (const double &s) const
- virtual double f_0 (const double &s) const
- virtual double f_t (const double &s) const

Static Public Member Functions

- static FormFactors< PToP > * make (const Parameters &parameters, unsigned)

template<typename Process_> class eos::BZ2004FormFactorsSplit< Process_ >

6.37.1 Constructor & Destructor Documentation

6.37.1.1 template<typename Process_ > eos::BZ2004FormFactorsSplit< Process_ >::BZ2004FormFactorsSplit (const Parameters & *p*, const Options &) `[inline]`

6.37.2 Member Function Documentation

6.37.2.1 template<typename Process_ > virtual double eos::BZ2004FormFactorsSplit< Process_ >::f_0 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.37.2.2 template<typename Process_ > virtual double eos::BZ2004FormFactorsSplit< Process_ >::f_p (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.37.2.3 template<typename Process_ > virtual double eos::BZ2004FormFactorsSplit< Process_ >::f_t (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.37.2.4 template<typename Process_ > static FormFactors<PToP>* eos::BZ2004FormFactorsSplit< Process_ >::make (const Parameters & *parameters*, unsigned) `[inline, static]`

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.38 eos::CartesianProduct< T_ > Class Template Reference

CartesianProduct is a template class to represent a Cartesian Product of an arbitrary number of T_. T_ should be a container providing a random access iterator.

```
#include <cartesian-product.hh>
```

Classes

- class **_Iterator**

Public Types

- typedef _Iterator< T_ > Iterator

  This is a random access iterator for CartesianPoduct.

Public Member Functions

- CartesianProduct ()
- void over (const T_ &new_container)
- Iterator begin () const
- Iterator end () const
- size_t size () const

6.38.1 Detailed Description

template<typename T_> class eos::CartesianProduct< T_ >

CartesianProduct is a template class to represent a Cartesian Product of an arbitrary number of T_. T_ should be a container providing a random access iterator.

6.38.2 Member Typedef Documentation

6.38.2.1 template<typename T_ > typedef _Iterator<T_> eos::CartesianProduct< T_ >::Iterator

This is a random access iterator for CartesianPoduct.

6.38.3 Constructor & Destructor Documentation

6.38.3.1 template<typename T_ > eos::CartesianProduct< T_ >::CartesianProduct () [inline]

Initializes an empty CartesianProduct.

6.38.4 Member Function Documentation

6.38.4.1 template<typename T_ > Iterator eos::CartesianProduct< T_ >::begin () const [inline]

Returns an iterator for the first element.

6.38.4.2 template<typename T_ > Iterator eos::CartesianProduct< T_ >::end () const [inline]

Returns an iterator for the position after the last element.

6.38.4.3 template<typename T_ > void eos::CartesianProduct< T_ >::over (const T_ & *new_container*) [inline]

Adds another container of type T_ to the CartesianProduct. So in some sense a multiplication with new_-container.

Parameters:

new_container The new container to be added.

6.38.4.4 template<typename T_ > size_t eos::CartesianProduct< T_ >::size () const [inline]

Returns the number of elements stored in CartesianProduct. This is the product of the sizes of each stored container.

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

- /home/alexshires/EOS/eos-next/eos/utils/cartesian-product.hh

6.39 eos::CharmLoops Struct Reference

```
#include <charm-loops.hh>
```

Static Public Member Functions

- static complex< double > h (const double &mu, const double &s)
- static complex< double > h (const double &mu, const double &s, const double &m_q)
- static complex< double > A (const double &mu, const double &s, const double &m_b)
- static complex< double > B (const double &mu, const double &s, const double &m_b)
- static complex< double > C (const double &mu, const double &s)
- static complex< double > F17_massless (const double &mu, const double &s, const double &m_b)
- static complex< double > F19_massless (const double &mu, const double &s, const double &m_b)
- static complex< double > F27_massless (const double &mu, const double &s, const double &m_b)
- static complex< double > F29_massless (const double &mu, const double &s, const double &m_b)
- static complex< double > F87_massless (const double &mu, const double &s, const double &m_b)
- static complex< double > F89_massless (const double &s, const double &m_b)
- static complex< double > F17_massive (const double &mu, const double &s, const double &m_b, const double &m_c)
- static complex< double > F19_massive (const double &mu, const double &s, const double &m_b, const double &m_c)
- static complex< double > F27_massive (const double &mu, const double &s, const double &m_b, const double &m_c)
- static complex< double > F29_massive (const double &mu, const double &s, const double &m_b, const double &m_c)
- static complex< double > B0 (const double &s, const double &m_q)
- static complex< double > C0 (const double &s, const double &m_q)

6.39.1 Member Function Documentation

6.39.1.1 static complex<double> eos::CharmLoops::A (const double & *mu*, const double & *s*, const double & *m_b*) `[static]`

6.39.1.2 static complex<double> eos::CharmLoops::B (const double & *mu*, const double & *s*, const double & *m_b*) `[static]`

6.39.1.3 static complex<double> eos::CharmLoops::B0 (const double & *s*, const double & *m_q*) `[static]`

6.39.1.4 static complex<double> eos::CharmLoops::C (const double & *mu*, const double & *s*) `[static]`

6.39.1.5 static complex<double> eos::CharmLoops::C0 (const double & *s*, const double & *m_q*) `[static]`

6.39.1.6 static complex<double> eos::CharmLoops::F17_massive (const double & *mu*, const double & *s*, const double & *m_b*, const double & *m_c*) `[static]`

6.39.1.7 static complex<double> eos::CharmLoops::F17_massless (const double & *mu*, const double & *s*, const double & *m_b*) `[static]`

6.39.1.8 static complex<double> eos::CharmLoops::F19_massive (const double & *mu*, const double & *s*, const double & *m_b*, const double & *m_c*) `[static]`

6.39.1.9 static complex<double> eos::CharmLoops::F19_massless (const double & *mu*, const double & *s*, const double & *m_b*) `[static]`

6.39.1.10 static complex<double> eos::CharmLoops::F27_massive (const double & *mu*, const double & *s*, const double & *m_b*, const double & *m_c*) `[static]`

6.39.1.11 static complex<double> eos::CharmLoops::F27_massless (const double & *mu*, const double & *s*, const double & *m_b*) `[static]`

6.39.1.12 static complex<double> eos::CharmLoops::F29_massive (const double & *mu*, const double & *s*, const double & *m_b*, const double & *m_c*) `[static]`

6.39.1.13 static complex<double> eos::CharmLoops::F29_massless (const double & *mu*, const double & *s*, const double & *m_b*) `[static]`

6.39.1.14 static complex<double> eos::CharmLoops::F87_massless (const double & *mu*, const double & *s*, const double & *m_b*) `[static]`

6.39.1.15 static complex<double> eos::CharmLoops::F89_massless (const double & *s*, const double & *m_b*) `[static]`

6.39.1.16 static complex<double> eos::CharmLoops::h (const double & *mu*, const double & *s*, const double & *m_q*) `[static]`

6.39.1.17 static complex<double> eos::CharmLoops::h (const double & *mu*, const double & *s*) `[static]`

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/charm-loops.hh

6.40 eos::ChiSquared Struct Reference

```
#include <chi-squared.hh>
```

Public Types

- typedef std::function< double(double, double, double, double, double, double)> Function

Static Public Member Functions

- static double with_theory_offset (const double &theory_min, const double &theory_central, const double &theory_max, const double &experiment_min, const double &experiment_central, const double &experiment_max)
- static double with_combined_uncertainties (const double &theory_min, const double &theory_central, const double &theory_max, const double &experiment_min, const double &experiment_central, const double &experiment_max)
- static Function make (const std::string &name)

6.40.1 Member Typedef Documentation

6.40.1.1 typedef std::function<double (double, double, double, double, double, double)> eos::ChiSquared::Function

6.40.2 Member Function Documentation

6.40.2.1 static Function eos::ChiSquared::make (const std::string & *name*) [static]

6.40.2.2 static double eos::ChiSquared::with_combined_uncertainties (const double & *theory_min*, const double & *theory_central*, const double & *theory_max*, const double & *experiment_min*, const double & *experiment_central*, const double & *experiment_max*) [static]

6.40.2.3 static double eos::ChiSquared::with_theory_offset (const double & *theory_min*, const double & *theory_central*, const double & *theory_max*, const double & *experiment_min*, const double & *experiment_central*, const double & *experiment_max*) [static]

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

- /home/alexshires/EOS/eos-next/eos/utils/chi-squared.hh

6.41 eos::CKMScanComponent Class Reference

`#include <ckm_scan_model.hh>`

Inherits eos::ModelComponent< components::CKM >.

Inherited by eos::CKMScanModel.

Public Member Functions

- CKMScanComponent (const Parameters &, const Options &, ParameterUser &)
- virtual complex< double > ckm_cd () const
- virtual complex< double > ckm_cs () const
- virtual complex< double > ckm_cb () const
- virtual complex< double > ckm_ud () const
- virtual complex< double > ckm_us () const
- virtual complex< double > ckm_ub () const
- virtual complex< double > ckm_td () const
- virtual complex< double > ckm_ts () const
- virtual complex< double > ckm_tb () const

6.41.1 Constructor & Destructor Documentation

6.41.1.1 eos::CKMScanComponent::CKMScanComponent (const Parameters &, const Options &, ParameterUser &)

6.41.2 Member Function Documentation

6.41.2.1 virtual complex<double> eos::CKMScanComponent::ckm_cb () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.2 virtual complex<double> eos::CKMScanComponent::ckm_cd () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.3 virtual complex<double> eos::CKMScanComponent::ckm_cs () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.4 virtual complex<double> eos::CKMScanComponent::ckm_tb () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.5 virtual complex<double> eos::CKMScanComponent::ckm_td () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.6 virtual complex<double> eos::CKMScanComponent::ckm_ts () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.7 virtual complex<double> eos::CKMScanComponent::ckm_ub () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.8 virtual complex<double> eos::CKMScanComponent::ckm_ud () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.41.2.9 virtual complex<double> eos::CKMScanComponent::ckm_us () const [virtual]

Implements eos::ModelComponent< components::CKM >.

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

- /home/alexshires/EOS/eos-next/eos/utils/ckm_scan_model.hh

6.42 eos::CKMScanModel Class Reference

`#include <ckm_scan_model.hh>`

Inherits eos::Model, eos::CKMScanComponent, eos::SMComponent< components::QCD >, and eos::SMComponent< components::DeltaB1 >.

Public Member Functions

- CKMScanModel (const Parameters &, const Options &)
- virtual ~CKMScanModel ()

Static Public Member Functions

- static std::shared_ptr< Model > make (const Parameters &, const Options &)

6.42.1 Constructor & Destructor Documentation

6.42.1.1 eos::CKMScanModel::CKMScanModel (const Parameters &, const Options &)

6.42.1.2 virtual eos::CKMScanModel::~CKMScanModel () `[virtual]`

6.42.2 Member Function Documentation

6.42.2.1 static std::shared_ptr<Model> eos::CKMScanModel::make (const Parameters &, const Options &) `[static]`

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

- /home/alexshires/EOS/eos-next/eos/utils/ckm_scan_model.hh

6.43 eos::Cluster Class Reference

`#include <cluster.hh>`

Inherits PrivateImplementationPattern< Cluster >.

Public Types

- typedef std::function< double(const std::vector< double > &, const std::vector< double > &, const unsigned &)> RValueFunction
- typedef WrappedForwardIterator< IteratorTag, HistoryPtr > Iterator
- typedef WrappedForwardIterator< IteratorTag, unsigned > IndexIterator

Public Member Functions

- Cluster (const RValueFunction &, const double &max_rvalue, const HistoryPtr &initial_chain, const unsigned &index, const double &skip_initial)
- ~Cluster ()
- void add (const HistoryPtr &chain, const unsigned &index)
- std::vector< double > mean () const
- const std::vector< std::vector< double > > & means () const
- const std::vector< std::vector< double > > & variances () const
- bool overlaps (const HistoryPtr &chain) const
- void parameter_indices (const std::vector< unsigned > &indices)
- Iterator begin () const
- Iterator end () const
- IndexIterator begin_indices () const
- IndexIterator end_indices () const

6.43.1 Member Typedef Documentation

6.43.1.1 typedef WrappedForwardIterator<IteratorTag, unsigned> eos::Cluster::IndexIterator

6.43.1.2 typedef WrappedForwardIterator<IteratorTag, HistoryPtr> eos::Cluster::Iterator

6.43.1.3 typedef std::function<double (const std::vector<double> &, const std::vector<double> &, const unsigned &)> eos::Cluster::RValueFunction

6.43.2 Constructor & Destructor Documentation

6.43.2.1 eos::Cluster::Cluster (const RValueFunction &, const double & *max_rvalue*, const HistoryPtr & *initial_chain*, const unsigned & *index*, const double & *skip_initial*)

Construct a cluster, which checks if two chains overlap by using the R-value criterion. With only one chain, it naturally overlaps with itself.

Parameters:

rvalue_function Strict or relaxed version

max_rvalue

initial_chain

index

skip_initial Should be in [0,1]. The first portion of the chain's history is ignored when computing the R-value.

6.43.2.2 eos::Cluster::~Cluster ()

6.43.3 Member Function Documentation

6.43.3.1 void eos::Cluster::add (const HistoryPtr & *chain*, const unsigned & *index*)

Add a chain (its history) to a cluster. Record its index.

6.43.3.2 Iterator eos::Cluster::begin () const

6.43.3.3 IndexIterator eos::Cluster::begin_indices () const

6.43.3.4 Iterator eos::Cluster::end () const

6.43.3.5 IndexIterator eos::Cluster::end_indices () const

6.43.3.6 std::vector<double> eos::Cluster::mean () const

Calculate the cluster mean from the means of individual chains.

Note:

Assumes that all chains have equal length.

6.43.3.7 const std::vector<std::vector<double> >& eos::Cluster::means () const

6.43.3.8 bool eos::Cluster::overlaps (const HistoryPtr & *chain*) const

Check if chain overlaps with this cluster. Internally, use the history of the chain, but to allow for burn-in, the first iterations can be skipped when computing means and variances.

Parameters:

chain

skip_initial A percentage, should be in range [0,1]

Returns:

6.43.3.9 void eos::Cluster::parameter_indices (const std::vector< unsigned > & *indices*)

Set indices of parameters whose R-value ought to be checked. By default, all dimensions are checked.

6.43.3.10 const std::vector<std::vector<double> >& eos::Cluster::variances () const

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

- /home/alexshires/EOS/eos-next/eos/utils/cluster.hh

6.44 eos::HierarchicalClustering::Component Class Reference

```
#include <hierarchical-clustering.hh>
```

Inherits PrivateImplementationPattern< HierarchicalClustering::Component >.

Public Member Functions

- Component (const std::vector< double > &mean, const std::vector< double > &covariance, const double &weight)
- Component (const gsl_vector *mean, const gsl_matrix *covariance, const double &weight)
- ~Component ()
- gsl_matrix * covariance () const
- const gsl_matrix * inverse_covariance () const
- const double & determinant () const
- gsl_vector * mean () const
- double & weight () const

6.44.1 Detailed Description

Describe a component of a Gaussian mixture density, characterized by mean, covariance and weight.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 eos::HierarchicalClustering::Component::Component (const std::vector< double > & *mean*, const std::vector< double > & *covariance*, const double & *weight*)

6.44.2.2 eos::HierarchicalClustering::Component::Component (const gsl_vector * *mean*, const gsl_matrix * *covariance*, const double & *weight*)

6.44.2.3 eos::HierarchicalClustering::Component::~Component ()

6.44.3 Member Function Documentation

6.44.3.1 gsl_matrix* eos::HierarchicalClustering::Component::covariance () const

6.44.3.2 const double& eos::HierarchicalClustering::Component::determinant () const

6.44.3.3 const gsl_matrix* eos::HierarchicalClustering::Component::inverse_covariance () const

6.44.3.4 gsl_vector* eos::HierarchicalClustering::Component::mean () const

6.44.3.5 double& eos::HierarchicalClustering::Component::weight () const

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

- /home/alexshires/EOS/eos-next/eos/utils/hierarchical-clustering.hh

6.45 eos::hdf5::Composite< T_ > Class Template Reference

```
#include <hdf5.hh>
```

Inherits eos::hdf5::Type.

Public Types

- typedef std::tuple< typename T_::Type...> Type

Public Member Functions

- Composite (const std::string &name, const T_ &...t)
- ~Composite ()
- virtual hid_t type_id () const
- virtual const std::string & name () const
- virtual hsize_t size () const
- virtual void copy_to_hdf5 (const void *src, void *dest) const
- virtual void copy_from_hdf5 (const void *src, void *dest) const

template<typename... T_> class eos::hdf5::Composite< T_ >

6.45.1 Member Typedef Documentation

6.45.1.1 template<typename... T_> typedef std::tuple<typename T_::Type ...> eos::hdf5::Composite< T_ >::Type

6.45.2 Constructor & Destructor Documentation

6.45.2.1 template<typename... T_> eos::hdf5::Composite< T_ >::Composite (const std::string & *name*, const T_ &... *t*) `[inline]`

6.45.2.2 template<typename... T_> eos::hdf5::Composite< T_ >::~Composite () `[inline]`

6.45.3 Member Function Documentation

6.45.3.1 template<typename... T_> virtual void eos::hdf5::Composite< T_ >::copy_from_hdf5 (const void * *src*, void * *dest*) const `[inline, virtual]`

Implements eos::hdf5::Type.

6.45.3.2 template<typename... T_> virtual void eos::hdf5::Composite< T_ >::copy_to_hdf5 (const void * *src*, void * *dest*) const `[inline, virtual]`

Implements eos::hdf5::Type.

6.45.3.3 template<typename... T_> virtual const std::string& eos::hdf5::Composite< T_ >::name () const `[inline, virtual]`

Implements eos::hdf5::Type.

6.45.3.4 template<typename... T_> virtual hsize_t eos::hdf5::Composite< T_ >::size () const [inline, virtual]

Implements eos::hdf5::Type.

6.45.3.5 template<typename... T_> virtual hid_t eos::hdf5::Composite< T_ >::type_id () const [inline, virtual]

Implements eos::hdf5::Type.

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.46 eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static std::ptrdiff_t copy_to_hdf5 (const TypeTuple_ &types, const Tuple_ &src, char *dest)
- static std::ptrdiff_t copy_from_hdf5 (const TypeTuple_ &types, const char *src, Tuple_ &dest)

template<size_t index_, typename TypeTuple_, typename Tuple_> struct eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ >

6.46.1 Member Function Documentation

6.46.1.1 template<size_t index_, typename TypeTuple_ , typename Tuple_ > static std::ptrdiff_t eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ >::copy_from_hdf5 (const TypeTuple_ & *types*, const char * *src*, Tuple_ & *dest*) `[inline, static]`

6.46.1.2 template<size_t index_, typename TypeTuple_ , typename Tuple_ > static std::ptrdiff_t eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ >::copy_to_hdf5 (const TypeTuple_ & *types*, const Tuple_ & *src*, char * *dest*) `[inline, static]`

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.47 eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static std::ptrdiff_t copy_to_hdf5 (const TypeTuple_ &types, const Tuple_ &src, char *dest)
- static std::ptrdiff_t copy_from_hdf5 (const TypeTuple_ &types, const char *src, Tuple_ &dest)

template<typename TypeTuple_, typename Tuple_> struct eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ >

6.47.1 Member Function Documentation

6.47.1.1 template<typename TypeTuple_ , typename Tuple_ > static std::ptrdiff_t eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ >::copy_from_hdf5 (const TypeTuple_ & *types*, const char * *src*, Tuple_ & *dest*) `[inline, static]`

6.47.1.2 template<typename TypeTuple_ , typename Tuple_ > static std::ptrdiff_t eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ >::copy_to_hdf5 (const TypeTuple_ & *types*, const Tuple_ & *src*, char * *dest*) `[inline, static]`

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.48 eos::ConcreteObservable< Decay_, Args_ > Class Template Reference

`#include <concrete_observable.hh>`

Inherits eos::Observable.

Public Member Functions

- ConcreteObservable (const std::string &name, const Parameters &parameters, const Kinematics &kinematics, const Options &options, const std::function< double(const Decay_ *, const Args_-&...)> &function, const std::tuple< typename impl::ConvertTo< Args_, const char * >::Type...> &kinematics_names)
- virtual const std::string & name () const
- virtual double evaluate () const
- virtual Parameters parameters ()
- virtual Kinematics kinematics ()
- virtual Options options ()
- virtual ObservablePtr clone () const
- virtual ObservablePtr clone (const Parameters &parameters) const

template<typename Decay_, typename... Args_> class eos::ConcreteObservable< Decay_, Args_ >

6.48.1 Constructor & Destructor Documentation

6.48.1.1 template<typename Decay_ , typename... Args_> eos::ConcreteObservable< Decay_, Args_ >::ConcreteObservable (const std::string & *name*, const Parameters & *parameters*, const Kinematics & *kinematics*, const Options & *options*, const std::function< double(const Decay_ *, const Args_ &...)> & *function*, const std::tuple< typename impl::ConvertTo< Args_, const char * >::Type...> & *kinematics_names*) `[inline]`

6.48.2 Member Function Documentation

6.48.2.1 template<typename Decay_ , typename... Args_> virtual ObservablePtr eos::ConcreteObservable< Decay_, Args_ >::clone (const Parameters & *parameters*) const `[inline, virtual]`

Implements eos::Observable.

6.48.2.2 template<typename Decay_ , typename... Args_> virtual ObservablePtr eos::ConcreteObservable< Decay_, Args_ >::clone () const `[inline, virtual]`

Implements eos::Observable.

6.48.2.3 template<typename Decay_ , typename... Args_> virtual double eos::ConcreteObservable< Decay_, Args_ >::evaluate () const `[inline, virtual]`

Implements eos::Observable.

6.48.2.4 template<typename Decay_ , typename... Args_> virtual Kinematics eos::ConcreteObservable< Decay_, Args_ >::kinematics () [inline, virtual]

Implements eos::Observable.

6.48.2.5 template<typename Decay_ , typename... Args_> virtual const std::string& eos::ConcreteObservable< Decay_, Args_ >::name () const [inline, virtual]

Implements eos::Observable.

6.48.2.6 template<typename Decay_ , typename... Args_> virtual Options eos::ConcreteObservable< Decay_, Args_ >::options () [inline, virtual]

Implements eos::Observable.

6.48.2.7 template<typename Decay_ , typename... Args_> virtual Parameters eos::ConcreteObservable< Decay_, Args_ >::parameters () [inline, virtual]

Implements eos::Observable.

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

- /home/alexshires/EOS/eos-next/eos/utils/concrete_observable.hh

6.49 eos::ConcreteObservableFactory< Decay_, Args_ > Class Template Reference

`#include <concrete_observable.hh>`

Inherits eos::ObservableFactory.

Public Member Functions

- ConcreteObservableFactory (const std::string &name, const std::function< double(const Decay_ *, const Args_ &...)> &function, const std::tuple< typename impl::ConvertTo< Args_, const char * >::Type...> &kinematics_names)
- ~ConcreteObservableFactory ()
- virtual ObservablePtr make (const Parameters &parameters, const Kinematics &kinematics, const Options &options) const

template<typename Decay_, typename... Args_> class eos::ConcreteObservableFactory< Decay_, Args_ >

6.49.1 Constructor & Destructor Documentation

6.49.1.1 template<typename Decay_ , typename... Args_> eos::ConcreteObservableFactory< Decay_, Args_ >::ConcreteObservableFactory (const std::string & *name*, const std::function< double(const Decay_ *, const Args_ &...)> & *function*, const std::tuple< typename impl::ConvertTo< Args_, const char * >::Type...> & *kinematics_names*) `[inline]`

6.49.1.2 template<typename Decay_ , typename... Args_> eos::ConcreteObservableFactory< Decay_, Args_ >::~ConcreteObservableFactory () `[inline]`

6.49.2 Member Function Documentation

6.49.2.1 template<typename Decay_ , typename... Args_> virtual ObservablePtr eos::ConcreteObservableFactory< Decay_, Args_ >::make (const Parameters & *parameters*, const Kinematics & *kinematics*, const Options & *options*) const `[inline, virtual]`

Implements eos::ObservableFactory.

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

- /home/alexshires/EOS/eos-next/eos/utils/concrete_observable.hh

6.50 eos::ConditionVariable Class Reference

```
#include <condition_variable.hh>
```

Public Member Functions

- ConditionVariable ()
  Constructor.
- ~ConditionVariable ()
  Destructor.
- void broadcast ()
  Broadcast a wake-up to all waiting threads.
- void signal ()
  Sign a wake-up to the waiting thread.
- void acquire_then_signal (Mutex &)
  Acquire a lock for signaling.
- void wait (Mutex &)
  Wait until condition is met.

6.50.1 Constructor & Destructor Documentation

6.50.1.1 eos::ConditionVariable::ConditionVariable ()

Constructor.

6.50.1.2 eos::ConditionVariable::~ConditionVariable ()

Destructor.

6.50.2 Member Function Documentation

6.50.2.1 void eos::ConditionVariable::acquire_then_signal (Mutex &)

Acquire a lock for signaling.

6.50.2.2 void eos::ConditionVariable::broadcast ()

Broadcast a wake-up to all waiting threads.

6.50.2.3 void eos::ConditionVariable::signal ()

Sign a wake-up to the waiting thread.

6.50.2.4 void eos::ConditionVariable::wait (Mutex &)

Wait until condition is met.

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

- /home/alexshires/EOS/eos-next/eos/utils/condition_variable.hh

6.51 eos::MarkovChainSampler::Config Class Reference

`#include <markov_chain_sampler.hh>`

Static Public Member Functions

Basic Function

- static Config Default ()
- static Config Quick ()

Public Attributes

Basic options

Number of Markov chains

- VerifiedRange< unsigned > number_of_chains
- unsigned long seed
- bool parallelize

Convergence options

- VerifiedRange< double > min_efficiency
  accepted / trials should be between 0.15 and 0.50
- VerifiedRange< double > max_efficiency
  accepted / trials should be between 0.15 and 0.50
- VerifiedRange< double > rvalue_criterion_param
  rvalues close to 1 indicates multichain convergence
- VerifiedRange< double > rvalue_criterion_posterior
  accepted / trials should be between 0.15 and 0.50
- bool use_strict_rvalue_definition
- bool use_posterior_rvalue
- bool scale_automatic
- bool scale_nuisance
- double scale_reduction
  Value by which sqrt(covariance) of scan parameters is reduced for initial guesses of the proposal functions.

Prerun options

- bool find_modes
- bool need_prerun
- unsigned prerun_iterations_update
- unsigned prerun_iterations_min
- unsigned prerun_iterations_max
- std::string proposal
  Which local proposal function is chosen.
- std::vector< std::string > block_proposal_parameters

- VerifiedRange< double > student_t_degrees_of_freedom
- bool store_prerun

  Whether to store prerun samples.

- unsigned prerun_chains_per_partition
- std::vector< std::vector< std::tuple< std::string, double, double > > > partitions

Main run options

- unsigned adapt_iterations
- unsigned chunks

  Number of chunks of sampling.

- unsigned chunk_size

  Number of iterations per chunk.

- std::shared_ptr< proposal_functions::GlobalLocal::Config > global_local_config

  Summarize all options pertaining to the GlobalLocal proposal function.

- bool need_main_run

  Turn off main run, so only prerun is performed.

- VerifiedRange< double > skip_initial
- bool store

  Whether to store collected samples.

Output options

- std::string output_file
- bool store_observables_and_proposals

6.51.1 Detailed Description

Stores all configuration options for a MarkovChainSampler.

6.51.2 Member Function Documentation

6.51.2.1 static Config eos::MarkovChainSampler::Config::Default () `[static]`

Named constructor

MarkovChainSampler settings with reasonably chosen default values.

6.51.2.2 static Config eos::MarkovChainSampler::Config::Quick () `[static]`

Named constructor

MarkovChainSampler settings with values optimized for quick chain convergence and evaluation.

Note:

The convergence is not very reliable. Use with care! If in doubt, use MarkovChainSampler::Config::Default

6.51.3 Member Data Documentation

6.51.3.1 unsigned eos::MarkovChainSampler::Config::adapt_iterations

Proposal function is adapted during the first iterations. Default: 0, i.e. don't adapt

6.51.3.2 std::vector<std::string> eos::MarkovChainSampler::Config::block_proposal_-parameters

One proposal for the whole space or a decomposition into multivariate part and 1D, uncorrelated part. Map a fixed proposal in 1D to a parameter

6.51.3.3 unsigned eos::MarkovChainSampler::Config::chunk_size

Number of iterations per chunk.

6.51.3.4 unsigned eos::MarkovChainSampler::Config::chunks

Number of chunks of sampling.

6.51.3.5 bool eos::MarkovChainSampler::Config::find_modes

Find the local maxima after the prerun, using the point with the highest posterior in each chain as a starting point

6.51.3.6 std::shared_ptr<proposal_functions::GlobalLocal::Config> eos::MarkovChainSampler::Config::global_local_config

Summarize all options pertaining to the GlobalLocal proposal function.

6.51.3.7 VerifiedRange<double> eos::MarkovChainSampler::Config::max_efficiency

accepted / trials should be between 0.15 and 0.50

6.51.3.8 VerifiedRange<double> eos::MarkovChainSampler::Config::min_efficiency

accepted / trials should be between 0.15 and 0.50

6.51.3.9 bool eos::MarkovChainSampler::Config::need_main_run

Turn off main run, so only prerun is performed.

6.51.3.10 bool eos::MarkovChainSampler::Config::need_prerun

Find the local maxima after the prerun, using the point with the highest posterior in each chain as a starting point

6.51.3.11 VerifiedRange<unsigned> eos::MarkovChainSampler::Config::number_of_chains

The seed that is used to initialize the random number generator. Independant runs with identical seeds will produce

6.51.3.12 std::string eos::MarkovChainSampler::Config::output_file

The HDF5 output file to store the markov chains.

6.51.3.13 bool eos::MarkovChainSampler::Config::parallelize

If true, use as many threads as there are cores available. If false, use only one thread.

6.51.3.14 std::vector<std::vector<std::tuple<std::string, double, double> > > eos::MarkovChainSampler::Config::partitions

Find the local maxima after the prerun, using the point with the highest posterior in each chain as a starting point

6.51.3.15 unsigned eos::MarkovChainSampler::Config::prerun_chains_per_partition

In case there is only one partition, this is ignored and number_of_chains is used. Else, this many chains are used per partition.

6.51.3.16 unsigned eos::MarkovChainSampler::Config::prerun_iterations_max

Find the local maxima after the prerun, using the point with the highest posterior in each chain as a starting point

6.51.3.17 unsigned eos::MarkovChainSampler::Config::prerun_iterations_min

Find the local maxima after the prerun, using the point with the highest posterior in each chain as a starting point

6.51.3.18 unsigned eos::MarkovChainSampler::Config::prerun_iterations_update

Find the local maxima after the prerun, using the point with the highest posterior in each chain as a starting point

6.51.3.19 std::string eos::MarkovChainSampler::Config::proposal

Which local proposal function is chosen.

6.51.3.20 VerifiedRange<double> eos::MarkovChainSampler::Config::rvalue_criterion_param

rvalues close to 1 indicates multichain convergence

6.51.3.21 VerifiedRange<double> eos::MarkovChainSampler::Config::rvalue_criterion_posterior

accepted / trials should be between 0.15 and 0.50

6.51.3.22 bool eos::MarkovChainSampler::Config::scale_automatic

Rescale multivariate proposal functions' covariance depending on the dimensionality of the parameter space.

6.51.3.23 bool eos::MarkovChainSampler::Config::scale_nuisance

Decide whether only scan parameters or all parameters' initial variance should be rescaled in order to increase the efficiency.

6.51.3.24 double eos::MarkovChainSampler::Config::scale_reduction

Value by which sqrt(covariance) of scan parameters is reduced for initial guesses of the proposal functions.

6.51.3.25 unsigned long eos::MarkovChainSampler::Config::seed

The seed that is used to initialize the random number generator. Independant runs with identical seeds will produce

6.51.3.26 VerifiedRange<double> eos::MarkovChainSampler::Config::skip_initial

When computing R-values, one can skip the first (skip_initial)% of the iterations. This provides more robust results, as it discards an initial burn in where the chain may be very far off from likely regions

6.51.3.27 bool eos::MarkovChainSampler::Config::store

Whether to store collected samples.

6.51.3.28 bool eos::MarkovChainSampler::Config::store_observables_and_proposals

The HDF5 output file to store the markov chains.

6.51.3.29 bool eos::MarkovChainSampler::Config::store_prerun

Whether to store prerun samples.

6.51.3.30 VerifiedRange<double> eos::MarkovChainSampler::Config::student_t_degrees_of_freedom

The number of degrees of freedom for a local proposal function of type MultivariateStudentT

Note:

a value of one corresponds to a (multivariate) Cauchy function

6.51.3.31 bool eos::MarkovChainSampler::Config::use_posterior_rvalue

often the chains have mixed in parameter space, but not in posterior space. Thus, this R-value impedes convergence declaration.

6.51.3.32 bool eos::MarkovChainSampler::Config::use_strict_rvalue_definition

Use Gelman/Rubin definition if true, and relaxed BAT definition if false.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain_sampler.hh

6.52 eos::EquationSolver::Config Struct Reference

```
#include <equation_solver.hh>
```

Static Public Member Functions

- static Config Default ()

Public Attributes

- unsigned maximum_steps

  Maximum number of evaluations.

- VerifiedRange< unsigned > strategy

  [0 = low, 1 = medium, 2 = high] precision

- VerifiedRange< double > tolerance

  Required precision.

6.52.1 Member Function Documentation

6.52.1.1 static Config eos::EquationSolver::Config::Default () [static]

6.52.2 Member Data Documentation

6.52.2.1 unsigned eos::EquationSolver::Config::maximum_steps

Maximum number of evaluations.

6.52.2.2 VerifiedRange<unsigned> eos::EquationSolver::Config::strategy

[0 = low, 1 = medium, 2 = high] precision

6.52.2.3 VerifiedRange<double> eos::EquationSolver::Config::tolerance

Required precision.

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

- /home/alexshires/EOS/eos-next/eos/utils/equation_solver.hh

6.53 eos::PopulationMonteCarloSampler::Config Struct Reference

```
#include <population_monte_carlo_sampler.hh>
```

Static Public Member Functions

Basic Function

- static Config Default ()
- static Config Quick ()

Public Attributes

Basic options

- unsigned long seed
- bool parallelize
- unsigned number_of_workers

Proposal density options

- bool block_decomposition
- std::vector< double > component_weights
- std::vector< std::vector< double > > component_means

  n entries.

- double component_offset

  Shift component by this number of sigmas.

- std::vector< std::vector< double > > component_variances

  n^2 entries.

- VerifiedRange< unsigned > components_per_cluster
- VerifiedRange< int > degrees_of_freedom
- VerifiedRange< double > minimum_overlap

  Used during filtering of components, if too many samples fall outside of box, remove components.

- VerifiedRange< double > mode_distance
- bool override_global_local_proposal
- bool random_start
- int single_cluster
- VerifiedRange< double > skip_initial

  Skip this percentage from beginning of a chain's history.

- unsigned starting_points
- VerifiedRange< double > std_dev_reduction

Hierarchical clustering options

- VerifiedRange< double > group_by_r_value
- std::vector< unsigned > ignore_groups
- bool patch_around_local_mode
- bool r_value_no_nuisance
- unsigned sliding_window

- bool store_input_components
  Store the components created from chain patches.

- bool store_hc_initial
  Store initial guess for clustering from long patches.

- unsigned super_clusters

Pre run options

- bool adjust_sample_size
  Change number of samples according to number of live components.

- unsigned chunks
  The number of updates to proposal functions (mixture components).

- unsigned chunk_size
  Number of iterations per chunk and component.

- unsigned crop_highest_weights
- bool need_prerun
  If false, no adaptions to proposal densities are made.

- bool store_prerun
  store samples of prerun steps (statistics are always stored)

Convergence options

- VerifiedRange< double > convergence_eff_sample_size
  Declare convergence if both perplexity and effective sample size large enough.

- VerifiedRange< double > convergence_perplexity
  Declare convergence if both perplexity and effective sample size large enough.

- bool ignore_eff_sample_size
- VerifiedRange< double > minimum_eff_sample_size
  Declare converge if both perplexity and effective sample size large enough and do not rise anymore.

- VerifiedRange< double > minimum_perplexity
  Declare converge if both perplexity and effective sample size large enough and do not rise anymore.

- VerifiedRange< unsigned > minimum_steps
  Consider the last steps to decide if convergence statistics do not rise (significantly) anymore.

- VerifiedRange< double > maximum_relative_std_deviation
  Relative variance of last steps should be small to ascertain convergence.

Main run options

- long final_chunk_size
  Number of iterations used in final step after adaption is finished.

- bool store

Whether to store collected samples.

Output options

- std::string output_file
- VerifiedRange< unsigned > print_steps

6.53.1 Detailed Description

Stores all configuration options for a PopulationMonteCarloSampler.

6.53.2 Member Function Documentation

6.53.2.1 static Config eos::PopulationMonteCarloSampler::Config::Default () [static]

Named constructor

PopulationMonteCarloSampler settings with reasonably chosen default values.

6.53.2.2 static Config eos::PopulationMonteCarloSampler::Config::Quick () [static]

Named constructor

PopulationMonteCarloSampler settings with values optimized for quick chain convergence and evaluation.

Note:

The convergence is not very reliable. Use with care! If in doubt, use PopulationMonteCarloSampler::Config::Default

6.53.3 Member Data Documentation

6.53.3.1 bool eos::PopulationMonteCarloSampler::Config::adjust_sample_size

Change number of samples according to number of live components.

6.53.3.2 bool eos::PopulationMonteCarloSampler::Config::block_decomposition

Perform a decomposition into scan and nuisance parameters and use prior variance for initial proposal of nuisance. The (co)variance in scan direction is unaffected. In addition, set the component's mean to the prior's mean.

6.53.3.3 unsigned eos::PopulationMonteCarloSampler::Config::chunk_size

Number of iterations per chunk and component.

6.53.3.4 unsigned eos::PopulationMonteCarloSampler::Config::chunks

The number of updates to proposal functions (mixture components).

6.53.3.5 std::vector<std::vector<double> > eos::PopulationMonteCarloSampler::Config::component_means

n entries.

6.53.3.6 double eos::PopulationMonteCarloSampler::Config::component_offset

Shift component by this number of sigmas.

6.53.3.7 std::vector<std::vector<double> > eos::PopulationMonteCarloSampler::Config::component_variances

n^2 entries.

6.53.3.8 std::vector<double> eos::PopulationMonteCarloSampler::Config::component_weights

Each component has an associated weight, which determines how many samples it may contribute in the next sampling phase.

6.53.3.9 VerifiedRange<unsigned> eos::PopulationMonteCarloSampler::Config::components_-per_cluster

When initializing from global local, draw this many components for each cluster.

6.53.3.10 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::convergence_-eff_sample_size

Declare convergence if both perplexity and effective sample size large enough.

6.53.3.11 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::convergence_-perplexity

Declare convergence if both perplexity and effective sample size large enough.

6.53.3.12 unsigned eos::PopulationMonteCarloSampler::Config::crop_highest_weights

If > 0, find the samples with highest weight and ignore them during a PMC update. This can be useful to remove outliers which dominate over the bulk of samples and lead to many components dying out, as well as to dramatically shrinking components.

6.53.3.13 VerifiedRange<int> eos::PopulationMonteCarloSampler::Config::degrees_of_freedom

Degrees of freedom of a multivariate t-distribution. Special value of -1 corresponds to the Gaussian distribution.

6.53.3.14 long eos::PopulationMonteCarloSampler::Config::final_chunk_size

Number of iterations used in final step after adaption is finished.

6.53.3.15 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::group_by_r_-value

If true, use the Gelman-Rubin R-value to combine chains into groups. N initial guess clusters are drawn from each group, no matter how many chains are in it.

Default: 1, all chains belong to same group.

6.53.3.16 bool eos::PopulationMonteCarloSampler::Config::ignore_eff_sample_size

Can declare convergence w/o considering the effective sample size, just by looking at perplexity.

6.53.3.17 std::vector<unsigned> eos::PopulationMonteCarloSampler::Config::ignore_groups

Ignore groups (identified by index) from clustering. Default: no groups are ignored.

Note:

Use with great care. Important parts of the posterior may be ignored.

6.53.3.18 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::maximum_-relative_std_deviation

Relative variance of last steps should be small to ascertain convergence.

6.53.3.19 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::minimum_eff_-sample_size

Declare converge if both perplexity and effective sample size large enough and do not rise anymore.

6.53.3.20 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::minimum_-overlap

Used during filtering of components, if too many samples fall outside of box, remove components.

6.53.3.21 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::minimum_-perplexity

Declare converge if both perplexity and effective sample size large enough and do not rise anymore.

6.53.3.22 VerifiedRange<unsigned> eos::PopulationMonteCarloSampler::Config::minimum_-steps

Consider the last steps to decide if convergence statistics do not rise (significantly) anymore.

6.53.3.23 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::mode_distance

Perform a decomposition into scan and nuisance parameters and use prior variance for initial proposal of nuisance. The (co)variance in scan direction is unaffected. In addition, set the component's mean to the prior's mean.

6.53.3.24 bool eos::PopulationMonteCarloSampler::Config::need_prerun

If false, no adaptions to proposal densities are made.

6.53.3.25 unsigned eos::PopulationMonteCarloSampler::Config::number_of_workers

How many workers to use. If parallelize is true, this is the number of threads used. The default value is 0, implying that the proper number is drawn from the number of available cores on the machine.

6.53.3.26 std::string eos::PopulationMonteCarloSampler::Config::output_file

The HDF5 output file to store the markov chains.

6.53.3.27 bool eos::PopulationMonteCarloSampler::Config::override_global_local_proposal

If true, use same local proposal as supplied by GlobalLocal input. Else use degrees_of_freedom defined above.

6.53.3.28 bool eos::PopulationMonteCarloSampler::Config::parallelize

If true, use as many threads as there are cores available. If false, use only one thread.

6.53.3.29 bool eos::PopulationMonteCarloSampler::Config::patch_around_local_mode

When creating patch from a Markov chain history, place the patches around the point with highest posterior in the selected range of point. If false, center on the mean of all points in range.

6.53.3.30 VerifiedRange<unsigned> eos::PopulationMonteCarloSampler::Config::print_steps

The HDF5 output file to store the markov chains.

6.53.3.31 bool eos::PopulationMonteCarloSampler::Config::r_value_no_nuisance

When grouping with R-value, consider only the scan parameters.

6.53.3.32 bool eos::PopulationMonteCarloSampler::Config::random_start

Initialize the components randomly at the beginning.

6.53.3.33 unsigned long eos::PopulationMonteCarloSampler::Config::seed

The seed that is used to initialize the random number generator. Independent runs with identical seeds will produce

6.53.3.34 int eos::PopulationMonteCarloSampler::Config::single_cluster

Use only history points from a particular cluster defined by its index. Default value -1: take from all clusters

6.53.3.35 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::skip_initial

Skip this percentage from beginning of a chain's history.

6.53.3.36 unsigned eos::PopulationMonteCarloSampler::Config::sliding_window

With hierarchical clustering, use this many samples to form a patch from a single Markov chain.

6.53.3.37 unsigned eos::PopulationMonteCarloSampler::Config::starting_points

Number of starting positions. Start optimization from each one to identify local modes and covariances.

6.53.3.38 VerifiedRange<double> eos::PopulationMonteCarloSampler::Config::std_dev_-reduction

For random start, take the std. deviation as parameter range divided by std_dev_reduction.

6.53.3.39 bool eos::PopulationMonteCarloSampler::Config::store

Whether to store collected samples.

6.53.3.40 bool eos::PopulationMonteCarloSampler::Config::store_hc_initial

Store initial guess for clustering from long patches.

6.53.3.41 bool eos::PopulationMonteCarloSampler::Config::store_input_components

Store the components created from chain patches.

6.53.3.42 bool eos::PopulationMonteCarloSampler::Config::store_prerun

store samples of prerun steps (statistics are always stored)

6.53.3.43 unsigned eos::PopulationMonteCarloSampler::Config::super_clusters

Perform the hierarchical_clustering algorithm on a set of input prerun markov chains to form an initial proposal density. Create this many components for PMC.

If group_by_r_value is active, choose this many components per group of chains.

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

- /home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_sampler.hh

6.54 eos::PriorSampler::Config Struct Reference

```
#include <prior_sampler.hh>
```

Static Public Member Functions

- static Config Default ()

  Constructor with the default settings.

Public Attributes

- unsigned n_samples

  Total number of samples.

- unsigned n_workers

  Number of worker threads.

- std::shared_ptr< hdf5::File > output_file

  The file where the observables are stored.

- bool parallelize
- unsigned seed

  Seed for the random number generator.

- bool store_parameters

6.54.1 Detailed Description

Store configuration options

6.54.2 Member Function Documentation

6.54.2.1 static Config eos::PriorSampler::Config::Default () [static]

Constructor with the default settings.

6.54.3 Member Data Documentation

6.54.3.1 unsigned eos::PriorSampler::Config::n_samples

Total number of samples.

6.54.3.2 unsigned eos::PriorSampler::Config::n_workers

Number of worker threads.

6.54.3.3 std::shared_ptr<hdf5::File> eos::PriorSampler::Config::output_file

The file where the observables are stored.

6.54.3.4 bool eos::PriorSampler::Config::parallelize

If true, use as many threads as there are cores available. If false, use only one thread.

6.54.3.5 unsigned eos::PriorSampler::Config::seed

Seed for the random number generator.

6.54.3.6 bool eos::PriorSampler::Config::store_parameters

If true, the parameter values are stored together with the observable values. If false, only the observable values are stored.

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

- /home/alexshires/EOS/eos-next/eos/utils/prior_sampler.hh

6.55 eos::proposal_functions::GlobalLocal::Config Struct Reference

```
#include <proposal_functions.hh>
```

Public Types

- enum HistoryPointWeighting { posterior, log_posterior, equal }

Static Public Member Functions

- static Config Default ()

  Named constructor.

Public Attributes

Construction options

- VerifiedRange< double > clustering_maximum_r_value
- bool clustering_strict_r_value

  Use strict definition of R or relaxed version.

- bool equal_weight_components

  Use same probability to draw from each cluster, at least initially.

- bool join_chains_symmetrically
- unsigned history_points
- unsigned history_points_local_covariance_size
- bool history_points_ordered
- HistoryPointWeighting history_point_weighting
- VerifiedRange< double > minimum_relative_cluster_weight
- bool perform_clustering
- VerifiedRange< double > rescale_local_covariance
- VerifiedRange< double > skip_initial

  Skip this percentage from beginning of a chain's history.

Runtime behavior options

- double cooling_power
- VerifiedRange< double > local_jump_probability

  How often is a local jump proposed instead of a global one.

- std::vector< unsigned > long_jump_indices

6.55.1 Member Enumeration Documentation

6.55.1.1 enum eos::proposal_functions::GlobalLocal::Config::HistoryPointWeighting

Choices on how to choose relative weight of a history point

Enumerator:

posterior

log_posterior use posterior as weight

equal use log(posterior) as weight all points have same probability

6.55.2 Member Function Documentation

6.55.2.1 static Config eos::proposal_functions::GlobalLocal::Config::Default () `[static]`

Named constructor.

6.55.3 Member Data Documentation

6.55.3.1 VerifiedRange<double> eos::proposal_functions::GlobalLocal::Config::clustering_maximum_r_value

When chains have a common R-value less than this, they are considered to overlap sufficiently.

6.55.3.2 bool eos::proposal_functions::GlobalLocal::Config::clustering_strict_r_value

Use strict definition of R or relaxed version.

6.55.3.3 double eos::proposal_functions::GlobalLocal::Config::cooling_power

Combinining the component probabilities of this and the previous step, limit the effect of samples in the distant past by forming a weighted average of last component probability, and the frequency encountered in the last chunk, . In the (n+1)-th step,

{n+1} = (1 - 1/n^{cooling_power}) + 1/n^{cooling_power} ∗ f_n

cf. arxiv:0903.0837, Eq. (23)

6.55.3.4 bool eos::proposal_functions::GlobalLocal::Config::equal_weight_components

Use same probability to draw from each cluster, at least initially.

6.55.3.5 HistoryPointWeighting eos::proposal_functions::GlobalLocal::Config::history_point_weighting

Use log(posterior) to assign relative weight to history points within each cluster. Else use posterior.

6.55.3.6 unsigned eos::proposal_functions::GlobalLocal::Config::history_points

The number of points selected from a single chain's history, useful for initializing population Monte Carlo

6.55.3.7 unsigned eos::proposal_functions::GlobalLocal::Config::history_points_local_-covariance_size

If non-zero, compute covariance around each history point locally, with the number of samples given.

6.55.3.8 bool eos::proposal_functions::GlobalLocal::Config::history_points_ordered

Choose the points from the ones with highest posterior or on a fixed grid (e.g. every 500th point)

6.55.3.9 bool eos::proposal_functions::GlobalLocal::Config::join_chains_symmetrically

Join chains symmetrically such that each chain's history (excluding the part from skip_initial) has equal contribution to the covariance matrix

6.55.3.10 VerifiedRange<double> eos::proposal_functions::GlobalLocal::Config::local_jump_-probability

How often is a local jump proposed instead of a global one.

6.55.3.11 std::vector<unsigned> eos::proposal_functions::GlobalLocal::Config::long_jump_-indices

If non-empty, all dimensions are masked and changes occur only in the dimensions specified by the indices.

6.55.3.12 VerifiedRange<double> eos::proposal_functions::GlobalLocal::Config::minimum_-relative_cluster_weight

If a cluster's weight relative to the "heaviest" cluster falls below this threshold, the cluster is discarded.

6.55.3.13 bool eos::proposal_functions::GlobalLocal::Config::perform_clustering

Compare chains by their R-values and merge them together. Else just put all chains from one partition together.

6.55.3.14 VerifiedRange<double> eos::proposal_functions::GlobalLocal::Config::rescale_local_-covariance

In order to increase efficiency, the local proposal covariance scale can be divided. Default value: 1.0, i.e. don't rescale.

Note:

Only meaningful if join_chains_symmetrically is true

6.55.3.15 VerifiedRange<double> eos::proposal_functions::GlobalLocal::Config::skip_initial

Skip this percentage from beginning of a chain's history.

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.56 eos::HierarchicalClustering::Config Class Reference

```
#include <hierarchical-clustering.hh>
```

Static Public Member Functions

Basic Function

- static Config Default ()
- static Config Quick ()

Public Attributes

- bool equal_weights

  Set component weights equal before the start of the clustering.

- bool kill_clusters

  If a cluster has zero weight, it is removed;.

- unsigned maximum_steps

  Perform a maximum number of update steps.

- double precision

6.56.1 Detailed Description

Stores all configuration options for a HierarchicalClustering.

6.56.2 Member Function Documentation

6.56.2.1 static Config eos::HierarchicalClustering::Config::Default () [static]

Named constructor

HierarchicalClustering settings with reasonably chosen default values.

6.56.2.2 static Config eos::HierarchicalClustering::Config::Quick () [static]

Named constructor

HierarchicalClustering settings with values optimized for quick chain convergence and evaluation.

Note:

The convergence is not very reliable. Use with care! If in doubt, use HierarchicalClustering::Config::Default

6.56.3 Member Data Documentation

6.56.3.1 bool eos::HierarchicalClustering::Config::equal_weights

Set component weights equal before the start of the clustering.

6.56.3.2 bool eos::HierarchicalClustering::Config::kill_clusters

If a cluster has zero weight, it is removed;.

6.56.3.3 unsigned eos::HierarchicalClustering::Config::maximum_steps

Perform a maximum number of update steps.

6.56.3.4 double eos::HierarchicalClustering::Config::precision

If relative change of distance between current and last step falls below precision, declare convergence.

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

- /home/alexshires/EOS/eos-next/eos/utils/hierarchical-clustering.hh

6.57 eos::Constraint Class Reference

`#include <constraint.hh>`

Inherits PrivateImplementationPattern< Constraint >.

Public Member Functions

- Constraint (const std::string &name, const std::vector< ObservablePtr > &observable, const std::vector< LogLikelihoodBlockPtr > &blocks)

  Constructor.

- ~Constraint ()

  Destructor.

Access

- std::string name () const

Static Public Member Functions

- static Constraint make (const std::string &name, const Options &options)

Iteration over Observables

- typedef WrappedForwardIterator< ObservableIteratorTag, ObservablePtr > ObservableIterator
- ObservableIterator begin_observables () const
- ObservableIterator end_observables () const

Iteration over Blocks

- typedef WrappedForwardIterator< BlockIteratorTag, LogLikelihoodBlockPtr > BlockIterator
- BlockIterator begin_blocks () const
- BlockIterator end_blocks () const

6.57.1 Detailed Description

Constraint models experimental constraints via one or more LogLikelihoodBlock objects that depend on one more Observable objects.

6.57.2 Member Typedef Documentation

6.57.2.1 typedef WrappedForwardIterator<BlockIteratorTag, LogLikelihoodBlockPtr> eos::Constraint::BlockIterator

6.57.2.2 typedef WrappedForwardIterator<ObservableIteratorTag, ObservablePtr> eos::Constraint::ObservableIterator

6.57.3 Constructor & Destructor Documentation

6.57.3.1 eos::Constraint::Constraint (const std::string & *name*, const std::vector< ObservablePtr > & *observable*, const std::vector< LogLikelihoodBlockPtr > & *blocks*)

Constructor.

6.57.3.2 eos::Constraint::~Constraint ()

Destructor.

6.57.4 Member Function Documentation

6.57.4.1 BlockIterator eos::Constraint::begin_blocks () const

6.57.4.2 ObservableIterator eos::Constraint::begin_observables () const

6.57.4.3 BlockIterator eos::Constraint::end_blocks () const

6.57.4.4 ObservableIterator eos::Constraint::end_observables () const

6.57.4.5 static Constraint eos::Constraint::make (const std::string & *name*, const Options & *options*) `[static]`

Create one of the builtin constraints by name.

Constraint names are structured as PROCESS::NAME-YEAR

Note:

You need to clone each LogLikelihoodBlock of a Constraint to actually be able to use it.

6.57.4.6 std::string eos::Constraint::name () const

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

- /home/alexshires/EOS/eos-next/eos/constraint.hh

6.58 eos::hdf5::DataSet< T_ > Class Template Reference

```
#include <hdf5.hh>
```

Public Types

- typedef T_::Type RecordType

Public Member Functions

Basic Functions

Destructor.

- ~DataSet ()

Metadata Access

Name of the data set

- std::string name () const
  Number of rows in data set.
- unsigned records () const
  Number of rows in data set.

Data Access

- void end ()
  Set index to last record.
- RecordType operator[ ] (const unsigned &index)
- void set_index (const unsigned &index)
  Set index to last record.

Attribute Access

- template<typename U_ >
  Attribute< U_ > create_attribute (const std::string &name, const U_ &u)
- template<typename U_ >
  Attribute< U_ > create_or_open_attribute (const std::string &name, const U_ &u)
- template<typename U_ >
  Attribute< U_ > open_attribute (const std::string &name, const U_ &u)

Friends

- class hdf5::File
- DataSet< T_ > & operator<< (DataSet< T_ > &, const typename T_::Type &)
- DataSet< T_ > & operator>> (DataSet< T_ > &, typename T_::Type &)

6.58.1 Detailed Description

template<typename T_> class eos::hdf5::DataSet< T_ >

DataSet<> represents one of the data sets within an hdf5 file.

6.58.2 Member Typedef Documentation

6.58.2.1 template<typename T_> typedef T_::Type eos::hdf5::DataSet< T_ >::RecordType

6.58.3 Constructor & Destructor Documentation

6.58.3.1 template<typename T_> eos::hdf5::DataSet< T_ >::~DataSet () [inline]

6.58.4 Member Function Documentation

6.58.4.1 template<typename T_> template<typename U_ > Attribute<U_> eos::hdf5::DataSet< T_ >::create_attribute (const std::string & *name*, const U_ & *u*) [inline]

6.58.4.2 template<typename T_> template<typename U_ > Attribute<U_> eos::hdf5::DataSet< T_ >::create_or_open_attribute (const std::string & *name*, const U_ & *u*) [inline]

6.58.4.3 template<typename T_> void eos::hdf5::DataSet< T_ >::end () [inline]

Set index to last record.

6.58.4.4 template<typename T_> std::string eos::hdf5::DataSet< T_ >::name () const

Number of rows in data set.

6.58.4.5 template<typename T_> template<typename U_ > Attribute<U_> eos::hdf5::DataSet< T_ >::open_attribute (const std::string & *name*, const U_ & *u*) [inline]

6.58.4.6 template<typename T_> RecordType eos::hdf5::DataSet< T_ >::operator[ ] (const unsigned & *index*)

Retrieve a record.

Parameters:

index Index of the record that shall be retrieved.

6.58.4.7 template<typename T_> unsigned eos::hdf5::DataSet< T_ >::records () const [inline]

Number of rows in data set.

6.58.4.8 template<typename T_> void eos::hdf5::DataSet< T_ >::set_index (const unsigned & *index*) [inline]

Set index to last record.

6.58.5 Friends And Related Function Documentation

6.58.5.1 template<typename T_> friend class hdf5::File [friend]

6.58.5.2 template<typename T_> DataSet<T_>& operator<< (DataSet< T_ > &, const typename T_::Type &) [friend]

6.58.5.3 template<typename T_> DataSet<T_>& operator>> (DataSet< T_ > &, typename T_::Type &) [friend]

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.59 eos::hdf5::DataSetHandle Class Reference

`#include <hdf5.hh>`

Inherits PrivateImplementationPattern< hdf5::DataSetHandle >.

Public Member Functions

- ~DataSetHandle ()
- hid_t set_id () const
- hid_t space_id () const
- hid_t type_id () const
- hsize_t size () const
- void select (hsize_t start, hsize_t count)
- void write_one (const void *buffer)
- void read_one (void *buffer)
- AttributeHandle create_attribute (const std::string &name, const hid_t &type_id)
- AttributeHandle open_attribute (const std::string &name, const hid_t &type_id)

Friends

- class File

6.59.1 Constructor & Destructor Documentation

6.59.1.1 eos::hdf5::DataSetHandle::~DataSetHandle ()

6.59.2 Member Function Documentation

6.59.2.1 AttributeHandle eos::hdf5::DataSetHandle::create_attribute (const std::string & *name*, const hid_t & *type_id*)

6.59.2.2 AttributeHandle eos::hdf5::DataSetHandle::open_attribute (const std::string & *name*, const hid_t & *type_id*)

6.59.2.3 void eos::hdf5::DataSetHandle::read_one (void * *buffer*)

6.59.2.4 void eos::hdf5::DataSetHandle::select (hsize_t *start*, hsize_t *count*)

6.59.2.5 hid_t eos::hdf5::DataSetHandle::set_id () const

6.59.2.6 hsize_t eos::hdf5::DataSetHandle::size () const

6.59.2.7 hid_t eos::hdf5::DataSetHandle::space_id () const

6.59.2.8 hid_t eos::hdf5::DataSetHandle::type_id () const

6.59.2.9 void eos::hdf5::DataSetHandle::write_one (const void * *buffer*)

6.59.3 Friends And Related Function Documentation

6.59.3.1 friend class File `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.60 eos::hdf5::DataType< T_ > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static hid_t type_id ()

template<typename T_> struct eos::hdf5::DataType< T_ >

6.60.1 Member Function Documentation

6.60.1.1 template<typename T_ > static hid_t eos::hdf5::DataType< T_ >::type_id () [static]

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.61 eos::hdf5::DataType< const char * > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static hid_t type_id ()

template<> struct eos::hdf5::DataType< const char * >

6.61.1 Member Function Documentation

6.61.1.1 static hid_t eos::hdf5::DataType< const char * >::type_id () [inline, static]

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.62 eos::hdf5::DataType< double > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static hid_t type_id ()

template<> struct eos::hdf5::DataType< double >

6.62.1 Member Function Documentation

6.62.1.1 static hid_t eos::hdf5::DataType< double >::type_id () [inline, static]

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.63 eos::hdf5::DataType< int > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static hid_t type_id ()

template<> struct eos::hdf5::DataType< int >

6.63.1 Member Function Documentation

6.63.1.1 static hid_t eos::hdf5::DataType< int >::type_id () [inline, static]

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.64 eos::hdf5::DataType< short > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static hid_t type_id ()

template<> struct eos::hdf5::DataType< short >

6.64.1 Member Function Documentation

6.64.1.1 static hid_t eos::hdf5::DataType< short >::type_id () [inline, static]

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.65 eos::hdf5::DataType< unsigned > Struct Template Reference

```
#include <hdf5.hh>
```

Static Public Member Functions

- static hid_t type_id ()

template<> struct eos::hdf5::DataType< unsigned >

6.65.1 Member Function Documentation

6.65.1.1 static hid_t eos::hdf5::DataType< unsigned >::type_id () [inline, static]

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.66 eos::InstantiationPolicy::DeleteOnDestruction< T_ > Class Template Reference

```
#include <instantiation_policy-impl.hh>
```

template<typename T_> class eos::InstantiationPolicy::DeleteOnDestruction< T_ >

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

- /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy-impl.hh

6.67 eos::Diagnostics Class Reference

`#include <diagnostics.hh>`

Inherits PrivateImplementationPattern< Diagnostics >.

Classes

- struct Entry

Public Member Functions

Basic Operations.

Constructor.

- Diagnostics ()

  Destructor.

- ~Diagnostics ()

  Destructor.

Access to Diagnostic Entries

Add an entry to the container.

- typedef WrappedForwardIterator< IteratorTag, const Entry > Iterator

  Iterator over entries.

- void add (const Entry &)

  Iterator over entries.

- Iterator begin () const

  Returns Iterator pointing to the first entry.

- Iterator end () const

  Returns Iterator pointing beyond the last entry.

- unsigned size () const

  Return the total number of entries.

6.67.1 Detailed Description

Container for diagnostic values and corresponding metadata. Used as part of the unit tests.

6.67.2 Member Typedef Documentation

6.67.2.1 typedef WrappedForwardIterator<IteratorTag, const Entry> eos::Diagnostics::Iterator

Iterator over entries.

6.67.3 Constructor & Destructor Documentation

6.67.3.1 eos::Diagnostics::Diagnostics ()

Destructor.

6.67.3.2 eos::Diagnostics::~Diagnostics ()

Destructor.

6.67.4 Member Function Documentation

6.67.4.1 void eos::Diagnostics::add (const Entry &)

Iterator over entries.

6.67.4.2 Iterator eos::Diagnostics::begin () const

Returns Iterator pointing to the first entry.

6.67.4.3 Iterator eos::Diagnostics::end () const

Returns Iterator pointing beyond the last entry.

6.67.4.4 unsigned eos::Diagnostics::size () const

Return the total number of entries.

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

- /home/alexshires/EOS/eos-next/eos/utils/diagnostics.hh

6.68 eos::implementation::DoStringify< T_ > Struct Template Reference

```
#include <stringify.hh>
```

Static Public Member Functions

- static std::string stringify (const T_ &x, unsigned precision)

template<typename T_> struct eos::implementation::DoStringify< T_ >

6.68.1 Member Function Documentation

6.68.1.1 template<typename T_ > static std::string eos::implementation::DoStringify< T_ >::stringify (const T_ & *x*, unsigned *precision*) [inline, static]

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

- /home/alexshires/EOS/eos-next/eos/utils/stringify.hh

6.69 eos::EMContributions Struct Reference

```
#include <em-contributions.hh>
```

Static Public Member Functions

- static double omegaem_22 (const double &s_hat, const double &log_m_l_hat, const double &mu)
- static complex< double > omegaem_27 (const double &s_hat, const double &log_m_l_hat, const double &mu)
- static complex< double > omegaem_29 (const double &s_hat, const double &log_m_l_hat, const double &mu)
- static double omegaem_77 (const double &s_hat, const double &log_m_l_hat)
- static double omegaem_79 (const double &s_hat, const double &log_m_l_hat)
- static double omegaem_99 (const double &s_hat, const double &log_m_l_hat)
- static double omegaem_1010 (const double &s_hat, const double &log_m_l_hat)

6.69.1 Member Function Documentation

6.69.1.1 static double eos::EMContributions::omegaem_1010 (const double & *s_hat*, const double & *log_m_l_hat*) [static]

6.69.1.2 static double eos::EMContributions::omegaem_22 (const double & *s_hat*, const double & *log_m_l_hat*, const double & *mu*) [static]

6.69.1.3 static complex<double> eos::EMContributions::omegaem_27 (const double & *s_hat*, const double & *log_m_l_hat*, const double & *mu*) [static]

6.69.1.4 static complex<double> eos::EMContributions::omegaem_29 (const double & *s_hat*, const double & *log_m_l_hat*, const double & *mu*) [static]

6.69.1.5 static double eos::EMContributions::omegaem_77 (const double & *s_hat*, const double & *log_m_l_hat*) [static]

6.69.1.6 static double eos::EMContributions::omegaem_79 (const double & *s_hat*, const double & *log_m_l_hat*) [static]

6.69.1.7 static double eos::EMContributions::omegaem_99 (const double & *s_hat*, const double & *log_m_l_hat*) [static]

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/em-contributions.hh

6.70 eos::Diagnostics::Entry Struct Reference

```
#include <diagnostics.hh>
```

Public Attributes

- double value

  Numeric value for this entry.

