

# Sapphire

v1.0

Generated by Doxygen 1.8.5

Thu Apr 9 2020 14:00:09



# Contents

<b>1</b>	<b>README</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class List . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Namespace Documentation</b>	<b>11</b>
6.1	boost Namespace Reference . . . . .	11
6.2	boost::serialization Namespace Reference . . . . .	11
6.2.1	Function Documentation . . . . .	11
6.2.1.1	serialize . . . . .	11
6.2.1.2	serialize . . . . .	11
6.3	std Namespace Reference . . . . .	11
6.4	std::tr1 Namespace Reference . . . . .	12
<b>7</b>	<b>Class Documentation</b>	<b>13</b>
7.1	BrinkAxelGSF Class Reference . . . . .	13
7.1.1	Detailed Description . . . . .	13
7.1.2	Constructor & Destructor Documentation . . . . .	13
7.1.2.1	BrinkAxelGSF . . . . .	13
7.1.3	Member Function Documentation . . . . .	13
7.1.3.1	CalcStrengthFunction . . . . .	14
7.2	CDFEntry Class Reference . . . . .	14
7.2.1	Constructor & Destructor Documentation . . . . .	14
7.2.1.1	CDFEntry . . . . .	14
7.2.2	Member Data Documentation . . . . .	14

7.2.2.1	energy_ . . . . .	14
7.2.2.2	pairIndex_ . . . . .	14
7.2.2.3	value_ . . . . .	14
7.3	CoulFunc Class Reference . . . . .	14
7.3.1	Constructor & Destructor Documentation . . . . .	15
7.3.1.1	CoulFunc . . . . .	15
7.3.2	Member Function Documentation . . . . .	15
7.3.2.1	coulLast . . . . .	15
7.3.2.2	energyLast . . . . .	15
7.3.2.3	GSLErrorHandler . . . . .	15
7.3.2.4	lLast . . . . .	15
7.3.2.5	operator() . . . . .	15
7.3.2.6	Penetrability . . . . .	16
7.3.2.7	PEShift . . . . .	16
7.3.2.8	PEShift_dE . . . . .	16
7.3.2.9	radiusLast . . . . .	16
7.3.2.10	redmass . . . . .	16
7.3.2.11	setLast . . . . .	16
7.3.2.12	z1 . . . . .	16
7.3.2.13	z2 . . . . .	16
7.4	Coulomb_wave_functions Class Reference . . . . .	17
7.4.1	Constructor & Destructor Documentation . . . . .	17
7.4.1.1	Coulomb_wave_functions . . . . .	17
7.4.1.2	~Coulomb_wave_functions . . . . .	17
7.4.2	Member Function Documentation . . . . .	17
7.4.2.1	F_dF . . . . .	17
7.4.2.2	F_dF_init . . . . .	17
7.4.2.3	G_dG . . . . .	17
7.4.2.4	H_dH . . . . .	18
7.4.2.5	H_dH_scaled . . . . .	18
7.4.3	Member Data Documentation . . . . .	18
7.4.3.1	eta . . . . .	18
7.4.3.2	is_it_normalized . . . . .	18
7.4.3.3	l . . . . .	18
7.5	CoulWaves Struct Reference . . . . .	18
7.5.1	Member Data Documentation . . . . .	18
7.5.1.1	dF . . . . .	18
7.5.1.2	dG . . . . .	18
7.5.1.3	F . . . . .	19
7.5.1.4	G . . . . .	19

7.6	CrossSection Class Reference	19
7.6.1	Constructor & Destructor Documentation	19
7.6.1.1	CrossSection	19
7.6.2	Member Function Documentation	20
7.6.2.1	CalcAverageDWaveResWidth	20
7.6.2.2	CalcAveragePWaveResWidth	20
7.6.2.3	CalcAverageSWaveResWidth	20
7.6.2.4	Calculate	20
7.6.2.5	CalculateReactionRates	20
7.6.2.6	CreateMACSEnergiesVector	20
7.6.2.7	CreateTempVector	20
7.6.2.8	IsValid	20
7.6.2.9	PrintCrossSections	20
7.6.2.10	PrintReactionRates	20
7.6.2.11	PrintTransmissionTerms	21
7.6.2.12	SetCalculateGammaCutoff	21
7.6.2.13	SetResidualAlpha	21
7.6.2.14	SetResidualGamma	21
7.6.2.15	SetResidualNeutron	21
7.6.2.16	SetResidualProton	21
7.7	CrossSectionValues Class Reference	21
7.7.1	Constructor & Destructor Documentation	22
7.7.1.1	CrossSectionValues	22
7.7.2	Member Data Documentation	22
7.7.2.1	alpha_	22
7.7.2.2	alphaStellar_	22
7.7.2.3	gamma_	22
7.7.2.4	gammaStellar_	22
7.7.2.5	neutron_	22
7.7.2.6	neutronStellar_	22
7.7.2.7	proton_	22
7.7.2.8	protonStellar_	22
7.8	DecayController Class Reference	22
7.8.1	Constructor & Destructor Documentation	22
7.8.1.1	DecayController	22
7.8.2	Member Function Documentation	22
7.8.2.1	Decay	22
7.8.2.2	DecayProducts	23
7.8.2.3	PrintDecays	23
7.9	DecayData Class Reference	23

7.9.1	Constructor & Destructor Documentation	23
7.9.1.1	DecayData	23
7.9.1.2	DecayData	23
7.9.2	Member Function Documentation	23
7.9.2.1	alphaEntranceWidth	23
7.9.2.2	alphaTotalWidth	23
7.9.2.3	energy	23
7.9.2.4	gammaEntranceWidth	23
7.9.2.5	gammaTotalWidth	24
7.9.2.6	neutronEntranceWidth	24
7.9.2.7	neutronTotalWidth	24
7.9.2.8	protonEntranceWidth	24
7.9.2.9	protonTotalWidth	24
7.10	Decayer Class Reference	24
7.10.1	Constructor & Destructor Documentation	25
7.10.1.1	Decayer	25
7.10.1.2	~Decayer	25
7.10.2	Member Function Documentation	25
7.10.2.1	AlphaEntranceWidth	25
7.10.2.2	AlphaTotalWidth	25
7.10.2.3	CorrectWidthFluctuations	25
7.10.2.4	Decay	25
7.10.2.5	GammaEntranceWidth	25
7.10.2.6	GammaTotalWidth	25
7.10.2.7	GetMaxL	25
7.10.2.8	NeutronEntranceWidth	25
7.10.2.9	NeutronTotalWidth	26
7.10.2.10	PrintCDF	26
7.10.2.11	PrintFunctions	26
7.10.2.12	ProtonEntranceWidth	26
7.10.2.13	ProtonTotalWidth	26
7.10.2.14	SetCrossSection	26
7.10.2.15	SetMaxL	26
7.10.3	Friends And Related Function Documentation	26
7.10.3.1	CrossSection	26
7.11	DecayProduct Class Reference	26
7.11.1	Constructor & Destructor Documentation	27
7.11.1.1	DecayProduct	27
7.11.1.2	DecayProduct	27
7.11.2	Member Data Documentation	27

7.11.2.1	<a href="#">A_</a>	27
7.11.2.2	<a href="#">excitationEnergy_</a>	27
7.11.2.3	<a href="#">fragmentEnergy_</a>	27
7.11.2.4	<a href="#">fragmentEnergyCM_</a>	27
7.11.2.5	<a href="#">fragmentMomentumX_</a>	27
7.11.2.6	<a href="#">fragmentMomentumY_</a>	27
7.11.2.7	<a href="#">fragmentMomentumZ_</a>	28
7.11.2.8	<a href="#">J_</a>	28
7.11.2.9	<a href="#">particleEnergy_</a>	28
7.11.2.10	<a href="#">particleEnergyCM_</a>	28
7.11.2.11	<a href="#">particleMomentumX_</a>	28
7.11.2.12	<a href="#">particleMomentumY_</a>	28
7.11.2.13	<a href="#">particleMomentumZ_</a>	28
7.11.2.14	<a href="#">particlePhiCM_</a>	28
7.11.2.15	<a href="#">particleThetaCM_</a>	28
7.11.2.16	<a href="#">particleType_</a>	28
7.11.2.17	<a href="#">Pi_</a>	28
7.11.2.18	<a href="#">Z_</a>	28
7.12	<a href="#">DecayResults Class Reference</a>	29
7.12.1	<a href="#">Constructor &amp; Destructor Documentation</a>	29
7.12.1.1	<a href="#">DecayResults</a>	29
7.12.1.2	<a href="#">~DecayResults</a>	29
7.12.2	<a href="#">Member Function Documentation</a>	29
7.12.2.1	<a href="#">AddResults</a>	29
7.13	<a href="#">EntrancePairs Struct Reference</a>	29
7.13.1	<a href="#">Constructor &amp; Destructor Documentation</a>	29
7.13.1.1	<a href="#">EntrancePairs</a>	29
7.13.2	<a href="#">Member Data Documentation</a>	30
7.13.2.1	<a href="#">A_</a>	30
7.13.2.2	<a href="#">pType_</a>	30
7.13.2.3	<a href="#">Z_</a>	30
7.14	<a href="#">std::equal_to&lt; MassKey &gt; Struct Template Reference</a>	30
7.14.1	<a href="#">Member Function Documentation</a>	30
7.14.1.1	<a href="#">operator()</a>	30
7.15	<a href="#">EquivSquareWell Class Reference</a>	30
7.15.1	<a href="#">Constructor &amp; Destructor Documentation</a>	31
7.15.1.1	<a href="#">EquivSquareWell</a>	31
7.15.1.2	<a href="#">~EquivSquareWell</a>	31
7.15.2	<a href="#">Member Function Documentation</a>	31
7.15.2.1	<a href="#">CalcTransmission</a>	31

7.16	GammaTransition Class Reference	31
7.16.1	Constructor & Destructor Documentation	31
7.16.1.1	GammaTransition	31
7.16.2	Member Data Documentation	31
7.16.2.1	energy_	32
7.16.2.2	levelIndex_	32
7.16.2.3	probability_	32
7.17	GammaTransmissionFunc Class Reference	32
7.17.1	Constructor & Destructor Documentation	33
7.17.1.1	GammaTransmissionFunc	33
7.17.1.2	~GammaTransmissionFunc	33
7.17.2	Member Function Documentation	33
7.17.2.1	CalcStrengthFunction	33
7.17.2.2	CreateGammaTransmissionFunc	33
7.17.2.3	InitializeGDRParameters	33
7.17.2.4	IsValid	33
7.17.2.5	operator()	33
7.17.2.6	SetEGDRType	33
7.17.2.7	SetPorterThomas	33
7.17.3	Member Data Documentation	34
7.17.3.1	egdrType_	34
7.17.3.2	egqrType_	34
7.17.3.3	gdrParameters_	34
7.17.3.4	gdrTable_	34
7.17.3.5	mgdrType_	34
7.17.3.6	porterThomas_	34
7.18	GDRParameters Class Reference	34
7.18.1	Constructor & Destructor Documentation	35
7.18.1.1	GDRParameters	35
7.18.1.2	GDRParameters	35
7.18.2	Member Data Documentation	35
7.18.2.1	E_	35
7.18.2.2	eta_	35
7.18.2.3	kSigmaGamma_	35
7.18.2.4	W_	35
7.19	gsl_partfunc_params Struct Reference	35
7.19.1	Member Data Documentation	35
7.19.1.1	density	35
7.19.1.2	temperature	35
7.20	gsl_reactionrate_params Struct Reference	36



7.20.1	Member Data Documentation . . . . .	36
7.20.1.1	graph . . . . .	36
7.20.1.2	temperature . . . . .	36
7.20.1.3	useSpline . . . . .	36
7.21	std::tr1::hash< MassKey > Struct Template Reference . . . . .	36
7.21.1	Member Function Documentation . . . . .	36
7.21.1.1	operator() . . . . .	36
7.22	InitialNucleusData Class Reference . . . . .	36
7.22.1	Constructor & Destructor Documentation . . . . .	37
7.22.1.1	InitialNucleusData . . . . .	37
7.22.1.2	InitialNucleusData . . . . .	37
7.22.2	Member Function Documentation . . . . .	37
7.22.2.1	A . . . . .	37
7.22.2.2	highEnergy . . . . .	37
7.22.2.3	J . . . . .	37
7.22.2.4	lowEnergy . . . . .	37
7.22.2.5	Pi . . . . .	37
7.22.2.6	preEq . . . . .	37
7.22.2.7	Z . . . . .	37
7.22.3	Friends And Related Function Documentation . . . . .	37
7.22.3.1	boost::serialization::access . . . . .	37
7.23	int_double_pair_compare Struct Reference . . . . .	37
7.23.1	Member Function Documentation . . . . .	38
7.23.1.1	operator() . . . . .	38
7.24	JLMPotential Class Reference . . . . .	38
7.24.1	Constructor & Destructor Documentation . . . . .	38
7.24.1.1	JLMPotential . . . . .	38
7.24.2	Member Function Documentation . . . . .	39
7.24.2.1	Calculate . . . . .	39
7.24.2.2	CalculateDensity . . . . .	39
7.24.2.3	GetA . . . . .	39
7.25	KopeckyUhlGSF Class Reference . . . . .	39
7.25.1	Constructor & Destructor Documentation . . . . .	39
7.25.1.1	KopeckyUhlGSF . . . . .	39
7.25.2	Member Function Documentation . . . . .	40
7.25.2.1	CalcStrengthFunction . . . . .	40
7.26	Level Class Reference . . . . .	40
7.26.1	Constructor & Destructor Documentation . . . . .	40
7.26.1.1	Level . . . . .	40
7.26.2	Member Data Documentation . . . . .	40

7.26.2.1	energy_ . . . . .	40
7.26.2.2	gammas_ . . . . .	40
7.26.2.3	J_ . . . . .	40
7.26.2.4	Pi_ . . . . .	40
7.27	LevelDensity Class Reference . . . . .	41
7.27.1	Constructor & Destructor Documentation . . . . .	41
7.27.1.1	LevelDensity . . . . .	41
7.27.1.2	~LevelDensity . . . . .	42
7.27.2	Member Function Documentation . . . . .	42
7.27.2.1	CalcBackShift . . . . .	42
7.27.2.2	CalcConstantTempTerms . . . . .	42
7.27.2.3	CalcDensityParam . . . . .	42
7.27.2.4	CalcNuclearTemp . . . . .	42
7.27.2.5	operator() . . . . .	42
7.27.2.6	TotalLevelDensity . . . . .	42
7.27.3	Friends And Related Function Documentation . . . . .	42
7.27.3.1	KopeckyUhlGSF . . . . .	42
7.27.4	Member Data Documentation . . . . .	42
7.27.4.1	A_ . . . . .	42
7.27.4.2	backshift_ . . . . .	43
7.27.4.3	constAngTerm_ . . . . .	43
7.27.4.4	criticalU_ . . . . .	43
7.27.4.5	e0_ . . . . .	43
7.27.4.6	J_ . . . . .	43
7.27.4.7	nuclearTemp_ . . . . .	43
7.27.4.8	r0_ . . . . .	43
7.27.4.9	Z_ . . . . .	43
7.27.4.10	zeta_ . . . . .	43
7.28	LevelsContainer Class Reference . . . . .	43
7.28.1	Constructor & Destructor Documentation . . . . .	44
7.28.1.1	LevelsContainer . . . . .	44
7.28.1.2	LevelsContainer . . . . .	44
7.28.2	Member Data Documentation . . . . .	44
7.28.2.1	levels_ . . . . .	44
7.29	MassEntry Class Reference . . . . .	44
7.29.1	Constructor & Destructor Documentation . . . . .	44
7.29.1.1	MassEntry . . . . .	44
7.29.1.2	MassEntry . . . . .	44
7.29.2	Member Data Documentation . . . . .	44
7.29.2.1	expMass_ . . . . .	45

7.29.2.2	mask_ . . . . .	45
7.29.2.3	microEnergyCorr_ . . . . .	45
7.29.2.4	thMass_ . . . . .	45
7.30	MassKey Class Reference . . . . .	45
7.30.1	Constructor & Destructor Documentation . . . . .	45
7.30.1.1	MassKey . . . . .	45
7.30.2	Member Function Documentation . . . . .	45
7.30.2.1	operator< . . . . .	45
7.30.3	Member Data Documentation . . . . .	45
7.30.3.1	A_ . . . . .	45
7.30.3.2	Z_ . . . . .	46
7.31	McCullaghGSF Class Reference . . . . .	46
7.31.1	Constructor & Destructor Documentation . . . . .	46
7.31.1.1	McCullaghGSF . . . . .	46
7.31.2	Member Function Documentation . . . . .	46
7.31.2.1	CalcStrengthFunction . . . . .	46
7.32	McFaddenSatchlerPotential Class Reference . . . . .	46
7.32.1	Constructor & Destructor Documentation . . . . .	47
7.32.1.1	McFaddenSatchlerPotential . . . . .	47
7.32.2	Member Function Documentation . . . . .	47
7.32.2.1	Calculate . . . . .	47
7.33	NuclearLevels Class Reference . . . . .	47
7.33.1	Member Function Documentation . . . . .	48
7.33.1.1	FindLevels . . . . .	48
7.33.1.2	InitializeLevels . . . . .	48
7.33.1.3	PrintLevels . . . . .	48
7.34	NuclearMass Class Reference . . . . .	48
7.34.1	Member Function Documentation . . . . .	48
7.34.1.1	CalculateLDMMass . . . . .	48
7.34.1.2	FindElement . . . . .	48
7.34.1.3	FindMass . . . . .	49
7.34.1.4	FindZ . . . . .	49
7.34.1.5	HighestBoundEnergy . . . . .	49
7.34.1.6	InitializeElements . . . . .	49
7.34.1.7	InitializeMasses . . . . .	49
7.34.1.8	MassDifference . . . . .	49
7.34.1.9	MicroEnergyCorr . . . . .	49
7.34.1.10	NeutronPairingGap . . . . .	49
7.34.1.11	ProtonPairingGap . . . . .	49
7.34.1.12	QValue . . . . .	49

7.35 ODE_integration Class Reference . . . . .	50
7.35.1 Constructor & Destructor Documentation . . . . .	50
7.35.1.1 ODE_integration . . . . .	50
7.35.2 Member Function Documentation . . . . .	50
7.35.2.1 operator() . . . . .	50
7.36 ParticleHoleLevelDensity Class Reference . . . . .	50
7.36.1 Constructor & Destructor Documentation . . . . .	51
7.36.1.1 ParticleHoleLevelDensity . . . . .	51
7.36.2 Member Function Documentation . . . . .	51
7.36.2.1 FiniteDepth . . . . .	51
7.36.2.2 gNu . . . . .	51
7.36.2.3 gPi . . . . .	51
7.36.2.4 operator() . . . . .	51
7.36.2.5 PairingCorrection . . . . .	51
7.36.2.6 PauliCorrection . . . . .	51
7.37 ParticleTransmissionFunc Class Reference . . . . .	51
7.37.1 Constructor & Destructor Documentation . . . . .	52
7.37.1.1 ParticleTransmissionFunc . . . . .	52
7.37.1.2 ~ParticleTransmissionFunc . . . . .	52
7.37.2 Member Function Documentation . . . . .	52
7.37.2.1 CalcSLDependentFunctions . . . . .	52
7.37.2.2 CalcTransmission . . . . .	53
7.37.2.3 CreateParticleTransmissionFunc . . . . .	53
7.37.2.4 IsValid . . . . .	53
7.37.2.5 operator() . . . . .	53
7.37.2.6 operator() . . . . .	53
7.37.2.7 SetAlphaFormalism . . . . .	53
7.37.2.8 SetNeutronFormalism . . . . .	53
7.37.2.9 SetPorterThomas . . . . .	53
7.37.2.10 SetProtonFormalism . . . . .	53
7.37.3 Member Data Documentation . . . . .	54
7.37.3.1 m1_ . . . . .	54
7.37.3.2 parity_ . . . . .	54
7.37.3.3 pType_ . . . . .	54
7.37.3.4 redmass_ . . . . .	54
7.37.3.5 spin_ . . . . .	54
7.37.3.6 z1_ . . . . .	54
7.38 Potential Class Reference . . . . .	54
7.38.1 Constructor & Destructor Documentation . . . . .	55
7.38.1.1 Potential . . . . .	55

7.38.1.2	<a href="#">~Potential</a>	55
7.38.2	<a href="#">Member Function Documentation</a>	55
7.38.2.1	<a href="#">CalcBeta</a>	55
7.38.2.2	<a href="#">CalcTransmission</a>	55
7.38.2.3	<a href="#">Calculate</a>	56
7.38.2.4	<a href="#">GetBoundaryRadius</a>	56
7.38.2.5	<a href="#">GetRedMass</a>	56
7.38.2.6	<a href="#">GetRMax</a>	56
7.38.2.7	<a href="#">GetZ1Z2</a>	56
7.38.2.8	<a href="#">NormalizeInternally</a>	56
7.38.2.9	<a href="#">NormalizeOverAllSpace</a>	56
7.38.2.10	<a href="#">Solve</a>	56
7.38.3	<a href="#">Member Data Documentation</a>	56
7.38.3.1	<a href="#">boundaryRadius_</a>	56
7.38.3.2	<a href="#">coulFunc_</a>	57
7.38.3.3	<a href="#">coulombRadius_</a>	57
7.39	<a href="#">PreEqCDFEntry Class Reference</a>	57
7.39.1	<a href="#">Constructor &amp; Destructor Documentation</a>	57
7.39.1.1	<a href="#">PreEqCDFEntry</a>	57
7.39.2	<a href="#">Member Data Documentation</a>	57
7.39.2.1	<a href="#">energy_</a>	57
7.39.2.2	<a href="#">pairIndex_</a>	57
7.39.2.3	<a href="#">value_</a>	57
7.40	<a href="#">PreEqDecayer Class Reference</a>	57
7.40.1	<a href="#">Constructor &amp; Destructor Documentation</a>	58
7.40.1.1	<a href="#">PreEqDecayer</a>	58
7.40.1.2	<a href="#">~PreEqDecayer</a>	58
7.40.2	<a href="#">Member Function Documentation</a>	58
7.40.2.1	<a href="#">Decay</a>	58
7.40.2.2	<a href="#">GetMaxL</a>	58
7.40.2.3	<a href="#">PrintCDF</a>	58
7.40.2.4	<a href="#">SetCrossSection</a>	58
7.40.2.5	<a href="#">SetMaxL</a>	58
7.41	<a href="#">PreEqSpinRatePair Class Reference</a>	58
7.41.1	<a href="#">Constructor &amp; Destructor Documentation</a>	59
7.41.1.1	<a href="#">PreEqSpinRatePair</a>	59
7.41.2	<a href="#">Member Data Documentation</a>	59
7.41.2.1	<a href="#">A_</a>	59
7.41.2.2	<a href="#">integral_</a>	59
7.41.2.3	<a href="#">neutronHoleNumber_</a>	59

7.41.2.4	neutronNumber_ . . . . .	59
7.41.2.5	parity_ . . . . .	59
7.41.2.6	protonHoleNumber_ . . . . .	59
7.41.2.7	protonNumber_ . . . . .	59
7.41.2.8	qValue_ . . . . .	59
7.41.2.9	rateFunc_ . . . . .	59
7.41.2.10	spin_ . . . . .	59
7.41.2.11	Z_ . . . . .	59
7.42	PreEqTransitionRateFunc Class Reference . . . . .	60
7.42.1	Constructor & Destructor Documentation . . . . .	60
7.42.1.1	PreEqTransitionRateFunc . . . . .	60
7.42.1.2	~PreEqTransitionRateFunc . . . . .	60
7.42.2	Member Function Documentation . . . . .	60
7.42.2.1	CumulativeSum . . . . .	60
7.42.2.2	Integral . . . . .	60
7.43	RauscherLevelDensity Class Reference . . . . .	60
7.43.1	Constructor & Destructor Documentation . . . . .	61
7.43.1.1	RauscherLevelDensity . . . . .	61
7.43.1.2	~RauscherLevelDensity . . . . .	61
7.43.2	Member Function Documentation . . . . .	61
7.43.2.1	CalcBackShift . . . . .	61
7.43.2.2	CalcDensityParam . . . . .	61
7.43.2.3	CalcNuclearTemp . . . . .	61
7.44	SLPair Class Reference . . . . .	61
7.44.1	Constructor & Destructor Documentation . . . . .	62
7.44.1.1	SLPair . . . . .	62
7.44.2	Member Function Documentation . . . . .	62
7.44.2.1	operator< . . . . .	62
7.44.3	Member Data Documentation . . . . .	62
7.44.3.1	I_ . . . . .	62
7.44.3.2	s_ . . . . .	62
7.45	SpinRatePair Class Reference . . . . .	62
7.45.1	Constructor & Destructor Documentation . . . . .	62
7.45.1.1	SpinRatePair . . . . .	62
7.45.2	Member Data Documentation . . . . .	62
7.45.2.1	A_ . . . . .	62
7.45.2.2	integral_ . . . . .	63
7.45.2.3	parity_ . . . . .	63
7.45.2.4	qValue_ . . . . .	63
7.45.2.5	rateFunc_ . . . . .	63

7.45.2.6	spin_ . . . . .	63
7.45.2.7	Z_ . . . . .	63
7.46	TransitionRateFunc Class Reference . . . . .	63
7.46.1	Constructor & Destructor Documentation . . . . .	63
7.46.1.1	TransitionRateFunc . . . . .	63
7.46.1.2	~TransitionRateFunc . . . . .	64
7.46.2	Member Function Documentation . . . . .	64
7.46.2.1	CalcLevelDensity . . . . .	64
7.46.2.2	CalcTotalLevelDensity . . . . .	64
7.46.2.3	CalcTransmissionFunc . . . . .	64
7.46.2.4	CumulativeSum . . . . .	64
7.46.2.5	ExclusiveBranching . . . . .	64
7.46.2.6	Function . . . . .	64
7.46.2.7	GetGammaCutoffEnergy . . . . .	64
7.46.2.8	GroundStateTransmission . . . . .	64
7.46.2.9	Integral . . . . .	64
7.46.2.10	SetGammaCutoffEnergy . . . . .	64
7.47	TransmissionFunc Class Reference . . . . .	64
7.47.1	Constructor & Destructor Documentation . . . . .	65
7.47.1.1	TransmissionFunc . . . . .	65
7.47.1.2	~TransmissionFunc . . . . .	65
7.47.2	Member Function Documentation . . . . .	65
7.47.2.1	IsValid . . . . .	65
7.47.2.2	operator() . . . . .	65
7.47.3	Member Data Documentation . . . . .	66
7.47.3.1	jFinal_ . . . . .	66
7.47.3.2	jInitial_ . . . . .	66
7.47.3.3	m2_ . . . . .	66
7.47.3.4	maxL_ . . . . .	66
7.47.3.5	piFinal_ . . . . .	66
7.47.3.6	piInitial_ . . . . .	66
7.47.3.7	previous_ . . . . .	66
7.47.3.8	totalWidthForCorrection_ . . . . .	66
7.47.3.9	uncorrTotalWidthForCorrection_ . . . . .	66
7.47.3.10	uncorrTotalWidthSqrForCorrection_ . . . . .	66
7.47.3.11	z2_ . . . . .	66
7.48	XYPair Class Reference . . . . .	67
7.48.1	Constructor & Destructor Documentation . . . . .	67
7.48.1.1	XYPair . . . . .	67
7.48.2	Member Data Documentation . . . . .	67

7.48.2.1	X_ . . . . .	67
7.48.2.2	Y_ . . . . .	67
<b>8</b>	<b>File Documentation</b>	<b>69</b>
8.1	CMakeFiles/3.13.2/CompilerIdC/CMakeCCompilerId.c File Reference . . . . .	69
8.1.1	Macro Definition Documentation . . . . .	69
8.1.1.1	ARCHITECTURE_ID . . . . .	69
8.1.1.2	C_DIALECT . . . . .	69
8.1.1.3	COMPILER_ID . . . . .	69
8.1.1.4	DEC . . . . .	69
8.1.1.5	HEX . . . . .	70
8.1.1.6	PLATFORM_ID . . . . .	70
8.1.1.7	STRINGIFY . . . . .	70
8.1.1.8	STRINGIFY_HELPER . . . . .	70
8.1.2	Function Documentation . . . . .	70
8.1.2.1	main . . . . .	70
8.1.3	Variable Documentation . . . . .	70
8.1.3.1	info_arch . . . . .	70
8.1.3.2	info_compiler . . . . .	70
8.1.3.3	info_language_dialect_default . . . . .	70
8.1.3.4	info_platform . . . . .	70
8.2	CMakeFiles/3.13.2/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference . . . . .	71
8.2.1	Macro Definition Documentation . . . . .	71
8.2.1.1	ARCHITECTURE_ID . . . . .	71
8.2.1.2	COMPILER_ID . . . . .	71
8.2.1.3	CXX_STD . . . . .	71
8.2.1.4	DEC . . . . .	71
8.2.1.5	HEX . . . . .	71
8.2.1.6	PLATFORM_ID . . . . .	72
8.2.1.7	STRINGIFY . . . . .	72
8.2.1.8	STRINGIFY_HELPER . . . . .	72
8.2.2	Function Documentation . . . . .	72
8.2.2.1	main . . . . .	72
8.2.3	Variable Documentation . . . . .	72
8.2.3.1	info_arch . . . . .	72
8.2.3.2	info_compiler . . . . .	72
8.2.3.3	info_language_dialect_default . . . . .	72
8.2.3.4	info_platform . . . . .	72
8.3	CMakeFiles/feature_tests.c File Reference . . . . .	72
8.3.1	Function Documentation . . . . .	72



8.3.1.1	main	72
8.3.2	Variable Documentation	73
8.3.2.1	features	73
8.4	CMakeFiles/feature_tests.cxx File Reference	73
8.4.1	Function Documentation	73
8.4.1.1	main	73
8.4.2	Variable Documentation	73
8.4.2.1	features	73
8.5	CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference	73
8.5.1	Function Documentation	73
8.5.1.1	main	73
8.5.2	Variable Documentation	74
8.5.2.1	ompver_str	74
8.6	CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference	74
8.6.1	Function Documentation	74
8.6.1.1	main	74
8.6.2	Variable Documentation	74
8.6.2.1	ompver_str	74
8.7	CMakeFiles/FindOpenMP/OpenMPTryFlag.c File Reference	74
8.7.1	Function Documentation	75
8.7.1.1	main	75
8.8	CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp File Reference	75
8.8.1	Function Documentation	75
8.8.1.1	main	75
8.9	generated/Sapphire_config.h File Reference	75
8.9.1	Function Documentation	75
8.9.1.1	sourceDirectory	75
8.10	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/generated/Sapphire_config.h File Reference	75
8.10.1	Function Documentation	76
8.10.1.1	sourceDirectory	76
8.11	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/complex_functions.H File Reference	76
8.11.1	Macro Definition Documentation	77
8.11.1.1	SIGN	77
8.11.2	Function Documentation	77
8.11.2.1	exp_l_omega_chi_calc	77
8.11.2.2	expm1	77
8.11.2.3	inf_norm	77
8.11.2.4	isfinite	77
8.11.2.5	log1p	77

