

My Project

Generated by Doxygen 1.7.6.1

Mon Oct 28 2013 09:25:38

Contents

1	Todo List	1
2	Namespace Index	3
2.1	Namespace List	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	PRPSError Namespace Reference	9
5.1.1	Variable Documentation	9
5.1.1.1	critical	9
5.1.1.2	general	9
5.1.1.3	okay	9
5.2	PRPSError::FileIO Namespace Reference	9
5.2.1	Variable Documentation	10
5.2.1.1	fnf	10
5.2.1.2	generalError	10
5.2.1.3	inputmalformed	10
5.2.1.4	okay	10
5.3	PRPSEvolution Namespace Reference	10
5.3.1	Detailed Description	11
5.3.2	Enumeration Type Documentation	11
5.3.2.1	NormalizationMethods	11

5.3.3	Function Documentation	12
5.3.3.1	ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION	12
5.3.3.2	ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION	12
5.3.3.3	idealPhaseFromInput	12
5.3.3.4	rCoords	12
5.3.3.5	rPoints	12
5.3.4	Variable Documentation	12
5.3.4.1	ANTENNA_AMOUNT	12
5.3.4.2	CALIBRATION_POINTS_AVAILABLE	12
5.3.4.3	DATA_NV	12
5.3.4.4	DEFAULT_MIN_GROUP_SIZE	12
5.3.4.5	EXPECTED_LINES_CALIBRATION_FILE	12
5.3.4.6	EXPECTED_LINES_COORD_FILE	13
5.3.4.7	EXPECTED_LINES_MEASUREMENT_FILE	13
5.3.4.8	EXPECTED_LINES_SYSTEM_INI_FILE	13
5.3.4.9	EXPECTED_VALUES_CALIBRATION_FILE	13
5.3.4.10	EXPECTED_VALUES_COORD_FILE	13
5.3.4.11	EXPECTED_VALUES_MEASUREMENT_FILE	13
5.3.4.12	MAT_COLS	13
5.3.4.13	MAT_ROWS	13
5.3.4.14	pi	13
5.4	PRPSEvolution::Calibration Namespace Reference	13
5.5	PRPSEvolution::Exceptions Namespace Reference	13
5.6	PRPSEvolution::Exceptions::Calibration Namespace Reference	13
5.7	PRPSEvolution::Exceptions::FileIO Namespace Reference	13
5.8	PRPSEvolution::Exceptions::General Namespace Reference	14
5.9	PRPSEvolution::Exceptions::Permutation Namespace Reference	14
5.10	PRPSEvolution::Exceptions::Solve Namespace Reference	14
5.11	PRPSEvolution::Models Namespace Reference	14
5.11.1	Function Documentation	15
5.11.1.1	ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION	15
5.11.1.2	ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION	15
5.11.1.3	ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION	15
5.11.1.4	ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION	15

5.11.1.5	ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION	15
5.11.2	Variable Documentation	15
5.11.2.1	printYN	15
5.12	PRPSEvolution::Permute Namespace Reference	15
5.12.1	Function Documentation	15
5.12.1.1	Factorial	15
5.12.1.2	next_combination	15
5.12.2	Variable Documentation	16
5.12.2.1	MAX_PERMUTATION_AMOUNT	16
5.13	PRPSEvolution::Positioning Namespace Reference	16
5.14	PRPSEvolution::Solve Namespace Reference	16
5.14.1	Enumeration Type Documentation	17
5.14.1.1	ESStrategy	17
5.14.1.2	Models	17
5.14.1.3	SelectBy	17
5.14.2	Function Documentation	18
5.14.2.1	meanFromVector	18
5.14.3	Variable Documentation	18
5.14.3.1	_i_	18
5.14.3.2	nConfigsForProcessing	18
5.14.3.3	wMutex	18
5.15	PRPSEvolution::Support Namespace Reference	18
6	Class Documentation	19
6.1	PRPSEvolution::Permute::AntennaPermutations< N_MAT, T > Struct Template Reference	19
6.1.1	Constructor & Destructor Documentation	19
6.1.1.1	AntennaPermutations	19
6.1.2	Member Function Documentation	20
6.1.2.1	dump_matrix	20
6.1.2.2	dump_matrix_2_file	20
6.1.3	Member Data Documentation	20
6.1.3.1	mat	20
6.1.3.2	names	20