- std::string description

  Description for this entry, e.g. a reference to the underlying formula.

6.70.1 Detailed Description

Data structure representing one entry of several diagnostic evaluations. Used as part of the unit tests.

6.70.2 Member Data Documentation

6.70.2.1 std::string eos::Diagnostics::Entry::description

Description for this entry, e.g. a reference to the underlying formula.

6.70.2.2 double eos::Diagnostics::Entry::value

Numeric value for this entry.

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

- /home/alexshires/EOS/eos-next/eos/utils/diagnostics.hh

6.71 eos::EquationSolver Class Reference

```
#include <equation_solver.hh>
```

Inherits PrivateImplementationPattern< EquationSolver >.

Classes

- struct Config
- struct Solution

Public Types

- typedef std::function< double(const std::vector< double > &)> Equation

Public Member Functions

- EquationSolver (const EquationSolver::Config &config)
- ~EquationSolver ()
- void add (const Equation &equation)
- void add (const std::string &name, const double &initial_value, const double &error)
- void add (const std::string &name, const double &initial_value, const double &error, const double &min, const double &max)
- Solution solve ()

6.71.1 Member Typedef Documentation

6.71.1.1 typedef std::function<double (const std::vector<double> &)> eos::EquationSolver::Equation

6.71.2 Constructor & Destructor Documentation

6.71.2.1 eos::EquationSolver::EquationSolver (const EquationSolver::Config & *config*)

6.71.2.2 eos::EquationSolver::~EquationSolver ()

6.71.3 Member Function Documentation

6.71.3.1 void eos::EquationSolver::add (const std::string & *name*, const double & *initial_value*, const double & *error*, const double & *min*, const double & *max*)

6.71.3.2 void eos::EquationSolver::add (const std::string & *name*, const double & *initial_value*, const double & *error*)

6.71.3.3 void eos::EquationSolver::add (const Equation & *equation*)

6.71.3.4 Solution eos::EquationSolver::solve ()

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

- /home/alexshires/EOS/eos-next/eos/utils/equation_solver.hh

6.72 eos::Exception Class Reference

`#include <exception.hh>`

Inherits std::exception.

Inherited by eos::HDF5Error, eos::InternalError, eos::NoSuchChiSquaredError, eos::NoSuchModelError, eos::ObservableNameError, eos::proposal_functions::UnknownProposalError, eos::UnknownConstraintError, eos::UnknownKinematicVariableError, eos::UnknownOptionError, eos::UnknownParameterError, and eos::VerifiedRangeError.

Public Member Functions

- ∼Exception () throw ()
- virtual const char ∗ what () const throw ()

Protected Member Functions

- Exception (const std::string &message) throw ()

6.72.1 Constructor & Destructor Documentation

6.72.1.1 eos::Exception::Exception (const std::string & *message*) throw () [protected]

6.72.1.2 eos::Exception::∼Exception () throw ()

6.72.2 Member Function Documentation

6.72.2.1 virtual const char∗ eos::Exception::what () const throw () [virtual]

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

- /home/alexshires/EOS/eos-next/eos/utils/exception.hh

6.73 eos::proposal_functions::Factory Struct Reference

Abstract factory to read in a proposal density from file.

`#include <proposal_functions.hh>`

Static Public Member Functions

- static ProposalFunctionPtr make (const hdf5::File &file, const std::string &data_set_base_name, const std::string &proposal_name, const unsigned &dimension)

6.73.1 Detailed Description

Abstract factory to read in a proposal density from file.

6.73.2 Member Function Documentation

6.73.2.1 static ProposalFunctionPtr eos::proposal_functions::Factory::make (const hdf5::File & *file*, const std::string & *data_set_base_name*, const std::string & *proposal_name*, const unsigned & *dimension*) [static]

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.74 eos::hdf5::File Class Reference

```
#include <hdf5.hh>
```

Public Member Functions

Data Access

- template<typename T_ >
  DataSet< T_ > create_data_set (const std::string &name, const T_ &t)
- template<typename T_ >
  DataSet< T_ > open_data_set (const std::string &name, const T_ &t)
- template<typename T_ >
  DataSet< T_ > create_or_open_data_set (const std::string &name, const T_ &t)

Group Operations

- void copy (const std::string &source_directory, File &destination_file, const std::string &destination_directory="") const
- void create_group (const std::string &name)
- bool group_exists (const std::string &name)
- unsigned number_of_objects (const std::string &name)

  List how many objects, i.e. groups or data sets, are in a subdirectory.

Basic Functions

Destructor.

- ~File ()

  Access to file name.

- const std::string & name () const

  Access to file name.

- static File Create (const std::string &filename)
- static File Open (const std::string &file_name, unsigned mode=H5F_ACC_RDONLY)
- static bool Exists (const std::string &filename)

  Access to file name.

6.74.1 Constructor & Destructor Documentation

6.74.1.1 eos::hdf5::File::~File ()

Access to file name.

6.74.2 Member Function Documentation

6.74.2.1 void eos::hdf5::File::copy (const std::string & *source_directory*, File & *destination_file*, const std::string & *destination_directory* = "") const

Copy all objects under source directory in this file to the destination file. If the destination directory is omitted, the source directory is recreated in the destination file.

6.74.2.2 static File eos::hdf5::File::Create (const std::string & *filename*) [static]

Named constructor to create a new file.

Parameters:

filename File which shall be created.

creator Name of the creating program.

6.74.2.3 template<typename T_ > DataSet<T_> eos::hdf5::File::create_data_set (const std::string & *name*, const T_ & *t*) [inline]

Create a new data set by name.

Parameters:

name Absolute name of the new data set.

t Instance of any of Scalar, Array or Composite that represents this data set's underlying data type.

6.74.2.4 void eos::hdf5::File::create_group (const std::string & *name*)

Create a group within this HDF5 file.

Parameters:

name Name of the group that shall be created.

6.74.2.5 template<typename T_ > DataSet<T_> eos::hdf5::File::create_or_open_data_set (const std::string & *name*, const T_ & *t*) [inline]

Open a data set by name. Create the data set if it doesn't exist.

Note:

The error handling of the HDF5 library is suppressed for the duration of the execution. Do not use this method when parallel access to a file is expected.

Parameters:

name Absolute name of the data set.

t Instance of any of Scalar, Array or Composite that represents this data set's underlying data type.

6.74.2.6 static bool eos::hdf5::File::Exists (const std::string & *filename*) [static]

Access to file name.

6.74.2.7 bool eos::hdf5::File::group_exists (const std::string & *name*)

Copy all objects under source directory in this file to the destination file. If the destination directory is omitted, the source directory is recreated in the destination file.

6.74.2.8 const std::string& eos::hdf5::File::name () const [inline]

Access to file name.

6.74.2.9 unsigned eos::hdf5::File::number_of_objects (const std::string & *name*)

List how many objects, i.e. groups or data sets, are in a subdirectory.

6.74.2.10 static File eos::hdf5::File::Open (const std::string & *file_name*, unsigned *mode* = H5F_ACC_RDONLY) [static]

Named constructor to open an existing file.

Parameters:

file_name File which shall be opened.

6.74.2.11 template<typename T_ > DataSet<T_> eos::hdf5::File::open_data_set (const std::string & *name*, const T_ & *t*) [inline]

Open an existing data set by name.

Parameters:

name Absolute name of the data set.

t Instance of any of Scalar, Array or Composite that represents this data set's underlying data type.

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.75 eos::hdf5::FileHandle Class Reference

`#include <hdf5.hh>`

Inherits PrivateImplementationPattern< hdf5::FileHandle >.

Public Member Functions

- ~FileHandle ()
- hid_t id () const
- bool read_only () const

Friends

- class File

6.75.1 Constructor & Destructor Documentation

6.75.1.1 eos::hdf5::FileHandle::~FileHandle ()

6.75.2 Member Function Documentation

6.75.2.1 hid_t eos::hdf5::FileHandle::id () const

6.75.2.2 bool eos::hdf5::FileHandle::read_only () const

6.75.3 Friends And Related Function Documentation

6.75.3.1 friend class File `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.76 eos::FormFactorAdapter< Transition_ > Class Template Reference

`#include <form-factor-adapter.hh>`

Inherits eos::Observable.

Public Member Functions

- FormFactorAdapter (const std::string &name, const std::string &process, const Parameters &parameters, const Kinematics &kinematics, const Options &options, const std::function< double(const FormFactors< Transition_ > *, const double &)> &form_factor_function)
- virtual const std::string & name () const
- virtual double evaluate () const
- virtual Parameters parameters ()
- virtual Kinematics kinematics ()
- virtual Options options ()
- virtual ObservablePtr clone () const
- virtual ObservablePtr clone (const Parameters &parameters) const

template<typename Transition_> class eos::FormFactorAdapter< Transition_ >

6.76.1 Constructor & Destructor Documentation

6.76.1.1 template<typename Transition_ > eos::FormFactorAdapter< Transition_ >::FormFactorAdapter (const std::string & *name*, const std::string & *process*, const Parameters & *parameters*, const Kinematics & *kinematics*, const Options & *options*, const std::function< double(const FormFactors< Transition_ > *, const double &)> & *form_factor_function*) `[inline]`

6.76.2 Member Function Documentation

6.76.2.1 template<typename Transition_ > virtual ObservablePtr eos::FormFactorAdapter< Transition_ >::clone (const Parameters & *parameters*) const `[inline, virtual]`

Implements eos::Observable.

6.76.2.2 template<typename Transition_ > virtual ObservablePtr eos::FormFactorAdapter< Transition_ >::clone () const `[inline, virtual]`

Implements eos::Observable.

6.76.2.3 template<typename Transition_ > virtual double eos::FormFactorAdapter< Transition_ >::evaluate () const `[inline, virtual]`

Implements eos::Observable.

6.76.2.4 template<typename Transition_ > virtual Kinematics eos::FormFactorAdapter< Transition_ >::kinematics () [inline, virtual]

Implements eos::Observable.

6.76.2.5 template<typename Transition_ > virtual const std::string& eos::FormFactorAdapter< Transition_ >::name () const [inline, virtual]

Implements eos::Observable.

6.76.2.6 template<typename Transition_ > virtual Options eos::FormFactorAdapter< Transition_ >::options () [inline, virtual]

Implements eos::Observable.

6.76.2.7 template<typename Transition_ > virtual Parameters eos::FormFactorAdapter< Transition_ >::parameters () [inline, virtual]

Implements eos::Observable.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh

6.77 eos::FormFactorAdapterFactory< Transition_ > Class Template Reference

`#include <form-factor-adapter.hh>`

Inherits eos::ObservableFactory.

Public Member Functions

- FormFactorAdapterFactory (const std::string &name, const std::string &process, const std::function< double(const FormFactors< Transition_ > *, const double &)> &form_factor_function)
- ~FormFactorAdapterFactory ()
- virtual ObservablePtr make (const Parameters &parameters, const Kinematics &kinematics, const Options &options) const

template<typename Transition_> class eos::FormFactorAdapterFactory< Transition_ >

6.77.1 Constructor & Destructor Documentation

6.77.1.1 template<typename Transition_ > eos::FormFactorAdapterFactory< Transition_ >::FormFactorAdapterFactory (const std::string & *name*, const std::string & *process*, const std::function< double(const FormFactors< Transition_ > *, const double &)> & *form_factor_function*) `[inline]`

6.77.1.2 template<typename Transition_ > eos::FormFactorAdapterFactory< Transition_ >::~FormFactorAdapterFactory () `[inline]`

6.77.2 Member Function Documentation

6.77.2.1 template<typename Transition_ > virtual ObservablePtr eos::FormFactorAdapterFactory< Transition_ >::make (const Parameters & *parameters*, const Kinematics & *kinematics*, const Options & *options*) const `[inline, virtual]`

Implements eos::ObservableFactory.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh

6.78 eos::FormFactorFactory< OneHalfPlusToOneHalfPlus > Class Template Reference

```
#include <baryonic.hh>
```

Static Public Member Functions

- static std::shared_ptr< FormFactors< OneHalfPlusToOneHalfPlus > > create (const std::string &label, const Parameters &parameters)

template<> class eos::FormFactorFactory< OneHalfPlusToOneHalfPlus >

6.78.1 Member Function Documentation

6.78.1.1 static std::shared_ptr<FormFactors<OneHalfPlusToOneHalfPlus> > eos::FormFactorFactory< OneHalfPlusToOneHalfPlus >::create (const std::string & *label*, const Parameters & *parameters*) [static]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/baryonic.hh

6.79 eos::FormFactorFactory< PToP > Class Template Reference

`#include <mesonic.hh>`

Static Public Member Functions

- static std::shared_ptr< FormFactors< PToP > > create (const std::string &label, const Parameters &parameters)

template<> class eos::FormFactorFactory< PToP >

6.79.1 Member Function Documentation

6.79.1.1 static std::shared_ptr<FormFactors<PToP> > eos::FormFactorFactory< PToP >::create (const std::string & *label*, const Parameters & *parameters*) [static]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh

6.80 eos::FormFactorFactory< PToV > Class Template Reference

```
#include <mesonic.hh>
```

Static Public Member Functions

- static std::shared_ptr< FormFactors< PToV > > create (const std::string &label, const Parameters &parameters)

template<> class eos::FormFactorFactory< PToV >

6.80.1 Member Function Documentation

6.80.1.1 static std::shared_ptr<FormFactors<PToV> > eos::FormFactorFactory< PToV >::create (const std::string & *label*, const Parameters & *parameters*) [static]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh

6.81 eos::FormFactorRatioAdapter< Transition_ > Class Template Reference

`#include <form-factor-adapter.hh>`

Inherits eos::Observable.

Public Member Functions

- FormFactorRatioAdapter (const std::string &name, const std::string &process, const Parameters &parameters, const Kinematics &kinematics, const Options &options, const std::function< double(const FormFactors< Transition_ > *, const double &)> &form_factor_numerator, const std::function< double(const FormFactors< Transition_ > *, const double &)> &form_factor_denominator)
- virtual const std::string & name () const
- virtual double evaluate () const
- virtual Parameters parameters ()
- virtual Kinematics kinematics ()
- virtual Options options ()
- virtual ObservablePtr clone () const
- virtual ObservablePtr clone (const Parameters &parameters) const

template<typename Transition_> class eos::FormFactorRatioAdapter< Transition_ >

6.81.1 Constructor & Destructor Documentation

6.81.1.1 template<typename Transition_ > eos::FormFactorRatioAdapter< Transition_ >::FormFactorRatioAdapter (const std::string & *name*, const std::string & *process*, const Parameters & *parameters*, const Kinematics & *kinematics*, const Options & *options*, const std::function< double(const FormFactors< Transition_ > *, const double &)> & *form_factor_numerator*, const std::function< double(const FormFactors< Transition_ > *, const double &)> & *form_factor_denominator*) `[inline]`

6.81.2 Member Function Documentation

6.81.2.1 template<typename Transition_ > virtual ObservablePtr eos::FormFactorRatioAdapter< Transition_ >::clone (const Parameters & *parameters*) const `[inline, virtual]`

Implements eos::Observable.

6.81.2.2 template<typename Transition_ > virtual ObservablePtr eos::FormFactorRatioAdapter< Transition_ >::clone () const `[inline, virtual]`

Implements eos::Observable.

6.81.2.3 template<typename Transition_ > virtual double eos::FormFactorRatioAdapter< Transition_ >::evaluate () const `[inline, virtual]`

Implements eos::Observable.

6.81.2.4 template<typename Transition_ > virtual Kinematics eos::FormFactorRatioAdapter< Transition_ >::kinematics () [inline, virtual]

Implements eos::Observable.

6.81.2.5 template<typename Transition_ > virtual const std::string& eos::FormFactorRatioAdapter< Transition_ >::name () const [inline, virtual]

Implements eos::Observable.

6.81.2.6 template<typename Transition_ > virtual Options eos::FormFactorRatioAdapter< Transition_ >::options () [inline, virtual]

Implements eos::Observable.

6.81.2.7 template<typename Transition_ > virtual Parameters eos::FormFactorRatioAdapter< Transition_ >::parameters () [inline, virtual]

Implements eos::Observable.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh

6.82 eos::FormFactorRatioAdapterFactory< Transition_ > Class Template Reference

`#include <form-factor-adapter.hh>`

Inherits eos::ObservableFactory.

Public Member Functions

- FormFactorRatioAdapterFactory (const std::string &name, const std::string &process, const std::function< double(const FormFactors< Transition_ > *, const double &)> &form_factor_numerator, const std::function< double(const FormFactors< Transition_ > *, const double &)> &form_factor_denominator)
- ~FormFactorRatioAdapterFactory ()
- virtual ObservablePtr make (const Parameters &parameters, const Kinematics &kinematics, const Options &options) const

template<typename Transition_> class eos::FormFactorRatioAdapterFactory< Transition_ >

6.82.1 Constructor & Destructor Documentation

6.82.1.1 template<typename Transition_ > eos::FormFactorRatioAdapterFactory< Transition_ >::FormFactorRatioAdapterFactory (const std::string & *name*, const std::string & *process*, const std::function< double(const FormFactors< Transition_ > *, const double &)> & *form_factor_numerator*, const std::function< double(const FormFactors< Transition_ > *, const double &)> & *form_factor_denominator*) `[inline]`

6.82.1.2 template<typename Transition_ > eos::FormFactorRatioAdapterFactory< Transition_ >::~FormFactorRatioAdapterFactory () `[inline]`

6.82.2 Member Function Documentation

6.82.2.1 template<typename Transition_ > virtual ObservablePtr eos::FormFactorRatioAdapterFactory< Transition_ >::make (const Parameters & *parameters*, const Kinematics & *kinematics*, const Options & *options*) const `[inline, virtual]`

Implements eos::ObservableFactory.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh

6.83 eos::FormFactors< OneHalfPlusToOneHalfPlus > Class Template Reference

`#include <baryonic.hh>`

Inherits eos::ParameterUser.

Public Member Functions

- virtual ~FormFactors ()
- virtual double f_long_v (const double &s) const =0
- virtual double f_perp_v (const double &s) const =0
- virtual double f_long_a (const double &s) const =0
- virtual double f_perp_a (const double &s) const =0

template<> class eos::FormFactors< OneHalfPlusToOneHalfPlus >

6.83.1 Constructor & Destructor Documentation

6.83.1.1 virtual eos::FormFactors< OneHalfPlusToOneHalfPlus >::~FormFactors () [virtual]

6.83.2 Member Function Documentation

6.83.2.1 virtual double eos::FormFactors< OneHalfPlusToOneHalfPlus >::f_long_a (const double & *s*) const [pure virtual]

6.83.2.2 virtual double eos::FormFactors< OneHalfPlusToOneHalfPlus >::f_long_v (const double & *s*) const [pure virtual]

6.83.2.3 virtual double eos::FormFactors< OneHalfPlusToOneHalfPlus >::f_perp_a (const double & *s*) const [pure virtual]

6.83.2.4 virtual double eos::FormFactors< OneHalfPlusToOneHalfPlus >::f_perp_v (const double & *s*) const [pure virtual]

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

- /home/alexshires/EOS/eos-next/eos/form-factors/baryonic.hh

6.84 eos::FormFactors< PToP > Class Template Reference

```
#include <mesonic.hh>
```

Inherits eos::ParameterUser.

Inherited by eos::AnalyticFormFactorBToPiDKMMO2008, eos::BFW2010FormFactors< PToP >, eos::BZ2004FormFactors< Process_, PToP >, eos::BZ2004FormFactorsSplit< Process_ >, and eos::KMPW2010FormFactors< PToP >.

Public Member Functions

- virtual ~FormFactors ()
- virtual double f_p (const double &s) const =0
- virtual double f_0 (const double &s) const =0
- virtual double f_t (const double &s) const =0
- virtual double f_p_d1 (const double &s) const
- virtual double f_p_d2 (const double &s) const

template<> class eos::FormFactors< PToP >

6.84.1 Constructor & Destructor Documentation

6.84.1.1 virtual eos::FormFactors< PToP >::~FormFactors () [virtual]

6.84.2 Member Function Documentation

6.84.2.1 virtual double eos::FormFactors< PToP >::f_0 (const double & *s*) const [pure virtual]

Implemented in eos::AnalyticFormFactorBToPiDKMMO2008, eos::BZ2004FormFactors< Process_, PToP >, eos::BZ2004FormFactorsSplit< Process_ >, eos::KMPW2010FormFactors< PToP >, and eos::BFW2010FormFactors< PToP >.

6.84.2.2 virtual double eos::FormFactors< PToP >::f_p (const double & *s*) const [pure virtual]

Implemented in eos::AnalyticFormFactorBToPiDKMMO2008, eos::BZ2004FormFactors< Process_, PToP >, eos::BZ2004FormFactorsSplit< Process_ >, eos::KMPW2010FormFactors< PToP >, and eos::BFW2010FormFactors< PToP >.

6.84.2.3 double eos::FormFactors< PToP >::f_p_d1 (const double & *s*) const [virtual]

6.84.2.4 double eos::FormFactors< PToP >::f_p_d2 (const double & *s*) const [virtual]

6.84.2.5 virtual double eos::FormFactors< PToP >::f_t (const double & *s*) const [pure virtual]

Implemented in eos::AnalyticFormFactorBToPiDKMMO2008, eos::BZ2004FormFactors< Process_, PToP >, eos::BZ2004FormFactorsSplit< Process_ >, eos::KMPW2010FormFactors< PToP >, and eos::BFW2010FormFactors< PToP >.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh
- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.85 eos::FormFactors< PToV > Class Template Reference

`#include <mesonic.hh>`

Inherits eos::ParameterUser.

Inherited by eos::BZ2004FormFactors< Process_, PToV >, and eos::KMPW2010FormFactors< PToV >.

Public Member Functions

- virtual ~FormFactors ()
- virtual double v (const double &s) const =0
- virtual double a_0 (const double &s) const =0
- virtual double a_1 (const double &s) const =0
- virtual double a_2 (const double &s) const =0
- virtual double a_12 (const double &s) const =0

template<> class eos::FormFactors< PToV >

6.85.1 Constructor & Destructor Documentation

6.85.1.1 virtual eos::FormFactors< PToV >::~FormFactors () `[virtual]`

6.85.2 Member Function Documentation

6.85.2.1 virtual double eos::FormFactors< PToV >::a_0 (const double & *s*) const `[pure virtual]`

Implemented in eos::BZ2004FormFactors< Process_, PToV >, and eos::KMPW2010FormFactors< PToV >.

6.85.2.2 virtual double eos::FormFactors< PToV >::a_1 (const double & *s*) const `[pure virtual]`

Implemented in eos::BZ2004FormFactors< Process_, PToV >, and eos::KMPW2010FormFactors< PToV >.

6.85.2.3 virtual double eos::FormFactors< PToV >::a_12 (const double & *s*) const `[pure virtual]`

Implemented in eos::BZ2004FormFactors< Process_, PToV >, and eos::KMPW2010FormFactors< PToV >.

6.85.2.4 virtual double eos::FormFactors< PToV >::a_2 (const double & *s*) const `[pure virtual]`

Implemented in eos::BZ2004FormFactors< Process_, PToV >, and eos::KMPW2010FormFactors< PToV >.

6.85.2.5 virtual double eos::FormFactors< PToV >::v (const double & *s*) const [pure virtual]

Implemented in eos::BZ2004FormFactors< Process_, PToV >, and eos::KMPW2010FormFactors< PToV >.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh

6.86 eos::proposal_functions::GlobalLocal Class Reference

`#include <proposal_functions.hh>`

Inherits eos::MarkovChain::ProposalFunction.

Classes

- struct Config

Public Types

- typedef hdf5::Composite< hdf5::Scalar< unsigned >, hdf5::Array< 1, double > > ComponentType

  Types of data sets for reading/writing the state of the proposal density from/to disk.

- typedef hdf5::Composite< hdf5::Array< 1, double >, hdf5::Scalar< double >, hdf5::Scalar< double >, hdf5::Scalar< double > > HistoryType
- typedef hdf5::Composite< hdf5::Array< 1, double >, hdf5::Scalar< double > > JumpType
- typedef hdf5::Composite< hdf5::Array< 1, double >, hdf5::Array< 1, double > > LocalCovarianceType

Public Member Functions

- GlobalLocal (const std::vector< HistoryPtr > &chains, const std::vector< ProposalFunctionPtr > &proposals, const std::vector< MarkovChain::Stats > &stats, const Config &config, const unsigned &prerun_chains_per_cluster)
- GlobalLocal (const std::vector< double > &component_probabilities, const unsigned &adaptations, const std::vector< MarkovChain::State > &jump_states, const std::vector< MarkovChain::State > &modes, const std::vector< std::vector< MarkovChain::State >> &history_states, const std::vector< std::vector< double >> &history_point_probabilities, const std::vector< std::vector< std::vector< double >>> &history_local_covariances, const std::vector< ProposalFunctionPtr > &proposals)
- virtual ~GlobalLocal ()
- virtual void adapt (const MarkovChain::State::Iterator &begin, const MarkovChain::State::Iterator &end, const double &efficiency, const double &efficiency_min, const double &efficiency_max)

  No adaptation currently foreseen.

- virtual MarkovChain::State mode () const
- const std::vector< double > & component_probabilities () const
- const std::vector< std::vector< MarkovChain::State > > & history_states () const
- const std::vector< std::vector< std::vector< double > > > & local_covariances () const
- virtual void config (const Config &config)

  Reset config options.

- virtual ProposalFunctionPtr clone () const

  Create independent copy.

- virtual void dump_state (hdf5::File &file, const std::string &data_set_base_name) const

  Store state in the file under the given base name.

- virtual double evaluate (const MarkovChain::State &x, const MarkovChain::State &y) const

  Evaluate the density to propose x given y.

- virtual void propose (MarkovChain::State &x, const MarkovChain::State &y, gsl_rng *rng) const

  Obtain from the density a proposal x given y.

Static Public Member Functions

- static ComponentType component_type (const unsigned &dimension)
- static HistoryType history_type (const unsigned &dimension)
- static JumpType jump_type (const unsigned &dimension)
- static LocalCovarianceType local_covariance_type (const unsigned &dimension)

6.86.1 Detailed Description

After a suitable amount of prerun iterations, combine the information of all chains into one, common proposal function.

6.86.2 Member Typedef Documentation

6.86.2.1 typedef hdf5::Composite<hdf5::Scalar<unsigned>, hdf5::Array<1, double> > eos::proposal_functions::GlobalLocal::ComponentType

Types of data sets for reading/writing the state of the proposal density from/to disk.

6.86.2.2 typedef hdf5::Composite<hdf5::Array<1, double>, hdf5::Scalar<double>, hdf5::Scalar<double>, hdf5::Scalar<double> > eos::proposal_functions::GlobalLocal::HistoryType

6.86.2.3 typedef hdf5::Composite<hdf5::Array<1, double>, hdf5::Scalar<double> > eos::proposal_functions::GlobalLocal::JumpType

6.86.2.4 typedef hdf5::Composite<hdf5::Array<1, double>, hdf5::Array<1, double> > eos::proposal_functions::GlobalLocal::LocalCovarianceType

6.86.3 Constructor & Destructor Documentation

6.86.3.1 eos::proposal_functions::GlobalLocal::GlobalLocal (const std::vector< HistoryPtr > & *chains*, const std::vector< ProposalFunctionPtr > & *proposals*, const std::vector< MarkovChain::Stats > & *stats*, const Config & *config*, const unsigned & *prerun_chains_per_cluster*)

Take the important parts of MarkovChains, and combine them into a global local proposal density. It is assumed that in each vector, the parts at a given index belong to the same chain. If their histories only contain the last chunk, variability of the algorithm is limited.

chains Use pointers to the histories, such that unnecessary copying is avoided.

proposal The proposal density of each chain.

stats The statistics of each chain.

local_jump_probability In order to allow chains to explore more distant regions,

a value close to one is advised.

history_points the number of points selected from a single chain's history

6.86.3.2 eos::proposal_functions::GlobalLocal::GlobalLocal (const std::vector< double > & *component_probabilities*, const unsigned & *adaptations*, const std::vector< MarkovChain::State > & *jump_states*, const std::vector< MarkovChain::State > & *modes*, const std::vector< std::vector< MarkovChain::State >> & *history_states*, const std::vector< std::vector< double >> & *history_point_probabilities*, const std::vector< std::vector< std::vector< double >>> & *history_local_covariances*, const std::vector< ProposalFunctionPtr > & *proposals*)

6.86.3.3 virtual eos::proposal_functions::GlobalLocal::~GlobalLocal () `[virtual]`

6.86.4 Member Function Documentation

6.86.4.1 virtual void eos::proposal_functions::GlobalLocal::adapt (const MarkovChain::State::Iterator & *begin*, const MarkovChain::State::Iterator & *end*, const double & *efficiency*, const double & *efficiency_min*, const double & *efficiency_max*) `[virtual]`

No adaptation currently foreseen.

Implements eos::MarkovChain::ProposalFunction.

6.86.4.2 virtual ProposalFunctionPtr eos::proposal_functions::GlobalLocal::clone () const `[virtual]`

Create independent copy.

Implements eos::MarkovChain::ProposalFunction.

6.86.4.3 const std::vector<double>& eos::proposal_functions::GlobalLocal::component_probabilities () const [inline]

6.86.4.4 static ComponentType eos::proposal_functions::GlobalLocal::component_type (const unsigned & *dimension*) [static]

6.86.4.5 virtual void eos::proposal_functions::GlobalLocal::config (const Config & *config*) [virtual]

Reset config options.

6.86.4.6 virtual void eos::proposal_functions::GlobalLocal::dump_state (hdf5::File & *file*, const std::string & *data_set_base_name*) const [virtual]

Store state in the file under the given base name.

Implements eos::MarkovChain::ProposalFunction.

6.86.4.7 virtual double eos::proposal_functions::GlobalLocal::evaluate (const MarkovChain::State & *x*, const MarkovChain::State & *y*) const [virtual]

Evaluate the density to propose x given y.

Implements eos::MarkovChain::ProposalFunction.

6.86.4.8 const std::vector<std::vector<MarkovChain::State> >& eos::proposal_functions::GlobalLocal::history_states () const [inline]

6.86.4.9 static HistoryType eos::proposal_functions::GlobalLocal::history_type (const unsigned & *dimension*) [static]

6.86.4.10 static JumpType eos::proposal_functions::GlobalLocal::jump_type (const unsigned & *dimension*) [static]

6.86.4.11 static LocalCovarianceType eos::proposal_functions::GlobalLocal::local_covariance_type (const unsigned & *dimension*) [static]

6.86.4.12 const std::vector<std::vector<std::vector<double> > >& eos::proposal_functions::GlobalLocal::local_covariances () const [inline]

6.86.4.13 virtual MarkovChain::State eos::proposal_functions::GlobalLocal::mode () const [virtual]

The state with the highest posterior of all components with parameter values and correct log posterior.

6.86.4.14 virtual void eos::proposal_functions::GlobalLocal::propose (MarkovChain::State & *x*, const MarkovChain::State & *y*, gsl_rng * *rng*) const [virtual]

Obtain from the density a proposal x given y.

Implements eos::MarkovChain::ProposalFunction.

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.87 eos::HardScattering Struct Reference

```
#include <hard-scattering.hh>
```

Static Public Member Functions

- static complex< double > I1 (const double &s, const double &u, const double &m_q, const double &m_B)

6.87.1 Member Function Documentation

6.87.1.1 static complex<double> eos::HardScattering::I1 (const double & *s*, const double & *u*, const double & *m_q*, const double & *m_B*) [static]

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/hard-scattering.hh

6.88 eos::HDF5Error Class Reference

```
#include <hdf5.hh>
```

Inherits eos::Exception.

Public Member Functions

- HDF5Error (const std::string &message)

6.88.1 Detailed Description

HDF5Error is parent to all exceptions thrown by entities in namespace eos::hdf5, e.g when creating, opening or accessing hdf5 file data.

6.88.2 Constructor & Destructor Documentation

6.88.2.1 eos::HDF5Error::HDF5Error (const std::string & *message*)

Constructor.

Parameters:

message The error message.

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.89 eos::HierarchicalClustering Class Reference

`#include <hierarchical-clustering.hh>`

Inherits PrivateImplementationPattern< HierarchicalClustering >.

Classes

- class Component
- class Config

Public Types

- typedef std::vector< HierarchicalClustering::Component > MixtureDensity
- typedef WrappedForwardIterator< IteratorTag, Component > ComponentIterator

  Loop over input components.

- typedef WrappedForwardIterator< IteratorTag, Component > ClusterIterator

  Loop over final clusters (determined during clustering).

- typedef WrappedForwardIterator< IteratorTag, unsigned > MapIterator

  To which cluster is a component mapped?

Public Member Functions

- void add (const Component &component)
- void initial_guess (const MixtureDensity &density)
- void run ()
- ComponentIterator begin_components () const
- ComponentIterator end_components () const
- ClusterIterator begin_clusters () const
- ClusterIterator end_clusters () const
- MapIterator begin_map () const
- MapIterator end_map () const

Basic Functions

- HierarchicalClustering (const HierarchicalClustering::Config &config)
- ~HierarchicalClustering ()

  Destructor.

6.89.1 Detailed Description

Implement the hierarchical clustering explained in [GR2004].

A mixture density is determined as a reduced representation of input components.

6.89.2 Member Typedef Documentation

6.89.2.1 typedef WrappedForwardIterator<IteratorTag, Component> eos::HierarchicalClustering::ClusterIterator

Loop over final clusters (determined during clustering).

6.89.2.2 typedef WrappedForwardIterator<IteratorTag, Component> eos::HierarchicalClustering::ComponentIterator

Loop over input components.

6.89.2.3 typedef WrappedForwardIterator<IteratorTag, unsigned> eos::HierarchicalClustering::MapIterator

To which cluster is a component mapped?

6.89.2.4 typedef std::vector<HierarchicalClustering::Component> eos::HierarchicalClustering::MixtureDensity

6.89.3 Constructor & Destructor Documentation

6.89.3.1 eos::HierarchicalClustering::HierarchicalClustering (const HierarchicalClustering::Config & *config*)

Constructor.

Parameters:

config The configuration of the samples.

6.89.3.2 eos::HierarchicalClustering::~HierarchicalClustering ()

Destructor.

6.89.4 Member Function Documentation

6.89.4.1 void eos::HierarchicalClustering::add (const Component & *component*)

Add an input component

6.89.4.2 ClusterIterator eos::HierarchicalClustering::begin_clusters () const

6.89.4.3 ComponentIterator eos::HierarchicalClustering::begin_components () const

6.89.4.4 MapIterator eos::HierarchicalClustering::begin_map () const

6.89.4.5 ClusterIterator eos::HierarchicalClustering::end_clusters () const

6.89.4.6 ComponentIterator eos::HierarchicalClustering::end_components () const

6.89.4.7 MapIterator eos::HierarchicalClustering::end_map () const

6.89.4.8 void eos::HierarchicalClustering::initial_guess (const MixtureDensity & *density*)

Add an initial guess for the clusters to be determined

Parameters:

density

6.89.4.9 void eos::HierarchicalClustering::run ()

Perform the clustering.

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

- /home/alexshires/EOS/eos-next/eos/utils/hierarchical-clustering.hh

6.90 eos::Histogram< 1 > Class Template Reference

`#include <histogram.hh>`

Inherits PrivateImplementationPattern< Histogram< 1 > >.

Classes

- struct Bin

Public Member Functions

Insertion and Retrieval

Inserts a bin into the histogram.

- void insert (const Bin &bin)

  Inserts a value into the histogram's matching bin.

- void insert (const double &value)

  Inserts a value into the histogram's matching bin.

- unsigned entries () const

  Returns the number of entries in the histogram.

Iteration

- typedef WrappedForwardIterator< ConstIteratorTag, const Bin > ConstIterator
- ConstIterator begin () const
- ConstIterator end () const

Basic Functions

- Histogram ()
- ~Histogram ()

  Destructor.

- static Histogram WithEqualBinning (const double &start, const double &end, const unsigned &count)

template<> class eos::Histogram< 1 >

6.90.1 Member Typedef Documentation

6.90.1.1 typedef WrappedForwardIterator<ConstIteratorTag, const Bin> eos::Histogram< 1 >::ConstIterator

6.90.2 Constructor & Destructor Documentation

6.90.2.1 eos::Histogram< 1 >::Histogram ()

Constructor.

Creates an empty Histogram, i.e. a Histogram without any bins.

6.90.2.2 eos::Histogram< 1 >::~Histogram ()

Destructor.

6.90.3 Member Function Documentation

6.90.3.1 ConstIterator eos::Histogram< 1 >::begin () const

6.90.3.2 ConstIterator eos::Histogram< 1 >::end () const

6.90.3.3 unsigned eos::Histogram< 1 >::entries () const

Returns the number of entries in the histogram.

6.90.3.4 void eos::Histogram< 1 >::insert (const double & *value*)

Inserts a value into the histogram's matching bin.

6.90.3.5 void eos::Histogram< 1 >::insert (const Bin & *bin*)

Inserts a value into the histogram's matching bin.

6.90.3.6 static Histogram eos::Histogram< 1 >::WithEqualBinning (const double & *start*, const double & *end*, const unsigned & *count*) `[static]`

Named constructor.

Creates a Histogram with pre-existing bins of identical width.

Parameters:

start Left-most value that shall be covered by the Histogram.

end Right-most value that shall be covered by the Histogram.

count Number of Bins in the Histogram.

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

- /home/alexshires/EOS/eos-next/eos/utils/histogram.hh

6.91 eos::Histogram< 2 > Class Template Reference

`#include <histogram.hh>`

Inherits PrivateImplementationPattern< Histogram< 2 > >.

Classes

- struct Bin

Public Member Functions

Insertion and Retrieval

Inserts a bin into the histogram.

- void insert (const Bin &bin)

  Returns the number of entries in the histogram.

- unsigned entries () const

  Returns the number of entries in the histogram.

Iteration

- typedef WrappedForwardIterator< IteratorTag, Bin > Iterator
- Iterator begin ()
- Iterator find (const std::array< double, 2 > &coordinates)
- Iterator end ()

Constant Iteration

- typedef WrappedForwardIterator< ConstIteratorTag, const Bin > ConstIterator
- ConstIterator cbegin () const
- ConstIterator cend () const

Basic Functions

- Histogram ()
- ~Histogram ()

  Destructor.

- static Histogram WithEqualBinning (const std::array< double, 2 > &start, const std::array< double, 2 > &end, const std::array< unsigned, 2 > &count)

template<> class eos::Histogram< 2 >

6.91.1 Member Typedef Documentation

6.91.1.1 typedef WrappedForwardIterator<ConstIteratorTag, const Bin> eos::Histogram< 2 >::ConstIterator

6.91.1.2 typedef WrappedForwardIterator<IteratorTag, Bin> eos::Histogram< 2 >::Iterator

6.91.2 Constructor & Destructor Documentation

6.91.2.1 eos::Histogram< 2 >::Histogram ()

Constructor.

Creates an empty Histogram, i.e. a Histogram without any bins.

6.91.2.2 eos::Histogram< 2 >::~Histogram ()

Destructor.

6.91.3 Member Function Documentation

6.91.3.1 Iterator eos::Histogram< 2 >::begin ()

6.91.3.2 ConstIterator eos::Histogram< 2 >::cbegin () const

6.91.3.3 ConstIterator eos::Histogram< 2 >::cend () const

6.91.3.4 Iterator eos::Histogram< 2 >::end ()

6.91.3.5 unsigned eos::Histogram< 2 >::entries () const

Returns the number of entries in the histogram.

6.91.3.6 Iterator eos::Histogram< 2 >::find (const std::array< double, 2 > & *coordinates*)

6.91.3.7 void eos::Histogram< 2 >::insert (const Bin & *bin*)

Returns the number of entries in the histogram.

6.91.3.8 static Histogram eos::Histogram< 2 >::WithEqualBinning (const std::array< double, 2 > & *start*, const std::array< double, 2 > & *end*, const std::array< unsigned, 2 > & *count*) `[static]`

Named constructor.

Creates a Histogram with pre-existing bins of identical width.

Parameters:

start Left-most value for each dimension that shall be covered by the Histogram.

end Right-most value for each dimension that shall be covered by the Histogram.

count Number of Bins for each dimension.

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

- /home/alexshires/EOS/eos-next/eos/utils/histogram.hh

6.92 eos::MarkovChain::History Struct Reference

```
#include <markov_chain.hh>
```

Public Member Functions

- const MarkovChain::State & local_mode (const MarkovChain::State::Iterator &begin, const MarkovChain::State::Iterator &end) const
- void mean_and_variance (const MarkovChain::State::Iterator &begin, const MarkovChain::State::Iterator &end, std::vector< double > &mean, std::vector< double > &variance) const
- void mean_and_covariance (const MarkovChain::State::Iterator &begin, const MarkovChain::State::Iterator &end, std::vector< double > &mean, std::vector< double > &variance) const

Public Attributes

- bool keep

  flag: if false => don't store numbers

- std::vector< MarkovChain::State > states

  All states.

6.92.1 Detailed Description

Holds the entire history of a run of a MarkovChain

6.92.2 Member Function Documentation

6.92.2.1 const MarkovChain::State& eos::MarkovChain::History::local_mode (const MarkovChain::State::Iterator & *begin*, const MarkovChain::State::Iterator & *end*) const

Return state with highest posterior probability in selected range

6.92.2.2 void eos::MarkovChain::History::mean_and_covariance (const MarkovChain::State::Iterator & *begin*, const MarkovChain::State::Iterator & *end*, std::vector< double > & *mean*, std::vector< double > & *variance*) const

Compute the mean vector and the sample covariance in the given range of the chain.

Parameters:

begin

end

mean

variance

6.92.2.3 void eos::MarkovChain::History::mean_and_variance (const MarkovChain::State::Iterator & *begin*, const MarkovChain::State::Iterator & *end*, std::vector< double > & *mean*, std::vector< double > & *variance*) const

Compute mean and variance of the states' parameters between begin and end using Welford's method for all parameters. Results are stored in the vectors.

For details, check out http://www.johndcook.com/standard_deviation.html .

6.92.3 Member Data Documentation

6.92.3.1 bool eos::MarkovChain::History::keep

flag: if false => don't store numbers

6.92.3.2 std::vector<MarkovChain::State> eos::MarkovChain::History::states

All states.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh

6.93 eos::HLMW2005 Struct Reference

```
#include <inclusive-b-to-s-dilepton.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-dilepton.hh

6.94 eos::MarkovChain::HyperParameter Struct Reference

```
#include <markov_chain.hh>
```

Public Attributes

- unsigned component

  Indicating the current component.

6.94.1 Member Data Documentation

6.94.1.1 unsigned eos::MarkovChain::HyperParameter::component

Indicating the current component.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh

6.95 eos::InstantiationPolicy< T_, NonCopyable > Class Template Reference

```
#include <instantiation_policy.hh>
```

Public Member Functions

- InstantiationPolicy ()
  Default constructor.

template<typename T_> class eos::InstantiationPolicy< T_, NonCopyable >

6.95.1 Constructor & Destructor Documentation

6.95.1.1 template<typename T_ > eos::InstantiationPolicy< T_, NonCopyable >::InstantiationPolicy () [inline]

Default constructor.

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

- /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy.hh

6.96 eos::InstantiationPolicy< T_, NonInstantiable > Class Template Reference

```
#include <instantiation_policy.hh>
```

template<typename T_> class eos::InstantiationPolicy< T_, NonInstantiable >

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

- /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy.hh

6.97 eos::InstantiationPolicy< T_, Singleton > Class Template Reference

```
#include <instantiation_policy.hh>
```

Static Public Member Functions

- static T_ * instance ()

  Returns the instance.

Protected Member Functions

- InstantiationPolicy ()

  Default constructor.

Friends

- class DeleteOnDestruction

template<typename T_> class eos::InstantiationPolicy< T_, Singleton >

6.97.1 Constructor & Destructor Documentation

6.97.1.1 template<typename T_ > eos::InstantiationPolicy< T_, Singleton >::InstantiationPolicy () [inline, protected]

Default constructor.

6.97.2 Member Function Documentation

6.97.2.1 template<typename T_ > T_ * eos::InstantiationPolicy< T_, Singleton >::instance () [inline, static]

Returns the instance.

6.97.3 Friends And Related Function Documentation

6.97.3.1 template<typename T_ > friend class DeleteOnDestruction [friend]

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

- /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy.hh
- /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy-impl.hh

6.98 eos::InternalError Struct Reference

```
#include <exception.hh>
```

Inherits eos::Exception.

Public Member Functions

- InternalError (const std::string &message) throw ()

6.98.1 Constructor & Destructor Documentation

6.98.1.1 eos::InternalError::InternalError (const std::string & *message*) throw ()

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

- /home/alexshires/EOS/eos-next/eos/utils/exception.hh

6.99 eos::Kinematics Class Reference

`#include <kinematic.hh>`

Inherits PrivateImplementationPattern< Kinematics >.

Public Member Functions

Basic Functions

Constructor.

- Kinematics ()
- Kinematics (const std::initializer_list< std::pair< std::string, double >> &variables)
- ~Kinematics ()

  Destructor.

- Kinematics clone () const
- bool operator== (const Kinematics &rhs) const

  Equality comparison operator.

- bool operator!= (const Kinematics &rhs) const

  Inequality comparison operator.

Variable access

- KinematicVariable declare (const std::string &name, const double &value=0.0)
- void set (const std::string &variable, const double &value)
- KinematicVariable operator[ ] (const std::string &variable) const

Output

- std::string as_string () const

6.99.1 Detailed Description

Kinematics keeps the set of all kinematic variables for any Observable.

Access to any KinematicVariable or their values is coherent, i.e., changes to a KinematicVariable object will propagate to every other object with the same parent Kinematics and which handle the same variable by name.

6.99.2 Constructor & Destructor Documentation

6.99.2.1 eos::Kinematics::Kinematics ()

Constructor.

Create an instance of Kinematics with a given set of initial kinematic variables.

Parameters:

variables The set of initial kinematics variables from which this object shall be constructed.

6.99.2.2 eos::Kinematics::Kinematics (const std::initializer_list< std::pair< std::string, double >> & *variables*)

Constructor.

Create an instance of Kinematics with a given set of initial kinematic variables.

Parameters:

variables The set of initial kinematics variables from which this object shall be constructed.

6.99.2.3 eos::Kinematics::~Kinematics ()

Destructor.

6.99.3 Member Function Documentation

6.99.3.1 std::string eos::Kinematics::as_string () const

Retrieve a string representation of the set of kinematic variables.

6.99.3.2 Kinematics eos::Kinematics::clone () const

Create an independent copy of this Kinematics object.

6.99.3.3 KinematicVariable eos::Kinematics::declare (const std::string & *name*, const double & *value* = 0.0)

Declare a new kinematic variable.

Parameters:

name Name of the new variable to be declared.

value (Optional) value for the new variable.

6.99.3.4 bool eos::Kinematics::operator!= (const Kinematics & *rhs*) const

Inequality comparison operator.

6.99.3.5 bool eos::Kinematics::operator== (const Kinematics & *rhs*) const

Equality comparison operator.

6.99.3.6 KinematicVariable eos::Kinematics::operator[ ] (const std::string & *variable*) const

Retrieve a variable's KinematicVariable object by name.

Parameters:

name The name of the KinematicVariable that shall be retrieved.

6.99.3.7 void eos::Kinematics::set (const std::string & *variable*, const double & *value*)

Set a kinematic variable's numeric value.

Parameters:

name The name of the variable whose numeric value shall be changed.

value The variable's new numeric value.

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

- /home/alexshires/EOS/eos-next/eos/utils/kinematic.hh

6.100 eos::KinematicVariable Class Reference

```
#include <kinematic.hh>
```

Public Member Functions

Access & Modification of the Numeric Value

Cast a KinematicVariable's numeric value to a double.

- operator double () const
  Retrieve a KinematicVariable's numeric value.
- double operator() () const
  Retrieve a KinematicVariable's numeric value.
- const KinematicVariable & operator= (const double &)
  Set a KinematicVariable's numeric value.

Friends

- class Kinematics
- struct Implementation< Kinematics >

Basic Functions

- ~KinematicVariable ()

6.100.1 Detailed Description

KinematicVariable is the class that holds all information of one of KinematicVariables' parameters.

6.100.2 Constructor & Destructor Documentation

6.100.2.1 eos::KinematicVariable::~KinematicVariable ()

6.100.3 Member Function Documentation

6.100.3.1 eos::KinematicVariable::operator double () const

Retrieve a KinematicVariable's numeric value.

6.100.3.2 double eos::KinematicVariable::operator() () const

Retrieve a KinematicVariable's numeric value.

6.100.3.3 const KinematicVariable& eos::KinematicVariable::operator= (const double &)

Set a KinematicVariable's numeric value.

6.100.4 Friends And Related Function Documentation

6.100.4.1 friend struct Implementation< Kinematics > `[friend]`

6.100.4.2 friend class Kinematics `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/kinematic.hh

6.101 eos::KMPW2010FormFactors< PToP > Class Template Reference

`#include <mesonic-impl.hh>`

Inherits eos::FormFactors< PToP >.

Public Member Functions

- KMPW2010FormFactors (const Parameters &p, const Options &)
- virtual double f_p (const double &s) const
- virtual double f_0 (const double &s) const
- virtual double f_t (const double &s) const

Static Public Member Functions

- static FormFactors< PToP > * make (const Parameters &parameters, unsigned)

template<> class eos::KMPW2010FormFactors< PToP >

6.101.1 Constructor & Destructor Documentation

6.101.1.1 eos::KMPW2010FormFactors< PToP >::KMPW2010FormFactors (const Parameters & *p*, const Options &) `[inline]`

6.101.2 Member Function Documentation

6.101.2.1 virtual double eos::KMPW2010FormFactors< PToP >::f_0 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.101.2.2 virtual double eos::KMPW2010FormFactors< PToP >::f_p (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.101.2.3 virtual double eos::KMPW2010FormFactors< PToP >::f_t (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToP >.

6.101.2.4 static FormFactors<PToP>* eos::KMPW2010FormFactors< PToP >::make (const Parameters & *parameters*, unsigned) `[inline, static]`

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.102 eos::KMPW2010FormFactors< PToV > Class Template Reference

`#include <mesonic-impl.hh>`

Inherits eos::FormFactors< PToV >.

Public Member Functions

- KMPW2010FormFactors (const Parameters &p, const Options &)
- virtual double v (const double &s) const
- virtual double a_0 (const double &s) const
- virtual double a_1 (const double &s) const
- virtual double a_2 (const double &s) const
- virtual double a_12 (const double &s) const

Static Public Member Functions

- static FormFactors< PToV > * make (const Parameters &parameters, unsigned)

template<> class eos::KMPW2010FormFactors< PToV >

6.102.1 Constructor & Destructor Documentation

6.102.1.1 eos::KMPW2010FormFactors< PToV >::KMPW2010FormFactors (const Parameters & *p*, const Options &) `[inline]`

6.102.2 Member Function Documentation

6.102.2.1 virtual double eos::KMPW2010FormFactors< PToV >::a_0 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.102.2.2 virtual double eos::KMPW2010FormFactors< PToV >::a_1 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.102.2.3 virtual double eos::KMPW2010FormFactors< PToV >::a_12 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.102.2.4 virtual double eos::KMPW2010FormFactors< PToV >::a_2 (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

6.102.2.5 static FormFactors<PToV>* eos::KMPW2010FormFactors< PToV >::make (const Parameters & *parameters*, unsigned) `[inline, static]`

6.102.2.6 virtual double eos::KMPW2010FormFactors< PToV >::v (const double & *s*) const `[inline, virtual]`

Implements eos::FormFactors< PToV >.

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh

6.103 eos::LambdaBToLambdaDilepton< LargeRecoil > Class Template Reference

```
#include <lambda-b-to-lambda-dilepton.hh>
```

Inherits eos::ParameterUser, and PrivateImplementationPattern< LambdaBToLambdaDilepton< LargeRecoil > >.

Public Member Functions

- LambdaBToLambdaDilepton (const Parameters &, const Options &)
- ~LambdaBToLambdaDilepton ()
- double differential_branching_ratio (const double &s) const
- double differential_a_fb_leptonic (const double &s) const
- double differential_a_fb_hadronic (const double &s) const
- double differential_a_fb_combined (const double &s) const
- double differential_fzero (const double &s) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- double integrated_a_fb_leptonic (const double &s_min, const double &s_max) const
- double integrated_a_fb_hadronic (const double &s_min, const double &s_max) const
- double integrated_a_fb_combined (const double &s_min, const double &s_max) const
- double integrated_fzero (const double &s_min, const double &s_max) const

template<> class eos::LambdaBToLambdaDilepton< LargeRecoil >

6.103.1 Constructor & Destructor Documentation

6.103.1.1 eos::LambdaBToLambdaDilepton< LargeRecoil >::LambdaBToLambdaDilepton (const Parameters &, const Options &)

6.103.1.2 eos::LambdaBToLambdaDilepton< LargeRecoil >::~LambdaBToLambdaDilepton ()

6.103.2 Member Function Documentation

6.103.2.1 double eos::LambdaBToLambdaDilepton< LargeRecoil >::differential_a_fb_combined (const double & *s*) const

6.103.2.2 double eos::LambdaBToLambdaDilepton< LargeRecoil >::differential_a_fb_hadronic (const double & *s*) const

6.103.2.3 double eos::LambdaBToLambdaDilepton< LargeRecoil >::differential_a_fb_leptonic (const double & *s*) const

6.103.2.4 double eos::LambdaBToLambdaDilepton< LargeRecoil >::differential_branching_-ratio (const double & *s*) const

6.103.2.5 double eos::LambdaBToLambdaDilepton< LargeRecoil >::differential_fzero (const double & *s*) const

6.103.2.6 double eos::LambdaBToLambdaDilepton< LargeRecoil >::integrated_a_fb_combined (const double & *s_min*, const double & *s_max*) const

6.103.2.7 double eos::LambdaBToLambdaDilepton< LargeRecoil >::integrated_a_fb_hadronic (const double & *s_min*, const double & *s_max*) const

6.103.2.8 double eos::LambdaBToLambdaDilepton< LargeRecoil >::integrated_a_fb_leptonic (const double & *s_min*, const double & *s_max*) const

6.103.2.9 double eos::LambdaBToLambdaDilepton< LargeRecoil >::integrated_branching_ratio (const double & *s_min*, const double & *s_max*) const

6.103.2.10 double eos::LambdaBToLambdaDilepton< LargeRecoil >::integrated_fzero (const double & *s_min*, const double & *s_max*) const

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/lambda-b-to-lambda-dilepton.hh

6.104 eos::LambdaBToLambdaDilepton< LowRecoil > Class Template Reference

```
#include <lambda-b-to-lambda-dilepton.hh>
```

Inherits eos::ParameterUser, and PrivateImplementationPattern< LambdaBToLambdaDilepton< LowRecoil > >.

Public Member Functions

- LambdaBToLambdaDilepton (const Parameters &, const Options &)
- ~LambdaBToLambdaDilepton ()
- double differential_branching_ratio (const double &s) const
- double differential_a_fb_leptonic (const double &s) const
- double differential_a_fb_hadronic (const double &s) const
- double differential_a_fb_combined (const double &s) const
- double differential_fzero (const double &s) const
- double integrated_branching_ratio (const double &s_min, const double &s_max) const
- double integrated_a_fb_leptonic (const double &s_min, const double &s_max) const
- double integrated_a_fb_hadronic (const double &s_min, const double &s_max) const
- double integrated_a_fb_combined (const double &s_min, const double &s_max) const
- double integrated_fzero (const double &s_min, const double &s_max) const

template<> class eos::LambdaBToLambdaDilepton< LowRecoil >

6.104.1 Constructor & Destructor Documentation

6.104.1.1 eos::LambdaBToLambdaDilepton< LowRecoil >::LambdaBToLambdaDilepton (const Parameters &, const Options &)

6.104.1.2 eos::LambdaBToLambdaDilepton< LowRecoil >::~LambdaBToLambdaDilepton ()

6.104.2 Member Function Documentation

6.104.2.1 double eos::LambdaBToLambdaDilepton< LowRecoil >::differential_a_fb_combined (const double & *s*) const

6.104.2.2 double eos::LambdaBToLambdaDilepton< LowRecoil >::differential_a_fb_hadronic (const double & *s*) const

6.104.2.3 double eos::LambdaBToLambdaDilepton< LowRecoil >::differential_a_fb_leptonic (const double & *s*) const

6.104.2.4 double eos::LambdaBToLambdaDilepton< LowRecoil >::differential_branching_ratio (const double & *s*) const

6.104.2.5 double eos::LambdaBToLambdaDilepton< LowRecoil >::differential_fzero (const double & *s*) const

6.104.2.6 double eos::LambdaBToLambdaDilepton< LowRecoil >::integrated_a_fb_combined (const double & *s_min*, const double & *s_max*) const

6.104.2.7 double eos::LambdaBToLambdaDilepton< LowRecoil >::integrated_a_fb_hadronic (const double & *s_min*, const double & *s_max*) const

6.104.2.8 double eos::LambdaBToLambdaDilepton< LowRecoil >::integrated_a_fb_leptonic (const double & *s_min*, const double & *s_max*) const

6.104.2.9 double eos::LambdaBToLambdaDilepton< LowRecoil >::integrated_branching_ratio (const double & *s_min*, const double & *s_max*) const

6.104.2.10 double eos::LambdaBToLambdaDilepton< LowRecoil >::integrated_fzero (const double & *s_min*, const double & *s_max*) const

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/lambda-b-to-lambda-dilepton.hh

6.105 eos::LargeRecoil Struct Reference

```
#include <decays.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/decays.hh

6.106 eos::LCDA Struct Reference

```
#include <lcda.hh>
```

Static Public Member Functions

- static double evolve_gegenbauer_moment (const double &a_n_0, const unsigned &n, const double &eta, const QCD::BetaFunction &beta)

6.106.1 Detailed Description

Groups all functions related to LightCone Distribution Amplitudes (LCDAs).

6.106.2 Member Function Documentation

6.106.2.1 static double eos::LCDA::evolve_gegenbauer_moment (const double & *a_n_0*, const unsigned & *n*, const double & *eta*, const QCD::BetaFunction & *beta*) [static]

Evolve the nth Gegenbauer moment a_n_0 from a scale mu_0 to a scale mu according to the LL RGE.

Calculation according to [BBL2006].

Parameters:

a_n_0 The Gegenbauer moment at the scale mu_0.

n The index of the Gegenbauer moment.

eta The ratio alpha_s(mu)/alpha_s(mu_0).

beta The coefficients of the QCD beta function.

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

- /home/alexshires/EOS/eos-next/eos/utils/lcda.hh

6.107 eos::Lock Class Reference

```
#include <lock.hh>
```

Public Member Functions

- Lock (Mutex &)

  (Explicit) constructor.

- ~Lock ()

  Destructor.

6.107.1 Constructor & Destructor Documentation

6.107.1.1 eos::Lock::Lock (Mutex &) [explicit]

(Explicit) constructor.

6.107.1.2 eos::Lock::~Lock ()

Destructor.

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

- /home/alexshires/EOS/eos-next/eos/utils/lock.hh

6.108 eos::Log Class Reference

`#include <log.hh>`

Inherits PrivateImplementationPattern< Log >.

Public Member Functions

Access

- const LogLevel & get_log_level () const
  Get the current log level.

- void set_log_level (const LogLevel &)
- void set_log_stream (std::ostream *)
- void set_program_name (const std::string &)
- LogMessageHandler message (const std::string &id, const LogLevel &log_level) __attribute__-((warn_unused_result))

Friends

- class LogMessageHandler
- class InstantiationPolicy< Log, Singleton >

Basic Functions

Destructor.

- ~Log ()

6.108.1 Detailed Description

Facility to emit messages to the user, with a filter according to interest and urgency of the messages at hand.

6.108.2 Constructor & Destructor Documentation

6.108.2.1 eos::Log::~Log ()

6.108.3 Member Function Documentation

6.108.3.1 const LogLevel& eos::Log::get_log_level () const

Get the current log level.

6.108.3.2 LogMessageHandler eos::Log::message (const std::string & *id*, const LogLevel & *log_level*)

Return a stream-like object to which message parts can be appended via its overloaded operator<<. The message will be completed upon destruction of the return value.

6.108.3.3 void eos::Log::set_log_level (const LogLevel &)

Set the log level.

6.108.3.4 void eos::Log::set_log_stream (std::ostream *)

Set the output stream.

6.108.3.5 void eos::Log::set_program_name (const std::string &)

Set the program's name.

6.108.4 Friends And Related Function Documentation

6.108.4.1 friend class InstantiationPolicy< Log, Singleton > [friend]

6.108.4.2 friend class LogMessageHandler [friend]

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

- /home/alexshires/EOS/eos-next/eos/utils/log.hh

6.109 eos::LogLikelihood Class Reference

`#include <log_likelihood.hh>`

Inherits PrivateImplementationPattern< LogLikelihood >.

Public Member Functions

- std::pair< double, double > bootstrap_p_value (const unsigned &datasets)
- LogLikelihood clone () const
- unsigned number_of_observations () const
- Parameters parameters () const
- ObservableCache observable_cache () const
- double operator() ()
- double operator() (const Parameter::Id &id)
- void reset ()

Basic Functions

- LogLikelihood (const Parameters &parameters)
- ~LogLikelihood ()

Access and evaluation

- void add (const ObservablePtr &observable, const double &min, const double &central, const double &max, const unsigned &number_of_observations=1u)
- void add (const Constraint &constraint)

Iteration and Access

- typedef WrappedForwardIterator< ConstraintIteratorTag, Constraint > ConstraintIterator

  Iterator to the first constraint.

- ConstraintIterator begin () const

  Iterator to the first constraint.

- ConstraintIterator end () const

  Iterator pointing past the last constraint.

6.109.1 Detailed Description

LogLikelihood handles a set of ObservablePtr with associated measurement data.

Access to any LogLikelihood is coherent, i.e., changes to one object will propagate to every other object copy. To create an independent instance, use clone().

6.109.2 Member Typedef Documentation

6.109.2.1 typedef WrappedForwardIterator<ConstraintIteratorTag, Constraint> eos::LogLikelihood::ConstraintIterator

Iterator to the first constraint.

6.109.3 Constructor & Destructor Documentation

6.109.3.1 eos::LogLikelihood::LogLikelihood (const Parameters & *parameters*)

Constructor.

Parameters:

parameters The Parameters object to which all further ObservablePtr objects must be bound.

6.109.3.2 eos::LogLikelihood::~LogLikelihood ()

Destructor.

6.109.4 Member Function Documentation

6.109.4.1 void eos::LogLikelihood::add (const Constraint & *constraint*)

Add one of the libraries experimental constraints.

Parameters:

constraint The experimental constraint, cf. Constraint::make

6.109.4.2 void eos::LogLikelihood::add (const ObservablePtr & *observable*, const double & *min*, const double & *central*, const double & *max*, const unsigned & *number_of_observations* = `1u`)

Add an observable and its associated measurement.

Parameters:

observable The Observable that shall be added to the calculation of the likelihood.

min The lower bound on the measurement.

central The central value of the measurement.

max The upper bound on the measurement.

number_of_observations The number of observations associated with the measurement. Defaults to 1.

6.109.4.3 ConstraintIterator eos::LogLikelihood::begin () const

Iterator to the first constraint.