8.11.2.6	<a href="#">log_Cl_eta_calc</a>	77
8.11.2.7	<a href="#">log_cut_constant_AS_calc</a>	78
8.11.2.8	<a href="#">log_cut_constant_CFa_calc</a>	78
8.11.2.9	<a href="#">log_cut_constant_CFb_calc</a>	78
8.11.2.10	<a href="#">log_Gamma</a>	78
8.11.2.11	<a href="#">operator!=</a>	78
8.11.2.12	<a href="#">operator!=</a>	78
8.11.2.13	<a href="#">operator!=</a>	78
8.11.2.14	<a href="#">operator!=</a>	78
8.11.2.15	<a href="#">operator*</a>	78
8.11.2.16	<a href="#">operator*</a>	78
8.11.2.17	<a href="#">operator*</a>	78
8.11.2.18	<a href="#">operator*</a>	78
8.11.2.19	<a href="#">operator+</a>	78
8.11.2.20	<a href="#">operator+</a>	78
8.11.2.21	<a href="#">operator+</a>	78
8.11.2.22	<a href="#">operator+</a>	78
8.11.2.23	<a href="#">operator-</a>	78
8.11.2.24	<a href="#">operator-</a>	78
8.11.2.25	<a href="#">operator-</a>	78
8.11.2.26	<a href="#">operator-</a>	78
8.11.2.27	<a href="#">operator/</a>	78
8.11.2.28	<a href="#">operator/</a>	79
8.11.2.29	<a href="#">operator/</a>	79
8.11.2.30	<a href="#">operator/</a>	79
8.11.2.31	<a href="#">operator==</a>	79
8.11.2.32	<a href="#">operator==</a>	79
8.11.2.33	<a href="#">operator==</a>	79
8.11.2.34	<a href="#">operator==</a>	79
8.11.2.35	<a href="#">sigma_l_calc</a>	79
8.11.2.36	<a href="#">sin_chi_calc</a>	79
8.11.3	<a href="#">Variable Documentation</a>	79
8.11.3.1	<a href="#">precision</a>	79
8.11.3.2	<a href="#">sqrt_precision</a>	79
8.12	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/cwfcomp.H File Reference</a>	79
8.13	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/ode_int.H File Reference</a>	80
8.14	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/complex_functions.cpp File Reference</a>	80
8.14.1	<a href="#">Function Documentation</a>	80
8.14.1.1	<a href="#">exp_l_omega_chi_calc</a>	80
8.14.1.2	<a href="#">expm1</a>	80

8.14.1.3	<a href="#">log1p</a>	81
8.14.1.4	<a href="#">log_Cl_eta_calc</a>	81
8.14.1.5	<a href="#">log_cut_constant_AS_calc</a>	81
8.14.1.6	<a href="#">log_cut_constant_CFa_calc</a>	81
8.14.1.7	<a href="#">log_cut_constant_CFb_calc</a>	81
8.14.1.8	<a href="#">log_Gamma</a>	81
8.14.1.9	<a href="#">sigma_l_calc</a>	81
8.14.1.10	<a href="#">sin_chi_calc</a>	81
8.15	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/cwfcomp.cpp File Reference</a>	81
8.16	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/ode_int.cpp File Reference</a>	82
8.17	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/BrinkAxelGSF.h File Reference</a>	82
8.17.1	Detailed Description	82
8.18	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Constants.h File Reference</a>	82
8.18.1	Typedef Documentation	83
8.18.1.1	<a href="#">complex</a>	83
8.18.1.2	<a href="#">matrix_c</a>	83
8.18.1.3	<a href="#">matrix_r</a>	83
8.18.1.4	<a href="#">vector_c</a>	83
8.18.1.5	<a href="#">vector_matrix_c</a>	83
8.18.1.6	<a href="#">vector_matrix_r</a>	83
8.18.1.7	<a href="#">vector_r</a>	83
8.18.2	Variable Documentation	83
8.18.2.1	<a href="#">avagadroNum</a>	83
8.18.2.2	<a href="#">boltzConst</a>	83
8.18.2.3	<a href="#">eMass</a>	83
8.18.2.4	<a href="#">fstruc</a>	83
8.18.2.5	<a href="#">hbarc</a>	83
8.18.2.6	<a href="#">lightSpeedInCmPerS</a>	84
8.18.2.7	<a href="#">pi</a>	84
8.18.2.8	<a href="#">uconv</a>	84
8.19	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/CoulFunc.h File Reference</a>	84
8.20	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/CrossSection.h File Reference</a>	84
8.20.1	Typedef Documentation	85
8.20.1.1	<a href="#">DecayerVector</a>	85
8.20.1.2	<a href="#">int_double_pair</a>	85
8.21	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayController.h File Reference</a>	85
8.22	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Decayer.h File Reference</a>	85
8.23	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayProduct.h File Reference</a>	85
8.24	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayResults.h File Reference</a>	85
8.25	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/elements.h File Reference</a>	86

8.26	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/EquivSquareWell.h File Reference . .	86
8.27	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/GammaTransmissionFunc.h File Reference . . . . .	86
8.27.1	Typedef Documentation . . . . .	86
8.27.1.1	GDRTTable . . . . .	86
8.28	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/JLMPotential.h File Reference . . . .	86
8.29	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/KopeckyUhlGSF.h File Reference . .	87
8.30	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/LevelDensity.h File Reference . . . .	87
8.31	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/McCullaghGSF.h File Reference . . .	87
8.32	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/McFaddenSatchlerPotential.h File Reference . . . . .	87
8.33	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearLevels.h File Reference . . .	87
8.33.1	Typedef Documentation . . . . .	88
8.33.1.1	LevelsTable . . . . .	88
8.34	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearMass.h File Reference . . . .	88
8.34.1	Typedef Documentation . . . . .	89
8.34.1.1	ElementTable . . . . .	89
8.34.1.2	MassTable . . . . .	89
8.35	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ParticleHoleLevelDensity.h File Reference . . . . .	89
8.36	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ParticleTransmissionFunc.h File Reference . . . . .	89
8.37	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Potential.h File Reference . . . . .	89
8.38	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqDecayer.h File Reference . . .	89
8.39	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqTransitionRateFunc.h File Reference . . . . .	90
8.40	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/RauscherLevelDensity.h File Reference	90
8.41	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/SapphireMPITypes.h File Reference .	90
8.41.1	Enumeration Type Documentation . . . . .	91
8.41.1.1	SapphireTags_t . . . . .	91
8.41.2	Function Documentation . . . . .	91
8.41.2.1	BOOST_IS_MPI_DATATYPE . . . . .	91
8.41.2.2	BOOST_IS_MPI_DATATYPE . . . . .	91
8.41.2.3	BOOST_IS_MPI_DATATYPE . . . . .	91
8.42	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransitionRateFunc.h File Reference	91
8.43	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransmissionFunc.h File Reference .	91
8.44	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/README.md File Reference . . . . .	92
8.45	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/BrinkAxelGSF.cpp File Reference . . . .	92
8.46	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CoulFunc.cpp File Reference . . . . .	92
8.47	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CrossSection.cpp File Reference . . . .	92
8.47.1	Function Documentation . . . . .	93

8.47.1.1	<a href="#">gsl_partfunc_integrand</a>	93
8.47.1.2	<a href="#">gsl_reactionrate_integrand</a>	93
8.48	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayController.cpp File Reference</a>	93
8.48.1	Variable Documentation	93
8.48.1.1	<a href="#">randomSeed</a>	93
8.49	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Decayer.cpp File Reference</a>	93
8.49.1	Variable Documentation	94
8.49.1.1	<a href="#">randomSeed</a>	94
8.50	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayResults.cpp File Reference</a>	94
8.51	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/EquivSquareWell.cpp File Reference</a>	94
8.52	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/GammaTransmissionFunc.cpp File Reference</a>	94
8.52.1	Variable Documentation	94
8.52.1.1	<a href="#">randomSeed</a>	94
8.53	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/JLMPotential.cpp File Reference</a>	95
8.54	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/KopeckyUhlGSF.cpp File Reference</a>	95
8.55	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/LevelDensity.cpp File Reference</a>	95
8.56	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McCullaghGSF.cpp File Reference</a>	95
8.57	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McFaddenSatchlerPotential.cpp File Reference</a>	95
8.58	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearLevels.cpp File Reference</a>	95
8.59	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearMass.cpp File Reference</a>	96
8.59.1	Macro Definition Documentation	96
8.59.1.1	<a href="#">ELEMENT</a>	96
8.59.1.2	<a href="#">HAS_EXP_MASS</a>	96
8.59.1.3	<a href="#">HAS_TH_MASS</a>	96
8.60	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleHoleLevelDensity.cpp File Reference</a>	96
8.61	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleTransmissionFunc.cpp File Reference</a>	97
8.61.1	Variable Documentation	97
8.61.1.1	<a href="#">randomSeed</a>	97
8.62	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Potential.cpp File Reference</a>	97
8.63	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqDecayer.cpp File Reference</a>	97
8.63.1	Variable Documentation	98
8.63.1.1	<a href="#">randomSeed</a>	98
8.64	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqTransitionRateFunc.cpp File Reference</a>	98
8.65	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/RauscherLevelDensity.cpp File Reference</a>	98
8.66	<a href="#">/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Sapphire.cpp File Reference</a>	98
8.66.1	Typedef Documentation	99
8.66.1.1	<a href="#">EntrancePairs</a>	99

8.66.2	Function Documentation	99
8.66.2.1	Initialize	99
8.66.2.2	main	99
8.66.2.3	parseCommandLineForDecay	99
8.66.2.4	parseCommandLineForOptions	100
8.66.2.5	parseCommandLineForXS	100
8.66.2.6	printHelp	100
8.66.3	Variable Documentation	100
8.66.3.1	randomSeed	100
8.67	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp File Reference	100
8.67.1	Function Documentation	101
8.67.1.1	Initialize	101
8.68	/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/TransitionRateFunc.cpp File Reference	101

## Index

102

# Chapter 1

## README

This is the original version of Sapphire by [Mary Beard](#). Only the README-file has been changed.

- [General Information](#)
- [Compilation & Execution](#)
  - [Requirements](#)
  - [Build](#)
  - [Run](#)
- [Features](#)
- [References](#)
  - [About Sapphire](#)
  - [Using Sapphire](#)

### General Information

### Compilation & Execution

#### Requirements

Compiling Sapphire requires [CMake](#), [ROOT](#), and [GNU Scientific Library \(GSL\)](#). Check the files in `cmake/Modules` whether `cmake` is looking for these packages in the right place.

For an automatized generation of documentation files, one needs [doxygen](#) installed on the system.

For the Message Passing Interface (MPI) build, `boost 1.4` libraries are required. The MPI build is optional and can be enabled in the `CMakeLists.txt`.

#### Build

1. Create a build directory under the main Sapphire directory and change to that directory. `“bash mkdir build cd build “`
2. Run CMake against the main directory, optionally specifying the desired C++ compiler. `“bash cmake -DCMAKE_CXX_COMPILER=icpc -DCMAKE_C_COMPILER=icc “`
3. Type `make install` to build Sapphire. The executable is put in the build directory. `“bash make install “`

Be aware that Sapphire links the paths to the needed tables at compile time. While the executable can be moved, the main Sapphire directory should stay in place. If moved, the build process should be repeated.

## Run

To execute the code just enter

```
“ ./sapphire X+a “
```

where X is the heavy nucleus and a is the projectile. Examples are  $^{25}\text{Mg}+a$  or  $^{60}\text{Fe}+n$ .

## Features

### Calculate Cross Sections

## References

### About Sapphire

If you are using *Sapphire* you should cite one or more of the references below.

- M. Beard, E. Uberseder, R. Crowter, and M. Wiescher, *Comparison of statistical model calculations for stable isotope neutron capture*, Physical Review C **90**, 034619 (2014), DOI: [10.1103/PhysRevC.90.034619](https://doi.org/10.1103/PhysRevC.90.034619)
- Mary Beard, Ethan Uberseder, and M. Wiescher, *Statistical Model Calculations for (n,g) Reactions*, EPJ Web of Conferences **93**, 04001 (2015), DOI: [10.1103/PhysRevC.90.034619](https://doi.org/10.1103/PhysRevC.90.034619)

### Using Sapphire

If you are using *Sapphire* for your research, please send us a reference of your work and we will list it here.

- X. Fang, W. P. Tan, M. Beard, R. J. deBoer, G. Gilardy, H. Jung, Q. Liu, S. Lyons, D. Robertson, K. Setood-ehnia, C. Seymour, E. Stech, B. Vande Kolk, M. Wiescher, R. T. deSouza, S. Hudan, V. Singh, X. D. Tang, and E. Uberseder, *Experimental measurement of  $^{12}\text{C}+^{16}\text{O}$  fusion at stellar energies*, Physical Review C **96**, 045804 (2017), DOI: [10.1103/PhysRevC.96.045804](https://doi.org/10.1103/PhysRevC.96.045804)
- M. Beard, E. Uberseder, R. Crowter, and M. Wiescher, *Comparison of statistical model calculations for stable isotope neutron capture*, Physical Review C **90**, 034619 (2014), DOI: [10.1103/PhysRevC.90.034619](https://doi.org/10.1103/PhysRevC.90.034619)
- Mary Beard, Ethan Uberseder, and M. Wiescher, *Statistical Model Calculations for (n,g) Reactions*, EPJ Web of Conferences **93**, 04001 (2015), DOI: [10.1103/PhysRevC.90.034619](https://doi.org/10.1103/PhysRevC.90.034619)



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">boost</a>	.....	??
<a href="#">boost::serialization</a>	.....	??
<a href="#">std</a>	.....	??
<a href="#">std::tr1</a>	.....	??



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

CDFEntry . . . . .	??
CoulFunc . . . . .	??
Coulomb_wave_functions . . . . .	??
CoulWaves . . . . .	??
CrossSection . . . . .	??
CrossSectionValues . . . . .	??
DecayController . . . . .	??
DecayData . . . . .	??
Decayer . . . . .	??
DecayProduct . . . . .	??
DecayResults . . . . .	??
EntrancePairs . . . . .	??
std::equal_to< MassKey > . . . . .	??
GammaTransition . . . . .	??
GDRParameters . . . . .	??
gsl_partfunc_params . . . . .	??
gsl_reactionrate_params . . . . .	??
std::tr1::hash< MassKey > . . . . .	??
InitialNucleusData . . . . .	??
int_double_pair_compare . . . . .	??
Level . . . . .	??
LevelDensity . . . . .	??
RauscherLevelDensity . . . . .	??
LevelsContainer . . . . .	??
MassEntry . . . . .	??
MassKey . . . . .	??
NuclearLevels . . . . .	??
NuclearMass . . . . .	??
ODE_integration . . . . .	??
ParticleHoleLevelDensity . . . . .	??
PreEqCDFEntry . . . . .	??
PreEqDecayer . . . . .	??
PreEqSpinRatePair . . . . .	??
PreEqTransitionRateFunc . . . . .	??
SLPair . . . . .	??
SpinRatePair . . . . .	??
TransitionRateFunc . . . . .	??
TransmissionFunc . . . . .	??

GammaTransmissionFunc . . . . .	??
BrinkAxelGSF . . . . .	??
KopeckyUhlGSF . . . . .	??
McCullaghGSF . . . . .	??
ParticleTransmissionFunc . . . . .	??
EquivSquareWell . . . . .	??
Potential . . . . .	??
JLMPotential . . . . .	??
McFaddenSatchlerPotential . . . . .	??
XYPair . . . . .	??

## Chapter 4

# Class Index

### 4.1 Class List

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

BrinkAxelGSF	??
CDFEntry	??
CoulFunc	??
Coulomb_wave_functions	??
CoulWaves	??
CrossSection	??
CrossSectionValues	??
DecayController	??
DecayData	??
Decayer	??
DecayProduct	??
DecayResults	??
EntrancePairs	??
std::equal_to< MassKey >	??
EquivSquareWell	??
GammaTransition	??
GammaTransmissionFunc	??
GDRParameters	??
gsl_partfunc_params	??
gsl_reactionrate_params	??
std::tr1::hash< MassKey >	??
InitialNucleusData	??
int_double_pair_compare	??
JLMPotential	??
KopeckyUhlGSF	??
Level	??
LevelDensity	??
LevelsContainer	??
MassEntry	??
MassKey	??
McCullaghGSF	??
McFaddenSatchlerPotential	??
NuclearLevels	??
NuclearMass	??
ODE_integration	??
ParticleHoleLevelDensity	??
ParticleTransmissionFunc	??
Potential	??

---

PreEqCDFEntry . . . . .	??
PreEqDecayer . . . . .	??
PreEqSpinRatePair . . . . .	??
PreEqTransitionRateFunc . . . . .	??
RauscherLevelDensity . . . . .	??
SLPair . . . . .	??
SpinRatePair . . . . .	??
TransitionRateFunc . . . . .	??
TransmissionFunc . . . . .	??
XYPair . . . . .	??

## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

CMakeFiles/ <a href="#">feature_tests.c</a> . . . . .	??
CMakeFiles/ <a href="#">feature_tests.cxx</a> . . . . .	??
CMakeFiles/3.13.2/CompilerIdC/ <a href="#">CMakeCCompilerId.c</a> . . . . .	??
CMakeFiles/3.13.2/CompilerIdCXX/ <a href="#">CMakeCXXCompilerId.cpp</a> . . . . .	??
CMakeFiles/FindOpenMP/ <a href="#">OpenMPCheckVersion.c</a> . . . . .	??
CMakeFiles/FindOpenMP/ <a href="#">OpenMPCheckVersion.cpp</a> . . . . .	??
CMakeFiles/FindOpenMP/ <a href="#">OpenMPTryFlag.c</a> . . . . .	??
CMakeFiles/FindOpenMP/ <a href="#">OpenMPTryFlag.cpp</a> . . . . .	??
generated/ <a href="#">Sapphire_config.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/ <a href="#">complex_functions.H</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/ <a href="#">cwfcomp.H</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/ <a href="#">ode_int.H</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/ <a href="#">complex_functions.cpp</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/ <a href="#">cwfcomp.cpp</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/ <a href="#">ode_int.cpp</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/generated/ <a href="#">Sapphire_config.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">BrinkAxelGSF.h</a> . . . . .	??
Header file for the Brink-Axel-GSF class . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">Constants.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">CoulFunc.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">CrossSection.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">DecayController.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">Decayer.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">DecayProduct.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">DecayResults.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">elements.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">EquivSquareWell.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">GammaTransmissionFunc.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">JLMPotential.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">KopeckyUhlGSF.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">LevelDensity.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">McCullaghGSF.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">McFaddenSatchlerPotential.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">NuclearLevels.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">NuclearMass.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">ParticleHoleLevelDensity.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">ParticleTransmissionFunc.h</a> . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ <a href="#">Potential.h</a> . . . . .	??

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqDecayer.h . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqTransitionRateFunc.h . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/RauscherLevelDensity.h . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/SapphireMPITypes.h . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransitionRateFunc.h . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransmissionFunc.h . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/BrinkAxelGSF.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CoulFunc.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CrossSection.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayController.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Decayer.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayResults.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/EquivSquareWell.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/GammaTransmissionFunc.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/JLMPotential.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/KopeckyUhlGSF.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/LevelDensity.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McCullaghGSF.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McFaddenSatchlerPotential.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearLevels.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearMass.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleHoleLevelDensity.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleTransmissionFunc.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Potential.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqDecayer.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqTransitionRateFunc.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/RauscherLevelDensity.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Sapphire.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp . . . . .	??
/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/TransitionRateFunc.cpp . . . . .	??



## Chapter 6

# Namespace Documentation

### 6.1 boost Namespace Reference

#### Namespaces

- [serialization](#)

### 6.2 boost::serialization Namespace Reference

#### Functions

- void [serialize](#) (Archive &ar, [DecayData](#) &g, const unsigned int version)
- void [serialize](#) (Archive &ar, [DecayProduct](#) &g, const unsigned int version)

#### 6.2.1 Function Documentation

6.2.1.1 void boost::serialization::serialize ( Archive &ar, [DecayData](#) &g, const unsigned int *version* )

6.2.1.2 void boost::serialization::serialize ( Archive &ar, [DecayProduct](#) &g, const unsigned int *version* )

References [DecayProduct::A\\_](#), [DecayProduct::excitationEnergy\\_](#), [DecayProduct::fragmentEnergy\\_](#), [DecayProduct::fragmentEnergyCM\\_](#), [DecayProduct::fragmentMomentumX\\_](#), [DecayProduct::fragmentMomentumY\\_](#), [DecayProduct::fragmentMomentumZ\\_](#), [DecayProduct::J\\_](#), [DecayProduct::particleEnergy\\_](#), [DecayProduct::particleEnergyCM\\_](#), [DecayProduct::particleMomentumX\\_](#), [DecayProduct::particleMomentumY\\_](#), [DecayProduct::particleMomentumZ\\_](#), [DecayProduct::particlePhiCM\\_](#), [DecayProduct::particleThetaCM\\_](#), [DecayProduct::particleType\\_](#), [DecayProduct::Pi\\_](#), and [DecayProduct::Z\\_](#).

### 6.3 std Namespace Reference

#### Namespaces

- [tr1](#)

#### Classes

- struct [equal\\_to< MassKey >](#)

## 6.4 std::tr1 Namespace Reference

### Classes

- struct [hash](#)< [MassKey](#) >

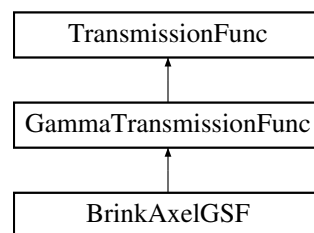
## Chapter 7

# Class Documentation

### 7.1 BrinkAxelGSF Class Reference

```
#include <BrinkAxelGSF.h>
```

Inheritance diagram for BrinkAxelGSF:



#### Public Member Functions

- [BrinkAxelGSF](#) (int, int, double, int, double, int, double, double, double, double, [TransmissionFunc](#) \*)
- double [CalcStrengthFunction](#) (double)

#### Additional Inherited Members

##### 7.1.1 Detailed Description

Class for the Brink-Axel-GSF

##### 7.1.2 Constructor & Destructor Documentation

7.1.2.1 **BrinkAxelGSF::BrinkAxelGSF** ( int *z2*, int *m2*, double *jInitial*, int *piInitial*, double *jFinal*, int *piFinal*, double *maxL*, double *totalWidthForCorrection*, double *uncorrTotalWidthForCorrection*, double *uncorrTotalWidthSqrdForCorrection*, [TransmissionFunc](#) \* *previous* )

[BrinkAxelGSF.h](#)

##### 7.1.3 Member Function Documentation

### 7.1.3.1 `double BrinkAxelGSF::CalcStrengthFunction ( double energy ) [virtual]`

Implements [GammaTransmissionFunc](#).

References `GDRParameters::E_`, `GammaTransmissionFunc::gdrParameters_`, `GDRParameters::kSigmaGamma_`, and `TransmissionFunc::maxL_`.

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/BrinkAxelGSF.h>
- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/BrinkAxelGSF.cpp>

## 7.2 CDFEntry Class Reference

```
#include <Decayer.h>
```

### Public Member Functions

- [CDFEntry](#) (int pairIndex, double energy, double value)

### Public Attributes

- int [pairIndex\\_](#)
- double [energy\\_](#)
- double [value\\_](#)

### 7.2.1 Constructor & Destructor Documentation

#### 7.2.1.1 `CDFEntry::CDFEntry ( int pairIndex, double energy, double value ) [inline]`

### 7.2.2 Member Data Documentation

#### 7.2.2.1 `double CDFEntry::energy_`

#### 7.2.2.2 `int CDFEntry::pairIndex_`

#### 7.2.2.3 `double CDFEntry::value_`

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Decayer.h>

## 7.3 CoulFunc Class Reference

```
#include <CoulFunc.h>
```

### Public Member Functions

- [CoulFunc](#) (int [z1](#), int [z2](#), double [redmass](#), bool useGSLFunctions)
- int [z1](#) () const
- int [z2](#) () const

- double [redmass](#) () const
- int [lLast](#) () const
- double [radiusLast](#) () const
- double [energyLast](#) () const
- struct [CoulWaves coulLast](#) () const
- void [setLast](#) (int, double, double, [CoulWaves](#))
- [CoulWaves operator\(\)](#) (int, double, double)
- double [Penetrability](#) (int, double, double)
- double [PEShift](#) (int, double, double)
- double [PEShift\\_dE](#) (int, double, double)

### Static Public Member Functions

- static void [GSLErrorHandler](#) (const char \*, const char \*, int, int)

### 7.3.1 Constructor & Destructor Documentation

#### 7.3.1.1 [CoulFunc::CoulFunc](#) ( int *z1*, int *z2*, double *redmass*, bool *useGSLFunctions* )

References [CoulWaves::dF](#), [CoulWaves::dG](#), [CoulWaves::F](#), and [CoulWaves::G](#).

### 7.3.2 Member Function Documentation

#### 7.3.2.1 [struct CoulWaves CoulFunc::coulLast](#) ( ) const

Returns the last Coulomb functions which were calculated.

Referenced by [operator\(\)](#)().

#### 7.3.2.2 [double CoulFunc::energyLast](#) ( ) const

Returns the last energy value at which the Coulomb functions were calculated.

Referenced by [operator\(\)](#)().

#### 7.3.2.3 [void CoulFunc::GSLErrorHandler](#) ( const char \* *reason*, const char \* *file*, int *line*, int *errorCode* ) [static]

Referenced by [Initialize\(\)](#).

#### 7.3.2.4 [int CoulFunc::lLast](#) ( ) const

Returns the last orbital angular momentum value at which the Coulomb functions were calculated.

Referenced by [operator\(\)](#)(), and [setLast\(\)](#).

#### 7.3.2.5 [CoulWaves CoulFunc::operator\(\)](#) ( int *l*, double *radius*, double *energy* )

The parenthesis operator is defined to make the class instance callable as a function. The orbital angular momentum, radius, and energy in the center of mass system are the dependent variables. The function returns the Coulomb waves.

References [coulLast\(\)](#), [CoulWaves::dF](#), [CoulWaves::dG](#), [energyLast\(\)](#), [CoulWaves::F](#), [Coulomb\\_wave\\_functions::F\\_dF\(\)](#), [fstruc](#), [CoulWaves::G](#), [Coulomb\\_wave\\_functions::G\\_dG\(\)](#), [hbarc](#), [lLast\(\)](#), [radiusLast\(\)](#), [redmass\(\)](#), [uconv](#), [z1\(\)](#), and [z2\(\)](#).

Referenced by Penetrability(), and PEShift().

#### 7.3.2.6 double CoulFunc::Penetrability ( int *l*, double *radius*, double *energy* )

Returns the penetrability as a function of orbital angular momentum, radius, and energy in the center of mass system.

References CoulWaves::F, CoulWaves::G, hbarc, operator>(), redmass(), and uconv.

Referenced by EquivSquareWell::CalcTransmission().

#### 7.3.2.7 double CoulFunc::PEShift ( int *l*, double *radius*, double *energy* )

Returns the positive energy shift function a function of orbital angular momentum, radius, and energy in the center of mass system.

References CoulWaves::dF, CoulWaves::dG, CoulWaves::F, CoulWaves::G, hbarc, operator>(), redmass(), and uconv.

#### 7.3.2.8 double CoulFunc::PEShift\_dE ( int *l*, double *radius*, double *energy* )

Returns the energy derivative of the shift function a function of orbital angular momentum, radius, and energy in the center of mass system.

References CoulWaves::F.

#### 7.3.2.9 double CoulFunc::radiusLast ( ) const

Returns the last radius value at which the Coulomb functions were calculated.

Referenced by operator>().

#### 7.3.2.10 double CoulFunc::redmass ( ) const

Returns the reduced mass of the particle pair.

Referenced by operator>(), Penetrability(), and PEShift().

#### 7.3.2.11 void CoulFunc::setLast ( int *lLast*, double *rLast*, double *eLast*, CoulWaves *coulLast* )

Sets the last calculated Coulomb waves and the values for which they were calculated.

References CoulWaves::dF, CoulWaves::dG, CoulWaves::F, CoulWaves::G, and lLast().

#### 7.3.2.12 int CoulFunc::z1 ( ) const

Returns the atomic number of the first particle in the pair.

Referenced by operator>().

#### 7.3.2.13 int CoulFunc::z2 ( ) const

Returns the atomic number of the second particle in the pair.

Referenced by operator>().

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/CoulFunc.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CoulFunc.cpp](#)

## 7.4 Coulomb\_wave\_functions Class Reference

```
#include <cwfcomp.H>
```

### Public Member Functions

- [Coulomb\\_wave\\_functions](#) (const bool is\_it\_normalized\_c, const [std::complex](#)< double > &l\_c, const [std::complex](#)< double > &eta\_c)
- [~Coulomb\\_wave\\_functions](#) (void)
- void [F\\_dF\\_init](#) (const [std::complex](#)< double > &z, const [std::complex](#)< double > &F, const [std::complex](#)< double > &dF)
- void [F\\_dF](#) (const [std::complex](#)< double > &z, [std::complex](#)< double > &F, [std::complex](#)< double > &dF)
- void [G\\_dG](#) (const [std::complex](#)< double > &z, [std::complex](#)< double > &G, [std::complex](#)< double > &dG)
- void [H\\_dH](#) (const int omega, const [std::complex](#)< double > &z, [std::complex](#)< double > &H, [std::complex](#)< double > &dH)
- void [H\\_dH\\_scaled](#) (const int omega, const [std::complex](#)< double > &z, [std::complex](#)< double > &H, [std::complex](#)< double > &dH)

### Public Attributes

- const [std::complex](#)< double > l
- const [std::complex](#)< double > eta
- const bool is\_it\_normalized

### 7.4.1 Constructor & Destructor Documentation

7.4.1.1 [Coulomb\\_wave\\_functions::Coulomb\\_wave\\_functions](#) ( const bool *is\_it\_normalized\_c*, const [std::complex](#)< double > &l\_c, const [std::complex](#)< double > &eta\_c ) [inline]

References eta, and is\_it\_normalized.

7.4.1.2 [Coulomb\\_wave\\_functions::~~Coulomb\\_wave\\_functions](#) ( void ) [inline]

### 7.4.2 Member Function Documentation

7.4.2.1 void [Coulomb\\_wave\\_functions::F\\_dF](#) ( const [std::complex](#)< double > &z, [std::complex](#)< double > &F, [std::complex](#)< double > &dF )

References is\_it\_normalized, isfinite(), SIGN, and sqrt\_precision.

Referenced by G\_dG(), H\_dH(), H\_dH\_scaled(), and CoulFunc::operator()().

7.4.2.2 void [Coulomb\\_wave\\_functions::F\\_dF\\_init](#) ( const [std::complex](#)< double > &z, const [std::complex](#)< double > &F, const [std::complex](#)< double > &dF )

7.4.2.3 void [Coulomb\\_wave\\_functions::G\\_dG](#) ( const [std::complex](#)< double > &z, [std::complex](#)< double > &G, [std::complex](#)< double > &dG )

References F\_dF(), H\_dH(), is\_it\_normalized, isfinite(), and sqrt\_precision.

Referenced by CoulFunc::operator()().

**7.4.2.4** void Coulomb\_wave\_functions::H\_dH ( const int *omega*, const std::complex< double > & z, std::complex< double > & H, std::complex< double > & dH )

References F\_dF(), is\_it\_normalized, isfinite(), and sqrt\_precision.