6.2	PRPSEvolution::Constants Struct Reference	20
6.2.1	Constructor & Destructor Documentation	20
6.2.1.1	Constants	20
6.2.1.2	Constants	21
6.2.2	Member Data Documentation	21
6.2.2.1	a_1	21
6.2.2.2	a_2	21
6.2.2.3	c_0	21
6.2.2.4	f_mess	21
6.2.2.5	lambda	21
6.3	PRPSEvolution::Positioning::CoordContainer< N, T > Struct Template Reference	21
6.3.1	Member Typedef Documentation	22
6.3.1.1	value_type	22
6.3.2	Constructor & Destructor Documentation	22
6.3.2.1	CoordContainer	22
6.3.2.2	CoordContainer	22
6.3.3	Member Function Documentation	22
6.3.3.1	operator[]	22
6.3.4	Member Data Documentation	22
6.3.4.1	x_	22
6.3.4.2	y_	23
6.3.4.3	z_	23
6.4	PRPSEvolution::Models::EvolutionaryCalibration Struct Reference	23
6.4.1	Member Typedef Documentation	23
6.4.1.1	base_type	23
6.4.1.2	ObjectiveFunctionType	24
6.4.2	Constructor & Destructor Documentation	24
6.4.2.1	EvolutionaryCalibration	24
6.4.3	Member Function Documentation	24
6.4.3.1	configure	24
6.4.3.2	eval	24
6.4.3.3	hasScalableDimensionality	24
6.4.3.4	mkII	24

6.4.3.5	name	24
6.4.3.6	numberOfVariables	24
6.4.3.7	proposeStartingPoint	24
6.4.3.8	setMat	24
6.4.3.9	setNumberOfVariables	25
6.4.3.10	setParams	25
6.4.3.11	setVec	25
6.5	PRPSEvolution::Exceptions::FileIO::FileNotFoundException Reference . . .	25
6.5.1	Member Function Documentation	25
6.5.1.1	what	25
6.6	PRPSEvolution::Support::FitnessPlaneCalculator< N > Class - Template Reference	26
6.6.1	Constructor & Destructor Documentation	26
6.6.1.1	FitnessPlaneCalculator	26
6.6.2	Member Function Documentation	26
6.6.2.1	calculate	27
6.6.2.2	rPlaneLimitsFromFile	27
6.6.2.3	setLimits	27
6.6.3	Member Data Documentation	27
6.6.3.1	f_count	27
6.6.3.2	f_pathBase	27
6.6.3.3	limits	27
6.7	PRPSEvolution::Support::FitnessPlaneLimits< N > Struct Template - Reference	28
6.7.1	Member Function Documentation	28
6.7.1.1	dump	28
6.7.2	Member Data Documentation	28
6.7.2.1	idx1	28
6.7.2.2	idx2	28
6.7.2.3	increment	28
6.7.2.4	lower	28
6.7.2.5	upper	28
6.8	PRPSEvolution::Exceptions::FileIO::MalformedInput Struct Reference . . .	29
6.8.1	Member Function Documentation	29

6.8.1.1	what	29
6.9	PRPSEvolution::Normalizer< N, T > Struct Template Reference	29
6.9.1	Detailed Description	29
6.9.2	Constructor & Destructor Documentation	29
6.9.2.1	Normalizer	29
6.9.3	Member Function Documentation	30
6.9.3.1	normalize	30
6.9.3.2	setLambda	30
6.9.3.3	setSelectIdealPoint	30
6.10	PRPSEvolution::Exceptions::General::NotImplemented Struct Reference	30
6.10.1	Detailed Description	30
6.10.2	Member Function Documentation	31
6.10.2.1	what	31
6.11	PRPSEvolution::Exceptions::FileIO::OutputFailure Struct Reference	31
6.11.1	Member Function Documentation	31
6.11.1.1	what	31
6.12	PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T > Struct Template Reference	31
6.12.1	Detailed Description	32
6.12.2	Constructor & Destructor Documentation	32
6.12.2.1	performCalibration	33
6.12.3	Member Data Documentation	33
6.12.3.1	A	33
6.12.3.2	AntennaCoordinates	33
6.12.3.3	c_k0	33
6.12.3.4	CalibrationPointsCoordinates	33
6.12.3.5	distances	33
6.13	PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T > Struct Template Reference	34
6.13.1	Detailed Description	35
6.13.2	Constructor & Destructor Documentation	35
6.13.2.1	permuteAntennas	35
6.13.3	Member Function Documentation	35
6.13.3.1	compute_d_k0_Mat	35