6.109.4.4 std::pair<double, double> eos::LogLikelihood::bootstrap_p_value (const unsigned & *datasets*)

Calculate a p-value based on the ^2 test statistic for the current setting of the parameters.

Note:

The p-value is _not_ corrected for degrees of freedom.

Parameters:

datasets The number of simulated data sets

Returns:

<p-value, uncertainty>, where the uncertainty is estimated from the standard posterior for a Bernoulli experiment.

6.109.4.5 LogLikelihood eos::LogLikelihood::clone () const

Create an independent instance of this LogLikelihood that uses the same set of observables and measurements.

6.109.4.6 ConstraintIterator eos::LogLikelihood::end () const

Iterator pointing past the last constraint.

6.109.4.7 unsigned eos::LogLikelihood::number_of_observations () const

The number of independent observations used in the likelihood.

Note:

This may differ from the number of observables in case two experiments reported their results on the same observable.

6.109.4.8 ObservableCache eos::LogLikelihood::observable_cache () const

Retrieve the cache of observables associated with this LogLikelihood.

6.109.4.9 double eos::LogLikelihood::operator() (const Parameter::Id & *id*)

Evaluate the likelihood, but recalculate only those observables which depend on the parameter with the given id

Warning:

Do not use this call before __all__ observables have been added to the likelihood! Since the dependence is checked only the first time a parameter id is encountered, adding an observable later will not establish the link between the id and the observable, leading to incorrect values of the likelihood

6.109.4.10 double eos::LogLikelihood::operator() ()

Evaluate the log likelihood, i.e., return

$$\log \mathcal{L} = \log P(D|\vec{\theta}, M) = -\frac{\chi^2}{2} + C$$

.

Note:

: all observables are recalculated

6.109.4.11 Parameters eos::LogLikelihood::parameters () const

Retrieve the underlying Parameters object.

6.109.4.12 void eos::LogLikelihood::reset ()

Reload previous observable values.

Note:

this is only useful in conjunction with operator()(const Parameter& par) and the Markov Chain sampler

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

- /home/alexshires/EOS/eos-next/eos/utils/log_likelihood.hh

6.110 eos::LogLikelihoodBlock Class Reference

```
#include <log_likelihood.hh>
```

Public Member Functions

- virtual ~LogLikelihoodBlock ()=0

  Destructor.

- virtual std::string as_string () const =0
- virtual LogLikelihoodBlockPtr clone (ObservableCache cache) const =0

  Clone this block.

- virtual double evaluate () const =0

  Compute the logarithm of the likelihood for this block.

- virtual unsigned number_of_observations () const =0

  The number of experimental observations (not observables!) used in this block.

- virtual void prepare_sampling ()
- virtual double sample (gsl_rng *rng) const =0
- virtual double significance () const =0

Static Public Member Functions

- static LogLikelihoodBlockPtr Gaussian (ObservableCache cache, const ObservablePtr &observable, const double &min, const double &central, const double &max, const unsigned &number_of_observations=1u)
- static LogLikelihoodBlockPtr LogGamma (ObservableCache cache, const ObservablePtr &observable, const double &min, const double &central, const double &max, const unsigned &number_of_observations=1u)
- static LogLikelihoodBlockPtr LogGamma (ObservableCache cache, const ObservablePtr &observable, const double &min, const double &central, const double &max, const double &lambda, const double &alpha, const unsigned &number_of_observations=1u)
- static LogLikelihoodBlockPtr AmorosoLimit (ObservableCache cache, const ObservablePtr &observable, const double &physical_limit, const double &upper_limit_90, const double &upper_limit_95, const double &theta, const double &alpha, const unsigned &number_of_observations=1u)
- static LogLikelihoodBlockPtr AmorosoMode (ObservableCache cache, const ObservablePtr &observable, const double &physical_limit, const double &mode, const double &upper_limit_90, const double &upper_limit_95, const double &theta, const double &alpha, const double &beta, const unsigned &number_of_observations=1u)
- static LogLikelihoodBlockPtr Amoroso (ObservableCache cache, const ObservablePtr &observable, const double &physical_limit, const double &upper_limit_10, const double &upper_limit_50, const double &upper_limit_90, const double &theta, const double &alpha, const double &beta, const unsigned &number_of_observations=1u)
- static LogLikelihoodBlockPtr Amoroso (ObservableCache cache, const ObservablePtr &observable, const double &physical_limit, const double &theta, const double &alpha, const double &beta, const unsigned &number_of_observations=1u)
- static LogLikelihoodBlockPtr Mixture (const std::vector< LogLikelihoodBlockPtr > &components, const std::vector< double > &weights)

- template<std::size_t n_>
  static LogLikelihoodBlockPtr MultivariateGaussian (ObservableCache cache, const std::array< ObservablePtr, n_ > &observables, const std::array< double, n_ > &mean, const std::array< std::array< double, n_ >, n_ > &covariance, const unsigned &number_of_observations=n_)
- template<std::size_t n_>
  static LogLikelihoodBlockPtr MultivariateGaussian (ObservableCache cache, const std::array< ObservablePtr, n_ > &observables, const std::array< double, n_ > &mean, const std::array< double, n_ > &variances, const std::array< std::array< double, n_ >, n_ > &correlation, const unsigned &number_of_observations=n_)

6.110.1 Detailed Description

LogLikelihoodBlock models the logarithm of the likelihood for a given number of correlated observables which are independent of all other observables such that total likelihood is just the product of the independent blocks.

Access to any LogLikelihoodBlock is coherent, i.e., changes to one object will propagate to every other object copy. To create an independent instance, use clone().

6.110.2 Constructor & Destructor Documentation

6.110.2.1 virtual eos::LogLikelihoodBlock::~LogLikelihoodBlock () `[pure virtual]`

Destructor.

6.110.3 Member Function Documentation

6.110.3.1 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::Amoroso (ObservableCache *cache*, const ObservablePtr & *observable*, const double & *physical_limit*, const double & *theta*, const double & *alpha*, const double & *beta*, const unsigned & *number_of_observations* = `1u`) `[static]`

A likelihood contribution representing an upper limit on a quantity x.

Internally, it is represented by an Amoroso distribution [C2004] with location parameter a set to the physical limit, scale parameter and shape parameters , supplied by the user.

Note:

The mode of the distribution is typically not at the physical limit.

For example, consider a yet unobserved branching ratio, which has to be non-negative, x>= 0.

Parameters:

cache The Observable cache from which we draw the predictions.

observable The Observable whose distribution we model.

mode The lower, physical limit. A branching ratio has to be >= 0.

theta scale parameter

alpha 1st shape parameter

beta 2nd shape parameter

Returns:

6.110.3.2 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::Amoroso (ObservableCache *cache*, const ObservablePtr & *observable*, const double & *physical_limit*, const double & *upper_limit_10*, const double & *upper_limit_50*, const double & *upper_limit_90*, const double & *theta*, const double & *alpha*, const double & *beta*, const unsigned & *number_of_observations* = `1u`) `[static]`

A likelihood contribution representing an upper limit on a quantity x.

Internally, it is represented by an Amoroso distribution [C2004] with location parameter a set to the physical limit, scale parameter and shape parameters , supplied by the user.

Note:

The mode of the distribution is typically not at the physical limit.

The limit values are required in order to check consistency with the parameter values.

For example, consider a yet unobserved branching ratio, which has to be non-negative, x>= 0.

Parameters:

cache The Observable cache from which we draw the predictions.

observable The Observable whose distribution we model.

mode The lower, physical limit. A branching ratio has to be >= 0.

upper_limit_10 With 10% probability, x < upper_limit_10.

upper_limit_50 With 50% probability, x < upper_limit_50.

upper_limit_90 With 90% probability, x < upper_limit_90.

theta scale parameter

alpha 1st shape parameter

beta 2nd shape parameter

Returns:

6.110.3.3 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::AmorosoLimit (ObservableCache *cache*, const ObservablePtr & *observable*, const double & *physical_limit*, const double & *upper_limit_90*, const double & *upper_limit_95*, const double & *theta*, const double & *alpha*, const unsigned & *number_of_observations* = `1u`) `[static]`

A likelihood contribution representing an upper limit on a quantity x.

Internally, it is represented by an Amoroso distribution [C2004] with location parameter a set to the physical limit, scale parameter and first shape parameter supplied by the user, and 2nd shape parameter set to the inverse of , in order to ensure that the maximum of the density is at the physical limit.

The limit values are required in order to check consistency with the parameter values.

For example, consider a yet unobserved branching ratio, which has to be non-negative, x>= 0.

Parameters:

cache The Observable cache from which we draw the predictions.

observable The Observable whose distribution we model.

mode The lower, physical limit. A branching ratio has to be >= 0. It is the maximum of the pdf.

upper_limit_90 With 90% probability, x < upper_limit_90.

upper_limit_95 With 95% probability, x < upper_limit_95.

theta scale parameter

alpha shape parameter

Returns:

6.110.3.4 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::AmorosoMode (ObservableCache *cache*, const ObservablePtr & *observable*, const double & *physical_limit*, const double & *mode*, const double & *upper_limit_90*, const double & *upper_limit_95*, const double & *theta*, const double & *alpha*, const double & *beta*, const unsigned & *number_of_observations* = `1u`) `[static]`

A likelihood contribution representing an upper limit on a quantity x.

Internally, it is represented by an Amoroso distribution [C2004] with location parameter a set to the physical limit, scale parameter and shape parameters , supplied by the user.

The limit values / mode are required in order to check consistency with the parameter values.

For example, consider a yet unobserved branching ratio, which has to be non-negative, x>= 0.

Parameters:

cache The Observable cache from which we draw the predictions.

observable The Observable whose distribution we model.

physical_limit The lower, physical limit. A branching ratio has to be >= 0.

mode The maximum of the distribution.

upper_limit_90 With 90% probability, x < upper_limit_90.

upper_limit_95 With 95% probability, x < upper_limit_95.

theta scale parameter

alpha 1st shape parameter

beta 2nd shape parameter

Returns:

6.110.3.5 virtual std::string eos::LogLikelihoodBlock::as_string () const `[pure virtual]`

6.110.3.6 virtual LogLikelihoodBlockPtr eos::LogLikelihoodBlock::clone (ObservableCache *cache*) const `[pure virtual]`

Clone this block.

6.110.3.7 virtual double eos::LogLikelihoodBlock::evaluate () const [pure virtual]

Compute the logarithm of the likelihood for this block.

6.110.3.8 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::Gaussian (ObservableCache *cache*, const ObservablePtr & *observable*, const double & *min*, const double & *central*, const double & *max*, const unsigned & *number_of_observations* = 1u) [static]

Create a new LogLikelihoodBlock for one normally distributed observable.

Note:

For every dimension, this template and the corresponding implementation have to be instantiated explicitly.

Parameters:

cache The Observable cache from which we draw the predictions.

observable The Observable whose distribution we model.

min The value one sigma below the mean of the experimental distribution.

central The mean value of the experimental distribution.

max The value one sigma above the mean of the experimental distribution.

6.110.3.9 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::LogGamma (ObservableCache *cache*, const ObservablePtr & *observable*, const double & *min*, const double & *central*, const double & *max*, const double & *lambda*, const double & *alpha*, const unsigned & *number_of_observations* = 1u) [static]

Create a new LogLikelihoodBlock for one a single observable with asymmetric uncertainties.

Note:

The only difference to the constructor without the and arguments is that now the LogGamma distribution is defined explicitly, and no numerical equation solving is needed. However, consistency is checked.

Parameters:

lambda The scale parameter of a LogGamma distribution.

alpha The shape parameter of a LogGamma distribution.

Returns:

6.110.3.10 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::LogGamma (ObservableCache *cache*, const ObservablePtr & *observable*, const double & *min*, const double & *central*, const double & *max*, const unsigned & *number_of_observations* = 1u) [static]

Create a new LogLikelihoodBlock for one a single observable with asymmetric uncertainties.

By construction, it is constructed to behave similarly to a Gaussian:

- the mode is at the central value
- the interval [min, max] contains 68% probability
- the density at min is the same as at max For details see [C2004].

Note:

Finding the correct parameter values is accomplished by solving a set of two equations numerically. This is unstable if max and min uncertainty differ by less than 5%. In that case, just a Gaussian instead.

Parameters:

cache The Observable cache from which we draw the predictions.

observable The Observable whose distribution we model.

min The value one sigma below the mean of the experimental distribution.

central The mean value of the experimental distribution.

max The value one sigma above the mean of the experimental distribution.

6.110.3.11 static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::Mixture (const std::vector< LogLikelihoodBlockPtr > & *components*, const std::vector< double > & *weights*) `[static]`

6.110.3.12 template<std::size_t n_> static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::MultivariateGaussian (ObservableCache *cache*, const std::array< ObservablePtr, n_ > & *observables*, const std::array< double, n_ > & *mean*, const std::array< double, n_ > & *variances*, const std::array< std::array< double, n_ >, n_ > & *correlation*, const unsigned & *number_of_observations* = n_) `[inline, static]`

Create a new LogLikelihoodBlock for n observables distributed according to a multivariate normal distribution.

Note:

For every dimension, this template and the corresponding implementation have to be instantiated explicitly.

Parameters:

cache The Observable cache from which we draw the predictions.

observables The Observables whose distribution we model.

mean The vector of means.

variances The vector of variances.

correlation The correlation matrix. Diagonal is assumed to be one.

6.110.3.13 template<std::size_t n_> static LogLikelihoodBlockPtr eos::LogLikelihoodBlock::MultivariateGaussian (ObservableCache *cache*, const std::array< ObservablePtr, n_ > & *observables*, const std::array< double, n_ > & *mean*, const std::array< std::array< double, n_ >, n_ > & *covariance*, const unsigned & *number_of_observations* = n_) `[inline, static]`

Create a new LogLikelihoodBlock for n observables distributed according to a multivariate normal distribution.

Note:

For every dimension, this template and the corresponding implementation have to be instantiated explicitly.

Parameters:

cache The Observable cache from which we draw the predictions.

observables The Observables whose distribution we model.

mean The vector of means.

covariance The covariance matrix

6.110.3.14 virtual unsigned eos::LogLikelihoodBlock::number_of_observations () const [pure virtual]

The number of experimental observations (not observables!) used in this block.

6.110.3.15 virtual void eos::LogLikelihoodBlock::prepare_sampling () [virtual]

Fix the predictions for fixed parameters within given model. Calculate any normalization constants to fix the sampling distribution in order to speed up repeated calls of sample().

6.110.3.16 virtual double eos::LogLikelihoodBlock::sample (gsl_rng * *rng*) const [pure virtual]

Sample from the logarithm of the likelihood for this block.

Warning:

Call prepare_sampling() before a call to sample() to ensure that one really gets a log likelihood value from the correct distribution, i.e. llh $\sim$ llh(model, fixed parameters)

Parameters:

rng The random number generator.

6.110.3.17 virtual double eos::LogLikelihoodBlock::significance () const [pure virtual]

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

- /home/alexshires/EOS/eos-next/eos/utils/log_likelihood.hh

6.111 eos::LogMessageHandler Class Reference

`#include <log.hh>`

Public Member Functions

- template<typename T_ >
  LogMessageHandler & operator<< (const T_ &t)

Basic Function

Destructor.

- ~LogMessageHandler ()

Friends

- LogMessageHandler Log::message (const std::string &, const LogLevel &)

6.111.1 Detailed Description

Proxy returned by Log::message.

6.111.2 Constructor & Destructor Documentation

6.111.2.1 eos::LogMessageHandler::~LogMessageHandler ()

6.111.3 Member Function Documentation

6.111.3.1 template<typename T_ > LogMessageHandler& eos::LogMessageHandler::operator<< (const T_ & *t*) [inline]

Append to our message.

6.111.4 Friends And Related Function Documentation

6.111.4.1 LogMessageHandler Log::message (const std::string &, const LogLevel &) [friend]

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

- /home/alexshires/EOS/eos-next/eos/utils/log.hh

6.112 eos::LogPrior Class Reference

```
#include <log_prior.hh>
```

Public Member Functions

Basic Functions

- LogPrior (const Parameters &parameters)
- virtual ~LogPrior ()
- virtual std::string as_string () const =0
- virtual LogPriorPtr clone (const Parameters &parameters) const =0

Accessors

- virtual double operator() () const =0
- virtual double sample (gsl_rng *rng) const =0
- virtual double mean () const =0
- virtual double variance () const =0

Static Public Member Functions

Named constructors for 1D prior distributions

- static LogPriorPtr Discrete (const Parameters &parameters, const std::string &name, const std::set< double > &values)
- static LogPriorPtr Flat (const Parameters &parameters, const std::string &name, const ParameterRange &range)
- static LogPriorPtr Gauss (const Parameters &parameters, const std::string &name, const ParameterRange &range, const double &lower, const double &central, const double &upper)
- static LogPriorPtr LogGamma (const Parameters &parameters, const std::string &name, const ParameterRange &range, const double &lower, const double &central, const double &upper)
- static LogPriorPtr Make (const Parameters &parameters, const std::string &serialization)

Protected Attributes

- Parameters _parameters

  Our associated Parameters object.

- std::vector< ParameterDescription > _parameter_descriptions

  All parameters for which this prior provides information.

Iteration over descriptions

- typedef WrappedForwardIterator< IteratorTag, ParameterDescription > Iterator
- Iterator begin ()
- Iterator end ()

6.112.1 Detailed Description

Base class for log(prior) distributions.

Has a container of subclasses describing independent 1..k dimensional prior distributions. Taken together, they specify the full N dimensional prior. Any actual calculation is done by the subclasses.

6.112.2 Member Typedef Documentation

6.112.2.1 typedef WrappedForwardIterator<IteratorTag, ParameterDescription> eos::LogPrior::Iterator

6.112.3 Constructor & Destructor Documentation

6.112.3.1 eos::LogPrior::LogPrior (const Parameters & *parameters*)

Constructor.

Parameters:

parameters The Parameters objects from which we evaluate.

6.112.3.2 virtual eos::LogPrior::~LogPrior () `[inline, virtual]`

Constructor.

Parameters:

parameters The Parameters objects from which we evaluate.

6.112.4 Member Function Documentation

6.112.4.1 virtual std::string eos::LogPrior::as_string () const `[pure virtual]`

Constructor.

Parameters:

parameters The Parameters objects from which we evaluate.

6.112.4.2 Iterator eos::LogPrior::begin ()

6.112.4.3 virtual LogPriorPtr eos::LogPrior::clone (const Parameters & *parameters*) const `[pure virtual]`

Create a clone (independent object) of this LogPrior object,

Parameters:

parameters The Parameters object from which the clone shall evaluate.

Returns:

The clone

6.112.4.4 static LogPriorPtr eos::LogPrior::Discrete (const Parameters & *parameters*, const std::string & *name*, const std::set< double > & *values*) [static]

The LogGamma distribution is useful to model continuous, unimodal, asymmetric priors in one dimension.

Note:

The construction will typically fail if the asymmetry is less than ca. 5 %. Use a Gaussian instead.

By construction, it is constructed to behave similarly to a Gaussian:

- the mode is at the central value
- the interval [lower, upper] contains 68% probability
- the density at lower is the same as at upper

Parameters:

parameters The object from which the value of the parameter is retrieved.

name The parameter name.

range Total allowed range. The value of the pdf is rescaled such that the prior integrates to one over the range. For efficiency reasons, no checking if input parameter is in range. But conceptually, the prior is zero outside.

lower (central - lower) roughly corresponds to a Gaussian standard deviation for values left of the mode.

central The mode of the prior.

upper (upper - central) roughly corresponds to a Gaussian standard deviation for values right of the mode.

Returns:

6.112.4.5 Iterator eos::LogPrior::end ()

6.112.4.6 static LogPriorPtr eos::LogPrior::Flat (const Parameters & *parameters*, const std::string & *name*, const ParameterRange & *range*) [static]

The LogGamma distribution is useful to model continuous, unimodal, asymmetric priors in one dimension.

Note:

The construction will typically fail if the asymmetry is less than ca. 5 %. Use a Gaussian instead.

By construction, it is constructed to behave similarly to a Gaussian:

- the mode is at the central value
- the interval [lower, upper] contains 68% probability
- the density at lower is the same as at upper

Parameters:

parameters The object from which the value of the parameter is retrieved.

name The parameter name.

range Total allowed range. The value of the pdf is rescaled such that the prior integrates to one over the range. For efficiency reasons, no checking if input parameter is in range. But conceptually, the prior is zero outside.

lower (central - lower) roughly corresponds to a Gaussian standard deviation for values left of the mode.

central The mode of the prior.

upper (upper - central) roughly corresponds to a Gaussian standard deviation for values right of the mode.

Returns:

6.112.4.7 static LogPriorPtr eos::LogPrior::Gauss (const Parameters & *parameters*, const std::string & *name*, const ParameterRange & *range*, const double & *lower*, const double & *central*, const double & *upper*) `[static]`

The LogGamma distribution is useful to model continuous, unimodal, asymmetric priors in one dimension.

Note:

The construction will typically fail if the asymmetry is less than ca. 5 %. Use a Gaussian instead.

By construction, it is constructed to behave similarly to a Gaussian:

- the mode is at the central value
- the interval [lower, upper] contains 68% probability
- the density at lower is the same as at upper

Parameters:

parameters The object from which the value of the parameter is retrieved.

name The parameter name.

range Total allowed range. The value of the pdf is rescaled such that the prior integrates to one over the range. For efficiency reasons, no checking if input parameter is in range. But conceptually, the prior is zero outside.

lower (central - lower) roughly corresponds to a Gaussian standard deviation for values left of the mode.

central The mode of the prior.

upper (upper - central) roughly corresponds to a Gaussian standard deviation for values right of the mode.

Returns:

6.112.4.8 static LogPriorPtr eos::LogPrior::LogGamma (const Parameters & *parameters*, const std::string & *name*, const ParameterRange & *range*, const double & *lower*, const double & *central*, const double & *upper*) `[static]`

The LogGamma distribution is useful to model continuous, unimodal, asymmetric priors in one dimension.

Note:

The construction will typically fail if the asymmetry is less than ca. 5 %. Use a Gaussian instead.

By construction, it is constructed to behave similarly to a Gaussian:

- the mode is at the central value
- the interval [lower, upper] contains 68% probability
- the density at lower is the same as at upper

Parameters:

parameters The object from which the value of the parameter is retrieved.

name The parameter name.

range Total allowed range. The value of the pdf is rescaled such that the prior integrates to one over the range. For efficiency reasons, no checking if input parameter is in range. But conceptually, the prior is zero outside.

lower (central - lower) roughly corresponds to a Gaussian standard deviation for values left of the mode.

central The mode of the prior.

upper (upper - central) roughly corresponds to a Gaussian standard deviation for values right of the mode.

Returns:

6.112.4.9 static LogPriorPtr eos::LogPrior::Make (const Parameters & *parameters*, const std::string & *serialization*) `[static]`

Construct a prior from its string representation.

Parameters:

parameters The object from which the value of the parameter is retrieved.

serialization The prior in string form.

Returns:

Pointer to the ready made prior.

6.112.4.10 virtual double eos::LogPrior::mean () const `[pure virtual]`

Return the mean of the distribution.

6.112.4.11 virtual double eos::LogPrior::operator() () const [pure virtual]

Evaluate the natural logarithm of the prior.

6.112.4.12 virtual double eos::LogPrior::sample (gsl_rng * *rng*) const [pure virtual]

Parameters:

rng the random number engine

Returns:

a sample according to this prior distribution

6.112.4.13 virtual double eos::LogPrior::variance () const [pure virtual]

Return the variance of this distribution.

6.112.5 Member Data Documentation

6.112.5.1 std::vector<ParameterDescription> eos::LogPrior::_parameter_descriptions [protected]

All parameters for which this prior provides information.

6.112.5.2 Parameters eos::LogPrior::_parameters [protected]

Our associated Parameters object.

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

- /home/alexshires/EOS/eos-next/eos/utils/log_prior.hh

6.113 eos::LongDistance Struct Reference

```
#include <long-distance.hh>
```

Static Public Member Functions

- static complex< double > g_had_ccbar (const double &s, const double &m_c)

6.113.1 Member Function Documentation

6.113.1.1 static complex<double> eos::LongDistance::g_had_ccbar (const double & *s*, const double & *m_c*) [static]

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/long-distance.hh

6.114 eos::LowRecoil Struct Reference

```
#include <decays.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/decays.hh

6.115 eos::MarkovChain Class Reference

`#include <markov_chain.hh>`

Inherits PrivateImplementationPattern< MarkovChain >.

Classes

- struct History
- struct HyperParameter
- struct ProposalFunction
- struct State
- struct Stats

Public Member Functions

- void clear ()

  Remove existing history of this chain.

- const State & current_state () const

  Retrieve information regarding the current state.

- void dump_description (hdf5::File &file, const std::string &data_set) const
- void dump_history (hdf5::File &file, const std::string &data_set_name, const unsigned &last_-iterations) const
- void dump_proposal (hdf5::File &file, const std::string &data_set_name) const
- MarkovChain::HyperParameter & hyper_parameter (bool current=true) const
- const unsigned & iterations_last_run () const

  Retrieve the number of iterations used in the last run.

- const History & history () const

  Retrieve the chain's detailed history.

- void set_analysis (const Analysis &analysis)

  Set analysis.

- void keep_history (bool keep_samples, bool keep_observables_and_proposals=false)
- void normalized_density (std::tuple< double, double > &result, const std::vector< double > &point, const unsigned &posterior_evaluations) const
- const std::vector< ParameterDescription > & parameter_descriptions () const

  Retrieve the descriptions of all parameters that are explored by this chain.

- bool proposal_accepted () const

  Check whether the most recently proposed move was accepted.

- std::shared_ptr< MarkovChain::ProposalFunction > proposal_function () const

  Access to the proposal function.

- void proposal_function (const std::shared_ptr< MarkovChain::ProposalFunction > &)

  Set the proposal function.

- const State & proposed_state () const

  Retrieve information regarding the most recently proposed state.

- void reset (bool hard=false)
- void run (const unsigned &iterations)
- void set_mode (hdf5::File &file, const std::string &data_base_name, const std::vector< double > &point, const double &posterior)
- void set_point (const std::vector< double > &point, const MarkovChain::HyperParameter &hyper_-parameter)
- const Stats & statistics () const

  Retrieve statistical data that summarizes the evolution of the chain up to the current point.

Basic Functions

- MarkovChain (const Analysis &analysis, unsigned long seed, const std::shared_ptr< MarkovChain::ProposalFunction > &proposal_function)
- ~MarkovChain ()

  Destructor.

Static Public Member Functions

- static void read_data (hdf5::File &file, const std::string &data_base_name, MarkovChain::History &history, std::shared_ptr< MarkovChain::ProposalFunction > &proposal, std::string &proposal_-type, MarkovChain::Stats &stats)
- static void read_descriptions (hdf5::File &file, const std::string &data_base_name, std::vector< Pa-rameterDescription > &descriptions, std::vector< std::string > &priors, std::vector< std::string > &constraints, std::string &hash)

6.115.1 Constructor & Destructor Documentation

6.115.1.1 eos::MarkovChain::MarkovChain (const Analysis & *analysis*, unsigned long *seed*, const std::shared_ptr< MarkovChain::ProposalFunction > & *proposal_function*)

Constructor

Parameters:

analysis The analysis for which we run.

seed The initial seed for the RNG.

6.115.1.2 eos::MarkovChain::~MarkovChain ()

Destructor.

6.115.2 Member Function Documentation

6.115.2.1 void eos::MarkovChain::clear ()

Remove existing history of this chain.

6.115.2.2 const State& eos::MarkovChain::current_state () const

Retrieve information regarding the current state.

6.115.2.3 void eos::MarkovChain::dump_description (hdf5::File & *file*, const std::string & *data_set*) const

6.115.2.4 void eos::MarkovChain::dump_history (hdf5::File & *file*, const std::string & *data_set_name*, const unsigned & *last_iterations*) const

Dump a part of the most recent history in the HDF5 file under the given group name.

Parameters:

file

data_set_name All output is stored below this directory.

last_iterations Dump only this many iterations.

6.115.2.5 void eos::MarkovChain::dump_proposal (hdf5::File & *file*, const std::string & *data_set_name*) const

6.115.2.6 const History& eos::MarkovChain::history () const

Retrieve the chain's detailed history.

6.115.2.7 MarkovChain::HyperParameter& eos::MarkovChain::hyper_parameter (bool *current* = `true`) const

6.115.2.8 const unsigned& eos::MarkovChain::iterations_last_run () const

Retrieve the number of iterations used in the last run.

6.115.2.9 void eos::MarkovChain::keep_history (bool *keep_samples*, bool *keep_observables_and_proposals* = `false`)

Set whether the chain stores samples in runs to come.

Parameters:

keep_samples If true, the history of the runs to come will be kept.

keep_observables_and_proposals For debugging purposes, store the proposed points and the observable values.

6.115.2.10 void eos::MarkovChain::normalized_density (std::tuple< double, double > & *result*, const std::vector< double > & *point*, const unsigned & *posterior_evaluations*) const

Compute estimate of normalized posterior density of this chain according to method described in Chib, Jeliazkov (2001), Eq. (9)

Parameters:

<numerator,denominator> of [CJ2001], Eq. (9)

point where density is sought

posterior_evaluations the number of additional posterior evaluations needed to estimate the normalized posterior density, see denominator of [CJ2001], Eq. (9)

6.115.2.11 const std::vector<ParameterDescription>& eos::MarkovChain::parameter_descriptions () const

Retrieve the descriptions of all parameters that are explored by this chain.

6.115.2.12 bool eos::MarkovChain::proposal_accepted () const

Check whether the most recently proposed move was accepted.

6.115.2.13 void eos::MarkovChain::proposal_function (const std::shared_ptr< MarkovChain::ProposalFunction > &)

Set the proposal function.

6.115.2.14 std::shared_ptr<MarkovChain::ProposalFunction> eos::MarkovChain::proposal_function () const

Access to the proposal function.

6.115.2.15 const State& eos::MarkovChain::proposed_state () const

Retrieve information regarding the most recently proposed state.

6.115.2.16 static void eos::MarkovChain::read_data (hdf5::File & *file*, const std::string & *data_base_name*, MarkovChain::History & *history*, std::shared_ptr< MarkovChain::ProposalFunction > & *proposal*, std::string & *proposal_type*, MarkovChain::Stats & *stats*) `[static]`

Read part of the output of a chain's prerun from hdf5 file.

Parameters:

file

data_base_name The directory in the file under which the data is parsed.

history The parsed chain's history is returned by reference, assumed to be empty initially.

proposal The local proposal density is recreated and returned by reference.

proposal_type The type of the local proposal function used.

stats Only the mode of the posterior is restored and returned by reference.

6.115.2.17 static void eos::MarkovChain::read_descriptions (hdf5::File & *file*, const std::string & *data_base_name*, std::vector< ParameterDescription > & *descriptions*, std::vector< std::string > & *priors*, std::vector< std::string > & *constraints*, std::string & *hash*) `[static]`

Read the description part of chain's prerun from hdf5 file.

Parameters:

file

data_base_name The directory in the file under which the data is parsed.

descriptions All parameter ranges etc. Beware, the association to the underlying Parameters object is independent.

priors The string representation of a prior distribution.

constraints The string representation of an individual constraint.

hash The EOS version used to create the file.

6.115.2.18 void eos::MarkovChain::reset (bool *hard* = `false`)

Clear all statistics and counter.

Do not change current position nor scale.

Parameters:

hard If true, rease all statistics (use to clear prerun data before main run)

6.115.2.19 void eos::MarkovChain::run (const unsigned & *iterations*)

Perform a number of iterations.

Parameters:

iterations The number of iterations that shall be performed.

6.115.2.20 void eos::MarkovChain::set_analysis (const Analysis & *analysis*)

Set analysis.

6.115.2.21 void eos::MarkovChain::set_mode (hdf5::File & *file*, const std::string & *data_base_name*, const std::vector< double > & *point*, const double & *posterior*)

Set the stats at mode to a point found outside of the chain. Triggers writing to the HDF5 file as another row in the stats section.

Parameters:

point

posterior

6.115.2.22 void eos::MarkovChain::set_point (const std::vector< double > & *point*, const MarkovChain::HyperParameter & *hyper_parameter*)

Set the chain to continue its walk from the given point.

Parameters:

point The point in parameter space.

6.115.2.23 const Stats& eos::MarkovChain::statistics () const

Retrieve statistical data that summarizes the evolution of the chain up to the current point.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh

6.116 eos::MarkovChainSampler Class Reference

`#include <markov_chain_sampler.hh>`

Inherits PrivateImplementationPattern< MarkovChainSampler >.

Classes

- class Config
- struct PreRunInfo

Public Member Functions

Sampling

- void massive_mode_finding (const Analysis::OptimizationOptions &options)
- PreRunInfo pre_run_info ()

  Retrieve information about the prerun performance.

- void run ()

  Start the Markov chain sampling.

- const MarkovChainSampler::Config & config ()

  Retrieve the configuration from which this sampler was constructed.

Basic Functions

- MarkovChainSampler (const Analysis &analysis, const MarkovChainSampler::Config &config)
- ~MarkovChainSampler ()

  Destructor.

- void resume (const hdf5::File &)
- static std::vector< HistoryPtr > build_global_local (const std::string &output_file_name, const std::vector< std::shared_ptr< hdf5::File >> input_files, const proposal_functions::GlobalLocal::Config &config, AnalysisPtr analysis=AnalysisPtr())

6.116.1 Constructor & Destructor Documentation

6.116.1.1 eos::MarkovChainSampler::MarkovChainSampler (const Analysis & *analysis*, const MarkovChainSampler::Config & *config*)

Constructor.

Parameters:

analysis The analysis for which we draw samples.

config The configuration of the samples.

6.116.1.2 eos::MarkovChainSampler::~MarkovChainSampler ()

Destructor.

6.116.2 Member Function Documentation

6.116.2.1 static std::vector<HistoryPtr> eos::MarkovChainSampler::build_global_local (const std::string & *output_file_name*, const std::vector< std::shared_ptr< hdf5::File >> *input_files*, const proposal_functions::GlobalLocal::Config & *config*, AnalysisPtr *analysis* = AnalysisPtr ()) [static]

Take the output from several independent preruns as input and store a global local proposal function to the output file in directory '/global local' In addition, the meta information about constraints, parameters and their priors is stored in '/descriptions'.

Parameters:

output_file_name The name of the HDF5 file containing the global local proposal function. if empty, do not create the proposal nor any output.

input_files HDF5 files containing the history, descriptions, stats, and local propopal densities as created by EOS.

config The configuration options influencing how GlobalLocal is created.

analysis Compare the order of parameters in files vs the one given in analysis.

Returns:

The full histories of all chains found in the input files

6.116.2.2 const MarkovChainSampler::Config& eos::MarkovChainSampler::config ()

Retrieve the configuration from which this sampler was constructed.

6.116.2.3 void eos::MarkovChainSampler::massive_mode_finding (const Analysis::OptimizationOptions & *options*)

Use minuit to find local minima.

Take as many starting points as there are chains

Parameters:

options Pass options on to Minuit.

6.116.2.4 PreRunInfo eos::MarkovChainSampler::pre_run_info ()

Retrieve information about the prerun performance.

6.116.2.5 void eos::MarkovChainSampler::resume (const hdf5::File &)

Copy the settings such as proposal density from the output of a (successful) prerun to prepare for calling run(), where the main run is started immediately.

6.116.2.6 void eos::MarkovChainSampler::run ()

Start the Markov chain sampling.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain_sampler.hh

6.117 eos::MemoisationControl Class Reference

```
#include <memoise.hh>
```

Public Member Functions

- MemoisationControl ()
- ~MemoisationControl ()
- void register_clear_function (const std::function< void()> &clear_function)
- void clear ()

6.117.1 Constructor & Destructor Documentation

6.117.1.1 eos::MemoisationControl::MemoisationControl ()

6.117.1.2 eos::MemoisationControl::~MemoisationControl ()

6.117.2 Member Function Documentation

6.117.2.1 void eos::MemoisationControl::clear ()

6.117.2.2 void eos::MemoisationControl::register_clear_function (const std::function< void()> & *clear_function*)

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

- /home/alexshires/EOS/eos-next/eos/utils/memoise.hh

6.118 eos::Memoiser< Result_, Params_ > Class Template Reference

```
#include <memoise.hh>
```

Public Types

- typedef Result_(∗ FunctionType)(const Params_ &...)
- typedef std::tuple< FunctionType, Params_...> KeyType

Public Member Functions

- Memoiser ()
- ∼Memoiser ()
- Result_ operator() (const FunctionType &f, const Params_ &...p)
- void clear ()
- unsigned number_of_memoisations () const

template<typename Result_, typename... Params_> class eos::Memoiser< Result_, Params_ >

6.118.1 Member Typedef Documentation

6.118.1.1 template<typename Result_ , typename... Params_> typedef Result_(∗ eos::Memoiser< Result_, Params_ >::FunctionType)(const Params_ &...)

6.118.1.2 template<typename Result_ , typename... Params_> typedef std::tuple<FunctionType, Params_...> eos::Memoiser< Result_, Params_ >::KeyType

6.118.2 Constructor & Destructor Documentation

6.118.2.1 template<typename Result_ , typename... Params_> eos::Memoiser< Result_, Params_ >::Memoiser () [inline]

6.118.2.2 template<typename Result_ , typename... Params_> eos::Memoiser< Result_, Params_ >::∼Memoiser () [inline]

6.118.3 Member Function Documentation

6.118.3.1 template<typename Result_ , typename... Params_> void eos::Memoiser< Result_, Params_ >::clear () [inline]

6.118.3.2 template<typename Result_ , typename... Params_> unsigned eos::Memoiser< Result_, Params_ >::number_of_memoisations () const [inline]

6.118.3.3 template<typename Result_ , typename... Params_> Result_ eos::Memoiser< Result_, Params_ >::operator() (const FunctionType & *f*, const Params_ &... *p*) [inline]

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

- /home/alexshires/EOS/eos-next/eos/utils/memoise.hh

6.119 eos::Minimal Struct Reference

```
#include <inclusive-b-to-s-gamma.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh

6.120 eos::Model Class Reference

`#include <model.hh>`

Inherits eos::ParameterUser, eos::ModelComponent< components::CKM >, eos::ModelComponent< components::QCD >, and eos::ModelComponent< components::DeltaB1 >.

Inherited by eos::CKMScanModel, eos::StandardModel, and eos::WilsonScanModel.

Public Member Functions

- virtual ~Model ()=0

Static Public Member Functions

- static std::shared_ptr< Model > make (const std::string &name, const Parameters &parameters, const Options &options)

6.120.1 Detailed Description

Base class for all models.

6.120.2 Constructor & Destructor Documentation

6.120.2.1 virtual eos::Model::~Model () `[pure virtual]`

6.120.3 Member Function Documentation

6.120.3.1 static std::shared_ptr<Model> eos::Model::make (const std::string & *name*, const Parameters & *parameters*, const Options & *options*) `[static]`

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

- /home/alexshires/EOS/eos-next/eos/utils/model.hh

6.121 eos::ModelComponent< components::CKM > Class Template Reference

`#include <model.hh>`

Inherited by eos::CKMScanComponent [virtual], eos::Model [virtual], and eos::SMComponent< components::CKM > [virtual].

Public Member Functions

- virtual complex< double > ckm_cd () const =0
- virtual complex< double > ckm_cs () const =0
- virtual complex< double > ckm_cb () const =0
- virtual complex< double > ckm_ud () const =0
- virtual complex< double > ckm_us () const =0
- virtual complex< double > ckm_ub () const =0
- virtual complex< double > ckm_td () const =0
- virtual complex< double > ckm_ts () const =0
- virtual complex< double > ckm_tb () const =0

6.121.1 Detailed Description

template<> class eos::ModelComponent< components::CKM >

Base class for the CKM component of models.

6.121.2 Member Function Documentation

6.121.2.1 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_cb () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.2 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_cd () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.3 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_cs () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.4 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_tb () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.5 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_td () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.6 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_ts () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.7 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_ub () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.8 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_ud () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

6.121.2.9 virtual complex<double> eos::ModelComponent< components::CKM >::ckm_us () const [pure virtual]

Implemented in eos::CKMScanComponent, and eos::SMComponent< components::CKM >.

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

- /home/alexshires/EOS/eos-next/eos/utils/model.hh

6.122 eos::ModelComponent< components::DeltaB1 > Class Template Reference

`#include <model.hh>`

Inherited by eos::Model `[virtual]`, eos::SMComponent< components::DeltaB1 > `[virtual]`, and eos::WilsonScanComponent `[virtual]`.

Public Member Functions

- virtual WilsonCoefficients< BToS > wilson_coefficients_b_to_s (const bool &cp_conjugate=false) const =0

6.122.1 Detailed Description

template<> class eos::ModelComponent< components::DeltaB1 >

Base class for the Delta B = 1 FCNC component of models.

6.122.2 Member Function Documentation

6.122.2.1 virtual WilsonCoefficients<BToS> eos::ModelComponent< components::DeltaB1 >::wilson_coefficients_b_to_s (const bool & *cp_conjugate* = `false`) const `[pure virtual]`

Implemented in eos::SMComponent< components::DeltaB1 >, and eos::WilsonScanComponent.

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

- /home/alexshires/EOS/eos-next/eos/utils/model.hh

6.123 eos::ModelComponent< components::QCD > Class Template Reference

`#include <model.hh>`

Inherited by eos::Model `[virtual]`, and eos::SMComponent< components::QCD > `[virtual]`.

Public Member Functions

- virtual double alpha_s (const double &mu) const =0
- virtual double m_t_msbar (const double &mu) const =0
- virtual double m_t_pole () const =0
- virtual double m_b_kin (const double &mu_kin) const =0
- virtual double m_b_msbar (const double &mu) const =0
- virtual double m_b_pole () const =0
- virtual double m_b_ps (const double &mu_f) const =0
- virtual double m_c_kin (const double &mu_kin) const =0
- virtual double m_c_msbar (const double &mu) const =0
- virtual double m_c_pole () const =0
- virtual double m_s_msbar (const double &mu) const =0
- virtual double m_ud_msbar (const double &mu) const =0

6.123.1 Detailed Description

template<> class eos::ModelComponent< components::QCD >

Base class for the QCD component of models.

6.123.2 Member Function Documentation

6.123.2.1 virtual double eos::ModelComponent< components::QCD >::alpha_s (const double & *mu*) const `[pure virtual]`

Implemented in eos::SMComponent< components::QCD >.

6.123.2.2 virtual double eos::ModelComponent< components::QCD >::m_b_kin (const double & *mu_kin*) const `[pure virtual]`

Implemented in eos::SMComponent< components::QCD >.

6.123.2.3 virtual double eos::ModelComponent< components::QCD >::m_b_msbar (const double & *mu*) const `[pure virtual]`

Implemented in eos::SMComponent< components::QCD >.

6.123.2.4 virtual double eos::ModelComponent< components::QCD >::m_b_pole () const `[pure virtual]`

Implemented in eos::SMComponent< components::QCD >.

6.123.2.5 virtual double eos::ModelComponent< components::QCD >::m_b_ps (const double & *mu_f*) const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

6.123.2.6 virtual double eos::ModelComponent< components::QCD >::m_c_kin (const double & *mu_kin*) const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

6.123.2.7 virtual double eos::ModelComponent< components::QCD >::m_c_msbar (const double & *mu*) const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

6.123.2.8 virtual double eos::ModelComponent< components::QCD >::m_c_pole () const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

6.123.2.9 virtual double eos::ModelComponent< components::QCD >::m_s_msbar (const double & *mu*) const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

6.123.2.10 virtual double eos::ModelComponent< components::QCD >::m_t_msbar (const double & *mu*) const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

6.123.2.11 virtual double eos::ModelComponent< components::QCD >::m_t_pole () const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

6.123.2.12 virtual double eos::ModelComponent< components::QCD >::m_ud_msbar (const double & *mu*) const [pure virtual]

Implemented in eos::SMComponent< components::QCD >.

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

- /home/alexshires/EOS/eos-next/eos/utils/model.hh

6.124 eos::proposal_functions::Multivariate Class Reference

`#include <proposal_functions.hh>`

Inherits eos::MarkovChain::ProposalFunction.

Inherited by eos::proposal_functions::MultivariateGaussian, and eos::proposal_functions::MultivariateStudentT.

Public Types

- typedef hdf5::Array< 1, double > CovarianceType

Public Member Functions

- unsigned dimension () const

  The dimension of the space for which samples are proposed.

- const gsl_matrix * covariance () const
- Multivariate (const unsigned &dimension, const std::vector< double > &covariance, const bool &automatic_scaling)
- virtual ~Multivariate ()
- virtual void adapt (const MarkovChain::State::Iterator &begin, const MarkovChain::State::Iterator &end, const double &efficiency, const double &efficiency_min, const double &efficiency_max)
- virtual void rescale (const double &rescale_factor)
- virtual void reset (const std::vector< HistoryPtr > &histories, const double &scale, const double &skip_initial)
- virtual void set_indices (const std::vector< unsigned > &index_list)

Static Public Member Functions

- static CovarianceType covariance_type (const unsigned &dimension)

Public Attributes

- unsigned adaptations

  Record how often an adaption to data has been performed.

- double covariance_scale

  Rescale the sample _covariance to form the proposal _covariance.

- double cooling_power
- double norm

Static Public Attributes

- static const double covariance_scale_min

  Scale enforced to exceed a minimum value.

- static const double covariance_scale_max

  Scale enforced to lie below a maximum value.

- static const double covariance_scale_update_factor

Protected Member Functions

- void _compute_cholesky_and_inverse ()
- virtual void _compute_norm ()=0
- void _copy (const Multivariate &)
- void _dump_covariance (hdf5::File &file, const std::string &data_set_base_name, const std::string &proposal_type_name) const

Protected Attributes

- gsl_vector * _tmp_left
- gsl_vector * _tmp_right
- gsl_matrix * _tmp_sample_covariance_current
- gsl_matrix * _covariance
- gsl_matrix * _covariance_inverse
- gsl_matrix * _covariance_chol
- const unsigned _dimension
- std::vector< unsigned > _index_list

6.124.1 Member Typedef Documentation

6.124.1.1 typedef hdf5::Array<1, double> eos::proposal_functions::Multivariate::CovarianceType

6.124.2 Constructor & Destructor Documentation

6.124.2.1 eos::proposal_functions::Multivariate::Multivariate (const unsigned & *dimension*, const std::vector< double > & *covariance*, const bool & *automatic_scaling*)

Construct with given dimension and an initial proposal covariance matrix. If automatic scaling is desired, the given matrix is interpreted as an estimate of the target covariance, and it is rescaled by a factor of 2.38^2 / dimension. Else is the matrix is used as is for proposing points.

Note:

The automatic scaling can be shown to lead to the asymptotically (!) perfect value of the efficiency of 0.238 for a Gaussian proposal and a Gaussian target density with unlimited parameter ranges. These requirements are usually not fullfilled.

6.124.2.2 virtual eos::proposal_functions::Multivariate::~Multivariate () [virtual]

6.124.3 Member Function Documentation

6.124.3.1 void eos::proposal_functions::Multivariate::_compute_cholesky_and_inverse () [protected]

6.124.3.2 virtual void eos::proposal_functions::Multivariate::_compute_norm () [protected, pure virtual]

Implemented in eos::proposal_functions::MultivariateGaussian, and eos::proposal_functions::MultivariateStudentT.

6.124.3.3 void eos::proposal_functions::Multivariate::_copy (const Multivariate &) [protected]

6.124.3.4 void eos::proposal_functions::Multivariate::_dump_covariance (hdf5::File & *file*, const std::string & *data_set_base_name*, const std::string & *proposal_type_name*) const [protected]

6.124.3.5 virtual void eos::proposal_functions::Multivariate::adapt (const MarkovChain::State::Iterator & *begin*, const MarkovChain::State::Iterator & *end*, const double & *efficiency*, const double & *efficiency_min*, const double & *efficiency_max*) [virtual]

Note:

Expect history to have only the most recent part, to which proposal function has not adapted yet. Accordingly, the efficiency of that last part is required. Do not push the entire history of a chain in here repeatedly!

Implements eos::MarkovChain::ProposalFunction.

6.124.3.6 const gsl_matrix* eos::proposal_functions::Multivariate::covariance () const

6.124.3.7 static CovarianceType eos::proposal_functions::Multivariate::covariance_type (const unsigned & *dimension*) [static]

6.124.3.8 unsigned eos::proposal_functions::Multivariate::dimension () const

The dimension of the space for which samples are proposed.

6.124.3.9 virtual void eos::proposal_functions::Multivariate::rescale (const double & *rescale_factor*) [virtual]

Rescale the covariance scale factor

Parameters:

rescale_factor

6.124.3.10 virtual void eos::proposal_functions::Multivariate::reset (const std::vector< HistoryPtr > & *histories*, const double & *scale*, const double & *skip_initial*) [virtual]

Reset the internal covariance to zero, and recompute it from the histories given. Apply a new scale factor for proposals.

Parameters:

histories

scale

6.124.3.11 virtual void eos::proposal_functions::Multivariate::set_indices (const std::vector< unsigned > & *index_list*) [virtual]

Set an index list, such that the proposal will consider only a subspace for it covariance calculation.

6.124.4 Member Data Documentation

6.124.4.1 gsl_matrix* eos::proposal_functions::Multivariate::_covariance [protected]

6.124.4.2 gsl_matrix* eos::proposal_functions::Multivariate::_covariance_chol [protected]

6.124.4.3 gsl_matrix* eos::proposal_functions::Multivariate::_covariance_inverse [protected]

6.124.4.4 const unsigned eos::proposal_functions::Multivariate::_dimension [protected]

6.124.4.5 std::vector<unsigned> eos::proposal_functions::Multivariate::_index_list [protected]

6.124.4.6 gsl_vector* eos::proposal_functions::Multivariate::_tmp_left [protected]

6.124.4.7 gsl_vector* eos::proposal_functions::Multivariate::_tmp_right [protected]

6.124.4.8 gsl_matrix* eos::proposal_functions::Multivariate::_tmp_sample_covariance_current [protected]

6.124.4.9 unsigned eos::proposal_functions::Multivariate::adaptations

Record how often an adaption to data has been performed.

6.124.4.10 double eos::proposal_functions::Multivariate::cooling_power

Combinining the proposal _covariance of this and the previous step, limit the effect of samples in the distant past by forming a weighted average. In the n-th step,

= (1 - 1/n^{cooling_power}) + 1/n^{cooling_power} * S_n

cf. arxiv:0903.0837, Eq. (23)

6.124.4.11 double eos::proposal_functions::Multivariate::covariance_scale

Rescale the sample _covariance to form the proposal _covariance.

6.124.4.12 const double eos::proposal_functions::Multivariate::covariance_scale_max [static]

Scale enforced to lie below a maximum value.

6.124.4.13 const double eos::proposal_functions::Multivariate::covariance_scale_min [static]

Scale enforced to exceed a minimum value.

6.124.4.14 const double eos::proposal_functions::Multivariate::covariance_scale_update_factor [static]

During an adaptation, the scale is multiplied/divided by this factor if the efficiency is too high/low .

6.124.4.15 double eos::proposal_functions::Multivariate::norm

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.125 eos::proposal_functions::MultivariateAccess Struct Reference

```
#include <proposal_functions.hh>
```

Static Public Member Functions

- static MultivariateProposalPtr access (const ProposalFunctionPtr &p)

6.125.1 Member Function Documentation

6.125.1.1 static MultivariateProposalPtr eos::proposal_functions::MultivariateAccess::access (const ProposalFunctionPtr & *p*) [static]

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.126 eos::proposal_functions::MultivariateGaussian Class Reference

`#include <proposal_functions.hh>`

Inherits eos::proposal_functions::Multivariate.

Public Types

- typedef hdf5::Composite< hdf5::Scalar< double >, hdf5::Scalar< double >, hdf5::Scalar< unsigned > > ScalarsType

Public Member Functions

- MultivariateGaussian (const unsigned &dimension, const std::vector< double > &covariance, const bool &automatic_scaling=true)
- virtual ~MultivariateGaussian ()
- virtual ProposalFunctionPtr clone () const

  Create independent copy.

- virtual void dump_state (hdf5::File &file, const std::string &data_set_base_name) const

  Store state in the file under the given base name.

- virtual double evaluate (const MarkovChain::State &x, const MarkovChain::State &y) const

  Evaluate the density to propose x given y.

- virtual void propose (MarkovChain::State &x, const MarkovChain::State &y, gsl_rng *rng) const

  Obtain from the density a proposal x given y.

Static Public Member Functions

- static ScalarsType scalars_type ()

Protected Member Functions

- virtual void _compute_norm ()

6.126.1 Member Typedef Documentation

6.126.1.1 typedef hdf5::Composite<hdf5::Scalar<double>, hdf5::Scalar<double>, hdf5::Scalar<unsigned> > eos::proposal_functions::MultivariateGaussian::ScalarsType

6.126.2 Constructor & Destructor Documentation

6.126.2.1 eos::proposal_functions::MultivariateGaussian::MultivariateGaussian (const unsigned & *dimension*, const std::vector< double > & *covariance*, const bool & *automatic_scaling* = `true`)

6.126.2.2 virtual eos::proposal_functions::MultivariateGaussian::~MultivariateGaussian () `[virtual]`

6.126.3 Member Function Documentation

6.126.3.1 virtual void eos::proposal_functions::MultivariateGaussian::_compute_norm () `[protected, virtual]`

Implements eos::proposal_functions::Multivariate.

6.126.3.2 virtual ProposalFunctionPtr eos::proposal_functions::MultivariateGaussian::clone () const `[virtual]`

Create independent copy.

Implements eos::MarkovChain::ProposalFunction.

6.126.3.3 virtual void eos::proposal_functions::MultivariateGaussian::dump_state (hdf5::File & *file*, const std::string & *data_set_base_name*) const `[virtual]`

Store state in the file under the given base name.

Implements eos::MarkovChain::ProposalFunction.

6.126.3.4 virtual double eos::proposal_functions::MultivariateGaussian::evaluate (const MarkovChain::State & *x*, const MarkovChain::State & *y*) const `[virtual]`

Evaluate the density to propose x given y.

Implements eos::MarkovChain::ProposalFunction.

6.126.3.5 virtual void eos::proposal_functions::MultivariateGaussian::propose (MarkovChain::State & *x*, const MarkovChain::State & *y*, gsl_rng * *rng*) const `[virtual]`

Obtain from the density a proposal x given y.

Implements eos::MarkovChain::ProposalFunction.

6.126.3.6 static ScalarsType eos::proposal_functions::MultivariateGaussian::scalars_type () [static]

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.127 eos::proposal_functions::MultivariateStudentT Class Reference

`#include <proposal_functions.hh>`

Inherits eos::proposal_functions::Multivariate.

Public Types

- typedef hdf5::Composite< hdf5::Scalar< double >, hdf5::Scalar< double >, hdf5::Scalar< unsigned >, hdf5::Scalar< double > > ScalarsType

Public Member Functions

- MultivariateStudentT (const unsigned &dimension, const std::vector< double > &covariance, const double &degree_of_freedom, const bool &automatic_scaling=true)
- virtual ~MultivariateStudentT ()
- virtual ProposalFunctionPtr clone () const

  Create independent copy.

- virtual void dump_state (hdf5::File &file, const std::string &data_set_base_name) const

  Store state in the file under the given base name.

- virtual double evaluate (const MarkovChain::State &x, const MarkovChain::State &y) const

  Evaluate the density to propose x given y.

- virtual void propose (MarkovChain::State &x, const MarkovChain::State &y, gsl_rng *rng) const

  Obtain from the density a proposal x given y.

Static Public Member Functions

- static ScalarsType scalars_type ()

Public Attributes

- VerifiedRange< double > dof

Protected Member Functions

- virtual void _compute_norm ()

6.127.1 Member Typedef Documentation

6.127.1.1 typedef hdf5::Composite<hdf5::Scalar<double>, hdf5::Scalar<double>, hdf5::Scalar<unsigned>, hdf5::Scalar<double> > eos::proposal_functions::MultivariateStudentT::ScalarsType

6.127.2 Constructor & Destructor Documentation

6.127.2.1 eos::proposal_functions::MultivariateStudentT::MultivariateStudentT (const unsigned & *dimension*, const std::vector< double > & *covariance*, const double & *degree_of_freedom*, const bool & *automatic_scaling* = `true`)

6.127.2.2 virtual eos::proposal_functions::MultivariateStudentT::~MultivariateStudentT () `[virtual]`

6.127.3 Member Function Documentation

6.127.3.1 virtual void eos::proposal_functions::MultivariateStudentT::_compute_norm () `[protected, virtual]`

Implements eos::proposal_functions::Multivariate.

6.127.3.2 virtual ProposalFunctionPtr eos::proposal_functions::MultivariateStudentT::clone () const `[virtual]`

Create independent copy.

Implements eos::MarkovChain::ProposalFunction.

6.127.3.3 virtual void eos::proposal_functions::MultivariateStudentT::dump_state (hdf5::File & *file*, const std::string & *data_set_base_name*) const `[virtual]`

Store state in the file under the given base name.

Implements eos::MarkovChain::ProposalFunction.

6.127.3.4 virtual double eos::proposal_functions::MultivariateStudentT::evaluate (const MarkovChain::State & *x*, const MarkovChain::State & *y*) const `[virtual]`

Evaluate the density to propose x given y.

Implements eos::MarkovChain::ProposalFunction.

6.127.3.5 virtual void eos::proposal_functions::MultivariateStudentT::propose (MarkovChain::State & *x*, const MarkovChain::State & *y*, gsl_rng * *rng*) const `[virtual]`

Obtain from the density a proposal x given y.

Implements eos::MarkovChain::ProposalFunction.

6.127.3.6 static ScalarsType eos::proposal_functions::MultivariateStudentT::scalars_type () [static]

6.127.4 Member Data Documentation

6.127.4.1 VerifiedRange<double> eos::proposal_functions::MultivariateStudentT::dof

Degree of freedom. Required to be >= 1.

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.128 eos::Mutable Class Reference

Base class for all mutable entities, e.g. a Parameter.

`#include <mutable.hh>`

Inherited by eos::Parameter.

Public Member Functions

Basic Operatios

Destructor.

- virtual ~Mutable ()=0
  Make a copy of this Mutable.
- virtual MutablePtr clone () const =0
  Make a copy of this Mutable.

Access & Modification of the numeric Value

Cast a Mutable to a double.

- virtual operator double () const =0
  Retrieve a Mutable's numeric value.
- virtual double operator() () const =0
  Retrieve a Mutable's numeric value.
- virtual const Mutable & operator= (const double &)=0
  Set a Mutable's numeric value.

Access to Meta Data

Retrieve the Parameter's name.

- virtual const std::string & name () const =0

6.128.1 Detailed Description

Base class for all mutable entities, e.g. a Parameter.

6.128.2 Constructor & Destructor Documentation

6.128.2.1 virtual eos::Mutable::~Mutable () `[pure virtual]`

Make a copy of this Mutable.

6.128.3 Member Function Documentation

6.128.3.1 virtual MutablePtr eos::Mutable::clone () const [pure virtual]

Make a copy of this Mutable.

Implemented in eos::Parameter.

6.128.3.2 virtual const std::string& eos::Mutable::name () const [pure virtual]

Implemented in eos::Parameter.

6.128.3.3 virtual eos::Mutable::operator double () const [pure virtual]

Retrieve a Mutable's numeric value.

Implemented in eos::Parameter.

6.128.3.4 virtual double eos::Mutable::operator() () const [pure virtual]

Retrieve a Mutable's numeric value.

Implemented in eos::Parameter.

6.128.3.5 virtual const Mutable& eos::Mutable::operator= (const double &) [pure virtual]

Set a Mutable's numeric value.

Implemented in eos::Parameter.

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

- /home/alexshires/EOS/eos-next/eos/utils/mutable.hh

6.129 eos::Mutex Class Reference

```
#include <mutex.hh>
```

Public Member Functions

- Mutex ()

  (Explicit) constructor.

- ∼Mutex ()

  Destructor.

- pthread_mutex_t ∗ mutex ()

  Returns a pointer to our underlying posix mutex.

6.129.1 Constructor & Destructor Documentation

6.129.1.1 eos::Mutex::Mutex () [explicit]

(Explicit) constructor.

6.129.1.2 eos::Mutex::∼Mutex ()

Destructor.

6.129.2 Member Function Documentation

6.129.2.1 pthread_mutex_t∗ eos::Mutex::mutex () [inline]

Returns a pointer to our underlying posix mutex.