Referenced by G\_dG().

**7.4.2.5** void Coulomb\_wave\_functions::H\_dH\_scaled ( const int *omega*, const std::complex< double > & z, std::complex< double > & H, std::complex< double > & dH )

References F\_dF(), is\_it\_normalized, isfinite(), and sqrt\_precision.

### 7.4.3 Member Data Documentation

**7.4.3.1** const std::complex<double> Coulomb\_wave\_functions::eta

Referenced by Coulomb\_wave\_functions().

**7.4.3.2** const bool Coulomb\_wave\_functions::is\_it\_normalized

Referenced by Coulomb\_wave\_functions(), F\_dF(), G\_dG(), H\_dH(), and H\_dH\_scaled().

**7.4.3.3** const std::complex<double> Coulomb\_wave\_functions::l

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/cwfcamp.H
- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/cwfcamp.cpp

## 7.5 CoulWaves Struct Reference

```
#include <CoulFunc.h>
```

### Public Attributes

- double [F](#)
- double [dF](#)
- double [G](#)
- double [dG](#)

### 7.5.1 Member Data Documentation

**7.5.1.1** double CoulWaves::dF

Referenced by Potential::CalcTransmission(), CoulFunc::CoulFunc(), CoulFunc::operator()(), CoulFunc::PEShift(), and CoulFunc::setLast().

**7.5.1.2** double CoulWaves::dG

Referenced by Potential::CalcTransmission(), CoulFunc::CoulFunc(), CoulFunc::operator()(), CoulFunc::PEShift(), and CoulFunc::setLast().



## 7.5.1.3 double CoulWaves::F

Referenced by Potential::CalcTransmission(), CoulFunc::CoulFunc(), CoulFunc::operator>(), CoulFunc::Penetrability(), CoulFunc::PEShift(), CoulFunc::PEShift\_dE(), and CoulFunc::setLast().

## 7.5.1.4 double CoulWaves::G

Referenced by Potential::CalcTransmission(), CoulFunc::CoulFunc(), CoulFunc::operator>(), CoulFunc::Penetrability(), CoulFunc::PEShift(), and CoulFunc::setLast().

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/CoulFunc.h>

## 7.6 CrossSection Class Reference

```
#include <CrossSection.h>
```

### Public Member Functions

- [CrossSection](#) (int, int, int, std::string, bool, int entranceState=0, std::vector< int > exitStates=std::vector< int >(4,-1))
- bool [IsValid](#) () const
- void [Calculate](#) ()
- void [PrintCrossSections](#) ()
- void [PrintTransmissionTerms](#) ()
- std::pair< double, double > [CalcAverageSWaveResWidth](#) ()
- std::pair< double, double > [CalcAveragePWaveResWidth](#) ()
- std::pair< double, double > [CalcAverageDWaveResWidth](#) ()
- void [CalculateReactionRates](#) (bool)
- void [PrintReactionRates](#) (bool)

### Static Public Member Functions

- static void [SetResidualGamma](#) (bool residual)
- static void [SetResidualNeutron](#) (bool residual)
- static void [SetResidualProton](#) (bool residual)
- static void [SetResidualAlpha](#) (bool residual)
- static void [SetCalculateGammaCutoff](#) (bool calc)
- static void [CreateTempVector](#) ()
- static void [CreateMACSEnergiesVector](#) ()

### 7.6.1 Constructor & Destructor Documentation

7.6.1.1 [CrossSection::CrossSection](#) ( int Z, int A, int pType, std::string *energyFile*, bool *forRates*, int *entranceState* = 0, std::vector< int > *exitStates* = std::vector<int>(4,-1) )

References Level::energy\_, NuclearLevels::FindLevels(), hbarc, Level::J\_, pi, Level::Pi\_, NuclearMass::QValue(), and uconv.

## 7.6.2 Member Function Documentation

### 7.6.2.1 `std::pair< double, double > CrossSection::CalcAverageDWaveResWidth ( )`

References `pi`.

Referenced by `main()`.

### 7.6.2.2 `std::pair< double, double > CrossSection::CalcAveragePWaveResWidth ( )`

References `pi`.

Referenced by `main()`.

### 7.6.2.3 `std::pair< double, double > CrossSection::CalcAverageSWaveResWidth ( )`

References `pi`.

Referenced by `main()`.

### 7.6.2.4 `void CrossSection::Calculate ( )`

References `TransitionRateFunc::SetGammaCutoffEnergy()`.

Referenced by `main()`.

### 7.6.2.5 `void CrossSection::CalculateReactionRates ( bool macs )`

References `avagadroNum`, `boltzConst`, `gsl_reactionrate_integrand()`, `lightSpeedInCmPerS`, `pi`, and `uconv`.

Referenced by `main()`.

### 7.6.2.6 `void CrossSection::CreateMACSEnergiesVector ( ) [static]`

Referenced by `Initialize()`.

### 7.6.2.7 `void CrossSection::CreateTempVector ( ) [static]`

Referenced by `Initialize()`.

### 7.6.2.8 `bool CrossSection::IsValid ( ) const [inline]`

Referenced by `main()`.

### 7.6.2.9 `void CrossSection::PrintCrossSections ( )`

References `NuclearMass::FindElement()`.

Referenced by `main()`.

### 7.6.2.10 `void CrossSection::PrintReactionRates ( bool macs )`

References `NuclearMass::FindElement()`.

Referenced by `main()`.

**7.6.2.11 void CrossSection::PrintTransmissionTerms ( )**

References NuclearMass::FindElement().

Referenced by main().

**7.6.2.12 static void CrossSection::SetCalculateGammaCutoff ( bool *calc* ) [inline],[static]**

Referenced by Initialize(), and parseCommandLineForOptions().

**7.6.2.13 static void CrossSection::SetResidualAlpha ( bool *residual* ) [inline],[static]**

Referenced by Initialize(), and parseCommandLineForOptions().

**7.6.2.14 static void CrossSection::SetResidualGamma ( bool *residual* ) [inline],[static]**

Referenced by Initialize(), and parseCommandLineForOptions().

**7.6.2.15 static void CrossSection::SetResidualNeutron ( bool *residual* ) [inline],[static]**

Referenced by Initialize(), and parseCommandLineForOptions().

**7.6.2.16 static void CrossSection::SetResidualProton ( bool *residual* ) [inline],[static]**

Referenced by Initialize(), and parseCommandLineForOptions().

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[CrossSection.h](#)
- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/[CrossSection.cpp](#)
- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/[Setup.cpp](#)

## 7.7 CrossSectionValues Class Reference

```
#include <CrossSection.h>
```

### Public Member Functions

- [CrossSectionValues](#) (double gamma, double neutron, double proton, double alpha, double gammaStellar, double neutronStellar, double protonStellar, double alphaStellar)

### Public Attributes

- double [gamma\\_](#)
- double [neutron\\_](#)
- double [proton\\_](#)
- double [alpha\\_](#)
- double [gammaStellar\\_](#)
- double [neutronStellar\\_](#)
- double [protonStellar\\_](#)
- double [alphaStellar\\_](#)

### 7.7.1 Constructor & Destructor Documentation

7.7.1.1 `CrossSectionValues::CrossSectionValues ( double gamma, double neutron, double proton, double alpha, double gammaStellar, double neutronStellar, double protonStellar, double alphaStellar ) [inline]`

### 7.7.2 Member Data Documentation

7.7.2.1 `double CrossSectionValues::alpha_`

7.7.2.2 `double CrossSectionValues::alphaStellar_`

7.7.2.3 `double CrossSectionValues::gamma_`

7.7.2.4 `double CrossSectionValues::gammaStellar_`

7.7.2.5 `double CrossSectionValues::neutron_`

7.7.2.6 `double CrossSectionValues::neutronStellar_`

7.7.2.7 `double CrossSectionValues::proton_`

7.7.2.8 `double CrossSectionValues::protonStellar_`

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/CrossSection.h>

## 7.8 DecayController Class Reference

```
#include <DecayController.h>
```

### Public Member Functions

- [DecayController](#) (int *Z*, int *A*, double *jInitial*, int *piInitial*, double *energy*, int *initialNeutronNumber*=-1, int *initialNeutronHoleNumber*=-1, int *initialProtonNumber*=-1, int *initialProtonHoleNumber*=-1)
- bool [Decay](#) (double &, double &, double &, double &, double &, double &, double &, double &)
- std::vector< [DecayProduct](#) > [DecayProducts](#) () const
- void [PrintDecays](#) ()

### 7.8.1 Constructor & Destructor Documentation

7.8.1.1 `DecayController::DecayController ( int Z, int A, double jInitial, int piInitial, double energy, int initialNeutronNumber = -1, int initialNeutronHoleNumber = -1, int initialProtonNumber = -1, int initialProtonHoleNumber = -1 ) [inline]`

### 7.8.2 Member Function Documentation

7.8.2.1 `bool DecayController::Decay ( double & neutronEntrance, double & protonEntrance, double & alphaEntrance, double & gammaEntrance, double & neutronTotalWidth, double & protonTotalWidth, double & alphaTotalWidth, double & gammaTotalWidth )`

References [Decayer::AlphaEntranceWidth\(\)](#), [Decayer::AlphaTotalWidth\(\)](#), [Decayer::Decay\(\)](#), [PreEqDecayer::Decay\(\)](#), [Decayer::GammaEntranceWidth\(\)](#), [Decayer::GammaTotalWidth\(\)](#), [Decayer::NeutronEntranceWidth\(\)](#), [Decayer::NeutronTotalWidth\(\)](#), [Decayer::ProtonEntranceWidth\(\)](#), and [Decayer::ProtonTotalWidth\(\)](#).

Referenced by `main()`.

**7.8.2.2** `std::vector<DecayProduct> DecayController::DecayProducts ( ) const` `[inline]`

Referenced by `main()`.

**7.8.2.3** `void DecayController::PrintDecays ( )`

Referenced by `main()`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayController.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayController.cpp`

## 7.9 DecayData Class Reference

```
#include <DecayProduct.h>
```

### Public Member Functions

- `DecayData ( )`
- `DecayData (double energy, double neutronEntranceWidth, double protonEntranceWidth, double alphaEntranceWidth, double gammaEntranceWidth, double neutronTotalWidth, double protonTotalWidth, double alphaTotalWidth, double gammaTotalWidth)`
- `double energy ( ) const`
- `double neutronEntranceWidth ( ) const`
- `double protonEntranceWidth ( ) const`
- `double alphaEntranceWidth ( ) const`
- `double gammaEntranceWidth ( ) const`
- `double neutronTotalWidth ( ) const`
- `double protonTotalWidth ( ) const`
- `double alphaTotalWidth ( ) const`
- `double gammaTotalWidth ( ) const`

### 7.9.1 Constructor & Destructor Documentation

**7.9.1.1** `DecayData::DecayData ( )` `[inline]`

**7.9.1.2** `DecayData::DecayData ( double energy, double neutronEntranceWidth, double protonEntranceWidth, double alphaEntranceWidth, double gammaEntranceWidth, double neutronTotalWidth, double protonTotalWidth, double alphaTotalWidth, double gammaTotalWidth )` `[inline]`

### 7.9.2 Member Function Documentation

**7.9.2.1** `double DecayData::alphaEntranceWidth ( ) const` `[inline]`

**7.9.2.2** `double DecayData::alphaTotalWidth ( ) const` `[inline]`

**7.9.2.3** `double DecayData::energy ( ) const` `[inline]`

**7.9.2.4** `double DecayData::gammaEntranceWidth ( ) const` `[inline]`

7.9.2.5 `double DecayData::gammaTotalWidth ( ) const` `[inline]`

7.9.2.6 `double DecayData::neutronEntranceWidth ( ) const` `[inline]`

7.9.2.7 `double DecayData::neutronTotalWidth ( ) const` `[inline]`

7.9.2.8 `double DecayData::protonEntranceWidth ( ) const` `[inline]`

7.9.2.9 `double DecayData::protonTotalWidth ( ) const` `[inline]`

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayProduct.h>

## 7.10 Decayer Class Reference

```
#include <Decayer.h>
```

### Public Member Functions

- [Decayer](#) (int Z, int A, double jInitial, int piInitial, double energy, double totalWidthForCorrection=0., double uncorrTotalWidthForCorrection=0., double uncorrTotalWidthSqrdForCorrection=0., [Decayer](#) \*widthCorrected-Decayer=NULL)
- [~Decayer](#) ()
- bool [Decay](#) (int &, int &, double &, int &, double &, double &)
- void [PrintFunctions](#) ()
- void [PrintCDF](#) ()
- void [CorrectWidthFluctuations](#) ()
- double [NeutronEntranceWidth](#) () const
- double [ProtonEntranceWidth](#) () const
- double [AlphaEntranceWidth](#) () const
- double [GammaEntranceWidth](#) () const
- double [GammaTotalWidth](#) () const
- double [NeutronTotalWidth](#) () const
- double [AlphaTotalWidth](#) () const
- double [ProtonTotalWidth](#) () const

### Static Public Member Functions

- static void [SetCrossSection](#) (bool isCrossSection)
- static void [SetMaxL](#) (double maxL)
- static double [GetMaxL](#) ()

### Friends

- class [CrossSection](#)

### 7.10.1 Constructor & Destructor Documentation

7.10.1.1 **Decayer::Decayer** ( int *Z*, int *A*, double *jInitial*, int *piInitial*, double *energy*, double *totalWidthForCorrection* = 0., double *uncorrTotalWidthForCorrection* = 0., double *uncorrTotalWidthSqrForCorrection* = 0., **Decayer** \* *widthCorrectedDecayer* = NULL )

References NuclearLevels::FindLevels(), TransitionRateFunc::GroundStateTransmission(), TransitionRateFunc::Integral(), and NuclearMass::QValue().

Referenced by CorrectWidthFluctuations().

7.10.1.2 **Decayer::~~Decayer** ( )

### 7.10.2 Member Function Documentation

7.10.2.1 double **Decayer::AlphaEntranceWidth** ( ) const [inline]

Referenced by DecayController::Decay().

7.10.2.2 double **Decayer::AlphaTotalWidth** ( ) const [inline]

Referenced by DecayController::Decay().

7.10.2.3 void **Decayer::CorrectWidthFluctuations** ( )

References Decayer().

7.10.2.4 bool **Decayer::Decay** ( int & *Z*, int & *A*, double & *jFinal*, int & *piFinal*, double & *excitationEnergy*, double & *decayEnergy* )

References randomSeed.

Referenced by DecayController::Decay().

7.10.2.5 double **Decayer::GammaEntranceWidth** ( ) const [inline]

Referenced by DecayController::Decay().

7.10.2.6 double **Decayer::GammaTotalWidth** ( ) const [inline]

Referenced by DecayController::Decay().

7.10.2.7 static double **Decayer::GetMaxL** ( ) [inline],[static]

Referenced by parseCommandLineForOptions().

7.10.2.8 double **Decayer::NeutronEntranceWidth** ( ) const [inline]

Referenced by DecayController::Decay().

7.10.2.9 `double Decayer::NeutronTotalWidth ( ) const` `[inline]`

Referenced by `DecayController::Decay()`.

7.10.2.10 `void Decayer::PrintCDF ( )`

7.10.2.11 `void Decayer::PrintFunctions ( )`

7.10.2.12 `double Decayer::ProtonEntranceWidth ( ) const` `[inline]`

Referenced by `DecayController::Decay()`.

7.10.2.13 `double Decayer::ProtonTotalWidth ( ) const` `[inline]`

Referenced by `DecayController::Decay()`.

7.10.2.14 `static void Decayer::SetCrossSection ( bool isCrossSection )` `[inline],[static]`

Referenced by `Initialize()`, and `main()`.

7.10.2.15 `static void Decayer::SetMaxL ( double maxL )` `[inline],[static]`

Referenced by `Initialize()`, and `parseCommandLineForOptions()`.

## 7.10.3 Friends And Related Function Documentation

7.10.3.1 `friend class CrossSection` `[friend]`

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Decayer.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Decayer.cpp](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp](#)

## 7.11 DecayProduct Class Reference

```
#include <DecayProduct.h>
```

### Public Member Functions

- [DecayProduct](#) ()
- [DecayProduct](#) (int Z, int A, double J, int Pi, double excitationEnergy, double fragmentEnergyCM, double fragmentEnergy, double fragmentMomentumX, double fragmentMomentumY, double fragmentMomentumZ, int decayType, double particleThetaCM, double particlePhiCM, double particleEnergyCM, double particleEnergy, double particleMomentumX, double particleMomentumY, double particleMomentumZ)

### Public Attributes

- int [Z\\_](#)
- int [A\\_](#)



- int [Pi\\_](#)
- int [particleType\\_](#)
- double [J\\_](#)
- double [excitationEnergy\\_](#)
- double [fragmentEnergyCM\\_](#)
- double [fragmentEnergy\\_](#)
- double [fragmentMomentumX\\_](#)
- double [fragmentMomentumY\\_](#)
- double [fragmentMomentumZ\\_](#)
- double [particleThetaCM\\_](#)
- double [particlePhiCM\\_](#)
- double [particleEnergyCM\\_](#)
- double [particleEnergy\\_](#)
- double [particleMomentumX\\_](#)
- double [particleMomentumY\\_](#)
- double [particleMomentumZ\\_](#)

### 7.11.1 Constructor & Destructor Documentation

7.11.1.1 `DecayProduct::DecayProduct ( )` `[inline]`

7.11.1.2 `DecayProduct::DecayProduct ( int Z, int A, double J, int Pi, double excitationEnergy, double fragmentEnergyCM, double fragmentEnergy, double fragmentMomentumX, double fragmentMomentumY, double fragmentMomentumZ, int decayType, double particleThetaCM, double particlePhiCM, double particleEnergyCM, double particleEnergy, double particleMomentumX, double particleMomentumY, double particleMomentumZ )` `[inline]`

### 7.11.2 Member Data Documentation

7.11.2.1 `int DecayProduct::A_`

Referenced by `boost::serialization::serialize()`.

7.11.2.2 `double DecayProduct::excitationEnergy_`

Referenced by `boost::serialization::serialize()`.

7.11.2.3 `double DecayProduct::fragmentEnergy_`

Referenced by `boost::serialization::serialize()`.

7.11.2.4 `double DecayProduct::fragmentEnergyCM_`

Referenced by `boost::serialization::serialize()`.

7.11.2.5 `double DecayProduct::fragmentMomentumX_`

Referenced by `boost::serialization::serialize()`.

7.11.2.6 `double DecayProduct::fragmentMomentumY_`

Referenced by `boost::serialization::serialize()`.

#### 7.11.2.7 double DecayProduct::fragmentMomentumZ\_

Referenced by boost::serialization::serialize().

#### 7.11.2.8 double DecayProduct::J\_

Referenced by boost::serialization::serialize().

#### 7.11.2.9 double DecayProduct::particleEnergy\_

Referenced by boost::serialization::serialize().

#### 7.11.2.10 double DecayProduct::particleEnergyCM\_

Referenced by boost::serialization::serialize().

#### 7.11.2.11 double DecayProduct::particleMomentumX\_

Referenced by boost::serialization::serialize().

#### 7.11.2.12 double DecayProduct::particleMomentumY\_

Referenced by boost::serialization::serialize().

#### 7.11.2.13 double DecayProduct::particleMomentumZ\_

Referenced by boost::serialization::serialize().

#### 7.11.2.14 double DecayProduct::particlePhiCM\_

Referenced by boost::serialization::serialize().

#### 7.11.2.15 double DecayProduct::particleThetaCM\_

Referenced by boost::serialization::serialize().

#### 7.11.2.16 int DecayProduct::particleType\_

Referenced by boost::serialization::serialize().

#### 7.11.2.17 int DecayProduct::Pi\_

Referenced by boost::serialization::serialize().

#### 7.11.2.18 int DecayProduct::Z\_

Referenced by boost::serialization::serialize().

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[DecayProduct.h](#)

## 7.12 DecayResults Class Reference

```
#include <DecayResults.h>
```

### Public Member Functions

- [DecayResults](#) (int, int, double, int, double, double, int)
- [~DecayResults](#) ()
- void [AddResults](#) (std::vector< std::pair< [DecayData](#), std::vector< [DecayProduct](#) > > > &)

### 7.12.1 Constructor & Destructor Documentation

7.12.1.1 `DecayResults::DecayResults ( int Z, int A, double J, int Pi, double initialEnergyLow, double initialEnergyHigh, int suffixNo )`

References `NuclearMass::FindElement()`.

7.12.1.2 `DecayResults::~~DecayResults ( )`

### 7.12.2 Member Function Documentation

7.12.2.1 `void DecayResults::AddResults ( std::vector< std::pair< DecayData, std::vector< DecayProduct > > > & results )`

Referenced by `main()`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayResults.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayResults.cpp`

## 7.13 EntrancePairs Struct Reference

### Public Member Functions

- [EntrancePairs](#) (int *Z*, int *A*, int *pType*)

### Public Attributes

- int [Z\\_](#)
- int [A\\_](#)
- int [pType\\_](#)

### 7.13.1 Constructor & Destructor Documentation

7.13.1.1 `EntrancePairs::EntrancePairs ( int Z, int A, int pType )` `[inline]`

References `A_`, `pType_`, and `Z_`.

### 7.13.2 Member Data Documentation

#### 7.13.2.1 int EntrancePairs::A\_

Referenced by EntrancePairs().

#### 7.13.2.2 int EntrancePairs::pType\_

Referenced by EntrancePairs().

#### 7.13.2.3 int EntrancePairs::Z\_

Referenced by EntrancePairs().

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/[Sapphire.cpp](#)

## 7.14 std::equal\_to< MassKey > Struct Template Reference

```
#include <NuclearMass.h>
```

### Public Member Functions

- bool [operator\(\)](#) ([MassKey](#) const &left, [MassKey](#) const &right) const

### 7.14.1 Member Function Documentation

#### 7.14.1.1 bool std::equal\_to< MassKey >::operator() ( MassKey const & left, MassKey const & right ) const [inline]

References [MassKey::A\\_](#), and [MassKey::Z\\_](#).

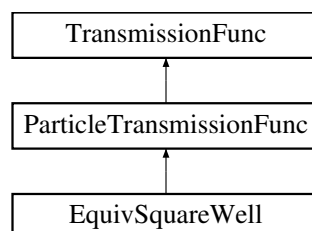
The documentation for this struct was generated from the following file:

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[NuclearMass.h](#)

## 7.15 EquivSquareWell Class Reference

```
#include <EquivSquareWell.h>
```

Inheritance diagram for EquivSquareWell:



## Public Member Functions

- [EquivSquareWell](#) (int *z1*, int *m1*, int *z2*, int *m2*, double *jInitial*, int *piInitial*, double *jFinal*, int *piFinal*, double *spin*, int *parity*, double *maxL*, double *totalWidthForCorrection*, double *uncorrTotalWidthForCorrection*, double *uncorrTotalWidthSqrdForCorrection*, [TransmissionFunc](#) \*previous)
- [~EquivSquareWell](#) ()
- double [CalcTransmission](#) (double, int, double)

## Additional Inherited Members

### 7.15.1 Constructor & Destructor Documentation

7.15.1.1 `EquivSquareWell::EquivSquareWell ( int z1, int m1, int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double spin, int parity, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrdForCorrection, TransmissionFunc * previous )` `[inline]`

References [ParticleTransmissionFunc::redmass\\_](#), [ParticleTransmissionFunc::z1\\_](#), and [TransmissionFunc::z2\\_](#).

7.15.1.2 `EquivSquareWell::~~EquivSquareWell ( )` `[inline]`

### 7.15.2 Member Function Documentation

7.15.2.1 `double EquivSquareWell::CalcTransmission ( double s, int l, double energy )` `[virtual]`

Implements [ParticleTransmissionFunc](#).

References [hbarc](#), [CoulFunc::Penetrability\(\)](#), [pi](#), [ParticleTransmissionFunc::redmass\\_](#), and [uconv](#).

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/EquivSquareWell.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/EquivSquareWell.cpp](#)

## 7.16 GammaTransition Class Reference

```
#include <NuclearLevels.h>
```

## Public Member Functions

- [GammaTransition](#) (int *levelIndex*, double *energy*, double *probability*)

## Public Attributes

- int [levelIndex\\_](#)
- double [energy\\_](#)
- double [probability\\_](#)

### 7.16.1 Constructor & Destructor Documentation

7.16.1.1 `GammaTransition::GammaTransition ( int levelIndex, double energy, double probability )` `[inline]`

### 7.16.2 Member Data Documentation

7.16.2.1 double GammaTransition::energy\_

7.16.2.2 int GammaTransition::levelIndex\_

7.16.2.3 double GammaTransition::probability\_

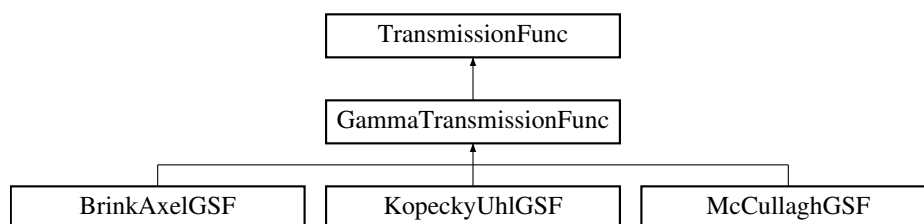
The documentation for this class was generated from the following file:

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[NuclearLevels.h](#)

## 7.17 GammaTransmissionFunc Class Reference

```
#include <GammaTransmissionFunc.h>
```

Inheritance diagram for GammaTransmissionFunc:



### Public Member Functions

- [GammaTransmissionFunc](#) (int, int, double, int, double, int, double, double, double, double, [TransmissionFunc](#) \*)
- virtual [~GammaTransmissionFunc](#) ()
- bool [IsValid](#) ()
- double [operator\(\)](#) (double)
- virtual double [CalcStrengthFunction](#) (double)=0

### Static Public Member Functions

- static [GammaTransmissionFunc](#) \* [CreateGammaTransmissionFunc](#) (int, int, double, int, double, int, double, [LevelDensity](#) \*, double, double, double, [TransmissionFunc](#) \*, double)
- static void [InitializeGDRParameters](#) (std::string)
- static void [SetEGDRType](#) (int)
- static void [SetPorterThomas](#) (bool)

### Protected Attributes

- [GDRParameters](#) [gdrParameters\\_](#)

### Static Protected Attributes

- static [GDRTable](#) [gdrTable\\_](#)
- static int [egdrType\\_](#)
- static const int [mgdrType\\_](#) =0
- static const int [egqrType\\_](#) =0
- static bool [porterThomas\\_](#)

### 7.17.1 Constructor & Destructor Documentation

7.17.1.1 **GammaTransmissionFunc::GammaTransmissionFunc** ( *int* *z2*, *int* *m2*, *double* *jInitial*, *int* *piInitial*, *double* *jFinal*, *int* *piFinal*, *double* *maxL*, *double* *totalWidthForCorrection*, *double* *uncorrTotalWidthForCorrection*, *double* *uncorrTotalWidthSqrForCorrection*, *TransmissionFunc* \* *previous* )

References [NuclearMass::FindMass\(\)](#), [fstruc](#), [gdrParameters\\_](#), [gdrTable\\_](#), [hbarc](#), [GDRParameters::kSigmaGamma\\_](#), [TransmissionFunc::m2\\_](#), [pi](#), and [TransmissionFunc::z2\\_](#).

7.17.1.2 **virtual GammaTransmissionFunc::~~GammaTransmissionFunc** ( ) [\[inline\]](#), [\[virtual\]](#)

### 7.17.2 Member Function Documentation

7.17.2.1 **virtual double GammaTransmissionFunc::CalcStrengthFunction** ( *double* ) [\[pure virtual\]](#)

Implemented in [BrinkAxelGSF](#), [KopeckyUhlGSF](#), and [McCullaghGSF](#).

Referenced by [operator\(\)\(\)](#).

7.17.2.2 **GammaTransmissionFunc \* GammaTransmissionFunc::CreateGammaTransmissionFunc** ( *int* *z2*, *int* *m2*, *double* *jInitial*, *int* *piInitial*, *double* *jFinal*, *int* *piFinal*, *double* *maxL*, *LevelDensity* \* *levelDensity*, *double* *totalWidthForCorrection*, *double* *uncorrTotalWidthForCorrection*, *double* *uncorrTotalWidthSqrForCorrection*, *TransmissionFunc* \* *previous*, *double* *compoundE* ) [\[static\]](#)

References [egdrType\\_](#), [egqrType\\_](#), and [mgdrType\\_](#).

Referenced by [TransitionRateFunc::TransitionRateFunc\(\)](#).

7.17.2.3 **void GammaTransmissionFunc::InitializeGDRParameters** ( *std::string* *filename* ) [\[static\]](#)

References [gdrTable\\_](#).

Referenced by [Initialize\(\)](#).

7.17.2.4 **bool GammaTransmissionFunc::IsValid** ( ) [\[inline\]](#), [\[virtual\]](#)

Implements [TransmissionFunc](#).

7.17.2.5 **double GammaTransmissionFunc::operator()** ( *double* *energy* ) [\[virtual\]](#)

Implements [TransmissionFunc](#).

References [CalcStrengthFunction\(\)](#), [TransmissionFunc::maxL\\_](#), [pi](#), [porterThomas\\_](#), [TransmissionFunc::previous\\_](#), [randomSeed](#), [TransmissionFunc::totalWidthForCorrection\\_](#), [TransmissionFunc::uncorrTotalWidthForCorrection\\_](#), and [TransmissionFunc::uncorrTotalWidthSqrForCorrection\\_](#).

7.17.2.6 **void GammaTransmissionFunc::SetEGDRType** ( *int* *type* ) [\[static\]](#)

References [egdrType\\_](#).

Referenced by [Initialize\(\)](#), and [parseCommandLineForOptions\(\)](#).

7.17.2.7 **void GammaTransmissionFunc::SetPorterThomas** ( *bool* *yn* ) [\[static\]](#)

References [porterThomas\\_](#).

Referenced by `Initialize()`, and `parseCommandLineForOptions()`.

### 7.17.3 Member Data Documentation

**7.17.3.1** `int GammaTransmissionFunc::egdrType_` `[static]`, `[protected]`

Referenced by `CreateGammaTransmissionFunc()`, and `SetEGDRType()`.

**7.17.3.2** `const int GammaTransmissionFunc::egqrType_ = 0` `[static]`, `[protected]`

Referenced by `CreateGammaTransmissionFunc()`.

**7.17.3.3** **GDRParameters** `GammaTransmissionFunc::gdrParameters_` `[protected]`

Referenced by `McCullaghGSF::CalcStrengthFunction()`, `KopeckyUhlGSF::CalcStrengthFunction()`, `BrinkAxelGSF::CalcStrengthFunction()`, and `GammaTransmissionFunc()`.

**7.17.3.4** **GDRTable** `GammaTransmissionFunc::gdrTable_` `[static]`, `[protected]`

Referenced by `GammaTransmissionFunc()`, and `InitializeGDRParameters()`.

**7.17.3.5** `const int GammaTransmissionFunc::mgdrType_ = 0` `[static]`, `[protected]`

Referenced by `CreateGammaTransmissionFunc()`.