6.13.3.2	computeMatrix	36
6.13.3.3	computePermutations	36
6.13.3.4	dump_matrices_2_file	36
6.13.3.5	dumpConfigurationsToFile	37
6.13.3.6	rCoordFile	37
6.13.4	Member Data Documentation	37
6.13.4.1	AntennaCoordinates	37
6.13.4.2	configurations	37
6.13.4.3	d_k0_mat	37
6.13.4.4	ref	37
6.13.4.5	systemConstants	38
6.14	PRPSEvolution::Solve::PostProcessing Class Reference	38
6.14.1	Constructor & Destructor Documentation	38
6.14.1.1	PostProcessing	38
6.15	PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_-Measure > Class Template Reference	38
6.15.1	Constructor & Destructor Documentation	39
6.15.1.1	PreProcessing	39
6.15.2	Member Function Documentation	40
6.15.2.1	possibleGroupSize	40
6.15.3	Member Data Documentation	40
6.15.3.1	antennas	40
6.15.3.2	antennasPerGroup	40
6.15.3.3	matGroups	40
6.15.3.4	matrices	40
6.15.3.5	nameGroups	41
6.15.3.6	names	41
6.15.3.7	vectorGroups	41
6.15.3.8	vectors	41
6.16	PRPSEvolution::Solve::ProblemDimensions Struct Reference	41
6.16.1	Detailed Description	42
6.16.2	Member Data Documentation	42
6.16.2.1	Calibration	42
6.16.2.2	Rosenbrock	42

6.16.2.3	Sphere	42
6.16.2.4	WholeTomato	42
6.16.2.5	WholeTomatoMkl	42
6.16.2.6	WholeTomatoMkl_A	42
6.16.2.7	WholeTomatoMkl_B	42
6.16.2.8	WholeTomatoMkII	42
6.16.2.9	WholeTomatoReduced	42
6.17	PRPSEvolution::Solve::Process Class Reference	42
6.17.1	Detailed Description	43
6.17.2	Constructor & Destructor Documentation	43
6.17.2.1	Process	43
6.17.2.2	Process	43
6.17.3	Member Function Documentation	43
6.17.3.1	findSolution	44
6.17.3.2	findSolutionCMA_ES_Mkl	44
6.17.3.3	findSolutionCMA_ES_MkII	44
6.17.3.4	findSolutionSolveSingle	45
6.17.3.5	findSolutionSphere	45
6.17.3.6	getLastSolutionFitness	46
6.17.3.7	incrementFileCounter	46
6.17.3.8	resetFileCounter	46
6.17.3.9	setMinSolutionFitness	47
6.17.3.10	setSeed	47
6.17.3.11	sq	47
6.17.4	Member Data Documentation	47
6.17.4.1	f_count	47
6.18	PRPSEvolution::Solve::Process_MkII Class Reference	48
6.18.1	Constructor & Destructor Documentation	49
6.18.1.1	Process_MkII	49
6.18.1.2	Process_MkII	49
6.18.1.3	Process_MkII	49
6.18.1.4	Process_MkII	50
6.18.1.5	Process_MkII	50
6.18.1.6	Process_MkII	50

6.18.2 Member Function Documentation	50
6.18.2.1 calcFitnessMkII	51
6.18.2.2 calcFitnessMkIIReduced	51
6.18.2.3 cWholeTomatoMkII	52
6.18.2.4 EvolutionaryCalibration	52
6.18.2.5 incrementFileCounter	53
6.18.2.6 init	53
6.18.2.7 Process_MkII_test	54
6.18.2.8 resetFileCounter	54
6.18.2.9 setAntennaCoords	54
6.18.2.10 setEpsilon	54
6.18.2.11 setFileCounter	54
6.18.2.12 setMaxEvaluations	54
6.18.2.13 setOutputFilePath	55
6.18.2.14 setOutputFilePathBase	55
6.18.2.15 setPrintLastOnly	55
6.18.2.16 toggleVariant	55
6.18.2.17 WholeTomatoMkI_A	56
6.18.2.18 WholeTomatoMkI_B	56
6.18.2.19 WholeTomatoMkII	56
6.18.2.20 WholeTomatoMkII_B	57
6.18.2.21 WholeTomatoReduced	58
6.19 PRPSEvolution::Solvresult_t< T_Store1, T_Store2, T_Return > Struct Template Reference	58
6.19.1 Detailed Description	59
6.19.2 Member Data Documentation	59
6.19.2.1 converged	59
6.19.2.2 duration	59
6.19.2.3 fitness	59
6.19.2.4 iterations	59
6.19.2.5 valCont	59
6.19.2.6 valDis	59
6.20 PRPSEvolution::System Struct Reference	60
6.20.1 Constructor & Destructor Documentation	60