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

- /home/alexshires/EOS/eos-next/eos/utils/mutex.hh

6.130 eos::NLO Struct Reference

```
#include <inclusive-b-to-s-gamma.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh

6.131 eos::NoSuchChiSquaredError Struct Reference

```
#include <chi-squared.hh>
```

Inherits eos::Exception.

Public Member Functions

- NoSuchChiSquaredError (const std::string &name)

6.131.1 Constructor & Destructor Documentation

6.131.1.1 eos::NoSuchChiSquaredError::NoSuchChiSquaredError (const std::string & *name*)

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

- /home/alexshires/EOS/eos-next/eos/utils/chi-squared.hh

6.132 eos::NoSuchModelError Struct Reference

```
#include <model.hh>
```

Inherits eos::Exception.

Public Member Functions

- NoSuchModelError (const std::string &name)

6.132.1 Constructor & Destructor Documentation

6.132.1.1 eos::NoSuchModelError::NoSuchModelError (const std::string & *name*)

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

- /home/alexshires/EOS/eos-next/eos/utils/model.hh

6.133 eos::Observable Class Reference

`#include <observable.hh>`

Inherits eos::ParameterUser.

Inherited by eos::ConcreteObservable< Decay_, Args_ >, eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, and eos::TestObservable.

Public Member Functions

- virtual const std::string & name () const =0
- virtual double evaluate () const =0
- virtual Kinematics kinematics ()=0
- virtual Parameters parameters ()=0
- virtual Options options ()=0
- virtual ObservablePtr clone () const =0
- virtual ObservablePtr clone (const Parameters &parameters) const =0

Static Public Member Functions

- static ObservablePtr make (const std::string &name, const Parameters &parameters, const Kinematics &kinematics, const Options &options)

6.133.1 Member Function Documentation

6.133.1.1 virtual ObservablePtr eos::Observable::clone (const Parameters & *parameters*) const `[pure virtual]`

Implemented in eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, eos::TestObservable, eos::AbsoluteTestObservable, and eos::ConcreteObservable< Decay_, Args_ >.

6.133.1.2 virtual ObservablePtr eos::Observable::clone () const `[pure virtual]`

Implemented in eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, eos::TestObservable, eos::AbsoluteTestObservable, and eos::ConcreteObservable< Decay_, Args_ >.

6.133.1.3 virtual double eos::Observable::evaluate () const `[pure virtual]`

Implemented in eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, eos::TestObservable, eos::AbsoluteTestObservable, and eos::ConcreteObservable< Decay_, Args_ >.

6.133.1.4 virtual Kinematics eos::Observable::kinematics () `[pure virtual]`

Implemented in eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, eos::TestObservable, and eos::ConcreteObservable< Decay_, Args_ >.

6.133.1.5 static ObservablePtr eos::Observable::make (const std::string & *name*, const Parameters & *parameters*, const Kinematics & *kinematics*, const Options & *options*) [static]

6.133.1.6 virtual const std::string& eos::Observable::name () const [pure virtual]

Implemented in eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, eos::TestObservable, and eos::ConcreteObservable< Decay_, Args_ >.

6.133.1.7 virtual Options eos::Observable::options () [pure virtual]

Implemented in eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, eos::TestObservable, and eos::ConcreteObservable< Decay_, Args_ >.

6.133.1.8 virtual Parameters eos::Observable::parameters () [pure virtual]

Implemented in eos::FormFactorAdapter< Transition_ >, eos::FormFactorRatioAdapter< Transition_ >, eos::TestObservable, and eos::ConcreteObservable< Decay_, Args_ >.

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

- /home/alexshires/EOS/eos-next/eos/observable.hh

6.134 eos::ObservableCache Class Reference

`#include <observable_cache.hh>`

Inherits PrivateImplementationPattern< ObservableCache >.

Public Member Functions

Basic Functions

- ObservableCache (const Parameters &parameters)
- ~ObservableCache ()

  Destructor.

Access

- typedef unsigned Id
- Id add (const ObservablePtr &observable)
- void update ()

  Update the predictions for all observables.

- void update (const Parameter::Id &id)
- Parameters parameters () const

  Retrieve the cache's common Parameters object.

- void reset ()

  Reset the cache to the previous predictions.

- ObservablePtr observable (const ObservableCache::Id &id) const
- double operator[ ] (const ObservableCache::Id &id) const
- unsigned size () const

  Retrieve the number of independent predictions from the cache.

- ObservableCache clone (const Parameters &parameters) const

  Clone this cache whilst keeping the observables in the given order, i.e. all ids remain valid.

6.134.1 Member Typedef Documentation

6.134.1.1 typedef unsigned eos::ObservableCache::Id

Add a given observable to the cache and return its unique Id.

Parameters:

observable The observable which shall be added to the cache.

6.134.2 Constructor & Destructor Documentation

6.134.2.1 eos::ObservableCache::ObservableCache (const Parameters & *parameters*)

Constructor.

Parameters:

parameters The Parameters object common to all cached Observables.

6.134.2.2 eos::ObservableCache::∼ObservableCache ()

Destructor.

6.134.3 Member Function Documentation

6.134.3.1 Id eos::ObservableCache::add (const ObservablePtr & *observable*)

Add a given observable to the cache and return its unique Id.

Parameters:

observable The observable which shall be added to the cache.

6.134.3.2 ObservableCache eos::ObservableCache::clone (const Parameters & *parameters*) const

Clone this cache whilst keeping the observables in the given order, i.e. all ids remain valid.

6.134.3.3 ObservablePtr eos::ObservableCache::observable (const ObservableCache::Id & *id*) const

Retrieve a unique observable by its ObservableCache::Id.

Parameters:

id The ObservableCache::Id whose associated ObservablePtr shall be retrieved.

6.134.3.4 double eos::ObservableCache::operator[ ] (const ObservableCache::Id & *id*) const

Retrieve the prediction for a given observable from the cache.

Parameters:

id The unique ObservableCache::Id whose associated observable's prediction shall be retrieved.

6.134.3.5 Parameters eos::ObservableCache::parameters () const

Retrieve the cache's common Parameters object.

6.134.3.6 void eos::ObservableCache::reset ()

Reset the cache to the previous predictions.

6.134.3.7 unsigned eos::ObservableCache::size () const

Retrieve the number of independent predictions from the cache.

6.134.3.8 void eos::ObservableCache::update (const Parameter::Id & *id*)

Update the predicition for all observables that depend on a given Parameter.

Parameters:

id The Parameter::Id for which all depending observables shall be updated.

6.134.3.9 void eos::ObservableCache::update ()

Update the predictions for all observables.

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

- /home/alexshires/EOS/eos-next/eos/utils/observable_cache.hh

6.135 eos::ObservableFactory Class Reference

```
#include <observable.hh>
```

Inherited by eos::ConcreteObservableFactory< Decay_, Args_ >, eos::FormFactorAdapterFactory< Transition_ >, and eos::FormFactorRatioAdapterFactory< Transition_ >.

Public Member Functions

- ObservableFactory ()
- virtual ~ObservableFactory ()
- virtual ObservablePtr make (const Parameters &, const Kinematics &, const Options &) const =0

6.135.1 Constructor & Destructor Documentation

6.135.1.1 eos::ObservableFactory::ObservableFactory ()

6.135.1.2 virtual eos::ObservableFactory::~ObservableFactory () [virtual]

6.135.2 Member Function Documentation

6.135.2.1 virtual ObservablePtr eos::ObservableFactory::make (const Parameters &, const Kinematics &, const Options &) const [pure virtual]

Implemented in eos::FormFactorAdapterFactory< Transition_ >, eos::FormFactorRatioAdapterFactory< Transition_ >, and eos::ConcreteObservableFactory< Decay_, Args_ >.

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

- /home/alexshires/EOS/eos-next/eos/observable.hh

6.136 eos::ObservableNameError Struct Reference

```
#include <observable.hh>
```

Inherits eos::Exception.

Public Member Functions

Basic Functions

- ObservableNameError (const std::string &name)

6.136.1 Detailed Description

ObservableNameError is thrown when Observable::make encounters a malformed observable name.

6.136.2 Constructor & Destructor Documentation

6.136.2.1 eos::ObservableNameError::ObservableNameError (const std::string & *name*)

Constructor.

Parameters:

name The offending malformed observable name.

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

- /home/alexshires/EOS/eos-next/eos/observable.hh

6.137 eos::ObservableSet Class Reference

`#include <observable_set.hh>`

Inherits PrivateImplementationPattern< ObservableSet >.

Public Member Functions

Basic Functions

Constructor.

- ObservableSet ()
  Destructor.
- ~ObservableSet ()
  Destructor.

Iteration and Access

- typedef WrappedForwardIterator< IteratorTag, ObservablePtr > Iterator
  Iterator to the first observable.
- Iterator begin () const
  Iterator to the first observable.
- Iterator end () const
  Iterator pointing past the last observable.
- ObservablePtr & operator[ ] (const unsigned &index) const
- std::pair< unsigned, bool > add (const ObservablePtr &observable)
- Parameters parameters ()
  Access to the underlying Parameters object.
- unsigned size () const
  The total number of elements.

6.137.1 Detailed Description

A container class to hold only unique observables, i.e. they differ by at least one of the following: name, kinematics, or options.

6.137.2 Member Typedef Documentation

6.137.2.1 typedef WrappedForwardIterator<IteratorTag, ObservablePtr> eos::ObservableSet::Iterator

Iterator to the first observable.

6.137.3 Constructor & Destructor Documentation

6.137.3.1 eos::ObservableSet::ObservableSet ()

Destructor.

6.137.3.2 eos::ObservableSet::~ObservableSet ()

Destructor.

6.137.4 Member Function Documentation

6.137.4.1 std::pair<unsigned, bool> eos::ObservableSet::add (const ObservablePtr & *observable*)

Add an observable to the vector.

Parameters:

observable The observable to be added.

Returns:

If observable is found to be an existing observable, the second value will be false. If the observable is added successfully, the second value is true, and the index of the new element in the vector is returned for use with operator[].

6.137.4.2 Iterator eos::ObservableSet::begin () const

Iterator to the first observable.

6.137.4.3 Iterator eos::ObservableSet::end () const

Iterator pointing past the last observable.

6.137.4.4 ObservablePtr& eos::ObservableSet::operator[ ] (const unsigned & *index*) const

Random access to an observable.

Parameters:

index The position in the vector.

6.137.4.5 Parameters eos::ObservableSet::parameters ()

Access to the underlying Parameters object.

6.137.4.6 unsigned eos::ObservableSet::size () const

The total number of elements.

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

- /home/alexshires/EOS/eos-next/eos/utils/observable_set.hh

6.138 eos::OneHalfPlusToOneHalfPlus Struct Reference

```
#include <baryonic.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/form-factors/baryonic.hh

6.139 eos::OneOf< Types_ > Class Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- OneOf ()
- template<typename Type_ >
  OneOf (const Type_ &value)
- Visitor_ Result_ accept_returning (Visitor_ &visitor) const

template<typename... Types_> class eos::OneOf< Types_ >

6.139.1 Constructor & Destructor Documentation

6.139.1.1 template<typename... Types_> eos::OneOf< Types_ >::OneOf () [inline]

6.139.1.2 template<typename... Types_> template<typename Type_ > eos::OneOf< Types_ >::OneOf (const Type_ & *value*) [inline]

6.139.2 Member Function Documentation

6.139.2.1 template<typename... Types_> Visitor_ Result_ eos::OneOf< Types_ >::accept_returning (Visitor_ & *visitor*) const [inline]

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.140 eos::OneOfValue< Type_, Types_ > Struct Template Reference

`#include <one-of.hh>`

Inherits OneOfValueBase< Types_...>.

Public Member Functions

- OneOfValue (const Type_ &type)
- virtual void accept (OneOfVisitor< Types_...> &visitor)

Public Attributes

- Type_ value

template<typename Type_, typename... Types_> struct eos::OneOfValue< Type_, Types_ >

6.140.1 Constructor & Destructor Documentation

6.140.1.1 template<typename Type_ , typename... Types_> eos::OneOfValue< Type_, Types_ >::OneOfValue (const Type_ & *type*) `[inline]`

6.140.2 Member Function Documentation

6.140.2.1 template<typename Type_ , typename... Types_> virtual void eos::OneOfValue< Type_, Types_ >::accept (OneOfVisitor< Types_...> & *visitor*) `[inline, virtual]`

Implements eos::OneOfValueBase< Types_...>.

6.140.3 Member Data Documentation

6.140.3.1 template<typename Type_ , typename... Types_> Type_ eos::OneOfValue< Type_, Types_ >::value

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.141 eos::OneOfValueBase< Types_ > Struct Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- virtual ~OneOfValueBase ()=0 = default
- virtual void accept (OneOfVisitor< Types_...> &)=0

template<typename... Types_> struct eos::OneOfValueBase< Types_ >

6.141.1 Constructor & Destructor Documentation

6.141.1.1 template<typename... Types_> eos::OneOfValueBase< Types_ >::~OneOfValueBase () = default [inline, pure virtual]

6.141.2 Member Function Documentation

6.141.2.1 template<typename... Types_> virtual void eos::OneOfValueBase< Types_ >::accept (OneOfVisitor< Types_...> &) [pure virtual]

Implemented in eos::OneOfValue< Type_, Types_ >.

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.142 eos::OneOfVisitor< Types_ > Struct Template Reference

```
#include <one-of.hh>
```

Inherits OneOfVisitorVisit< Types_ >.

template<typename... Types_> struct eos::OneOfVisitor< Types_ >

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.143 eos::OneOfVisitorVisit< Type_ > Struct Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- virtual void visit (Type_ &)=0

template<typename Type_> struct eos::OneOfVisitorVisit< Type_ >

6.143.1 Member Function Documentation

6.143.1.1 template<typename Type_> virtual void eos::OneOfVisitorVisit< Type_ >::visit (Type_ &) [pure virtual]

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.144 eos::OneOfVisitorWrapper< Underlying_, Result_, Types_ > Struct Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- OneOfVisitorWrapper (Underlying_ &u)

template<typename Underlying_, typename Result_, typename... Types_> struct eos::OneOfVisitorWrapper< Underlying_, Result_, Types_ >

6.144.1 Constructor & Destructor Documentation

6.144.1.1 template<typename Underlying_ , typename Result_ , typename... Types_> eos::OneOfVisitorWrapper< Underlying_, Result_, Types_ >::OneOfVisitorWrapper (Underlying_ & *u*) `[inline]`

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.145 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ > Struct Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- OneOfVisitorWrapperVisit (Underlying_ &u)

Public Attributes

- Underlying_ & underlying
- Result_ result

template<typename Visitor_, typename Underlying_, typename Result_> struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >

6.145.1 Constructor & Destructor Documentation

6.145.1.1 template<typename Visitor_ , typename Underlying_ , typename Result_ > eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >::OneOfVisitorWrapperVisit (Underlying_ & *u*) [inline]

6.145.2 Member Data Documentation

6.145.2.1 template<typename Visitor_ , typename Underlying_ , typename Result_ > Result_ eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >::result

6.145.2.2 template<typename Visitor_ , typename Underlying_ , typename Result_ > Underlying_& eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >::underlying

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.146 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...> Struct Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- OneOfVisitorWrapperVisit (Underlying_ &u)
- virtual void visit (Type_ &t)

template<typename Visitor_, typename Underlying_, typename Result_, typename Type_, typename... Rest_> struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>

6.146.1 Constructor & Destructor Documentation

6.146.1.1 template<typename Visitor_ , typename Underlying_ , typename Result_ , typename Type_ , typename... Rest_> eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>::OneOfVisitorWrapperVisit (Underlying_ & *u*) `[inline]`

6.146.2 Member Function Documentation

6.146.2.1 template<typename Visitor_ , typename Underlying_ , typename Result_ , typename Type_ , typename... Rest_> virtual void eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>::visit (Type_ & *t*) `[inline, virtual]`

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.147 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void > Struct Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- OneOfVisitorWrapperVisit (Underlying_ &u)

Public Attributes

- Underlying_ & underlying

template<typename Visitor_, typename Underlying_> struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >

6.147.1 Constructor & Destructor Documentation

6.147.1.1 template<typename Visitor_ , typename Underlying_ > eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >::OneOfVisitorWrapperVisit (Underlying_ & *u*) [inline]

6.147.2 Member Data Documentation

6.147.2.1 template<typename Visitor_ , typename Underlying_ > Underlying_& eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >::underlying

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.148 eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...> Struct Template Reference

```
#include <one-of.hh>
```

Public Member Functions

- OneOfVisitorWrapperVisit (Underlying_ &u)
- virtual void visit (Type_ &t)

template<typename Visitor_, typename Underlying_, typename Type_, typename... Rest_> struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>

6.148.1 Constructor & Destructor Documentation

6.148.1.1 template<typename Visitor_ , typename Underlying_ , typename Type_ , typename... Rest_> eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>::OneOfVisitorWrapperVisit (Underlying_ & *u*) `[inline]`

6.148.2 Member Function Documentation

6.148.2.1 template<typename Visitor_ , typename Underlying_ , typename Type_ , typename... Rest_> virtual void eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>::visit (Type_ & *t*) `[inline, virtual]`

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.149 eos::Analysis::OptimizationOptions Struct Reference

`#include <analysis.hh>`

Static Public Member Functions

- static OptimizationOptions Defaults ()

Public Attributes

- std::string algorithm

  Options are: "migrad", "minimize", "scan", "simplex" from minuit2.

- bool fix_flat_nuisance
- VerifiedRange< double > initial_step_size
- unsigned maximum_iterations
- bool mcmc_pre_run
- VerifiedRange< double > tolerance
- VerifiedRange< double > splitting_tolerance
- VerifiedRange< unsigned > strategy_level

  0 - low, 1 - medium, 2 - high precision

6.149.1 Member Function Documentation

6.149.1.1 static OptimizationOptions eos::Analysis::OptimizationOptions::Defaults () [static]

6.149.2 Member Data Documentation

6.149.2.1 std::string eos::Analysis::OptimizationOptions::algorithm

Options are: "migrad", "minimize", "scan", "simplex" from minuit2.

6.149.2.2 bool eos::Analysis::OptimizationOptions::fix_flat_nuisance

Keep the value of nuisance parameters with a flat prior fixed at the current value during optimization, to avoid flat directions that cause Migrad to fail.

6.149.2.3 VerifiedRange<double> eos::Analysis::OptimizationOptions::initial_step_size

Fraction of parameter range, in [0,1]. Useful only for simplex method

6.149.2.4 unsigned eos::Analysis::OptimizationOptions::maximum_iterations

If algorithm doesn't converge before, quit after maximum_iterations.

6.149.2.5 bool eos::Analysis::OptimizationOptions::mcmc_pre_run

If non-zero, perform MCMC iterations first, before Minuit2 is invoked from the last point of the chain.

Note:

This is only useful when call from MarkovChainSampler, further control of the chain is taken from the MarkovChainSampler::Config object.

6.149.2.6 VerifiedRange<double> eos::Analysis::OptimizationOptions::splitting_tolerance

When comparing two modes found by minuit to decide whether they correspond to the same mode, the splitting_tolerance decides how far in relative units their distance may be.

6.149.2.7 VerifiedRange<unsigned> eos::Analysis::OptimizationOptions::strategy_level

0 - low, 1 - medium, 2 - high precision

6.149.2.8 VerifiedRange<double> eos::Analysis::OptimizationOptions::tolerance

Once the algorithm has shrunk the probe simplex below this size, convergence is declared.

For minuit, it is just their tolerance parameter

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

- /home/alexshires/EOS/eos-next/eos/utils/analysis.hh

6.150 eos::Options Class Reference

`#include <options.hh>`

Inherits PrivateImplementationPattern< Options >.

Public Member Functions

Basic Functions

- Options ()
  Constructor.
- Options (const std::initializer_list< std::pair< std::string, std::string >> &options)
- ~Options ()
  Destructor.
- bool operator== (const Options &rhs) const
  Equality comparison operator.
- bool operator!= (const Options &rhs) const
  Inequality comparison operator.

Access

- const std::string & operator[ ] (const std::string &key) const
- bool has (const std::string &key) const
- void set (const std::string &key, const std::string &value="")
- std::string get (const std::string &key, const std::string &default_value="") const
- std::string as_string () const

Friends

- Options operator+ (const Options &, const Options &)

6.150.1 Detailed Description

Options keeps the set of all string options for any Observable.

6.150.2 Constructor & Destructor Documentation

6.150.2.1 eos::Options::Options ()

Constructor.

6.150.2.2 eos::Options::Options (const std::initializer_list< std::pair< std::string, std::string >> & *options*)

Constructor.

Create an instance of Kinematics with a given set of initial options.

Parameters:

options The set of initial options from which this object shall be constructed.

6.150.2.3 eos::Options::~Options ()

Destructor.

6.150.3 Member Function Documentation

6.150.3.1 std::string eos::Options::as_string () const

6.150.3.2 std::string eos::Options::get (const std::string & *key*, const std::string & *default_value* = "") const

6.150.3.3 bool eos::Options::has (const std::string & *key*) const

6.150.3.4 bool eos::Options::operator!= (const Options & *rhs*) const

Inequality comparison operator.

6.150.3.5 bool eos::Options::operator== (const Options & *rhs*) const

Equality comparison operator.

6.150.3.6 const std::string& eos::Options::operator[ ] (const std::string & *key*) const

6.150.3.7 void eos::Options::set (const std::string & *key*, const std::string & *value* = "")

6.150.4 Friends And Related Function Documentation

6.150.4.1 Options operator+ (const Options &, const Options &) `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/options.hh

6.151 eos::PopulationMonteCarloSampler::Output Struct Reference

```
#include <population_monte_carlo_sampler.hh>
```

Public Types

- typedef hdf5::Composite< hdf5::Scalar< double >, hdf5::Array< 1, double >, hdf5::Array< 1, double > > ComponentType
- typedef hdf5::Scalar< short > IgnoreType
- typedef hdf5::Array< 1, double > SampleType
- typedef hdf5::Composite< hdf5::Scalar< double >, hdf5::Scalar< double >, hdf5::Scalar< double > > StatisticsType
- typedef hdf5::Composite< hdf5::Scalar< double >, hdf5::Scalar< double > > WeightType

Static Public Member Functions

- static ComponentType component_type (const unsigned &dimension)
- static IgnoreType ignore_type ()
- static SampleType sample_type (const unsigned &dimension)
- static StatisticsType statistics_type ()
- static WeightType weight_type ()
- static std::tuple< double, std::vector< double >, std::vector< double > > component_record (const unsigned &dimension)
- static short ignore_record ()
- static std::vector< double > sample_record (const unsigned &dimension)
- static std::tuple< double, double, double > statistics_record ()
- static std::tuple< double, double > weight_record ()

6.151.1 Detailed Description

Access to the HDF5 types used in output.

6.151.2 Member Typedef Documentation

6.151.2.1 typedef hdf5::Composite<hdf5::Scalar<double>, hdf5::Array<1, double>, hdf5::Array<1, double> > eos::PopulationMonteCarloSampler::Output::ComponentType

6.151.2.2 typedef hdf5::Scalar<short> eos::PopulationMonteCarloSampler::Output::IgnoreType

6.151.2.3 typedef hdf5::Array<1, double> eos::PopulationMonteCarloSampler::Output::SampleType

6.151.2.4 typedef hdf5::Composite<hdf5::Scalar<double>, hdf5::Scalar<double>, hdf5::Scalar<double> > eos::PopulationMonteCarloSampler::Output::StatisticsType

6.151.2.5 typedef hdf5::Composite<hdf5::Scalar<double>, hdf5::Scalar<double> > eos::PopulationMonteCarloSampler::Output::WeightType

6.151.3 Member Function Documentation

6.151.3.1 static std::tuple<double, std::vector<double>, std::vector<double> > eos::PopulationMonteCarloSampler::Output::component_record (const unsigned & *dimension*) `[static]`

6.151.3.2 static ComponentType eos::PopulationMonteCarloSampler::Output::component_type (const unsigned & *dimension*) `[static]`

6.151.3.3 static short eos::PopulationMonteCarloSampler::Output::ignore_record () `[static]`

6.151.3.4 static IgnoreType eos::PopulationMonteCarloSampler::Output::ignore_type () `[static]`

6.151.3.5 static std::vector<double> eos::PopulationMonteCarloSampler::Output::sample_record (const unsigned & *dimension*) `[static]`

6.151.3.6 static SampleType eos::PopulationMonteCarloSampler::Output::sample_type (const unsigned & *dimension*) `[static]`

6.151.3.7 static std::tuple<double, double, double> eos::PopulationMonteCarloSampler::Output::statistics_record () `[static]`

6.151.3.8 static StatisticsType eos::PopulationMonteCarloSampler::Output::statistics_type () `[static]`

6.151.3.9 static std::tuple<double, double> eos::PopulationMonteCarloSampler::Output::weight_record () `[static]`

6.151.3.10 static WeightType eos::PopulationMonteCarloSampler::Output::weight_type () `[static]`

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

- /home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_sampler.hh

6.152 eos::Analysis::Output Struct Reference

```
#include <analysis.hh>
```

Public Types

- typedef hdf5::Composite< hdf5::Scalar< const char * >, hdf5::Scalar< double >, hdf5::Scalar< double >, hdf5::Scalar< int >, hdf5::Scalar< const char * > > DescriptionType

Static Public Member Functions

- static DescriptionType description_type ()
- static std::tuple< const char *, double, double, int, const char * > description_record ()

6.152.1 Member Typedef Documentation

6.152.1.1 typedef hdf5::Composite<hdf5::Scalar<const char *>, hdf5::Scalar<double>, hdf5::Scalar<double>, hdf5::Scalar<int>, hdf5::Scalar<const char *> > eos::Analysis::Output::DescriptionType

6.152.2 Member Function Documentation

6.152.2.1 static std::tuple<const char *, double, double, int, const char *> eos::Analysis::Output::description_record () [static]

6.152.2.2 static DescriptionType eos::Analysis::Output::description_type () [static]

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

- /home/alexshires/EOS/eos-next/eos/utils/analysis.hh

6.153 eos::Parameter Class Reference

`#include <parameters.hh>`

Inherits eos::Mutable.

Inherited by eos::UsedParameter.

Public Types

- typedef unsigned Id

Public Member Functions

Access & Modification of the Numeric Value

Cast a Parameter's numeric value to a double.

- virtual operator double () const
  Retrieve a Parameter's numeric value.
- virtual double operator() () const
  Retrieve a Parameter's numeric value.
- virtual const Parameter & operator= (const double &)
  Set a Parameter's numeric value.

Access to Meta Data

Retrieve the Parameter's name.

- virtual const std::string & name () const
  Retrieve the Parameter's (default) central value.
- const double & central () const
  Retrieve the Parameter's (default) central value.
- const double & max () const
  Retrieve the Parameter's (default) maximal value.
- const double & min () const
  Retrieve the Parameter's (default) minimal value.
- Id id () const
  Retrieve the Parameter's id.

Friends

- class Parameters
- struct Implementation< Parameters >

Basic Functions

Copy-constructor.

- Parameter (const Parameter &other)

  Destructor.

- ~Parameter ()

  Destructor.

- MutablePtr clone () const

  Make a copy of this Parameter as a MutablePtr.

6.153.1 Detailed Description

Parameter is the class that holds all information of one of Parameters' parameters.

6.153.2 Member Typedef Documentation

6.153.2.1 typedef unsigned eos::Parameter::Id

A unique number that identifies this parameter at run time.

6.153.3 Constructor & Destructor Documentation

6.153.3.1 eos::Parameter::Parameter (const Parameter & *other*)

Destructor.

6.153.3.2 eos::Parameter::~Parameter ()

Destructor.

6.153.4 Member Function Documentation

6.153.4.1 const double& eos::Parameter::central () const

Retrieve the Parameter's (default) central value.

6.153.4.2 MutablePtr eos::Parameter::clone () const `[virtual]`

Make a copy of this Parameter as a MutablePtr.

Implements eos::Mutable.

6.153.4.3 Id eos::Parameter::id () const

Retrieve the Parameter's id.

6.153.4.4 const double& eos::Parameter::max () const

Retrieve the Parameter's (default) maximal value.

6.153.4.5 const double& eos::Parameter::min () const

Retrieve the Parameter's (default) minimal value.

6.153.4.6 virtual const std::string& eos::Parameter::name () const [virtual]

Retrieve the Parameter's (default) central value.

Implements eos::Mutable.

6.153.4.7 virtual eos::Parameter::operator double () const [virtual]

Retrieve a Parameter's numeric value.

Implements eos::Mutable.

6.153.4.8 virtual double eos::Parameter::operator() () const [virtual]

Retrieve a Parameter's numeric value.

Implements eos::Mutable.

6.153.4.9 virtual const Parameter& eos::Parameter::operator= (const double &) [virtual]

Set a Parameter's numeric value.

Implements eos::Mutable.

6.153.5 Friends And Related Function Documentation

6.153.5.1 friend struct Implementation< Parameters > [friend]

6.153.5.2 friend class Parameters [friend]

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

- /home/alexshires/EOS/eos-next/eos/utils/parameters.hh

6.154 eos::ParameterDescription Struct Reference

```
#include <parameters.hh>
```

Public Attributes

- Parameter parameter
- double min
- double max
- bool nuisance
- bool discrete

6.154.1 Member Data Documentation

6.154.1.1 bool eos::ParameterDescription::discrete

6.154.1.2 double eos::ParameterDescription::max

6.154.1.3 double eos::ParameterDescription::min

6.154.1.4 bool eos::ParameterDescription::nuisance

6.154.1.5 Parameter eos::ParameterDescription::parameter

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

- /home/alexshires/EOS/eos-next/eos/utils/parameters.hh

6.155 eos::ParameterRange Struct Reference

```
#include <parameters.hh>
```

Public Attributes

- double min
- double max

6.155.1 Member Data Documentation

6.155.1.1 double eos::ParameterRange::max

6.155.1.2 double eos::ParameterRange::min

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

- /home/alexshires/EOS/eos-next/eos/utils/parameters.hh

6.156 eos::Parameters Class Reference

`#include <parameters.hh>`

Inherits PrivateImplementationPattern< Parameters >.

Public Member Functions

- bool operator!= (const Parameters &rhs) const

Parameter access

- Parameter declare (const std::string &name, const double &value=0.0)
- void set (const std::string &name, const double &value)
- Parameter operator[ ] (const std::string &name) const
- Parameter operator[ ] (const unsigned &id) const

Friends

- class Parameter
- struct Implementation< Parameter >
- struct Implementation< Parameters >

Iteration

- typedef WrappedForwardIterator< IteratorTag, Parameter > Iterator
- Iterator begin () const
- Iterator end () const

Basic Functions

- static Parameters Defaults ()
- Parameters clone () const
- ~Parameters ()

6.156.1 Detailed Description

Parameters keeps the set of all numeric parameters for any Observable.

Access to any Parameter or their values is coherent, i.e., changes to a Parameter object will propagate to every other object with the same parent Parameters and which handle the same parameter by name.

6.156.2 Member Typedef Documentation

6.156.2.1 typedef WrappedForwardIterator<IteratorTag, Parameter> eos::Parameters::Iterator

6.156.3 Constructor & Destructor Documentation

6.156.3.1 eos::Parameters::~Parameters ()

Destructor.

6.156.4 Member Function Documentation

6.156.4.1 Iterator eos::Parameters::begin () const

6.156.4.2 Parameters eos::Parameters::clone () const

Constructor (private).

Creates an instance of Parameters from a given implementation. To create an instance of Parameters, see one of the named constructors.

Parameters:

impl Implementation from which the Parameters object shall be constructed from.

6.156.4.3 Parameter eos::Parameters::declare (const std::string & *name*, const double & *value* = `0.0`)

Declare a new parameter.

Parameters:

name Name of the new parameter to be declared.

value (Optional) value for the new parameter.

6.156.4.4 static Parameters eos::Parameters::Defaults () `[static]`

Named constructor.

Creates an instance of Parameters with default values filled in.

6.156.4.5 Iterator eos::Parameters::end () const

6.156.4.6 bool eos::Parameters::operator!= (const Parameters & *rhs*) const

Compare two instances of Parameters on inequality of their underlying implementations.

Parameters:

rhs The right hand side of the binary != operator.

6.156.4.7 Parameter eos::Parameters::operator[ ] (const unsigned & *id*) const

Retrieve a parameter's Parameter object by id.

Parameters:

id The id of the Parameter that shall be retrived.

6.156.4.8 Parameter eos::Parameters::operator[ ] (const std::string & *name*) const

Retrieve a parameter's Parameter object by name.

Parameters:

name The name of the Parameter that shall be retrieved.

6.156.4.9 void eos::Parameters::set (const std::string & *name*, const double & *value*)

Set a parameter's numeric value.

Parameters:

name The name of the parameter whose numeric value shall be changed.

value The parameter's new numeric value.

6.156.5 Friends And Related Function Documentation

6.156.5.1 friend struct Implementation< Parameter > `[friend]`

6.156.5.2 friend struct Implementation< Parameters > `[friend]`

6.156.5.3 friend class Parameter `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/parameters.hh

6.157 eos::ParameterUser Class Reference

`#include <parameters.hh>`

Inherited by eos::BMesonProperties, eos::BToDilepton, eos::BToKDilepton< LargeRecoil >, eos::BToKDilepton< LowRecoil >, eos::BToKstarDilepton< LargeRecoil >, eos::BToKstarDilepton< LowRecoil >, eos::BToKstarGamma, eos::BToPiLeptonNeutrino, eos::BToXsDilepton< HLMW2005 >, eos::BToXsGamma< Minimal >, eos::BToXsGamma< NLO >, eos::FormFactors< OneHalfPlusToOneHalfPlus >, eos::FormFactors< PToP >, eos::FormFactors< PToV >, eos::LambdaBToLambdaDilepton< LargeRecoil >, eos::LambdaBToLambdaDilepton< LowRecoil >, eos::Model, eos::Observable, and eos::PionLCDAs.

Public Member Functions

Access

- void uses (const Parameter::Id &id)
- void uses (const ParameterUser &user)

Protected Attributes

- std::set< Parameter::Id > _ids

Iteration over ids

- typedef WrappedForwardIterator< ConstIteratorTag, const Parameter::Id > ConstIterator
- ConstIterator begin () const
- ConstIterator end () const

6.157.1 Detailed Description

Base class for all users of Parameter objects.

6.157.2 Member Typedef Documentation

6.157.2.1 typedef WrappedForwardIterator<ConstIteratorTag, const Parameter::Id> eos::ParameterUser::ConstIterator

6.157.3 Member Function Documentation

6.157.3.1 ConstIterator eos::ParameterUser::begin () const

6.157.3.2 ConstIterator eos::ParameterUser::end () const

6.157.3.3 void eos::ParameterUser::uses (const ParameterUser & *user*)

Copy parameter ids of another ParameterUser to our list of used ids.

Parameters:

user The other ParameterUser whose ids we are going to copy.

6.157.3.4 void eos::ParameterUser::uses (const Parameter::Id & *id*)

Add a given parameter id to our list of used ids.

Parameters:

id The parameter id that we use.

6.157.4 Member Data Documentation

6.157.4.1 std::set<Parameter::Id> eos::ParameterUser::_ids [protected]

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

- /home/alexshires/EOS/eos-next/eos/utils/parameters.hh

6.158 eos::PionLCDAs Class Reference

`#include <pi-lcdas.hh>`

Inherits eos::ParameterUser, and PrivateImplementationPattern< PionLCDAs >.

Public Member Functions

- PionLCDAs (const Parameters &, const Options &)
- ~PionLCDAs ()
- double a2pi (const double &mu) const
- double a4pi (const double &mu) const
- double mupi (const double &mu) const
- double f3pi (const double &mu) const
- double eta3pi (const double &mu) const
- double omega3pi (const double &mu) const
- double deltapipi (const double &mu) const
- double omega4pi (const double &mu) const
- double phi (const double &u, const double &mu) const
- double phi3p (const double &u, const double &mu) const
- double phi3s (const double &u, const double &mu) const
- double phi3s_d1 (const double &u, const double &mu) const
- double phi4 (const double &u, const double &mu) const
- double phi4_d1 (const double &u, const double &mu) const
- double phi4_d2 (const double &u, const double &mu) const
- double psi4 (const double &u, const double &mu) const
- double psi4_i (const double &u, const double &mu) const
- Diagnostics diagnostics () const

6.158.1 Constructor & Destructor Documentation

6.158.1.1 eos::PionLCDAs::PionLCDAs (const Parameters &, const Options &)

6.158.1.2 eos::PionLCDAs::~PionLCDAs ()

6.158.2 Member Function Documentation

6.158.2.1 double eos::PionLCDAs::a2pi (const double & *mu*) const

6.158.2.2 double eos::PionLCDAs::a4pi (const double & *mu*) const

6.158.2.3 double eos::PionLCDAs::deltapipi (const double & *mu*) const

6.158.2.4 Diagnostics eos::PionLCDAs::diagnostics () const

6.158.2.5 double eos::PionLCDAs::eta3pi (const double & *mu*) const

6.158.2.6 double eos::PionLCDAs::f3pi (const double & *mu*) const

6.158.2.7 double eos::PionLCDAs::mupi (const double & *mu*) const

6.158.2.8 double eos::PionLCDAs::omega3pi (const double & *mu*) const

6.158.2.9 double eos::PionLCDAs::omega4pi (const double & *mu*) const

6.158.2.10 double eos::PionLCDAs::phi (const double & *u*, const double & *mu*) const

6.158.2.11 double eos::PionLCDAs::phi3p (const double & *u*, const double & *mu*) const

6.158.2.12 double eos::PionLCDAs::phi3s (const double & *u*, const double & *mu*) const

6.158.2.13 double eos::PionLCDAs::phi3s_d1 (const double & *u*, const double & *mu*) const

6.158.2.14 double eos::PionLCDAs::phi4 (const double & *u*, const double & *mu*) const

6.158.2.15 double eos::PionLCDAs::phi4_d1 (const double & *u*, const double & *mu*) const

6.158.2.16 double eos::PionLCDAs::phi4_d2 (const double & *u*, const double & *mu*) const

6.158.2.17 double eos::PionLCDAs::psi4 (const double & *u*, const double & *mu*) const

6.158.2.18 double eos::PionLCDAs::psi4_i (const double & *u*, const double & *mu*) const

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

- /home/alexshires/EOS/eos-next/eos/form-factors/pi-lcdas.hh

6.159 eos::PopulationMonteCarloSampler Class Reference

`#include <population_monte_carlo_sampler.hh>`

Inherits PrivateImplementationPattern< PopulationMonteCarloSampler >.

Classes

- struct Config
- struct Output
- struct Status

Public Member Functions

Basic Functions

- PopulationMonteCarloSampler (const Analysis &analysis, const PopulationMonteCarloSampler::Config &config)
- PopulationMonteCarloSampler (const Analysis &analysis, const hdf5::File &file, const PopulationMonteCarloSampler::Config &config, const bool &update=false)
- ~PopulationMonteCarloSampler ()

  Destructor.

Sampling

- void calculate_weights (const std::string &sample_file, const unsigned &min_index, const unsigned &max_index) const
- const PopulationMonteCarloSampler::Config & config () const

  Retrieve the configuration from which this sampler was constructed.

- void draw_samples ()
- void run ()

  Start the Markov chain sampling.

- const PopulationMonteCarloSampler::Status & status () const

  Retrieve the current status.

- bool status (const PopulationMonteCarloSampler::Status &new_status, bool check_convergence=false)
- static void read_samples (const std::string &sample_file, const std::string &base, const unsigned &min, const unsigned &max, std::vector< std::vector< double >> &samples)

6.159.1 Constructor & Destructor Documentation

6.159.1.1 eos::PopulationMonteCarloSampler::PopulationMonteCarloSampler (const Analysis & *analysis*, const PopulationMonteCarloSampler::Config & *config*)

Constructor.

Parameters:

analysis The analysis for which we draw samples.

config The configuration of the samples.

6.159.1.2 eos::PopulationMonteCarloSampler::PopulationMonteCarloSampler (const Analysis & *analysis*, const hdf5::File & *file*, const PopulationMonteCarloSampler::Config & *config*, const bool & *update* = `false`)

Initialize the PMC from a HDF5 file.

Several kinds of input files are (automatically) recognized:

a) GlobalLocal proposal function, stored in file. b) MCMC prerun, used for hierarchical clustering. c) Serialized status of a PMC proposal function from a previous PMC run.

All options regarding construction of the PMC in config are ignored.

6.159.1.3 eos::PopulationMonteCarloSampler::∼PopulationMonteCarloSampler ()

Destructor.

6.159.2 Member Function Documentation

6.159.2.1 void eos::PopulationMonteCarloSampler::calculate_weights (const std::string & *sample_file*, const unsigned & *min_index*, const unsigned & *max_index*) const

Calculate the posterior and importance weights for the range of parameter samples given by min and max indices from the sample file.

6.159.2.2 const PopulationMonteCarloSampler::Config& eos::PopulationMonteCarloSampler::config () const

Retrieve the configuration from which this sampler was constructed.

6.159.2.3 void eos::PopulationMonteCarloSampler::draw_samples ()

Draw parameter samples from the proposal density and store them, along with the full status of all components.

Parameters:

output_file Name of the HDF5 output file

6.159.2.4 static void eos::PopulationMonteCarloSampler::read_samples (const std::string & *sample_file*, const std::string & *base*, const unsigned & *min*, const unsigned & *max*, std::vector< std::vector< double >> & *samples*) `[static]`

Read in a slice of samples from a previous PMC dump.

Parameters:

sample_file Name of HDF5 file containing the samples.

base Directory name within HDF5 file.

min First element to parse.

max Index of one-past last element to parse.

samples Upon return, contains all samples stored. When passing in, there must be at least one sample in there to convey the parameter dimension

6.159.2.5 void eos::PopulationMonteCarloSampler::run ()

Start the Markov chain sampling.

6.159.2.6 bool eos::PopulationMonteCarloSampler::status (const PopulationMonteCarloSampler::Status & *new_status*, bool *check_convergence* = `false`)

Set the current status.

Parameters:

new_status The new status.

check_convergence If true, compute convergence based on the new status and return true if converged. This option is for testing only!

Returns:

True

6.159.2.7 const PopulationMonteCarloSampler::Status& eos::PopulationMonteCarloSampler::status () const

Retrieve the current status.

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

- /home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_sampler.hh

6.160 eos::MarkovChainSampler::PreRunInfo Struct Reference

```
#include <markov_chain_sampler.hh>
```

Public Attributes

- bool converged

  Convergence status after performing the prerun.

- unsigned iterations

  The number of iterations that were performed in the prerun.

- unsigned iterations_at_convergence

  The number of iterations after which convergence was declared.

- double rvalue_posterior

  R-value of posterior.

- std::vector< double > rvalue_parameters

  R-values of individual parameters.

6.160.1 Detailed Description

Holds convergence information of the prerun.

6.160.2 Member Data Documentation

6.160.2.1 bool eos::MarkovChainSampler::PreRunInfo::converged

Convergence status after performing the prerun.

6.160.2.2 unsigned eos::MarkovChainSampler::PreRunInfo::iterations

The number of iterations that were performed in the prerun.

6.160.2.3 unsigned eos::MarkovChainSampler::PreRunInfo::iterations_at_convergence

The number of iterations after which convergence was declared.

6.160.2.4 std::vector<double> eos::MarkovChainSampler::PreRunInfo::rvalue_parameters

R-values of individual parameters.

6.160.2.5 double eos::MarkovChainSampler::PreRunInfo::rvalue_posterior

R-value of posterior.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain_sampler.hh

6.161 eos::PriorSampler Class Reference

`#include <prior_sampler.hh>`

Inherits PrivateImplementationPattern< PriorSampler >.

Classes

- struct Config

Public Types

- typedef hdf5::Array< 1, double > ObservablesType
- typedef std::vector< std::vector< double > > SamplesList

Public Member Functions

- PriorSampler (const ObservableSet &observables, const Config &config)
- ~PriorSampler ()
- bool add (const LogPriorPtr &prior)
- bool add (const ObservablePtr &observable)
- void run ()
- void run (const SamplesList &samples, const std::vector< ParameterDescription > &)

Static Public Member Functions

- static ObservablesType observables_type (const unsigned &dimension)

6.161.1 Detailed Description

Perform simple uncertainty propagation by defining parameters to be varied, and the observables whose variation one is interested in. Parameter values are sampled directly from 1D priors. All observable values are stored to disk.

6.161.2 Member Typedef Documentation

6.161.2.1 typedef hdf5::Array<1, double> eos::PriorSampler::ObservablesType

6.161.2.2 typedef std::vector<std::vector<double> > eos::PriorSampler::SamplesList

6.161.3 Constructor & Destructor Documentation

6.161.3.1 eos::PriorSampler::PriorSampler (const ObservableSet & *observables*, const Config & *config*)

Constructor.

Parameters:

observables list of observables, can be extended through add().

config configuration options.

6.161.3.2 eos::PriorSampler::~PriorSampler ()

Destructor.

6.161.4 Member Function Documentation

6.161.4.1 bool eos::PriorSampler::add (const ObservablePtr & *observable*)

Add an observable for which the the uncertainty is to be evaluated.

Returns:

false if identical to an existing observable.

6.161.4.2 bool eos::PriorSampler::add (const LogPriorPtr & *prior*)

Add a parameter with its prior to sample from.

Returns:

false if prior was not added successfully.

6.161.4.3 static ObservablesType eos::PriorSampler::observables_type (const unsigned & *dimension*) `[static]`

6.161.4.4 void eos::PriorSampler::run (const SamplesList & *samples*, const std::vector< ParameterDescription > &)

Calculate observables at the given parameter samples. The order and meaning of parameters in a sample are specified by the parameter descriptions.

Note:

No new samples are drawn from the priors.

6.161.4.5 void eos::PriorSampler::run ()

Start the sampling process and store the calculated observable values to disk.

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

- /home/alexshires/EOS/eos-next/eos/utils/prior_sampler.hh

6.162 eos::PrivateImplementationPattern< T_ > Class Template Reference

```
#include <private_implementation_pattern.hh>
```

Public Member Functions

- PrivateImplementationPattern (Implementation< T_ > *imp)
- ~PrivateImplementationPattern ()

Protected Attributes

- std::shared_ptr< Implementation< T_ > > _imp

template<typename T_> class eos::PrivateImplementationPattern< T_ >

6.162.1 Constructor & Destructor Documentation

6.162.1.1 template<typename T_> eos::PrivateImplementationPattern< T_ >::PrivateImplementationPattern (Implementation< T_ > * *imp*) [inline, explicit]

6.162.1.2 template<typename T_ > eos::PrivateImplementationPattern< T_ >::~PrivateImplementationPattern () [inline]

6.162.2 Member Data Documentation

6.162.2.1 template<typename T_> std::shared_ptr<Implementation<T_> > eos::PrivateImplementationPattern< T_ >::_imp [protected]

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

- /home/alexshires/EOS/eos-next/eos/utils/private_implementation_pattern.hh
- /home/alexshires/EOS/eos-next/eos/utils/private_implementation_pattern-impl.hh

6.163 eos::MarkovChain::ProposalFunction Struct Reference

`#include <markov_chain.hh>`

Inherited by eos::proposal_functions::BlockDecomposition, eos::proposal_functions::GlobalLocal, and eos::proposal_functions::Multivariate.

Public Member Functions

- virtual ~ProposalFunction ()=0

  (virtual) Destructor.

- virtual void adapt (const MarkovChain::State::Iterator &begin, const MarkovChain::State::Iterator &end, const double &efficiency, const double &efficiency_min, const double &efficiency_max)=0
- virtual ProposalFunctionPtr clone () const =0

  Create independent copy.

- virtual void dump_state (hdf5::File &file, const std::string &data_set_base_name) const =0

  Store state in the file under the given base name.

- virtual double evaluate (const MarkovChain::State &x, const MarkovChain::State &y) const =0

  Evaluate the density to propose x given y.

- virtual void propose (MarkovChain::State &x, const MarkovChain::State &y, gsl_rng *rng) const =0

  Obtain from the density a proposal x given y.

6.163.1 Detailed Description

Interface to proposal functions for a MarkovChain.

6.163.2 Constructor & Destructor Documentation

6.163.2.1 virtual eos::MarkovChain::ProposalFunction::~ProposalFunction () [pure virtual]

(virtual) Destructor.

6.163.3 Member Function Documentation

6.163.3.1 virtual void eos::MarkovChain::ProposalFunction::adapt (const MarkovChain::State::Iterator & *begin*, const MarkovChain::State::Iterator & *end*, const double & *efficiency*, const double & *efficiency_min*, const double & *efficiency_max*) [pure virtual]

Adapt the proposal function to the chain's current state and history.

Note:

adapt() always uses the full history passed as an argument. If only a subset of an existing history is to be used, the caller is responsible for removing the unneeded parts from history before calling adapt().

Parameters:

history

Implemented in eos::proposal_functions::GlobalLocal, eos::proposal_functions::Multivariate, and eos::proposal_functions::BlockDecomposition.

6.163.3.2 virtual ProposalFunctionPtr eos::MarkovChain::ProposalFunction::clone () const [pure virtual]

Create independent copy.

Implemented in eos::proposal_functions::GlobalLocal, eos::proposal_functions::MultivariateGaussian, eos::proposal_functions::MultivariateStudentT, and eos::proposal_functions::BlockDecomposition.

6.163.3.3 virtual void eos::MarkovChain::ProposalFunction::dump_state (hdf5::File & *file*, const std::string & *data_set_base_name*) const [pure virtual]

Store state in the file under the given base name.

Implemented in eos::proposal_functions::GlobalLocal, eos::proposal_functions::MultivariateGaussian, eos::proposal_functions::MultivariateStudentT, and eos::proposal_functions::BlockDecomposition.

6.163.3.4 virtual double eos::MarkovChain::ProposalFunction::evaluate (const MarkovChain::State & *x*, const MarkovChain::State & *y*) const [pure virtual]

Evaluate the density to propose x given y.

Implemented in eos::proposal_functions::GlobalLocal, eos::proposal_functions::MultivariateGaussian, eos::proposal_functions::MultivariateStudentT, and eos::proposal_functions::BlockDecomposition.

6.163.3.5 virtual void eos::MarkovChain::ProposalFunction::propose (MarkovChain::State & *x*, const MarkovChain::State & *y*, gsl_rng * *rng*) const [pure virtual]

Obtain from the density a proposal x given y.

Implemented in eos::proposal_functions::GlobalLocal, eos::proposal_functions::MultivariateGaussian, eos::proposal_functions::MultivariateStudentT, and eos::proposal_functions::BlockDecomposition.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh

6.164 eos::PToP Struct Reference

```
#include <mesonic.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh

6.165 eos::PToV Struct Reference

```
#include <mesonic.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh

6.166 eos::QCD Class Reference

```
#include <qcd.hh>
```

Public Types

- typedef std::array< double, 4 > AnomalousMassDimension

  The four leading coefficients of the α_s expansion of the quark mass' anomalous dimension γ_m.

- typedef std::array< double, 4 > BetaFunction

  The four leading coefficient of the α_s expansion of the beta function of QCD.

Static Public Member Functions

- static double alpha_s (const double &mu, const double &alpha_s_0, const double &mu_0, const BetaFunction &beta)
- static double m_q_msbar (const double &m_q, const double &alpha_s_0, const double &alpha_s_-mu, const BetaFunction &beta, const AnomalousMassDimension &gamma_m)
- static double m_q_msbar (const double &m_q_pole, const double &alpha_s, const double &nf)
- static double m_q_pole (const double &m_q, const double &alpha_s, const double &nf)
- static double m_q_ps (const double &m_q, const double &alpha_s, const double &mu_f, const double &nf, const BetaFunction &beta)
- static double m_q_kin (const double &m_q, const double &alpha_s, const double &mu, const BetaFunction &beta)

Static Public Attributes

- static const double casimir_f
- static const double casimir_a

  - static const BetaFunction beta_function_nf_6
  - static const BetaFunction beta_function_nf_5
  - static const AnomalousMassDimension gamma_m_nf_5
  - static const BetaFunction beta_function_nf_4
  - static const AnomalousMassDimension gamma_m_nf_4
  - static const BetaFunction beta_function_nf_3
  - static const AnomalousMassDimension gamma_m_nf_3

6.166.1 Member Typedef Documentation

6.166.1.1 typedef std::array<double, 4> eos::QCD::AnomalousMassDimension

The four leading coefficients of the α_s expansion of the quark mass' anomalous dimension γ_m.

6.166.1.2 typedef std::array<double, 4> eos::QCD::BetaFunction

The four leading coefficient of the α_s expansion of the beta function of QCD.

6.166.2 Member Function Documentation

6.166.2.1 static double eos::QCD::alpha_s (const double & *mu*, const double & *alpha_s_0*, const double & *mu_0*, const BetaFunction & *beta*) [static]

Calculate RGE running of strong coupling alpha_s from scale mu_0 down to scale mu in the MSbar scheme.

Calculation according to [CKS2000].

Parameters:

mu scale at which alpha_s shall be evaluated

alpha_s_0 alpha_s at the initial scale mu_0

mu_0 initial scale mu_0

beta parameters of QCD beta function that control the running

6.166.2.2 static double eos::QCD::m_q_kin (const double & *m_q*, const double & *alpha_s*, const double & *mu*, const BetaFunction & *beta*) [static]

Calculate the shift from MSbar scheme to kinetic scheme (kin mass)

Calculation according to [BBMU2003].

Parameters:

m_q quark mass in the MSbar scheme at the scale m_q

alpha_s alpha_s at the scale m_q

mu scale at which the mass shall be evaluated

beta parameters of QCD beta function that control the calculation

6.166.2.3 static double eos::QCD::m_q_msbar (const double & *m_q_pole*, const double & *alpha_s*, const double & *nf*) [static]

Calculate the shift from pole mass scheme to MSbar.

Calculation according to [MvR1999].

Parameters:

m_q_pole quark mass in the pole mass scheme

alpha_s alpha_s at scale mu = m_q_pole

nf number of active QCD flavors that control the calculation

6.166.2.4 static double eos::QCD::m_q_msbar (const double & *m_q*, const double & *alpha_s_0*, const double & *alpha_s_mu*, const BetaFunction & *beta*, const AnomalousMassDimension & *gamma_m*) [static]

Calculate RGE running of quark mass m_q in the MSbar scheme.

Calculation according to [CKS2000].

Parameters:

m_q quark mass in the MSbar scheme at the scale m_q

alpha_s_0 alpha_s at the scale mu_0 = m_q

alpha_s_mu alpha_s at the scale mu

beta parameters of QCD beta function that control the running

gamma_m parameters of QCD anomalous mass dimension that control the running

6.166.2.5 static double eos::QCD::m_q_pole (const double & *m_q*, const double & *alpha_s*, const double & *nf*) `[static]`

Calculate the shift from MSbar scheme to pole mass.

Calculation according to [CERN2003-002].

Parameters:

m_q quark mass in the MSbar scheme at the scale m_q

alpha_s alpha_s at the scale m_q

nf number of active QCD flavors that control the calculation

6.166.2.6 static double eos::QCD::m_q_ps (const double & *m_q*, const double & *alpha_s*, const double & *mu_f*, const double & *nf*, const BetaFunction & *beta*) `[static]`

Calculate the shift from MSbar scheme to potential-Subtracted mass (PS mass).

Calculation according to [B1998].

Parameters:

m_q quark mass in the MSbar scheme at the scale m_q

alpha_s alpha_s at the scale m_q

mu_f Factorization scale

nf number of active QCD flavors that control the calculation

beta parameters of QCD beta function that control the calculation

6.166.3 Member Data Documentation

6.166.3.1 const BetaFunction eos::QCD::beta_function_nf_3 `[static]`

The 3-flavor QCD constants.

6.166.3.2 const BetaFunction eos::QCD::beta_function_nf_4 `[static]`

The 4-flavor-QCD constants.

6.166.3.3 const BetaFunction eos::QCD::beta_function_nf_5 `[static]`

The 5-flavor-QCD constants.

6.166.3.4 const BetaFunction eos::QCD::beta_function_nf_6 [static]

The 6-flavor-QCD constants.

6.166.3.5 const double eos::QCD::casimir_a [static]

The quadratic casimir operator for the adjoint representation SU(3).

6.166.3.6 const double eos::QCD::casimir_f [static]

The quadratic casimir operator for the fundamental representation of SU(3).

6.166.3.7 const AnomalousMassDimension eos::QCD::gamma_m_nf_3 [static]

The 6-flavor-QCD constants.

6.166.3.8 const AnomalousMassDimension eos::QCD::gamma_m_nf_4 [static]

The 6-flavor-QCD constants.

6.166.3.9 const AnomalousMassDimension eos::QCD::gamma_m_nf_5 [static]

The 6-flavor-QCD constants.

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

- /home/alexshires/EOS/eos-next/eos/utils/qcd.hh

6.167 eos::QCDFIntegrals Class Reference

```
#include <qcdf_integrals.hh>
```

Classes

- struct Results

Static Public Member Functions

- static Results photon_bottom_case (const double &m_b, const double &m_B, const double &m_V, const double &mu, const double &a_1_perp, const double &a_2_perp, const double &a_1_parallel, const double &a_2_parallel)
- static Results photon_charm_case (const double &m_c, const double &m_B, const double &m_V, const double &mu, const double &a_1_perp, const double &a_2_perp, const double &a_1_parallel, const double &a_2_parallel)
- static Results photon_massless_case (const double &m_B, const double &m_V, const double &mu, const double &a_1_perp, const double &a_2_perp, const double &a_1_parallel, const double &a_-2_parallel)
- static Results dilepton_bottom_case (const double &s, const double &m_b, const double &m_B, const double &m_V, const double &mu, const double &a_1_perp, const double &a_2_perp, const double &a_1_parallel, const double &a_2_parallel)
- static Results dilepton_charm_case (const double &s, const double &m_c, const double &m_B, const double &m_V, const double &mu, const double &a_1_perp, const double &a_2_perp, const double &a_1_parallel, const double &a_2_parallel)
- static Results dilepton_massless_case (const double &s, const double &m_B, const double &m_V, const double &mu, const double &a_1_perp, const double &a_2_perp, const double &a_1_parallel, const double &a_2_parallel)

6.167.1 Member Function Documentation

6.167.1.1 static Results eos::QCDFIntegrals::dilepton_bottom_case (const double & *s*, const double & *m_b*, const double & *m_B*, const double & *m_V*, const double & *mu*, const double & *a_1_perp*, const double & *a_2_perp*, const double & *a_1_parallel*, const double & *a_2_parallel*) `[static]`

Return all QCDF Integrals for a b quark-antiquark loop, according to [vD2011].

Parameters:

s Invariant quark-antiquark mass square.

m_b Pole mass of the b quark.

m_B Mass of the parent B quark.

mu Renormalization scale mu.

a_1_perp First Gegenbauer moment for the perpendicular amplitude.

a_2_perp Second Gegenbauer moment for the perpendicular amplitude.

a_1_parallel First Gegenbauer moment for the paralle amplitude.

a_2_parallel Second Gegenbauer moment for the parallel amplitude.

6.167.1.2 static Results eos::QCDFIntegrals::dilepton_charm_case (const double & *s*, const double & *m_c*, const double & *m_B*, const double & *m_V*, const double & *mu*, const double & *a_1_perp*, const double & *a_2_perp*, const double & *a_1_parallel*, const double & *a_2_parallel*) `[static]`

Return all QCDF Integrals for a c quark-antiquark loop, according to [vD2011].

Parameters:

s Invariant quark-antiquark mass square.

m_c Pole mass of the c quark.

m_B Mass of the parent B quark.

mu Renormalization scale mu.

a_1_perp First Gegenbauer moment for the perpendicular amplitude.

a_2_perp Second Gegenbauer moment for the perpendicular amplitude.

a_1_parallel First Gegenbauer moment for the paralle amplitude.

a_2_parallel Second Gegenbauer moment for the parallel amplitude.

6.167.1.3 static Results eos::QCDFIntegrals::dilepton_massless_case (const double & *s*, const double & *m_B*, const double & *m_V*, const double & *mu*, const double & *a_1_perp*, const double & *a_2_perp*, const double & *a_1_parallel*, const double & *a_2_parallel*) `[static]`

Return all QCDF Integrals for a u,d,s (i.e. massless) quark-antiquark loop for massless quarks, according to [vD2011].

Parameters:

s Invariant quark-antiquark mass square.

m_B Mass of the parent B quark.

mu Renormalization scale.

a_1_perp First Gegenbauer moment for the perpendicular amplitude.

a_2_perp Second Gegenbauer moment for the perpendicular amplitude.

a_1_parallel First Gegenbauer moment for the paralle amplitude.

a_2_parallel Second Gegenbauer moment for the parallel amplitude.

6.167.1.4 static Results eos::QCDFIntegrals::photon_bottom_case (const double & *m_b*, const double & *m_B*, const double & *m_V*, const double & *mu*, const double & *a_1_perp*, const double & *a_2_perp*, const double & *a_1_parallel*, const double & *a_2_parallel*) `[static]`

Return all QCDF Integrals for a b quark-antiquark loop with s = 0, according to [vD2011].

Parameters:

m_b Pole mass of the b quark.

m_B Mass of the parent B quark.

mu Renormalization scale mu.

a_1_perp First Gegenbauer moment for the perpendicular amplitude.

a_2_perp Second Gegenbauer moment for the perpendicular amplitude.

a_1_parallel First Gegenbauer moment for the paralle amplitude.

a_2_parallel Second Gegenbauer moment for the parallel amplitude.

6.167.1.5 static Results eos::QCDFIntegrals::photon_charm_case (const double & *m_c*, const double & *m_B*, const double & *m_V*, const double & *mu*, const double & *a_1_perp*, const double & *a_2_perp*, const double & *a_1_parallel*, const double & *a_2_parallel*) `[static]`

Return all QCDF Integrals for a c quark-antiquark loop with s = 0, according to [vD2011].

Parameters:

m_c Pole mass of the c quark.

m_B Mass of the parent B quark.

mu Renormalization scale mu.

a_1_perp First Gegenbauer moment for the perpendicular amplitude.

a_2_perp Second Gegenbauer moment for the perpendicular amplitude.

a_1_parallel First Gegenbauer moment for the paralle amplitude.

a_2_parallel Second Gegenbauer moment for the parallel amplitude.

6.167.1.6 static Results eos::QCDFIntegrals::photon_massless_case (const double & *m_B*, const double & *m_V*, const double & *mu*, const double & *a_1_perp*, const double & *a_2_perp*, const double & *a_1_parallel*, const double & *a_2_parallel*) `[static]`

Return all QCDF Integrals for a u,d,s (i.e. massless) quark-antiquark loops with s = 0, according to [vD2011].

Parameters:

m_B Mass of the parent B quark.

mu Renormalization scale mu.

a_1_perp First Gegenbauer moment for the perpendicular amplitude.

a_2_perp Second Gegenbauer moment for the perpendicular amplitude.

a_1_parallel First Gegenbauer moment for the paralle amplitude.

a_2_parallel Second Gegenbauer moment for the parallel amplitude.

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/qcdf_integrals.hh

6.168 eos::RandomNumberGenerator Class Reference

`#include <random_number_generator.hh>`

Inherits PrivateImplementationPattern< RandomNumberGenerator >.

Public Member Functions

- double operator() ()

  Obtain a pseudo-random number in the range [0.0, 1.0).

- double max () const

  Return the maximal value that can be drawn.

- double min () const

  Return the minimal value that can be drawn.

Basic functions.

- RandomNumberGenerator (const unsigned long &seed=0ul)
- ~RandomNumberGenerator ()

  Destructor.

6.168.1 Detailed Description

Generate pseudo-random floating point numbers in the range 0 ... 1.

RandomNumberGenerator keeps its state/seed on copying.

6.168.2 Constructor & Destructor Documentation

6.168.2.1 eos::RandomNumberGenerator::RandomNumberGenerator (const unsigned long & *seed* = `0ul`)

Constructor.

Parameters:

seed The numerical value used to seed the RNG.

6.168.2.2 eos::RandomNumberGenerator::~RandomNumberGenerator ()

Destructor.

6.168.3 Member Function Documentation

6.168.3.1 double eos::RandomNumberGenerator::max () const `[inline]`

Return the maximal value that can be drawn.

6.168.3.2 double eos::RandomNumberGenerator::min () const [inline]

Return the minimal value that can be drawn.

6.168.3.3 double eos::RandomNumberGenerator::operator() ()

Obtain a pseudo-random number in the range [0.0, 1.0).

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

- /home/alexshires/EOS/eos-next/eos/utils/random_number_generator.hh

6.169 eos::implementation::ResultOf< Result_(∗)(Args_...)> Struct Template Reference

```
#include <memoise.hh>
```

Public Types

- typedef Result_ Type

template<typename Result_, typename... Args_> struct eos::implementation::ResultOf< Result_-(∗)(Args_...)>

6.169.1 Member Typedef Documentation

6.169.1.1 template<typename Result_ , typename... Args_> typedef Result_ eos::implementation::ResultOf< Result_(∗)(Args_...)>::Type

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

- /home/alexshires/EOS/eos-next/eos/utils/memoise.hh

6.170 eos::implementation::ResultOf< Result_(Class_::*)(Args_...)> Struct Template Reference

```
#include <memoise.hh>
```

Public Types

- typedef Result_ Type

template<typename Result_, typename Class_, typename... Args_> struct eos::implementation::ResultOf< Result_(Class_::*)(Args_...)>

6.170.1 Member Typedef Documentation

6.170.1.1 template<typename Result_ , typename Class_ , typename... Args_> typedef Result_ eos::implementation::ResultOf< Result_(Class_::*)(Args_...)>::Type

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

- /home/alexshires/EOS/eos-next/eos/utils/memoise.hh

6.171 eos::QCDFIntegrals::Results Struct Reference

```
#include <qcdf_integrals.hh>
```

Public Attributes

- complex< double > j0_perp
- complex< double > j0bar_perp
- complex< double > j1_perp
- complex< double > j2_perp
- complex< double > j4_perp
- complex< double > j5_perp
- complex< double > j6_perp
- double j7_perp
- complex< double > j0_parallel
- complex< double > j1_parallel
- complex< double > j3_parallel
- complex< double > j4_parallel
- complex< double > jtilde1_perp
- complex< double > jtilde2_parallel

6.171.1 Detailed Description

Ensemble of the individual integral results.

6.171.2 Member Data Documentation

6.171.2.1 complex<double> eos::QCDFIntegrals::Results::j0_parallel

6.171.2.2 complex<double> eos::QCDFIntegrals::Results::j0_perp

6.171.2.3 complex<double> eos::QCDFIntegrals::Results::j0bar_perp

6.171.2.4 complex<double> eos::QCDFIntegrals::Results::j1_parallel

6.171.2.5 complex<double> eos::QCDFIntegrals::Results::j1_perp

6.171.2.6 complex<double> eos::QCDFIntegrals::Results::j2_perp

6.171.2.7 complex<double> eos::QCDFIntegrals::Results::j3_parallel

6.171.2.8 complex<double> eos::QCDFIntegrals::Results::j4_parallel

6.171.2.9 complex<double> eos::QCDFIntegrals::Results::j4_perp

6.171.2.10 complex<double> eos::QCDFIntegrals::Results::j5_perp

6.171.2.11 complex<double> eos::QCDFIntegrals::Results::j6_perp

6.171.2.12 double eos::QCDFIntegrals::Results::j7_perp

6.171.2.13 complex<double> eos::QCDFIntegrals::Results::jtilde1_perp

6.171.2.14 complex<double> eos::QCDFIntegrals::Results::jtilde2_parallel

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/qcdf_integrals.hh

6.172 eos::RValue Struct Reference

```
#include <rvalue.hh>
```

Static Public Member Functions

- static double gelman_rubin (const std::vector< double > &chain_means, const std::vector< double > &chain_variances, const unsigned &chain_length)
- static double approximation (const std::vector< double > &chain_means, const std::vector< double > &chain_variances, const unsigned &chain_length)

6.172.1 Member Function Documentation

6.172.1.1 static double eos::RValue::approximation (const std::vector< double > & *chain_means*, const std::vector< double > & *chain_variances*, const unsigned & *chain_length*) [static]

Approximate the R-value (actually sqrt(R)) for a given quantitiy x.