**7.17.3.6** `bool GammaTransmissionFunc::porterThomas_` `[static]`, `[protected]`

Referenced by `operator()()`, and `SetPorterThomas()`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/GammaTransmissionFunc.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/GammaTransmissionFunc.cpp`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp`

## 7.18 GDRParameters Class Reference

```
#include <GammaTransmissionFunc.h>
```

### Public Member Functions

- [GDRParameters](#) ()
- [GDRParameters](#) (double eta, double E1, double W1, double kSigmaGamma1, double E2, double W2, double kSigmaGamma2)

### Public Attributes

- double [eta\\_](#)
- double [E\\_](#) [2]
- double [W\\_](#) [2]
- double [kSigmaGamma\\_](#) [2]



### 7.18.1 Constructor & Destructor Documentation

#### 7.18.1.1 GDRParameters::GDRParameters ( ) [inline]

References `E_`, and `W_`.

#### 7.18.1.2 GDRParameters::GDRParameters ( double *eta*, double *E1*, double *W1*, double *kSigmaGamma1*, double *E2*, double *W2*, double *kSigmaGamma2* ) [inline]

References `E_`, `kSigmaGamma_`, and `W_`.

### 7.18.2 Member Data Documentation

#### 7.18.2.1 double GDRParameters::E\_[2]

Referenced by `McCullaghGSF::CalcStrengthFunction()`, `KopeckyUhlGSF::CalcStrengthFunction()`, `BrinkAxelGSF::CalcStrengthFunction()`, and `GDRParameters()`.

#### 7.18.2.2 double GDRParameters::eta\_

#### 7.18.2.3 double GDRParameters::kSigmaGamma\_[2]

Referenced by `McCullaghGSF::CalcStrengthFunction()`, `KopeckyUhlGSF::CalcStrengthFunction()`, `BrinkAxelGSF::CalcStrengthFunction()`, `GammaTransmissionFunc::GammaTransmissionFunc()`, and `GDRParameters()`.

#### 7.18.2.4 double GDRParameters::W\_[2]

Referenced by `KopeckyUhlGSF::CalcStrengthFunction()`, and `GDRParameters()`.

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/GammaTransmissionFunc.h>

## 7.19 gsl\_partfunc\_params Struct Reference

### Public Attributes

- double [temperature](#)
- [LevelDensity](#) \* [density](#)

### 7.19.1 Member Data Documentation

#### 7.19.1.1 LevelDensity\* gsl\_partfunc\_params::density

Referenced by `gsl_partfunc_integrand()`, `LevelDensity::operator()()`, and `LevelDensity::TotalLevelDensity()`.

#### 7.19.1.2 double gsl\_partfunc\_params::temperature

Referenced by `gsl_partfunc_integrand()`.

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CrossSection.cpp>

## 7.20 gsl\_reactionrate\_params Struct Reference

### Public Attributes

- double [temperature](#)
- TGraph \* [graph](#)
- bool [useSpline](#)

### 7.20.1 Member Data Documentation

#### 7.20.1.1 TGraph\* gsl\_reactionrate\_params::graph

Referenced by [gsl\\_reactionrate\\_integrand\(\)](#).

#### 7.20.1.2 double gsl\_reactionrate\_params::temperature

Referenced by [gsl\\_reactionrate\\_integrand\(\)](#).

#### 7.20.1.3 bool gsl\_reactionrate\_params::useSpline

Referenced by [gsl\\_reactionrate\\_integrand\(\)](#).

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CrossSection.cpp](#)

## 7.21 std::tr1::hash< MassKey > Struct Template Reference

```
#include <NuclearMass.h>
```

### Public Member Functions

- `std::size_t operator\(\) (MassKey const &key) const`

### 7.21.1 Member Function Documentation

#### 7.21.1.1 `std::size_t std::tr1::hash< MassKey >::operator() ( MassKey const & key ) const` `[inline]`

References [MassKey::A\\_](#), and [MassKey::Z\\_](#).

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearMass.h](#)

## 7.22 InitialNucleusData Class Reference

```
#include <SapphireMPITypes.h>
```

## Public Member Functions

- [InitialNucleusData](#) ()
- [InitialNucleusData](#) (int [Z](#), int [A](#), double [J](#), int [Pi](#), double [lowEnergy](#), double [highEnergy](#), bool [preEq](#))
- bool [preEq](#) () const
- int [Z](#) () const
- int [A](#) () const
- int [Pi](#) () const
- double [J](#) () const
- double [lowEnergy](#) () const
- double [highEnergy](#) () const

## Friends

- class [boost::serialization::access](#)

### 7.22.1 Constructor & Destructor Documentation

7.22.1.1 [InitialNucleusData::InitialNucleusData](#) ( ) [\[inline\]](#)

7.22.1.2 [InitialNucleusData::InitialNucleusData](#) ( int *Z*, int *A*, double *J*, int *Pi*, double *lowEnergy*, double *highEnergy*, bool *preEq* ) [\[inline\]](#)

### 7.22.2 Member Function Documentation

7.22.2.1 int [InitialNucleusData::A](#) ( ) const [\[inline\]](#)

7.22.2.2 double [InitialNucleusData::highEnergy](#) ( ) const [\[inline\]](#)

7.22.2.3 double [InitialNucleusData::J](#) ( ) const [\[inline\]](#)

7.22.2.4 double [InitialNucleusData::lowEnergy](#) ( ) const [\[inline\]](#)

7.22.2.5 int [InitialNucleusData::Pi](#) ( ) const [\[inline\]](#)

7.22.2.6 bool [InitialNucleusData::preEq](#) ( ) const [\[inline\]](#)

7.22.2.7 int [InitialNucleusData::Z](#) ( ) const [\[inline\]](#)

### 7.22.3 Friends And Related Function Documentation

7.22.3.1 friend class [boost::serialization::access](#) [\[friend\]](#)

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/SapphireMPITypes.h](#)

## 7.23 int\_double\_pair\_compare Struct Reference

```
#include <CrossSection.h>
```

## Public Member Functions

- bool [operator\(\)](#) (const [int\\_double\\_pair](#) &lhs, [int\\_double\\_pair](#) const &rhs)

### 7.23.1 Member Function Documentation

7.23.1.1 bool [int\\_double\\_pair\\_compare::operator\(\)](#) ( const [int\\_double\\_pair](#) & lhs, [int\\_double\\_pair](#) const & rhs )  
[inline]

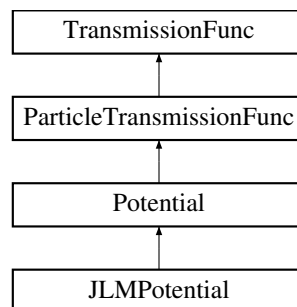
The documentation for this struct was generated from the following file:

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[CrossSection.h](#)

## 7.24 JLMPotential Class Reference

```
#include <JLMPotential.h>
```

Inheritance diagram for JLMPotential:



## Public Member Functions

- [JLMPotential](#) (int, int, int, int, double, int, double, int, double, int, double, double, double, double, double, [TransmissionFunc](#) \*)
- double [CalculateDensity](#) (double, int) const
- [std::complex](#)< double > [Calculate](#) (double, int, double, double, double, double) const

## Static Public Member Functions

- static double [GetA](#) (int i, int j)

## Additional Inherited Members

### 7.24.1 Constructor & Destructor Documentation

7.24.1.1 [JLMPotential::JLMPotential](#) ( int *z1*, int *m1*, int *z2*, int *m2*, double *jInitial*, int *piInitial*, double *jFinal*, int *piFinal*, double *spin*, int *parity*, double *maxL*, double *totalWidthForCorrection*, double *uncorrTotalWidthForCorrection*, double *uncorrTotalWidthSqrdForCorrection*, [TransmissionFunc](#) \* *previous* )

References [Potential::boundaryRadius\\_](#), [Potential::coulombRadius\\_](#), [TransmissionFunc::m2\\_](#), [pi](#), and [TransmissionFunc::z2\\_](#).

## 7.24.2 Member Function Documentation

7.24.2.1 `std::complex< double > JLMPotential::Calculate ( double r, int l, double s, double j, double E ) const`  
`[virtual]`

Implements [Potential](#).

References `CalculateDensity()`, `Potential::coulombRadius_`, `fstruc`, `Potential::GetZ1Z2()`, and `hbarc`.

7.24.2.2 `double JLMPotential::CalculateDensity ( double r, int which ) const`

Referenced by `Calculate()`.

7.24.2.3 `static double JLMPotential::GetA ( int i, int j )` `[inline], [static]`

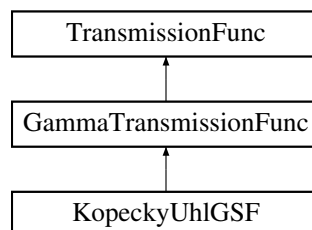
The documentation for this class was generated from the following files:

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/JLMPotential.h>
- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/JLMPotential.cpp>

## 7.25 KopecUhlGSF Class Reference

```
#include <KopecUhlGSF.h>
```

Inheritance diagram for KopecUhlGSF:



### Public Member Functions

- [KopecUhlGSF](#) (int, int, double, int, double, int, double, double, double, double, [TransmissionFunc](#) \*, [LevelDensity](#) \*, double)
- double [CalcStrengthFunction](#) (double)

### Additional Inherited Members

## 7.25.1 Constructor & Destructor Documentation

7.25.1.1 `KopecUhlGSF::KopecUhlGSF ( int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrForCorrection, TransmissionFunc * previous, LevelDensity * levelDensity, double compoundE )`

References `LevelDensity::backshift_`, `LevelDensity::CalcDensityParam()`, and `NuclearMass::QValue()`.

## 7.25.2 Member Function Documentation

### 7.25.2.1 `double KopecckyUhlGSF::CalcStrengthFunction ( double energy ) [virtual]`

Implements [GammaTransmissionFunc](#).

References `GDRParameters::E_`, `GammaTransmissionFunc::gdrParameters_`, `GDRParameters::kSigmaGamma_`, `pi`, and `GDRParameters::W_`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/KopecckyUhlGSF.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/KopecckyUhlGSF.cpp`

## 7.26 Level Class Reference

```
#include <NuclearLevels.h>
```

### Public Member Functions

- [Level](#) (double J, int Pi, double energy)

### Public Attributes

- int [Pi\\_](#)
- double [J\\_](#)
- double [energy\\_](#)
- `std::vector< GammaTransition > gammas_`

## 7.26.1 Constructor & Destructor Documentation

### 7.26.1.1 `Level::Level ( double J, int Pi, double energy ) [inline]`

## 7.26.2 Member Data Documentation

### 7.26.2.1 `double Level::energy_`

Referenced by `CrossSection::CrossSection()`.

### 7.26.2.2 `std::vector<GammaTransition> Level::gammas_`

Referenced by `LevelsContainer::LevelsContainer()`.

### 7.26.2.3 `double Level::J_`

Referenced by `CrossSection::CrossSection()`.

### 7.26.2.4 `int Level::Pi_`

Referenced by `CrossSection::CrossSection()`.

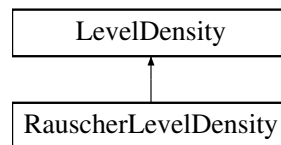
The documentation for this class was generated from the following file:

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearLevels.h`

## 7.27 LevelDensity Class Reference

```
#include <LevelDensity.h>
```

Inheritance diagram for LevelDensity:



### Public Member Functions

- [LevelDensity](#) (int Z, int A, double J)
- virtual [~LevelDensity](#) ()
- double [operator\(\)](#) (double)
- double [TotalLevelDensity](#) (double)

### Protected Member Functions

- virtual void [CalcBackShift](#) ()=0
- virtual double [CalcDensityParam](#) (double)=0
- virtual double [CalcNuclearTemp](#) (double)=0
- void [CalcConstantTempTerms](#) ()

### Protected Attributes

- int [Z\\_](#)
- int [A\\_](#)
- double [J\\_](#)
- double [backshift\\_](#)
- double [criticalU\\_](#)
- double [constAngTerm\\_](#)
- double [nuclearTemp\\_](#)
- double [e0\\_](#)

### Static Protected Attributes

- static constexpr double [zeta\\_](#) = 1.0
- static constexpr double [r0\\_](#) = 1.25

### Friends

- class [KopeckyUhlGSF](#)

### 7.27.1 Constructor & Destructor Documentation

7.27.1.1 [LevelDensity::LevelDensity](#) ( int Z, int A, double J ) `[inline]`

References [criticalU\\_](#).

7.27.1.2 `virtual LevelDensity::~~LevelDensity ( ) [inline],[virtual]`

## 7.27.2 Member Function Documentation

7.27.2.1 `virtual void LevelDensity::CalcBackShift ( ) [protected],[pure virtual]`

Implemented in [RauscherLevelDensity](#).

7.27.2.2 `void LevelDensity::CalcConstantTempTerms ( ) [protected]`

References `A_`, `backshift_`, `CalcDensityParam()`, `CalcNuclearTemp()`, `constAngTerm_`, `criticalU_`, `e0_`, `hbarc`, `J_`, `nuclearTemp_`, `r0_`, `uconv`, and `zeta_`.

Referenced by `RauscherLevelDensity::RauscherLevelDensity()`.

7.27.2.3 `virtual double LevelDensity::CalcDensityParam ( double ) [protected],[pure virtual]`

Implemented in [RauscherLevelDensity](#).

Referenced by `CalcConstantTempTerms()`, `KopeckyUhlGSF::KopeckyUhlGSF()`, `operator()()`, and `TotalLevelDensity()`.

7.27.2.4 `virtual double LevelDensity::CalcNuclearTemp ( double ) [protected],[pure virtual]`

Implemented in [RauscherLevelDensity](#).

Referenced by `CalcConstantTempTerms()`.

7.27.2.5 `double LevelDensity::operator() ( double E )`

References `A_`, `backshift_`, `CalcDensityParam()`, `constAngTerm_`, `criticalU_`, `gsl_partfunc_params::density`, `e0_`, `hbarc`, `J_`, `nuclearTemp_`, `r0_`, `uconv`, and `zeta_`.

7.27.2.6 `double LevelDensity::TotalLevelDensity ( double E )`

References `A_`, `backshift_`, `CalcDensityParam()`, `criticalU_`, `gsl_partfunc_params::density`, `e0_`, `hbarc`, `nuclearTemp_`, `r0_`, `uconv`, and `zeta_`.

Referenced by `TransitionRateFunc::CalcTotalLevelDensity()`.

## 7.27.3 Friends And Related Function Documentation

7.27.3.1 `friend class KopeckyUhlGSF [friend]`

## 7.27.4 Member Data Documentation

7.27.4.1 `int LevelDensity::A_ [protected]`

Referenced by `RauscherLevelDensity::CalcBackShift()`, `CalcConstantTempTerms()`, `operator()()`, and `TotalLevelDensity()`.



7.27.4.2 `double LevelDensity::backshift_` `[protected]`

Referenced by `RauscherLevelDensity::CalcBackShift()`, `CalcConstantTempTerms()`, `KopeckyUhlGSF::KopeckyUhlGSF()`, `operator()()`, and `TotalLevelDensity()`.

7.27.4.3 `double LevelDensity::constAngTerm_` `[protected]`

Referenced by `CalcConstantTempTerms()`, and `operator()()`.

7.27.4.4 `double LevelDensity::criticalU_` `[protected]`

Referenced by `CalcConstantTempTerms()`, `LevelDensity()`, `operator()()`, and `TotalLevelDensity()`.

7.27.4.5 `double LevelDensity::e0_` `[protected]`

Referenced by `CalcConstantTempTerms()`, `operator()()`, and `TotalLevelDensity()`.

7.27.4.6 `double LevelDensity::J_` `[protected]`

Referenced by `CalcConstantTempTerms()`, and `operator()()`.

7.27.4.7 `double LevelDensity::nuclearTemp_` `[protected]`

Referenced by `CalcConstantTempTerms()`, `operator()()`, and `TotalLevelDensity()`.

7.27.4.8 `constexpr double LevelDensity::r0_ = 1.25` `[static]`, `[protected]`

Referenced by `CalcConstantTempTerms()`, `operator()()`, and `TotalLevelDensity()`.

7.27.4.9 `int LevelDensity::Z_` `[protected]`

Referenced by `RauscherLevelDensity::CalcBackShift()`.

7.27.4.10 `constexpr double LevelDensity::zeta_ = 1.0` `[static]`, `[protected]`

Referenced by `CalcConstantTempTerms()`, `operator()()`, and `TotalLevelDensity()`.

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/LevelDensity.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/LevelDensity.cpp](#)

## 7.28 LevelsContainer Class Reference

```
#include <NuclearLevels.h>
```

### Public Member Functions

- [LevelsContainer](#) ()
- [LevelsContainer](#) (std::istream &, int, int)

## Public Attributes

- `std::vector< Level > levels_`

## 7.28.1 Constructor & Destructor Documentation

7.28.1.1 `LevelsContainer::LevelsContainer ( ) [inline]`

7.28.1.2 `LevelsContainer::LevelsContainer ( std::istream & in, int numLevels, int numComplete )`

References `Level::gammas_`, and `levels_`.

## 7.28.2 Member Data Documentation

7.28.2.1 `std::vector<Level> LevelsContainer::levels_`

Referenced by `NuclearLevels::InitializeLevels()`, and `LevelsContainer()`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearLevels.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearLevels.cpp`

## 7.29 MassEntry Class Reference

```
#include <NuclearMass.h>
```

## Public Member Functions

- `MassEntry ( )`
- `MassEntry (double expMass, double thMass, double microEnergyCorr, unsigned int mask)`

## Public Attributes

- double `expMass_`
- double `thMass_`
- double `microEnergyCorr_`
- unsigned int `mask_`

## 7.29.1 Constructor & Destructor Documentation

7.29.1.1 `MassEntry::MassEntry ( ) [inline]`

References `expMass_`, `mask_`, `microEnergyCorr_`, and `thMass_`.

7.29.1.2 `MassEntry::MassEntry ( double expMass, double thMass, double microEnergyCorr, unsigned int mask ) [inline]`

## 7.29.2 Member Data Documentation

## 7.29.2.1 double MassEntry::expMass\_

Referenced by MassEntry().

## 7.29.2.2 unsigned int MassEntry::mask\_

Referenced by MassEntry().

## 7.29.2.3 double MassEntry::microEnergyCorr\_

Referenced by MassEntry().

## 7.29.2.4 double MassEntry::thMass\_

Referenced by MassEntry().

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[NuclearMass.h](#)

## 7.30 MassKey Class Reference

```
#include <NuclearMass.h>
```

### Public Member Functions

- [MassKey](#) (int Z, int A)
- bool [operator<](#) (const [MassKey](#) &right) const

### Public Attributes

- int [Z\\_](#)
- int [A\\_](#)

### 7.30.1 Constructor & Destructor Documentation

7.30.1.1 [MassKey::MassKey](#) ( int Z, int A ) `[inline]`

### 7.30.2 Member Function Documentation

7.30.2.1 bool [MassKey::operator<](#) ( const [MassKey](#) &right ) const `[inline]`

References [A\\_](#), and [Z\\_](#).

### 7.30.3 Member Data Documentation

7.30.3.1 int [MassKey::A\\_](#)

Referenced by `std::tr1::hash< MassKey >::operator()()`, `std::equal_to< MassKey >::operator()()`, and `operator<()`.

### 7.30.3.2 `int MassKey::Z_`

Referenced by `std::tr1::hash< MassKey >::operator()()`, `std::equal_to< MassKey >::operator()()`, and `operator<()`.

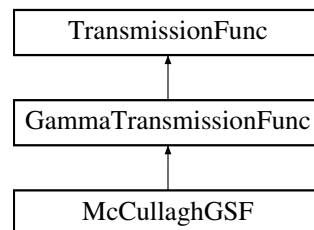
The documentation for this class was generated from the following file:

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearMass.h>

## 7.31 McCullaghGSF Class Reference

```
#include <McCullaghGSF.h>
```

Inheritance diagram for McCullaghGSF:



### Public Member Functions

- [McCullaghGSF](#) (int, int, double, int, double, int, double, double, double, double, [TransmissionFunc](#) \*)
- double [CalcStrengthFunction](#) (double)

### Additional Inherited Members

#### 7.31.1 Constructor & Destructor Documentation

7.31.1.1 `McCullaghGSF::McCullaghGSF ( int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrForCorrection, TransmissionFunc * previous )`

#### 7.31.2 Member Function Documentation

7.31.2.1 `double McCullaghGSF::CalcStrengthFunction ( double energy )` `[virtual]`

Implements [GammaTransmissionFunc](#).

References `GDRParameters::E_`, `GammaTransmissionFunc::gdrParameters_`, `GDRParameters::kSigmaGamma_`, and `TransmissionFunc::maxL_`.

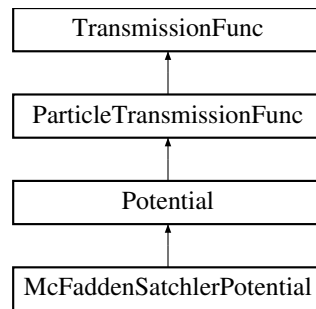
The documentation for this class was generated from the following files:

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/McCullaghGSF.h>
- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McCullaghGSF.cpp>

## 7.32 McFaddenSatchlerPotential Class Reference

```
#include <McFaddenSatchlerPotential.h>
```

Inheritance diagram for McFaddenSatchlerPotential:



## Public Member Functions

- [McFaddenSatchlerPotential](#) (int, int, int, int, double, int, double, int, double, int, double, double, double, double, double, [TransmissionFunc](#) \*)
- `std::complex< double >` [Calculate](#) (double, int, double, double, double, double) const

## Additional Inherited Members

### 7.32.1 Constructor & Destructor Documentation

7.32.1.1 `McFaddenSatchlerPotential::McFaddenSatchlerPotential ( int z1, int m1, int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double spin, int parity, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrForCorrection, TransmissionFunc * previous )`

References `Potential::boundaryRadius_`, `Potential::coulombRadius_`, and `TransmissionFunc::m2_`.

### 7.32.2 Member Function Documentation

7.32.2.1 `std::complex< double > McFaddenSatchlerPotential::Calculate ( double r, int l, double s, double j, double energy ) const` [virtual]

Implements [Potential](#).

References `Potential::coulombRadius_`, `fstruc`, `Potential::GetZ1Z2()`, and `hbarc`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/McFaddenSatchlerPotential.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McFaddenSatchlerPotential.cpp`

## 7.33 NuclearLevels Class Reference

```
#include <NuclearLevels.h>
```

## Static Public Member Functions

- static void [InitializeLevels](#) (std::string levelsDirectory, std::string spinFile)
- static void [PrintLevels](#) (int, int)
- static std::vector< [Level](#) > [FindLevels](#) (int, int)

### 7.33.1 Member Function Documentation

7.33.1.1 `std::vector< Level > NuclearLevels::FindLevels ( int Z, int A )` [static]

Referenced by `CrossSection::CrossSection()`, `Decayer::Decayer()`, and `TransitionRateFunc::TransitionRateFunc()`.

7.33.1.2 `void NuclearLevels::InitializeLevels ( std::string levelsDirectory, std::string spinFile )` [static]

References `LevelsContainer::levels_`.

Referenced by `Initialize()`.

7.33.1.3 `void NuclearLevels::PrintLevels ( int Z, int A )` [static]

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearLevels.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearLevels.cpp](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp](#)

## 7.34 NuclearMass Class Reference

```
#include <NuclearMass.h>
```

### Static Public Member Functions

- static void [InitializeElements](#) ()
- static void [InitializeMasses](#) (std::string)
- static int [FindZ](#) (std::string)
- static std::string [FindElement](#) (int)
- static bool [FindMass](#) (int, int, double &)
- static bool [MassDifference](#) (int, int, int, int, double &)
- static bool [QValue](#) (int, int, int, int, double &)
- static bool [NeutronPairingGap](#) (int, int, double &)
- static bool [ProtonPairingGap](#) (int, int, double &)
- static bool [MicroEnergyCorr](#) (int, int, double &)
- static bool [HighestBoundEnergy](#) (int, int, double &)
- static double [CalculateLDMMass](#) (int, int)

### 7.34.1 Member Function Documentation

7.34.1.1 `double NuclearMass::CalculateLDMMass ( int Z, int A )` [static]

Calculates liquid drop model mass based on TALYS parametrization.

References `uconv`.

7.34.1.2 `std::string NuclearMass::FindElement ( int Z )` [static]

Referenced by `DecayResults::DecayResults()`, `CrossSection::PrintCrossSections()`, `CrossSection::PrintReactionRates()`, and `CrossSection::PrintTransmissionTerms()`.

**7.34.1.3** `bool NuclearMass::FindMass ( int Z, int A, double & M ) [static]`

References HAS\_EXP\_MASS, and HAS\_TH\_MASS.

Referenced by GammaTransmissionFunc::GammaTransmissionFunc(), MassDifference(), ParticleTransmissionFunc::ParticleTransmissionFunc(), and QValue().

**7.34.1.4** `int NuclearMass::FindZ ( std::string element ) [static]`

Referenced by parseCommandLineForDecay(), and parseCommandLineForXS().

**7.34.1.5** `bool NuclearMass::HighestBoundEnergy ( int Z, int A, double & energy ) [static]`

References QValue().

Referenced by TransitionRateFunc::TransitionRateFunc().

**7.34.1.6** `void NuclearMass::InitializeElements ( ) [static]`

Referenced by Initialize().

**7.34.1.7** `void NuclearMass::InitializeMasses ( std::string filename ) [static]`

References HAS\_EXP\_MASS, HAS\_TH\_MASS, and uconv.

Referenced by Initialize().

**7.34.1.8** `bool NuclearMass::MassDifference ( int Z1, int A1, int Z2, int A2, double & difference ) [static]`

References FindMass().

Referenced by NeutronPairingGap(), ProtonPairingGap(), and QValue().

**7.34.1.9** `bool NuclearMass::MicroEnergyCorr ( int Z, int A, double & correction ) [static]`

Referenced by RauscherLevelDensity::RauscherLevelDensity().

**7.34.1.10** `bool NuclearMass::NeutronPairingGap ( int Z, int A, double & pairingGap ) [static]`

References MassDifference().

Referenced by RauscherLevelDensity::CalcBackShift().

**7.34.1.11** `bool NuclearMass::ProtonPairingGap ( int Z, int A, double & pairingGap ) [static]`

References MassDifference().

Referenced by RauscherLevelDensity::CalcBackShift().

**7.34.1.12** `bool NuclearMass::QValue ( int Z1, int A1, int Z2, int A2, double & qValue ) [static]`

References FindMass(), and MassDifference().

Referenced by CrossSection::CrossSection(), Decayer::Decayer(), HighestBoundEnergy(), KopeckyUhlGSF::KopeckyUhlGSF(), ParticleHoleLevelDensity::ParticleHoleLevelDensity(), and PreEqDecayer::PreEqDecayer().

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearMass.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearMass.cpp](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp](#)

## 7.35 ODE\_integration Class Reference

```
#include <ode_int.H>
```

### Public Member Functions

- [ODE\\_integration](#) (const [std::complex](#)< double > &l\_1, const [std::complex](#)< double > &two\_eta\_1)
- void [operator\(\)](#) (const [std::complex](#)< double > &r0, const [std::complex](#)< double > &u0, const [std::complex](#)< double > &du0, const [std::complex](#)< double > &r, [std::complex](#)< double > &u, [std::complex](#)< double > &du) const

### 7.35.1 Constructor & Destructor Documentation

7.35.1.1 [ODE\\_integration::ODE\\_integration](#) ( const [std::complex](#)< double > &l\_1, const [std::complex](#)< double > &two\_eta\_1 ) [inline]

### 7.35.2 Member Function Documentation

7.35.2.1 void [ODE\\_integration::operator\(\)](#) ( const [std::complex](#)< double > &r0, const [std::complex](#)< double > &u0, const [std::complex](#)< double > &du0, const [std::complex](#)< double > &r, [std::complex](#)< double > &u, [std::complex](#)< double > &du ) const

References [inf\\_norm\(\)](#), and [precision](#).

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

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/ode\\_int.H](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/ode\\_int.cpp](#)

## 7.36 ParticleHoleLevelDensity Class Reference

```
#include <ParticleHoleLevelDensity.h>
```

### Public Member Functions

- [ParticleHoleLevelDensity](#) (int, int, double, int, int, int, int)
- double [operator\(\)](#) (double energy, bool correct=true, bool spin=false)
- double [PauliCorrection](#) (int, int, int, int)
- double [PairingCorrection](#) (double energy)
- double [gNu](#) () const
- double [gPi](#) () const
- double [FiniteDepth](#) (double)



### 7.36.1 Constructor & Destructor Documentation

7.36.1.1 `ParticleHoleLevelDensity::ParticleHoleLevelDensity ( int Z, int A, double J, int neutronNumber, int neutronHoleNumber, int protonNumber, int protonHoleNumber )`

References `PauliCorrection()`, and `NuclearMass::QValue()`.

### 7.36.2 Member Function Documentation

7.36.2.1 `double ParticleHoleLevelDensity::FiniteDepth ( double energy )`

Referenced by `operator>()()`.

7.36.2.2 `double ParticleHoleLevelDensity::gNu ( ) const` `[inline]`

7.36.2.3 `double ParticleHoleLevelDensity::gPi ( ) const` `[inline]`

7.36.2.4 `double ParticleHoleLevelDensity::operator() ( double energy, bool correct = true, bool spin = false )`

References `FiniteDepth()`, and `PairingCorrection()`.

7.36.2.5 `double ParticleHoleLevelDensity::PairingCorrection ( double energy )`

Referenced by `operator>()()`.

7.36.2.6 `double ParticleHoleLevelDensity::PauliCorrection ( int neutronNumber, int neutronHoleNumber, int protonNumber, int protonHoleNumber )`

Referenced by `ParticleHoleLevelDensity()`.