6.20.1.1	System	60
6.20.1.2	System	61
6.20.2	Member Function Documentation	61
6.20.2.1	rPRPSIniFile	61
6.20.3	Member Data Documentation	61
6.20.3.1	constants	61
6.20.3.2	fn	61
6.21	PRPSEvolution::Solve::Ueber9000< T > Struct Template Reference . .	61
6.21.1	Detailed Description	62
6.21.2	Constructor & Destructor Documentation	63
6.21.2.1	Ueber9000	63
6.21.2.2	Ueber9000	63
6.21.2.3	Ueber9000	63
6.21.2.4	Ueber9000	63
6.21.2.5	Ueber9000	64
6.21.3	Member Function Documentation	64
6.21.3.1	fitnessAckley	64
6.21.3.2	fitnessRosenbrock	64
6.21.3.3	fitnessSphere	64
6.21.3.4	fitnessSphereMkII	64
6.21.3.5	parseldxFromNames	65
6.21.3.6	SuWi_PositionVariation	65
6.21.3.7	SuWi_WavenumberVariation	65
6.21.3.8	WholeTomato	65
6.21.3.9	WholeTomato	66
6.21.3.10	WholeTomatoMkl	66
6.21.3.11	WholeTomatoMkII	67
6.21.3.12	WholeTomatoMkII	67
6.21.3.13	WholeTomatoMkII	68
6.21.3.14	WholeTomatoMkII	68
6.21.4	Member Data Documentation	69
6.21.4.1	A	69
6.21.4.2	b	69
6.21.4.3	Dimension	69

6.21.4.4	evaluate	69
6.21.4.5	evaluateMkl	69
6.21.4.6	evaluateMkII	69
6.21.4.7	evaluateMkIII	70
6.21.4.8	evaluations	70
6.21.4.9	idxs	70
6.21.4.10	names	70
6.22	PRPSEvolution::WholeTomatoMkl_A Struct Reference	70
6.22.1	Constructor & Destructor Documentation	71
6.22.1.1	WholeTomatoMkl_A	71
6.22.2	Member Function Documentation	71
6.22.2.1	configure	71
6.22.2.2	eval	71
6.22.2.3	hasScalableDimensionality	71
6.22.2.4	mkl	71
6.22.2.5	name	71
6.22.2.6	numberOfVariables	71
6.22.2.7	proposeStartingPoint	71
6.22.2.8	setMat	72
6.22.2.9	setNumberOfVariables	72
6.22.2.10	setParams	72
6.22.2.11	setVec	72
6.23	PRPSEvolution::WholeTomatoMkl_B Struct Reference	72
6.23.1	Constructor & Destructor Documentation	72
6.23.1.1	WholeTomatoMkl_B	72
6.23.2	Member Function Documentation	72
6.23.2.1	configure	73
6.23.2.2	eval	73
6.23.2.3	hasScalableDimensionality	73
6.23.2.4	mkl	73
6.23.2.5	name	73
6.23.2.6	numberOfVariables	73
6.23.2.7	proposeStartingPoint	73
6.23.2.8	setMat	73

6.23.2.9	setNumberOfVariables	73
6.23.2.10	setParams	74
6.23.2.11	setVec	74
6.24	PRPSEvolution::Models::WholeTomatoMkII Struct Reference	74
6.24.1	Member Typedef Documentation	75
6.24.1.1	base_type	75
6.24.1.2	ObjectiveFunctionType	75
6.24.2	Constructor & Destructor Documentation	75
6.24.2.1	WholeTomatoMkII	75
6.24.3	Member Function Documentation	75
6.24.3.1	configure	75
6.24.3.2	constraints	75
6.24.3.3	eval	75
6.24.3.4	hasScalableDimensionality	75
6.24.3.5	mkII	76
6.24.3.6	name	76
6.24.3.7	numberOfVariables	76
6.24.3.8	proposeStartingPoint	76
6.24.3.9	setNumberOfVariables	76
6.24.3.10	setParams	77
6.24.3.11	setParams	77
6.25	PRPSEvolution::Models::WholeTomatoMkII_B Struct Reference	77
6.25.1	Member Typedef Documentation	78
6.25.1.1	base_type	78
6.25.1.2	ObjectiveFunctionType	78
6.25.2	Constructor & Destructor Documentation	78
6.25.2.1	WholeTomatoMkII_B	78
6.25.3	Member Function Documentation	78
6.25.3.1	configure	78
6.25.3.2	eval	78
6.25.3.3	hasScalableDimensionality	79
6.25.3.4	mkII	79
6.25.3.5	name	79
6.25.3.6	numberOfVariables	79