Here we use the approximation $R ^2 / W$, just as in BAT v0.4.

Parameters:

chain_means the mean of x in each chain

chain_variances the sample variance of x in each chain

chain_length The number of iterations used to calculate the means and variances within each chain. Usually the length of the prerun until now. Note that if only chain_length is increased and everything else is kept constant, the R-value should increase

6.172.1.2 static double eos::RValue::gelman_rubin (const std::vector< double > & *chain_means*, const std::vector< double > & *chain_variances*, const unsigned & *chain_length*) [static]

Calculate the R-value (actually the sqrt(R)) for a given quantity x (param, log(posterior), ...) according to [GR1992], Eqs. (3),(4), p. 461. Using their notation. Included DoF estimation for the t-distribution.

Parameters:

chain_means the mean of x in each chain

chain_variances the sample variance of x in each chain

chain_length The number of iterations used to calculate the means and variances within each chain. Usually the length of the prerun until now. Note that if only chain_length is increased and everything else is kept constant, the R-value should increase

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

- /home/alexshires/EOS/eos-next/eos/utils/rvalue.hh

6.173 eos::Save< V_, R_ > Class Template Reference

```
#include <save.hh>
```

Public Member Functions

- Save (V_ &variable, const R_ &replacement)
- ~Save ()

template<typename V_, typename R_ = V_> class eos::Save< V_, R_ >

6.173.1 Constructor & Destructor Documentation

6.173.1.1 template<typename V_ , typename R_ = V_> eos::Save< V_, R_ >::Save (V_ & *variable*, const R_ & *replacement*) [inline]

6.173.1.2 template<typename V_ , typename R_ = V_> eos::Save< V_, R_ >::~Save () [inline]

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

- /home/alexshires/EOS/eos-next/eos/utils/save.hh

6.174 eos::hdf5::Scalar< T_ > Class Template Reference

```
#include <hdf5.hh>
```

Inherits eos::hdf5::Type.

Public Types

- typedef T_ Type

Public Member Functions

- Scalar (const std::string &name)
- ~Scalar ()
- virtual hid_t type_id () const
- virtual const std::string & name () const
- virtual hsize_t size () const
- virtual void copy_to_hdf5 (const void *src, void *dest) const
- virtual void copy_from_hdf5 (const void *src, void *dest) const

6.174.1 Detailed Description

template<typename T_> class eos::hdf5::Scalar< T_ >

Scalar represents a scalar built-in HDF5 data type.

6.174.2 Member Typedef Documentation

6.174.2.1 template<typename T_ > typedef T_ eos::hdf5::Scalar< T_ >::Type

6.174.3 Constructor & Destructor Documentation

6.174.3.1 template<typename T_ > eos::hdf5::Scalar< T_ >::Scalar (const std::string & *name*) [inline]

6.174.3.2 template<typename T_ > eos::hdf5::Scalar< T_ >::~Scalar () [inline]

6.174.4 Member Function Documentation

6.174.4.1 template<typename T_ > virtual void eos::hdf5::Scalar< T_ >::copy_from_hdf5 (const void * *src*, void * *dest*) const [inline, virtual]

Implements eos::hdf5::Type.

6.174.4.2 template<typename T_ > virtual void eos::hdf5::Scalar< T_ >::copy_to_hdf5 (const void * *src*, void * *dest*) const [inline, virtual]

Implements eos::hdf5::Type.

6.174.4.3 template<typename T_ > virtual const std::string& eos::hdf5::Scalar< T_ >::name () const [inline, virtual]

Implements eos::hdf5::Type.

6.174.4.4 template<typename T_ > virtual hsize_t eos::hdf5::Scalar< T_ >::size () const [inline, virtual]

Implements eos::hdf5::Type.

6.174.4.5 template<typename T_ > virtual hid_t eos::hdf5::Scalar< T_ >::type_id () const [inline, virtual]

Implements eos::hdf5::Type.

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.175 eos::SelectOneOfType< Want_ > Struct Template Reference

```
#include <one-of.hh>
```

Public Types

- typedef UnknownTypeForOneOf Type

template<typename Want_> struct eos::SelectOneOfType< Want_ >

6.175.1 Member Typedef Documentation

6.175.1.1 template<typename Want_ > typedef UnknownTypeForOneOf eos::SelectOneOfType< Want_ >::Type

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.176 eos::SelectOneOfType< Want_, Try_, Rest_...> Struct Template Reference

```
#include <one-of.hh>
```

Public Types

- typedef std::conditional< std::is_same< Want_, Try_ >::value, Try_, typename SelectOneOfType< Want_, Rest_...>::Type >::type Type

template<typename Want_, typename Try_, typename... Rest_> struct eos::SelectOneOfType< Want_, Try_, Rest_...>

6.176.1 Member Typedef Documentation

6.176.1.1 template<typename Want_ , typename Try_ , typename... Rest_> typedef std::conditional< std::is_same<Want_, Try_>::value, Try_, typename SelectOneOfType<Want_, Rest_ ...>::Type >::type eos::SelectOneOfType< Want_, Try_, Rest_...>::Type

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

- /home/alexshires/EOS/eos-next/eos/utils/one-of.hh

6.177 eos::ShortDistanceLowRecoil Struct Reference

```
#include <charm-loops.hh>
```

Static Public Member Functions

- static complex< double > c7eff (const double &s, const double &mu, const double &alpha_s, const double &m_b_PS, bool use_nlo, const WilsonCoefficients< BToS > &wc)
- static complex< double > c9eff (const double &s, const double &mu, const double &alpha_s, const double &m_b_PS, const double &m_c_MSbar, bool use_nlo, bool ccbar_resonance, const complex< double > &lambda_hat_u, const WilsonCoefficients< BToS > &wc)

6.177.1 Member Function Documentation

6.177.1.1 static complex<double> eos::ShortDistanceLowRecoil::c7eff (const double & *s*, const double & *mu*, const double & *alpha_s*, const double & *m_b_PS*, bool *use_nlo*, const WilsonCoefficients< BToS > & *wc*) [static]

Effective Wilson coefficient c7 in the region of low hadronic recoil.

Parameters:

s dilepton invariant mass

mu renormalization scale

alpha_s strong coupling evaluated at the scale mu

m_b_PS PS mass of the bottom quark

use_nlo true, if NLO contributions shall be used

wc the Wilson coefficients

For the calculation, cf. [GP2004], Eq. (56)

6.177.1.2 static complex<double> eos::ShortDistanceLowRecoil::c9eff (const double & *s*, const double & *mu*, const double & *alpha_s*, const double & *m_b_PS*, const double & *m_c_MSbar*, bool *use_nlo*, bool *ccbar_resonance*, const complex< double > & *lambda_hat_u*, const WilsonCoefficients< BToS > & *wc*) [static]

Effective Wilson coefficient c9 in the region of low hadronic recoil.

Parameters:

s dilepton invariant mass

mu renormalization scale

alpha_s strong coupling evaluated at the scale mu

m_b_PS PS mass of the bottom quark

m_c MSbar mass of the charm quark

use_nlo true, if NLO contributions shall be used

ccbar_resonance true, if phenomenological data from e^+e^- -> ccbar resonance -> hadrons shall be used

lambda_hat_u certain combination of CKM matrix elements: V_ub V_us^∗ / (V_tb V_ts^∗)

wc the Wilson coefficients

For the calculation, cf. [GP2004], Eq. (55), p. 10

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

- /home/alexshires/EOS/eos-next/eos/rare-b-decays/charm-loops.hh

6.178 eos::SMComponent< components::CKM > Class Template Reference

`#include <standard-model.hh>`

Inherits eos::ModelComponent< components::CKM >.

Inherited by eos::StandardModel, and eos::WilsonScanModel.

Public Member Functions

- SMComponent (const Parameters &, ParameterUser &)
- virtual complex< double > ckm_cd () const
- virtual complex< double > ckm_cs () const
- virtual complex< double > ckm_cb () const
- virtual complex< double > ckm_ud () const
- virtual complex< double > ckm_us () const
- virtual complex< double > ckm_ub () const
- virtual complex< double > ckm_td () const
- virtual complex< double > ckm_ts () const
- virtual complex< double > ckm_tb () const

template<> class eos::SMComponent< components::CKM >

6.178.1 Constructor & Destructor Documentation

6.178.1.1 eos::SMComponent< components::CKM >::SMComponent (const Parameters &, ParameterUser &)

6.178.2 Member Function Documentation

6.178.2.1 virtual complex<double> eos::SMComponent< components::CKM >::ckm_cb () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.2 virtual complex<double> eos::SMComponent< components::CKM >::ckm_cd () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.3 virtual complex<double> eos::SMComponent< components::CKM >::ckm_cs () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.4 virtual complex<double> eos::SMComponent< components::CKM >::ckm_tb () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.5 virtual complex<double> eos::SMComponent< components::CKM >::ckm_td () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.6 virtual complex<double> eos::SMComponent< components::CKM >::ckm_ts () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.7 virtual complex<double> eos::SMComponent< components::CKM >::ckm_ub () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.8 virtual complex<double> eos::SMComponent< components::CKM >::ckm_ud () const [virtual]

Implements eos::ModelComponent< components::CKM >.

6.178.2.9 virtual complex<double> eos::SMComponent< components::CKM >::ckm_us () const [virtual]

Implements eos::ModelComponent< components::CKM >.

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

- /home/alexshires/EOS/eos-next/eos/utils/standard-model.hh

6.179 eos::SMComponent< components::DeltaB1 > Class Template Reference

`#include <standard-model.hh>`

Inherits eos::ModelComponent< components::DeltaB1 >.

Inherited by eos::CKMScanModel, and eos::StandardModel.

Public Member Functions

- SMComponent (const Parameters &, ParameterUser &)
- virtual WilsonCoefficients< BToS > wilson_coefficients_b_to_s (const bool &cp_conjugate) const

template<> class eos::SMComponent< components::DeltaB1 >

6.179.1 Constructor & Destructor Documentation

6.179.1.1 eos::SMComponent< components::DeltaB1 >::SMComponent (const Parameters &, ParameterUser &)

6.179.2 Member Function Documentation

6.179.2.1 virtual WilsonCoefficients<BToS> eos::SMComponent< components::DeltaB1 >::wilson_coefficients_b_to_s (const bool & *cp_conjugate*) const `[virtual]`

Implements eos::ModelComponent< components::DeltaB1 >.

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

- /home/alexshires/EOS/eos-next/eos/utils/standard-model.hh

6.180 eos::SMComponent< components::QCD > Class Template Reference

`#include <standard-model.hh>`

Inherits eos::ModelComponent< components::QCD >.

Inherited by eos::CKMScanModel, eos::StandardModel, and eos::WilsonScanModel.

Public Member Functions

- SMComponent (const Parameters &, ParameterUser &)
- virtual double alpha_s (const double &) const
- virtual double m_t_msbar (const double &mu) const
- virtual double m_t_pole () const
- virtual double m_b_kin (const double &mu_kin) const
- virtual double m_b_msbar (const double &mu) const
- virtual double m_b_pole () const
- virtual double m_b_ps (const double &mu_f) const
- virtual double m_c_kin (const double &mu_kin) const
- virtual double m_c_msbar (const double &mu) const
- virtual double m_c_pole () const
- virtual double m_s_msbar (const double &mu) const
- virtual double m_ud_msbar (const double &mu) const

template<> class eos::SMComponent< components::QCD >

6.180.1 Constructor & Destructor Documentation

6.180.1.1 eos::SMComponent< components::QCD >::SMComponent (const Parameters &, ParameterUser &)

6.180.2 Member Function Documentation

6.180.2.1 virtual double eos::SMComponent< components::QCD >::alpha_s (const double &) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.2 virtual double eos::SMComponent< components::QCD >::m_b_kin (const double & *mu_kin*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.3 virtual double eos::SMComponent< components::QCD >::m_b_msbar (const double & *mu*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.4 virtual double eos::SMComponent< components::QCD >::m_b_pole () const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.5 virtual double eos::SMComponent< components::QCD >::m_b_ps (const double & *mu_f*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.6 virtual double eos::SMComponent< components::QCD >::m_c_kin (const double & *mu_kin*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.7 virtual double eos::SMComponent< components::QCD >::m_c_msbar (const double & *mu*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.8 virtual double eos::SMComponent< components::QCD >::m_c_pole () const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.9 virtual double eos::SMComponent< components::QCD >::m_s_msbar (const double & *mu*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.10 virtual double eos::SMComponent< components::QCD >::m_t_msbar (const double & *mu*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.11 virtual double eos::SMComponent< components::QCD >::m_t_pole () const [virtual]

Implements eos::ModelComponent< components::QCD >.

6.180.2.12 virtual double eos::SMComponent< components::QCD >::m_ud_msbar (const double & *mu*) const [virtual]

Implements eos::ModelComponent< components::QCD >.

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

- /home/alexshires/EOS/eos-next/eos/utils/standard-model.hh

6.181 eos::EquationSolver::Solution Struct Reference

```
#include <equation_solver.hh>
```

Public Attributes

- std::vector< double > parameters

  The parameter values giving the best solution to the problem.

- double value

  Value of the function to be minimized at the minimum found.

- bool valid

  True if the solution finding succeeded.

6.181.1 Member Data Documentation

6.181.1.1 std::vector<double> eos::EquationSolver::Solution::parameters

The parameter values giving the best solution to the problem.

6.181.1.2 bool eos::EquationSolver::Solution::valid

True if the solution finding succeeded.

6.181.1.3 double eos::EquationSolver::Solution::value

Value of the function to be minimized at the minimum found.

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

- /home/alexshires/EOS/eos-next/eos/utils/equation_solver.hh

6.182 eos::StandardModel Class Reference

`#include <standard-model.hh>`

Inherits eos::Model, eos::SMComponent< components::CKM >, eos::SMComponent< components::QCD >, and eos::SMComponent< components::DeltaB1 >.

Public Member Functions

- StandardModel (const Parameters &)
- virtual ~StandardModel ()

Static Public Member Functions

- static std::shared_ptr< Model > make (const Parameters &, const Options &)

6.182.1 Constructor & Destructor Documentation

6.182.1.1 eos::StandardModel::StandardModel (const Parameters &)

6.182.1.2 virtual eos::StandardModel::~StandardModel () `[virtual]`

6.182.2 Member Function Documentation

6.182.2.1 static std::shared_ptr<Model> eos::StandardModel::make (const Parameters &, const Options &) `[static]`

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

- /home/alexshires/EOS/eos-next/eos/utils/standard-model.hh

6.183 eos::MarkovChain::State Struct Reference

```
#include <markov_chain.hh>
```

Public Types

- typedef std::vector< State >::const_iterator Iterator

Public Member Functions

- State ()
- State (const State &other)

Public Attributes

- std::vector< double > point

  position in parameter space

- double log_likelihood

  log likelihood at the point

- double log_prior

  log prior at the point

- double log_posterior

  log posterior at the points

- MarkovChain::HyperParameter hyper_parameter

  Contains possibly multidimensional hyperparameter information.

6.183.1 Detailed Description

Summarize info at current position in parameter space

6.183.2 Member Typedef Documentation

6.183.2.1 typedef std::vector<State>::const_iterator eos::MarkovChain::State::Iterator

6.183.3 Constructor & Destructor Documentation

6.183.3.1 eos::MarkovChain::State::State () `[inline]`

6.183.3.2 eos::MarkovChain::State::State (const State & *other*) `[inline]`

6.183.4 Member Data Documentation

6.183.4.1 MarkovChain::HyperParameter eos::MarkovChain::State::hyper_parameter

Contains possibly multidimensional hyperparameter information.

6.183.4.2 double eos::MarkovChain::State::log_likelihood

log likelihood at the point

6.183.4.3 double eos::MarkovChain::State::log_posterior

log posterior at the points

6.183.4.4 double eos::MarkovChain::State::log_prior

log prior at the point

6.183.4.5 std::vector<double> eos::MarkovChain::State::point

position in parameter space

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh

6.184 eos::MarkovChain::Stats Struct Reference

```
#include <markov_chain.hh>
```

Public Attributes

- unsigned iterations_total
- unsigned iterations_accepted
- unsigned iterations_invalid
- unsigned iterations_rejected
- double mode_of_posterior

  Maximum value of the posterior.

- std::vector< double > parameters_at_mode

  Parameter values at the maximum of the posterior.

- std::vector< double > mean_of_parameters

  Sample mean of parameter values.

- double mean_of_posterior

  Sample mean of posterior.

- std::vector< double > variance_of_parameters

  Sample variance of parameter values.

- double variance_of_posterior

  Sample variance of [log] posterior values.

6.184.1 Detailed Description

Holds statistical information of a run of a MarkovChain

6.184.2 Member Data Documentation

6.184.2.1 unsigned eos::MarkovChain::Stats::iterations_accepted

The number of accepted proposals.

Note:

accepted and rejected only add up to the number of iterations in the current run. The total number of samples used to calculated the variance maybe different from the sum of former two.

6.184.2.2 unsigned eos::MarkovChain::Stats::iterations_invalid

6.184.2.3 unsigned eos::MarkovChain::Stats::iterations_rejected

The number of iterations in which the proposed move of a parameter has been rejected. Reset each time run() is called

6.184.2.4 unsigned eos::MarkovChain::Stats::iterations_total

The total number of iterations per parameter that were collected before the current run, i.e. since the last reset of statistics.

6.184.2.5 std::vector<double> eos::MarkovChain::Stats::mean_of_parameters

Sample mean of parameter values.

6.184.2.6 double eos::MarkovChain::Stats::mean_of_posterior

Sample mean of posterior.

6.184.2.7 double eos::MarkovChain::Stats::mode_of_posterior

Maximum value of the posterior.

6.184.2.8 std::vector<double> eos::MarkovChain::Stats::parameters_at_mode

Parameter values at the maximum of the posterior.

6.184.2.9 std::vector<double> eos::MarkovChain::Stats::variance_of_parameters

Sample variance of parameter values.

6.184.2.10 double eos::MarkovChain::Stats::variance_of_posterior

Sample variance of [log] posterior values.

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

- /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh

6.185 eos::PopulationMonteCarloSampler::Status Struct Reference

```
#include <population_monte_carlo_sampler.hh>
```

Public Member Functions

- Status ()

Public Attributes

- unsigned chunk_size
- bool converged

  Whether sampling has converged.

- unsigned iterations_at_convergence

  The number of iterations after which convergence was declared.

- double evidence

  Evidence.

- double eff_sample_size

  Effective sample size.

- double perplexity

  Perplexity.

6.185.1 Detailed Description

Holds convergence information of the population sampling.

6.185.2 Constructor & Destructor Documentation

6.185.2.1 eos::PopulationMonteCarloSampler::Status::Status ()

6.185.3 Member Data Documentation

6.185.3.1 unsigned eos::PopulationMonteCarloSampler::Status::chunk_size

The actual number of samples drawn from mixture proposal density before the latter is updated.

6.185.3.2 bool eos::PopulationMonteCarloSampler::Status::converged

Whether sampling has converged.

6.185.3.3 double eos::PopulationMonteCarloSampler::Status::eff_sample_size

Effective sample size.

6.185.3.4 double eos::PopulationMonteCarloSampler::Status::evidence

Evidence.

6.185.3.5 unsigned eos::PopulationMonteCarloSampler::Status::iterations_at_convergence

The number of iterations after which convergence was declared.

6.185.3.6 double eos::PopulationMonteCarloSampler::Status::perplexity

Perplexity.

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

- /home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_sampler.hh

6.186 eos::TestObservable Struct Reference

```
#include <analysis_TEST.hh>
```

Inherits eos::Observable.

Inherited by eos::AbsoluteTestObservable.

Public Member Functions

- TestObservable (const Parameters &p, const Kinematics &k, const std::string &mass_name)
- virtual ~TestObservable ()
- virtual double evaluate () const
- virtual ObservablePtr clone () const
- virtual ObservablePtr clone (const Parameters &parameters) const
- virtual Parameters parameters ()
- virtual Kinematics kinematics ()
- virtual Options options ()
- const std::string & name () const
- void set_option (const std::string &key, const std::string &value="")

Public Attributes

- Parameters p
- Kinematics k
- Options o
- std::string n
- std::string mass_name
- UsedParameter mass

6.186.1 Constructor & Destructor Documentation

6.186.1.1 eos::TestObservable::TestObservable (const Parameters & *p*, const Kinematics & *k*, const std::string & *mass_name*) `[inline]`

6.186.1.2 virtual eos::TestObservable::~TestObservable () `[inline, virtual]`

6.186.2 Member Function Documentation

6.186.2.1 virtual ObservablePtr eos::TestObservable::clone (const Parameters & *parameters*) const `[inline, virtual]`

Implements eos::Observable.

Reimplemented in eos::AbsoluteTestObservable.

6.186.2.2 virtual ObservablePtr eos::TestObservable::clone () const `[inline, virtual]`

Implements eos::Observable.

Reimplemented in eos::AbsoluteTestObservable.

6.186.2.3 virtual double eos::TestObservable::evaluate () const [inline, virtual]

Implements eos::Observable.

Reimplemented in eos::AbsoluteTestObservable.

6.186.2.4 virtual Kinematics eos::TestObservable::kinematics () [inline, virtual]

Implements eos::Observable.

6.186.2.5 const std::string& eos::TestObservable::name () const [inline, virtual]

Implements eos::Observable.

6.186.2.6 virtual Options eos::TestObservable::options () [inline, virtual]

Implements eos::Observable.

6.186.2.7 virtual Parameters eos::TestObservable::parameters () [inline, virtual]

Implements eos::Observable.

6.186.2.8 void eos::TestObservable::set_option (const std::string & *key*, const std::string & *value* = "") [inline]

6.186.3 Member Data Documentation

6.186.3.1 Kinematics eos::TestObservable::k

6.186.3.2 UsedParameter eos::TestObservable::mass

6.186.3.3 std::string eos::TestObservable::mass_name

6.186.3.4 std::string eos::TestObservable::n

6.186.3.5 Options eos::TestObservable::o

6.186.3.6 Parameters eos::TestObservable::p

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

- /home/alexshires/EOS/eos-next/eos/utils/analysis_TEST.hh

6.187 eos::Thread Class Reference

`#include <thread.hh>`

Inherits PrivateImplementationPattern< Thread >.

Public Types

- typedef std::function< void()> Function

  Our function type.

Public Member Functions

- bool completed () const

  Return wether we have yet completed executing our function.

Constructor and destructor

- Thread (const Function &function)
- ~Thread ()

  Destructor.

6.187.1 Detailed Description

Thread uses TR1's function wrapper to execute a given function in a separate thread.

Thread uses POSIX threads internally and allows minimal synchronisation. Thread guarantees that its function is finalised after destruction.

6.187.2 Member Typedef Documentation

6.187.2.1 typedef std::function<void ()> eos::Thread::Function

Our function type.

6.187.3 Constructor & Destructor Documentation

6.187.3.1 eos::Thread::Thread (const Function & *function*)

Constructor.

Parameters:

function A Function object that wraps the thread's main function.

6.187.3.2 eos::Thread::∼Thread ()

Destructor.

6.187.4 Member Function Documentation

6.187.4.1 bool eos::Thread::completed () const

Return wether we have yet completed executing our function.

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

- /home/alexshires/EOS/eos-next/eos/utils/thread.hh

6.188 eos::ThreadPool Class Reference

`#include <thread_pool.hh>`

Inherits PrivateImplementationPattern< ThreadPool >.

Public Member Functions

- ThreadPool ()
- ~ThreadPool ()
- Ticket enqueue (const std::function< void(void)> &work)
- void wait_for_free_capacity ()
- unsigned number_of_threads () const

Static Public Member Functions

- static ThreadPool * instance ()

6.188.1 Constructor & Destructor Documentation

6.188.1.1 eos::ThreadPool::ThreadPool ()

6.188.1.2 eos::ThreadPool::~ThreadPool ()

6.188.2 Member Function Documentation

6.188.2.1 Ticket eos::ThreadPool::enqueue (const std::function< void(void)> & *work*)

6.188.2.2 static ThreadPool* eos::ThreadPool::instance () `[static]`

6.188.2.3 unsigned eos::ThreadPool::number_of_threads () const

6.188.2.4 void eos::ThreadPool::wait_for_free_capacity ()

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

- /home/alexshires/EOS/eos-next/eos/utils/thread_pool.hh

6.189 eos::Ticket Class Reference

`#include <ticket.hh>`

Inherits PrivateImplementationPattern< Ticket >.

Public Member Functions

- void mark ()
  Mark ticket as completed.
- void wait () const
  Wait for ticket completion.

Basic Operations

- Ticket ()
  Constructor.
- ~Ticket ()
  Destructor.

Friends

Friends of Ticket

- class TicketList

6.189.1 Detailed Description

Ticket is used by all asynchronous function calls to relay/query information on a function's completion status.

6.189.2 Constructor & Destructor Documentation

6.189.2.1 eos::Ticket::Ticket ()

Constructor.

6.189.2.2 eos::Ticket::~Ticket ()

Destructor.

6.189.3 Member Function Documentation

6.189.3.1 void eos::Ticket::mark ()

Mark ticket as completed.

6.189.3.2 void eos::Ticket::wait () const

Wait for ticket completion.

6.189.4 Friends And Related Function Documentation

6.189.4.1 friend class TicketList `[friend]`

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

- /home/alexshires/EOS/eos-next/eos/utils/ticket.hh

6.190 eos::TicketList Class Reference

`#include <ticket.hh>`

Inherits PrivateImplementationPattern< TicketList >.

Public Member Functions

- void push_back (const Ticket &ticket)
  Push a ticket to the back of the list.
- void wait () const
  Wait for ticket completion.

Basic Operations

- TicketList ()
  Constructor.
- ~TicketList ()
  Destructor.

6.190.1 Detailed Description

TicketList is used when multiple tickets are needed.

6.190.2 Constructor & Destructor Documentation

6.190.2.1 eos::TicketList::TicketList ()

Constructor.

6.190.2.2 eos::TicketList::~TicketList ()

Destructor.

6.190.3 Member Function Documentation

6.190.3.1 void eos::TicketList::push_back (const Ticket & *ticket*)

Push a ticket to the back of the list.

6.190.3.2 void eos::TicketList::wait () const

Wait for ticket completion.

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

- /home/alexshires/EOS/eos-next/eos/utils/ticket.hh

6.191 eos::TopLoops Struct Reference

```
#include <top-loops.hh>
```

Static Public Member Functions

- static double A0 (const double &x_t)
- static double B0 (const double &x_t)
- static double C0 (const double &x_t)
- static double D0 (const double &x_t)
- static double E0 (const double &x_t)
- static double F0 (const double &x_t)
- static double A1 (const double &x_t, const double &log_t)
- static double B1 (const double &x_t, const double &log_t)
- static double C1 (const double &x_t, const double &log_t)
- static double D1 (const double &x_t, const double &log_t)
- static double E1 (const double &x_t, const double &log_t)
- static double F1 (const double &x_t, const double &log_t)
- static double G1 (const double &x_t, const double &log_t)

6.191.1 Detailed Description

Helper class to bundle top-loop functions in the SM which are relevant to $b \to s$ transitions.

6.191.2 Member Function Documentation

6.191.2.1 static double eos::TopLoops::A0 (const double & *x_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.2 static double eos::TopLoops::A1 (const double & *x_t*, const double & *log_t*) [static]

Two-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$
log_t $= \ln(\mu_t/m_t(\mu_t))$

6.191.2.3 static double eos::TopLoops::B0 (const double & *x_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.4 static double eos::TopLoops::B1 (const double & *x_t*, const double & *log_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.5 static double eos::TopLoops::C0 (const double & *x_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.6 static double eos::TopLoops::C1 (const double & *x_t*, const double & *log_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.7 static double eos::TopLoops::D0 (const double & *x_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.8 static double eos::TopLoops::D1 (const double & *x_t*, const double & *log_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.9 static double eos::TopLoops::E0 (const double & *x_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.10 static double eos::TopLoops::E1 (const double & *x_t*, const double & *log_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.11 static double eos::TopLoops::F0 (const double & *x_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.12 static double eos::TopLoops::F1 (const double & *x_t*, const double & *log_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

6.191.2.13 static double eos::TopLoops::G1 (const double & *x_t*, const double & *log_t*) [static]

One-loop functions in the MSbar scheme at the renormalization scale μ_t.

Parameters:

x_t $= m_t(\mu_t)^2/m_W^2$

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

- /home/alexshires/EOS/eos-next/eos/utils/top-loops.hh

6.192 eos::TryLock Class Reference

```
#include <lock.hh>
```

Public Member Functions

- TryLock (Mutex &e)
  (Explicit) constructor.
- ~TryLock ()
  Destructor.
- bool operator() () const
  Return true if the lock worked.

6.192.1 Constructor & Destructor Documentation

6.192.1.1 eos::TryLock::TryLock (Mutex & *e*) [explicit]

(Explicit) constructor.

6.192.1.2 eos::TryLock::~TryLock ()

Destructor.

6.192.2 Member Function Documentation

6.192.2.1 bool eos::TryLock::operator() () const [inline]

Return true if the lock worked.

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

- /home/alexshires/EOS/eos-next/eos/utils/lock.hh

6.193 eos::deriv::TwoSided Class Reference

```
#include <derivative.hh>
```

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

- /home/alexshires/EOS/eos-next/eos/utils/derivative.hh

6.194 eos::hdf5::Type Struct Reference

```
#include <hdf5.hh>
```

Inherited by eos::hdf5::Array< rank_, T_ >, eos::hdf5::Composite< T_ >, and eos::hdf5::Scalar< T_ >.

Public Member Functions

- virtual ~Type ()=0
- virtual hid_t type_id () const =0
- virtual const std::string & name () const =0
- virtual hsize_t size () const =0
- virtual void copy_to_hdf5 (const void *src, void *dest) const =0
- virtual void copy_from_hdf5 (const void *src, void *dest) const =0

6.194.1 Constructor & Destructor Documentation

6.194.1.1 virtual eos::hdf5::Type::~Type () [pure virtual]

6.194.2 Member Function Documentation

6.194.2.1 virtual void eos::hdf5::Type::copy_from_hdf5 (const void * *src*, void * *dest*) const [pure virtual]

Implemented in eos::hdf5::Scalar< T_ >, eos::hdf5::Array< rank_, T_ >, and eos::hdf5::Composite< T_ >.

6.194.2.2 virtual void eos::hdf5::Type::copy_to_hdf5 (const void * *src*, void * *dest*) const [pure virtual]

Implemented in eos::hdf5::Scalar< T_ >, eos::hdf5::Array< rank_, T_ >, and eos::hdf5::Composite< T_ >.

6.194.2.3 virtual const std::string& eos::hdf5::Type::name () const [pure virtual]

Implemented in eos::hdf5::Scalar< T_ >, eos::hdf5::Array< rank_, T_ >, and eos::hdf5::Composite< T_ >.

6.194.2.4 virtual hsize_t eos::hdf5::Type::size () const [pure virtual]

Implemented in eos::hdf5::Scalar< T_ >, eos::hdf5::Array< rank_, T_ >, and eos::hdf5::Composite< T_ >.

6.194.2.5 virtual hid_t eos::hdf5::Type::type_id () const [pure virtual]

Implemented in eos::hdf5::Scalar< T_ >, eos::hdf5::Array< rank_, T_ >, and eos::hdf5::Composite< T_ >.

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

- /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh

6.195 eos::UnknownConstraintError Struct Reference

```
#include <constraint.hh>
```

Inherits eos::Exception.

Public Member Functions

Basic Functions

- UnknownConstraintError (const std::string &name)

6.195.1 Detailed Description

UnknownConstraintError is thrown when Constraint::make encounters an unknown constraint name.

6.195.2 Constructor & Destructor Documentation

6.195.2.1 eos::UnknownConstraintError::UnknownConstraintError (const std::string & *name*)

Constructor.

Parameters:

name The offending constraint name.

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

- /home/alexshires/EOS/eos-next/eos/constraint.hh

6.196 eos::UnknownKinematicVariableError Struct Reference

```
#include <kinematic.hh>
```

Inherits eos::Exception.

Public Member Functions

- UnknownKinematicVariableError (const std::string &variable) throw ()

6.196.1 Detailed Description

UnknownKinematicVariableError is thrown when no parameter of a given name could be found.

6.196.2 Constructor & Destructor Documentation

6.196.2.1 eos::UnknownKinematicVariableError::UnknownKinematicVariableError (const std::string & *variable*) throw ()

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

- /home/alexshires/EOS/eos-next/eos/utils/kinematic.hh

6.197 eos::UnknownOptionError Struct Reference

```
#include <options.hh>
```

Inherits eos::Exception.

Public Member Functions

- UnknownOptionError (const std::string &key) throw ()

6.197.1 Detailed Description

UnknownOptionError is thrown when no option of a given name could be found.

6.197.2 Constructor & Destructor Documentation

6.197.2.1 eos::UnknownOptionError::UnknownOptionError (const std::string & *key*) throw ()

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

- /home/alexshires/EOS/eos-next/eos/utils/options.hh

6.198 eos::UnknownParameterError Struct Reference

```
#include <parameters.hh>
```

Inherits eos::Exception.

Public Member Functions

- UnknownParameterError (const std::string &variable) throw ()

6.198.1 Detailed Description

UnknownParameterError is thrown when no parameter of a given name could be found.

6.198.2 Constructor & Destructor Documentation

6.198.2.1 eos::UnknownParameterError::UnknownParameterError (const std::string & *variable*) throw ()

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

- /home/alexshires/EOS/eos-next/eos/utils/parameters.hh

6.199 eos::proposal_functions::UnknownProposalError Struct Reference

```
#include <proposal_functions.hh>
```

Inherits eos::Exception.

Public Member Functions

Basic Functions

- UnknownProposalError (const std::string &name)

6.199.1 Detailed Description

UnknownProposalError is thrown when Proposal::make encounters an unknown proposal density name.

6.199.2 Constructor & Destructor Documentation

6.199.2.1 eos::proposal_functions::UnknownProposalError::UnknownProposalError (const std::string & *name*)

Constructor.

Parameters:

name The offending constraint name.

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

- /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh

6.200 eos::UsedParameter Class Reference

```
#include <parameters.hh>
```

Inherits eos::Parameter.

Public Member Functions

- UsedParameter (const Parameter &parameter, ParameterUser &user)

6.200.1 Detailed Description

Wrapper class to automate usage tracking of Parameter objects.

6.200.2 Constructor & Destructor Documentation

6.200.2.1 eos::UsedParameter::UsedParameter (const Parameter & *parameter*, ParameterUser & *user*)

Constructor.

Constructs a Parameter object and registers its usage with a ParameterUser.

Parameters:

parameter The parameter which is used.

user The user of above parameter.

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

- /home/alexshires/EOS/eos-next/eos/utils/parameters.hh

6.201 eos::VerifiedRange< T_ > Class Template Reference

```
#include <verify.hh>
```

Public Member Functions

Basic Functions

- VerifiedRange (const T_ &min, const T_ &max, const T_ &value)

Operations

*Conversion operator its native type T_*

- operator T_ () const
- const T_ & operator= (const T_ &rhs)
- T_ min () const

  Retrieve the minimal value that is allowed for this object.

- T_ max () const

  Retrieve the maximal value that is allowed for this object.

6.201.1 Detailed Description

template<typename T_> class eos::VerifiedRange< T_ >

VerifiedRange is a wrapper around a variable that allows assignment only within a given range.

```
VerifiedRange<double> value_between_zero_and_one(0.0, 1.0, 0.5);
test_range = +10; // not allowed, throws a VerifiedRangeOverflow
test_range = -10; // not allowed, throws a VerifiedRangeUnderflow
```

6.201.2 Constructor & Destructor Documentation

6.201.2.1 template<typename T_> eos::VerifiedRange< T_ >::VerifiedRange (const T_ & *min*, const T_ & *max*, const T_ & *value*) [inline]

Constructor.

Parameters:

min Minimal value that is allowed for any value of this object.

max Maximal value that is allowed for any value of this object.

value Initial value of this object.

6.201.3 Member Function Documentation

6.201.3.1 template<typename T_> T_ eos::VerifiedRange< T_ >::max () const [inline]

Retrieve the maximal value that is allowed for this object.

6.201.3.2 template<typename T_> T_ eos::VerifiedRange< T_ >::min () const [inline]

Retrieve the minimal value that is allowed for this object.

6.201.3.3 template<typename T_> eos::VerifiedRange< T_ >::operator T_ () const [inline]

Assignment operator.

Parameters:

rhs Right-hand-side of an assignment to this object.

6.201.3.4 template<typename T_> const T_& eos::VerifiedRange< T_ >::operator= (const T_ & *rhs*) [inline]

Assignment operator.

Parameters:

rhs Right-hand-side of an assignment to this object.

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

- /home/alexshires/EOS/eos-next/eos/utils/verify.hh

6.202 eos::VerifiedRangeError Class Reference

```
#include <verify.hh>
```

Inherits eos::Exception.

Inherited by eos::VerifiedRangeOverflow, and eos::VerifiedRangeUnderflow.

Protected Member Functions

- VerifiedRangeError (const std::string &message)

6.202.1 Detailed Description

VerifiedRangeError is thrown whenever a VerifiedRange is assigned a value that exceeds its allowed range.

6.202.2 Constructor & Destructor Documentation

6.202.2.1 eos::VerifiedRangeError::VerifiedRangeError (const std::string & *message*) [protected]

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

- /home/alexshires/EOS/eos-next/eos/utils/verify.hh

6.203 eos::VerifiedRangeOverflow Struct Reference

```
#include <verify.hh>
```

Inherits eos::VerifiedRangeError.

Public Member Functions

- VerifiedRangeOverflow (const std::string &value, const std::string &maximum)

6.203.1 Detailed Description

VerifiedRangeOverflow is thrown whenever a VerifiedRange is assigned a value that is larger than its allowed maximum.

6.203.2 Constructor & Destructor Documentation

6.203.2.1 eos::VerifiedRangeOverflow::VerifiedRangeOverflow (const std::string & *value*, const std::string & *maximum*)

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

- /home/alexshires/EOS/eos-next/eos/utils/verify.hh

6.204 eos::VerifiedRangeUnderflow Struct Reference

```
#include <verify.hh>
```

Inherits eos::VerifiedRangeError.

Public Member Functions

- VerifiedRangeUnderflow (const std::string &value, const std::string &minimum)

6.204.1 Detailed Description

VerifiedRangeUnderflow is thrown whenever a VerifiedRange is assigned a value that is smaller than its allowed minimum.

6.204.2 Constructor & Destructor Documentation

6.204.2.1 eos::VerifiedRangeUnderflow::VerifiedRangeUnderflow (const std::string & *value*, const std::string & *minimum*)

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

- /home/alexshires/EOS/eos-next/eos/utils/verify.hh

6.205 eos::Welford Struct Reference

```
#include <welford.hh>
```

Public Member Functions

- Welford ()
- void add (const double &value)
- double mean () const
- unsigned number_of_elements () const
- double std_deviation () const
- double variance () const

6.205.1 Detailed Description

Calculate running mean and variance according to Welford's method, cf. Knuth, D. TAOCP, vol. 2, 3rd edition, p. 232, or http://www.johndcook.com/standard%5Fdeviation.html

6.205.2 Constructor & Destructor Documentation

6.205.2.1 eos::Welford::Welford ()

6.205.3 Member Function Documentation

6.205.3.1 void eos::Welford::add (const double & *value*)

6.205.3.2 double eos::Welford::mean () const

6.205.3.3 unsigned eos::Welford::number_of_elements () const

6.205.3.4 double eos::Welford::std_deviation () const

6.205.3.5 double eos::Welford::variance () const

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

- /home/alexshires/EOS/eos-next/eos/utils/welford.hh

6.206 eos::WilsonCoefficients< BToS > Struct Template Reference

```
#include <wilson_coefficients.hh>
```

Public Member Functions

- WilsonCoefficients ()
- complex< double > c1 () const
- complex< double > c2 () const
- complex< double > c3 () const
- complex< double > c4 () const
- complex< double > c5 () const
- complex< double > c6 () const
- complex< double > cq3 () const
- complex< double > cq4 () const
- complex< double > cq5 () const
- complex< double > cq6 () const
- complex< double > c2b () const
- complex< double > c7 () const
- complex< double > c8 () const
- complex< double > c9 () const
- complex< double > c10 () const
- complex< double > c7prime () const
- complex< double > c8prime () const
- complex< double > c9prime () const
- complex< double > c10prime () const
- complex< double > cS () const
- complex< double > cSprime () const
- complex< double > cP () const
- complex< double > cPprime () const
- complex< double > cT () const
- complex< double > cT5 () const

Public Attributes

- std::array< complex< double >, 15 > _sm_like_coefficients
- std::array< complex< double >, 15 > _primed_coefficients
- std::array< complex< double >, 6 > _scalar_tensor_coefficients
- double _alpha_s

template<> struct eos::WilsonCoefficients< BToS >

6.206.1 Constructor & Destructor Documentation

6.206.1.1 eos::WilsonCoefficients< BToS >::WilsonCoefficients ()

Default ctor

6.206.2 Member Function Documentation

6.206.2.1 complex<double> eos::WilsonCoefficients< BToS >::c1 () const [inline]

6.206.2.2 complex<double> eos::WilsonCoefficients< BToS >::c10 () const [inline]

6.206.2.3 complex<double> eos::WilsonCoefficients< BToS >::c10prime () const [inline]

6.206.2.4 complex<double> eos::WilsonCoefficients< BToS >::c2 () const [inline]

6.206.2.5 complex<double> eos::WilsonCoefficients< BToS >::c2b () const [inline]

6.206.2.6 complex<double> eos::WilsonCoefficients< BToS >::c3 () const [inline]

6.206.2.7 complex<double> eos::WilsonCoefficients< BToS >::c4 () const [inline]

6.206.2.8 complex<double> eos::WilsonCoefficients< BToS >::c5 () const [inline]

6.206.2.9 complex<double> eos::WilsonCoefficients< BToS >::c6 () const [inline]

6.206.2.10 complex<double> eos::WilsonCoefficients< BToS >::c7 () const [inline]

6.206.2.11 complex<double> eos::WilsonCoefficients< BToS >::c7prime () const [inline]

6.206.2.12 complex<double> eos::WilsonCoefficients< BToS >::c8 () const [inline]

6.206.2.13 complex<double> eos::WilsonCoefficients< BToS >::c8prime () const [inline]

6.206.2.14 complex<double> eos::WilsonCoefficients< BToS >::c9 () const [inline]

6.206.2.15 complex<double> eos::WilsonCoefficients< BToS >::c9prime () const [inline]

6.206.2.16 complex<double> eos::WilsonCoefficients< BToS >::cP () const [inline]

6.206.2.17 complex<double> eos::WilsonCoefficients< BToS >::cPprime () const [inline]

6.206.2.18 complex<double> eos::WilsonCoefficients< BToS >::cq3 () const [inline]

6.206.2.19 complex<double> eos::WilsonCoefficients< BToS >::cq4 () const [inline]

6.206.2.20 complex<double> eos::WilsonCoefficients< BToS >::cq5 () const [inline]

6.206.2.21 complex<double> eos::WilsonCoefficients< BToS >::cq6 () const [inline]

6.206.2.22 complex<double> eos::WilsonCoefficients< BToS >::cS () const [inline]

6.206.2.23 complex<double> eos::WilsonCoefficients< BToS >::cSprime () const [inline]

6.206.2.24 complex<double> eos::WilsonCoefficients< BToS >::cT () const [inline]

6.206.2.25 complex<double> eos::WilsonCoefficients< BToS >::cT5 () const [inline]

6.206.3 Member Data Documentation

6.206.3.1 double eos::WilsonCoefficients< BToS >::_alpha_s

6.206.3.2 std::array<complex<double>, 15> eos::WilsonCoefficients< BToS >::_primed_coefficients

6.206.3.3 std::array<complex<double>, 6> eos::WilsonCoefficients< BToS >::_scalar_tensor_coefficients

- /home/alexshires/EOS/eos-next/eos/utils/wilson_coefficients.hh

6.207 eos::WilsonPolynomialCloner Class Reference

```
#include <wilson-polynomial.hh>
```

Public Member Functions

- WilsonPolynomialCloner (const Parameters &parameters)
- WilsonPolynomial visit (const Constant &c)
- WilsonPolynomial visit (const Sum &s)
- WilsonPolynomial visit (const Product &p)
- WilsonPolynomial visit (const Sine &s)
- WilsonPolynomial visit (const Cosine &s)
- WilsonPolynomial visit (const Parameter &p)

6.207.1 Constructor & Destructor Documentation

6.207.1.1 eos::WilsonPolynomialCloner::WilsonPolynomialCloner (const Parameters & *parameters*)

6.207.2 Member Function Documentation

6.207.2.1 WilsonPolynomial eos::WilsonPolynomialCloner::visit (const Parameter & *p*)

6.207.2.2 WilsonPolynomial eos::WilsonPolynomialCloner::visit (const Cosine & *s*)

6.207.2.3 WilsonPolynomial eos::WilsonPolynomialCloner::visit (const Sine & *s*)

6.207.2.4 WilsonPolynomial eos::WilsonPolynomialCloner::visit (const Product & *p*)

6.207.2.5 WilsonPolynomial eos::WilsonPolynomialCloner::visit (const Sum & *s*)

6.207.2.6 WilsonPolynomial eos::WilsonPolynomialCloner::visit (const Constant & *c*)

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

- /home/alexshires/EOS/eos-next/eos/utils/wilson-polynomial.hh

6.208 eos::WilsonPolynomialEvaluator Class Reference

```
#include <wilson-polynomial.hh>
```

Public Member Functions

- double visit (const Constant &c)
- double visit (const Parameter &p)
- double visit (const Sum &s)
- double visit (const Product &p)
- double visit (const Sine &s)
- double visit (const Cosine &c)

6.208.1 Member Function Documentation

6.208.1.1 double eos::WilsonPolynomialEvaluator::visit (const Cosine & *c*)

6.208.1.2 double eos::WilsonPolynomialEvaluator::visit (const Sine & *s*)

6.208.1.3 double eos::WilsonPolynomialEvaluator::visit (const Product & *p*)

6.208.1.4 double eos::WilsonPolynomialEvaluator::visit (const Sum & *s*)

6.208.1.5 double eos::WilsonPolynomialEvaluator::visit (const Parameter & *p*)

6.208.1.6 double eos::WilsonPolynomialEvaluator::visit (const Constant & *c*)

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

- /home/alexshires/EOS/eos-next/eos/utils/wilson-polynomial.hh

6.209 eos::WilsonPolynomialPrinter Class Reference

```
#include <wilson-polynomial.hh>
```

Public Member Functions

- WilsonPolynomialPrinter (const bool &pretty=true)
- std::string visit (const Constant &c)
- std::string visit (const Sum &s)
- std::string visit (const Product &p)
- std::string visit (const Sine &s)
- std::string visit (const Cosine &s)
- std::string visit (const Parameter &p)

6.209.1 Constructor & Destructor Documentation

6.209.1.1 eos::WilsonPolynomialPrinter::WilsonPolynomialPrinter (const bool & *pretty* = `true`)

6.209.2 Member Function Documentation

6.209.2.1 std::string eos::WilsonPolynomialPrinter::visit (const Parameter & *p*)

6.209.2.2 std::string eos::WilsonPolynomialPrinter::visit (const Cosine & *s*)

6.209.2.3 std::string eos::WilsonPolynomialPrinter::visit (const Sine & *s*)

6.209.2.4 std::string eos::WilsonPolynomialPrinter::visit (const Product & *p*)

6.209.2.5 std::string eos::WilsonPolynomialPrinter::visit (const Sum & *s*)

6.209.2.6 std::string eos::WilsonPolynomialPrinter::visit (const Constant & *c*)

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

- /home/alexshires/EOS/eos-next/eos/utils/wilson-polynomial.hh

6.210 eos::WilsonScanComponent Class Reference

`#include <wilson_scan_model.hh>`

Inherits eos::ModelComponent< components::DeltaB1 >.

Inherited by eos::WilsonScanModel.

Public Member Functions

- WilsonScanComponent (const Parameters &, const Options &, ParameterUser &)
- virtual WilsonCoefficients< BToS > wilson_coefficients_b_to_s (const bool &cp_conjugate) const

6.210.1 Constructor & Destructor Documentation

6.210.1.1 eos::WilsonScanComponent::WilsonScanComponent (const Parameters &, const Options &, ParameterUser &)

6.210.2 Member Function Documentation

6.210.2.1 virtual WilsonCoefficients<BToS> eos::WilsonScanComponent::wilson_coefficients_b_to_s (const bool & *cp_conjugate*) const `[virtual]`

Implements eos::ModelComponent< components::DeltaB1 >.

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

- /home/alexshires/EOS/eos-next/eos/utils/wilson_scan_model.hh

6.211 eos::WilsonScanModel Class Reference

`#include <wilson_scan_model.hh>`

Inherits eos::Model, eos::SMComponent< components::CKM >, eos::SMComponent< components::QCD >, and eos::WilsonScanComponent.

Public Member Functions

- WilsonScanModel (const Parameters &, const Options &)
- virtual ~WilsonScanModel ()

Static Public Member Functions

- static std::shared_ptr< Model > make (const Parameters &, const Options &)

6.211.1 Constructor & Destructor Documentation

6.211.1.1 eos::WilsonScanModel::WilsonScanModel (const Parameters &, const Options &)

6.211.1.2 virtual eos::WilsonScanModel::~WilsonScanModel () [virtual]

6.211.2 Member Function Documentation

6.211.2.1 static std::shared_ptr<Model> eos::WilsonScanModel::make (const Parameters &, const Options &) [static]

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

- /home/alexshires/EOS/eos-next/eos/utils/wilson_scan_model.hh

6.212 eos::WrappedForwardIterator< Tag_, Value_ > Class Template Reference

```
#include <wrapped_forward_iterator.hh>
```

Classes

- struct Base
- struct BaseImpl

Public Types

Standard library typedefs

- typedef std::remove_reference< Value_ >::type & value_type
- typedef std::remove_reference< Value_ >::type & reference
- typedef std::remove_reference< Value_ >::type * pointer
- typedef std::ptrdiff_t difference_type
- typedef std::forward_iterator_tag iterator_category

Public Member Functions

Basic operations

- WrappedForwardIterator ()
- ~WrappedForwardIterator ()
- WrappedForwardIterator (const WrappedForwardIterator &)
- template<typename T_ >
  WrappedForwardIterator (const T_ &)
- WrappedForwardIterator & operator= (const WrappedForwardIterator &)

Increment

- WrappedForwardIterator & operator++ ()
- WrappedForwardIterator operator++ (int)

Dereference

- pointer operator-> () const
- reference operator* () const

Equality

- bool operator== (const WrappedForwardIterator &) const
- bool operator!= (const WrappedForwardIterator &) const

Underlying iterator

- template<typename T_ >
  T_ & underlying_iterator ()
- template<typename T_ >
  const T_ & underlying_iterator () const

6.212.1 Detailed Description

template<typename Tag_, typename Value_> class eos::WrappedForwardIterator< Tag_, Value_ >

A WrappedForwardIterator is a generic wrapper around a forward iterator, hiding the underlying base iterator.

6.212.2 Member Typedef Documentation

6.212.2.1 template<typename Tag_, typename Value_> typedef std::ptrdiff_t eos::WrappedForwardIterator< Tag_, Value_ >::difference_type

6.212.2.2 template<typename Tag_, typename Value_> typedef std::forward_iterator_tag eos::WrappedForwardIterator< Tag_, Value_ >::iterator_category

6.212.2.3 template<typename Tag_, typename Value_> typedef std::remove_-reference<Value_>::type* eos::WrappedForwardIterator< Tag_, Value_ >::pointer

6.212.2.4 template<typename Tag_, typename Value_> typedef std::remove_-reference<Value_>::type& eos::WrappedForwardIterator< Tag_, Value_ >::reference

6.212.2.5 template<typename Tag_, typename Value_> typedef std::remove_-reference<Value_>::type& eos::WrappedForwardIterator< Tag_, Value_ >::value_type

6.212.3 Constructor & Destructor Documentation

6.212.3.1 template<typename Tag_ , typename Value_ > eos::WrappedForwardIterator< Tag_, Value_ >::WrappedForwardIterator () `[inline]`

6.212.3.2 template<typename Tag_ , typename Value_ > eos::WrappedForwardIterator< Tag_, Value_ >::~WrappedForwardIterator () `[inline]`

6.212.3.3 template<typename Tag_ , typename Value_ > eos::WrappedForwardIterator< Tag_, Value_ >::WrappedForwardIterator (const WrappedForwardIterator< Tag_, Value_ > & *other*) `[inline]`

6.212.3.4 template<typename Tag_ , typename Value_ > template<typename T_ > eos::WrappedForwardIterator< Tag_, Value_ >::WrappedForwardIterator (const T_ & *base*) `[inline]`

6.212.4 Member Function Documentation

6.212.4.1 template<typename Tag_ , typename Value_ > bool eos::WrappedForwardIterator< Tag_, Value_ >::operator!= (const WrappedForwardIterator< Tag_, Value_ > & *other*) const `[inline]`

6.212.4.2 template<typename Tag_ , typename Value_ > WrappedForwardIterator< Tag_, Value_ >::reference eos::WrappedForwardIterator< Tag_, Value_ >::operator* () const `[inline]`

6.212.4.3 template<typename Tag_ , typename Value_ > WrappedForwardIterator< Tag_, Value_ > eos::WrappedForwardIterator< Tag_, Value_ >::operator++ (int) `[inline]`

6.212.4.4 template<typename Tag_ , typename Value_ > WrappedForwardIterator< Tag_, Value_ > & eos::WrappedForwardIterator< Tag_, Value_ >::operator++ () `[inline]`

6.212.4.5 template<typename Tag_ , typename Value_ > WrappedForwardIterator< Tag_, Value_ >::pointer eos::WrappedForwardIterator< Tag_, Value_ >::operator-> () const `[inline]`

6.212.4.6 template<typename Tag_ , typename Value_ > WrappedForwardIterator< Tag_, Value_ > & eos::WrappedForwardIterator< Tag_, Value_ >::operator= (const WrappedForwardIterator< Tag_, Value_ > & *other*) `[inline]`

- /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator.hh
- /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-impl.hh

Chapter 7

File Documentation

7.1 /home/alexshires/EOS/eos-next/eos/b-decays/b-to-pi-l-nu.hh File Reference

```
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::BToPiLeptonNeutrino

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_B_DECAYS_EXCLUSIVE_B_TO_U_L_NU_HH 1

7.1.1 Define Documentation

7.1.1.1 #define EOS_GUARD_EOS_B_DECAYS_EXCLUSIVE_B_TO_U_L_NU_HH 1

7.2 /home/alexshires/EOS/eos-next/eos/b-decays/properties.hh File Reference

```
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::BMesonProperties

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_B_DECAYS_PROPERTIES_HH 1

7.2.1 Define Documentation

7.2.1.1 #define EOS_GUARD_EOS_B_DECAYS_PROPERTIES_HH 1

7.3 /home/alexshires/EOS/eos-next/eos/constraint.hh File Reference

```
#include <eos/observable.hh>
#include <eos/utils/log_likelihood-fwd.hh>
#include <eos/utils/observable_cache.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/wrapped_forward_iterator.hh>
#include <vector>
```

Classes

- class eos::Constraint
- struct eos::UnknownConstraintError

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_CONSTRAINT_HH 1

7.3.1 Define Documentation

7.3.1.1 #define EOS_GUARD_SRC_CONSTRAINT_HH 1

7.4 /home/alexshires/EOS/eos-next/eos/form-factors/analytic-b-to-pi.hh File Reference

```
#include <eos/form-factors/mesonic.hh>
#include <eos/utils/diagnostics.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/options.hh>
```

Classes

- class eos::AnalyticFormFactorBToPiDKMMO2008

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_FORM_FACTORS_ANALYTIC_B_TO_PI_HH 1

7.4.1 Define Documentation

7.4.1.1 #define EOS_GUARD_EOS_FORM_FACTORS_ANALYTIC_B_TO_PI_HH 1

7.5 /home/alexshires/EOS/eos-next/eos/form-factors/baryonic-impl.hh File Reference

```
#include <eos/form-factors/baryonic.hh>
#include <eos/utils/kinematic.hh>
#include <eos/utils/options.hh>
#include <eos/utils/power_of.hh>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_FORM_FACTORS_BARYONIC_IMPL_HH 1

7.5.1 Define Documentation

7.5.1.1 #define EOS_GUARD_EOS_FORM_FACTORS_BARYONIC_IMPL_HH 1

7.6 /home/alexshires/EOS/eos-next/eos/form-factors/baryonic.hh File Reference

```
#include <eos/form-factors/form-factors-fwd.hh>
#include <eos/utils/parameters.hh>
#include <memory>
#include <string>
```

Classes

- struct eos::OneHalfPlusToOneHalfPlus
- class eos::FormFactors< OneHalfPlusToOneHalfPlus >
- class eos::FormFactorFactory< OneHalfPlusToOneHalfPlus >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_FORM_FACTORS_BARYONIC_HH 1

7.6.1 Define Documentation

7.6.1.1 #define EOS_GUARD_EOS_FORM_FACTORS_BARYONIC_HH 1

7.7 /home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh File Reference

```
#include <eos/observable.hh>
#include <eos/form-factors/form-factors.hh>
```

Classes

- class eos::FormFactorAdapter< Transition_ >
- class eos::FormFactorAdapterFactory< Transition_ >
- class eos::FormFactorRatioAdapter< Transition_ >
- class eos::FormFactorRatioAdapterFactory< Transition_ >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTOR_ADAPTER_HH 1

7.7.1 Define Documentation

7.7.1.1 #define EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTOR_ADAPTER_HH 1

7.8 /home/alexshires/EOS/eos-next/eos/form-factors/form-factors-fwd.hh File Reference

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTORS_FWD_HH 1

7.8.1 Define Documentation

7.8.1.1 #define EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTORS_FWD_HH 1

7.9 /home/alexshires/EOS/eos-next/eos/form-factors/form-factors.hh File Reference

```
#include <eos/form-factors/baryonic.hh>
#include <eos/form-factors/mesonic.hh>
```

Defines

- #define EOS_GUARD_SRC_FORM_FACTORS_FORM_FACTORS_HH 1

7.9.1 Define Documentation

7.9.1.1 #define EOS_GUARD_SRC_FORM_FACTORS_FORM_FACTORS_HH 1

7.10 /home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh File Reference

```
#include <eos/form-factors/mesonic.hh>
#include <eos/form-factors/analytic-b-to-pi.hh>
#include <eos/utils/derivative.hh>
#include <eos/utils/kinematic.hh>
#include <eos/utils/options.hh>
#include <eos/utils/power_of.hh>
```

Classes

- struct eos::BToKstar
- struct eos::BsToPhi
- class eos::BZ2004FormFactors< Process_, PToV >
- class eos::KMPW2010FormFactors< PToV >
- struct eos::BToK
- struct eos::BToPi
- class eos::BCL2008FormFactors< Process_ >
- class eos::BZ2004FormFactors< Process_, PToP >
- class eos::BZ2004FormFactorsSplit< Process_ >
- class eos::KMPW2010FormFactors< PToP >
- class eos::BFW2010FormFactors< PToP >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_FORM_FACTORS_IMPL_HH 1

7.10.1 Define Documentation

7.10.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_FORM_FACTORS_IMPL_HH 1

7.11 /home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh File Reference

```
#include <eos/form-factors/form-factors-fwd.hh>
#include <eos/utils/parameters.hh>
#include <memory>
#include <string>
```

Classes

- struct eos::PToV
- struct eos::PToP
- class eos::FormFactors< PToV >
- class eos::FormFactorFactory< PToV >
- class eos::FormFactors< PToP >
- class eos::FormFactorFactory< PToP >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_FORM_FACTORS_MESONIC_HH 1

7.11.1 Define Documentation

7.11.1.1 #define EOS_GUARD_SRC_FORM_FACTORS_MESONIC_HH 1

7.12 /home/alexshires/EOS/eos-next/eos/form-factors/pi-lcdas.hh File Reference

```
#include <eos/utils/diagnostics.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/options.hh>
```

Classes

- class eos::PionLCDAs

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_FORM_FACTORS_PI_LCDAS_HH 1

7.12.1 Define Documentation

7.12.1.1 #define EOS_GUARD_EOS_FORM_FACTORS_PI_LCDAS_HH 1

7.13 /home/alexshires/EOS/eos-next/eos/observable.hh File Reference

```
#include <eos/utils/exception.hh>
#include <eos/utils/kinematic.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <string>
#include <memory>
```

Classes

- class eos::Observable
- class eos::ObservableFactory
- struct eos::ObservableNameError

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_OBSERVABLE_HH 1

Typedefs

- typedef std::shared_ptr< Observable > eos::ObservablePtr

7.13.1 Define Documentation

7.13.1.1 #define EOS_GUARD_SRC_UTILS_OBSERVABLE_HH 1

7.14 /home/alexshires/EOS/eos-next/eos/rare-b-decays/bremsstrahlung.hh File Reference

```
#include <eos/utils/complex.hh>
```

Classes

- struct eos::Bremsstrahlung

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_BREMSSTRAHLUNG_HH 1

7.14.1 Define Documentation

7.14.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_BREMSSTRAHLUNG_HH 1

7.15 /home/alexshires/EOS/eos-next/eos/rare-b-decays/charm-loops.hh File Reference

```
#include <eos/utils/complex.hh>
#include <eos/utils/model.hh>
```

Classes

- struct eos::CharmLoops
- struct eos::ShortDistanceLowRecoil

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_CHARM_LOOPS_HH 1

7.15.1 Define Documentation

7.15.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_CHARM_LOOPS_HH 1

7.16 /home/alexshires/EOS/eos-next/eos/rare-b-decays/decays.hh File Reference

Classes

- struct eos::LargeRecoil
- struct eos::LowRecoil

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_DECAYS_HH 1

Enumerations

- enum eos::Helicity { eos::left_handed = -1, eos::right_handed = +1 }

7.16.1 Define Documentation

7.16.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_DECAYS_HH 1

7.17 /home/alexshires/EOS/eos-next/eos/rare-b-decays/em-contributions.hh File Reference

```
#include <eos/utils/complex.hh>
```

Classes

- struct eos::EMContributions

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_EM_CONTRIBUTIONS_HH 1

7.17.1 Define Documentation

7.17.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_EM_CONTRIBUTIONS_HH 1

7.18 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-dilepton.hh File Reference

```
#include <eos/observable.hh>
#include <eos/utils/complex.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::BToDilepton

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_DILEPTON_HH 1

7.18.1 Define Documentation

7.18.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_DILEPTON_HH 1

7.19 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-large-recoil.hh File Reference

```
#include <eos/rare-b-decays/decays.hh>
#include <eos/utils/complex.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::BToKstarDilepton< LargeRecoil >
- class eos::BToKDilepton< LargeRecoil >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_DILEPTON_-LARGE_RECOIL_HH 1

7.19.1 Define Documentation

7.19.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_DILEPTON_-LARGE_RECOIL_HH 1

7.20 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-low-recoil.hh File Reference

```
#include <eos/rare-b-decays/decays.hh>
#include <eos/utils/complex.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::BToKstarDilepton< LowRecoil >
- class eos::BToKDilepton< LowRecoil >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_DILEPTON_LOW_-RECOIL_HH 1

7.20.1 Define Documentation

7.20.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_DILEPTON_-LOW_RECOIL_HH 1

7.21 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton.hh File Reference

```
#include <eos/utils/power_of.hh>
#include <array>
```

Classes

- struct eos::btovll::Amplitudes
- struct eos::btovll::AngularCoefficients

Namespaces

- namespace eos
- namespace eos::btovll

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_DILEPTON_HH 1

Functions

- AngularCoefficients eos::btovll::array_to_angular_coefficients (const std::array< double, 12 > &arr)
- double eos::btovll::decay_width (const AngularCoefficients &a_c)
- std::array< double, 12 > eos::btovll::angular_coefficients_array (const Amplitudes &amplitudes, const double &s, const double &m_l)

7.21.1 Define Documentation

7.21.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_DILEPTON_HH 1

7.22 /home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-gamma.hh File Reference

```
#include <eos/rare-b-decays/decays.hh>
#include <eos/utils/complex.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::BToKstarGamma

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_GAMMA_HH 1

7.22.1 Define Documentation

7.22.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_EXCLUSIVE_B_TO_S_GAMMA_HH 1

7.23 /home/alexshires/EOS/eos-next/eos/rare-b-decays/hard-scattering.hh File Reference