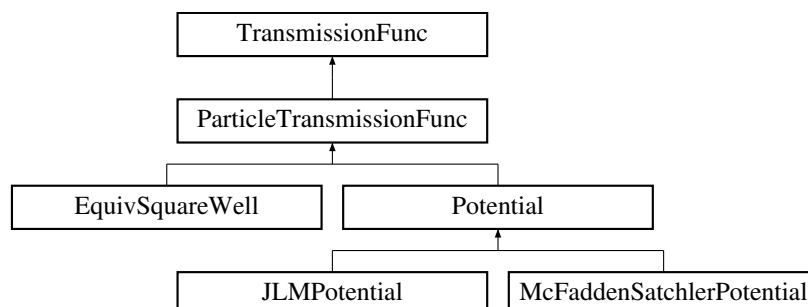
The documentation for this class was generated from the following files:

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ParticleHoleLevelDensity.h>
- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleHoleLevelDensity.cpp>

## 7.37 ParticleTransmissionFunc Class Reference

```
#include <ParticleTransmissionFunc.h>
```

Inheritance diagram for ParticleTransmissionFunc:



## Public Member Functions

- [ParticleTransmissionFunc](#) (int z1, int m1, int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double spin, int parity, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrdForCorrection, [TransmissionFunc](#) \*previous)
- virtual [~ParticleTransmissionFunc](#) ()
- bool [IsValid](#) ()
- double [operator\(\)](#) (double)
- double [operator\(\)](#) (double, int)
- void [CalcSLDependentFunctions](#) (double, std::map< [SLPair](#), double > &)

## Static Public Member Functions

- static [ParticleTransmissionFunc](#) \* [CreateParticleTransmissionFunc](#) (int, int, int, int, double, int, double, int, double, int, double, double, double, double, [TransmissionFunc](#) \*)
- static void [SetAlphaFormalism](#) (int formalism)
- static void [SetNeutronFormalism](#) (int formalism)
- static void [SetProtonFormalism](#) (int formalism)
- static void [SetPorterThomas](#) (bool)

## Protected Member Functions

- virtual double [CalcTransmission](#) (double, int, double)=0

## Protected Attributes

- int [z1\\_](#)
- int [m1\\_](#)
- int [pType\\_](#)
- int [parity\\_](#)
- double [redmass\\_](#)
- double [spin\\_](#)

### 7.37.1 Constructor & Destructor Documentation

7.37.1.1 [ParticleTransmissionFunc::ParticleTransmissionFunc](#) ( int *z1*, int *m1*, int *z2*, int *m2*, double *jInitial*, int *piInitial*, double *jFinal*, int *piFinal*, double *spin*, int *parity*, double *maxL*, double *totalWidthForCorrection*, double *uncorrTotalWidthForCorrection*, double *uncorrTotalWidthSqrdForCorrection*, [TransmissionFunc](#) \* *previous* )  
[inline]

References [NuclearMass::FindMass\(\)](#), [m1\\_](#), [TransmissionFunc::m2\\_](#), [pType\\_](#), [redmass\\_](#), [uconv](#), [z1\\_](#), and [TransmissionFunc::z2\\_](#).

7.37.1.2 virtual [ParticleTransmissionFunc::~~ParticleTransmissionFunc](#) ( ) [inline],[virtual]

### 7.37.2 Member Function Documentation

7.37.2.1 void [ParticleTransmissionFunc::CalcSLDependentFunctions](#) ( double *energy*, std::map< [SLPair](#), double > & *functions* )

References [CalcTransmission\(\)](#), [TransmissionFunc::jFinal\\_](#), [TransmissionFunc::jInitial\\_](#), [TransmissionFunc::maxL\\_](#), [parity\\_](#), [TransmissionFunc::piFinal\\_](#), [TransmissionFunc::piInitial\\_](#), and [spin\\_](#).

Referenced by [operator\(\)\(\)](#).

**7.37.2.2** `virtual double ParticleTransmissionFunc::CalcTransmission ( double , int , double )` `[protected]`, `[pure virtual]`

Implemented in [EquivSquareWell](#), and [Potential](#).

Referenced by `CalcSLDependentFunctions()`.

**7.37.2.3** `ParticleTransmissionFunc * ParticleTransmissionFunc::CreateParticleTransmissionFunc ( int z1, int m1, int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double spin, int parity, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrdForCorrection, TransmissionFunc * previous )` `[static]`

Referenced by `TransitionRateFunc::TransitionRateFunc()`.

**7.37.2.4** `bool ParticleTransmissionFunc::IsValid ( )` `[inline]`, `[virtual]`

Implements [TransmissionFunc](#).

References `pType_`, and `redmass_`.

**7.37.2.5** `double ParticleTransmissionFunc::operator() ( double energy )` `[virtual]`

Implements [TransmissionFunc](#).

References `CalcSLDependentFunctions()`, `TransmissionFunc::previous_`, `randomSeed`, `TransmissionFunc::totalWidthForCorrection_`, `TransmissionFunc::uncorrTotalWidthForCorrection_`, and `TransmissionFunc::uncorrTotalWidthSqrdForCorrection_`.

**7.37.2.6** `double ParticleTransmissionFunc::operator() ( double energy, int which )`

References `CalcSLDependentFunctions()`, `TransmissionFunc::previous_`, `TransmissionFunc::totalWidthForCorrection_`, `TransmissionFunc::uncorrTotalWidthForCorrection_`, and `TransmissionFunc::uncorrTotalWidthSqrdForCorrection_`.

**7.37.2.7** `static void ParticleTransmissionFunc::SetAlphaFormalism ( int formalism )` `[inline]`, `[static]`

Referenced by `Initialize()`, and `parseCommandLineForOptions()`.

**7.37.2.8** `static void ParticleTransmissionFunc::SetNeutronFormalism ( int formalism )` `[inline]`, `[static]`

Referenced by `Initialize()`, and `parseCommandLineForOptions()`.

**7.37.2.9** `void ParticleTransmissionFunc::SetPorterThomas ( bool yn )` `[static]`

Referenced by `Initialize()`, and `parseCommandLineForOptions()`.

**7.37.2.10** `static void ParticleTransmissionFunc::SetProtonFormalism ( int formalism )` `[inline]`, `[static]`

Referenced by `Initialize()`, and `parseCommandLineForOptions()`.

### 7.37.3 Member Data Documentation

#### 7.37.3.1 `int ParticleTransmissionFunc::m1_` [protected]

Referenced by `Potential::CalcTransmission()`, and `ParticleTransmissionFunc()`.

#### 7.37.3.2 `int ParticleTransmissionFunc::parity_` [protected]

Referenced by `CalcSLDependentFunctions()`.

#### 7.37.3.3 `int ParticleTransmissionFunc::pType_` [protected]

Referenced by `IsValid()`, and `ParticleTransmissionFunc()`.

#### 7.37.3.4 `double ParticleTransmissionFunc::redmass_` [protected]

Referenced by `EquivSquareWell::CalcTransmission()`, `EquivSquareWell::EquivSquareWell()`, `Potential::GetRedMass()`, `IsValid()`, and `ParticleTransmissionFunc()`.

#### 7.37.3.5 `double ParticleTransmissionFunc::spin_` [protected]

Referenced by `CalcSLDependentFunctions()`.

#### 7.37.3.6 `int ParticleTransmissionFunc::z1_` [protected]

Referenced by `EquivSquareWell::EquivSquareWell()`, `Potential::GetZ1Z2()`, and `ParticleTransmissionFunc()`.

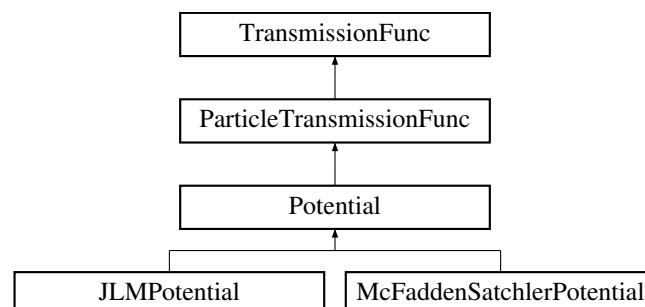
The documentation for this class was generated from the following files:

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ParticleTransmissionFunc.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleTransmissionFunc.cpp`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp`

## 7.38 Potential Class Reference

```
#include <Potential.h>
```

Inheritance diagram for Potential:



## Public Member Functions

- [Potential](#) (int, int, int, int, double, int, double, int, double, int, double, double, double, double, [TransmissionFunc](#) \*)
- [~Potential](#) ()
- int [GetZ1Z2](#) () const
- double [GetBoundaryRadius](#) () const
- double [GetRedMass](#) () const
- double [GetRMax](#) () const
- virtual `std::complex< double >` [Calculate](#) (double, int, double, double, double) const =0
- `std::complex< double >` [CalcBeta](#) (double, int, double, double, double) const
- double [CalcTransmission](#) (double, int, double)
- void [NormalizeInternally](#) (std::vector< `std::complex< double >` > &, double) const
- void [NormalizeOverAllSpace](#) (std::vector< `std::complex< double >` > &, double) const
- std::vector< `std::complex< double >` > [Solve](#) (double, int, double, double, double) const

## Protected Attributes

- `CoulFunc` \* [coulFunc\\_](#)
- double [boundaryRadius\\_](#)
- double [coulombRadius\\_](#)

## Additional Inherited Members

### 7.38.1 Constructor & Destructor Documentation

**7.38.1.1** `Potential::Potential ( int z1, int m1, int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double spin, int parity, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrForCorrection, TransmissionFunc * previous )`

References [coulFunc\\_](#), and [GetRedMass\(\)](#).

**7.38.1.2** `Potential::~~Potential ( )`

References [coulFunc\\_](#).

### 7.38.2 Member Function Documentation

**7.38.2.1** `std::complex< double > Potential::CalcBeta ( double energy, int l, double s, double j, double rStep ) const`

References [GetBoundaryRadius\(\)](#), [NormalizeInternally\(\)](#), and [Solve\(\)](#).

Referenced by [CalcTransmission\(\)](#).

**7.38.2.2** `double Potential::CalcTransmission ( double s, int l, double energy ) [virtual]`

Implements [ParticleTransmissionFunc](#).

References [CalcBeta\(\)](#), [coulFunc\\_](#), [CoulWaves::dF](#), [CoulWaves::dG](#), [CoulWaves::F](#), [CoulWaves::G](#), [GetBoundaryRadius\(\)](#), [GetRedMass\(\)](#), [hbarc](#), [TransmissionFunc::jInitial\\_](#), [ParticleTransmissionFunc::m1\\_](#), and [uconv](#).

**7.38.2.3** `virtual std::complex<double> Potential::Calculate ( double , int , double , double , double ) const` `[pure virtual]`

Implemented in [JLMPotential](#), and [McFaddenSatchlerPotential](#).

Referenced by `Solve()`.

**7.38.2.4** `double Potential::GetBoundaryRadius ( ) const` `[inline]`

References `boundaryRadius_`.

Referenced by `CalcBeta()`, `CalcTransmission()`, `NormalizeInternally()`, and `Solve()`.

**7.38.2.5** `double Potential::GetRedMass ( ) const` `[inline]`

References `ParticleTransmissionFunc::redmass_`.

Referenced by `CalcTransmission()`, `Potential()`, and `Solve()`.

**7.38.2.6** `double Potential::GetRMax ( ) const` `[inline]`

References `boundaryRadius_`.

**7.38.2.7** `int Potential::GetZ1Z2 ( ) const` `[inline]`

References `ParticleTransmissionFunc::z1_`, and `TransmissionFunc::z2_`.

Referenced by `JLMPotential::Calculate()`, and `McFaddenSatchlerPotential::Calculate()`.

**7.38.2.8** `void Potential::NormalizeInternally ( std::vector< std::complex< double > > & waveFunction, double rStep ) const`

References `GetBoundaryRadius()`.

Referenced by `CalcBeta()`.

**7.38.2.9** `void Potential::NormalizeOverAllSpace ( std::vector< std::complex< double > > & waveFunction, double rStep ) const`

**7.38.2.10** `std::vector< std::complex< double > > Potential::Solve ( double energy, int L, double S, double J, double rStep ) const`

References `Calculate()`, `GetBoundaryRadius()`, `GetRedMass()`, `hbarc`, and `uconv`.

Referenced by `CalcBeta()`.

### 7.38.3 Member Data Documentation

**7.38.3.1** `double Potential::boundaryRadius_` `[protected]`

Referenced by `GetBoundaryRadius()`, `GetRMax()`, `JLMPotential::JLMPotential()`, and `McFaddenSatchlerPotential::McFaddenSatchlerPotential()`.

### 7.38.3.2 CoulFunc\* Potential::coulFunc\_ [protected]

Referenced by CalcTransmission(), Potential(), and ~Potential().

### 7.38.3.3 double Potential::coulombRadius\_ [protected]

Referenced by JLMPotential::Calculate(), McFaddenSatchlerPotential::Calculate(), JLMPotential::JLMPotential(), and McFaddenSatchlerPotential::McFaddenSatchlerPotential().

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[Potential.h](#)
- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/[Potential.cpp](#)

## 7.39 PreEqCDFEntry Class Reference

```
#include <PreEqDecayer.h>
```

### Public Member Functions

- [PreEqCDFEntry](#) (int pairIndex, double energy, double value)

### Public Attributes

- int [pairIndex\\_](#)
- double [energy\\_](#)
- double [value\\_](#)

### 7.39.1 Constructor & Destructor Documentation

7.39.1.1 [PreEqCDFEntry::PreEqCDFEntry \( int pairIndex, double energy, double value \)](#) [inline]

### 7.39.2 Member Data Documentation

7.39.2.1 double [PreEqCDFEntry::energy\\_](#)

7.39.2.2 int [PreEqCDFEntry::pairIndex\\_](#)

7.39.2.3 double [PreEqCDFEntry::value\\_](#)

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[PreEqDecayer.h](#)

## 7.40 PreEqDecayer Class Reference

```
#include <PreEqDecayer.h>
```

## Public Member Functions

- [PreEqDecayer](#) (int initialNeutronNumber, int initialNeutronHoleNumber, int initialProtonNumber, int initialProtonHoleNumber, int Z, int A, double jInitial, int piInitial, double energy)
- [~PreEqDecayer](#) ()
- bool [Decay](#) (int &, int &, double &, int &, int &, int &, int &, double &, double &)
- void [PrintCDF](#) ()

## Static Public Member Functions

- static void [SetCrossSection](#) (bool isCrossSection)
- static void [SetMaxL](#) (double maxL)
- static double [GetMaxL](#) ()

### 7.40.1 Constructor & Destructor Documentation

7.40.1.1 `PreEqDecayer::PreEqDecayer ( int initialNeutronNumber, int initialNeutronHoleNumber, int initialProtonNumber, int initialProtonHoleNumber, int Z, int A, double jInitial, int piInitial, double energy )`

References `PreEqTransitionRateFunc::Integral()`, and `NuclearMass::QValue()`.

7.40.1.2 `PreEqDecayer::~~PreEqDecayer ( )`

### 7.40.2 Member Function Documentation

7.40.2.1 `bool PreEqDecayer::Decay ( int & Z, int & A, double & jFinal, int & piFinal, int & neutronNumber, int & neutronHoleNumber, int & protonNumber, int & protonHoleNumber, double & excitationEnergy, double & decayEnergy )`

References `randomSeed`.

Referenced by `DecayController::Decay()`.

7.40.2.2 `static double PreEqDecayer::GetMaxL ( )` `[inline],[static]`

7.40.2.3 `void PreEqDecayer::PrintCDF ( )`

7.40.2.4 `static void PreEqDecayer::SetCrossSection ( bool isCrossSection )` `[inline],[static]`

Referenced by `Initialize()`.

7.40.2.5 `static void PreEqDecayer::SetMaxL ( double maxL )` `[inline],[static]`

Referenced by `Initialize()`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqDecayer.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqDecayer.cpp`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp`

## 7.41 PreEqSpinRatePair Class Reference

```
#include <PreEqDecayer.h>
```



## Public Member Functions

- [PreEqSpinRatePair](#) (int neutronNumber, int neutronHoleNumber, int protonNumber, int protonHoleNumber, int Z, int A, double spin, int parity, double qValue, [PreEqTransitionRateFunc](#) \*rateFunc, double integral)

## Public Attributes

- [PreEqTransitionRateFunc](#) \* rateFunc\_
- int neutronNumber\_
- int neutronHoleNumber\_
- int protonNumber\_
- int protonHoleNumber\_
- int Z\_
- int A\_
- int parity\_
- double spin\_
- double qValue\_
- double integral\_

### 7.41.1 Constructor & Destructor Documentation

- 7.41.1.1 [PreEqSpinRatePair::PreEqSpinRatePair](#) ( int *neutronNumber*, int *neutronHoleNumber*, int *protonNumber*, int *protonHoleNumber*, int *Z*, int *A*, double *spin*, int *parity*, double *qValue*, [PreEqTransitionRateFunc](#) \* *rateFunc*, double *integral* ) [inline]

### 7.41.2 Member Data Documentation

- 7.41.2.1 int [PreEqSpinRatePair::A\\_](#)
- 7.41.2.2 double [PreEqSpinRatePair::integral\\_](#)
- 7.41.2.3 int [PreEqSpinRatePair::neutronHoleNumber\\_](#)
- 7.41.2.4 int [PreEqSpinRatePair::neutronNumber\\_](#)
- 7.41.2.5 int [PreEqSpinRatePair::parity\\_](#)
- 7.41.2.6 int [PreEqSpinRatePair::protonHoleNumber\\_](#)
- 7.41.2.7 int [PreEqSpinRatePair::protonNumber\\_](#)
- 7.41.2.8 double [PreEqSpinRatePair::qValue\\_](#)
- 7.41.2.9 [PreEqTransitionRateFunc](#)\* [PreEqSpinRatePair::rateFunc\\_](#)
- 7.41.2.10 double [PreEqSpinRatePair::spin\\_](#)
- 7.41.2.11 int [PreEqSpinRatePair::Z\\_](#)

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

- /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/[PreEqDecayer.h](#)

## 7.42 PreEqTransitionRateFunc Class Reference

```
#include <PreEqTransitionRateFunc.h>
```

### Public Member Functions

- [PreEqTransitionRateFunc](#) (int, int, int, int, int, int, int, int, int, int, int, int, double, int, double, int, double, int, double, double, double)
- [~PreEqTransitionRateFunc](#) ()
- double [Integral](#) () const
- std::vector< [XYPair](#) > const [CumulativeSum](#) ()

### 7.42.1 Constructor & Destructor Documentation

7.42.1.1 [PreEqTransitionRateFunc::PreEqTransitionRateFunc](#) ( int *z1*, int *m1*, int *z2*, int *m2*, int *initialNeutronNumber*, int *initialNeutronHoleNumber*, int *initialProtonNumber*, int *initialProtonHoleNumber*, int *finalNeutronNumber*, int *finalNeutronHoleNumber*, int *finalProtonNumber*, int *finalProtonHoleNumber*, double *jInitial*, int *piInitial*, double *jFinal*, int *piFinal*, double *spin*, int *parity*, double *maxL*, double *compoundE*, double *qValue* )

7.42.1.2 [PreEqTransitionRateFunc::~~PreEqTransitionRateFunc](#) ( ) [inline]

### 7.42.2 Member Function Documentation

7.42.2.1 [std::vector<XYPair> const PreEqTransitionRateFunc::CumulativeSum](#) ( ) [inline]

7.42.2.2 [double PreEqTransitionRateFunc::Integral](#) ( ) const [inline]

Referenced by [PreEqDecayer::PreEqDecayer](#)().

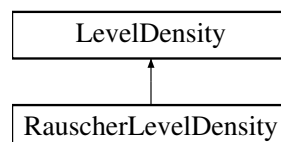
The documentation for this class was generated from the following files:

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqTransitionRateFunc.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqTransitionRateFunc.cpp](#)

## 7.43 RauscherLevelDensity Class Reference

```
#include <RauscherLevelDensity.h>
```

Inheritance diagram for [RauscherLevelDensity](#):



### Public Member Functions

- [RauscherLevelDensity](#) (int *Z*, int *A*, double *J*)
- [~RauscherLevelDensity](#) ()
- void [CalcBackShift](#) ()
- double [CalcDensityParam](#) (double)
- double [CalcNuclearTemp](#) (double)

## Additional Inherited Members

### 7.43.1 Constructor & Destructor Documentation

7.43.1.1 `RauscherLevelDensity::RauscherLevelDensity ( int Z, int A, double J )` `[inline]`

References `CalcBackShift()`, `LevelDensity::CalcConstantTempTerms()`, and `NuclearMass::MicroEnergyCorr()`.

7.43.1.2 `RauscherLevelDensity::~~RauscherLevelDensity ( )` `[inline]`

### 7.43.2 Member Function Documentation

7.43.2.1 `void RauscherLevelDensity::CalcBackShift ( )` `[virtual]`

Implements [LevelDensity](#).

References `LevelDensity::A_`, `LevelDensity::backshift_`, `NuclearMass::NeutronPairingGap()`, `NuclearMass::ProtonPairingGap()`, and `LevelDensity::Z_`.

Referenced by `RauscherLevelDensity()`.

7.43.2.2 `double RauscherLevelDensity::CalcDensityParam ( double u )` `[virtual]`

Implements [LevelDensity](#).

Referenced by `CalcNuclearTemp()`.

7.43.2.3 `double RauscherLevelDensity::CalcNuclearTemp ( double u )` `[virtual]`

Implements [LevelDensity](#).

References `CalcDensityParam()`.

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

- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/RauscherLevelDensity.h`
- `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/RauscherLevelDensity.cpp`

## 7.44 SLPair Class Reference

```
#include <ParticleTransmissionFunc.h>
```

### Public Member Functions

- [SLPair](#) (double s, int l)
- bool [operator<](#) (const [SLPair](#) &right) const

### Public Attributes

- double [s\\_](#)
- int [l\\_](#)

### 7.44.1 Constructor & Destructor Documentation

7.44.1.1 `SLPair::SLPair ( double s, int I ) [inline]`

### 7.44.2 Member Function Documentation

7.44.2.1 `bool SLPair::operator< ( const SLPair & right ) const [inline]`

References `I_`, and `s_`.

### 7.44.3 Member Data Documentation

7.44.3.1 `int SLPair::I_`

Referenced by `operator<()`.

7.44.3.2 `double SLPair::s_`

Referenced by `operator<()`.

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ParticleTransmissionFunc.h>

## 7.45 SpinRatePair Class Reference

```
#include <Decayer.h>
```

### Public Member Functions

- [SpinRatePair](#) (int Z, int A, double spin, int parity, double qValue, [TransitionRateFunc](#) \*rateFunc, double integral)

### Public Attributes

- [TransitionRateFunc](#) \* [rateFunc\\_](#)
- int [Z\\_](#)
- int [A\\_](#)
- int [parity\\_](#)
- double [spin\\_](#)
- double [qValue\\_](#)
- double [integral\\_](#)

### 7.45.1 Constructor & Destructor Documentation

7.45.1.1 `SpinRatePair::SpinRatePair ( int Z, int A, double spin, int parity, double qValue, TransitionRateFunc * rateFunc, double integral ) [inline]`

### 7.45.2 Member Data Documentation

7.45.2.1 `int SpinRatePair::A_`

7.45.2.2 double SpinRatePair::integral\_

7.45.2.3 int SpinRatePair::parity\_

7.45.2.4 double SpinRatePair::qValue\_

7.45.2.5 TransitionRateFunc\* SpinRatePair::rateFunc\_

7.45.2.6 double SpinRatePair::spin\_

7.45.2.7 int SpinRatePair::Z\_

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Decayer.h>

## 7.46 TransitionRateFunc Class Reference

```
#include <TransitionRateFunc.h>
```

### Public Member Functions

- [TransitionRateFunc](#) (int, int, int, int, double, int, double, int, double, int, double, double, double, double, double, double, double, [TransitionRateFunc](#) \*, bool)
- [~TransitionRateFunc](#) ()
- std::vector< [XYPair](#) > const [Function](#) ()
- std::vector< [XYPair](#) > const [CumulativeSum](#) ()
- double [Integral](#) () const
- double [CalcLevelDensity](#) (double energy)
- double [CalcTransmissionFunc](#) (double energy)
- double [CalcTotalLevelDensity](#) (double energy)
- double [ExclusiveBranching](#) () const
- double [GroundStateTransmission](#) () const

### Static Public Member Functions

- static void [SetGammaCutoffEnergy](#) (double energy)
- static double [GetGammaCutoffEnergy](#) ()

### 7.46.1 Constructor & Destructor Documentation

7.46.1.1 [TransitionRateFunc::TransitionRateFunc](#) ( int z1, int m1, int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double spin, int parity, double maxL, double compoundE, double qValue, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrForCorrection, [TransitionRateFunc](#) \* previous, bool isCrossSection )

References [CalcLevelDensity\(\)](#), [CalcTransmissionFunc\(\)](#), [GammaTransmissionFunc::CreateGammaTransmissionFunc\(\)](#), [ParticleTransmissionFunc::CreateParticleTransmissionFunc\(\)](#), [NuclearLevels::FindLevels\(\)](#), [NuclearMass::HighestBoundEnergy\(\)](#), and [TransmissionFunc::IsValid\(\)](#).

7.46.1.2 `TransitionRateFunc::~~TransitionRateFunc ( )` `[inline]`

## 7.46.2 Member Function Documentation

7.46.2.1 `double TransitionRateFunc::CalcLevelDensity ( double energy )` `[inline]`

Referenced by `TransitionRateFunc()`.

7.46.2.2 `double TransitionRateFunc::CalcTotalLevelDensity ( double energy )` `[inline]`

References `LevelDensity::TotalLevelDensity()`.

7.46.2.3 `double TransitionRateFunc::CalcTransmissionFunc ( double energy )` `[inline]`

Referenced by `TransitionRateFunc()`.

7.46.2.4 `std::vector<XYPair> const TransitionRateFunc::CumulativeSum ( )` `[inline]`

7.46.2.5 `double TransitionRateFunc::ExclusiveBranching ( ) const` `[inline]`

7.46.2.6 `std::vector<XYPair> const TransitionRateFunc::Function ( )` `[inline]`

7.46.2.7 `static double TransitionRateFunc::GetGammaCutoffEnergy ( )` `[inline],[static]`

Referenced by `parseCommandLineForOptions()`.

7.46.2.8 `double TransitionRateFunc::GroundStateTransmission ( ) const` `[inline]`

Referenced by `Decayer::Decayer()`.

7.46.2.9 `double TransitionRateFunc::Integral ( ) const` `[inline]`

Referenced by `Decayer::Decayer()`.

7.46.2.10 `static void TransitionRateFunc::SetGammaCutoffEnergy ( double energy )` `[inline],[static]`

Referenced by `CrossSection::Calculate()`, `Initialize()`, and `parseCommandLineForOptions()`.

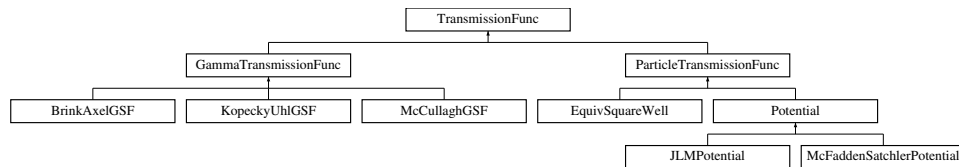
The documentation for this class was generated from the following files:

- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransitionRateFunc.h](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp](#)
- [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/TransitionRateFunc.cpp](#)

## 7.47 TransmissionFunc Class Reference

```
#include <TransmissionFunc.h>
```

Inheritance diagram for `TransmissionFunc`:



## Public Member Functions

- [TransmissionFunc](#) (int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrdForCorrection, [TransmissionFunc](#) \*previous)
- virtual [~TransmissionFunc](#) ()
- virtual double [operator\(\)](#) (double)=0
- virtual bool [IsValid](#) ()=0

## Protected Attributes

- int [z2\\_](#)
- int [m2\\_](#)
- int [piInitial\\_](#)
- int [piFinal\\_](#)
- double [jInitial\\_](#)
- double [jFinal\\_](#)
- double [maxL\\_](#)
- double [totalWidthForCorrection\\_](#)
- double [uncorrTotalWidthForCorrection\\_](#)
- double [uncorrTotalWidthSqrdForCorrection\\_](#)
- [TransmissionFunc](#) \* [previous\\_](#)

### 7.47.1 Constructor & Destructor Documentation

7.47.1.1 `TransmissionFunc::TransmissionFunc ( int z2, int m2, double jInitial, int piInitial, double jFinal, int piFinal, double maxL, double totalWidthForCorrection, double uncorrTotalWidthForCorrection, double uncorrTotalWidthSqrdForCorrection, TransmissionFunc * previous ) [inline]`

7.47.1.2 `virtual TransmissionFunc::~~TransmissionFunc ( ) [inline],[virtual]`

### 7.47.2 Member Function Documentation

7.47.2.1 `virtual bool TransmissionFunc::IsValid ( ) [pure virtual]`

Implemented in [GammaTransmissionFunc](#), and [ParticleTransmissionFunc](#).

Referenced by `TransitionRateFunc::TransitionRateFunc()`.

7.47.2.2 `virtual double TransmissionFunc::operator() ( double ) [pure virtual]`

Implemented in [GammaTransmissionFunc](#), and [ParticleTransmissionFunc](#).

### 7.47.3 Member Data Documentation

#### 7.47.3.1 `double TransmissionFunc::jFinal_` [protected]

Referenced by `ParticleTransmissionFunc::CalcSLDependentFunctions()`.

#### 7.47.3.2 `double TransmissionFunc::jInitial_` [protected]

Referenced by `ParticleTransmissionFunc::CalcSLDependentFunctions()`, and `Potential::CalcTransmission()`.

#### 7.47.3.3 `int TransmissionFunc::m2_` [protected]

Referenced by `GammaTransmissionFunc::GammaTransmissionFunc()`, `JLMPotential::JLMPotential()`, `McFaddenSatchlerPotential::McFaddenSatchlerPotential()`, and `ParticleTransmissionFunc::ParticleTransmissionFunc()`.

#### 7.47.3.4 `double TransmissionFunc::maxL_` [protected]

Referenced by `ParticleTransmissionFunc::CalcSLDependentFunctions()`, `McCullaghGSF::CalcStrengthFunction()`, `BrinkAxelGSF::CalcStrengthFunction()`, and `GammaTransmissionFunc::operator()()`.

#### 7.47.3.5 `int TransmissionFunc::piFinal_` [protected]

Referenced by `ParticleTransmissionFunc::CalcSLDependentFunctions()`.

#### 7.47.3.6 `int TransmissionFunc::piInitial_` [protected]

Referenced by `ParticleTransmissionFunc::CalcSLDependentFunctions()`.

#### 7.47.3.7 `TransmissionFunc* TransmissionFunc::previous_` [protected]

Referenced by `ParticleTransmissionFunc::operator()()`, and `GammaTransmissionFunc::operator()()`.

#### 7.47.3.8 `double TransmissionFunc::totalWidthForCorrection_` [protected]

Referenced by `ParticleTransmissionFunc::operator()()`, and `GammaTransmissionFunc::operator()()`.

#### 7.47.3.9 `double TransmissionFunc::uncorrTotalWidthForCorrection_` [protected]

Referenced by `ParticleTransmissionFunc::operator()()`, and `GammaTransmissionFunc::operator()()`.

#### 7.47.3.10 `double TransmissionFunc::uncorrTotalWidthSqrdForCorrection_` [protected]

Referenced by `ParticleTransmissionFunc::operator()()`, and `GammaTransmissionFunc::operator()()`.

#### 7.47.3.11 `int TransmissionFunc::z2_` [protected]

Referenced by `EquivSquareWell::EquivSquareWell()`, `GammaTransmissionFunc::GammaTransmissionFunc()`, `Potential::GetZ1Z2()`, `JLMPotential::JLMPotential()`, and `ParticleTransmissionFunc::ParticleTransmissionFunc()`.