6.25.3.7	proposeStartingPoint	79
6.25.3.8	setIdx	79
6.25.3.9	setMats	79
6.25.3.10	setNames	79
6.25.3.11	setNumberOfVariables	80
6.25.3.12	setParams	80
6.25.3.13	setParams	80
6.25.3.14	setVecs	80
6.26	PRPSEvolution::Models::WholeTomatoMkIII Struct Reference	80
6.26.1	Member Typedef Documentation	81
6.26.1.1	base_type	81
6.26.1.2	ObjectiveFunctionType	81
6.26.2	Constructor & Destructor Documentation	81
6.26.2.1	WholeTomatoMkIII	82
6.26.3	Member Function Documentation	82
6.26.3.1	configure	82
6.26.3.2	constrains	82
6.26.3.3	eval	82
6.26.3.4	hasScalableDimensionality	82
6.26.3.5	mkII	83
6.26.3.6	name	83
6.26.3.7	numberOfVariables	83
6.26.3.8	proposeStartingPoint	83
6.26.3.9	setIdx	83
6.26.3.10	setMats	83
6.26.3.11	setNames	83
6.26.3.12	setNumberOfVariables	83
6.26.3.13	setParams	83
6.26.3.14	setParams	83
6.26.3.15	setVecs	84
6.27	PRPSEvolution::Models::WholeTomatoReduced Struct Reference	84
6.27.1	Member Typedef Documentation	85
6.27.1.1	base_type	85
6.27.1.2	ObjectiveFunctionType	85

6.27.2	Constructor & Destructor Documentation	85
6.27.2.1	WholeTomatoReduced	85
6.27.3	Member Function Documentation	85
6.27.3.1	calcWavenumbers2	85
6.27.3.2	configure	85
6.27.3.3	constrains	85
6.27.3.4	eval	86
6.27.3.5	hasScalableDimensionality	86
6.27.3.6	name	86
6.27.3.7	numberOfVariables	86
6.27.3.8	proposeStartingPoint	86
6.27.3.9	reduced	86
6.27.3.10	setNumberOfVariables	87
6.27.3.11	setParams	87
6.27.3.12	setParams	87
7	File Documentation	89
7.1	trunk/CMakeFiles/CompilerIdC/CMakeCCompilerId.c File Reference	89
7.1.1	Define Documentation	89
7.1.1.1	ARCHITECTURE_ID	89
7.1.1.2	COMPILER_ID	89
7.1.1.3	PLATFORM_ID	89
7.1.2	Function Documentation	89
7.1.2.1	main	90
7.1.3	Variable Documentation	90
7.1.3.1	info_arch	90
7.1.3.2	info_compiler	90
7.1.3.3	info_platform	90
7.2	trunk/CMakeFiles/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference	90
7.2.1	Define Documentation	90
7.2.1.1	ARCHITECTURE_ID	90
7.2.1.2	COMPILER_ID	90
7.2.1.3	PLATFORM_ID	90