```
#include <complex>
```

Classes

- struct eos::HardScattering

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_HARD_SCATTERING_HH 1

7.23.1 Define Documentation

7.23.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_HARD_SCATTERING_HH 1

7.24 /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-dilepton.hh File Reference

```
#include <eos/rare-b-decays/decays.hh>
#include <eos/utils/diagnostics.hh>
#include <eos/utils/complex.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- struct eos::HLMW2005
- class eos::BToXsDilepton< HLMW2005 >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_INCLUSIVE_B_TO_S_DILEPTON_HH 1

7.24.1 Define Documentation

7.24.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_INCLUSIVE_B_TO_S_DILEPTON_HH 1

7.25 /home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh File Reference

```
#include <eos/rare-b-decays/decays.hh>
#include <eos/utils/diagnostics.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- struct eos::Minimal
- class eos::BToXsGamma< Minimal >
- struct eos::NLO
- class eos::BToXsGamma< NLO >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_INCLUSIVE_B_TO_S_GAMMA_HH 1

7.25.1 Define Documentation

7.25.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_INCLUSIVE_B_TO_S_GAMMA_HH 1

7.26 /home/alexshires/EOS/eos-next/eos/rare-b-decays/lambda-b-to-lambda-dilepton.hh File Reference

```
#include <eos/rare-b-decays/decays.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::LambdaBToLambdaDilepton< LargeRecoil >
- class eos::LambdaBToLambdaDilepton< LowRecoil >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_RARE_B_DECAYS_BARYONIC_B_TO_S_DILEPTON_HH 1

7.26.1 Define Documentation

7.26.1.1 #define EOS_GUARD_EOS_RARE_B_DECAYS_BARYONIC_B_TO_S_DILEPTON_HH 1

7.27 /home/alexshires/EOS/eos-next/eos/rare-b-decays/long-distance.hh File Reference

```
#include <complex>
```

Classes

- struct eos::LongDistance

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_RARE_B_DECAYS_LONG_DISTANCE_HH 1

7.27.1 Define Documentation

7.27.1.1 #define EOS_GUARD_SRC_RARE_B_DECAYS_LONG_DISTANCE_HH 1

7.28 /home/alexshires/EOS/eos-next/eos/rare-b-decays/qcdf_-integrals.hh File Reference

```
#include <eos/utils/complex.hh>
```

Classes

- class eos::QCDFIntegrals
- struct eos::QCDFIntegrals::Results

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_RARE_B_DECAYS_QCDF_INTEGRALS_HH 1

7.28.1 Define Documentation

7.28.1.1 #define EOS_GUARD_EOS_RARE_B_DECAYS_QCDF_INTEGRALS_HH 1

7.29 /home/alexshires/EOS/eos-next/eos/utils/accumulator.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::Accumulator

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_ACCUMULATOR_HH 1

7.29.1 Define Documentation

7.29.1.1 #define EOS_GUARD_SRC_UTILS_ACCUMULATOR_HH 1

7.30 /home/alexshires/EOS/eos-next/eos/utils/analysis-fwd.hh File Reference

```
#include <memory>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_ANALYSIS_FWD_HH 1

Typedefs

- typedef std::shared_ptr< Analysis > eos::AnalysisPtr

7.30.1 Define Documentation

7.30.1.1 #define EOS_GUARD_SRC_UTILS_ANALYSIS_FWD_HH 1

7.31 /home/alexshires/EOS/eos-next/eos/utils/analysis.hh File Reference

```
#include <eos/utils/analysis-fwd.hh>
#include <eos/utils/hdf5-fwd.hh>
#include <eos/utils/log_likelihood.hh>
#include <eos/utils/log_prior.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/verify.hh>
#include <vector>
#include <set>
#include <gsl/gsl_multimin.h>
```

Classes

- class eos::Analysis
- struct eos::Analysis::OptimizationOptions
- struct eos::Analysis::Output

Namespaces

- namespace ROOT
- namespace ROOT::Minuit2
- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_ANALYSIS_HH 1

7.31.1 Define Documentation

7.31.1.1 #define EOS_GUARD_SRC_UTILS_ANALYSIS_HH 1

7.32 /home/alexshires/EOS/eos-next/eos/utils/analysis_TEST.hh File Reference

```
#include <eos/utils/analysis.hh>
#include <test/test.hh>
```

Classes

- struct eos::TestObservable
- struct eos::AbsoluteTestObservable

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_ANALYSIS_TEST_HH 1

Functions

- Analysis eos::make_analysis (bool flat)

7.32.1 Define Documentation

7.32.1.1 #define EOS_GUARD_SRC_UTILS_ANALYSIS_TEST_HH 1

7.33 /home/alexshires/EOS/eos-next/eos/utils/apply.hh File Reference

```
#include <functional>
#include <tuple>
```

Namespaces

- namespace eos
- namespace eos::impl

Defines

- #define EOS_GUARD_SRC_UTILS_APPLY_HH 1

Functions

- template<typename Result_ , typename... FunctionArgs_, typename... TupleElements_>
  Result_ eos::apply (Result_(∗f)(FunctionArgs_...), const std::tuple< TupleElements_...> &t)
- template<typename Result_ , typename... FunctionArgs_, typename... TupleElements_>
  Result_ eos::apply (const std::function< Result_(FunctionArgs_...)> &f, const std::tuple< TupleElements_...> &t)
- template<typename Result_ , typename Class_ , typename... FunctionArgs_, typename... TupleElements_>
  Result_ eos::apply (Result_(Class_::∗f)(FunctionArgs_...), const std::tuple< Class_ ∗, TupleElements_...> &t)

7.33.1 Define Documentation

7.33.1.1 #define EOS_GUARD_SRC_UTILS_APPLY_HH 1

7.34 /home/alexshires/EOS/eos-next/eos/utils/cartesian-product.hh File Reference

```
#include <vector>
```

Classes

- class eos::CartesianProduct< T_ >

  CartesianProduct is a template class to represent a Cartesian Product of an arbitrary number of T_. T_ should be a container providing a random access iterator.

- class **eos::CartesianProduct< T_ >::_Iterator< U_ >**

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_CARTESIAN_PRODUCT_HH 1

7.34.1 Define Documentation

7.34.1.1 #define EOS_GUARD_SRC_UTILS_CARTESIAN_PRODUCT_HH 1

7.35 /home/alexshires/EOS/eos-next/eos/utils/chi-squared.hh File Reference

```
#include <eos/utils/exception.hh>
#include <functional>
```

Classes

- struct eos::ChiSquared
- struct eos::NoSuchChiSquaredError

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_CHI_SQUARED_HH 1

7.35.1 Define Documentation

7.35.1.1 #define EOS_GUARD_SRC_UTILS_CHI_SQUARED_HH 1

7.36 /home/alexshires/EOS/eos-next/eos/utils/ckm_scan_model.hh File Reference

```
#include <eos/utils/model.hh>
#include <eos/utils/standard-model.hh>
```

Classes

- class eos::CKMScanComponent
- class eos::CKMScanModel

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_CKM_SCAN_MODEL_HH 1

7.36.1 Define Documentation

7.36.1.1 #define EOS_GUARD_SRC_UTILS_CKM_SCAN_MODEL_HH 1

7.37 /home/alexshires/EOS/eos-next/eos/utils/cluster.hh File Reference

```
#include <eos/utils/markov_chain.hh>
#include <eos/utils/private_implementation_pattern-impl.hh>
#include <eos/utils/wrapped_forward_iterator-impl.hh>
```

Classes

- class eos::Cluster

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_CLUSTER_HH 1

7.37.1 Define Documentation

7.37.1.1 #define EOS_GUARD_EOS_UTILS_CLUSTER_HH 1

7.38 /home/alexshires/EOS/eos-next/eos/utils/complex.hh File Reference

```
#include <complex>
```

Namespaces

- namespace eos

Defines

- #define WILSON_FITTER_GUARD_SRC_UTILS_COMPLEX_HH 1

7.38.1 Define Documentation

7.38.1.1 #define WILSON_FITTER_GUARD_SRC_UTILS_COMPLEX_HH 1

7.39 /home/alexshires/EOS/eos-next/eos/utils/concrete_observable.hh File Reference

```
#include <eos/observable.hh>
#include <eos/utils/apply.hh>
#include <functional>
#include <string>
```

Classes

- class eos::ConcreteObservable< Decay_, Args_ >
- class eos::ConcreteObservableFactory< Decay_, Args_ >

Namespaces

- namespace eos
- namespace eos::impl

Defines

- #define EOS_GUARD_SRC_UTILS_CONCRETE_OBSERVABLE_HH 1

Functions

- template<typename Decay_ , typename Tuple_ , typename... Args_>
  ObservableFactory * eos::make_concrete_observable_factory (const std::string &name, double(Decay_::*function)(const Args_ &...) const, const Tuple_ &kinematics_names=std::make_tuple())

7.39.1 Define Documentation

7.39.1.1 #define EOS_GUARD_SRC_UTILS_CONCRETE_OBSERVABLE_HH 1

7.40 /home/alexshires/EOS/eos-next/eos/utils/condition_variable.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <eos/utils/mutex.hh>
#include <pthread.h>
```

Classes

- class eos::ConditionVariable

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_CONDITION_VARIABLE_HH 1

7.40.1 Define Documentation

7.40.1.1 #define EOS_GUARD_SRC_UTILS_CONDITION_VARIABLE_HH 1

7.41 /home/alexshires/EOS/eos-next/eos/utils/derivative.hh File Reference

```
#include <functional>
```

Classes

- class eos::deriv::TwoSided

Namespaces

- namespace eos
- namespace eos::deriv

Defines

- #define BTOPI_FF_GUARD_EOS_UTILS_DERIVATIVE_HH 1

Functions

- template<unsigned order_, typename Method_ >
  double eos::derivative (const std::function< double(const double &)> &f, const double &x0)
- template<>
  double eos::derivative< 1u, deriv::TwoSided > (const std::function< double(const double &)> &f, const double &x0)
- template<>
  double eos::derivative< 2u, deriv::TwoSided > (const std::function< double(const double &)> &f, const double &x0)

7.41.1 Define Documentation

7.41.1.1 #define BTOPI_FF_GUARD_EOS_UTILS_DERIVATIVE_HH 1

7.42 /home/alexshires/EOS/eos-next/eos/utils/destringify.hh File Reference

```
#include <sstream>
```

Namespaces

- namespace eos
- namespace eos::impl

Defines

- #define EOS_GUARD_SRC_UTILS_DESTRINGIFY_HH 1

Functions

- template<typename T_ >
  T_ eos::destringify (const std::string &input)

7.42.1 Define Documentation

7.42.1.1 #define EOS_GUARD_SRC_UTILS_DESTRINGIFY_HH 1

7.43 /home/alexshires/EOS/eos-next/eos/utils/diagnostics.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/wrapped_forward_iterator.hh>
#include <string>
```

Classes

- class eos::Diagnostics
- struct eos::Diagnostics::Entry

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_DIAGNOSTICS_HH 1

7.43.1 Define Documentation

7.43.1.1 #define EOS_GUARD_EOS_UTILS_DIAGNOSTICS_HH 1

7.44 /home/alexshires/EOS/eos-next/eos/utils/equation_solver.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/verify.hh>
#include <functional>
#include <vector>
```

Classes

- class eos::EquationSolver
- struct eos::EquationSolver::Config
- struct eos::EquationSolver::Solution

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_EQUATION_SOLVER_HH 1

7.44.1 Define Documentation

7.44.1.1 #define EOS_GUARD_EOS_UTILS_EQUATION_SOLVER_HH 1

7.45 /home/alexshires/EOS/eos-next/eos/utils/exception.hh File Reference

```
#include <exception>
#include <string>
```

Classes

- class eos::Exception
- struct eos::InternalError

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_EXCEPTION_HH 1

7.45.1 Define Documentation

7.45.1.1 #define EOS_GUARD_SRC_UTILS_EXCEPTION_HH 1

7.46 /home/alexshires/EOS/eos-next/eos/utils/hdf5-fwd.hh File Reference

Namespaces

- namespace eos
- namespace eos::hdf5

Defines

- #define EOS_GUARD_SRC_UTILS_HDF5_FWD_HH 1

Functions

- template<typename T_ >
  DataSet< T_ > & eos::hdf5::operator<< (DataSet< T_ > &, const typename T_::Type &)
- template<typename T_ >
  DataSet< T_ > & eos::hdf5::operator>> (DataSet< T_ > &, typename T_::Type &)

7.46.1 Define Documentation

7.46.1.1 #define EOS_GUARD_SRC_UTILS_HDF5_FWD_HH 1

7.47 /home/alexshires/EOS/eos-next/eos/utils/hdf5.hh File Reference

```
#include <eos/utils/exception.hh>
#include <eos/utils/hdf5-fwd.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/wrapped_forward_iterator.hh>
#include <hdf5.h>
#include <cstring>
#include <vector>
```

Classes

- class eos::HDF5Error
- struct eos::hdf5::DataType< T_ >
- struct eos::hdf5::DataType< double >
- struct eos::hdf5::DataType< unsigned >
- struct eos::hdf5::DataType< int >
- struct eos::hdf5::DataType< short >
- struct eos::hdf5::DataType< const char * >
- struct eos::hdf5::Type
- class eos::hdf5::Scalar< T_ >
- class eos::hdf5::Array< rank_, T_ >
- struct eos::hdf5::CompositeCopier< index_, TypeTuple_, Tuple_ >
- struct eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ >
- class eos::hdf5::Composite< T_ >
- class eos::hdf5::FileHandle
- class eos::hdf5::DataSetHandle
- class eos::hdf5::AttributeHandle
- class eos::hdf5::File
- class eos::hdf5::DataSet< T_ >
- class eos::hdf5::Attribute< T_ >

Namespaces

- namespace eos
- namespace eos::hdf5

Defines

- #define EOS_GUARD_SRC_UTILS_HDF5_HH 1

Typedefs

- typedef int eos::hdf5::hid_t
- typedef std::shared_ptr< Type > eos::hdf5::TypePtr

Functions

- template<typename T_ >
  DataSet< T_ > & eos::hdf5::operator<< (DataSet< T_ > &, const typename T_::Type &)
- template<typename T_ >
  DataSet< T_ > & eos::hdf5::operator>> (DataSet< T_ > &, typename T_::Type &)

7.47.1 Define Documentation

7.47.1.1 #define EOS_GUARD_SRC_UTILS_HDF5_HH 1

7.48 /home/alexshires/EOS/eos-next/eos/utils/hierarchical-clustering.hh File Reference

```
#include <eos/utils/wrapped_forward_iterator-impl.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <vector>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_vector.h>
```

Classes

- class eos::HierarchicalClustering
- class eos::HierarchicalClustering::Component
- class eos::HierarchicalClustering::Config

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_HIERARCHICAL_CLUSTERING_HH 1

Functions

- std::ostream & eos::operator<< (std::ostream &lhs, const HierarchicalClustering::Component &rhs)

7.48.1 Define Documentation

7.48.1.1 #define EOS_GUARD_SRC_UTILS_HIERARCHICAL_CLUSTERING_HH 1

7.49 /home/alexshires/EOS/eos-next/eos/utils/histogram.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/wrapped_forward_iterator.hh>
#include <array>
```

Classes

- class eos::Histogram< 1 >
- struct eos::Histogram< 1 >::Bin
- class eos::Histogram< 2 >
- struct eos::Histogram< 2 >::Bin

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_HISTROGRAM_HH 1

Functions

- Histogram< 1 > eos::estimate_cumulative_distribution (const Histogram< 1 > &distribution)

7.49.1 Define Documentation

7.49.1.1 #define EOS_GUARD_SRC_UTILS_HISTROGRAM_HH 1

7.50 /home/alexshires/EOS/eos-next/eos/utils/instantiation_policy-impl.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <eos/utils/lock.hh>
#include <eos/utils/mutex.hh>
```

Classes

- class eos::InstantiationPolicy::DeleteOnDestruction< T_ >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_INSTANTIATION_POLICY_IMPL_HH 1

Variables

- T_ **const _ptr

7.50.1 Define Documentation

7.50.1.1 #define EOS_GUARD_SRC_UTILS_INSTANTIATION_POLICY_IMPL_HH 1

7.50.2 Variable Documentation

7.50.2.1 T_* * const _ptr

7.51 /home/alexshires/EOS/eos-next/eos/utils/instantiation_-policy.hh File Reference

Classes

- class eos::InstantiationPolicy< T_, NonCopyable >
- class eos::InstantiationPolicy< T_, NonInstantiable >
- class eos::InstantiationPolicy< T_, Singleton >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_INSTANTIATION_POLICY_HH 1

7.51.1 Define Documentation

7.51.1.1 #define EOS_GUARD_SRC_UTILS_INSTANTIATION_POLICY_HH 1

7.52 /home/alexshires/EOS/eos-next/eos/utils/integrate-impl.hh File Reference

```
#include <eos/utils/integrate.hh>
#include <eos/utils/matrix.hh>
#include <vector>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_INTEGRATE_IMPL_HH 1

Functions

- template<std::size_t k>
  std::array< double, k > eos::integrate (const std::function< std::array< double, k >(const double &)> &f, unsigned n, const double &a, const double &b)

7.52.1 Define Documentation

7.52.1.1 #define EOS_GUARD_EOS_UTILS_INTEGRATE_IMPL_HH 1

7.53 /home/alexshires/EOS/eos-next/eos/utils/integrate.hh File Reference

```
#include <eos/utils/complex.hh>
#include <array>
#include <functional>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_INTEGRATE_HH 1

Functions

- double eos::integrate (const std::function< double(const double &)> &f, unsigned n, const double &a, const double &b)
- complex< double > eos::integrate (const std::function< complex< double >(const double &)> &f, unsigned n, const double &a, const double &b)
- template<std::size_t k>
  std::array< double, k > eos::integrate (const std::function< std::array< double, k >(const double &)> &f, unsigned n, const double &a, const double &b)

7.53.1 Define Documentation

7.53.1.1 #define EOS_GUARD_SRC_UTILS_INTEGRATE_HH 1

7.54 /home/alexshires/EOS/eos-next/eos/utils/kinematic.hh File Reference

```
#include <eos/utils/exception.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- struct eos::UnknownKinematicVariableError
- class eos::Kinematics
- class eos::KinematicVariable

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_KINEMATIC_HH 1

Functions

- double eos::lambda (const double &a, const double &b, const double &c)

7.54.1 Define Documentation

7.54.1.1 #define EOS_GUARD_SRC_UTILS_KINEMATIC_HH 1

7.55 /home/alexshires/EOS/eos-next/eos/utils/lcda.hh File Reference

```
#include <eos/utils/qcd.hh>
```

Classes

- struct eos::LCDA

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_LCDA_HH 1

7.55.1 Define Documentation

7.55.1.1 #define EOS_GUARD_SRC_UTILS_LCDA_HH 1

7.56 /home/alexshires/EOS/eos-next/eos/utils/lock.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <eos/utils/mutex.hh>
```

Classes

- class eos::Lock
- class eos::TryLock

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_LOCK_HH 1

7.56.1 Define Documentation

7.56.1.1 #define EOS_GUARD_SRC_UTILS_LOCK_HH 1

7.57 /home/alexshires/EOS/eos-next/eos/utils/log.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/stringify.hh>
```

Classes

- class eos::Log
- class eos::LogMessageHandler

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_LOG_HH 1

Enumerations

- enum eos::LogLevel {
  eos::ll_silent, eos::ll_error, eos::ll_warning, eos::ll_informational,
  eos::ll_debug, eos::ll_last }

Functions

- std::ostream & eos::operator<< (std::ostream &lhs, const LogLevel &rhs)
- std::istream & eos::operator>> (std::istream &lhs, LogLevel &rhs)

7.57.1 Define Documentation

7.57.1.1 #define EOS_GUARD_SRC_UTILS_LOG_HH 1

7.58 /home/alexshires/EOS/eos-next/eos/utils/log_likelihood-fwd.hh File Reference

```
#include <memory>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_LOG_LIKELIHOOD_FWD_HH 1

Typedefs

- typedef std::shared_ptr< LogLikelihoodBlock > eos::LogLikelihoodBlockPtr

7.58.1 Define Documentation

7.58.1.1 #define EOS_GUARD_SRC_UTILS_LOG_LIKELIHOOD_FWD_HH 1

7.59 /home/alexshires/EOS/eos-next/eos/utils/log_likelihood.hh File Reference

```
#include <eos/constraint.hh>
#include <eos/observable.hh>
#include <eos/utils/log_likelihood-fwd.hh>
#include <eos/utils/matrix.hh>
#include <eos/utils/observable_cache.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <gsl/gsl_rng.h>
```

Classes

- class eos::LogLikelihoodBlock
- class eos::LogLikelihood

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_LIKELIHOOD_HH 1

7.59.1 Define Documentation

7.59.1.1 #define EOS_GUARD_SRC_UTILS_LIKELIHOOD_HH 1

7.60 /home/alexshires/EOS/eos-next/eos/utils/log_prior-fwd.hh File Reference

```
#include <memory>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_PRIOR_FWD_HH 1

Typedefs

- typedef std::shared_ptr< LogPrior > eos::LogPriorPtr

7.60.1 Define Documentation

7.60.1.1 #define EOS_GUARD_SRC_UTILS_PRIOR_FWD_HH 1

7.61 /home/alexshires/EOS/eos-next/eos/utils/log_prior.hh File Reference

```
#include <eos/utils/log_prior-fwd.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/wrapped_forward_iterator.hh>
#include <vector>
#include <memory>
#include <gsl/gsl_rng.h>
```

Classes

- class eos::LogPrior

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_PRIOR_HH 1

7.61.1 Define Documentation

7.61.1.1 #define EOS_GUARD_SRC_UTILS_PRIOR_HH 1

7.62 /home/alexshires/EOS/eos-next/eos/utils/markov_chain.hh File Reference

```
#include <eos/utils/analysis-fwd.hh>
#include <eos/utils/hdf5-fwd.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/stringify.hh>
#include <vector>
#include <gsl/gsl_rng.h>
```

Classes

- class eos::MarkovChain
- struct eos::MarkovChain::HyperParameter
- struct eos::MarkovChain::State
- struct eos::MarkovChain::Stats
- struct eos::MarkovChain::History
- struct eos::MarkovChain::ProposalFunction

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_MARKOV_CHAIN_HH 1

Typedefs

- typedef std::shared_ptr< MarkovChain::History > eos::HistoryPtr
- typedef std::shared_ptr< MarkovChain::ProposalFunction > eos::ProposalFunctionPtr

Functions

- std::ostream & eos::operator<< (std::ostream &lhs, const MarkovChain::State &rhs)

7.62.1 Define Documentation

7.62.1.1 #define EOS_GUARD_SRC_UTILS_MARKOV_CHAIN_HH 1

7.63 /home/alexshires/EOS/eos-next/eos/utils/markov_chain_-sampler.hh File Reference

```
#include <eos/utils/analysis.hh>
#include <eos/utils/markov_chain.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/proposal_functions.hh>
#include <eos/utils/verify.hh>
```

Classes

- class eos::MarkovChainSampler
- class eos::MarkovChainSampler::Config
- struct eos::MarkovChainSampler::PreRunInfo

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_MC_SAMPLER_HH 1

Functions

- std::ostream & eos::operator<< (std::ostream &, const MarkovChainSampler::Config &config)

7.63.1 Define Documentation

7.63.1.1 #define EOS_GUARD_SRC_UTILS_MC_SAMPLER_HH 1

7.64 /home/alexshires/EOS/eos-next/eos/utils/matrix.hh File Reference

```
#include <array>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_MATRIX_HH 1

Functions

- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< std::array< T_, n_ >, m_ > eos::operator+ (const std::array< std::array< T_, n_ >, m_ > &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< std::array< T_, n_ >, m_ > eos::operator- (const std::array< std::array< T_, n_ >, m_ > &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t m_>
  std::array< T_, m_ > eos::operator+ (const std::array< T_, m_ > &x, const std::array< T_, m_ > &y)
- template<typename T_ , std::size_t m_>
  std::array< T_, m_ > eos::operator- (const std::array< T_, m_ > &x, const std::array< T_, m_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_, std::size_t o_>
  std::array< std::array< T_, n_ >, m_ > eos::operator∗ (const std::array< std::array< T_, o_ >, m_ > &x, const std::array< std::array< T_, n_ >, o_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< T_, m_ > eos::operator∗ (const std::array< std::array< T_, n_ >, m_ > &x, const std::array< T_, n_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< T_, n_ > eos::operator^ (const std::array< T_, m_ > &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t m_, std::size_t n_>
  std::array< std::array< T_, n_ >, m_ > eos::operator∗ (const T_ &x, const std::array< std::array< T_, n_ >, m_ > &y)
- template<typename T_ , std::size_t n_>
  std::array< T_, n_ > eos::operator∗ (const T_ &x, const std::array< T_, n_ > &y)
- template<typename T_ , std::size_t n_>
  T_ eos::dot (const std::array< T_, n_ > &x, const std::array< T_, n_ > &y)
- template<typename T_ , std::size_t n_>
  std::array< T_, n_ > eos::mult (const std::array< T_, n_ > &x, const std::array< T_, n_ > &y)
- template<typename T_ , std::size_t n_>
  std::array< T_, n_ > eos::divide (const std::array< T_, n_ > &x, const std::array< T_, n_ > &y)

7.64.1 Define Documentation

7.64.1.1 #define EOS_GUARD_SRC_UTILS_MATRIX_HH 1

7.65 /home/alexshires/EOS/eos-next/eos/utils/memoise.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <eos/utils/instantiation_policy-impl.hh>
#include <eos/utils/lock.hh>
#include <eos/utils/mutex.hh>
#include <cstdint>
#include <functional>
#include <tuple>
#include <unordered_map>
#include <vector>
```

Classes

- struct eos::implementation::ResultOf< Result_(Class_::*)(Args_...)>
- struct eos::implementation::ResultOf< Result_(*)(Args_...)>
- class eos::MemoisationControl
- class eos::Memoiser< Result_, Params_ >

Namespaces

- namespace std

  STL namespace.

- namespace eos
- namespace eos::implementation

Defines

- #define EOS_GUARD_SRC_UTILS_MEMOISE_HH 1

Functions

- template<typename FunctionType_ , typename... Params>
  implementation::ResultOf< FunctionType_ >::Type eos::memoise (FunctionType_ f, const Params &...p)
- template<typename FunctionType_ , typename... Params>
  unsigned eos::number_of_memoisations (FunctionType_, const Params &...)

7.65.1 Define Documentation

7.65.1.1 #define EOS_GUARD_SRC_UTILS_MEMOISE_HH 1

7.66 /home/alexshires/EOS/eos-next/eos/utils/model.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/options.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/wilson_coefficients.hh>
#include <complex>
```

Classes

- class eos::ModelComponent< components::CKM >
- class eos::ModelComponent< components::QCD >
- class eos::ModelComponent< components::DeltaB1 >
- class eos::Model
- struct eos::NoSuchModelError

Namespaces

- namespace eos
- namespace eos::components

Defines

- #define EOS_GUARD_SRC_UTILS_MODEL_HH 1

7.66.1 Define Documentation

7.66.1.1 #define EOS_GUARD_SRC_UTILS_MODEL_HH 1

7.67 /home/alexshires/EOS/eos-next/eos/utils/mutable-fwd.hh File Reference

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_MUTABLE_FWD_HH 1

7.67.1 Define Documentation

7.67.1.1 #define EOS_GUARD_EOS_UTILS_MUTABLE_FWD_HH 1

7.68 /home/alexshires/EOS/eos-next/eos/utils/mutable.hh File Reference

```
#include <eos/utils/mutable-fwd.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <memory>
```

Classes

- class eos::Mutable

  Base class for all mutable entities, e.g. a Parameter.

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_MUTABLE_HH 1

Typedefs

- typedef std::shared_ptr< Mutable > eos::MutablePtr

7.68.1 Define Documentation

7.68.1.1 #define EOS_GUARD_EOS_UTILS_MUTABLE_HH 1

7.69 /home/alexshires/EOS/eos-next/eos/utils/mutex.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <pthread.h>
```

Classes

- class eos::Mutex

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_MUTEX_HH 1

7.69.1 Define Documentation

7.69.1.1 #define EOS_GUARD_SRC_UTILS_MUTEX_HH 1

7.70 /home/alexshires/EOS/eos-next/eos/utils/observable_cache.hh File Reference

```
#include <eos/observable.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::ObservableCache

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_OBSERVABLE_CACHE_HH 1

7.70.1 Define Documentation

7.70.1.1 #define EOS_GUARD_SRC_UTILS_OBSERVABLE_CACHE_HH 1

7.71 /home/alexshires/EOS/eos-next/eos/utils/observable_set.hh File Reference

```
#include <eos/observable.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/wrapped_forward_iterator.hh>
```

Classes

- class eos::ObservableSet

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_OBSERVABLE_SET_HH 1

7.71.1 Define Documentation

7.71.1.1 #define EOS_GUARD_SRC_UTILS_OBSERVABLE_SET_HH 1

7.72 /home/alexshires/EOS/eos-next/eos/utils/one-of.hh File Reference

```
#include <cstdlib>
#include <memory>
#include <string>
#include <utility>
```

Classes

- struct eos::SelectOneOfType< Want_ >
- struct eos::SelectOneOfType< Want_, Try_, Rest_...>
- struct eos::OneOfVisitorVisit< Type_ >
- struct eos::OneOfVisitor< Types_ >
- struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >
- struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >
- struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>
- struct eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>
- struct eos::OneOfVisitorWrapper< Underlying_, Result_, Types_ >
- struct eos::OneOfValueBase< Types_ >
- struct eos::OneOfValue< Type_, Types_ >
- class eos::OneOf< Types_ >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_ONE_OF_HH 1

7.72.1 Define Documentation

7.72.1.1 #define EOS_GUARD_SRC_UTILS_ONE_OF_HH 1

7.73 /home/alexshires/EOS/eos-next/eos/utils/options.hh File Reference

```
#include <eos/utils/exception.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <string>
```

Classes

- struct eos::UnknownOptionError
- class eos::Options

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_OPTIONS_HH 1

Functions

- Options eos::operator+ (const Options &lhs, const Options &rhs)

  Merge operator.

7.73.1 Define Documentation

7.73.1.1 #define EOS_GUARD_SRC_UTILS_OPTIONS_HH 1

7.74 /home/alexshires/EOS/eos-next/eos/utils/parameters-fwd.hh File Reference

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_PARAMETERS_FWD_HH 1

7.74.1 Define Documentation

7.74.1.1 #define EOS_GUARD_EOS_UTILS_PARAMETERS_FWD_HH 1

7.75 /home/alexshires/EOS/eos-next/eos/utils/parameters.hh File Reference

```
#include <eos/utils/exception.hh>
#include <eos/utils/mutable.hh>
#include <eos/utils/parameters-fwd.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/random_number_generator.hh>
#include <eos/utils/wrapped_forward_iterator.hh>
#include <set>
```

Classes

- struct eos::UnknownParameterError
- class eos::Parameters
- class eos::Parameter
- class eos::ParameterUser
- class eos::UsedParameter
- struct eos::ParameterDescription
- struct eos::ParameterRange

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_PARAMETERS_HH 1

7.75.1 Define Documentation

7.75.1.1 #define EOS_GUARD_SRC_UTILS_PARAMETERS_HH 1

7.76 /home/alexshires/EOS/eos-next/eos/utils/polylog.hh File Reference

```
#include <eos/utils/complex.hh>
```

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_POLYLOG_HH 1

Functions

- complex< double > eos::dilog (const complex< double > &z) __attribute__((pure))
- complex< double > eos::trilog (const complex< double > &z) __attribute__((pure))

7.76.1 Define Documentation

7.76.1.1 #define EOS_GUARD_EOS_UTILS_POLYLOG_HH 1

7.77 /home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_sampler.hh File Reference

```
#include <eos/utils/analysis.hh>
#include <eos/utils/hdf5-fwd.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::PopulationMonteCarloSampler
- struct eos::PopulationMonteCarloSampler::Config
- struct eos::PopulationMonteCarloSampler::Output
- struct eos::PopulationMonteCarloSampler::Status

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_PMC_SAMPLER_HH 1

Functions

- std::ostream & eos::operator<< (std::ostream &, const PopulationMonteCarloSampler::Config &)

7.77.1 Define Documentation

7.77.1.1 #define EOS_GUARD_SRC_UTILS_PMC_SAMPLER_HH 1

7.78 /home/alexshires/EOS/eos-next/eos/utils/power_of.hh File Reference

```
#include <complex>
```

Namespaces

- namespace eos
- namespace eos::impl

Defines

- #define EOS_GUARD_SRC_UTILS_POWER_OF_HH 1

Functions

- template<unsigned n_, typename T_ >
  T_ eos::power_of (const T_ &x)

7.78.1 Define Documentation

7.78.1.1 #define EOS_GUARD_SRC_UTILS_POWER_OF_HH 1

7.79 /home/alexshires/EOS/eos-next/eos/utils/prior_sampler.hh File Reference

```
#include <eos/utils/log_prior-fwd.hh>
#include <eos/utils/observable_set.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/hdf5-fwd.hh>
#include <vector>
```

Classes

- class eos::PriorSampler
- struct eos::PriorSampler::Config

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_PRIOR_SAMPLER_HH 1

7.79.1 Define Documentation

7.79.1.1 #define EOS_GUARD_SRC_UTILS_PRIOR_SAMPLER_HH 1

7.80 /home/alexshires/EOS/eos-next/eos/utils/private_-implementation_pattern-impl.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
```

Namespaces

- namespace eos

Defines

- #define WILSON_FITTER_GUARD_UTILS_PRIVATE_IMPLEMENTATION_PATTERN_-IMPL_HH 1

7.80.1 Define Documentation

7.80.1.1 #define WILSON_FITTER_GUARD_UTILS_PRIVATE_IMPLEMENTATION_-PATTERN_IMPL_HH 1

7.81 /home/alexshires/EOS/eos-next/eos/utils/private_-implementation_pattern.hh File Reference

```
#include <memory>
```

Classes

- class eos::PrivateImplementationPattern< T_ >

Namespaces

- namespace eos

Defines

- #define WILSON_FITTER_GUARD_UTILS_PRIVATE_IMPLEMENTATION_PATTERN_HH 1

7.81.1 Define Documentation

7.81.1.1 #define WILSON_FITTER_GUARD_UTILS_PRIVATE_IMPLEMENTATION_-PATTERN_HH 1

7.82 /home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh File Reference

```
#include <eos/utils/cluster.hh>
#include <eos/utils/hdf5-fwd.hh>
#include <eos/utils/log_prior-fwd.hh>
#include <eos/utils/markov_chain.hh>
#include <eos/utils/verify.hh>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_vector.h>
```

Classes

- struct eos::proposal_functions::UnknownProposalError
- struct eos::proposal_functions::Factory

  Abstract factory to read in a proposal density from file.

- class eos::proposal_functions::AdjacencyMatrix
- class eos::proposal_functions::GlobalLocal
- struct eos::proposal_functions::GlobalLocal::Config
- class eos::proposal_functions::Multivariate
- class eos::proposal_functions::MultivariateGaussian
- class eos::proposal_functions::MultivariateStudentT
- struct eos::proposal_functions::MultivariateAccess
- class eos::proposal_functions::BlockDecomposition

Namespaces

- namespace eos
- namespace eos::proposal_functions

Defines

- #define EOS_GUARD_EOS_UTILS_PROPROSAL_FUNCTIONS_HH 1

Typedefs

- typedef hdf5::Composite< hdf5::Scalar< const char * >, hdf5::Scalar< unsigned > > eos::proposal_functions::MetaType
- typedef std::tuple< const char *, unsigned > eos::proposal_functions::MetaRecord
- typedef std::shared_ptr< Multivariate > eos::proposal_functions::MultivariateProposalPtr

Functions

- unsigned eos::proposal_functions::random_index (const std::vector< double > &cumulative, gsl_rng *rng)
- void eos::proposal_functions::sliding_window (const unsigned &K, const unsigned &size, const unsigned &j, unsigned &j_min, unsigned &j_max)
- MetaType eos::proposal_functions::meta_type ()
- MetaRecord eos::proposal_functions::meta_record ()

7.82.1 Define Documentation

7.82.1.1 #define EOS_GUARD_EOS_UTILS_PROPROSAL_FUNCTIONS_HH 1

7.83 /home/alexshires/EOS/eos-next/eos/utils/qcd.hh File Reference

```
#include <array>
```

Classes

- class eos::QCD

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_QCD_HH 1

7.83.1 Define Documentation

7.83.1.1 #define EOS_GUARD_SRC_UTILS_QCD_HH 1

7.84 /home/alexshires/EOS/eos-next/eos/utils/random_number_-generator.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
#include <cstdint>
```

Classes

- class eos::RandomNumberGenerator

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_RANDOM_NUMBER_GENERATOR_HH 1

7.84.1 Define Documentation

7.84.1.1 #define EOS_GUARD_SRC_UTILS_RANDOM_NUMBER_GENERATOR_HH 1

7.85 /home/alexshires/EOS/eos-next/eos/utils/rvalue.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <vector>
```

Classes

- struct eos::RValue

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_RVALUE_HH 1

7.85.1 Define Documentation

7.85.1.1 #define EOS_GUARD_SRC_UTILS_RVALUE_HH 1

7.86 /home/alexshires/EOS/eos-next/eos/utils/save.hh File Reference

Classes

- class eos::Save< V_, R_ >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_SAVE_HH 1

7.86.1 Define Documentation

7.86.1.1 #define EOS_GUARD_SRC_UTILS_SAVE_HH 1

7.87 /home/alexshires/EOS/eos-next/eos/utils/standard-model.hh File Reference

```
#include <eos/utils/model.hh>
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::SMComponent< components::CKM >
- class eos::SMComponent< components::QCD >
- class eos::SMComponent< components::DeltaB1 >
- class eos::StandardModel

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_STANDARD_MODEL_HH 1

7.87.1 Define Documentation

7.87.1.1 #define EOS_GUARD_SRC_UTILS_STANDARD_MODEL_HH 1

7.88 /home/alexshires/EOS/eos-next/eos/utils/stringify.hh File Reference

```
#include <string>
#include <sstream>
```

Classes

- struct eos::implementation::DoStringify< T_ >

Namespaces

- namespace eos
- namespace eos::implementation

Defines

- #define EOS_GUARD_SRC_UTILS_STRINGIFY_HH 1

Functions

- template<typename T_ >
  std::string eos::stringify (const T_ &x, const unsigned &precision=10)
- template<typename Iterator_ >
  std::string eos::stringify (const Iterator_ &begin, const Iterator_ &end, const unsigned &precision=10)
- template<typename T_ >
  std::string eos::stringify (const T_ *m, const unsigned &dim, const unsigned &precision=10)
- template<typename Container_ >
  std::string eos::stringify_container (const Container_ &container, unsigned precision=10)

7.88.1 Define Documentation

7.88.1.1 #define EOS_GUARD_SRC_UTILS_STRINGIFY_HH 1

7.89 /home/alexshires/EOS/eos-next/eos/utils/thread.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <tr1/functional>
```

Classes

- class eos::Thread

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_THREAD_HH 1

7.89.1 Define Documentation

7.89.1.1 #define EOS_GUARD_SRC_UTILS_THREAD_HH 1

7.90 /home/alexshires/EOS/eos-next/eos/utils/thread_pool.hh File Reference

```
#include <eos/utils/instantiation_policy.hh>
#include <eos/utils/private_implementation_pattern.hh>
#include <eos/utils/ticket.hh>
#include <tr1/functional>
```

Classes

- class eos::ThreadPool

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_THREAD_POOL_HH 1

7.90.1 Define Documentation

7.90.1.1 #define EOS_GUARD_SRC_UTILS_THREAD_POOL_HH 1

7.91 /home/alexshires/EOS/eos-next/eos/utils/ticket.hh File Reference

```
#include <eos/utils/private_implementation_pattern.hh>
```

Classes

- class eos::Ticket
- class eos::TicketList

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_TICKET_HH 1

7.91.1 Define Documentation

7.91.1.1 #define EOS_GUARD_SRC_UTILS_TICKET_HH 1

7.92 /home/alexshires/EOS/eos-next/eos/utils/top-loops.hh File Reference

Classes

- struct eos::TopLoops

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_TOP_LOOPS_HH 1

7.92.1 Define Documentation

7.92.1.1 #define EOS_GUARD_SRC_UTILS_TOP_LOOPS_HH 1

7.93 /home/alexshires/EOS/eos-next/eos/utils/verify.hh File Reference

```
#include <eos/utils/exception.hh>
#include <eos/utils/stringify.hh>
```

Classes

- class eos::VerifiedRangeError
- struct eos::VerifiedRangeOverflow
- struct eos::VerifiedRangeUnderflow
- class eos::VerifiedRange< T_ >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_VERIFY_HH 1

7.93.1 Define Documentation

7.93.1.1 #define EOS_GUARD_SRC_UTILS_VERIFY_HH 1

7.94 /home/alexshires/EOS/eos-next/eos/utils/welford.hh File Reference

Classes

- struct eos::Welford

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_EOS_UTILS_WELFORD_HH 1

7.94.1 Define Documentation

7.94.1.1 #define EOS_GUARD_EOS_UTILS_WELFORD_HH 1

7.95 /home/alexshires/EOS/eos-next/eos/utils/wilson-polynomial.hh File Reference

```
#include <eos/observable.hh>
#include <eos/utils/one-of.hh>
#include <list>
#include <string>
```

Classes

- class eos::WilsonPolynomialCloner
- class eos::WilsonPolynomialPrinter
- class eos::WilsonPolynomialEvaluator

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_WILSON_POLYNOMIAL_HH 1

Typedefs

- typedef OneOf< Constant, Sum, Product, Sine, Cosine, Parameter > eos::WilsonPolynomial

Functions

- WilsonPolynomial eos::make_polynomial (const ObservablePtr &, const std::list< std::string > &)
- ObservablePtr eos::make_polynomial_observable (const WilsonPolynomial &polynomial, const Parameters &parameters)
- ObservablePtr eos::make_polynomial_ratio (const WilsonPolynomial &numerator, const WilsonPolynomial &denominator, const Parameters &parameters)
- ObservablePtr eos::make_polynomial_ht_like_ratio (const WilsonPolynomial &numerator, const WilsonPolynomial &denominator, const WilsonPolynomial &denominator2, const Parameters &parameters)

7.95.1 Define Documentation

7.95.1.1 #define EOS_GUARD_SRC_UTILS_WILSON_POLYNOMIAL_HH 1

7.96 /home/alexshires/EOS/eos-next/eos/utils/wilson_coefficients.hh File Reference

```
#include <eos/utils/complex.hh>
#include <eos/utils/parameters.hh>
#include <eos/utils/qcd.hh>
#include <array>
#include <cmath>
```

Classes

- struct eos::BToS
- struct eos::WilsonCoefficients< BToS >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_WILSON_COEFFICIENTS_HH 1

Functions

- WilsonCoefficients< BToS > eos::evolve (const std::array< complex< double >, 15 > &wc_qcd_-0, const std::array< complex< double >, 15 > &wc_qcd_1, const std::array< complex< double >, 15 > &wc_qcd_2, const double &alpha_s_0, const double &alpha_s, const double &nf, const QCD::BetaFunction &beta)

7.96.1 Define Documentation

7.96.1.1 #define EOS_GUARD_SRC_UTILS_WILSON_COEFFICIENTS_HH 1

7.97 /home/alexshires/EOS/eos-next/eos/utils/wilson_scan_model.hh File Reference

```
#include <eos/utils/model.hh>
#include <eos/utils/standard-model.hh>
```

Classes

- class eos::WilsonScanComponent
- class eos::WilsonScanModel

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_SRC_UTILS_WILSON_SCAN_MODEL_HH 1

7.97.1 Define Documentation

7.97.1.1 #define EOS_GUARD_SRC_UTILS_WILSON_SCAN_MODEL_HH 1

7.98 /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_-iterator-fwd.hh File Reference

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_UTIL_WRAPPED_FORWARD_ITERATOR_FWD_HH 1

7.98.1 Define Documentation

7.98.1.1 #define EOS_GUARD_UTIL_WRAPPED_FORWARD_ITERATOR_FWD_HH 1

7.99 /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_-iterator-impl.hh File Reference

```
#include <eos/utils/wrapped_forward_iterator.hh>
```

Classes

- struct eos::WrappedForwardIterator< Tag_, Value_ >::Base< Tag_, Value_ >
- struct eos::WrappedForwardIterator< Tag_, Value_ >::BaseImpl< Tag_, Value_ >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_UTIL_WRAPPED_FORWARD_ITERATOR_IMPL_HH 1

7.99.1 Define Documentation

7.99.1.1 #define EOS_GUARD_UTIL_WRAPPED_FORWARD_ITERATOR_IMPL_HH 1

7.100 /home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_-iterator.hh File Reference

```
#include <eos/utils/wrapped_forward_iterator-fwd.hh>
#include <functional>
#include <iterator>
#include <type_traits>
```

Classes

- class eos::WrappedForwardIterator< Tag_, Value_ >

Namespaces

- namespace eos

Defines

- #define EOS_GUARD_UTILS_WRAPPED_FORWARD_ITERATOR_HH 1

7.100.1 Define Documentation

7.100.1.1 #define EOS_GUARD_UTILS_WRAPPED_FORWARD_ITERATOR_HH 1

Chapter 8

Example Documentation

8.1 For

Calculate the significance of the deviation between the observables' current value and the mode in units of the standard Gaussian distribution.

a Gaussian around x=1 with sigma = 0.5, a current value of x=2 will yield a significance of 2.

Note:

The significance is >= 0.

8.2 if

Maximum relative distance between two local modes that allows them to be treated as one mode. A rescaling of coordinates to the unit hypercube is performed in order to avoid problems when parameter ranges differ by several orders of magnitude.

|x - y|/|y| < eps, they are just one mode

8.3 index_list

Ignore any differences in dimensions other than those given.

= { 2 }; then only jumps occur only in the 2nd dimension.

8.4 print_steps

Determines how often PMC prints outs "Done x%" during sampling of one chunk. = 20, then pmc prints Done 20% Done 40% Done 60% Done 80%

Index

~AbsoluteTestObservable
eos::AbsoluteTestObservable, 51
~Accumulator
eos::Accumulator, 53
~Analysis
eos::Analysis, 58
~AnalyticFormFactorBToPiDKMMO2008
eos::AnalyticFormFactorBToPiDKMMO2008, 62
~AttributeHandle
eos::hdf5::AttributeHandle, 68
~BMesonProperties
eos::BMesonProperties, 82
~BToDilepton
eos::BToDilepton, 86
~BToKDilepton
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
~BToKstarDilepton
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
~BToKstarGamma
eos::BToKstarGamma, 104
~BToPiLeptonNeutrino
eos::BToPiLeptonNeutrino, 107
~BToXsDilepton
eos::BToXsDilepton< HLMW2005 >, 109
~BToXsGamma
eos::BToXsGamma< Minimal >, 110
eos::BToXsGamma< NLO >, 111
~Base
eos::WrappedForwardIterator::Base, 69
~Bin
eos::Histogram< 1 >::Bin, 77
eos::Histogram< 2 >::Bin, 75
~BlockDecomposition
eos::proposal_functions::BlockDecomposition, 80
~CKMScanModel
eos::CKMScanModel, 125
~Cluster
eos::Cluster, 127
~Component
eos::HierarchicalClustering::Component, 129
~Composite
eos::hdf5::Composite, 130
~ConcreteObservableFactory
eos::ConcreteObservableFactory, 136
~ConditionVariable
eos::ConditionVariable, 137
~Constraint
eos::Constraint, 163
~DataSet
eos::hdf5::DataSet, 165
~DataSetHandle
eos::hdf5::DataSetHandle, 168
~Diagnostics
eos::Diagnostics, 177
~EquationSolver
eos::EquationSolver, 181
~Exception
eos::Exception, 183
~File
eos::hdf5::File, 185
~FileHandle
eos::hdf5::FileHandle, 188
~FormFactorAdapterFactory
eos::FormFactorAdapterFactory, 191
~FormFactorRatioAdapterFactory
eos::FormFactorRatioAdapterFactory, 197
~FormFactors
eos::FormFactors< OneHalfPlusToOneHalfPlus >, 198
eos::FormFactors< PToP >, 199
eos::FormFactors< PToV >, 201
~GlobalLocal
eos::proposal_functions::GlobalLocal, 205
~HierarchicalClustering
eos::HierarchicalClustering, 211
~Histogram
eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
~KinematicVariable
eos::KinematicVariable, 230
~Kinematics
eos::Kinematics, 228
~LambdaBToLambdaDilepton
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236

eos::LambdaBToLambdaDilepton< LowRecoil >, 238
~Lock
eos::Lock, 241
~Log
eos::Log, 242
~LogLikelihood
eos::LogLikelihood, 245
~LogLikelihoodBlock
eos::LogLikelihoodBlock, 249
~LogMessageHandler
eos::LogMessageHandler, 255
~LogPrior
eos::LogPrior, 257
~MarkovChain
eos::MarkovChain, 265
~MarkovChainSampler
eos::MarkovChainSampler, 270
~MemoisationControl
eos::MemoisationControl, 273
~Memoiser
eos::Memoiser, 274
~Model
eos::Model, 276
~Multivariate
eos::proposal_functions::Multivariate, 283
~MultivariateGaussian
eos::proposal_functions::MultivariateGaussian, 289
~MultivariateStudentT
eos::proposal_functions::MultivariateStudentT, 292
~Mutable
eos::Mutable, 294
~Mutex
eos::Mutex, 296
~ObservableCache
eos::ObservableCache, 303
~ObservableFactory
eos::ObservableFactory, 305
~ObservableSet
eos::ObservableSet, 308
~OneOfValueBase
eos::OneOfValueBase, 313
~Options
eos::Options, 324
~Parameter
eos::Parameter, 330
~Parameters
eos::Parameters, 335
~PionLCDAs
eos::PionLCDAs, 340
~PopulationMonteCarloSampler
eos::PopulationMonteCarloSampler, 342
~PriorSampler
eos::PriorSampler, 347
~PrivateImplementationPattern
eos::PrivateImplementationPattern, 348
~ProposalFunction
eos::MarkovChain::ProposalFunction, 349
~RandomNumberGenerator
eos::RandomNumberGenerator, 360
~Save
eos::Save, 367
~Scalar
eos::hdf5::Scalar, 368
~StandardModel
eos::StandardModel, 380
~TestObservable
eos::TestObservable, 387
~Thread
eos::Thread, 389
~ThreadPool
eos::ThreadPool, 391
~Ticket
eos::Ticket, 392
~TicketList
eos::TicketList, 394
~TryLock
eos::TryLock, 399
~Type
eos::hdf5::Type, 401
~WilsonScanModel
eos::WilsonScanModel, 423
~WrappedForwardIterator
eos::WrappedForwardIterator, 427
/home/alexshires/EOS/eos-next/eos/b-decays/b-to-pi-l-nu.hh, 429
/home/alexshires/EOS/eos-next/eos/b-decays/properties.hh, 430
/home/alexshires/EOS/eos-next/eos/constraint.hh, 431
/home/alexshires/EOS/eos-next/eos/form-factors/analytic-b-to-pi.hh, 432
/home/alexshires/EOS/eos-next/eos/form-factors/baryonic-impl.hh, 433
/home/alexshires/EOS/eos-next/eos/form-factors/baryonic.hh, 434
/home/alexshires/EOS/eos-next/eos/form-factors/form-factor-adapter.hh, 435
/home/alexshires/EOS/eos-next/eos/form-factors/form-factors-fwd.hh, 436
/home/alexshires/EOS/eos-next/eos/form-factors/form-factors.hh, 437
/home/alexshires/EOS/eos-next/eos/form-factors/mesonic-impl.hh, 438
/home/alexshires/EOS/eos-next/eos/form-factors/mesonic.hh, 439

/home/alexshires/EOS/eos-next/eos/form-factors/pi-lcdas.hh, 440
/home/alexshires/EOS/eos-next/eos/observable.hh, 441
/home/alexshires/EOS/eos-next/eos/rare-b-decays/bremsstrahlung.hh, 442
/home/alexshires/EOS/eos-next/eos/rare-b-decays/charm-loops.hh, 443
/home/alexshires/EOS/eos-next/eos/rare-b-decays/decays.hh, 444
/home/alexshires/EOS/eos-next/eos/rare-b-decays/em-contributions.hh, 445
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-dilepton.hh, 446
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-large-recoil.hh, 447
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton-low-recoil.hh, 448
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-dilepton.hh, 449
/home/alexshires/EOS/eos-next/eos/rare-b-decays/exclusive-b-to-s-gamma.hh, 450
/home/alexshires/EOS/eos-next/eos/rare-b-decays/hard-scattering.hh, 451
/home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-dilepton.hh, 452
/home/alexshires/EOS/eos-next/eos/rare-b-decays/inclusive-b-to-s-gamma.hh, 453
/home/alexshires/EOS/eos-next/eos/rare-b-decays/lambda-b-to-lambda-dilepton.hh, 454
/home/alexshires/EOS/eos-next/eos/rare-b-decays/long-distance.hh, 455
/home/alexshires/EOS/eos-next/eos/rare-b-decays/qcdf_integrals.hh, 456
/home/alexshires/EOS/eos-next/eos/utils/accumulator.hh, 457
/home/alexshires/EOS/eos-next/eos/utils/analysis-fwd.hh, 458
/home/alexshires/EOS/eos-next/eos/utils/analysis.hh, 459
/home/alexshires/EOS/eos-next/eos/utils/analysis_-TEST.hh, 460
/home/alexshires/EOS/eos-next/eos/utils/apply.hh, 461
/home/alexshires/EOS/eos-next/eos/utils/cartesian-product.hh, 462
/home/alexshires/EOS/eos-next/eos/utils/chi-squared.hh, 463
/home/alexshires/EOS/eos-next/eos/utils/ckm_-scan_model.hh, 464
/home/alexshires/EOS/eos-next/eos/utils/cluster.hh, 465
/home/alexshires/EOS/eos-next/eos/utils/complex.hh, 466
/home/alexshires/EOS/eos-next/eos/utils/concrete_-observable.hh, 467
/home/alexshires/EOS/eos-next/eos/utils/condition_variable.hh, 468
/home/alexshires/EOS/eos-next/eos/utils/derivative.hh, 469
/home/alexshires/EOS/eos-next/eos/utils/destringify.hh, 470
/home/alexshires/EOS/eos-next/eos/utils/diagnostics.hh, 471
/home/alexshires/EOS/eos-next/eos/utils/equation_solver.hh, 472
/home/alexshires/EOS/eos-next/eos/utils/exception.hh, 473
/home/alexshires/EOS/eos-next/eos/utils/hdf5-fwd.hh, 474
/home/alexshires/EOS/eos-next/eos/utils/hdf5.hh, 475
/home/alexshires/EOS/eos-next/eos/utils/hierarchical-clustering.hh, 477
/home/alexshires/EOS/eos-next/eos/utils/histogram.hh, 478
/home/alexshires/EOS/eos-next/eos/utils/instantiation_policy-impl.hh, 479
/home/alexshires/EOS/eos-next/eos/utils/instantiation_policy.hh, 480
/home/alexshires/EOS/eos-next/eos/utils/integrate-impl.hh, 481
/home/alexshires/EOS/eos-next/eos/utils/integrate.hh, 482
/home/alexshires/EOS/eos-next/eos/utils/kinematic.hh, 483
/home/alexshires/EOS/eos-next/eos/utils/lcda.hh, 484
/home/alexshires/EOS/eos-next/eos/utils/lock.hh, 485
/home/alexshires/EOS/eos-next/eos/utils/log.hh, 486
/home/alexshires/EOS/eos-next/eos/utils/log_-likelihood-fwd.hh, 487
/home/alexshires/EOS/eos-next/eos/utils/log_-likelihood.hh, 488

/home/alexshires/EOS/eos-next/eos/utils/log_prior-fwd.hh, 489
/home/alexshires/EOS/eos-next/eos/utils/log_-prior.hh, 490
/home/alexshires/EOS/eos-next/eos/utils/markov_-chain.hh, 491
/home/alexshires/EOS/eos-next/eos/utils/markov_-chain_sampler.hh, 492
/home/alexshires/EOS/eos-next/eos/utils/matrix.hh, 493
/home/alexshires/EOS/eos-next/eos/utils/memoise.hh, 495
/home/alexshires/EOS/eos-next/eos/utils/model.hh, 496
/home/alexshires/EOS/eos-next/eos/utils/mutable-fwd.hh, 497
/home/alexshires/EOS/eos-next/eos/utils/mutable.hh, 498
/home/alexshires/EOS/eos-next/eos/utils/mutex.hh, 499
/home/alexshires/EOS/eos-next/eos/utils/observable_cache.hh, 500
/home/alexshires/EOS/eos-next/eos/utils/observable_set.hh, 501
/home/alexshires/EOS/eos-next/eos/utils/one-of.hh, 502
/home/alexshires/EOS/eos-next/eos/utils/options.hh, 503
/home/alexshires/EOS/eos-next/eos/utils/parameters-fwd.hh, 504
/home/alexshires/EOS/eos-next/eos/utils/parameters.hh, 505
/home/alexshires/EOS/eos-next/eos/utils/polylog.hh, 506
/home/alexshires/EOS/eos-next/eos/utils/population_monte_carlo_-sampler.hh, 507
/home/alexshires/EOS/eos-next/eos/utils/power_-of.hh, 508
/home/alexshires/EOS/eos-next/eos/utils/prior_-sampler.hh, 509
/home/alexshires/EOS/eos-next/eos/utils/private_-implementation_pattern-impl.hh, 510
/home/alexshires/EOS/eos-next/eos/utils/private_-implementation_pattern.hh, 511
/home/alexshires/EOS/eos-next/eos/utils/proposal_functions.hh, 512
/home/alexshires/EOS/eos-next/eos/utils/qcd.hh, 514
/home/alexshires/EOS/eos-next/eos/utils/random_-number_generator.hh, 515
/home/alexshires/EOS/eos-next/eos/utils/rvalue.hh, 516
/home/alexshires/EOS/eos-next/eos/utils/save.hh, 517
/home/alexshires/EOS/eos-next/eos/utils/standard-model.hh, 518
/home/alexshires/EOS/eos-next/eos/utils/stringify.hh, 519
/home/alexshires/EOS/eos-next/eos/utils/thread.hh, 520
/home/alexshires/EOS/eos-next/eos/utils/thread_-pool.hh, 521
/home/alexshires/EOS/eos-next/eos/utils/ticket.hh, 522
/home/alexshires/EOS/eos-next/eos/utils/top-loops.hh, 523
/home/alexshires/EOS/eos-next/eos/utils/verify.hh, 524
/home/alexshires/EOS/eos-next/eos/utils/welford.hh, 525
/home/alexshires/EOS/eos-next/eos/utils/wilson-polynomial.hh, 526
/home/alexshires/EOS/eos-next/eos/utils/wilson_-coefficients.hh, 527
/home/alexshires/EOS/eos-next/eos/utils/wilson_-scan_model.hh, 528
/home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-fwd.hh, 529
/home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_iterator-impl.hh, 530
/home/alexshires/EOS/eos-next/eos/utils/wrapped_forward_-iterator.hh, 531
_alpha_s
eos::WilsonCoefficients< BToS >, 417
_compute_cholesky_and_inverse
eos::proposal_functions::Multivariate, 284
_compute_norm
eos::proposal_functions::Multivariate, 284
eos::proposal_-functions::MultivariateGaussian, 289
eos::proposal_-functions::MultivariateStudentT, 292
_copy
eos::proposal_functions::Multivariate, 284
_covariance
eos::proposal_functions::Multivariate, 285
_covariance_chol
eos::proposal_functions::Multivariate, 285
_covariance_inverse
eos::proposal_functions::Multivariate, 285
_dimension

eos::proposal_functions::Multivariate, 285
_dump_covariance
eos::proposal_functions::Multivariate, 284
_ids
eos::ParameterUser, 338
_imp
eos::PrivateImplementationPattern, 348
_index_list
eos::proposal_functions::Multivariate, 285
_parameter_descriptions
eos::LogPrior, 261
_parameters
eos::LogPrior, 261
_primed_coefficients
eos::WilsonCoefficients< BToS >, 417
_ptr
instantiation_policy-impl.hh, 479
_scalar_tensor_coefficients
eos::WilsonCoefficients< BToS >, 417
_sm_like_coefficients
eos::WilsonCoefficients< BToS >, 417
_tmp_left
eos::proposal_functions::Multivariate, 285
_tmp_right
eos::proposal_functions::Multivariate, 285
_tmp_sample_covariance_current
eos::proposal_functions::Multivariate, 285

A
eos::CharmLoops, 120
A0
eos::TopLoops, 396
A1
eos::TopLoops, 396
a2pi
eos::PionLCDAs, 340
a4pi
eos::PionLCDAs, 340
a_0
eos::BZ2004FormFactors< Process_, PToV >, 113
eos::FormFactors< PToV >, 201
eos::KMPW2010FormFactors< PToV >, 233
a_1
eos::BZ2004FormFactors< Process_, PToV >, 113
eos::FormFactors< PToV >, 201
eos::KMPW2010FormFactors< PToV >, 233
a_12
eos::BZ2004FormFactors< Process_, PToV >, 113
eos::FormFactors< PToV >, 201
eos::KMPW2010FormFactors< PToV >, 233
a_2
eos::BZ2004FormFactors< Process_, PToV >, 113
eos::FormFactors< PToV >, 201
eos::KMPW2010FormFactors< PToV >, 233
a_fb_zero_crossing
eos::BToKstarDilepton< LargeRecoil >, 97
a_l
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
a_long
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
a_long_left
eos::btovll::Amplitudes, 56
a_long_right
eos::btovll::Amplitudes, 56
a_par
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
a_par_left
eos::btovll::Amplitudes, 56
a_par_right
eos::btovll::Amplitudes, 56
a_perp
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
a_perp_left
eos::btovll::Amplitudes, 56
a_perp_right
eos::btovll::Amplitudes, 56
a_timelike
eos::btovll::Amplitudes, 56
AbsoluteTestObservable
eos::AbsoluteTestObservable, 51
accept
eos::OneOfValue, 312
eos::OneOfValueBase, 313
accept_returning
eos::OneOf, 311
access
eos::proposal_functions::MultivariateAccess, 287
Accumulator
eos::Accumulator, 53
accumulator.hh
EOS_GUARD_SRC_UTILS_-ACCUMULATOR_HH, 457
acquire_then_signal
eos::ConditionVariable, 137
adapt
eos::MarkovChain::ProposalFunction, 349
eos::proposal_-functions::BlockDecomposition, 80
eos::proposal_functions::GlobalLocal, 205

eos::proposal_functions::Multivariate, 284
adapt_iterations
eos::MarkovChainSampler::Config, 141
adaptations
eos::proposal_functions::Multivariate, 285
add
eos::Analysis, 58
eos::Cluster, 127
eos::Diagnostics, 177
eos::EquationSolver, 181
eos::HierarchicalClustering, 211
eos::LogLikelihood, 245
eos::ObservableCache, 303
eos::ObservableSet, 308
eos::PriorSampler, 347
eos::proposal_functions::AdjacencyMatrix, 54
eos::proposal_-functions::BlockDecomposition, 80
eos::Welford, 414
AdjacencyMatrix
eos::proposal_functions::AdjacencyMatrix, 54
adjust_sample_size
eos::PopulationMonteCarloSampler::Config, 148
algorithm
eos::Analysis::OptimizationOptions, 321
alpha_s
eos::ModelComponent< components::QCD >, 280
eos::QCD, 354
eos::SMComponent< components::QCD >, 377
Amoroso
eos::LogLikelihoodBlock, 249, 250
AmorosoLimit
eos::LogLikelihoodBlock, 250
AmorosoMode
eos::LogLikelihoodBlock, 251
Analysis
eos::Analysis, 58
analysis-fwd.hh
EOS_GUARD_SRC_UTILS_ANALYSIS_-FWD_HH, 458
analysis.hh
EOS_GUARD_SRC_UTILS_ANALYSIS_-HH, 459
analysis_TEST.hh
EOS_GUARD_SRC_UTILS_ANALYSIS_-TEST_HH, 460
AnalysisPtr
eos, 27
analytic-b-to-pi.hh
EOS_GUARD_EOS_FORM_FACTORS_-ANALYTIC_B_TO_PI_HH, 432
AnalyticFormFactorBToPiDKMMO2008
eos::AnalyticFormFactorBToPiDKMMO2008, 62
angular_coefficients_array
eos::btovll, 34
AnomalousMassDimension
eos::QCD, 353
apply
eos, 28
apply.hh
EOS_GUARD_SRC_UTILS_APPLY_HH, 461
approximation
eos::RValue, 366
Array
eos::hdf5::Array, 65
array_to_angular_coefficients
eos::btovll, 34
as_string
eos::Kinematics, 228
eos::LogLikelihoodBlock, 251
eos::LogPrior, 257
eos::Options, 324
attribute_id
eos::hdf5::AttributeHandle, 68