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



- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransmissionFunc.h>

## 7.48 XYPair Class Reference

```
#include <TransitionRateFunc.h>
```

### Public Member Functions

- [XYPair](#) (double X, double Y)

### Public Attributes

- double [X\\_](#)
- double [Y\\_](#)

### 7.48.1 Constructor & Destructor Documentation

7.48.1.1 [XYPair::XYPair \( double X, double Y \)](#) `[inline]`

### 7.48.2 Member Data Documentation

7.48.2.1 `double XYPair::X_`

7.48.2.2 `double XYPair::Y_`

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

- </afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransitionRateFunc.h>



## Chapter 8

# File Documentation

### 8.1 CMakeFiles/3.13.2/CompilerIdC/CMakeCCompilerId.c File Reference

#### Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define C_DIALECT`

#### Functions

- `int main (int argc, char *argv[])`

#### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_dialect_default`

#### 8.1.1 Macro Definition Documentation

##### 8.1.1.1 `#define ARCHITECTURE_ID`

##### 8.1.1.2 `#define C_DIALECT`

##### 8.1.1.3 `#define COMPILER_ID ""`

##### 8.1.1.4 `#define DEC( n )`

**Value:**

```

('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))

```

#### 8.1.1.5 #define HEX( n )

**Value:**

```

('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))

```

#### 8.1.1.6 #define PLATFORM\_ID

#### 8.1.1.7 #define STRINGIFY( X ) STRINGIFY\_HELPER(X)

#### 8.1.1.8 #define STRINGIFY\_HELPER( X ) #X

### 8.1.2 Function Documentation

#### 8.1.2.1 int main ( int argc, char \* argv[] )

References info\_arch, info\_compiler, info\_language\_dialect\_default, and info\_platform.

### 8.1.3 Variable Documentation

#### 8.1.3.1 char const\* info\_arch = "INFO" ":" "arch[" ARCHITECTURE\_ID "]"

Referenced by main().

#### 8.1.3.2 char const\* info\_compiler = "INFO" ":" "compiler[" COMPILER\_ID "]"

Referenced by main().

#### 8.1.3.3 const char\* info\_language\_dialect\_default

**Initial value:**

```

=
"INFO" ":" "dialect_default[" C_DIALECT "]"

```

Referenced by main().

#### 8.1.3.4 char const\* info\_platform = "INFO" ":" "platform[" PLATFORM\_ID "]"

Referenced by main().

## 8.2 CMakeFiles/3.13.2/CompilerIdCXX/CompilerIdCXX.cpp File Reference

### Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD __cplusplus`

### Functions

- `int main (int argc, char *argv[])`

### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_dialect_default`

### 8.2.1 Macro Definition Documentation

#### 8.2.1.1 `#define ARCHITECTURE_ID`

#### 8.2.1.2 `#define COMPILER_ID ""`

#### 8.2.1.3 `#define CXX_STD __cplusplus`

#### 8.2.1.4 `#define DEC( n )`

**Value:**

```
( '0' + ((n) / 10000000) % 10 ), \
( '0' + ((n) / 1000000) % 10 ), \
( '0' + ((n) / 100000) % 10 ), \
( '0' + ((n) / 10000) % 10 ), \
( '0' + ((n) / 1000) % 10 ), \
( '0' + ((n) / 100) % 10 ), \
( '0' + ((n) / 10) % 10 ), \
( '0' + ((n) % 10) )
```

#### 8.2.1.5 `#define HEX( n )`

**Value:**

```
( '0' + ((n) >> 28 & 0xF) ), \
( '0' + ((n) >> 24 & 0xF) ), \
( '0' + ((n) >> 20 & 0xF) ), \
( '0' + ((n) >> 16 & 0xF) ), \
( '0' + ((n) >> 12 & 0xF) ), \
( '0' + ((n) >> 8 & 0xF) ), \
( '0' + ((n) >> 4 & 0xF) ), \
( '0' + ((n) & 0xF) )
```

8.2.1.6 `#define PLATFORM_ID`

8.2.1.7 `#define STRINGIFY( X ) STRINGIFY_HELPER(X)`

8.2.1.8 `#define STRINGIFY_HELPER( X ) #X`

## 8.2.2 Function Documentation

8.2.2.1 `int main ( int argc, char * argv[] )`

References `info_compiler`, `info_language_dialect_default`, and `info_platform`.

## 8.2.3 Variable Documentation

8.2.3.1 `char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

8.2.3.2 `char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

8.2.3.3 `const char* info_language_dialect_default`

**Initial value:**

```
= "INFO" ":" "dialect_default["
```

```
    "98"
    "]"
```

8.2.3.4 `char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

## 8.3 CMakeFiles/feature\_tests.c File Reference

### Functions

- `int main (int argc, char **argv)`

### Variables

- `const char features []`

## 8.3.1 Function Documentation

8.3.1.1 `int main ( int argc, char ** argv )`

References `features`.

### 8.3.2 Variable Documentation

#### 8.3.2.1 const char features[]

Referenced by main().

## 8.4 CMakeFiles/feature\_tests.cxx File Reference

### Functions

- int [main](#) (int argc, char \*\*argv)

### Variables

- const char [features](#) []

### 8.4.1 Function Documentation

#### 8.4.1.1 int main ( int *argc*, char \*\* *argv* )

References features.

### 8.4.2 Variable Documentation

#### 8.4.2.1 const char features[]

## 8.5 CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference

```
#include <stdio.h>
#include <omp.h>
```

### Functions

- int [main](#) (void)

### Variables

- const char [ompver\\_str](#) []

### 8.5.1 Function Documentation

#### 8.5.1.1 int main ( void )

References ompver\_str.

## 8.5.2 Variable Documentation

### 8.5.2.1 `const char ompver_str[]`

#### Initial value:

```
= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'P', '-', 'd', 'a', 't', 'e', '[',
    ('0' + ((_OPENMP/100000)%10)),
    ('0' + ((_OPENMP/10000)%10)),
    ('0' + ((_OPENMP/1000)%10)),
    ('0' + ((_OPENMP/100)%10)),
    ('0' + ((_OPENMP/10)%10)),
    ('0' + ((_OPENMP/1)%10)),
    ']', '\0' }
```

Referenced by `main()`.

## 8.6 CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference

```
#include <stdio.h>
#include <omp.h>
```

### Functions

- `int main (void)`

### Variables

- `const char ompver_str []`

## 8.6.1 Function Documentation

### 8.6.1.1 `int main ( void )`

References `ompver_str`.

## 8.6.2 Variable Documentation

### 8.6.2.1 `const char ompver_str[]`

#### Initial value:

```
= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'P', '-', 'd', 'a', 't', 'e', '[',
    ('0' + ((_OPENMP/100000)%10)),
    ('0' + ((_OPENMP/10000)%10)),
    ('0' + ((_OPENMP/1000)%10)),
    ('0' + ((_OPENMP/100)%10)),
    ('0' + ((_OPENMP/10)%10)),
    ('0' + ((_OPENMP/1)%10)),
    ']', '\0' }
```

## 8.7 CMakeFiles/FindOpenMP/OpenMPTryFlag.c File Reference

```
#include <omp.h>
```



## Functions

- int [main](#) (void)

### 8.7.1 Function Documentation

#### 8.7.1.1 int main ( void )

## 8.8 CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp File Reference

```
#include <omp.h>
```

## Functions

- int [main](#) (void)

### 8.8.1 Function Documentation

#### 8.8.1.1 int main ( void )

## 8.9 generated/Sapphire\_config.h File Reference

```
#include <string>
```

## Functions

- std::string [sourceDirectory](#) ()

### 8.9.1 Function Documentation

#### 8.9.1.1 std::string sourceDirectory ( )

Referenced by Initialize().

## 8.10 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/generated/Sapphire\_config.h File Reference

```
#include <string>
```

## Functions

- std::string [sourceDirectory](#) ()

## 8.10.1 Function Documentation

### 8.10.1.1 `std::string sourceDirectory ( )`

## 8.11 `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/complex_functions.H` File Reference

```
#include <complex>
#include <iostream>
#include <cstdlib>
```

### Macros

- `#define SIGN(a) (((a) < 0) ? (-1) : (1))`

### Functions

- `double inf_norm (const std::complex< double > &z)`
- `bool isfinite (const std::complex< double > &z)`
- `std::complex< double > operator+ (const std::complex< double > &z, const int n)`
- `std::complex< double > operator- (const std::complex< double > &z, const int n)`
- `std::complex< double > operator* (const std::complex< double > &z, const int n)`
- `std::complex< double > operator/ (const std::complex< double > &z, const int n)`
- `std::complex< double > operator+ (const int n, const std::complex< double > &z)`
- `std::complex< double > operator- (const int n, const std::complex< double > &z)`
- `std::complex< double > operator* (const int n, const std::complex< double > &z)`
- `std::complex< double > operator/ (const int n, const std::complex< double > &z)`
- `std::complex< double > operator+ (const std::complex< double > &z, const unsigned int n)`
- `std::complex< double > operator- (const std::complex< double > &z, const unsigned int n)`
- `std::complex< double > operator* (const std::complex< double > &z, const unsigned int n)`
- `std::complex< double > operator/ (const std::complex< double > &z, const unsigned int n)`
- `std::complex< double > operator+ (const unsigned int n, const std::complex< double > &z)`
- `std::complex< double > operator- (const unsigned int n, const std::complex< double > &z)`
- `std::complex< double > operator* (const unsigned int n, const std::complex< double > &z)`
- `std::complex< double > operator/ (const unsigned int n, const std::complex< double > &z)`
- `bool operator== (const std::complex< double > &z, const int n)`
- `bool operator!= (const std::complex< double > &z, const int n)`
- `bool operator== (const int n, const std::complex< double > &z)`
- `bool operator!= (const int n, const std::complex< double > &z)`
- `bool operator== (const std::complex< double > &z, const unsigned int n)`
- `bool operator!= (const std::complex< double > &z, const unsigned int n)`
- `bool operator== (const unsigned int n, const std::complex< double > &z)`
- `bool operator!= (const unsigned int n, const std::complex< double > &z)`
- `std::complex< double > expm1 (const std::complex< double > &z)`
- `std::complex< double > log1p (const std::complex< double > &z)`
- `std::complex< double > log_Gamma (const std::complex< double > &z)`
- `std::complex< double > sigma_l_calc (const std::complex< double > &l, const std::complex< double > &eta)`
- `std::complex< double > log_Cl_eta_calc (const std::complex< double > &l, const std::complex< double > &eta)`
- `std::complex< double > log_cut_constant_AS_calc (const int omega, const std::complex< double > &l, const std::complex< double > &eta)`

- `std::complex< double > log_cut_constant_CFa_calc` (const bool is\_it\_normalized, const int omega, const `std::complex< double > &l`, const `std::complex< double > &eta`)
- `std::complex< double > log_cut_constant_CFb_calc` (const bool is\_it\_normalized, const int omega, const `std::complex< double > &l`, const `std::complex< double > &eta`)
- `std::complex< double > sin_chi_calc` (const `std::complex< double > &l`, const `std::complex< double > &eta`)
- `std::complex< double > exp_l_omega_chi_calc` (const int omega, const `std::complex< double > &l`, const `std::complex< double > &eta`)

## Variables

- const double `precision` = 1E-10
- const double `sqrt_precision` = 1E-5

## 8.11.1 Macro Definition Documentation

### 8.11.1.1 `#define SIGN( a ) (((a) < 0) ? (-1) : (1))`

Referenced by `exp_l_omega_chi_calc()`, and `Coulomb_wave_functions::F_dF()`.

## 8.11.2 Function Documentation

### 8.11.2.1 `std::complex<double> exp_l_omega_chi_calc ( const int omega, const std::complex< double > &l, const std::complex< double > &eta )`

References `sigma_l_calc()`, `SIGN`, and `sin_chi_calc()`.

### 8.11.2.2 `std::complex<double> expm1 ( const std::complex< double > &z )`

References `expm1()`.

Referenced by `expm1()`, `log_cut_constant_AS_calc()`, and `log_cut_constant_CFa_calc()`.

### 8.11.2.3 `double inf_norm ( const std::complex< double > &z ) [inline]`

Referenced by `ODE_integration::operator()()`.

### 8.11.2.4 `bool isfinite ( const std::complex< double > &z ) [inline]`

Referenced by `Coulomb_wave_functions::F_dF()`, `Coulomb_wave_functions::G_dG()`, `Coulomb_wave_functions::H_dH()`, `Coulomb_wave_functions::H_dH_scaled()`, and `log_Gamma()`.

### 8.11.2.5 `std::complex<double> log1p ( const std::complex< double > &z )`

References `log1p()`.

Referenced by `log1p()`, `log_cut_constant_AS_calc()`, and `log_cut_constant_CFa_calc()`.

### 8.11.2.6 `std::complex<double> log_Cl_eta_calc ( const std::complex< double > &l, const std::complex< double > &eta )`

References `log_Gamma()`.

Referenced by `log_cut_constant_CFa_calc()`, `log_cut_constant_CFb_calc()`, and `sin_chi_calc()`.

8.11.2.7 `std::complex<double> log_cut_constant_AS_calc ( const int omega, const std::complex< double > & l, const std::complex< double > & eta )`

References `expm1()`, and `log1p()`.

8.11.2.8 `std::complex<double> log_cut_constant_CFa_calc ( const bool is_it_normalized, const int omega, const std::complex< double > & l, const std::complex< double > & eta )`

References `expm1()`, `log1p()`, and `log_Cl_eta_calc()`.

8.11.2.9 `std::complex<double> log_cut_constant_CFb_calc ( const bool is_it_normalized, const int omega, const std::complex< double > & l, const std::complex< double > & eta )`

References `log_Cl_eta_calc()`.

8.11.2.10 `std::complex<double> log_Gamma ( const std::complex< double > & z )`

References `isfinite()`, and `log_Gamma()`.

Referenced by `log_Cl_eta_calc()`, `log_Gamma()`, and `sigma_l_calc()`.

8.11.2.11 `bool operator!= ( const std::complex< double > & z, const int n )` `[inline]`

8.11.2.12 `bool operator!= ( const int n, const std::complex< double > & z )` `[inline]`

8.11.2.13 `bool operator!= ( const std::complex< double > & z, const unsigned int n )` `[inline]`

8.11.2.14 `bool operator!= ( const unsigned int n, const std::complex< double > & z )` `[inline]`

8.11.2.15 `std::complex<double> operator* ( const std::complex< double > & z, const int n )` `[inline]`

8.11.2.16 `std::complex<double> operator* ( const int n, const std::complex< double > & z )` `[inline]`

8.11.2.17 `std::complex<double> operator* ( const std::complex< double > & z, const unsigned int n )` `[inline]`

8.11.2.18 `std::complex<double> operator* ( const unsigned int n, const std::complex< double > & z )` `[inline]`

8.11.2.19 `std::complex<double> operator+ ( const std::complex< double > & z, const int n )` `[inline]`

8.11.2.20 `std::complex<double> operator+ ( const int n, const std::complex< double > & z )` `[inline]`

8.11.2.21 `std::complex<double> operator+ ( const std::complex< double > & z, const unsigned int n )` `[inline]`

8.11.2.22 `std::complex<double> operator+ ( const unsigned int n, const std::complex< double > & z )` `[inline]`

8.11.2.23 `std::complex<double> operator- ( const std::complex< double > & z, const int n )` `[inline]`

8.11.2.24 `std::complex<double> operator- ( const int n, const std::complex< double > & z )` `[inline]`

8.11.2.25 `std::complex<double> operator- ( const std::complex< double > & z, const unsigned int n )` `[inline]`

8.11.2.26 `std::complex<double> operator- ( const unsigned int n, const std::complex< double > & z )` `[inline]`

8.11.2.27 `std::complex<double> operator/ ( const std::complex< double > & z, const int n )` `[inline]`

8.11.2.28 `std::complex<double> operator/ ( const int n, const std::complex< double > & z )` [inline]

8.11.2.29 `std::complex<double> operator/ ( const std::complex< double > & z, const unsigned int n )` [inline]

8.11.2.30 `std::complex<double> operator/ ( const unsigned int n, const std::complex< double > & z )` [inline]

8.11.2.31 `bool operator== ( const std::complex< double > & z, const int n )` [inline]

8.11.2.32 `bool operator== ( const int n, const std::complex< double > & z )` [inline]

8.11.2.33 `bool operator== ( const std::complex< double > & z, const unsigned int n )` [inline]

8.11.2.34 `bool operator== ( const unsigned int n, const std::complex< double > & z )` [inline]

8.11.2.35 `std::complex<double> sigma_l_calc ( const std::complex< double > & l, const std::complex< double > & eta )`

References `log_Gamma()`.

Referenced by `exp_l_omega_chi_calc()`.

8.11.2.36 `std::complex<double> sin_chi_calc ( const std::complex< double > & l, const std::complex< double > & eta )`

References `log_Cl_eta_calc()`.

Referenced by `exp_l_omega_chi_calc()`.

### 8.11.3 Variable Documentation

8.11.3.1 `const double precision = 1E-10`

Referenced by `ODE_integration::operator()()`.

8.11.3.2 `const double sqrt_precision = 1E-5`

Referenced by `Coulomb_wave_functions::F_dF()`, `Coulomb_wave_functions::G_dG()`, `Coulomb_wave_functions::H_dH()`, and `Coulomb_wave_functions::H_dH_scaled()`.

## 8.12 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/cwfcomp.H File Reference

```
#include "ode_int.H"
```

### Classes

- class [Coulomb\\_wave\\_functions](#)

## 8.13 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/ode\_int.H File Reference

```
#include "complex_functions.H"
```

### Classes

- class [ODE\\_integration](#)

## 8.14 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/complex\_functions.cpp File Reference

```
#include "complex_functions.H"
```

### Functions

- [std::complex< double > expm1](#) (const [std::complex< double > &z](#))
- [std::complex< double > log1p](#) (const [std::complex< double > &z](#))
- [std::complex< double > log\\_Gamma](#) (const [std::complex< double > &z](#))
- [std::complex< double > sigma\\_I\\_calc](#) (const [std::complex< double > &l](#), const [std::complex< double > &eta](#))
- [std::complex< double > log\\_Cl\\_eta\\_calc](#) (const [std::complex< double > &l](#), const [std::complex< double > &eta](#))
- [std::complex< double > log\\_cut\\_constant\\_AS\\_calc](#) (const int omega, const [std::complex< double > &l](#), const [std::complex< double > &eta](#))
- [std::complex< double > log\\_cut\\_constant\\_CFa\\_calc](#) (const bool is\_it\_normalized, const int omega, const [std::complex< double > &l](#), const [std::complex< double > &eta](#))
- [std::complex< double > log\\_cut\\_constant\\_CFb\\_calc](#) (const bool is\_it\_normalized, const int omega, const [std::complex< double > &l](#), const [std::complex< double > &eta](#))
- [std::complex< double > sin\\_chi\\_calc](#) (const [std::complex< double > &l](#), const [std::complex< double > &eta](#))
- [std::complex< double > exp\\_I\\_omega\\_chi\\_calc](#) (const int omega, const [std::complex< double > &l](#), const [std::complex< double > &eta](#))

### 8.14.1 Function Documentation

**8.14.1.1** [std::complex<double> exp\\_I\\_omega\\_chi\\_calc](#) ( const int *omega*, const [std::complex< double > &l](#), const [std::complex< double > &eta](#) )

References [sigma\\_I\\_calc\(\)](#), [SIGN](#), and [sin\\_chi\\_calc\(\)](#).

**8.14.1.2** [std::complex<double> expm1](#) ( const [std::complex< double > &z](#) )

References [expm1\(\)](#).

Referenced by [expm1\(\)](#), [log\\_cut\\_constant\\_AS\\_calc\(\)](#), and [log\\_cut\\_constant\\_CFa\\_calc\(\)](#).

**8.14.1.3** `std::complex<double> log1p ( const std::complex< double > & z )`

References `log1p()`.

Referenced by `log1p()`, `log_cut_constant_AS_calc()`, and `log_cut_constant_CFa_calc()`.

**8.14.1.4** `std::complex<double> log_Cl_eta_calc ( const std::complex< double > & l, const std::complex< double > & eta )`

References `log_Gamma()`.

Referenced by `log_cut_constant_CFa_calc()`, `log_cut_constant_CFb_calc()`, and `sin_chi_calc()`.

**8.14.1.5** `std::complex<double> log_cut_constant_AS_calc ( const int omega, const std::complex< double > & l, const std::complex< double > & eta )`

References `expm1()`, and `log1p()`.

**8.14.1.6** `std::complex<double> log_cut_constant_CFa_calc ( const bool is_it_normalized, const int omega, const std::complex< double > & l, const std::complex< double > & eta )`

References `expm1()`, `log1p()`, and `log_Cl_eta_calc()`.

**8.14.1.7** `std::complex<double> log_cut_constant_CFb_calc ( const bool is_it_normalized, const int omega, const std::complex< double > & l, const std::complex< double > & eta )`

References `log_Cl_eta_calc()`.

**8.14.1.8** `std::complex<double> log_Gamma ( const std::complex< double > & z )`

References `isfinite()`, and `log_Gamma()`.

Referenced by `log_Cl_eta_calc()`, `log_Gamma()`, and `sigma_l_calc()`.

**8.14.1.9** `std::complex<double> sigma_l_calc ( const std::complex< double > & l, const std::complex< double > & eta )`

References `log_Gamma()`.

Referenced by `exp_l_omega_chi_calc()`.

**8.14.1.10** `std::complex<double> sin_chi_calc ( const std::complex< double > & l, const std::complex< double > & eta )`

References `log_Cl_eta_calc()`.

Referenced by `exp_l_omega_chi_calc()`.

## 8.15 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/cwfcomp.cpp File Reference

```
#include "cwfcomp.H"
```

## 8.16 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/ode\_int.cpp File Reference

```
#include "ode_int.H"
```

## 8.17 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/BrinkAxelGSF.h File Reference

Header file for the Brink-Axel-GSF class.

```
#include "GammaTransmissionFunc.h"
```

### Classes

- class [BrinkAxelGSF](#)

### 8.17.1 Detailed Description

Header file for the Brink-Axel-GSF class.

## 8.18 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Constants.h File Reference

```
#include <complex>
#include <vector>
#include <cstdlib>
```

### Typedefs

- typedef std::complex< double > [complex](#)
- typedef std::vector< double > [vector\\_r](#)
- typedef std::vector< std::complex< double > > [vector\\_c](#)
- typedef std::vector< std::vector< double > > [matrix\\_r](#)
- typedef std::vector< std::vector< std::complex< double > > > [matrix\\_c](#)
- typedef std::vector< std::vector< std::vector< double > > > [vector\\_matrix\\_r](#)
- typedef std::vector< std::vector< std::vector< std::complex< double > > > > [vector\\_matrix\\_c](#)



## Variables

- const double `pi` =3.141592650
- const double `hbarc` =197.32696310
- const double `uconv` =931.4940880
- const double `fstruc` =1.00/137.0359996790
- const double `boltzConst` =8.6171e-2
- const double `lightSpeedInCmPerS` =29979245800.
- const double `avagadroNum` =6.02214179e23
- const double `eMass` = 548.579894

## 8.18.1 Typedef Documentation

8.18.1.1 `typedef std::complex<double> complex`

8.18.1.2 `typedef std::vector<std::vector<std::complex<double> > > matrix_c`

8.18.1.3 `typedef std::vector<std::vector<double> > matrix_r`

8.18.1.4 `typedef std::vector<std::complex<double> > vector_c`

8.18.1.5 `typedef std::vector<std::vector<std::vector<std::complex<double> > > > vector_matrix_c`

8.18.1.6 `typedef std::vector<std::vector<std::vector<double> > > vector_matrix_r`

8.18.1.7 `typedef std::vector<double> vector_r`

## 8.18.2 Variable Documentation

8.18.2.1 `const double avagadroNum` =6.02214179e23

Referenced by `CrossSection::CalculateReactionRates()`.

8.18.2.2 `const double boltzConst` =8.6171e-2

Referenced by `CrossSection::CalculateReactionRates()`, `gsl_partfunc_integrand()`, and `gsl_reactionrate_integrand()`.

8.18.2.3 `const double eMass` = 548.579894

8.18.2.4 `const double fstruc` =1.00/137.0359996790

Referenced by `JLMPotential::Calculate()`, `McFaddenSatchlerPotential::Calculate()`, `GammaTransmissionFunc::GammaTransmissionFunc()`, and `CoulFunc::operator()`.

8.18.2.5 `const double hbarc` =197.32696310

Referenced by `LevelDensity::CalcConstantTempTerms()`, `Potential::CalcTransmission()`, `EquivSquareWell::CalcTransmission()`, `McFaddenSatchlerPotential::Calculate()`, `JLMPotential::Calculate()`, `CrossSection::CrossSection()`, `GammaTransmissionFunc::GammaTransmissionFunc()`, `LevelDensity::operator()`, `CoulFunc::operator()`, `CoulFunc::Penetrability()`, `CoulFunc::PEShift()`, `Potential::Solve()`, and `LevelDensity::TotalLevelDensity()`.

8.18.2.6 `const double lightSpeedInCmPerS =29979245800.`

Referenced by `CrossSection::CalculateReactionRates()`.

8.18.2.7 `const double pi =3.141592650`

Referenced by `CrossSection::CalcAverageDWaveResWidth()`, `CrossSection::CalcAveragePWaveResWidth()`, `CrossSection::CalcAverageSWaveResWidth()`, `KopeckyUhlGSF::CalcStrengthFunction()`, `EquivSquareWell::CalcTransmission()`, `CrossSection::CalculateReactionRates()`, `CrossSection::CrossSection()`, `GammaTransmissionFunc::GammaTransmissionFunc()`, `JLMPotential::JLMPotential()`, and `GammaTransmissionFunc::operator()`.

8.18.2.8 `const double uconv =931.4940880`

Referenced by `LevelDensity::CalcConstantTempTerms()`, `Potential::CalcTransmission()`, `EquivSquareWell::CalcTransmission()`, `NuclearMass::CalculateLDMMass()`, `CrossSection::CalculateReactionRates()`, `CrossSection::CrossSection()`, `NuclearMass::InitializeMasses()`, `LevelDensity::operator()`, `CoulFunc::operator()`, `ParticleTransmissionFunc::ParticleTransmissionFunc()`, `CoulFunc::Penetrability()`, `CoulFunc::PEShift()`, `Potential::Solve()`, and `LevelDensity::TotalLevelDensity()`.

## 8.19 `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/CoulFunc.h` File Reference

### Classes

- struct [CoulWaves](#)
- class [CoulFunc](#)

## 8.20 `/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/CrossSection.h` File Reference

```
#include <vector>
#include <map>
#include <string>
```

### Classes

- struct [int\\_double\\_pair\\_compare](#)
- class [CrossSectionValues](#)
- class [CrossSection](#)

### Typedefs

- typedef `std::vector< std::pair< Decayer *, std::vector< SpinRatePair * > > >` [DecayerVector](#)
- typedef `std::pair< int, double >` [int\\_double\\_pair](#)

### 8.20.1 Typedef Documentation

8.20.1.1 `typedef std::vector<std::pair<Decayer*,std::vector<SpinRatePair*>>> > > DecayerVector`

8.20.1.2 `typedef std::pair<int,double> int_double_pair`

## 8.21 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayController.h File Reference

```
#include <vector>
#include "Decayer.h"
#include "DecayProduct.h"
```

### Classes

- class [DecayController](#)

## 8.22 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Decayer.h File Reference

```
#include <vector>
#include <cstdlib>
```

### Classes

- class [SpinRatePair](#)
- class [CDFEntry](#)
- class [Decayer](#)

## 8.23 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayProduct.h File Reference

### Classes

- class [DecayData](#)
- class [DecayProduct](#)

## 8.24 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/DecayResults.h File Reference

```
#include <TTree.h>
#include <TFile.h>
#include <vector>
```

## Classes

- class [DecayResults](#)

### 8.25 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/elements.h](#) File Reference

### 8.26 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/EquivSquareWell.h](#) File Reference

```
#include "CoulFunc.h"
#include "ParticleTransmissionFunc.h"
```

## Classes

- class [EquivSquareWell](#)

### 8.27 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/GammaTransmission-Func.h](#) File Reference

```
#include "TransmissionFunc.h"
#include "Constants.h"
#include "NuclearMass.h"
#include <math.h>
```

## Classes

- class [GDRParameters](#)
- class [GammaTransmissionFunc](#)

## Typedefs

- typedef  
std::tr1::unordered\_map  
< [MassKey](#), [GDRParameters](#) > [GDRTTable](#)

### 8.27.1 Typedef Documentation

8.27.1.1 typedef std::tr1::unordered\_map<[MassKey](#), [GDRParameters](#) > [GDRTTable](#)

### 8.28 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/JLMPotential.h](#) File Reference

```
#include "Potential.h"
```

## Classes

- class [JLMPotential](#)

## 8.29 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/KopeckyUhlGSF.h File Reference

```
#include "GammaTransmissionFunc.h"
```

## Classes

- class [KopeckyUhlGSF](#)

## 8.30 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/LevelDensity.h File Reference

## Classes

- class [LevelDensity](#)

## 8.31 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/McCullaghGSF.h File Reference

```
#include "GammaTransmissionFunc.h"
```

## Classes

- class [McCullaghGSF](#)

## 8.32 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/McFaddenSatchler-Potential.h File Reference

```
#include "Potential.h"
```

## Classes

- class [McFaddenSatchlerPotential](#)

## 8.33 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearLevels.h File Reference

```
#include "NuclearMass.h"
```

```
#include <vector>
#include <string>
```

## Classes

- class [GammaTransition](#)
- class [Level](#)
- class [LevelsContainer](#)
- class [NuclearLevels](#)

## Typedefs

- typedef  
std::tr1::unordered\_map  
< [MassKey](#), [LevelsContainer](#) > [LevelsTable](#)

### 8.33.1 Typedef Documentation

8.33.1.1 typedef std::tr1::unordered\_map<MassKey, LevelsContainer> LevelsTable

## 8.34 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/NuclearMass.h File Reference

```
#include <tr1/unordered_map>
#include <string>
```

## Classes

- class [MassKey](#)
- struct [std::tr1::hash< MassKey >](#)
- struct [std::equal\\_to< MassKey >](#)
- class [MassEntry](#)
- class [NuclearMass](#)

## Namespaces

- [std](#)
- [std::tr1](#)

## Typedefs

- typedef  
std::tr1::unordered\_map  
< [MassKey](#), [MassEntry](#) > [MassTable](#)
- typedef  
std::tr1::unordered\_map  
< std::string, int > [ElementTable](#)

### 8.34.1 Typedef Documentation

8.34.1.1 `typedef std::tr1::unordered_map<std::string, int > ElementTable`

8.34.1.2 `typedef std::tr1::unordered_map<MassKey, MassEntry> MassTable`

## 8.35 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ParticleHoleLevelDensity.h File Reference

### Classes

- class [ParticleHoleLevelDensity](#)

## 8.36 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/ParticleTransmissionFunc.h File Reference

```
#include "TransmissionFunc.h"
#include "NuclearMass.h"
#include "Constants.h"
#include <map>
```

### Classes

- class [SLPair](#)
- class [ParticleTransmissionFunc](#)

## 8.37 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/Potential.h File Reference

```
#include <vector>
#include <complex>
#include "ParticleTransmissionFunc.h"
```

### Classes

- class [Potential](#)

## 8.38 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqDecayer.h File Reference

```
#include <vector>
```

### Classes

- class [PreEqSpinRatePair](#)

- class [PreEqCDFEntry](#)
- class [PreEqDecayer](#)

### 8.39 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/PreEqTransitionRateFunc.h File Reference

```
#include "TransitionRateFunc.h"
#include "ParticleHoleLevelDensity.h"
```

#### Classes

- class [PreEqTransitionRateFunc](#)

### 8.40 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/RauscherLevelDensity.h File Reference

```
#include "LevelDensity.h"
#include "NuclearMass.h"
#include <iostream>
#include <stdlib.h>
#include <math.h>
```

#### Classes

- class [RauscherLevelDensity](#)

### 8.41 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/SapphireMPITypes.h File Reference

```
#include <boost/serialization/access.hpp>
#include <boost/serialization/vector.hpp>
#include <boost/serialization/utility.hpp>
#include <boost/mpi/datatype.hpp>
#include <vector>
#include "DecayProduct.h"
```

#### Classes

- class [InitialNucleusData](#)

#### Namespaces

- [boost](#)
- [boost::serialization](#)



## Enumerations

- enum [SapphireTags\\_t](#) { [SapphireTagProcess](#), [SapphireTagDone](#), [SapphireTagResults](#) }

## Functions

- [BOOST\\_IS\\_MPI\\_DATATYPE](#) ([InitialNucleusData](#))
- void [boost::serialization::serialize](#) (Archive &ar, [DecayData](#) &g, const unsigned int version)
- void [boost::serialization::serialize](#) (Archive &ar, [DecayProduct](#) &g, const unsigned int version)
- [BOOST\\_IS\\_MPI\\_DATATYPE](#) ([DecayData](#))
- [BOOST\\_IS\\_MPI\\_DATATYPE](#) ([DecayProduct](#))

### 8.41.1 Enumeration Type Documentation

#### 8.41.1.1 enum [SapphireTags\\_t](#)

Enumerator

***[SapphireTagProcess](#)***

***[SapphireTagDone](#)***

***[SapphireTagResults](#)***

### 8.41.2 Function Documentation

#### 8.41.2.1 [BOOST\\_IS\\_MPI\\_DATATYPE](#) ( [InitialNucleusData](#) )

#### 8.41.2.2 [BOOST\\_IS\\_MPI\\_DATATYPE](#) ( [DecayData](#) )

#### 8.41.2.3 [BOOST\\_IS\\_MPI\\_DATATYPE](#) ( [DecayProduct](#) )

## 8.42 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransitionRateFunc.h File Reference

```
#include <vector>
#include "LevelDensity.h"
#include "TransmissionFunc.h"
```

## Classes

- class [XYPair](#)
- class [TransitionRateFunc](#)

## 8.43 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/TransmissionFunc.h File Reference

## Classes

- class [TransmissionFunc](#)

#### 8.44 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/README.md](#) File Reference

#### 8.45 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/BrinkAxelGSF.cpp](#) File Reference

```
#include "BrinkAxelGSF.h"
```

#### 8.46 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CoulFunc.cpp](#) File Reference

```
#include "Constants.h"  
#include "CoulFunc.h"  
#include <iostream>  
#include "cwfcomp.H"  
#include <gsl/gsl_sf_coulomb.h>  
#include <gsl/gsl_deriv.h>  
#include <gsl/gsl_errno.h>
```

#### 8.47 [/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/CrossSection.cpp](#) File Reference

```
#include "NuclearLevels.h"  
#include "CrossSection.h"  
#include "Decayer.h"  
#include "TransitionRateFunc.h"  
#include "RauscherLevelDensity.h"  
#include "Constants.h"  
#include <math.h>  
#include <fstream>  
#include <iomanip>  
#include <time.h>  
#include <iostream>  
#include <sstream>  
#include <gsl/gsl_integration.h>  
#include <TGraph.h>  
#include <algorithm>
```

#### Classes

- struct [gsl\\_reactionrate\\_params](#)
- struct [gsl\\_partfunc\\_params](#)

#### Functions

- double [gsl\\_reactionrate\\_integrand](#) (double x, void \*p)
- double [gsl\\_partfunc\\_integrand](#) (double x, void \*p)

### 8.47.1 Function Documentation

#### 8.47.1.1 `double gsl_partfunc_integrand ( double x, void * p )`

References `boltzConst`, `gsl_partfunc_params::density`, and `gsl_partfunc_params::temperature`.