B
eos::CharmLoops, 120
b-to-pi-l-nu.hh
EOS_GUARD_EOS_B_DECAYS_-EXCLUSIVE_B_TO_U_L_NU_HH, 429
B0
eos::CharmLoops, 120
eos::TopLoops, 396
B1
eos::TopLoops, 396
b_l
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
baryonic-impl.hh
EOS_GUARD_EOS_FORM_FACTORS_-BARYONIC_IMPL_HH, 433
baryonic.hh
EOS_GUARD_EOS_FORM_FACTORS_-BARYONIC_HH, 434
BaseImpl
eos::WrappedForwardIterator::BaseImpl, 71
BCL2008FormFactors
eos::BCL2008FormFactors, 73
begin
eos::CartesianProduct, 117
eos::Cluster, 127
eos::Diagnostics, 177

eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
eos::LogLikelihood, 245
eos::LogPrior, 257
eos::ObservableSet, 308
eos::Parameters, 335
eos::ParameterUser, 337
begin_blocks
eos::Constraint, 163
begin_clusters
eos::HierarchicalClustering, 211
begin_components
eos::HierarchicalClustering, 212
begin_indices
eos::Cluster, 127
begin_map
eos::HierarchicalClustering, 212
begin_observables
eos::Constraint, 163
beta_function_nf_3
eos::QCD, 355
beta_function_nf_4
eos::QCD, 355
beta_function_nf_5
eos::QCD, 355
beta_function_nf_6
eos::QCD, 355
BetaFunction
eos::QCD, 353
BFW2010FormFactors
eos::BFW2010FormFactors< PToP >, 74
Bin
eos::Histogram< 1 >::Bin, 77
eos::Histogram< 2 >::Bin, 75
block_decomposition
eos::PopulationMonteCarloSampler::Config, 148
block_proposal_parameters
eos::MarkovChainSampler::Config, 141
BlockDecomposition
eos::proposal_functions::BlockDecomposition, 80
BlockIterator
eos::Constraint, 163
BMesonProperties
eos::BMesonProperties, 82
bootstrap_p_value
eos::LogLikelihood, 245
branching_ratio
eos::BToKstarGamma, 104
branching_ratio_cp_averaged
eos::BToKstarGamma, 104
branching_ratio_time_zero
eos::BToDilepton, 86
branching_ratio_untagged_integrated
eos::BToDilepton, 86
bremsstrahlung.hh
EOS_GUARD_SRC_RARE_B_DECAYS_BREMSSTRAHLUNG_HH, 442
broadcast
eos::ConditionVariable, 137
BToDilepton
eos::BToDilepton, 86
BToKDilepton
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
BToKstarDilepton
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
BToKstarGamma
eos::BToKstarGamma, 104
BTOPI_FF_GUARD_EOS_UTILS_DERIVATIVE_HH
derivative.hh, 469
BToPiLeptonNeutrino
eos::BToPiLeptonNeutrino, 107
BToXsDilepton
eos::BToXsDilepton< HLMW2005 >, 109
BToXsGamma
eos::BToXsGamma< Minimal >, 110
eos::BToXsGamma< NLO >, 111
build_global_local
eos::MarkovChainSampler, 271
BZ2004FormFactors
eos::BZ2004FormFactors< Process_, PToP >, 112
eos::BZ2004FormFactors< Process_, PToV >, 113
BZ2004FormFactorsSplit
eos::BZ2004FormFactorsSplit, 115

C
eos::CharmLoops, 120
C0
eos::CharmLoops, 120
eos::TopLoops, 397
C1
eos::TopLoops, 397
c1
eos::WilsonCoefficients< BToS >, 417
c10
eos::WilsonCoefficients< BToS >, 417
c10prime
eos::WilsonCoefficients< BToS >, 417
c2
eos::WilsonCoefficients< BToS >, 417
c2b
eos::WilsonCoefficients< BToS >, 417

c3
eos::WilsonCoefficients< BToS >, 417
c4
eos::WilsonCoefficients< BToS >, 417
c5
eos::WilsonCoefficients< BToS >, 417
c6
eos::WilsonCoefficients< BToS >, 417
c7
eos::WilsonCoefficients< BToS >, 417
c7eff
eos::ShortDistanceLowRecoil, 372
c7prime
eos::WilsonCoefficients< BToS >, 417
c8
eos::WilsonCoefficients< BToS >, 417
c8prime
eos::WilsonCoefficients< BToS >, 417
c9
eos::WilsonCoefficients< BToS >, 417
c9eff
eos::ShortDistanceLowRecoil, 372
c9prime
eos::WilsonCoefficients< BToS >, 417
c_kstar_gamma
eos::BToKstarGamma, 104
c_l
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
calculate_weights
eos::PopulationMonteCarloSampler, 342
cartesian-product.hh
EOS_GUARD_SRC_UTILS_CARTESIAN_-PRODUCT_HH, 462
CartesianProduct
eos::CartesianProduct, 116
casimir_a
eos::QCD, 356
casimir_f
eos::QCD, 356
cbegin
eos::Histogram< 2 >, 217
cend
eos::Histogram< 2 >, 217
central
eos::Parameter, 330
charm-loops.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-CHARM_LOOPS_HH, 443
chi-squared.hh
EOS_GUARD_SRC_UTILS_CHI_-SQUARED_HH, 463
chunk_size
eos::MarkovChainSampler::Config, 141
eos::PopulationMonteCarloSampler::Config, 148
eos::PopulationMonteCarloSampler::Status, 385
chunks
eos::MarkovChainSampler::Config, 141
eos::PopulationMonteCarloSampler::Config, 148
ckm_cb
eos::CKMScanComponent, 123
eos::ModelComponent< components::CKM >, 277
eos::SMComponent< components::CKM >, 374
ckm_cd
eos::CKMScanComponent, 123
eos::ModelComponent< components::CKM >, 277
eos::SMComponent< components::CKM >, 374
ckm_cs
eos::CKMScanComponent, 123
eos::ModelComponent< components::CKM >, 277
eos::SMComponent< components::CKM >, 374
ckm_scan_model.hh
EOS_GUARD_SRC_UTILS_CKM_SCAN_-MODEL_HH, 464
ckm_tb
eos::CKMScanComponent, 123
eos::ModelComponent< components::CKM >, 277
eos::SMComponent< components::CKM >, 374
ckm_td
eos::CKMScanComponent, 123
eos::ModelComponent< components::CKM >, 277
eos::SMComponent< components::CKM >, 374
ckm_ts
eos::CKMScanComponent, 123
eos::ModelComponent< components::CKM >, 278
eos::SMComponent< components::CKM >, 375
ckm_ub
eos::CKMScanComponent, 124
eos::ModelComponent< components::CKM >, 278
eos::SMComponent< components::CKM >, 375
ckm_ud

eos::CKMScanComponent, 124
eos::ModelComponent< components::CKM >, 278
eos::SMComponent< components::CKM >, 375
ckm_us
eos::CKMScanComponent, 124
eos::ModelComponent< components::CKM >, 278
eos::SMComponent< components::CKM >, 375
CKMScanComponent
eos::CKMScanComponent, 123
CKMScanModel
eos::CKMScanModel, 125
clear
eos::MarkovChain, 266
eos::MemoisationControl, 273
eos::Memoiser, 274
clone
eos::AbsoluteTestObservable, 51
eos::Analysis, 58
eos::ConcreteObservable, 134
eos::FormFactorAdapter, 189
eos::FormFactorRatioAdapter, 195
eos::Kinematics, 228
eos::LogLikelihood, 246
eos::LogLikelihoodBlock, 251
eos::LogPrior, 257
eos::MarkovChain::ProposalFunction, 350
eos::Mutable, 295
eos::Observable, 300
eos::ObservableCache, 303
eos::Parameter, 330
eos::Parameters, 335
eos::proposal_functions::BlockDecomposition, 80
eos::proposal_functions::GlobalLocal, 205
eos::proposal_functions::MultivariateGaussian, 289
eos::proposal_functions::MultivariateStudentT, 292
eos::TestObservable, 387
eos::WrappedForwardIterator::Base, 69
eos::WrappedForwardIterator::BaseImpl, 71
Cluster
eos::Cluster, 126
cluster.hh
EOS_GUARD_EOS_UTILS_CLUSTER_HH, 465
clustering_maximum_r_value
eos::proposal_functions::GlobalLocal::Config, 157
clustering_strict_r_value
eos::proposal_functions::GlobalLocal::Config, 157
ClusterIterator
eos::HierarchicalClustering, 211
completed
eos::Thread, 390
complex.hh
WILSON_FITTER_GUARD_SRC_UTILS_COMPLEX_HH, 466
Component
eos::HierarchicalClustering::Component, 129
component
eos::MarkovChain::HyperParameter, 222
component_means
eos::PopulationMonteCarloSampler::Config, 148
component_offset
eos::PopulationMonteCarloSampler::Config, 149
component_probabilities
eos::proposal_functions::GlobalLocal, 205
component_record
eos::PopulationMonteCarloSampler::Output, 326
component_type
eos::PopulationMonteCarloSampler::Output, 326
eos::proposal_functions::GlobalLocal, 206
component_variances
eos::PopulationMonteCarloSampler::Config, 149
component_weights
eos::PopulationMonteCarloSampler::Config, 149
ComponentIterator
eos::HierarchicalClustering, 211
components_per_cluster
eos::PopulationMonteCarloSampler::Config, 149
ComponentType
eos::PopulationMonteCarloSampler::Output, 326
eos::proposal_functions::GlobalLocal, 204
Composite
eos::hdf5::Composite, 130
concrete_observable.hh
EOS_GUARD_SRC_UTILS_CONCRETE_OBSERVABLE_HH, 467
ConcreteObservable
eos::ConcreteObservable, 134
ConcreteObservableFactory
eos::ConcreteObservableFactory, 136
condition_variable.hh

EOS_GUARD_SRC_UTILS_CONDITION_-VARIABLE_HH, 468
ConditionVariable
eos::ConditionVariable, 137
config
eos::MarkovChainSampler, 271
eos::PopulationMonteCarloSampler, 342
eos::proposal_functions::GlobalLocal, 206
ConstIterator
eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
eos::ParameterUser, 337
Constraint
eos::Constraint, 163
constraint.hh
EOS_GUARD_SRC_CONSTRAINT_HH, 431
ConstraintIterator
eos::LogLikelihood, 245
converged
eos::MarkovChainSampler::PreRunInfo, 344
eos::PopulationMonteCarloSampler::Status, 385
convergence_eff_sample_size
eos::PopulationMonteCarloSampler::Config, 149
convergence_perplexity
eos::PopulationMonteCarloSampler::Config, 149
cooling_power
eos::proposal_functions::GlobalLocal::Config, 157
eos::proposal_functions::Multivariate, 285
copy
eos::hdf5::File, 186
copy_from_hdf5
eos::hdf5::Array, 65
eos::hdf5::Composite, 130
eos::hdf5::CompositeCopier, 132
eos::hdf5::CompositeCopier< 0, TypeTuple_-, Tuple_ >, 133
eos::hdf5::Scalar, 368
eos::hdf5::Type, 401
copy_to_hdf5
eos::hdf5::Array, 65
eos::hdf5::Composite, 130
eos::hdf5::CompositeCopier, 132
eos::hdf5::CompositeCopier< 0, TypeTuple_-, Tuple_ >, 133
eos::hdf5::Scalar, 368
eos::hdf5::Type, 401
covariance
eos::HierarchicalClustering::Component, 129
eos::proposal_functions::Multivariate, 284
covariance_scale
eos::proposal_functions::Multivariate, 285
covariance_scale_max
eos::proposal_functions::Multivariate, 286
covariance_scale_min
eos::proposal_functions::Multivariate, 286
covariance_scale_update_factor
eos::proposal_functions::Multivariate, 286
covariance_type
eos::proposal_functions::Multivariate, 284
CovarianceType
eos::proposal_functions::Multivariate, 283
cP
eos::WilsonCoefficients< BToS >, 417
cp_asymmetry
eos::BToKstarGamma, 105
cPprime
eos::WilsonCoefficients< BToS >, 417
cq3
eos::WilsonCoefficients< BToS >, 417
cq4
eos::WilsonCoefficients< BToS >, 417
cq5
eos::WilsonCoefficients< BToS >, 417
cq6
eos::WilsonCoefficients< BToS >, 417
Create
eos::hdf5::File, 186
create
eos::FormFactorFactory< OneHalfPlus-ToOneHalfPlus >, 192
eos::FormFactorFactory< PToP >, 193
eos::FormFactorFactory< PToV >, 194
create_attribute
eos::hdf5::DataSet, 165
eos::hdf5::DataSetHandle, 168
create_data_set
eos::hdf5::File, 186
create_group
eos::hdf5::File, 186
create_or_open_attribute
eos::hdf5::DataSet, 165
create_or_open_data_set
eos::hdf5::File, 186
crop_highest_weights
eos::PopulationMonteCarloSampler::Config, 149
cS
eos::WilsonCoefficients< BToS >, 417
cSprime
eos::WilsonCoefficients< BToS >, 417
cT
eos::WilsonCoefficients< BToS >, 417
cT5

eos::WilsonCoefficients< BToS >, 417
current_state
eos::MarkovChain, 266

D0
eos::TopLoops, 397
D1
eos::TopLoops, 397
DataSet
eos::hdf5::Attribute, 67
DataSetHandle
eos::hdf5::AttributeHandle, 68
decay_width
eos::btovll, 34
decays.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-DECAYS_HH, 444
declare
eos::Kinematics, 228
eos::Parameters, 335
Default
eos::EquationSolver::Config, 145
eos::HierarchicalClustering::Config, 160
eos::MarkovChainSampler::Config, 140
eos::PopulationMonteCarloSampler::Config, 148
eos::PriorSampler::Config, 154
eos::proposal_functions::GlobalLocal::Config, 157
Defaults
eos::Analysis::OptimizationOptions, 321
eos::Parameters, 335
degrees_of_freedom
eos::PopulationMonteCarloSampler::Config, 149
DeleteOnDestruction
eos::InstantiationPolicy< T_, Singleton >, 225
Deltai_23
eos::Bremsstrahlung, 84
Deltai_27
eos::Bremsstrahlung, 84
deltapipi
eos::PionLCDAs, 340
derivative
eos, 28
derivative< 1u, deriv::TwoSided >
eos, 28
derivative< 2u, deriv::TwoSided >
eos, 28
derivative.hh
BTOPI_FF_GUARD_EOS_UTILS_-DERIVATIVE_HH, 469
description
eos::Diagnostics::Entry, 180
description_record
eos::Analysis::Output, 328
description_type
eos::Analysis::Output, 328
DescriptionType
eos::Analysis::Output, 328
destringify
eos, 28
destringify.hh
EOS_GUARD_SRC_UTILS_-DESTRINGIFY_HH, 470
determinant
eos::HierarchicalClustering::Component, 129
Diagnostics
eos::Diagnostics, 177
diagnostics
eos::AnalyticFormFactorBToPiDKMMO2008, 62
eos::BToXsDilepton< HLMW2005 >, 109
eos::BToXsGamma< NLO >, 111
eos::PionLCDAs, 340
diagnostics.hh
EOS_GUARD_EOS_UTILS_-DIAGNOSTICS_HH, 471
difference_type
eos::WrappedForwardIterator, 427
differential_a_fb_combined
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
differential_a_fb_hadronic
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
differential_a_fb_leptonic
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
differential_branching_ratio
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
eos::BToPiLeptonNeutrino, 107
eos::BToXsDilepton< HLMW2005 >, 109
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
differential_cp_asymmetry_1
eos::BToKstarDilepton< LowRecoil >, 102

differential_cp_asymmetry_2
eos::BToKstarDilepton< LowRecoil >, 102
differential_cp_asymmetry_3
eos::BToKstarDilepton< LowRecoil >, 102
differential_cp_asymmetry_mix
eos::BToKstarDilepton< LowRecoil >, 102
differential_decay_width
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_flat_term
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
differential_forward_backward_asymmetry
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_fzero
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
differential_h_1
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_h_2
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_h_3
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_h_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_h_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_isospin_asymmetry
eos::BToKstarDilepton< LargeRecoil >, 97
differential_j_1c
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_1s
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_2c
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_2s
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_3
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_3_normalized
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_3_normalized_cp_averaged
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_6c
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_6s
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_7
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_8
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_9
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_9_normalized
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_j_9_normalized_cp_averaged
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_longitudinal_polarisation
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_p_prime_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_p_prime_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_p_prime_6
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_ratio_muons_electrons
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
differential_transversal_polarisation
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_transverse_asymmetry_2
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102

differential_transverse_asymmetry_3
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_transverse_asymmetry_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_transverse_asymmetry_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_transverse_asymmetry_im
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_transverse_asymmetry_re
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
differential_zeta
eos::BToPiLeptonNeutrino, 107
dilepton_bottom_case
eos::QCDFIntegrals, 357
dilepton_charm_case
eos::QCDFIntegrals, 357
dilepton_massless_case
eos::QCDFIntegrals, 358
dilog
eos, 28
dimension
eos::proposal_functions::Multivariate, 284
Discrete
eos::LogPrior, 257
discrete
eos::ParameterDescription, 332
divide
eos, 28
dof
eos::proposal_-functions::MultivariateStudentT, 293
dot
eos, 28
draw_samples
eos::PopulationMonteCarloSampler, 342
dump_description
eos::MarkovChain, 266
dump_descriptions
eos::Analysis, 58
dump_history
eos::MarkovChain, 266
dump_proposal
eos::MarkovChain, 266
dump_state
eos::MarkovChain::ProposalFunction, 350
eos::proposal_-functions::BlockDecomposition, 80
eos::proposal_functions::GlobalLocal, 206
eos::proposal_-functions::MultivariateGaussian, 289
eos::proposal_-functions::MultivariateStudentT, 292

E0
eos::TopLoops, 397
E1
eos::TopLoops, 397
eff_sample_size
eos::PopulationMonteCarloSampler::Status, 385
em-contributions.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-EM_CONTRIBUTIONS_HH, 445
end
eos::CartesianProduct, 117
eos::Cluster, 127
eos::Diagnostics, 177
eos::hdf5::DataSet, 165
eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
eos::LogLikelihood, 246
eos::LogPrior, 258
eos::ObservableSet, 308
eos::Parameters, 335
eos::ParameterUser, 337
end_blocks
eos::Constraint, 163
end_clusters
eos::HierarchicalClustering, 212
end_components
eos::HierarchicalClustering, 212
end_indices
eos::Cluster, 127
end_map
eos::HierarchicalClustering, 212
end_observables
eos::Constraint, 163
enqueue
eos::ThreadPool, 391
entries
eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
eos, 21
AnalysisPtr, 27
apply, 28
derivative, 28
derivative< 1u, deriv::TwoSided >, 28
derivative< 2u, deriv::TwoSided >, 28
destringify, 28
dilog, 28
divide, 28
dot, 28

estimate_cumulative_distribution, 28
evolve, 28
Helicity, 27
HistoryPtr, 27
integrate, 29
lambda, 29
left_handed, 27
ll_debug, 27
ll_error, 27
ll_informational, 27
ll_last, 27
ll_silent, 27
ll_warning, 27
LogLevel, 27
LogLikelihoodBlockPtr, 27
LogPriorPtr, 27
make_analysis, 30
make_concrete_observable_factory, 30
make_polynomial, 30
make_polynomial_ht_like_ratio, 30
make_polynomial_observable, 30
make_polynomial_ratio, 30
memoise, 30
mult, 31
MutablePtr, 27
number_of_memoisations, 31
ObservablePtr, 27
operator<<, 32
operator>>, 32
operator∗, 31
operator^, 32
operator+, 31, 32
operator-, 32
power_of, 32
ProposalFunctionPtr, 27
right_handed, 27
stringify, 32, 33
stringify_container, 33
trilog, 33
WilsonPolynomial, 27
eos::AbsoluteTestObservable, 51
∼AbsoluteTestObservable, 51
AbsoluteTestObservable, 51
clone, 51
evaluate, 51
eos::Accumulator, 53
∼Accumulator, 53
Accumulator, 53
operator+=, 53
value, 53
eos::Analysis, 57
∼Analysis, 58
add, 58
Analysis, 58
clone, 58
dump_descriptions, 58
goodness_of_fit, 58
Implementation< Analysis >, 61
log_likelihood, 59
log_posterior, 59
log_prior, 59
nuisance, 59
optimize, 59
optimize_minuit, 60
parameter_descriptions, 60
parameters, 60
read_descriptions, 60
restrict, 60
eos::Analysis::OptimizationOptions, 321
algorithm, 321
Defaults, 321
fix_flat_nuisance, 321
initial_step_size, 321
maximum_iterations, 321
mcmc_pre_run, 321
splitting_tolerance, 322
strategy_level, 322
tolerance, 322
eos::Analysis::Output, 328
description_record, 328
description_type, 328
DescriptionType, 328
eos::AnalyticFormFactorBToPiDKMMO2008, 62
∼AnalyticFormFactorBToPiDKMMO2008, 62
AnalyticFormFactorBToPiDKMMO2008, 62
diagnostics, 62
f_0, 62
F_lo_tw2, 62
F_lo_tw3, 63
F_lo_tw4, 63
F_nlo_tw2, 63
F_nlo_tw3, 63
f_p, 63
f_t, 63
make, 63
MB_lcsr, 63
MB_svz, 63
eos::BCL2008FormFactors, 73
BCL2008FormFactors, 73
f_0, 73
f_p, 73
f_t, 73
make, 73
eos::BFW2010FormFactors< PToP >, 74
BFW2010FormFactors, 74
f_0, 74
f_p, 74

f_t, 74
make, 74
eos::BMesonProperties, 82
~BMesonProperties, 82
BMesonProperties, 82
mass_splitting_j1_j0, 82
eos::Bremsstrahlung, 83
Deltai_23, 84
Deltai_27, 84
G_0, 84
G_m1, 84
itau_22, 84
itau_27, 84
itau_28, 84
itau_29, 84
tau_22, 84
tau_27, 84
tau_28, 84
tau_29, 84
tau_78, 84
tau_88, 84
tau_89, 84
eos::BsToPhi, 85
mB, 85
mV, 85
eos::BToDilepton, 86
~BToDilepton, 86
branching_ratio_time_zero, 86
branching_ratio_untagged_integrated, 86
BToDilepton, 86
eos::BToK, 87
label, 87
m2_Br0m, 87
m2_Br1m, 87
m_B, 87
m_P, 87
tau_m, 87
tau_p, 87
Transition, 87
eos::BToKDilepton< LargeRecoil >, 88
~BToKDilepton, 89
a_l, 89
b_l, 89
BToKDilepton, 89
c_l, 89
differential_branching_ratio, 89
differential_flat_term, 89
differential_forward_backward_asymmetry, 89
differential_ratio_muons_electrons, 89
F_A, 89
F_P, 89
F_S, 89
F_T, 89
F_T5, 89
F_V, 89
integrated_branching_ratio, 89
integrated_branching_ratio_cp_averaged, 89
integrated_cp_asymmetry, 89
integrated_decay_width, 89
integrated_flat_term, 89
integrated_flat_term_cp_averaged, 89
integrated_forward_backward_asymmetry, 89
integrated_forward_backward_asymmetry_-cp_averaged, 89
integrated_ratio_muons_electrons, 89
two_differential_decay_width, 89
eos::BToKDilepton< LowRecoil >, 91
~BToKDilepton, 92
a_l, 92
b_l, 92
BToKDilepton, 92
c_l, 92
differential_branching_ratio, 92
differential_flat_term, 92
differential_forward_backward_asymmetry, 92
differential_ratio_muons_electrons, 92
imag_c7eff, 92
imag_c9eff, 92
integrated_branching_ratio, 92
integrated_branching_ratio_cp_averaged, 92
integrated_cp_asymmetry, 92
integrated_decay_width, 92
integrated_flat_term, 92
integrated_flat_term_cp_averaged, 92
integrated_forward_backward_asymmetry, 92
integrated_forward_backward_asymmetry_-cp_averaged, 92
integrated_ratio_muons_electrons, 92
real_c7eff, 92
real_c9eff, 92
two_differential_decay_width, 92
eos::BToKstar, 94
mB, 94
mV, 94
eos::BToKstarDilepton< LargeRecoil >, 95
~BToKstarDilepton, 97
a_fb_zero_crossing, 97
a_long, 97
a_par, 97
a_perp, 97
BToKstarDilepton, 97
differential_branching_ratio, 97
differential_decay_width, 97
differential_forward_backward_asymmetry, 97
differential_h_1, 97

differential_h_2, 97
differential_h_3, 97
differential_h_4, 97
differential_h_5, 97
differential_isospin_asymmetry, 97
differential_j_1c, 97
differential_j_1s, 97
differential_j_2c, 97
differential_j_2s, 97
differential_j_3, 97
differential_j_3_normalized, 97
differential_j_3_normalized_cp_averaged, 97
differential_j_4, 97
differential_j_5, 97
differential_j_6c, 97
differential_j_6s, 97
differential_j_7, 97
differential_j_8, 97
differential_j_9, 97
differential_j_9_normalized, 97
differential_j_9_normalized_cp_averaged, 97
differential_longitudinal_polarisation, 97
differential_p_prime_4, 97
differential_p_prime_5, 97
differential_p_prime_6, 97
differential_transversal_polarisation, 97
differential_transverse_asymmetry_2, 97
differential_transverse_asymmetry_3, 97
differential_transverse_asymmetry_4, 97
differential_transverse_asymmetry_5, 97
differential_transverse_asymmetry_im, 97
differential_transverse_asymmetry_re, 97
four_differential_decay_width, 97
integrated_a_9, 97
integrated_branching_ratio, 97
integrated_branching_ratio_cp_averaged, 97
integrated_cp_asymmetry, 97
integrated_decay_width, 97
integrated_forward_backward_asymmetry, 97
integrated_forward_backward_asymmetry_-cp_averaged, 97
integrated_h_1, 97
integrated_h_2, 97
integrated_h_3, 97
integrated_h_4, 97
integrated_h_5, 97
integrated_isospin_asymmetry, 97
integrated_j_1c, 97
integrated_j_1s, 97
integrated_j_2c, 97
integrated_j_2s, 97
integrated_j_3, 97
integrated_j_3_normalized, 97
integrated_j_3_normalized_cp_averaged, 97
integrated_j_4, 97
integrated_j_5, 97
integrated_j_6c, 97
integrated_j_6s, 97
integrated_j_7, 97
integrated_j_8, 97
integrated_j_9, 97
integrated_j_9_normalized, 97
integrated_j_9_normalized_cp_averaged, 97
integrated_longitudinal_polarisation, 97
integrated_longitudinal_polarisation_cp_-averaged, 97
integrated_p_prime_4, 97
integrated_p_prime_5, 97
integrated_p_prime_6, 97
integrated_transversal_polarisation, 97
integrated_transversal_polarisation_cp_-averaged, 97
integrated_transverse_asymmetry_2, 97
integrated_transverse_asymmetry_2_cp_-averaged, 97
integrated_transverse_asymmetry_3, 97
integrated_transverse_asymmetry_4, 97
integrated_transverse_asymmetry_5, 97
integrated_transverse_asymmetry_im, 97
integrated_transverse_asymmetry_re, 97
xi_para, 97
xi_perp, 97
eos::BToKstarDilepton< LowRecoil >, 99
~BToKstarDilepton, 102
a_long, 102
a_par, 102
a_perp, 102
BToKstarDilepton, 102
differential_branching_ratio, 102
differential_cp_asymmetry_1, 102
differential_cp_asymmetry_2, 102
differential_cp_asymmetry_3, 102
differential_cp_asymmetry_mix, 102
differential_decay_width, 102
differential_forward_backward_asymmetry, 102
differential_h_1, 102
differential_h_2, 102
differential_h_3, 102
differential_h_4, 102
differential_h_5, 102
differential_j_1c, 102
differential_j_1s, 102
differential_j_2c, 102
differential_j_2s, 102
differential_j_3, 102
differential_j_3_normalized, 102
differential_j_3_normalized_cp_averaged, 102

differential_j_4, 102
differential_j_5, 102
differential_j_6c, 102
differential_j_6s, 102
differential_j_7, 102
differential_j_8, 102
differential_j_9, 102
differential_j_9_normalized, 102
differential_j_9_normalized_cp_averaged, 102
differential_longitudinal_polarisation, 102
differential_p_prime_4, 102
differential_p_prime_5, 102
differential_p_prime_6, 102
differential_transversal_polarisation, 102
differential_transverse_asymmetry_2, 102
differential_transverse_asymmetry_3, 102
differential_transverse_asymmetry_4, 102
differential_transverse_asymmetry_5, 102
differential_transverse_asymmetry_im, 102
differential_transverse_asymmetry_re, 102
four_differential_decay_width, 102
imag_c7eff, 102
imag_c9eff, 102
imag_y, 102
integrated_a_9, 102
integrated_branching_ratio, 102
integrated_branching_ratio_cp_averaged, 102
integrated_cp_asymmetry, 102
integrated_cp_asymmetry_1, 102
integrated_cp_asymmetry_2, 102
integrated_cp_asymmetry_3, 102
integrated_cp_summed_decay_width, 102
integrated_decay_width, 102
integrated_forward_backward_asymmetry, 102
integrated_forward_backward_asymmetry_cp_averaged, 102
integrated_forward_backward_asymmetry_naive, 102
integrated_h_1, 102
integrated_h_1_naive, 102
integrated_h_2, 102
integrated_h_2_naive, 102
integrated_h_3, 102
integrated_h_3_naive, 102
integrated_h_4, 102
integrated_h_5, 102
integrated_j_1c, 102
integrated_j_1s, 102
integrated_j_2c, 102
integrated_j_2s, 102
integrated_j_3, 102
integrated_j_3_normalized, 102
integrated_j_3_normalized_cp_averaged, 102
integrated_j_4, 102
integrated_j_5, 102
integrated_j_6c, 102
integrated_j_6s, 102
integrated_j_7, 102
integrated_j_8, 102
integrated_j_9, 102
integrated_j_9_normalized, 102
integrated_j_9_normalized_cp_averaged, 102
integrated_longitudinal_polarisation, 102
integrated_longitudinal_polarisation_cp_averaged, 102
integrated_longitudinal_polarisation_naive, 102
integrated_p_prime_4, 102
integrated_p_prime_5, 102
integrated_p_prime_6, 102
integrated_transversal_polarisation, 102
integrated_transversal_polarisation_cp_averaged, 102
integrated_transverse_asymmetry_2, 102
integrated_transverse_asymmetry_2_cp_averaged, 102
integrated_transverse_asymmetry_2_naive, 102
integrated_transverse_asymmetry_3, 102
integrated_transverse_asymmetry_3_naive, 102
integrated_transverse_asymmetry_4, 102
integrated_transverse_asymmetry_4_naive, 102
integrated_transverse_asymmetry_5, 102
integrated_transverse_asymmetry_im, 102
integrated_transverse_asymmetry_re, 102
integrated_unnormalized_cp_asymmetry_1, 102
integrated_unnormalized_forward_backward_asymmetry, 102
real_c7eff, 102
real_c9eff, 102
real_y, 102
rho_1, 102
rho_2, 102
eos::BToKstarGamma, 104
~BToKstarGamma, 104
branching_ratio, 104
branching_ratio_cp_averaged, 104
BToKstarGamma, 104
c_kstar_gamma, 104
cp_asymmetry, 105
isospin_asymmetry, 105
s_kstar_gamma, 105
eos::BToPi, 106
label, 106

m2_Br0m, 106
m2_Br1m, 106
m_B, 106
m_P, 106
tau_m, 106
tau_p, 106
Transition, 106
eos::BToPiLeptonNeutrino, 107
~BToPiLeptonNeutrino, 107
BToPiLeptonNeutrino, 107
differential_branching_ratio, 107
differential_zeta, 107
integrated_branching_ratio, 107
integrated_zeta, 107
eos::BToS, 108
eos::btovll, 34
angular_coefficients_array, 34
array_to_angular_coefficients, 34
decay_width, 34
eos::btovll::Amplitudes, 56
a_long_left, 56
a_long_right, 56
a_par_left, 56
a_par_right, 56
a_perp_left, 56
a_perp_right, 56
a_timelike, 56
eos::btovll::AngularCoefficients, 64
j1c, 64
j1s, 64
j2c, 64
j2s, 64
j3, 64
j4, 64
j5, 64
j6c, 64
j6s, 64
j7, 64
j8, 64
j9, 64
eos::BToXsDilepton< HLMW2005 >, 109
~BToXsDilepton, 109
BToXsDilepton, 109
diagnostics, 109
differential_branching_ratio, 109
integrated_branching_ratio, 109
eos::BToXsGamma< Minimal >, 110
~BToXsGamma, 110
BToXsGamma, 110
integrated_branching_ratio, 110
eos::BToXsGamma< NLO >, 111
~BToXsGamma, 111
BToXsGamma, 111
diagnostics, 111
integrated_branching_ratio, 111
photon_energy_moment_1, 111
photon_energy_moment_2, 111
eos::BZ2004FormFactors< Process_, PToP >, 112
BZ2004FormFactors, 112
f_0, 112
f_p, 112
f_t, 112
make, 112
eos::BZ2004FormFactors< Process_, PToV >, 113
a_0, 113
a_1, 113
a_12, 113
a_2, 113
BZ2004FormFactors, 113
make, 113
v, 114
eos::BZ2004FormFactorsSplit, 115
BZ2004FormFactorsSplit, 115
f_0, 115
f_p, 115
f_t, 115
make, 115
eos::CartesianProduct, 116
begin, 117
CartesianProduct, 116
end, 117
Iterator, 116
over, 117
size, 117
eos::CharmLoops, 118
A, 120
B, 120
B0, 120
C, 120
C0, 120
F17_massive, 120
F17_massless, 120
F19_massive, 120
F19_massless, 120
F27_massive, 120
F27_massless, 120
F29_massive, 120
F29_massless, 120
F87_massless, 120
F89_massless, 120
h, 120
eos::ChiSquared, 122
Function, 122
make, 122
with_combined_uncertainties, 122
with_theory_offset, 122
eos::CKMScanComponent, 123
ckm_cb, 123

ckm_cd, 123
ckm_cs, 123
ckm_tb, 123
ckm_td, 123
ckm_ts, 123
ckm_ub, 124
ckm_ud, 124
ckm_us, 124
CKMScanComponent, 123
eos::CKMScanModel, 125
~CKMScanModel, 125
CKMScanModel, 125
make, 125
eos::Cluster, 126
~Cluster, 127
add, 127
begin, 127
begin_indices, 127
Cluster, 126
end, 127
end_indices, 127
IndexIterator, 126
Iterator, 126
mean, 127
means, 127
overlaps, 127
parameter_indices, 127
RValueFunction, 126
variances, 127
eos::components, 35
eos::ConcreteObservable, 134
clone, 134
ConcreteObservable, 134
evaluate, 134
kinematics, 134
name, 135
options, 135
parameters, 135
eos::ConcreteObservableFactory, 136
~ConcreteObservableFactory, 136
ConcreteObservableFactory, 136
make, 136
eos::ConditionVariable, 137
~ConditionVariable, 137
acquire_then_signal, 137
broadcast, 137
ConditionVariable, 137
signal, 137
wait, 137
eos::Constraint, 162
~Constraint, 163
begin_blocks, 163
begin_observables, 163
BlockIterator, 163
Constraint, 163
end_blocks, 163
end_observables, 163
make, 163
name, 163
ObservableIterator, 163
eos::deriv, 36
eos::deriv::TwoSided, 400
eos::Diagnostics, 176
~Diagnostics, 177
add, 177
begin, 177
Diagnostics, 177
end, 177
Iterator, 177
size, 177
eos::Diagnostics::Entry, 180
description, 180
value, 180
eos::EMContributions, 179
omegaem_1010, 179
omegaem_22, 179
omegaem_27, 179
omegaem_29, 179
omegaem_77, 179
omegaem_79, 179
omegaem_99, 179
eos::EquationSolver, 181
~EquationSolver, 181
add, 181
Equation, 181
EquationSolver, 181
solve, 181
eos::EquationSolver::Config, 145
Default, 145
maximum_steps, 145
strategy, 145
tolerance, 145
eos::EquationSolver::Solution, 379
parameters, 379
valid, 379
value, 379
eos::Exception, 183
~Exception, 183
Exception, 183
what, 183
eos::FormFactorAdapter, 189
clone, 189
evaluate, 189
FormFactorAdapter, 189
kinematics, 189
name, 190
options, 190
parameters, 190

eos::FormFactorAdapterFactory, 191
~FormFactorAdapterFactory, 191
FormFactorAdapterFactory, 191
make, 191
eos::FormFactorFactory< OneHalfPlusToOneHalfPlus >, 192
create, 192
eos::FormFactorFactory< PToP >, 193
create, 193
eos::FormFactorFactory< PToV >, 194
create, 194
eos::FormFactorRatioAdapter, 195
clone, 195
evaluate, 195
FormFactorRatioAdapter, 195
kinematics, 195
name, 196
options, 196
parameters, 196
eos::FormFactorRatioAdapterFactory, 197
~FormFactorRatioAdapterFactory, 197
FormFactorRatioAdapterFactory, 197
make, 197
eos::FormFactors< OneHalfPlusToOneHalfPlus >, 198
~FormFactors, 198
f_long_a, 198
f_long_v, 198
f_perp_a, 198
f_perp_v, 198
eos::FormFactors< PToP >, 199
~FormFactors, 199
f_0, 199
f_p, 199
f_p_d1, 199
f_p_d2, 199
f_t, 199
eos::FormFactors< PToV >, 201
~FormFactors, 201
a_0, 201
a_1, 201
a_12, 201
a_2, 201
v, 201
eos::HardScattering, 208
I1, 208
eos::hdf5, 37
hid_t, 37
operator<<, 37
operator>>, 38
TypePtr, 37
eos::hdf5::Array, 65
Array, 65
copy_from_hdf5, 65
copy_to_hdf5, 65
name, 65
size, 65
Type, 65
type_id, 66
eos::hdf5::Attribute, 67
DataSet, 67
operator=, 67
RecordType, 67
value, 67
eos::hdf5::AttributeHandle, 68
~AttributeHandle, 68
attribute_id, 68
DataSetHandle, 68
read, 68
type_id, 68
write, 68
eos::hdf5::Composite, 130
~Composite, 130
Composite, 130
copy_from_hdf5, 130
copy_to_hdf5, 130
name, 130
size, 130
Type, 130
type_id, 131
eos::hdf5::CompositeCopier, 132
copy_from_hdf5, 132
copy_to_hdf5, 132
eos::hdf5::CompositeCopier< 0, TypeTuple_, Tuple_ >, 133
copy_from_hdf5, 133
copy_to_hdf5, 133
eos::hdf5::DataSet, 164
~DataSet, 165
create_attribute, 165
create_or_open_attribute, 165
end, 165
hdf5::File, 166
name, 165
open_attribute, 165
operator<<, 166
operator>>, 166
records, 165
RecordType, 165
set_index, 165
eos::hdf5::DataSetHandle, 167
~DataSetHandle, 168
create_attribute, 168
File, 168
open_attribute, 168
read_one, 168
select, 168
set_id, 168

size, 168
space_id, 168
type_id, 168
write_one, 168
eos::hdf5::DataType, 169
type_id, 169
eos::hdf5::DataType< const char * >, 170
type_id, 170
eos::hdf5::DataType< double >, 171
type_id, 171
eos::hdf5::DataType< int >, 172
type_id, 172
eos::hdf5::DataType< short >, 173
type_id, 173
eos::hdf5::DataType< unsigned >, 174
type_id, 174
eos::hdf5::File, 185
~File, 185
copy, 186
Create, 186
create_data_set, 186
create_group, 186
create_or_open_data_set, 186
Exists, 186
group_exists, 187
name, 187
number_of_objects, 187
Open, 187
open_data_set, 187
eos::hdf5::FileHandle, 188
~FileHandle, 188
File, 188
id, 188
read_only, 188
eos::hdf5::Scalar, 368
~Scalar, 368
copy_from_hdf5, 368
copy_to_hdf5, 368
name, 368
Scalar, 368
size, 369
Type, 368
type_id, 369
eos::hdf5::Type, 401
~Type, 401
copy_from_hdf5, 401
copy_to_hdf5, 401
name, 401
size, 401
type_id, 401
eos::HDF5Error, 209
HDF5Error, 209
eos::HierarchicalClustering, 210
~HierarchicalClustering, 211
add, 211
begin_clusters, 211
begin_components, 212
begin_map, 212
ClusterIterator, 211
ComponentIterator, 211
end_clusters, 212
end_components, 212
end_map, 212
HierarchicalClustering, 211
initial_guess, 212
MapIterator, 211
MixtureDensity, 211
run, 212
eos::HierarchicalClustering::Component, 129
~Component, 129
Component, 129
covariance, 129
determinant, 129
inverse_covariance, 129
mean, 129
weight, 129
eos::HierarchicalClustering::Config, 160
Default, 160
equal_weights, 161
kill_clusters, 161
maximum_steps, 161
precision, 161
Quick, 160
eos::Histogram< 1 >, 213
~Histogram, 214
begin, 214
ConstIterator, 214
end, 214
entries, 214
Histogram, 214
insert, 214
WithEqualBinning, 214
eos::Histogram< 1 >::Bin, 77
~Bin, 77
Bin, 77
lower, 78
operator<, 78
upper, 78
value, 78
eos::Histogram< 2 >, 216
~Histogram, 217
begin, 217
cbegin, 217
cend, 217
ConstIterator, 217
end, 217
entries, 217
find, 217

Histogram, 217
insert, 217
Iterator, 217
WithEqualBinning, 217
eos::Histogram< 2 >::Bin, 75
~Bin, 75
Bin, 75
lower, 76
operator<, 76
upper, 76
value, 76
eos::HLMW2005, 221
eos::impl, 39
eos::implementation, 40
eos::implementation::DoStringify, 178
stringify, 178
eos::implementation::ResultOf< Result_(*)(Args_...)>, 362
Type, 362
eos::implementation::ResultOf< Result_(Class_::*)(Args_...)>, 363
Type, 363
eos::InstantiationPolicy, 41
eos::InstantiationPolicy< T_, NonCopyable >, 223
InstantiationPolicy, 223
eos::InstantiationPolicy< T_, NonInstantiable >, 224
eos::InstantiationPolicy< T_, Singleton >, 225
DeleteOnDestruction, 225
instance, 225
InstantiationPolicy, 225
eos::InstantiationPolicy::DeleteOnDestruction, 175
eos::InternalError, 226
InternalError, 226
eos::Kinematics, 227
~Kinematics, 228
as_string, 228
clone, 228
declare, 228
Kinematics, 227
operator==, 228
set, 228
eos::KinematicVariable, 230
~KinematicVariable, 230
Implementation< Kinematics >, 231
Kinematics, 231
operator double, 230
operator(), 230
operator=, 230
eos::KMPW2010FormFactors< PToP >, 232
f_0, 232
f_p, 232
f_t, 232
KMPW2010FormFactors, 232
make, 232
eos::KMPW2010FormFactors< PToV >, 233
a_0, 233
a_1, 233
a_12, 233
a_2, 233
KMPW2010FormFactors, 233
make, 233
v, 234
eos::LambdaBToLambdaDilepton< LargeRecoil >, 235
~LambdaBToLambdaDilepton, 236
differential_a_fb_combined, 236
differential_a_fb_hadronic, 236
differential_a_fb_leptonic, 236
differential_branching_ratio, 236
differential_fzero, 236
integrated_a_fb_combined, 236
integrated_a_fb_hadronic, 236
integrated_a_fb_leptonic, 236
integrated_branching_ratio, 236
integrated_fzero, 236
LambdaBToLambdaDilepton, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 237
~LambdaBToLambdaDilepton, 238
differential_a_fb_combined, 238
differential_a_fb_hadronic, 238
differential_a_fb_leptonic, 238
differential_branching_ratio, 238
differential_fzero, 238
integrated_a_fb_combined, 238
integrated_a_fb_hadronic, 238
integrated_a_fb_leptonic, 238
integrated_branching_ratio, 238
integrated_fzero, 238
LambdaBToLambdaDilepton, 238
eos::LargeRecoil, 239
eos::LCDA, 240
evolve_gegenbauer_moment, 240
eos::Lock, 241
~Lock, 241
Lock, 241
eos::Log, 242
~Log, 242
get_log_level, 242
InstantiationPolicy< Log, Singleton >, 243
LogMessageHandler, 243
message, 242
set_log_level, 242
set_log_stream, 243
set_program_name, 243
eos::LogLikelihood, 244
~LogLikelihood, 245

add, 245
begin, 245
bootstrap_p_value, 245
clone, 246
ConstraintIterator, 245
end, 246
LogLikelihood, 245
number_of_observations, 246
observable_cache, 246
operator(), 246
parameters, 247
reset, 247
eos::LogLikelihoodBlock, 248
~LogLikelihoodBlock, 249
Amoroso, 249, 250
AmorosoLimit, 250
AmorosoMode, 251
as_string, 251
clone, 251
evaluate, 251
Gaussian, 252
LogGamma, 252
Mixture, 253
MultivariateGaussian, 253
number_of_observations, 254
prepare_sampling, 254
sample, 254
significance, 254
eos::LogMessageHandler, 255
~LogMessageHandler, 255
Log::message, 255
operator<<, 255
eos::LogPrior, 256
~LogPrior, 257
_parameter_descriptions, 261
_parameters, 261
as_string, 257
begin, 257
clone, 257
Discrete, 257
end, 258
Flat, 258
Gauss, 259
Iterator, 257
LogGamma, 259
LogPrior, 257
Make, 260
mean, 260
operator(), 260
sample, 261
variance, 261
eos::LongDistance, 262
g_had_ccbar, 262
eos::LowRecoil, 263
eos::MarkovChain, 264
~MarkovChain, 265
clear, 266
current_state, 266
dump_description, 266
dump_history, 266
dump_proposal, 266
history, 266
hyper_parameter, 266
iterations_last_run, 266
keep_history, 266
MarkovChain, 265
normalized_density, 266
parameter_descriptions, 267
proposal_accepted, 267
proposal_function, 267
proposed_state, 267
read_data, 267
read_descriptions, 268
reset, 268
run, 268
set_analysis, 268
set_mode, 268
set_point, 269
statistics, 269
eos::MarkovChain::History, 219
keep, 220
local_mode, 219
mean_and_covariance, 219
mean_and_variance, 219
states, 220
eos::MarkovChain::HyperParameter, 222
component, 222
eos::MarkovChain::ProposalFunction, 349
~ProposalFunction, 349
adapt, 349
clone, 350
dump_state, 350
evaluate, 350
propose, 350
eos::MarkovChain::State, 381
hyper_parameter, 382
Iterator, 382
log_likelihood, 382
log_posterior, 382
log_prior, 382
point, 382
State, 382
eos::MarkovChain::Stats, 383
iterations_accepted, 383
iterations_invalid, 383
iterations_rejected, 383
iterations_total, 383
mean_of_parameters, 384

mean_of_posterior, 384
mode_of_posterior, 384
parameters_at_mode, 384
variance_of_parameters, 384
variance_of_posterior, 384
eos::MarkovChainSampler, 270
~MarkovChainSampler, 270
build_global_local, 271
config, 271
MarkovChainSampler, 270
massive_mode_finding, 271
pre_run_info, 271
resume, 271
run, 271
eos::MarkovChainSampler::Config, 139
adapt_iterations, 141
block_proposal_parameters, 141
chunk_size, 141
chunks, 141
Default, 140
find_modes, 141
global_local_config, 141
max_efficiency, 141
min_efficiency, 141
need_main_run, 141
need_prerun, 141
number_of_chains, 141
output_file, 142
parallelize, 142
partitions, 142
prerun_chains_per_partition, 142
prerun_iterations_max, 142
prerun_iterations_min, 142
prerun_iterations_update, 142
proposal, 142
Quick, 140
rvalue_criterion_param, 142
rvalue_criterion_posterior, 142
scale_automatic, 143
scale_nuisance, 143
scale_reduction, 143
seed, 143
skip_initial, 143
store, 143
store_observables_and_proposals, 143
store_prerun, 143
student_t_degrees_of_freedom, 143
use_posterior_rvalue, 144
use_strict_rvalue_definition, 144
eos::MarkovChainSampler::PreRunInfo, 344
converged, 344
iterations, 344
iterations_at_convergence, 344
rvalue_parameters, 344
rvalue_posterior, 344
eos::MemoisationControl, 273
~MemoisationControl, 273
clear, 273
MemoisationControl, 273
register_clear_function, 273
eos::Memoiser, 274
~Memoiser, 274
clear, 274
FunctionType, 274
KeyType, 274
Memoiser, 274
number_of_memoisations, 274
operator(), 274
eos::Minimal, 275
eos::Model, 276
~Model, 276
make, 276
eos::ModelComponent< components::CKM >, 277
ckm_cb, 277
ckm_cd, 277
ckm_cs, 277
ckm_tb, 277
ckm_td, 277
ckm_ts, 278
ckm_ub, 278
ckm_ud, 278
ckm_us, 278
eos::ModelComponent< components::DeltaB1 >, 279
wilson_coefficients_b_to_s, 279
eos::ModelComponent< components::QCD >, 280
alpha_s, 280
m_b_kin, 280
m_b_msbar, 280
m_b_pole, 280
m_b_ps, 280
m_c_kin, 281
m_c_msbar, 281
m_c_pole, 281
m_s_msbar, 281
m_t_msbar, 281
m_t_pole, 281
m_ud_msbar, 281
eos::Mutable, 294
~Mutable, 294
clone, 295
name, 295
operator double, 295
operator(), 295
operator=, 295
eos::Mutex, 296
~Mutex, 296
Mutex, 296

mutex, 296
eos::NLO, 297
eos::NoSuchChiSquaredError, 298
NoSuchChiSquaredError, 298
eos::NoSuchModelError, 299
NoSuchModelError, 299
eos::Observable, 300
clone, 300
evaluate, 300
kinematics, 300
make, 300
name, 301
options, 301
parameters, 301
eos::ObservableCache, 302
~ObservableCache, 303
add, 303
clone, 303
Id, 302
observable, 303
ObservableCache, 303
parameters, 303
reset, 303
size, 304
update, 304
eos::ObservableFactory, 305
~ObservableFactory, 305
make, 305
ObservableFactory, 305
eos::ObservableNameError, 306
ObservableNameError, 306
eos::ObservableSet, 307
~ObservableSet, 308
add, 308
begin, 308
end, 308
Iterator, 307
ObservableSet, 308
parameters, 308
size, 308
eos::OneHalfPlusToOneHalfPlus, 310
eos::OneOf, 311
accept_returning, 311
OneOf, 311
eos::OneOfValue, 312
accept, 312
OneOfValue, 312
value, 312
eos::OneOfValueBase, 313
~OneOfValueBase, 313
accept, 313
eos::OneOfVisitor, 314
eos::OneOfVisitorVisit, 315
visit, 315
eos::OneOfVisitorWrapper, 316
OneOfVisitorWrapper, 316
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >, 317
OneOfVisitorWrapperVisit, 317
result, 317
underlying, 317
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>, 318
OneOfVisitorWrapperVisit, 318
visit, 318
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >, 319
OneOfVisitorWrapperVisit, 319
underlying, 319
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>, 320
OneOfVisitorWrapperVisit, 320
visit, 320
eos::Options, 323
~Options, 324
as_string, 324
get, 324
has, 324
operator+, 324
operator==, 324
Options, 323
set, 324
eos::Parameter, 329
~Parameter, 330
central, 330
clone, 330
Id, 330
id, 330
Implementation< Parameters >, 331
max, 331
min, 331
name, 331
operator double, 331
operator(), 331
operator=, 331
Parameter, 330
Parameters, 331
eos::ParameterDescription, 332
discrete, 332
max, 332
min, 332
nuisance, 332
parameter, 332
eos::ParameterRange, 333
max, 333
min, 333

eos::Parameters, 334
~Parameters, 335
begin, 335
clone, 335
declare, 335
Defaults, 335
end, 335
Implementation< Parameter >, 336
Implementation< Parameters >, 336
Iterator, 335
Parameter, 336
set, 336
eos::ParameterUser, 337
_ids, 338
begin, 337
ConstIterator, 337
end, 337
uses, 337, 338
eos::PionLCDAs, 339
~PionLCDAs, 340
a2pi, 340
a4pi, 340
deltapipi, 340
diagnostics, 340
eta3pi, 340
f3pi, 340
mupi, 340
omega3pi, 340
omega4pi, 340
phi, 340
phi3p, 340
phi3s, 340
phi3s_d1, 340
phi4, 340
phi4_d1, 340
phi4_d2, 340
PionLCDAs, 340
psi4, 340
psi4_i, 340
eos::PopulationMonteCarloSampler, 341
~PopulationMonteCarloSampler, 342
calculate_weights, 342
config, 342
draw_samples, 342
PopulationMonteCarloSampler, 341, 342
read_samples, 342
run, 343
status, 343
eos::PopulationMonteCarloSampler::Config, 146
adjust_sample_size, 148
block_decomposition, 148
chunk_size, 148
chunks, 148
component_means, 148
component_offset, 149
component_variances, 149
component_weights, 149
components_per_cluster, 149
convergence_eff_sample_size, 149
convergence_perplexity, 149
crop_highest_weights, 149
Default, 148
degrees_of_freedom, 149
final_chunk_size, 149
group_by_r_value, 149
ignore_eff_sample_size, 150
ignore_groups, 150
maximum_relative_std_deviation, 150
minimum_eff_sample_size, 150
minimum_overlap, 150
minimum_perplexity, 150
minimum_steps, 150
mode_distance, 150
need_prerun, 151
number_of_workers, 151
output_file, 151
override_global_local_proposal, 151
parallelize, 151
patch_around_local_mode, 151
print_steps, 151
Quick, 148
r_value_no_nuisance, 151
random_start, 151
seed, 151
single_cluster, 152
skip_initial, 152
sliding_window, 152
starting_points, 152
std_dev_reduction, 152
store, 152
store_hc_initial, 152
store_input_components, 152
store_prerun, 152
super_clusters, 152
eos::PopulationMonteCarloSampler::Output, 325
component_record, 326
component_type, 326
ComponentType, 326
ignore_record, 326
ignore_type, 326
IgnoreType, 326
sample_record, 326
sample_type, 326
SampleType, 326
statistics_record, 326
statistics_type, 326
StatisticsType, 326
weight_record, 326

weight_type, 326
WeightType, 326
eos::PopulationMonteCarloSampler::Status, 385
chunk_size, 385
converged, 385
eff_sample_size, 385
evidence, 385
iterations_at_convergence, 386
perplexity, 386
Status, 385
eos::PriorSampler, 346
~PriorSampler, 347
add, 347
observables_type, 347
ObservablesType, 346
PriorSampler, 346
run, 347
SamplesList, 346
eos::PriorSampler::Config, 154
Default, 154
n_samples, 154
n_workers, 154
output_file, 154
parallelize, 155
seed, 155
store_parameters, 155
eos::PrivateImplementationPattern, 348
~PrivateImplementationPattern, 348
_imp, 348
PrivateImplementationPattern, 348
eos::proposal_functions::GlobalLocal::Config
equal, 157
log_posterior, 157
posterior, 157
eos::proposal_functions, 42
meta_record, 43
meta_type, 43
MetaRecord, 42
MetaType, 42
MultivariateProposalPtr, 42
random_index, 43
sliding_window, 43
eos::proposal_functions::AdjacencyMatrix, 54
add, 54
AdjacencyMatrix, 54
indices, 54
jump, 54
number_of_clusters, 55
reserve, 55
state, 55
eos::proposal_functions::BlockDecomposition, 79
~BlockDecomposition, 80
adapt, 80
add, 80
BlockDecomposition, 80
clone, 80
dump_state, 80
evaluate, 80
MultivariateAccess, 81
priors_type, 81
PriorsType, 80
propose, 81
eos::proposal_functions::Factory, 184
make, 184
eos::proposal_functions::GlobalLocal, 203
~GlobalLocal, 205
adapt, 205
clone, 205
component_probabilities, 205
component_type, 206
ComponentType, 204
config, 206
dump_state, 206
evaluate, 206
GlobalLocal, 204, 205
history_states, 206
history_type, 206
HistoryType, 204
jump_type, 206
JumpType, 204
local_covariance_type, 206
local_covariances, 206
LocalCovarianceType, 204
mode, 206
propose, 206
eos::proposal_functions::GlobalLocal::Config, 156
clustering_maximum_r_value, 157
clustering_strict_r_value, 157
cooling_power, 157
Default, 157
equal_weight_components, 157
history_point_weighting, 157
history_points, 157
history_points_local_covariance_size, 157
history_points_ordered, 158
HistoryPointWeighting, 156
join_chains_symmetrically, 158
local_jump_probability, 158
long_jump_indices, 158
minimum_relative_cluster_weight, 158
perform_clustering, 158
rescale_local_covariance, 158
skip_initial, 158
eos::proposal_functions::Multivariate, 282
~Multivariate, 283
_compute_cholesky_and_inverse, 284
_compute_norm, 284
_copy, 284

_covariance, 285
_covariance_chol, 285
_covariance_inverse, 285
_dimension, 285
_dump_covariance, 284
_index_list, 285
_tmp_left, 285
_tmp_right, 285
_tmp_sample_covariance_current, 285
adapt, 284
adaptations, 285
cooling_power, 285
covariance, 284
covariance_scale, 285
covariance_scale_max, 286
covariance_scale_min, 286
covariance_scale_update_factor, 286
covariance_type, 284
CovarianceType, 283
dimension, 284
Multivariate, 283
norm, 286
rescale, 284
reset, 284
set_indices, 285
eos::proposal_functions::MultivariateAccess, 287
access, 287
eos::proposal_functions::MultivariateGaussian, 288
~MultivariateGaussian, 289
_compute_norm, 289
clone, 289
dump_state, 289
evaluate, 289
MultivariateGaussian, 289
propose, 289
scalars_type, 289
ScalarsType, 289
eos::proposal_functions::MultivariateStudentT, 291
~MultivariateStudentT, 292
_compute_norm, 292
clone, 292
dof, 293
dump_state, 292
evaluate, 292
MultivariateStudentT, 292
propose, 292
scalars_type, 292
ScalarsType, 292
eos::proposal_functions::UnknownProposalError, 407
UnknownProposalError, 407
eos::PToP, 351
eos::PToV, 352
eos::QCD, 353
alpha_s, 354
AnomalousMassDimension, 353
beta_function_nf_3, 355
beta_function_nf_4, 355
beta_function_nf_5, 355
beta_function_nf_6, 355
BetaFunction, 353
casimir_a, 356
casimir_f, 356
gamma_m_nf_3, 356
gamma_m_nf_4, 356
gamma_m_nf_5, 356
m_q_kin, 354
m_q_msbar, 354
m_q_pole, 355
m_q_ps, 355
eos::QCDFIntegrals, 357
dilepton_bottom_case, 357
dilepton_charm_case, 357
dilepton_massless_case, 358
photon_bottom_case, 358
photon_charm_case, 359
photon_massless_case, 359
eos::QCDFIntegrals::Results, 364
j0_parallel, 365
j0_perp, 365
j0bar_perp, 365
j1_parallel, 365
j1_perp, 365
j2_perp, 365
j3_parallel, 365
j4_parallel, 365
j4_perp, 365
j5_perp, 365
j6_perp, 365
j7_perp, 365
jtilde1_perp, 365
jtilde2_parallel, 365
eos::RandomNumberGenerator, 360
~RandomNumberGenerator, 360
max, 360
min, 360
operator(), 361
RandomNumberGenerator, 360
eos::RValue, 366
approximation, 366
gelman_rubin, 366
eos::Save, 367
~Save, 367
Save, 367
eos::SelectOneOfType< Want_ >, 370
Type, 370
eos::SelectOneOfType< Want_, Try_, Rest_...>, 371

Type, 371
eos::ShortDistanceLowRecoil, 372
c7eff, 372
c9eff, 372
eos::SMComponent< components::CKM >, 374
ckm_cb, 374
ckm_cd, 374
ckm_cs, 374
ckm_tb, 374
ckm_td, 374
ckm_ts, 375
ckm_ub, 375
ckm_ud, 375
ckm_us, 375
SMComponent, 374
eos::SMComponent< components::DeltaB1 >, 376
SMComponent, 376
wilson_coefficients_b_to_s, 376
eos::SMComponent< components::QCD >, 377
alpha_s, 377
m_b_kin, 377
m_b_msbar, 377
m_b_pole, 377
m_b_ps, 378
m_c_kin, 378
m_c_msbar, 378
m_c_pole, 378
m_s_msbar, 378
m_t_msbar, 378
m_t_pole, 378
m_ud_msbar, 378
SMComponent, 377
eos::StandardModel, 380
~StandardModel, 380
make, 380
StandardModel, 380
eos::TestObservable, 387
~TestObservable, 387
clone, 387
evaluate, 387
k, 388
kinematics, 388
mass, 388
mass_name, 388
n, 388
name, 388
o, 388
options, 388
p, 388
parameters, 388
set_option, 388
TestObservable, 387
eos::Thread, 389
~Thread, 389
completed, 390
Function, 389
Thread, 389
eos::ThreadPool, 391
~ThreadPool, 391
enqueue, 391
instance, 391
number_of_threads, 391
ThreadPool, 391
wait_for_free_capacity, 391
eos::Ticket, 392
~Ticket, 392
mark, 392
Ticket, 392
TicketList, 393
wait, 392
eos::TicketList, 394
~TicketList, 394
push_back, 394
TicketList, 394
wait, 394
eos::TopLoops, 396
A0, 396
A1, 396
B0, 396
B1, 396
C0, 397
C1, 397
D0, 397
D1, 397
E0, 397
E1, 397
F0, 398
F1, 398
G1, 398
eos::TryLock, 399
~TryLock, 399
operator(), 399
TryLock, 399
eos::UnknownConstraintError, 403
UnknownConstraintError, 403
eos::UnknownKinematicVariableError, 404
UnknownKinematicVariableError, 404
eos::UnknownOptionError, 405
UnknownOptionError, 405
eos::UnknownParameterError, 406
UnknownParameterError, 406
eos::UsedParameter, 408
UsedParameter, 408
eos::VerifiedRange, 409
max, 409
min, 409
operator T_, 410
operator=, 410