#### 8.47.1.2 `double gsl_reactionrate_integrand ( double x, void * p )`

References `boltzConst`, `gsl_reactionrate_params::graph`, `gsl_reactionrate_params::temperature`, and `gsl_reactionrate_params::useSpline`.

Referenced by `CrossSection::CalculateReactionRates()`.

## 8.48 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayController.cpp File Reference

```
#include "DecayController.h"
#include "Constants.h"
#include "NuclearMass.h"
#include "PreEqDecayer.h"
#include <iostream>
#include <iomanip>
#include <TVector3.h>
#include <stdlib.h>
#include <omp.h>
```

### Variables

- unsigned int `randomSeed` [12]

### 8.48.1 Variable Documentation

#### 8.48.1.1 `unsigned int randomSeed[12]`

Referenced by `Decayer::Decay()`, `PreEqDecayer::Decay()`, `main()`, `ParticleTransmissionFunc::operator()()`, and `GammaTransmissionFunc::operator()()`.

## 8.49 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Decayer.cpp File Reference

```
#include "Decayer.h"
#include "TransitionRateFunc.h"
#include "NuclearMass.h"
#include "NuclearLevels.h"
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <math.h>
#include <omp.h>
```

## Variables

- unsigned int [randomSeed](#) [12]

### 8.49.1 Variable Documentation

#### 8.49.1.1 unsigned int randomSeed[12]

## 8.50 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/DecayResults.cpp File Reference

```
#include "DecayResults.h"
#include "DecayProduct.h"
#include "NuclearMass.h"
#include <fstream>
#include <iostream>
```

## 8.51 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/EquivSquareWell.cpp File Reference

```
#include "EquivSquareWell.h"
```

## 8.52 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/GammaTransmission-Func.cpp File Reference

```
#include "GammaTransmissionFunc.h"
#include "BrinkAxelGSF.h"
#include "KopeckyUhlGSF.h"
#include "McCullaghGSF.h"
#include "Constants.h"
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <stdlib.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
#include <omp.h>
```

## Variables

- unsigned int [randomSeed](#) [12]

### 8.52.1 Variable Documentation

#### 8.52.1.1 unsigned int randomSeed[12]

### 8.53 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/JLMPotential.cpp File Reference

```
#include "JLMPotential.h"  
#include <assert.h>
```

### 8.54 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/KopeckyUhlGSF.cpp File Reference

```
#include "KopeckyUhlGSF.h"  
#include "LevelDensity.h"  
#include <iostream>
```

### 8.55 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/LevelDensity.cpp File Reference

```
#include "LevelDensity.h"  
#include "Constants.h"  
#include <math.h>
```

### 8.56 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McCullaghGSF.cpp File Reference

```
#include "McCullaghGSF.h"
```

### 8.57 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/McFaddenSatchlerPotential.cpp File Reference

```
#include "McFaddenSatchlerPotential.h"
```

### 8.58 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearLevels.cpp File Reference

```
#include "NuclearLevels.h"  
#include <fstream>  
#include <sstream>  
#include <iostream>  
#include <iomanip>  
#include <math.h>  
#include <string.h>
```

## 8.59 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/NuclearMass.cpp File Reference

```
#include "NuclearMass.h"
#include "Constants.h"
#include <sstream>
#include <fstream>
#include <iostream>
#include <stdlib.h>
#include "elements.h"
```

### Macros

- `#define HAS_EXP_MASS 1`
- `#define HAS_TH_MASS 2`
- `#define ELEMENT(Z, EL) {elementTable_[std::string(EL)]=Z;}`

### 8.59.1 Macro Definition Documentation

8.59.1.1 `#define ELEMENT( Z, EL ) {elementTable_[std::string(EL)]=Z;}`

8.59.1.2 `#define HAS_EXP_MASS 1`

Referenced by `NuclearMass::FindMass()`, and `NuclearMass::InitializeMasses()`.

8.59.1.3 `#define HAS_TH_MASS 2`

Referenced by `NuclearMass::FindMass()`, and `NuclearMass::InitializeMasses()`.

## 8.60 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleHoleLevelDensity.cpp File Reference

```
#include "ParticleHoleLevelDensity.h"
#include "gsl/gsl_sf_gamma.h"
#include "math.h"
#include <algorithm>
#include <iostream>
#include "NuclearMass.h"
```

## 8.61 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/ParticleTransmissionFunc.cpp File Reference

```
#include "ParticleTransmissionFunc.h"
#include "EquivSquareWell.h"
#include "McFaddenSatchlerPotential.h"
#include "JLMPotential.h"
#include "Constants.h"
#include <iostream>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
#include <omp.h>
```

### Variables

- unsigned int [randomSeed](#) [12]

#### 8.61.1 Variable Documentation

8.61.1.1 unsigned int randomSeed[12]

## 8.62 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Potential.cpp File Reference

```
#include "Potential.h"
#include "Constants.h"
#include "CoulFunc.h"
#include <iostream>
#include <float.h>
```

## 8.63 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqDecayer.cpp File Reference

```
#include "PreEqDecayer.h"
#include "NuclearMass.h"
#include "PreEqTransitionRateFunc.h"
#include <iostream>
#include <math.h>
#include <fstream>
#include <iomanip>
#include <cstdlib>
#include <omp.h>
```

### Variables

- unsigned int [randomSeed](#) [12]

### 8.63.1 Variable Documentation

#### 8.63.1.1 unsigned int randomSeed[12]

## 8.64 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/PreEqTransitionRateFunc.cpp File Reference

```
#include "PreEqTransitionRateFunc.h"  
#include "ParticleTransmissionFunc.h"  
#include <iostream>
```

## 8.65 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/RauscherLevelDensity.cpp File Reference

```
#include "RauscherLevelDensity.h"  
#include <math.h>
```

## 8.66 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Sapphire.cpp File Reference

```
#include <iostream>  
#include <iomanip>  
#include <fstream>  
#include <sstream>  
#include <time.h>  
#include <stdlib.h>  
#include <string.h>  
#include <math.h>  
#include <algorithm>  
#include "DecayController.h"  
#include "NuclearMass.h"  
#include "DecayResults.h"  
#include "CrossSection.h"  
#include "omp.h"  
#include "TransitionRateFunc.h"  
#include "ParticleTransmissionFunc.h"  
#include "GammaTransmissionFunc.h"
```

### Classes

- struct [EntrancePairs](#)

### Typedefs

- typedef struct [EntrancePairs](#) EntrancePairs



## Functions

- void [Initialize](#) ()
- void [printHelp](#) ()
- void [parseCommandLineForOptions](#) (std::vector< std::string > &args, int &suffixNo, bool &preEq, int &numPiParticles, int &numPiHoles, int &numNuParticles, int &numNuHoles, bool &calcAverageWidth, bool &calcRates, bool &asciiIn, std::string &inFile, int &entranceState, std::vector< int > &exitStates, bool &printTrans)
- bool [parseCommandLineForDecay](#) (std::vector< std::string > &args, int &Z, int &A, double &J, int &Pi, double &lowEnergy, double &highEnergy, int &events)
- bool [parseCommandLineForXS](#) (std::vector< std::string > &args, int &Z, int &A, int &pType, std::string &energyFile, bool &asciiIn)
- int [main](#) (int argc, char \*argv[])

## Variables

- unsigned int [randomSeed](#) [12]

### 8.66.1 Typedef Documentation

#### 8.66.1.1 typedef struct EntrancePairs EntrancePairs

### 8.66.2 Function Documentation

#### 8.66.2.1 void Initialize ( )

... text ...

References [CrossSection::CreateMACSEnergiesVector\(\)](#), [CrossSection::CreateTempVector\(\)](#), [CoulFunc::GSL-ErrorHandler\(\)](#), [NuclearMass::InitializeElements\(\)](#), [GammaTransmissionFunc::InitializeGDRParameters\(\)](#), [NuclearLevels::InitializeLevels\(\)](#), [NuclearMass::InitializeMasses\(\)](#), [ParticleTransmissionFunc::SetAlphaFormalism\(\)](#), [CrossSection::SetCalculateGammaCutoff\(\)](#), [Decayer::SetCrossSection\(\)](#), [PreEqDecayer::SetCrossSection\(\)](#), [GammaTransmissionFunc::SetEGDRType\(\)](#), [TransitionRateFunc::SetGammaCutoffEnergy\(\)](#), [Decayer::SetMaxL\(\)](#), [PreEqDecayer::SetMaxL\(\)](#), [ParticleTransmissionFunc::SetNeutronFormalism\(\)](#), [GammaTransmissionFunc::SetPorterThomas\(\)](#), [ParticleTransmissionFunc::SetPorterThomas\(\)](#), [ParticleTransmissionFunc::SetProtonFormalism\(\)](#), [CrossSection::SetResidualAlpha\(\)](#), [CrossSection::SetResidualGamma\(\)](#), [CrossSection::SetResidualNeutron\(\)](#), [CrossSection::SetResidualProton\(\)](#), and [sourceDirectory\(\)](#).

Referenced by [main\(\)](#).

#### 8.66.2.2 int main ( int argc, char \* argv[] )

References [DecayResults::AddResults\(\)](#), [CrossSection::CalcAverageDWaveResWidth\(\)](#), [CrossSection::CalcAveragePWaveResWidth\(\)](#), [CrossSection::CalcAverageSWaveResWidth\(\)](#), [CrossSection::Calculate\(\)](#), [CrossSection::CalculateReactionRates\(\)](#), [DecayController::Decay\(\)](#), [DecayController::DecayProducts\(\)](#), [Initialize\(\)](#), [CrossSection::IsValid\(\)](#), [parseCommandLineForDecay\(\)](#), [parseCommandLineForOptions\(\)](#), [parseCommandLineForXS\(\)](#), [CrossSection::PrintCrossSections\(\)](#), [DecayController::PrintDecays\(\)](#), [printHelp\(\)](#), [CrossSection::PrintReactionRates\(\)](#), [CrossSection::PrintTransmissionTerms\(\)](#), [randomSeed](#), and [Decayer::SetCrossSection\(\)](#).

#### 8.66.2.3 bool parseCommandLineForDecay ( std::vector< std::string > &args, int &Z, int &A, double &J, int &Pi, double &lowEnergy, double &highEnergy, int &events )

References [NuclearMass::FindZ\(\)](#).

Referenced by [main\(\)](#).

**8.66.2.4** `void parseCommandLineForOptions ( std::vector< std::string > & args, int & suffixNo, bool & preEq, int & numPiParticles, int & numPiHoles, int & numNuParticles, int & numNuHoles, bool & calcAverageWidth, bool & calcRates, bool & asciiln, std::string & inFile, int & entranceState, std::vector< int > & exitStates, bool & printTrans )`

References `TransitionRateFunc::GetGammaCutoffEnergy()`, `Decayer::GetMaxL()`, `ParticleTransmissionFunc::SetAlphaFormalism()`, `CrossSection::SetCalculateGammaCutoff()`, `GammaTransmissionFunc::SetEGDRType()`, `TransitionRateFunc::SetGammaCutoffEnergy()`, `Decayer::SetMaxL()`, `ParticleTransmissionFunc::SetNeutronFormalism()`, `GammaTransmissionFunc::SetPorterThomas()`, `ParticleTransmissionFunc::SetPorterThomas()`, `ParticleTransmissionFunc::SetProtonFormalism()`, `CrossSection::SetResidualAlpha()`, `CrossSection::SetResidualGamma()`, `CrossSection::SetResidualNeutron()`, and `CrossSection::SetResidualProton()`.

Referenced by `main()`.

**8.66.2.5** `bool parseCommandLineForXS ( std::vector< std::string > & args, int & Z, int & A, int & pType, std::string & energyFile, bool asciiln )`

References `NuclearMass::FindZ()`.

Referenced by `main()`.

**8.66.2.6** `void printHelp ( )`

Referenced by `main()`.

## 8.66.3 Variable Documentation

**8.66.3.1** `unsigned int randomSeed[12]`

Referenced by `Decayer::Decay()`, `PreEqDecayer::Decay()`, `main()`, `ParticleTransmissionFunc::operator()()`, and `GammaTransmissionFunc::operator()()`.

## 8.67 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/Setup.cpp File Reference

```
#include "NuclearMass.h"
#include "GammaTransmissionFunc.h"
#include "NuclearLevels.h"
#include "Decayer.h"
#include "Sapphire_config.h"
#include "TransitionRateFunc.h"
#include "CrossSection.h"
#include "PreEqDecayer.h"
#include "ParticleTransmissionFunc.h"
#include "CoulFunc.h"
#include <iostream>
#include <gsl/gsl_errno.h>
```

## Functions

- void [Initialize](#) ()

## 8.67.1 Function Documentation

### 8.67.1.1 void Initialize ( )

... text ...

References CrossSection::CreateMACSEnergiesVector(), CrossSection::CreateTempVector(), CoulFunc::GSL-ErrorHandler(), NuclearMass::InitializeElements(), GammaTransmissionFunc::InitializeGDRParameters(), NuclearLevels::InitializeLevels(), NuclearMass::InitializeMasses(), ParticleTransmissionFunc::SetAlphaFormalism(), CrossSection::SetCalculateGammaCutoff(), Decayer::SetCrossSection(), PreEqDecayer::SetCrossSection(), GammaTransmissionFunc::SetEGDRType(), TransitionRateFunc::SetGammaCutoffEnergy(), Decayer::SetMaxL(), PreEqDecayer::SetMaxL(), ParticleTransmissionFunc::SetNeutronFormalism(), GammaTransmissionFunc::SetPorter-Thomas(), ParticleTransmissionFunc::SetPorterThomas(), ParticleTransmissionFunc::SetProtonFormalism(), CrossSection::SetResidualAlpha(), CrossSection::SetResidualGamma(), CrossSection::SetResidualNeutron(), CrossSection::SetResidualProton(), and sourceDirectory().

Referenced by main().

## 8.68 /afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/TransitionRateFunc.cpp File Reference

```
#include "TransitionRateFunc.h"
#include "ParticleTransmissionFunc.h"
#include "GammaTransmissionFunc.h"
#include "RauscherLevelDensity.h"
#include "NuclearLevels.h"
#include <iostream>
#include <stdlib.h>
```

# Index

~Coulomb\_wave\_functions  
    Coulomb\_wave\_functions, 17

~DecayResults  
    DecayResults, 29

~Decayer  
    Decayer, 25

~EquivSquareWell  
    EquivSquareWell, 31

~GammaTransmissionFunc  
    GammaTransmissionFunc, 33

~LevelDensity  
    LevelDensity, 41

~ParticleTransmissionFunc  
    ParticleTransmissionFunc, 52

~Potential  
    Potential, 55

~PreEqDecayer  
    PreEqDecayer, 58

~PreEqTransitionRateFunc  
    PreEqTransitionRateFunc, 60

~RauscherLevelDensity  
    RauscherLevelDensity, 61

~TransitionRateFunc  
    TransitionRateFunc, 63

~TransmissionFunc  
    TransmissionFunc, 65

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/-  
    README.md, 92

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/coulomb\_wave\_functions.H, 76

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/coulomb\_wave\_functions.H, 79

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/include/coulomb\_wave\_functions\_int.H, 80

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/complex\_functions.cpp, 80

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/cwfcomplex\_functions.cpp, 81

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/coul/src/ode\_functions\_int.cpp, 82

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/generated/-  
    Sapphire\_config.h, 75

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    BrinkAxelGSF.h, 82

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    Constants.h, 82

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    CoulFunc.h, 84

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    CrossSection.h, 84

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    DecayController.h, 85

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    DecayProduct.h, 85

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    DecayResults.h, 85

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    Decayer.h, 85

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    EquivSquareWell.h, 86

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    GammaTransmissionFunc.h, 86

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    JLMPotential.h, 86

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    KopeckyUhlGSF.h, 87

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    LevelDensity.h, 87

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    McCullaghGSF.h, 87

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    McFaddenSatchlerPotential.h, 87

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    NuclearLevels.h, 87

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    NuclearMass.h, 88

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    ParticleHoleLevelDensity.h, 89

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    ParticleTransmissionFunc.h, 89

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    Potential.h, 89

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    PreEqDecayer.h, 89

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    PreEqTransitionRateFunc.h, 90

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    RauscherLevelDensity.h, 90

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    SapphireMPITypes.h, 90

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    TransitionRateFunc.h, 91

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/-  
    TransmissionFunc.h, 91

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/include/elements.-  
    h, 86

/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-  
    BrinkAxelGSF.cpp, 92

[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- AddResults](#)  
     CoulFunc.cpp, [92](#)      DecayResults, [29](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- alpha\\_](#)  
     CrossSection.cpp, [92](#)      CrossSectionValues, [22](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- AlphaEntranceWidth](#)  
     DecayController.cpp, [93](#)      Decayer, [25](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- alphaEntranceWidth](#)  
     DecayResults.cpp, [94](#)      DecayData, [23](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- alphaStellar\\_](#)  
     Decayer.cpp, [93](#)      CrossSectionValues, [22](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- AlphaTotalWidth](#)  
     EquivSquareWell.cpp, [94](#)      Decayer, [25](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- alphaTotalWidth](#)  
     GammaTransmissionFunc.cpp, [94](#)      DecayData, [23](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/- avagadroNum](#)  
     JLMPotential.cpp, [95](#)      Constants.h, [83](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#)  
     KopeckyUHLGSF.cpp, [95](#)      BOOST\_IS\_MPI\_DATATYPE  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#)  
     LevelDensity.cpp, [95](#)      SapphireMPITypes.h, [91](#)  
     backshift\_  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#)  
     LevelDensity, [42](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#)  
     McCullaghGSF.cpp, [95](#)      boltzConst  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#)  
     Constants.h, [83](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#)  
     McFaddenSatchlerPotential.cpp, [95](#)      boost, [11](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) boost::serialization, [11](#)  
     NuclearLevels.cpp, [95](#)      serialize, [11](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) boost::serialization::access  
     NuclearMass.cpp, [96](#)      InitialNucleusData, [37](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) boundaryRadius\_  
     ParticleHoleLevelDensity.cpp, [96](#)      Potential, [56](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) BrinkAxelGSF, [13](#)  
     ParticleTransmissionFunc.cpp, [97](#)      BrinkAxelGSF, [13](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) BrinkAxelGSF, [13](#)  
     Potential.cpp, [97](#)      CalcStrengthFunction, [13](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) build/generated/Sapphire\_config.h  
     PreEqDecayer.cpp, [97](#)      sourceDirectory, [75](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#)  
     PreEqTransitionRateFunc.cpp, [98](#)      C\_DIALECT  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) CMakeCCompilerId.c, [69](#)  
     RauscherLevelDensity.cpp, [98](#)      CDFEntry, [14](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) CDFEntry, [14](#)  
     Sapphire.cpp, [98](#)      CDFEntry, [14](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) energy\_, [14](#)  
     Setup.cpp, [100](#)      pairIndex\_, [14](#)  
[/afs/crc.nd.edu/user/p/pscholz/Private/sapphire-devel/src/-](#) value\_, [14](#)  
     TransitionRateFunc.cpp, [101](#)      CMakeCCompilerId.c  
     ARCHITECTURE\_ID, [69](#)  
A      C\_DIALECT, [69](#)  
     InitialNucleusData, [37](#)      COMPILER\_ID, [69](#)  
A\_      DEC, [69](#)  
     DecayProduct, [27](#)      HEX, [70](#)  
     EntrancePairs, [30](#)      info\_arch, [70](#)  
     LevelDensity, [42](#)      info\_compiler, [70](#)  
     MassKey, [45](#)      info\_language\_dialect\_default, [70](#)  
     PreEqSpinRatePair, [59](#)      info\_platform, [70](#)  
     SpinRatePair, [62](#)      main, [70](#)  
ARCHITECTURE\_ID      PLATFORM\_ID, [70](#)  
     CMakeCCompilerId.c, [69](#)      STRINGIFY, [70](#)  
     CMakeCXXCompilerId.cpp, [71](#)      STRINGIFY\_HELPER, [70](#)

- CMakeCXXCompilerId.cpp
  - ARCHITECTURE\_ID, 71
  - COMPILER\_ID, 71
  - CXX\_STD, 71
  - DEC, 71
  - HEX, 71
  - info\_arch, 72
  - info\_compiler, 72
  - info\_language\_dialect\_default, 72
  - info\_platform, 72
  - main, 72
  - PLATFORM\_ID, 71
  - STRINGIFY, 72
  - STRINGIFY\_HELPER, 72
- CMakeFiles/3.13.2/CompilerIdC/CMakeCCompilerId.c, 69
- CMakeFiles/3.13.2/CompilerIdCXX/CMakeCXXCompilerId.cpp, 71
- CMakeFiles/FindOpenMP/OpenMPCheckVersion.c, 73
- CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp, 74
- CMakeFiles/FindOpenMP/OpenMPTryFlag.c, 74
- CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp, 75
- CMakeFiles/feature\_tests.c, 72
- CMakeFiles/feature\_tests.cxx, 73
- COMPILER\_ID
  - CMakeCCompilerId.c, 69
  - CMakeCXXCompilerId.cpp, 71
- CXX\_STD
  - CMakeCXXCompilerId.cpp, 71
- CalcAverageDWaveResWidth
  - CrossSection, 20
- CalcAveragePWaveResWidth
  - CrossSection, 20
- CalcAverageSWaveResWidth
  - CrossSection, 20
- CalcBackShift
  - LevelDensity, 42
  - RauscherLevelDensity, 61
- CalcBeta
  - Potential, 55
- CalcConstantTempTerms
  - LevelDensity, 42
- CalcDensityParam
  - LevelDensity, 42
  - RauscherLevelDensity, 61
- CalcLevelDensity
  - TransitionRateFunc, 64
- CalcNuclearTemp
  - LevelDensity, 42
  - RauscherLevelDensity, 61
- CalcSLDependentFunctions
  - ParticleTransmissionFunc, 52
- CalcStrengthFunction
  - BrinkAxelGSF, 13
  - GammaTransmissionFunc, 33
  - KopeckyUhlGSF, 40
  - McCullaghGSF, 46
- CalcTotalLevelDensity
  - TransitionRateFunc, 64
- CalcTransmission
  - EquivSquareWell, 31
  - ParticleTransmissionFunc, 52
  - Potential, 55
- CalcTransmissionFunc
  - TransitionRateFunc, 64
- Calculate
  - CrossSection, 20
  - JLMPotential, 39
  - McFaddenSatchlerPotential, 47
  - Potential, 55
- CalculateDensity
  - JLMPotential, 39
- CalculateLDMMass
  - NuclearMass, 48
- CalculateReactionRates
  - CrossSection, 20
- complex
  - Constants.h, 83
- complex\_functions.cpp
  - exp\_l\_omega\_chi\_calc, 80
  - expm1, 80
  - log1p, 80
  - log\_Cl\_eta\_calc, 81
  - log\_Gamma, 81
  - log\_cut\_constant\_AS\_calc, 81
  - log\_cut\_constant\_CFa\_calc, 81
  - log\_cut\_constant\_CFb\_calc, 81
  - sigma\_l\_calc, 81
  - sin\_chi\_calc, 81
- complex\_functions.H
  - exp\_l\_omega\_chi\_calc, 77
  - expm1, 77
  - inf\_norm, 77
  - isfinite, 77
  - log1p, 77
  - log\_Cl\_eta\_calc, 77
  - log\_Gamma, 78
  - log\_cut\_constant\_AS\_calc, 77
  - log\_cut\_constant\_CFa\_calc, 78
  - log\_cut\_constant\_CFb\_calc, 78
  - operator\*, 78
  - operator+, 78
  - operator-, 78
  - operator/, 78, 79
  - operator==, 79
  - precision, 79
  - SIGN, 77
  - sigma\_l\_calc, 79
  - sin\_chi\_calc, 79
  - sqrt\_precision, 79
- constAngTerm\_
  - LevelDensity, 43
- Constants.h
  - avadroNum, 83
  - boltzConst, 83

- complex, [83](#)
- eMass, [83](#)
- fstruc, [83](#)
- hbarc, [83](#)
- lightSpeedInCmPerS, [83](#)
- matrix\_c, [83](#)
- matrix\_r, [83](#)
- pi, [84](#)
- uconv, [84](#)
- vector\_c, [83](#)
- vector\_matrix\_c, [83](#)
- vector\_matrix\_r, [83](#)
- vector\_r, [83](#)
- CorrectWidthFluctuations
  - Decayer, [25](#)
- CoulFunc, [14](#)
  - CoulFunc, [15](#)
  - coulLast, [15](#)
  - CoulFunc, [15](#)
  - energyLast, [15](#)
  - GSLErrorHandler, [15](#)
  - lLast, [15](#)
  - operator(), [15](#)
  - PEShift, [16](#)
  - PEShift\_dE, [16](#)
  - Penetrability, [16](#)
  - radiusLast, [16](#)
  - redmass, [16](#)
  - setLast, [16](#)
  - z1, [16](#)
  - z2, [16](#)
- coulFunc\_
  - Potential, [56](#)
- coulLast
  - CoulFunc, [15](#)
- CoulWaves, [18](#)
  - dF, [18](#)
  - dG, [18](#)
  - F, [18](#)
  - G, [19](#)
- Coulomb\_wave\_functions, [17](#)
  - ~Coulomb\_wave\_functions, [17](#)
  - Coulomb\_wave\_functions, [17](#)
  - Coulomb\_wave\_functions, [17](#)
  - eta, [18](#)
  - F\_dF, [17](#)
  - F\_dF\_init, [17](#)
  - G\_dG, [17](#)
  - H\_dH, [17](#)
  - H\_dH\_scaled, [18](#)
  - is\_it\_normalized, [18](#)
  - I, [18](#)
- coulombRadius\_
  - Potential, [57](#)
- CreateGammaTransmissionFunc
  - GammaTransmissionFunc, [33](#)
- CreateMACSEnergiesVector
  - CrossSection, [20](#)
- CreateParticleTransmissionFunc
  - ParticleTransmissionFunc, [53](#)
- CreateTempVector
  - CrossSection, [20](#)
- criticalU\_
  - LevelDensity, [43](#)
- CrossSection, [19](#)
  - CalcAverageDWaveResWidth, [20](#)
  - CalcAveragePWaveResWidth, [20](#)
  - CalcAverageSWaveResWidth, [20](#)
  - Calculate, [20](#)
  - CalculateReactionRates, [20](#)
  - CreateMACSEnergiesVector, [20](#)
  - CreateTempVector, [20](#)
  - CrossSection, [19](#)
  - CrossSection, [19](#)
  - Decayer, [26](#)
  - IsValid, [20](#)
  - PrintCrossSections, [20](#)
  - PrintReactionRates, [20](#)
  - PrintTransmissionTerms, [20](#)
  - SetCalculateGammaCutoff, [21](#)
  - SetResidualAlpha, [21](#)
  - SetResidualGamma, [21](#)
  - SetResidualNeutron, [21](#)
  - SetResidualProton, [21](#)
- CrossSection.cpp
  - gsl\_partfunc\_integrand, [93](#)
  - gsl\_reactionrate\_integrand, [93](#)
- CrossSection.h
  - DecayerVector, [85](#)
  - int\_double\_pair, [85](#)
- CrossSectionValues, [21](#)
  - alpha\_, [22](#)
  - alphaStellar\_, [22](#)
  - CrossSectionValues, [22](#)
  - CrossSectionValues, [22](#)
  - gamma\_, [22](#)
  - gammaStellar\_, [22](#)
  - neutron\_, [22](#)
  - neutronStellar\_, [22](#)
  - proton\_, [22](#)
  - protonStellar\_, [22](#)
- CumulativeSum
  - PreEqTransitionRateFunc, [60](#)
  - TransitionRateFunc, [64](#)
- DEC
  - CMakeCCompilerId.c, [69](#)
  - CMakeCXXCompilerId.cpp, [71](#)
- dF
  - CoulWaves, [18](#)
- dG
  - CoulWaves, [18](#)
- Decay
  - DecayController, [22](#)
  - Decayer, [25](#)
  - PreEqDecayer, [58](#)
- DecayController, [22](#)

- Decay, [22](#)
- DecayController, [22](#)
- DecayProducts, [23](#)
- DecayController, [22](#)
- PrintDecays, [23](#)
- DecayController.cpp
  - randomSeed, [93](#)
- DecayData, [23](#)
  - alphaEntranceWidth, [23](#)
  - alphaTotalWidth, [23](#)
  - DecayData, [23](#)
  - DecayData, [23](#)
  - energy, [23](#)
  - gammaEntranceWidth, [23](#)
  - gammaTotalWidth, [23](#)
  - neutronEntranceWidth, [24](#)
  - neutronTotalWidth, [24](#)
  - protonEntranceWidth, [24](#)
  - protonTotalWidth, [24](#)
- DecayProduct, [26](#)
  - A\_, [27](#)
  - DecayProduct, [27](#)
  - DecayProduct, [27](#)
  - excitationEnergy\_, [27](#)
  - fragmentEnergy\_, [27](#)
  - fragmentEnergyCM\_, [27](#)
  - fragmentMomentumX\_, [27](#)
  - fragmentMomentumY\_, [27](#)
  - fragmentMomentumZ\_, [27](#)
  - J\_, [28](#)
  - particleEnergy\_, [28](#)
  - particleEnergyCM\_, [28](#)
  - particleMomentumX\_, [28](#)
  - particleMomentumY\_, [28](#)
  - particleMomentumZ\_, [28](#)
  - particlePhiCM\_, [28](#)
  - particleThetaCM\_, [28](#)
  - particleType\_, [28](#)
  - Pi\_, [28](#)
  - Z\_, [28](#)
- DecayProducts
  - DecayController, [23](#)
- DecayResults, [29](#)
  - ~DecayResults, [29](#)
  - AddResults, [29](#)
  - DecayResults, [29](#)
  - DecayResults, [29](#)
- Decayer, [24](#)
  - ~Decayer, [25](#)
  - AlphaEntranceWidth, [25](#)
  - AlphaTotalWidth, [25](#)
  - CorrectWidthFluctuations, [25](#)
  - CrossSection, [26](#)
  - Decay, [25](#)
  - Decayer, [25](#)
  - GammaEntranceWidth, [25](#)
  - GammaTotalWidth, [25](#)
  - GetMaxL, [25](#)
  - NeutronEntranceWidth, [25](#)
  - NeutronTotalWidth, [25](#)
  - PrintCDF, [26](#)
  - PrintFunctions, [26](#)
  - ProtonEntranceWidth, [26](#)
  - ProtonTotalWidth, [26](#)
  - SetCrossSection, [26](#)
  - SetMaxL, [26](#)
- Decayer.cpp
  - randomSeed, [94](#)
- DecayerVector
  - CrossSection.h, [85](#)
- density
  - gsl\_partfunc\_params, [35](#)
- e0\_
  - LevelDensity, [43](#)
- E\_
  - GDRParameters, [35](#)
- ELEMENT
  - NuclearMass.cpp, [96](#)
- eMass
  - Constants.h, [83](#)
- egdrType\_
  - GammaTransmissionFunc, [34](#)
- egqrType\_
  - GammaTransmissionFunc, [34](#)
- ElementTable
  - NuclearMass.h, [89](#)
- energy
  - DecayData, [23](#)
- energy\_
  - CDFEntry, [14](#)
  - GammaTransition, [31](#)
  - Level, [40](#)
  - PreEqCDFEntry, [57](#)
- energyLast
  - CoulFunc, [15](#)
- EntrancePairs, [29](#)
  - A\_, [30](#)
  - EntrancePairs, [29](#)
  - EntrancePairs, [29](#)
  - pType\_, [30](#)
  - Sapphire.cpp, [99](#)
  - Z\_, [30](#)
- EquivSquareWell, [30](#)
  - ~EquivSquareWell, [31](#)
  - CalcTransmission, [31](#)
  - EquivSquareWell, [31](#)
  - EquivSquareWell, [31](#)
- eta
  - Coulomb\_wave\_functions, [18](#)
- eta\_
  - GDRParameters, [35](#)
- excitationEnergy\_
  - DecayProduct, [27](#)
- ExclusiveBranching
  - TransitionRateFunc, [64](#)
- exp\_l\_omega\_chi\_calc



- complex\_functions.cpp, [80](#)
  - complex\_functions.H, [77](#)
- expMass\_
  - MassEntry, [44](#)
- expm1
  - complex\_functions.cpp, [80](#)
  - complex\_functions.H, [77](#)
- F
  - CoulWaves, [18](#)
- F\_dF
  - Coulomb\_wave\_functions, [17](#)
- F\_dF\_init
  - Coulomb\_wave\_functions, [17](#)
- feature\_tests.c
  - features, [73](#)
  - main, [72](#)
- feature\_tests.cxx
  - features, [73](#)
  - main, [73](#)
- features
  - feature\_tests.c, [73](#)
  - feature\_tests.cxx, [73](#)
- FindElement
  - NuclearMass, [48](#)
- FindLevels
  - NuclearLevels, [48](#)
- FindMass
  - NuclearMass, [48](#)
- FindZ
  - NuclearMass, [49](#)
- FiniteDepth
  - ParticleHoleLevelDensity, [51](#)
- fragmentEnergy\_
  - DecayProduct, [27](#)
- fragmentEnergyCM\_
  - DecayProduct, [27](#)
- fragmentMomentumX\_
  - DecayProduct, [27](#)
- fragmentMomentumY\_
  - DecayProduct, [27](#)
- fragmentMomentumZ\_
  - DecayProduct, [27](#)
- fstruc
  - Constants.h, [83](#)
- Function
  - TransitionRateFunc, [64](#)
- G
  - CoulWaves, [19](#)
- G\_dG
  - Coulomb\_wave\_functions, [17](#)
- GDRParameters, [34](#)
  - E\_, [35](#)
  - eta\_, [35](#)
  - GDRParameters, [35](#)
  - GDRParameters, [35](#)
  - kSigmaGamma\_, [35](#)
  - W\_, [35](#)
- GDRTable
  - GammaTransmissionFunc.h, [86](#)
- gNu
  - ParticleHoleLevelDensity, [51](#)
- gPi
  - ParticleHoleLevelDensity, [51](#)
- GSLErrorHandler
  - CoulFunc, [15](#)
- gamma\_
  - CrossSectionValues, [22](#)
- GammaEntranceWidth
  - Decayer, [25](#)
- gammaEntranceWidth
  - DecayData, [23](#)
- gammaStellar\_
  - CrossSectionValues, [22](#)
- GammaTotalWidth
  - Decayer, [25](#)
- gammaTotalWidth
  - DecayData, [23](#)
- GammaTransition, [31](#)
  - energy\_, [31](#)
  - GammaTransition, [31](#)
  - GammaTransition, [31](#)
  - levelIndex\_, [32](#)
  - probability\_, [32](#)
- GammaTransmissionFunc, [32](#)
  - ~GammaTransmissionFunc, [33](#)
  - CalcStrengthFunction, [33](#)
  - CreateGammaTransmissionFunc, [33](#)
  - egdrType\_, [34](#)
  - egqrType\_, [34](#)
  - GammaTransmissionFunc, [33](#)
  - GammaTransmissionFunc, [33](#)
  - gdrParameters\_, [34](#)
  - gdrTable\_, [34](#)
  - InitializeGDRParameters, [33](#)
  - IsValid, [33](#)
  - mgdrType\_, [34](#)
  - operator(), [33](#)
  - porterThomas\_, [34](#)
  - SetEGDRType, [33](#)
  - SetPorterThomas, [33](#)
- GammaTransmissionFunc.cpp
  - randomSeed, [94](#)
- GammaTransmissionFunc.h
  - GDRTable, [86](#)
- gammas\_
  - Level, [40](#)
- gdrParameters\_
  - GammaTransmissionFunc, [34](#)
- gdrTable\_
  - GammaTransmissionFunc, [34](#)
- generated/Sapphire\_config.h, [75](#)
  - sourceDirectory, [76](#)
- GetA
  - JLMPotential, [39](#)
- GetBoundaryRadius

- Potential, 56
- GetGammaCutoffEnergy
  - TransitionRateFunc, 64
- GetMaxL
  - Decayer, 25
  - PreEqDecayer, 58
- GetRMax
  - Potential, 56
- GetRedMass
  - Potential, 56
- GetZ1Z2
  - Potential, 56
- graph
  - gsl\_reactionrate\_params, 36
- GroundStateTransmission
  - TransitionRateFunc, 64
- gsl\_partfunc\_integrand
  - CrossSection.cpp, 93
- gsl\_partfunc\_params, 35
  - density, 35
  - temperature, 35
- gsl\_reactionrate\_integrand
  - CrossSection.cpp, 93
- gsl\_reactionrate\_params, 36
  - graph, 36
  - temperature, 36
  - useSpline, 36
- H\_dH
  - Coulomb\_wave\_functions, 17
- H\_dH\_scaled
  - Coulomb\_wave\_functions, 18
- HAS\_EXP\_MASS
  - NuclearMass.cpp, 96
- HAS\_TH\_MASS
  - NuclearMass.cpp, 96
- HEX
  - CMakeCCompilerId.c, 70
  - CMakeCXXCompilerId.cpp, 71
- hbarc
  - Constants.h, 83
- highEnergy
  - InitialNucleusData, 37
- HighestBoundEnergy
  - NuclearMass, 49
- inf\_norm
  - complex\_functions.H, 77
- info\_arch
  - CMakeCCompilerId.c, 70
  - CMakeCXXCompilerId.cpp, 72
- info\_compiler
  - CMakeCCompilerId.c, 70
  - CMakeCXXCompilerId.cpp, 72
- info\_language\_dialect\_default
  - CMakeCCompilerId.c, 70
  - CMakeCXXCompilerId.cpp, 72
- info\_platform
  - CMakeCCompilerId.c, 70
- CMakeCXXCompilerId.cpp, 72
- InitialNucleusData, 36
  - A, 37
  - boost::serialization::access, 37
  - highEnergy, 37
  - InitialNucleusData, 37
  - InitialNucleusData, 37
  - J, 37
  - lowEnergy, 37
  - Pi, 37
  - preEq, 37
  - Z, 37
- Initialize
  - Sapphire.cpp, 99
  - Setup.cpp, 101
- InitializeElements
  - NuclearMass, 49
- InitializeGDRParameters
  - GammaTransmissionFunc, 33
- InitializeLevels
  - NuclearLevels, 48
- InitializeMasses
  - NuclearMass, 49
- int\_double\_pair
  - CrossSection.h, 85
- int\_double\_pair\_compare, 37
  - operator(), 38
- Integral
  - PreEqTransitionRateFunc, 60
  - TransitionRateFunc, 64
- integral\_
  - PreEqSpinRatePair, 59
  - SpinRatePair, 62
- is\_it\_normalized
  - Coulomb\_wave\_functions, 18
- IsValid
  - CrossSection, 20
  - GammaTransmissionFunc, 33
  - ParticleTransmissionFunc, 53
  - TransmissionFunc, 65
- isfinite
  - complex\_functions.H, 77
- J
  - InitialNucleusData, 37
- J\_
  - DecayProduct, 28
  - Level, 40
  - LevelDensity, 43
- jFinal\_
  - TransmissionFunc, 66
- jInitial\_
  - TransmissionFunc, 66
- JLMPotential, 38
  - Calculate, 39
  - CalculateDensity, 39
  - GetA, 39
  - JLMPotential, 38
  - JLMPotential, 38

- kSigmaGamma\_
  - GDRParameters, 35
- KopeckyUhlGSF, 39
  - CalcStrengthFunction, 40
  - KopeckyUhlGSF, 39
  - KopeckyUhlGSF, 39
  - LevelDensity, 42
- l
  - Coulomb\_wave\_functions, 18
- l\_
  - SLPair, 62
- lLast
  - CoulFunc, 15
- Level, 40
  - energy\_, 40
  - gammas\_, 40
  - J\_, 40
  - Level, 40
  - Pi\_, 40
- LevelDensity, 41
  - ~LevelDensity, 41
  - A\_, 42
  - backshift\_, 42
  - CalcBackShift, 42
  - CalcConstantTempTerms, 42
  - CalcDensityParam, 42
  - CalcNuclearTemp, 42
  - constAngTerm\_, 43
  - criticalU\_, 43
  - e0\_, 43
  - J\_, 43
  - KopeckyUhlGSF, 42
  - LevelDensity, 41
  - LevelDensity, 41
  - nuclearTemp\_, 43
  - operator(), 42
  - r0\_, 43
  - TotalLevelDensity, 42
  - Z\_, 43
  - zeta\_, 43
- levelIndex\_
  - GammaTransition, 32
- levels\_
  - LevelsContainer, 44
- LevelsContainer, 43
  - levels\_, 44
  - LevelsContainer, 44
  - LevelsContainer, 44
- LevelsTable
  - NuclearLevels.h, 88
- lightSpeedInCmPerS
  - Constants.h, 83
- log1p
  - complex\_functions.cpp, 80
  - complex\_functions.H, 77
- log\_Cl\_eta\_calc
  - complex\_functions.cpp, 81
  - complex\_functions.H, 77
- log\_Gamma
  - complex\_functions.cpp, 81
  - complex\_functions.H, 78
- log\_cut\_constant\_AS\_calc
  - complex\_functions.cpp, 81
  - complex\_functions.H, 77
- log\_cut\_constant\_CFa\_calc
  - complex\_functions.cpp, 81
  - complex\_functions.H, 78
- log\_cut\_constant\_CFb\_calc
  - complex\_functions.cpp, 81
  - complex\_functions.H, 78
- lowEnergy
  - InitialNucleusData, 37
- m1\_
  - ParticleTransmissionFunc, 54
- m2\_
  - TransmissionFunc, 66
- main
  - CMakeCCompilerId.c, 70
  - CMakeCXXCompilerId.cpp, 72
  - feature\_tests.c, 72
  - feature\_tests.cxx, 73
  - OpenMPCheckVersion.c, 73
  - OpenMPCheckVersion.cpp, 74
  - OpenMPTryFlag.c, 75
  - OpenMPTryFlag.cpp, 75
  - Sapphire.cpp, 99
- mask\_
  - MassEntry, 45
- MassDifference
  - NuclearMass, 49
- MassEntry, 44
  - expMass\_, 44
  - mask\_, 45
  - MassEntry, 44
  - MassEntry, 44
  - microEnergyCorr\_, 45
  - thMass\_, 45
- MassKey, 45
  - A\_, 45
  - MassKey, 45
  - MassKey, 45
  - operator<, 45
  - Z\_, 45
- MassTable
  - NuclearMass.h, 89
- matrix\_c
  - Constants.h, 83
- matrix\_r
  - Constants.h, 83
- maxL\_
  - TransmissionFunc, 66
- McCullaghGSF, 46
  - CalcStrengthFunction, 46
  - McCullaghGSF, 46
  - McCullaghGSF, 46
- McFaddenSatchlerPotential, 46

- Calculate, [47](#)
- McFaddenSatchlerPotential, [47](#)
- McFaddenSatchlerPotential, [47](#)
- mgdrType\_
  - GammaTransmissionFunc, [34](#)
- MicroEnergyCorr
  - NuclearMass, [49](#)
- microEnergyCorr\_
  - MassEntry, [45](#)
- neutron\_
  - CrossSectionValues, [22](#)
- NeutronEntranceWidth
  - Decayer, [25](#)
- neutronEntranceWidth
  - DecayData, [24](#)
- neutronHoleNumber\_
  - PreEqSpinRatePair, [59](#)
- neutronNumber\_
  - PreEqSpinRatePair, [59](#)
- NeutronPairingGap
  - NuclearMass, [49](#)
- neutronStellar\_
  - CrossSectionValues, [22](#)
- NeutronTotalWidth
  - Decayer, [25](#)
- neutronTotalWidth
  - DecayData, [24](#)
- NormalizeInternally
  - Potential, [56](#)
- NormalizeOverAllSpace
  - Potential, [56](#)
- NuclearLevels, [47](#)
  - FindLevels, [48](#)
  - InitializeLevels, [48](#)
  - PrintLevels, [48](#)
- NuclearLevels.h
  - LevelsTable, [88](#)
- NuclearMass, [48](#)
  - CalculateLDMMass, [48](#)
  - FindElement, [48](#)
  - FindMass, [48](#)
  - FindZ, [49](#)
  - HighestBoundEnergy, [49](#)
  - InitializeElements, [49](#)
  - InitializeMasses, [49](#)
  - MassDifference, [49](#)
  - MicroEnergyCorr, [49](#)
  - NeutronPairingGap, [49](#)
  - ProtonPairingGap, [49](#)
  - QValue, [49](#)
- NuclearMass.cpp
  - ELEMENT, [96](#)
  - HAS\_EXP\_MASS, [96](#)
  - HAS\_TH\_MASS, [96](#)
- NuclearMass.h
  - ElementTable, [89](#)
  - MassTable, [89](#)
- nuclearTemp\_
  - LevelDensity, [43](#)
- ODE\_integration, [50](#)
  - ODE\_integration, [50](#)
  - ODE\_integration, [50](#)
  - operator(), [50](#)
- ompver\_str
  - OpenMPCheckVersion.c, [74](#)
  - OpenMPCheckVersion.cpp, [74](#)
- OpenMPCheckVersion.c
  - main, [73](#)
  - ompver\_str, [74](#)
- OpenMPCheckVersion.cpp
  - main, [74](#)
  - ompver\_str, [74](#)
- OpenMPTryFlag.c
  - main, [75](#)
- OpenMPTryFlag.cpp
  - main, [75](#)
- operator<
  - MassKey, [45](#)
  - SLPair, [62](#)
- operator\*
  - complex\_functions.H, [78](#)
- operator()
  - CoulFunc, [15](#)
  - GammaTransmissionFunc, [33](#)
  - int\_double\_pair\_compare, [38](#)
  - LevelDensity, [42](#)
  - ODE\_integration, [50](#)
  - ParticleHoleLevelDensity, [51](#)
  - ParticleTransmissionFunc, [53](#)
  - std::equal\_to< MassKey >, [30](#)
  - std::tr1::hash< MassKey >, [36](#)
  - TransmissionFunc, [65](#)
- operator+
  - complex\_functions.H, [78](#)
- operator-
  - complex\_functions.H, [78](#)
- operator/
  - complex\_functions.H, [78](#), [79](#)
- operator==
  - complex\_functions.H, [79](#)
- PEShift
  - CoulFunc, [16](#)
- PEShift\_dE
  - CoulFunc, [16](#)
- PLATFORM\_ID
  - CMakeCCompilerId.c, [70](#)
  - CMakeCXXCompilerId.cpp, [71](#)
- pType\_
  - EntrancePairs, [30](#)
  - ParticleTransmissionFunc, [54](#)
- pairIndex\_
  - CDFEntry, [14](#)
  - PreEqCDFEntry, [57](#)
- PairingCorrection
  - ParticleHoleLevelDensity, [51](#)

- parity\_
  - ParticleTransmissionFunc, 54
  - PreEqSpinRatePair, 59
  - SpinRatePair, 63
- parseCommandLineForDecay
  - Sapphire.cpp, 99
- parseCommandLineForOptions
  - Sapphire.cpp, 99
- parseCommandLineForXS
  - Sapphire.cpp, 100
- particleEnergy\_
  - DecayProduct, 28
- particleEnergyCM\_
  - DecayProduct, 28
- ParticleHoleLevelDensity, 50
  - FiniteDepth, 51
  - gNu, 51
  - gPi, 51
  - operator(), 51
  - PairingCorrection, 51
  - ParticleHoleLevelDensity, 51
  - ParticleHoleLevelDensity, 51
  - PauliCorrection, 51
- particleMomentumX\_
  - DecayProduct, 28
- particleMomentumY\_
  - DecayProduct, 28
- particleMomentumZ\_
  - DecayProduct, 28
- particlePhiCM\_
  - DecayProduct, 28
- particleThetaCM\_
  - DecayProduct, 28
- ParticleTransmissionFunc, 51
  - ~ParticleTransmissionFunc, 52
  - CalcSLDependentFunctions, 52
  - CalcTransmission, 52
  - CreateParticleTransmissionFunc, 53
  - IsValid, 53
  - m1\_, 54
  - operator(), 53
  - pType\_, 54
  - parity\_, 54
  - ParticleTransmissionFunc, 52
  - ParticleTransmissionFunc, 52
  - redmass\_, 54
  - SetAlphaFormalism, 53
  - SetNeutronFormalism, 53
  - SetPorterThomas, 53
  - SetProtonFormalism, 53
  - spin\_, 54
  - z1\_, 54
- ParticleTransmissionFunc.cpp
  - randomSeed, 97
- particleType\_
  - DecayProduct, 28
- PauliCorrection
  - ParticleHoleLevelDensity, 51
- Penetrability
  - CoulFunc, 16
- Pi
  - InitialNucleusData, 37
- pi
  - Constants.h, 84
- Pi\_
  - DecayProduct, 28
  - Level, 40
- piFinal\_
  - TransmissionFunc, 66
- piInitial\_
  - TransmissionFunc, 66
- porterThomas\_
  - GammaTransmissionFunc, 34
- Potential, 54
  - ~Potential, 55
  - boundaryRadius\_, 56
  - CalcBeta, 55
  - CalcTransmission, 55
  - Calculate, 55
  - coulFunc\_, 56
  - coulombRadius\_, 57
  - GetBoundaryRadius, 56
  - GetRMax, 56
  - GetRedMass, 56
  - GetZ1Z2, 56
  - NormalizeInternally, 56
  - NormalizeOverAllSpace, 56
  - Potential, 55
  - Solve, 56
- preEq
  - InitialNucleusData, 37
- PreEqCDFEntry, 57
  - energy\_, 57
  - pairIndex\_, 57
  - PreEqCDFEntry, 57
  - PreEqCDFEntry, 57
  - value\_, 57
- PreEqDecayer, 57
  - ~PreEqDecayer, 58
  - Decay, 58
  - GetMaxL, 58
  - PreEqDecayer, 58
  - PreEqDecayer, 58
  - PrintCDF, 58
  - SetCrossSection, 58
  - SetMaxL, 58
- PreEqDecayer.cpp
  - randomSeed, 98
- PreEqSpinRatePair, 58
  - A\_, 59
  - integral\_, 59
  - neutronHoleNumber\_, 59
  - neutronNumber\_, 59
  - parity\_, 59
  - PreEqSpinRatePair, 59
  - PreEqSpinRatePair, 59

- protonHoleNumber\_, 59
- protonNumber\_, 59
- qValue\_, 59
- rateFunc\_, 59
- spin\_, 59
- Z\_, 59
- PreEqTransitionRateFunc, 60
  - ~PreEqTransitionRateFunc, 60
  - CumulativeSum, 60
  - Integral, 60
  - PreEqTransitionRateFunc, 60
  - PreEqTransitionRateFunc, 60
- precision
  - complex\_functions.H, 79
- previous\_
  - TransmissionFunc, 66
- PrintCDF
  - Decayer, 26
  - PreEqDecayer, 58
- PrintCrossSections
  - CrossSection, 20
- PrintDecays
  - DecayController, 23
- PrintFunctions
  - Decayer, 26
- printHelp
  - Sapphire.cpp, 100
- PrintLevels
  - NuclearLevels, 48
- PrintReactionRates
  - CrossSection, 20
- PrintTransmissionTerms
  - CrossSection, 20
- probability\_
  - GammaTransition, 32
- proton\_
  - CrossSectionValues, 22
- ProtonEntranceWidth
  - Decayer, 26
- protonEntranceWidth
  - DecayData, 24
- protonHoleNumber\_
  - PreEqSpinRatePair, 59
- protonNumber\_
  - PreEqSpinRatePair, 59
- ProtonPairingGap
  - NuclearMass, 49
- protonStellar\_
  - CrossSectionValues, 22
- ProtonTotalWidth
  - Decayer, 26
- protonTotalWidth
  - DecayData, 24
- QValue
  - NuclearMass, 49
- qValue\_
  - PreEqSpinRatePair, 59
  - SpinRatePair, 63
- r0\_
  - LevelDensity, 43
- radiusLast
  - CoulFunc, 16
- randomSeed
  - DecayController.cpp, 93
  - Decayer.cpp, 94
  - GammaTransmissionFunc.cpp, 94
  - ParticleTransmissionFunc.cpp, 97
  - PreEqDecayer.cpp, 98
  - Sapphire.cpp, 100
- rateFunc\_
  - PreEqSpinRatePair, 59
  - SpinRatePair, 63
- RauscherLevelDensity, 60
  - ~RauscherLevelDensity, 61
  - CalcBackShift, 61
  - CalcDensityParam, 61
  - CalcNuclearTemp, 61
  - RauscherLevelDensity, 61
  - RauscherLevelDensity, 61
- redmass
  - CoulFunc, 16
- redmass\_
  - ParticleTransmissionFunc, 54
- s\_
  - SLPair, 62
- SIGN
  - complex\_functions.H, 77
- SLPair, 61
  - l\_, 62
  - operator<, 62
  - s\_, 62
  - SLPair, 62
  - SLPair, 62
- STRINGIFY
  - CMakeCCompilerId.c, 70
  - CMakeCXXCompilerId.cpp, 72
- STRINGIFY\_HELPER
  - CMakeCCompilerId.c, 70
  - CMakeCXXCompilerId.cpp, 72
- Sapphire.cpp
  - EntrancePairs, 99
  - Initialize, 99
  - main, 99
  - parseCommandLineForDecay, 99
  - parseCommandLineForOptions, 99
  - parseCommandLineForXS, 100
  - printHelp, 100
  - randomSeed, 100
- SapphireMPITypes.h
  - BOOST\_IS\_MPI\_DATATYPE, 91
  - SapphireTagDone, 91
  - SapphireTagProcess, 91
  - SapphireTagResults, 91
  - SapphireTags\_t, 91
- SapphireTagDone
  - SapphireMPITypes.h, 91

- SapphireTagProcess
  - SapphireMPITypes.h, 91
- SapphireTagResults
  - SapphireMPITypes.h, 91
- SapphireTags\_t
  - SapphireMPITypes.h, 91
- serialize
  - boost::serialization, 11
- SetAlphaFormalism
  - ParticleTransmissionFunc, 53
- SetCalculateGammaCutoff
  - CrossSection, 21
- SetCrossSection
  - Decayer, 26
  - PreEqDecayer, 58
- SetEGDRType
  - GammaTransmissionFunc, 33
- SetGammaCutoffEnergy
  - TransitionRateFunc, 64
- setLast
  - CoulFunc, 16
- SetMaxL
  - Decayer, 26
  - PreEqDecayer, 58
- SetNeutronFormalism
  - ParticleTransmissionFunc, 53
- SetPorterThomas
  - GammaTransmissionFunc, 33
  - ParticleTransmissionFunc, 53
- SetProtonFormalism
  - ParticleTransmissionFunc, 53
- SetResidualAlpha
  - CrossSection, 21
- SetResidualGamma
  - CrossSection, 21
- SetResidualNeutron
  - CrossSection, 21
- SetResidualProton
  - CrossSection, 21
- Setup.cpp
  - Initialize, 101
- sigma\_l\_calc
  - complex\_functions.cpp, 81
  - complex\_functions.H, 79
- sin\_chi\_calc
  - complex\_functions.cpp, 81
  - complex\_functions.H, 79
- Solve
  - Potential, 56
- sourceDirectory
  - build/generated/Sapphire\_config.h, 75
  - generated/Sapphire\_config.h, 76
- spin\_
  - ParticleTransmissionFunc, 54
  - PreEqSpinRatePair, 59
  - SpinRatePair, 63
- SpinRatePair, 62
  - A\_, 62
  - integral\_, 62
  - parity\_, 63
  - qValue\_, 63
  - rateFunc\_, 63
  - spin\_, 63
  - SpinRatePair, 62
  - SpinRatePair, 62
  - Z\_, 63
- sqrt\_precision
  - complex\_functions.H, 79
- std, 11
- std::equal\_to< MassKey >, 30
  - operator(), 30
- std::tr1, 12
- std::tr1::hash< MassKey >, 36
  - operator(), 36
- temperature
  - gsl\_partfunc\_params, 35
  - gsl\_reactionrate\_params, 36
- thMass\_
  - MassEntry, 45
- TotalLevelDensity
  - LevelDensity, 42
- totalWidthForCorrection\_
  - TransmissionFunc, 66
- TransitionRateFunc, 63
  - ~TransitionRateFunc, 63
  - CalcLevelDensity, 64
  - CalcTotalLevelDensity, 64
  - CalcTransmissionFunc, 64
  - CumulativeSum, 64
  - ExclusiveBranching, 64
  - Function, 64
  - GetGammaCutoffEnergy, 64
  - GroundStateTransmission, 64
  - Integral, 64
  - SetGammaCutoffEnergy, 64
  - TransitionRateFunc, 63
  - TransitionRateFunc, 63
- TransmissionFunc, 64
  - ~TransmissionFunc, 65
  - IsValid, 65
  - jFinal\_, 66
  - jInitial\_, 66
  - m2\_, 66
  - maxL\_, 66
  - operator(), 65
  - piFinal\_, 66
  - piInitial\_, 66
  - previous\_, 66
  - totalWidthForCorrection\_, 66
  - TransmissionFunc, 65
  - TransmissionFunc, 65
  - uncorrTotalWidthForCorrection\_, 66
  - uncorrTotalWidthSqrdForCorrection\_, 66
  - z2\_, 66
- uconv

- Constants.h, [84](#)
- uncorrTotalWidthForCorrection\_
  - TransmissionFunc, [66](#)
- uncorrTotalWidthSqrdForCorrection\_
  - TransmissionFunc, [66](#)
- useSpline
  - gsl\_reactionrate\_params, [36](#)
- value\_
  - CDFEntry, [14](#)
  - PreEqCDFEntry, [57](#)
- vector\_c
  - Constants.h, [83](#)
- vector\_matrix\_c
  - Constants.h, [83](#)
- vector\_matrix\_r
  - Constants.h, [83](#)
- vector\_r
  - Constants.h, [83](#)
- W\_
  - GDRParameters, [35](#)
- X\_
  - XYPair, [67](#)
- XYPair, [67](#)
  - X\_, [67](#)
  - XYPair, [67](#)
  - XYPair, [67](#)
  - Y\_, [67](#)
- Y\_
  - XYPair, [67](#)
- Z
  - InitialNucleusData, [37](#)
- z1
  - CoulFunc, [16](#)
- z1\_
  - ParticleTransmissionFunc, [54](#)
- z2
  - CoulFunc, [16](#)
- z2\_
  - TransmissionFunc, [66](#)
- Z\_
  - DecayProduct, [28](#)
  - EntrancePairs, [30](#)
  - LevelDensity, [43](#)
  - MassKey, [45](#)
  - PreEqSpinRatePair, [59](#)
  - SpinRatePair, [63](#)
- zeta\_
  - LevelDensity, [43](#)