VerifiedRange, 409
eos::VerifiedRangeError, 411
VerifiedRangeError, 411
eos::VerifiedRangeOverflow, 412
VerifiedRangeOverflow, 412
eos::VerifiedRangeUnderflow, 413
VerifiedRangeUnderflow, 413
eos::Welford, 414
add, 414
mean, 414
number_of_elements, 414
std_deviation, 414
variance, 414
Welford, 414
eos::WilsonCoefficients< BToS >, 415
_alpha_s, 417
_primed_coefficients, 417
_scalar_tensor_coefficients, 417
_sm_like_coefficients, 417
c1, 417
c10, 417
c10prime, 417
c2, 417
c2b, 417
c3, 417
c4, 417
c5, 417
c6, 417
c7, 417
c7prime, 417
c8, 417
c8prime, 417
c9, 417
c9prime, 417
cP, 417
cPprime, 417
cq3, 417
cq4, 417
cq5, 417
cq6, 417
cS, 417
cSprime, 417
cT, 417
cT5, 417
WilsonCoefficients, 415
eos::WilsonPolynomialCloner, 419
visit, 419
WilsonPolynomialCloner, 419
eos::WilsonPolynomialEvaluator, 420
visit, 420
eos::WilsonPolynomialPrinter, 421
visit, 421
WilsonPolynomialPrinter, 421
eos::WilsonScanComponent, 422
wilson_coefficients_b_to_s, 422
WilsonScanComponent, 422
eos::WilsonScanModel, 423
~WilsonScanModel, 423
make, 423
WilsonScanModel, 423
eos::WrappedForwardIterator, 424
~WrappedForwardIterator, 427
difference_type, 427
iterator_category, 427
operator*, 427
operator++, 427
operator->, 427
operator=, 427
operator==, 427
pointer, 427
reference, 427
underlying_iterator, 427
value_type, 427
WrappedForwardIterator, 427
eos::WrappedForwardIterator::Base, 69
~Base, 69
clone, 69
equal, 69
increment, 69
pointer, 69
reference, 70
underlying_iterator_ptr, 70
eos::WrappedForwardIterator::BaseImpl, 71
BaseImpl, 71
clone, 71
equal, 71
i, 72
increment, 71
pointer, 71
reference, 72
underlying_iterator_ptr, 72
EOS_GUARD_EOS_B_DECAYS_-EXCLUSIVE_B_TO_U_L_NU_HH
b-to-pi-l-nu.hh, 429
EOS_GUARD_EOS_B_DECAYS_-PROPERTIES_HH
properties.hh, 430
EOS_GUARD_EOS_FORM_FACTORS_-ANALYTIC_B_TO_PI_HH
analytic-b-to-pi.hh, 432
EOS_GUARD_EOS_FORM_FACTORS_-BARYONIC_HH
baryonic.hh, 434
EOS_GUARD_EOS_FORM_FACTORS_-BARYONIC_IMPL_HH
baryonic-impl.hh, 433
EOS_GUARD_EOS_FORM_FACTORS_FORM_-FACTOR_ADAPTER_HH

form-factor-adapter.hh, 435
EOS_GUARD_EOS_FORM_FACTORS_FORM_-FACTORS_FWD_HH
form-factors-fwd.hh, 436
EOS_GUARD_EOS_FORM_FACTORS_PI_-LCDAS_HH
pi-lcdas.hh, 440
EOS_GUARD_EOS_RARE_B_DECAYS_-BARYONIC_B_TO_S_DILEPTON_HH
lambda-b-to-lambda-dilepton.hh, 454
EOS_GUARD_EOS_RARE_B_DECAYS_-QCDF_INTEGRALS_HH
qcdf_integrals.hh, 456
EOS_GUARD_EOS_UTILS_CLUSTER_HH
cluster.hh, 465
EOS_GUARD_EOS_UTILS_DIAGNOSTICS_HH
diagnostics.hh, 471
EOS_GUARD_EOS_UTILS_EQUATION_-SOLVER_HH
equation_solver.hh, 472
EOS_GUARD_EOS_UTILS_INTEGRATE_-IMPL_HH
integrate-impl.hh, 481
EOS_GUARD_EOS_UTILS_MUTABLE_FWD_-HH
mutable-fwd.hh, 497
EOS_GUARD_EOS_UTILS_MUTABLE_HH
mutable.hh, 498
EOS_GUARD_EOS_UTILS_PARAMETERS_-FWD_HH
parameters-fwd.hh, 504
EOS_GUARD_EOS_UTILS_POLYLOG_HH
polylog.hh, 506
EOS_GUARD_EOS_UTILS_PROPROSAL_-FUNCTIONS_HH
proposal_functions.hh, 513
EOS_GUARD_EOS_UTILS_WELFORD_HH
welford.hh, 525
EOS_GUARD_SRC_CONSTRAINT_HH
constraint.hh, 431
EOS_GUARD_SRC_FORM_FACTORS_FORM_-FACTORS_HH
form-factors.hh, 437
EOS_GUARD_SRC_FORM_FACTORS_-MESONIC_HH
mesonic.hh, 439
EOS_GUARD_SRC_RARE_B_DECAYS_-BREMSSTRAHLUNG_HH
bremsstrahlung.hh, 442
EOS_GUARD_SRC_RARE_B_DECAYS_-CHARM_LOOPS_HH
charm-loops.hh, 443
EOS_GUARD_SRC_RARE_B_DECAYS_-DECAYS_HH
decays.hh, 444
EOS_GUARD_SRC_RARE_B_DECAYS_EM_-CONTRIBUTIONS_HH
em-contributions.hh, 445
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_DILEPTON_HH
exclusive-b-to-dilepton.hh, 446
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_DILEPTON_-HH
exclusive-b-to-s-dilepton.hh, 449
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_DILEPTON_-LARGE_RECOIL_HH
exclusive-b-to-s-dilepton-large-recoil.hh, 447
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_DILEPTON_-LOW_RECOIL_HH
exclusive-b-to-s-dilepton-low-recoil.hh, 448
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_GAMMA_HH
exclusive-b-to-s-gamma.hh, 450
EOS_GUARD_SRC_RARE_B_DECAYS_-FORM_FACTORS_IMPL_HH
mesonic-impl.hh, 438
EOS_GUARD_SRC_RARE_B_DECAYS_-HARD_SCATTERING_HH
hard-scattering.hh, 451
EOS_GUARD_SRC_RARE_B_DECAYS_-INCLUSIVE_B_TO_S_DILEPTON_HH
inclusive-b-to-s-dilepton.hh, 452
EOS_GUARD_SRC_RARE_B_DECAYS_-INCLUSIVE_B_TO_S_GAMMA_HH
inclusive-b-to-s-gamma.hh, 453
EOS_GUARD_SRC_RARE_B_DECAYS_-LONG_DISTANCE_HH
long-distance.hh, 455
EOS_GUARD_SRC_UTILS_ACCUMULATOR_-HH
accumulator.hh, 457
EOS_GUARD_SRC_UTILS_ANALYSIS_FWD_-HH
analysis-fwd.hh, 458
EOS_GUARD_SRC_UTILS_ANALYSIS_HH
analysis.hh, 459
EOS_GUARD_SRC_UTILS_ANALYSIS_TEST_-HH
analysis_TEST.hh, 460
EOS_GUARD_SRC_UTILS_APPLY_HH
apply.hh, 461
EOS_GUARD_SRC_UTILS_CARTESIAN_-PRODUCT_HH
cartesian-product.hh, 462

EOS_GUARD_SRC_UTILS_CHI_SQUARED_-HH
chi-squared.hh, 463
EOS_GUARD_SRC_UTILS_CKM_SCAN_-MODEL_HH
ckm_scan_model.hh, 464
EOS_GUARD_SRC_UTILS_CONCRETE_-OBSERVABLE_HH
concrete_observable.hh, 467
EOS_GUARD_SRC_UTILS_CONDITION_-VARIABLE_HH
condition_variable.hh, 468
EOS_GUARD_SRC_UTILS_DESTRINGIFY_HH
destringify.hh, 470
EOS_GUARD_SRC_UTILS_EXCEPTION_HH
exception.hh, 473
EOS_GUARD_SRC_UTILS_HDF5_FWD_HH
hdf5-fwd.hh, 474
EOS_GUARD_SRC_UTILS_HDF5_HH
hdf5.hh, 476
EOS_GUARD_SRC_UTILS_HIERARCHICAL_-CLUSTERING_HH
hierarchical-clustering.hh, 477
EOS_GUARD_SRC_UTILS_HISTROGRAM_HH
histogram.hh, 478
EOS_GUARD_SRC_UTILS_INSTANTIATION_-POLICY_HH
instantiation_policy.hh, 480
EOS_GUARD_SRC_UTILS_INSTANTIATION_-POLICY_IMPL_HH
instantiation_policy-impl.hh, 479
EOS_GUARD_SRC_UTILS_INTEGRATE_HH
integrate.hh, 482
EOS_GUARD_SRC_UTILS_KINEMATIC_HH
kinematic.hh, 483
EOS_GUARD_SRC_UTILS_LCDA_HH
lcda.hh, 484
EOS_GUARD_SRC_UTILS_LIKELIHOOD_HH
log_likelihood.hh, 488
EOS_GUARD_SRC_UTILS_LOCK_HH
lock.hh, 485
EOS_GUARD_SRC_UTILS_LOG_HH
log.hh, 486
EOS_GUARD_SRC_UTILS_LOG_-LIKELIHOOD_FWD_HH
log_likelihood-fwd.hh, 487
EOS_GUARD_SRC_UTILS_MARKOV_-CHAIN_HH
markov_chain.hh, 491
EOS_GUARD_SRC_UTILS_MATRIX_HH
matrix.hh, 494
EOS_GUARD_SRC_UTILS_MC_SAMPLER_-HH
markov_chain_sampler.hh, 492
EOS_GUARD_SRC_UTILS_MEMOISE_HH
memoise.hh, 495
EOS_GUARD_SRC_UTILS_MODEL_HH
model.hh, 496
EOS_GUARD_SRC_UTILS_MUTEX_HH
mutex.hh, 499
EOS_GUARD_SRC_UTILS_OBSERVABLE_-CACHE_HH
observable_cache.hh, 500
EOS_GUARD_SRC_UTILS_OBSERVABLE_HH
observable.hh, 441
EOS_GUARD_SRC_UTILS_OBSERVABLE_-SET_HH
observable_set.hh, 501
EOS_GUARD_SRC_UTILS_ONE_OF_HH
one-of.hh, 502
EOS_GUARD_SRC_UTILS_OPTIONS_HH
options.hh, 503
EOS_GUARD_SRC_UTILS_PARAMETERS_HH
parameters.hh, 505
EOS_GUARD_SRC_UTILS_PMC_SAMPLER_-HH
population_monte_carlo_sampler.hh, 507
EOS_GUARD_SRC_UTILS_POWER_OF_HH
power_of.hh, 508
EOS_GUARD_SRC_UTILS_PRIOR_FWD_HH
log_prior-fwd.hh, 489
EOS_GUARD_SRC_UTILS_PRIOR_HH
log_prior.hh, 490
EOS_GUARD_SRC_UTILS_PRIOR_-SAMPLER_HH
prior_sampler.hh, 509
EOS_GUARD_SRC_UTILS_QCD_HH
qcd.hh, 514
EOS_GUARD_SRC_UTILS_RANDOM_-NUMBER_GENERATOR_HH
random_number_generator.hh, 515
EOS_GUARD_SRC_UTILS_RVALUE_HH
rvalue.hh, 516
EOS_GUARD_SRC_UTILS_SAVE_HH
save.hh, 517
EOS_GUARD_SRC_UTILS_STANDARD_-MODEL_HH
standard-model.hh, 518
EOS_GUARD_SRC_UTILS_STRINGIFY_HH
stringify.hh, 519
EOS_GUARD_SRC_UTILS_THREAD_HH
thread.hh, 520
EOS_GUARD_SRC_UTILS_THREAD_POOL_-HH
thread_pool.hh, 521
EOS_GUARD_SRC_UTILS_TICKET_HH
ticket.hh, 522
EOS_GUARD_SRC_UTILS_TOP_LOOPS_HH

top-loops.hh, 523
EOS_GUARD_SRC_UTILS_VERIFY_HH
verify.hh, 524
EOS_GUARD_SRC_UTILS_WILSON_-COEFFICIENTS_HH
wilson_coefficients.hh, 527
EOS_GUARD_SRC_UTILS_WILSON_-POLYNOMIAL_HH
wilson-polynomial.hh, 526
EOS_GUARD_SRC_UTILS_WILSON_SCAN_-MODEL_HH
wilson_scan_model.hh, 528
EOS_GUARD_UTIL_WRAPPED_FORWARD_-ITERATOR_FWD_HH
wrapped_forward_iterator-fwd.hh, 529
EOS_GUARD_UTIL_WRAPPED_FORWARD_-ITERATOR_IMPL_HH
wrapped_forward_iterator-impl.hh, 530
EOS_GUARD_UTILS_WRAPPED_FORWARD_-ITERATOR_HH
wrapped_forward_iterator.hh, 531
equal
eos::proposal_functions::GlobalLocal::Config, 157
eos::WrappedForwardIterator::Base, 69
eos::WrappedForwardIterator::BaseImpl, 71
equal_weight_components
eos::proposal_functions::GlobalLocal::Config, 157
equal_weights
eos::HierarchicalClustering::Config, 161
Equation
eos::EquationSolver, 181
equation_solver.hh
EOS_GUARD_EOS_UTILS_EQUATION_-SOLVER_HH, 472
EquationSolver
eos::EquationSolver, 181
estimate_cumulative_distribution
eos, 28
eta3pi
eos::PionLCDAs, 340
evaluate
eos::AbsoluteTestObservable, 51
eos::ConcreteObservable, 134
eos::FormFactorAdapter, 189
eos::FormFactorRatioAdapter, 195
eos::LogLikelihoodBlock, 251
eos::MarkovChain::ProposalFunction, 350
eos::Observable, 300
eos::proposal_-functions::BlockDecomposition, 80
eos::proposal_functions::GlobalLocal, 206
eos::proposal_-functions::MultivariateGaussian, 289
eos::proposal_-functions::MultivariateStudentT, 292
eos::TestObservable, 387
evidence
eos::PopulationMonteCarloSampler::Status, 385
evolve
eos, 28
evolve_gegenbauer_moment
eos::LCDA, 240
Exception
eos::Exception, 183
exception.hh
EOS_GUARD_SRC_UTILS_EXCEPTION_-HH, 473
exclusive-b-to-dilepton.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_DILEPTON_HH, 446
exclusive-b-to-s-dilepton-large-recoil.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_DILEPTON_-LARGE_RECOIL_HH, 447
exclusive-b-to-s-dilepton-low-recoil.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_DILEPTON_-LOW_RECOIL_HH, 448
exclusive-b-to-s-dilepton.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_DILEPTON_-HH, 449
exclusive-b-to-s-gamma.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-EXCLUSIVE_B_TO_S_GAMMA_HH, 450
Exists
eos::hdf5::File, 186

F0
eos::TopLoops, 398
F1
eos::TopLoops, 398
F17_massive
eos::CharmLoops, 120
F17_massless
eos::CharmLoops, 120
F19_massive
eos::CharmLoops, 120
F19_massless
eos::CharmLoops, 120
F27_massive
eos::CharmLoops, 120

F27_massless
eos::CharmLoops, 120
F29_massive
eos::CharmLoops, 120
F29_massless
eos::CharmLoops, 120
f3pi
eos::PionLCDAs, 340
F87_massless
eos::CharmLoops, 120
F89_massless
eos::CharmLoops, 120
f_0
eos::AnalyticFormFactorBToPiDKMMO2008, 62
eos::BCL2008FormFactors, 73
eos::BFW2010FormFactors< PToP >, 74
eos::BZ2004FormFactors< Process_, PToP >, 112
eos::BZ2004FormFactorsSplit, 115
eos::FormFactors< PToP >, 199
eos::KMPW2010FormFactors< PToP >, 232
F_A
eos::BToKDilepton< LargeRecoil >, 89
F_lo_tw2
eos::AnalyticFormFactorBToPiDKMMO2008, 62
F_lo_tw3
eos::AnalyticFormFactorBToPiDKMMO2008, 63
F_lo_tw4
eos::AnalyticFormFactorBToPiDKMMO2008, 63
f_long_a
eos::FormFactors< OneHalfPlusToOneHalfPlus >, 198
f_long_v
eos::FormFactors< OneHalfPlusToOneHalfPlus >, 198
F_nlo_tw2
eos::AnalyticFormFactorBToPiDKMMO2008, 63
F_nlo_tw3
eos::AnalyticFormFactorBToPiDKMMO2008, 63
F_P
eos::BToKDilepton< LargeRecoil >, 89
f_p
eos::AnalyticFormFactorBToPiDKMMO2008, 63
eos::BCL2008FormFactors, 73
eos::BFW2010FormFactors< PToP >, 74
eos::BZ2004FormFactors< Process_, PToP >, 112
eos::BZ2004FormFactorsSplit, 115
eos::FormFactors< PToP >, 199
eos::KMPW2010FormFactors< PToP >, 232
f_p_d1
eos::FormFactors< PToP >, 199
f_p_d2
eos::FormFactors< PToP >, 199
f_perp_a
eos::FormFactors< OneHalfPlusToOneHalfPlus >, 198
f_perp_v
eos::FormFactors< OneHalfPlusToOneHalfPlus >, 198
F_S
eos::BToKDilepton< LargeRecoil >, 89
F_T
eos::BToKDilepton< LargeRecoil >, 89
f_t
eos::AnalyticFormFactorBToPiDKMMO2008, 63
eos::BCL2008FormFactors, 73
eos::BFW2010FormFactors< PToP >, 74
eos::BZ2004FormFactors< Process_, PToP >, 112
eos::BZ2004FormFactorsSplit, 115
eos::FormFactors< PToP >, 199
eos::KMPW2010FormFactors< PToP >, 232
F_T5
eos::BToKDilepton< LargeRecoil >, 89
F_V
eos::BToKDilepton< LargeRecoil >, 89
File
eos::hdf5::DataSetHandle, 168
eos::hdf5::FileHandle, 188
final_chunk_size
eos::PopulationMonteCarloSampler::Config, 149
find
eos::Histogram< 2 >, 217
find_modes
eos::MarkovChainSampler::Config, 141
fix_flat_nuisance
eos::Analysis::OptimizationOptions, 321
Flat
eos::LogPrior, 258
form-factor-adapter.hh
EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTOR_ADAPTER_HH, 435
form-factors-fwd.hh
EOS_GUARD_EOS_FORM_FACTORS_FORM_FACTORS_FWD_HH, 436
form-factors.hh

EOS_GUARD_SRC_FORM_FACTORS_-FORM_FACTORS_HH, 437
FormFactorAdapter
eos::FormFactorAdapter, 189
FormFactorAdapterFactory
eos::FormFactorAdapterFactory, 191
FormFactorRatioAdapter
eos::FormFactorRatioAdapter, 195
FormFactorRatioAdapterFactory
eos::FormFactorRatioAdapterFactory, 197
four_differential_decay_width
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
Function
eos::ChiSquared, 122
eos::Thread, 389
FunctionType
eos::Memoiser, 274

G1
eos::TopLoops, 398
G_0
eos::Bremsstrahlung, 84
g_had_ccbar
eos::LongDistance, 262
G_m1
eos::Bremsstrahlung, 84
gamma_m_nf_3
eos::QCD, 356
gamma_m_nf_4
eos::QCD, 356
gamma_m_nf_5
eos::QCD, 356
Gauss
eos::LogPrior, 259
Gaussian
eos::LogLikelihoodBlock, 252
gelman_rubin
eos::RValue, 366
get
eos::Options, 324
get_log_level
eos::Log, 242
global_local_config
eos::MarkovChainSampler::Config, 141
GlobalLocal
eos::proposal_functions::GlobalLocal, 204, 205
goodness_of_fit
eos::Analysis, 58
group_by_r_value
eos::PopulationMonteCarloSampler::Config, 149
group_exists
eos::hdf5::File, 187

h
eos::CharmLoops, 120
hard-scattering.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-HARD_SCATTERING_HH, 451
has
eos::Options, 324
hdf5-fwd.hh
EOS_GUARD_SRC_UTILS_HDF5_FWD_-HH, 474
hdf5.hh
EOS_GUARD_SRC_UTILS_HDF5_HH, 476
hdf5::File
eos::hdf5::DataSet, 166
HDF5Error
eos::HDF5Error, 209
Helicity
eos, 27
hid_t
eos::hdf5, 37
hierarchical-clustering.hh
EOS_GUARD_SRC_UTILS_-HIERARCHICAL_CLUSTERING_HH, 477
HierarchicalClustering
eos::HierarchicalClustering, 211
Histogram
eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
histogram.hh
EOS_GUARD_SRC_UTILS_-HISTROGRAM_HH, 478
history
eos::MarkovChain, 266
history_point_weighting
eos::proposal_functions::GlobalLocal::Config, 157
history_points
eos::proposal_functions::GlobalLocal::Config, 157
history_points_local_covariance_size
eos::proposal_functions::GlobalLocal::Config, 157
history_points_ordered
eos::proposal_functions::GlobalLocal::Config, 158
history_states
eos::proposal_functions::GlobalLocal, 206
history_type
eos::proposal_functions::GlobalLocal, 206
HistoryPointWeighting

eos::proposal_functions::GlobalLocal::Config, 156
HistoryPtr
eos, 27
HistoryType
eos::proposal_functions::GlobalLocal, 204
hyper_parameter
eos::MarkovChain, 266
eos::MarkovChain::State, 382

i
eos::WrappedForwardIterator::BaseImpl, 72
I1
eos::HardScattering, 208
Id
eos::ObservableCache, 302
eos::Parameter, 330
id
eos::hdf5::FileHandle, 188
eos::Parameter, 330
ignore_eff_sample_size
eos::PopulationMonteCarloSampler::Config, 150
ignore_groups
eos::PopulationMonteCarloSampler::Config, 150
ignore_record
eos::PopulationMonteCarloSampler::Output, 326
ignore_type
eos::PopulationMonteCarloSampler::Output, 326
IgnoreType
eos::PopulationMonteCarloSampler::Output, 326
imag_c7eff
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LowRecoil >, 102
imag_c9eff
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LowRecoil >, 102
imag_y
eos::BToKstarDilepton< LowRecoil >, 102
Implementation< Analysis >
eos::Analysis, 61
Implementation< Kinematics >
eos::KinematicVariable, 231
Implementation< Parameter >
eos::Parameters, 336
Implementation< Parameters >
eos::Parameter, 331
eos::Parameters, 336
inclusive-b-to-s-dilepton.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-INCLUSIVE_B_TO_S_DILEPTON_-HH, 452
inclusive-b-to-s-gamma.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-INCLUSIVE_B_TO_S_GAMMA_HH, 453
increment
eos::WrappedForwardIterator::Base, 69
eos::WrappedForwardIterator::BaseImpl, 71
IndexIterator
eos::Cluster, 126
indices
eos::proposal_functions::AdjacencyMatrix, 54
initial_guess
eos::HierarchicalClustering, 212
initial_step_size
eos::Analysis::OptimizationOptions, 321
insert
eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
instance
eos::InstantiationPolicy< T_, Singleton >, 225
eos::ThreadPool, 391
instantiation_policy-impl.hh
_ptr, 479
EOS_GUARD_SRC_UTILS_-INSTANTIATION_POLICY_IMPL_HH, 479
instantiation_policy.hh
EOS_GUARD_SRC_UTILS_-INSTANTIATION_POLICY_HH, 480
InstantiationPolicy
eos::InstantiationPolicy< T_, NonCopyable >, 223
eos::InstantiationPolicy< T_, Singleton >, 225
InstantiationPolicy< Log, Singleton >
eos::Log, 243
integrate
eos, 29
integrate-impl.hh
EOS_GUARD_EOS_UTILS_INTEGRATE_-IMPL_HH, 481
integrate.hh
EOS_GUARD_SRC_UTILS_INTEGRATE_-HH, 482
integrated_a_9
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_a_fb_combined
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236

eos::LambdaBToLambdaDilepton< LowRecoil >, 238
integrated_a_fb_hadronic
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
integrated_a_fb_leptonic
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
integrated_branching_ratio
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
eos::BToPiLeptonNeutrino, 107
eos::BToXsDilepton< HLMW2005 >, 109
eos::BToXsGamma< Minimal >, 110
eos::BToXsGamma< NLO >, 111
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
integrated_branching_ratio_cp_averaged
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_cp_asymmetry
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_cp_asymmetry_1
eos::BToKstarDilepton< LowRecoil >, 102
integrated_cp_asymmetry_2
eos::BToKstarDilepton< LowRecoil >, 102
integrated_cp_asymmetry_3
eos::BToKstarDilepton< LowRecoil >, 102
integrated_cp_summed_decay_width
eos::BToKstarDilepton< LowRecoil >, 102
integrated_decay_width
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_flat_term
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
integrated_flat_term_cp_averaged
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
integrated_forward_backward_asymmetry
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_forward_backward_asymmetry_cp_averaged
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_forward_backward_asymmetry_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_fzero
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
integrated_h_1
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_h_1_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_h_2
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_h_2_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_h_3
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_h_3_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_h_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_h_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_isospin_asymmetry
eos::BToKstarDilepton< LargeRecoil >, 97
integrated_j_1c
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_1s
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_2c
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_2s
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_3

eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_3_normalized
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_3_normalized_cp_averaged
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_6c
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_6s
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_7
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_8
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_9
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_9_normalized
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_j_9_normalized_cp_averaged
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_longitudinal_polarisation
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_longitudinal_polarisation_cp_averaged
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_longitudinal_polarisation_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_p_prime_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_p_prime_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_p_prime_6
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_ratio_muons_electrons
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
integrated_transversal_polarisation
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transversal_polarisation_cp_averaged
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_2
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_2_cp_averaged
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_2_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_3
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_3_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_4
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_4_naive
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_5
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_im
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_transverse_asymmetry_re
eos::BToKstarDilepton< LargeRecoil >, 97
eos::BToKstarDilepton< LowRecoil >, 102
integrated_unnormalized_cp_asymmetry_1
eos::BToKstarDilepton< LowRecoil >, 102
integrated_unnormalized_forward_backward_-asymmetry
eos::BToKstarDilepton< LowRecoil >, 102
integrated_zeta
eos::BToPiLeptonNeutrino, 107
InternalError
eos::InternalError, 226
inverse_covariance
eos::HierarchicalClustering::Component, 129
isospin_asymmetry
eos::BToKstarGamma, 105
itau_22
eos::Bremsstrahlung, 84
itau_27
eos::Bremsstrahlung, 84
itau_28
eos::Bremsstrahlung, 84
itau_29

eos::Bremsstrahlung, 84
iterations
eos::MarkovChainSampler::PreRunInfo, 344
iterations_accepted
eos::MarkovChain::Stats, 383
iterations_at_convergence
eos::MarkovChainSampler::PreRunInfo, 344
eos::PopulationMonteCarloSampler::Status, 386
iterations_invalid
eos::MarkovChain::Stats, 383
iterations_last_run
eos::MarkovChain, 266
iterations_rejected
eos::MarkovChain::Stats, 383
iterations_total
eos::MarkovChain::Stats, 383
Iterator
eos::CartesianProduct, 116
eos::Cluster, 126
eos::Diagnostics, 177
eos::Histogram< 2 >, 217
eos::LogPrior, 257
eos::MarkovChain::State, 382
eos::ObservableSet, 307
eos::Parameters, 335
iterator_category
eos::WrappedForwardIterator, 427

j0_parallel
eos::QCDFIntegrals::Results, 365
j0_perp
eos::QCDFIntegrals::Results, 365
j0bar_perp
eos::QCDFIntegrals::Results, 365
j1_parallel
eos::QCDFIntegrals::Results, 365
j1_perp
eos::QCDFIntegrals::Results, 365
j1c
eos::btovll::AngularCoefficients, 64
j1s
eos::btovll::AngularCoefficients, 64
j2_perp
eos::QCDFIntegrals::Results, 365
j2c
eos::btovll::AngularCoefficients, 64
j2s
eos::btovll::AngularCoefficients, 64
j3
eos::btovll::AngularCoefficients, 64
j3_parallel
eos::QCDFIntegrals::Results, 365
j4
eos::btovll::AngularCoefficients, 64
j4_parallel
eos::QCDFIntegrals::Results, 365
j4_perp
eos::QCDFIntegrals::Results, 365
j5
eos::btovll::AngularCoefficients, 64
j5_perp
eos::QCDFIntegrals::Results, 365
j6_perp
eos::QCDFIntegrals::Results, 365
j6c
eos::btovll::AngularCoefficients, 64
j6s
eos::btovll::AngularCoefficients, 64
j7
eos::btovll::AngularCoefficients, 64
j7_perp
eos::QCDFIntegrals::Results, 365
j8
eos::btovll::AngularCoefficients, 64
j9
eos::btovll::AngularCoefficients, 64
join_chains_symmetrically
eos::proposal_functions::GlobalLocal::Config, 158
jtilde1_perp
eos::QCDFIntegrals::Results, 365
jtilde2_parallel
eos::QCDFIntegrals::Results, 365
jump
eos::proposal_functions::AdjacencyMatrix, 54
jump_type
eos::proposal_functions::GlobalLocal, 206
JumpType
eos::proposal_functions::GlobalLocal, 204

k
eos::TestObservable, 388
keep
eos::MarkovChain::History, 220
keep_history
eos::MarkovChain, 266
KeyType
eos::Memoiser, 274
kill_clusters
eos::HierarchicalClustering::Config, 161
kinematic.hh
EOS_GUARD_SRC_UTILS_KINEMATIC_-HH, 483
Kinematics
eos::Kinematics, 227
eos::KinematicVariable, 231
kinematics

eos::ConcreteObservable, 134
eos::FormFactorAdapter, 189
eos::FormFactorRatioAdapter, 195
eos::Observable, 300
eos::TestObservable, 388
KMPW2010FormFactors
eos::KMPW2010FormFactors< PToP >, 232
eos::KMPW2010FormFactors< PToV >, 233

label
eos::BToK, 87
eos::BToPi, 106
lambda
eos, 29
lambda-b-to-lambda-dilepton.hh
EOS_GUARD_EOS_RARE_B_DECAYS_-BARYONIC_B_TO_S_DILEPTON_HH, 454
LambdaBToLambdaDilepton
eos::LambdaBToLambdaDilepton< LargeRecoil >, 236
eos::LambdaBToLambdaDilepton< LowRecoil >, 238
lcda.hh
EOS_GUARD_SRC_UTILS_LCDA_HH, 484
left_handed
eos, 27
ll_debug
eos, 27
ll_error
eos, 27
ll_informational
eos, 27
ll_last
eos, 27
ll_silent
eos, 27
ll_warning
eos, 27
local_covariance_type
eos::proposal_functions::GlobalLocal, 206
local_covariances
eos::proposal_functions::GlobalLocal, 206
local_jump_probability
eos::proposal_functions::GlobalLocal::Config, 158
local_mode
eos::MarkovChain::History, 219
LocalCovarianceType
eos::proposal_functions::GlobalLocal, 204
Lock
eos::Lock, 241
lock.hh
EOS_GUARD_SRC_UTILS_LOCK_HH, 485
log.hh
EOS_GUARD_SRC_UTILS_LOG_HH, 486
Log::message
eos::LogMessageHandler, 255
log_posterior
eos::proposal_functions::GlobalLocal::Config, 157
log_likelihood
eos::Analysis, 59
eos::MarkovChain::State, 382
log_likelihood-fwd.hh
EOS_GUARD_SRC_UTILS_LOG_-LIKELIHOOD_FWD_HH, 487
log_likelihood.hh
EOS_GUARD_SRC_UTILS_-LIKELIHOOD_HH, 488
log_posterior
eos::Analysis, 59
eos::MarkovChain::State, 382
log_prior
eos::Analysis, 59
eos::MarkovChain::State, 382
log_prior-fwd.hh
EOS_GUARD_SRC_UTILS_PRIOR_FWD_-HH, 489
log_prior.hh
EOS_GUARD_SRC_UTILS_PRIOR_HH, 490
LogGamma
eos::LogLikelihoodBlock, 252
eos::LogPrior, 259
LogLevel
eos, 27
LogLikelihood
eos::LogLikelihood, 245
LogLikelihoodBlockPtr
eos, 27
LogMessageHandler
eos::Log, 243
LogPrior
eos::LogPrior, 257
LogPriorPtr
eos, 27
long-distance.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-LONG_DISTANCE_HH, 455
long_jump_indices
eos::proposal_functions::GlobalLocal::Config, 158
lower
eos::Histogram< 1 >::Bin, 78
eos::Histogram< 2 >::Bin, 76

m2_Br0m
eos::BToK, 87
eos::BToPi, 106
m2_Br1m
eos::BToK, 87
eos::BToPi, 106
m_B
eos::BToK, 87
eos::BToPi, 106
m_b_kin
eos::ModelComponent< components::QCD >, 280
eos::SMComponent< components::QCD >, 377
m_b_msbar
eos::ModelComponent< components::QCD >, 280
eos::SMComponent< components::QCD >, 377
m_b_pole
eos::ModelComponent< components::QCD >, 280
eos::SMComponent< components::QCD >, 377
m_b_ps
eos::ModelComponent< components::QCD >, 280
eos::SMComponent< components::QCD >, 378
m_c_kin
eos::ModelComponent< components::QCD >, 281
eos::SMComponent< components::QCD >, 378
m_c_msbar
eos::ModelComponent< components::QCD >, 281
eos::SMComponent< components::QCD >, 378
m_c_pole
eos::ModelComponent< components::QCD >, 281
eos::SMComponent< components::QCD >, 378
m_P
eos::BToK, 87
eos::BToPi, 106
m_q_kin
eos::QCD, 354
m_q_msbar
eos::QCD, 354
m_q_pole
eos::QCD, 355
m_q_ps
eos::QCD, 355
m_s_msbar
eos::ModelComponent< components::QCD >, 281
eos::SMComponent< components::QCD >, 378
m_t_msbar
eos::ModelComponent< components::QCD >, 281
eos::SMComponent< components::QCD >, 378
m_t_pole
eos::ModelComponent< components::QCD >, 281
eos::SMComponent< components::QCD >, 378
m_ud_msbar
eos::ModelComponent< components::QCD >, 281
eos::SMComponent< components::QCD >, 378
Make
eos::LogPrior, 260
make
eos::AnalyticFormFactorBToPiDKMMO2008, 63
eos::BCL2008FormFactors, 73
eos::BFW2010FormFactors< PToP >, 74
eos::BZ2004FormFactors< Process_, PToP >, 112
eos::BZ2004FormFactors< Process_, PToV >, 113
eos::BZ2004FormFactorsSplit, 115
eos::ChiSquared, 122
eos::CKMScanModel, 125
eos::ConcreteObservableFactory, 136
eos::Constraint, 163
eos::FormFactorAdapterFactory, 191
eos::FormFactorRatioAdapterFactory, 197
eos::KMPW2010FormFactors< PToP >, 232
eos::KMPW2010FormFactors< PToV >, 233
eos::Model, 276
eos::Observable, 300
eos::ObservableFactory, 305
eos::proposal_functions::Factory, 184
eos::StandardModel, 380
eos::WilsonScanModel, 423
make_analysis
eos, 30
make_concrete_observable_factory
eos, 30
make_polynomial
eos, 30
make_polynomial_ht_like_ratio

eos, 30
make_polynomial_observable
eos, 30
make_polynomial_ratio
eos, 30
MapIterator
eos::HierarchicalClustering, 211
mark
eos::Ticket, 392
markov_chain.hh
EOS_GUARD_SRC_UTILS_MARKOV_-CHAIN_HH, 491
markov_chain_sampler.hh
EOS_GUARD_SRC_UTILS_MC_-SAMPLER_HH, 492
MarkovChain
eos::MarkovChain, 265
MarkovChainSampler
eos::MarkovChainSampler, 270
mass
eos::TestObservable, 388
mass_name
eos::TestObservable, 388
mass_splitting_j1_j0
eos::BMesonProperties, 82
massive_mode_finding
eos::MarkovChainSampler, 271
matrix.hh
EOS_GUARD_SRC_UTILS_MATRIX_HH, 494
max
eos::Parameter, 331
eos::ParameterDescription, 332
eos::ParameterRange, 333
eos::RandomNumberGenerator, 360
eos::VerifiedRange, 409
max_efficiency
eos::MarkovChainSampler::Config, 141
maximum_iterations
eos::Analysis::OptimizationOptions, 321
maximum_relative_std_deviation
eos::PopulationMonteCarloSampler::Config, 150
maximum_steps
eos::EquationSolver::Config, 145
eos::HierarchicalClustering::Config, 161
mB
eos::BsToPhi, 85
eos::BToKstar, 94
MB_lcsr
eos::AnalyticFormFactorBToPiDKMMO2008, 63
MB_svz
eos::AnalyticFormFactorBToPiDKMMO2008, 63
mcmc_pre_run
eos::Analysis::OptimizationOptions, 321
mean
eos::Cluster, 127
eos::HierarchicalClustering::Component, 129
eos::LogPrior, 260
eos::Welford, 414
mean_and_covariance
eos::MarkovChain::History, 219
mean_and_variance
eos::MarkovChain::History, 219
mean_of_parameters
eos::MarkovChain::Stats, 384
mean_of_posterior
eos::MarkovChain::Stats, 384
means
eos::Cluster, 127
MemoisationControl
eos::MemoisationControl, 273
memoise
eos, 30
memoise.hh
EOS_GUARD_SRC_UTILS_MEMOISE_-HH, 495
Memoiser
eos::Memoiser, 274
mesonic-impl.hh
EOS_GUARD_SRC_RARE_B_DECAYS_-FORM_FACTORS_IMPL_HH, 438
mesonic.hh
EOS_GUARD_SRC_FORM_FACTORS_-MESONIC_HH, 439
message
eos::Log, 242
meta_record
eos::proposal_functions, 43
meta_type
eos::proposal_functions, 43
MetaRecord
eos::proposal_functions, 42
MetaType
eos::proposal_functions, 42
min
eos::Parameter, 331
eos::ParameterDescription, 332
eos::ParameterRange, 333
eos::RandomNumberGenerator, 360
eos::VerifiedRange, 409
min_efficiency
eos::MarkovChainSampler::Config, 141
minimum_eff_sample_size

eos::PopulationMonteCarloSampler::Config, 150
minimum_overlap
eos::PopulationMonteCarloSampler::Config, 150
minimum_perplexity
eos::PopulationMonteCarloSampler::Config, 150
minimum_relative_cluster_weight
eos::proposal_functions::GlobalLocal::Config, 158
minimum_steps
eos::PopulationMonteCarloSampler::Config, 150
Mixture
eos::LogLikelihoodBlock, 253
MixtureDensity
eos::HierarchicalClustering, 211
mode
eos::proposal_functions::GlobalLocal, 206
mode_distance
eos::PopulationMonteCarloSampler::Config, 150
mode_of_posterior
eos::MarkovChain::Stats, 384
model.hh
EOS_GUARD_SRC_UTILS_MODEL_HH, 496
mult
eos, 31
Multivariate
eos::proposal_functions::Multivariate, 283
MultivariateAccess
eos::proposal_-functions::BlockDecomposition, 81
MultivariateGaussian
eos::LogLikelihoodBlock, 253
eos::proposal_-functions::MultivariateGaussian, 289
MultivariateProposalPtr
eos::proposal_functions, 42
MultivariateStudentT
eos::proposal_-functions::MultivariateStudentT, 292
mupi
eos::PionLCDAs, 340
mutable-fwd.hh
EOS_GUARD_EOS_UTILS_MUTABLE_-FWD_HH, 497
mutable.hh
EOS_GUARD_EOS_UTILS_MUTABLE_-HH, 498
MutablePtr
eos, 27
Mutex
eos::Mutex, 296
mutex
eos::Mutex, 296
mutex.hh
EOS_GUARD_SRC_UTILS_MUTEX_HH, 499
mV
eos::BsToPhi, 85
eos::BToKstar, 94

n
eos::TestObservable, 388
n_samples
eos::PriorSampler::Config, 154
n_workers
eos::PriorSampler::Config, 154
name
eos::ConcreteObservable, 135
eos::Constraint, 163
eos::FormFactorAdapter, 190
eos::FormFactorRatioAdapter, 196
eos::hdf5::Array, 65
eos::hdf5::Composite, 130
eos::hdf5::DataSet, 165
eos::hdf5::File, 187
eos::hdf5::Scalar, 368
eos::hdf5::Type, 401
eos::Mutable, 295
eos::Observable, 301
eos::Parameter, 331
eos::TestObservable, 388
need_main_run
eos::MarkovChainSampler::Config, 141
need_prerun
eos::MarkovChainSampler::Config, 141
eos::PopulationMonteCarloSampler::Config, 151
norm
eos::proposal_functions::Multivariate, 286
normalized_density
eos::MarkovChain, 266
NoSuchChiSquaredError
eos::NoSuchChiSquaredError, 298
NoSuchModelError
eos::NoSuchModelError, 299
nuisance
eos::Analysis, 59
eos::ParameterDescription, 332
number_of_chains
eos::MarkovChainSampler::Config, 141
number_of_clusters
eos::proposal_functions::AdjacencyMatrix, 55
number_of_elements

eos::Welford, 414
number_of_memoisations
eos, 31
eos::Memoiser, 274
number_of_objects
eos::hdf5::File, 187
number_of_observations
eos::LogLikelihood, 246
eos::LogLikelihoodBlock, 254
number_of_threads
eos::ThreadPool, 391
number_of_workers
eos::PopulationMonteCarloSampler::Config, 151

o
eos::TestObservable, 388
observable
eos::ObservableCache, 303
observable.hh
EOS_GUARD_SRC_UTILS_-OBSERVABLE_HH, 441
observable_cache
eos::LogLikelihood, 246
observable_cache.hh
EOS_GUARD_SRC_UTILS_-OBSERVABLE_CACHE_HH, 500
observable_set.hh
EOS_GUARD_SRC_UTILS_-OBSERVABLE_SET_HH, 501
ObservableCache
eos::ObservableCache, 303
ObservableFactory
eos::ObservableFactory, 305
ObservableIterator
eos::Constraint, 163
ObservableNameError
eos::ObservableNameError, 306
ObservablePtr
eos, 27
observables_type
eos::PriorSampler, 347
ObservableSet
eos::ObservableSet, 308
ObservablesType
eos::PriorSampler, 346
omega3pi
eos::PionLCDAs, 340
omega4pi
eos::PionLCDAs, 340
omegaem_1010
eos::EMContributions, 179
omegaem_22
eos::EMContributions, 179
omegaem_27
eos::EMContributions, 179
omegaem_29
eos::EMContributions, 179
omegaem_77
eos::EMContributions, 179
omegaem_79
eos::EMContributions, 179
omegaem_99
eos::EMContributions, 179
one-of.hh
EOS_GUARD_SRC_UTILS_ONE_OF_HH, 502
OneOf
eos::OneOf, 311
OneOfValue
eos::OneOfValue, 312
OneOfVisitorWrapper
eos::OneOfVisitorWrapper, 316
OneOfVisitorWrapperVisit
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >, 317
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>, 318
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >, 319
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>, 320
Open
eos::hdf5::File, 187
open_attribute
eos::hdf5::DataSet, 165
eos::hdf5::DataSetHandle, 168
open_data_set
eos::hdf5::File, 187
operator double
eos::KinematicVariable, 230
eos::Mutable, 295
eos::Parameter, 331
operator T_
eos::VerifiedRange, 410
operator<
eos::Histogram< 1 >::Bin, 78
eos::Histogram< 2 >::Bin, 76
operator<<
eos, 32
eos::hdf5, 37
eos::hdf5::DataSet, 166
eos::LogMessageHandler, 255
operator>>
eos, 32
eos::hdf5, 38
eos::hdf5::DataSet, 166

operator*
eos, 31
eos::WrappedForwardIterator, 427
operator^
eos, 32
operator()
eos::KinematicVariable, 230
eos::LogLikelihood, 246
eos::LogPrior, 260
eos::Memoiser, 274
eos::Mutable, 295
eos::Parameter, 331
eos::RandomNumberGenerator, 361
eos::TryLock, 399
operator+
eos, 31, 32
eos::Options, 324
operator++
eos::WrappedForwardIterator, 427
operator+=
eos::Accumulator, 53
operator-
eos, 32
operator->
eos::WrappedForwardIterator, 427
operator=
eos::hdf5::Attribute, 67
eos::KinematicVariable, 230
eos::Mutable, 295
eos::Parameter, 331
eos::VerifiedRange, 410
eos::WrappedForwardIterator, 427
operator==
eos::Kinematics, 228
eos::Options, 324
eos::WrappedForwardIterator, 427
optimize
eos::Analysis, 59
optimize_minuit
eos::Analysis, 60
Options
eos::Options, 323
options
eos::ConcreteObservable, 135
eos::FormFactorAdapter, 190
eos::FormFactorRatioAdapter, 196
eos::Observable, 301
eos::TestObservable, 388
options.hh
EOS_GUARD_SRC_UTILS_OPTIONS_HH, 503
output_file
eos::MarkovChainSampler::Config, 142
eos::PopulationMonteCarloSampler::Config, 151
eos::PriorSampler::Config, 154
over
eos::CartesianProduct, 117
overlaps
eos::Cluster, 127
override_global_local_proposal
eos::PopulationMonteCarloSampler::Config, 151

p
eos::TestObservable, 388
parallelize
eos::MarkovChainSampler::Config, 142
eos::PopulationMonteCarloSampler::Config, 151
eos::PriorSampler::Config, 155
Parameter
eos::Parameter, 330
eos::Parameters, 336
parameter
eos::ParameterDescription, 332
parameter_descriptions
eos::Analysis, 60
eos::MarkovChain, 267
parameter_indices
eos::Cluster, 127
Parameters
eos::Parameter, 331
parameters
eos::Analysis, 60
eos::ConcreteObservable, 135
eos::EquationSolver::Solution, 379
eos::FormFactorAdapter, 190
eos::FormFactorRatioAdapter, 196
eos::LogLikelihood, 247
eos::Observable, 301
eos::ObservableCache, 303
eos::ObservableSet, 308
eos::TestObservable, 388
parameters-fwd.hh
EOS_GUARD_EOS_UTILS_-PARAMETERS_FWD_HH, 504
parameters.hh
EOS_GUARD_SRC_UTILS_-PARAMETERS_HH, 505
parameters_at_mode
eos::MarkovChain::Stats, 384
partitions
eos::MarkovChainSampler::Config, 142
patch_around_local_mode
eos::PopulationMonteCarloSampler::Config, 151

perform_clustering
eos::proposal_functions::GlobalLocal::Config, 158
perplexity
eos::PopulationMonteCarloSampler::Status, 386
phi
eos::PionLCDAs, 340
phi3p
eos::PionLCDAs, 340
phi3s
eos::PionLCDAs, 340
phi3s_d1
eos::PionLCDAs, 340
phi4
eos::PionLCDAs, 340
phi4_d1
eos::PionLCDAs, 340
phi4_d2
eos::PionLCDAs, 340
photon_bottom_case
eos::QCDFIntegrals, 358
photon_charm_case
eos::QCDFIntegrals, 359
photon_energy_moment_1
eos::BToXsGamma< NLO >, 111
photon_energy_moment_2
eos::BToXsGamma< NLO >, 111
photon_massless_case
eos::QCDFIntegrals, 359
pi-lcdas.hh
EOS_GUARD_EOS_FORM_FACTORS_PI_-LCDAS_HH, 440
PionLCDAs
eos::PionLCDAs, 340
point
eos::MarkovChain::State, 382
pointer
eos::WrappedForwardIterator, 427
eos::WrappedForwardIterator::Base, 69
eos::WrappedForwardIterator::BaseImpl, 71
polylog.hh
EOS_GUARD_EOS_UTILS_POLYLOG_-HH, 506
population_monte_carlo_sampler.hh
EOS_GUARD_SRC_UTILS_PMC_-SAMPLER_HH, 507
PopulationMonteCarloSampler
eos::PopulationMonteCarloSampler, 341, 342
posterior
eos::proposal_functions::GlobalLocal::Config, 157
power_of
eos, 32
power_of.hh
EOS_GUARD_SRC_UTILS_POWER_OF_-HH, 508
pre_run_info
eos::MarkovChainSampler, 271
precision
eos::HierarchicalClustering::Config, 161
prepare_sampling
eos::LogLikelihoodBlock, 254
prerun_chains_per_partition
eos::MarkovChainSampler::Config, 142
prerun_iterations_max
eos::MarkovChainSampler::Config, 142
prerun_iterations_min
eos::MarkovChainSampler::Config, 142
prerun_iterations_update
eos::MarkovChainSampler::Config, 142
print_steps
eos::PopulationMonteCarloSampler::Config, 151
prior_sampler.hh
EOS_GUARD_SRC_UTILS_PRIOR_-SAMPLER_HH, 509
priors_type
eos::proposal_-functions::BlockDecomposition, 81
PriorSampler
eos::PriorSampler, 346
PriorsType
eos::proposal_-functions::BlockDecomposition, 80
private_implementation_pattern-impl.hh
WILSON_FITTER_GUARD_UTILS_-PRIVATE_IMPLEMENTATION_-PATTERN_IMPL_HH, 510
private_implementation_pattern.hh
WILSON_FITTER_GUARD_UTILS_-PRIVATE_IMPLEMENTATION_-PATTERN_HH, 511
PrivateImplementationPattern
eos::PrivateImplementationPattern, 348
properties.hh
EOS_GUARD_EOS_B_DECAYS_-PROPERTIES_HH, 430
proposal
eos::MarkovChainSampler::Config, 142
proposal_accepted
eos::MarkovChain, 267
proposal_function
eos::MarkovChain, 267
proposal_functions.hh
EOS_GUARD_EOS_UTILS_PROPROSAL_-FUNCTIONS_HH, 513
ProposalFunctionPtr

eos, 27
propose
eos::MarkovChain::ProposalFunction, 350
eos::proposal_-functions::BlockDecomposition, 81
eos::proposal_functions::GlobalLocal, 206
eos::proposal_-functions::MultivariateGaussian, 289
eos::proposal_-functions::MultivariateStudentT, 292
proposed_state
eos::MarkovChain, 267
psi4
eos::PionLCDAs, 340
psi4_i
eos::PionLCDAs, 340
push_back
eos::TicketList, 394

qcd.hh
EOS_GUARD_SRC_UTILS_QCD_HH, 514
qcdf_integrals.hh
EOS_GUARD_EOS_RARE_B_DECAYS_-QCDF_INTEGRALS_HH, 456
Quick
eos::HierarchicalClustering::Config, 160
eos::MarkovChainSampler::Config, 140
eos::PopulationMonteCarloSampler::Config, 148

r_value_no_nuisance
eos::PopulationMonteCarloSampler::Config, 151
random_index
eos::proposal_functions, 43
random_number_generator.hh
EOS_GUARD_SRC_UTILS_RANDOM_-NUMBER_GENERATOR_HH, 515
random_start
eos::PopulationMonteCarloSampler::Config, 151
RandomNumberGenerator
eos::RandomNumberGenerator, 360
read
eos::hdf5::AttributeHandle, 68
read_data
eos::MarkovChain, 267
read_descriptions
eos::Analysis, 60
eos::MarkovChain, 268
read_one
eos::hdf5::DataSetHandle, 168
read_only
eos::hdf5::FileHandle, 188
read_samples
eos::PopulationMonteCarloSampler, 342
real_c7eff
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LowRecoil >, 102
real_c9eff
eos::BToKDilepton< LowRecoil >, 92
eos::BToKstarDilepton< LowRecoil >, 102
real_y
eos::BToKstarDilepton< LowRecoil >, 102
records
eos::hdf5::DataSet, 165
RecordType
eos::hdf5::Attribute, 67
eos::hdf5::DataSet, 165
reference
eos::WrappedForwardIterator, 427
eos::WrappedForwardIterator::Base, 70
eos::WrappedForwardIterator::BaseImpl, 72
register_clear_function
eos::MemoisationControl, 273
rescale
eos::proposal_functions::Multivariate, 284
rescale_local_covariance
eos::proposal_functions::GlobalLocal::Config, 158
reserve
eos::proposal_functions::AdjacencyMatrix, 55
reset
eos::LogLikelihood, 247
eos::MarkovChain, 268
eos::ObservableCache, 303
eos::proposal_functions::Multivariate, 284
restrict
eos::Analysis, 60
result
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >, 317
resume
eos::MarkovChainSampler, 271
rho_1
eos::BToKstarDilepton< LowRecoil >, 102
rho_2
eos::BToKstarDilepton< LowRecoil >, 102
right_handed
eos, 27
ROOT, 44
ROOT::Minuit2, 45
run
eos::HierarchicalClustering, 212
eos::MarkovChain, 268
eos::MarkovChainSampler, 271
eos::PopulationMonteCarloSampler, 343
eos::PriorSampler, 347

rvalue.hh
EOS_GUARD_SRC_UTILS_RVALUE_HH, 516
rvalue_criterion_param
eos::MarkovChainSampler::Config, 142
rvalue_criterion_posterior
eos::MarkovChainSampler::Config, 142
rvalue_parameters
eos::MarkovChainSampler::PreRunInfo, 344
rvalue_posterior
eos::MarkovChainSampler::PreRunInfo, 344
RValueFunction
eos::Cluster, 126

s_kstar_gamma
eos::BToKstarGamma, 105
sample
eos::LogLikelihoodBlock, 254
eos::LogPrior, 261
sample_record
eos::PopulationMonteCarloSampler::Output, 326
sample_type
eos::PopulationMonteCarloSampler::Output, 326
SamplesList
eos::PriorSampler, 346
SampleType
eos::PopulationMonteCarloSampler::Output, 326
Save
eos::Save, 367
save.hh
EOS_GUARD_SRC_UTILS_SAVE_HH, 517
Scalar
eos::hdf5::Scalar, 368
scalars_type
eos::proposal_-functions::MultivariateGaussian, 289
eos::proposal_-functions::MultivariateStudentT, 292
ScalarsType
eos::proposal_-functions::MultivariateGaussian, 289
eos::proposal_-functions::MultivariateStudentT, 292
scale_automatic
eos::MarkovChainSampler::Config, 143
scale_nuisance
eos::MarkovChainSampler::Config, 143
scale_reduction
eos::MarkovChainSampler::Config, 143
seed
eos::MarkovChainSampler::Config, 143
eos::PopulationMonteCarloSampler::Config, 151
eos::PriorSampler::Config, 155
select
eos::hdf5::DataSetHandle, 168
set
eos::Kinematics, 228
eos::Options, 324
eos::Parameters, 336
set_analysis
eos::MarkovChain, 268
set_id
eos::hdf5::DataSetHandle, 168
set_index
eos::hdf5::DataSet, 165
set_indices
eos::proposal_functions::Multivariate, 285
set_log_level
eos::Log, 242
set_log_stream
eos::Log, 243
set_mode
eos::MarkovChain, 268
set_option
eos::TestObservable, 388
set_point
eos::MarkovChain, 269
set_program_name
eos::Log, 243
signal
eos::ConditionVariable, 137
significance
eos::LogLikelihoodBlock, 254
single_cluster
eos::PopulationMonteCarloSampler::Config, 152
size
eos::CartesianProduct, 117
eos::Diagnostics, 177
eos::hdf5::Array, 65
eos::hdf5::Composite, 130
eos::hdf5::DataSetHandle, 168
eos::hdf5::Scalar, 369
eos::hdf5::Type, 401
eos::ObservableCache, 304
eos::ObservableSet, 308
skip_initial
eos::MarkovChainSampler::Config, 143
eos::PopulationMonteCarloSampler::Config, 152
eos::proposal_functions::GlobalLocal::Config, 158
sliding_window

eos::PopulationMonteCarloSampler::Config, 152
eos::proposal_functions, 43
SMComponent
eos::SMComponent< components::CKM >, 374
eos::SMComponent< components::DeltaB1 >, 376
eos::SMComponent< components::QCD >, 377
solve
eos::EquationSolver, 181
space_id
eos::hdf5::DataSetHandle, 168
splitting_tolerance
eos::Analysis::OptimizationOptions, 322
standard-model.hh
EOS_GUARD_SRC_UTILS_STANDARD_-MODEL_HH, 518
StandardModel
eos::StandardModel, 380
starting_points
eos::PopulationMonteCarloSampler::Config, 152
State
eos::MarkovChain::State, 382
state
eos::proposal_functions::AdjacencyMatrix, 55
states
eos::MarkovChain::History, 220
statistics
eos::MarkovChain, 269
statistics_record
eos::PopulationMonteCarloSampler::Output, 326
statistics_type
eos::PopulationMonteCarloSampler::Output, 326
StatisticsType
eos::PopulationMonteCarloSampler::Output, 326
Status
eos::PopulationMonteCarloSampler::Status, 385
status
eos::PopulationMonteCarloSampler, 343
std, 46
std_dev_reduction
eos::PopulationMonteCarloSampler::Config, 152
std_deviation
eos::Welford, 414
store
eos::MarkovChainSampler::Config, 143
eos::PopulationMonteCarloSampler::Config, 152
store_hc_initial
eos::PopulationMonteCarloSampler::Config, 152
store_input_components
eos::PopulationMonteCarloSampler::Config, 152
store_observables_and_proposals
eos::MarkovChainSampler::Config, 143
store_parameters
eos::PriorSampler::Config, 155
store_prerun
eos::MarkovChainSampler::Config, 143
eos::PopulationMonteCarloSampler::Config, 152
strategy
eos::EquationSolver::Config, 145
strategy_level
eos::Analysis::OptimizationOptions, 322
stringify
eos, 32, 33
eos::implementation::DoStringify, 178
stringify.hh
EOS_GUARD_SRC_UTILS_STRINGIFY_-HH, 519
stringify_container
eos, 33
student_t_degrees_of_freedom
eos::MarkovChainSampler::Config, 143
super_clusters
eos::PopulationMonteCarloSampler::Config, 152

tau_22
eos::Bremsstrahlung, 84
tau_27
eos::Bremsstrahlung, 84
tau_28
eos::Bremsstrahlung, 84
tau_29
eos::Bremsstrahlung, 84
tau_78
eos::Bremsstrahlung, 84
tau_88
eos::Bremsstrahlung, 84
tau_89
eos::Bremsstrahlung, 84
tau_m
eos::BToK, 87
eos::BToPi, 106
tau_p
eos::BToK, 87
eos::BToPi, 106

TestObservable
eos::TestObservable, 387
Thread
eos::Thread, 389
thread.hh
EOS_GUARD_SRC_UTILS_THREAD_HH, 520
thread_pool.hh
EOS_GUARD_SRC_UTILS_THREAD_-POOL_HH, 521
ThreadPool
eos::ThreadPool, 391
Ticket
eos::Ticket, 392
ticket.hh
EOS_GUARD_SRC_UTILS_TICKET_HH, 522
TicketList
eos::Ticket, 393
eos::TicketList, 394
tolerance
eos::Analysis::OptimizationOptions, 322
eos::EquationSolver::Config, 145
top-loops.hh
EOS_GUARD_SRC_UTILS_TOP_LOOPS_-HH, 523
Transition
eos::BToK, 87
eos::BToPi, 106
trilog
eos, 33
TryLock
eos::TryLock, 399
two_differential_decay_width
eos::BToKDilepton< LargeRecoil >, 89
eos::BToKDilepton< LowRecoil >, 92
Type
eos::hdf5::Array, 65
eos::hdf5::Composite, 130
eos::hdf5::Scalar, 368
eos::implementation::ResultOf< Result_-(∗)(Args_...)>, 362
eos::implementation::ResultOf< Result_-(Class_::∗)(Args_...)>, 363
eos::SelectOneOfType< Want_ >, 370
eos::SelectOneOfType< Want_, Try_, Rest_-...>, 371
type_id
eos::hdf5::Array, 66
eos::hdf5::AttributeHandle, 68
eos::hdf5::Composite, 131
eos::hdf5::DataSetHandle, 168
eos::hdf5::DataType, 169
eos::hdf5::DataType< const char ∗ >, 170
eos::hdf5::DataType< double >, 171
eos::hdf5::DataType< int >, 172
eos::hdf5::DataType< short >, 173
eos::hdf5::DataType< unsigned >, 174
eos::hdf5::Scalar, 369
eos::hdf5::Type, 401
TypePtr
eos::hdf5, 37

underlying
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_ >, 317
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void >, 319
underlying_iterator
eos::WrappedForwardIterator, 427
underlying_iterator_ptr
eos::WrappedForwardIterator::Base, 70
eos::WrappedForwardIterator::BaseImpl, 72
UnknownConstraintError
eos::UnknownConstraintError, 403
UnknownKinematicVariableError
eos::UnknownKinematicVariableError, 404
UnknownOptionError
eos::UnknownOptionError, 405
UnknownParameterError
eos::UnknownParameterError, 406
UnknownProposalError
eos::proposal_-functions::UnknownProposalError, 407
update
eos::ObservableCache, 304
upper
eos::Histogram< 1 >::Bin, 78
eos::Histogram< 2 >::Bin, 76
use_posterior_rvalue
eos::MarkovChainSampler::Config, 144
use_strict_rvalue_definition
eos::MarkovChainSampler::Config, 144
UsedParameter
eos::UsedParameter, 408
uses
eos::ParameterUser, 337, 338

v
eos::BZ2004FormFactors< Process_, PToV >, 114
eos::FormFactors< PToV >, 201
eos::KMPW2010FormFactors< PToV >, 234
valid
eos::EquationSolver::Solution, 379
value
eos::Accumulator, 53

eos::Diagnostics::Entry, 180
eos::EquationSolver::Solution, 379
eos::hdf5::Attribute, 67
eos::Histogram< 1 >::Bin, 78
eos::Histogram< 2 >::Bin, 76
eos::OneOfValue, 312
value_type
eos::WrappedForwardIterator, 427
variance
eos::LogPrior, 261
eos::Welford, 414
variance_of_parameters
eos::MarkovChain::Stats, 384
variance_of_posterior
eos::MarkovChain::Stats, 384
variances
eos::Cluster, 127
VerifiedRange
eos::VerifiedRange, 409
VerifiedRangeError
eos::VerifiedRangeError, 411
VerifiedRangeOverflow
eos::VerifiedRangeOverflow, 412
VerifiedRangeUnderflow
eos::VerifiedRangeUnderflow, 413
verify.hh
EOS_GUARD_SRC_UTILS_VERIFY_HH, 524
visit
eos::OneOfVisitorVisit, 315
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, Result_, Type_, Rest_...>, 318
eos::OneOfVisitorWrapperVisit< Visitor_, Underlying_, void, Type_, Rest_...>, 320
eos::WilsonPolynomialCloner, 419
eos::WilsonPolynomialEvaluator, 420
eos::WilsonPolynomialPrinter, 421

wait
eos::ConditionVariable, 137
eos::Ticket, 392
eos::TicketList, 394
wait_for_free_capacity
eos::ThreadPool, 391
weight
eos::HierarchicalClustering::Component, 129
weight_record
eos::PopulationMonteCarloSampler::Output, 326
weight_type
eos::PopulationMonteCarloSampler::Output, 326
WeightType
eos::PopulationMonteCarloSampler::Output, 326
Welford
eos::Welford, 414
welford.hh
EOS_GUARD_EOS_UTILS_WELFORD_HH, 525
what
eos::Exception, 183
wilson-polynomial.hh
EOS_GUARD_SRC_UTILS_WILSON_POLYNOMIAL_HH, 526
wilson_coefficients.hh
EOS_GUARD_SRC_UTILS_WILSON_COEFFICIENTS_HH, 527
wilson_coefficients_b_to_s
eos::ModelComponent< components::DeltaB1 >, 279
eos::SMComponent< components::DeltaB1 >, 376
eos::WilsonScanComponent, 422
WILSON_FITTER_GUARD_SRC_UTILS_COMPLEX_HH
complex.hh, 466
WILSON_FITTER_GUARD_UTILS_PRIVATE_IMPLEMENTATION_PATTERN_HH
private_implementation_pattern.hh, 511
WILSON_FITTER_GUARD_UTILS_PRIVATE_IMPLEMENTATION_PATTERN_IMPL_HH
private_implementation_pattern-impl.hh, 510
wilson_scan_model.hh
EOS_GUARD_SRC_UTILS_WILSON_SCAN_MODEL_HH, 528
WilsonCoefficients
eos::WilsonCoefficients< BToS >, 415
WilsonPolynomial
eos, 27
WilsonPolynomialCloner
eos::WilsonPolynomialCloner, 419
WilsonPolynomialPrinter
eos::WilsonPolynomialPrinter, 421
WilsonScanComponent
eos::WilsonScanComponent, 422
WilsonScanModel
eos::WilsonScanModel, 423
with_combined_uncertainties
eos::ChiSquared, 122
with_theory_offset
eos::ChiSquared, 122
WithEqualBinning
eos::Histogram< 1 >, 214
eos::Histogram< 2 >, 217
wrapped_forward_iterator-fwd.hh

EOS_GUARD_UTIL_WRAPPED_-
FORWARD_ITERATOR_FWD_HH,
529
wrapped_forward_iterator-impl.hh
EOS_GUARD_UTIL_WRAPPED_-
FORWARD_ITERATOR_IMPL_HH,
530
wrapped_forward_iterator.hh
EOS_GUARD_UTILS_WRAPPED_-
FORWARD_ITERATOR_HH, 531
WrappedForwardIterator
eos::WrappedForwardIterator, 427
write
eos::hdf5::AttributeHandle, 68
write_one
eos::hdf5::DataSetHandle, 168

xi_para
eos::BToKstarDilepton< LargeRecoil >, 97
xi_perp
eos::BToKstarDilepton< LargeRecoil >, 97