7.2.2	Function Documentation	90
7.2.2.1	main	90
7.2.3	Variable Documentation	90
7.2.3.1	info_arch	90
7.2.3.2	info_compiler	91
7.2.3.3	info_platform	91
7.3	trunk/include/coords.h File Reference	91
7.4	trunk/include/prps.h File Reference	92
7.4.1	Variable Documentation	92
7.4.1.1	ANTENNA_AMOUNT	92
7.4.1.2	EXPECTED_LINES	92
7.4.1.3	EXPECTED_VALUES	92
7.5	trunk/include/PRPSError.h File Reference	92
7.5.1	Detailed Description	93
7.6	trunk/include/PRPSEvolution.h File Reference	93
7.6.1	Detailed Description	94
7.7	trunk/include/PRPSEvolutionCalibrationExceptions.h File Reference	94
7.8	trunk/include/PRPSEvolutionFIOExceptions.h File Reference	95
7.9	trunk/include/PRPSEvolutionGeneralExceptions.h File Reference	96
7.10	trunk/include/PRPSEvolutionPermutationExceptions.h File Reference	97
7.11	trunk/include/PRPSEvolutionSolveExceptions.h File Reference	99
7.12	trunk/libCalibration/calib.cpp File Reference	100
7.13	trunk/libCalibration/calib.h File Reference	100
7.13.1	Detailed Description	101
7.14	trunk/libFitnessPlanCalculator/FitnessPlaneCalculator.h File Reference	101
7.15	trunk/libNormalizer/normalizer.cpp File Reference	102
7.16	trunk/libNormalizer/normalizer.h File Reference	103
7.16.1	Detailed Description	103
7.17	trunk/libPermute/permute.cpp File Reference	104
7.17.1	Function Documentation	104
7.17.1.1	test2	104
7.18	trunk/libPermute/permute.h File Reference	104
7.18.1	Detailed Description	105
7.19	trunk/libPRPSSystem/prpsevolutionsystem.cpp File Reference	106

7.20	trunk/libPRPSSystem/prpsevolutionsystem.h File Reference	106
7.21	trunk/libSolve/ObjectFunctions.cpp File Reference	106
7.22	trunk/libSolve/ObjectFunctions.h File Reference	106
7.23	trunk/libSolve/Objectivefunctions/EvolutionaryCalibration.cpp File - Reference	107
7.24	trunk/libSolve/Objectivefunctions/EvolutionaryCalibration.h File Reference	108
7.25	trunk/libSolve/Objectivefunctions/WholeTomatoMkI.cpp File Reference .	109
7.26	trunk/libSolve/Objectivefunctions/WholeTomatoMkI_A.h File Reference .	109
7.27	trunk/libSolve/Objectivefunctions/WholeTomatoMkI_B.h File Reference .	111
7.28	trunk/libSolve/Objectivefunctions/WholeTomatoMkII.cpp File Reference .	112
7.29	trunk/libSolve/Objectivefunctions/WholeTomatoMkII.h File Reference .	113
7.30	trunk/libSolve/Objectivefunctions/WholeTomatoMkII_B.cpp File Reference	114
7.31	trunk/libSolve/Objectivefunctions/WholeTomatoMkII_B.h File Reference .	115
7.32	trunk/libSolve/Objectivefunctions/WholeTomatoMkIII.cpp File Reference .	116
7.33	trunk/libSolve/Objectivefunctions/WholeTomatoMkIII.h File Reference .	117
7.34	trunk/libSolve/Objectivefunctions/WholeTomatoReduced.cpp File - Reference	118
7.34.1	Define Documentation	119
7.34.1.1	_WT_CONSTRAIN_HARD_	119
7.35	trunk/libSolve/Objectivefunctions/WholeTomatoReduced.h File Reference	119
7.36	trunk/libSolve/postprocessing.cpp File Reference	120
7.37	trunk/libSolve/postprocessing.h File Reference	121
7.38	trunk/libSolve/preprocessing.cpp File Reference	122
7.39	trunk/libSolve/preprocessing.h File Reference	122
7.40	trunk/libSolve/process.cpp File Reference	123
7.41	trunk/libSolve/process.h File Reference	123
7.42	trunk/libSolve/processMkII.cpp File Reference	123
7.43	trunk/libSolve/processMkII.h File Reference	123
7.43.1	Define Documentation	124
7.43.1.1	SOLVE	125
7.43.1.2	SOLVE_AND_WRITE	125
7.43.1.3	STUFF	125
7.44	trunk/libSolve/solve.cpp File Reference	125
7.45	trunk/libSolve/solve.h File Reference	126

7.45.1	Detailed Description	127
7.46	trunk/libSolve/solveresult.h File Reference	127
7.46.1	Detailed Description	127
7.47	trunk/libSolve/ueber9000.cpp File Reference	127
7.48	trunk/libSolve/ueber9000.h File Reference	128
7.49	trunk/test/AntennaConfiguration.cpp File Reference	129
7.49.1	Detailed Description	130
7.49.2	Define Documentation	130
7.49.2.1	_DROP_BAD_	130
7.49.2.2	_PP_FORM_GROUPS	130
7.49.2.3	_REFINE_SELECTION	130
7.49.2.4	_USE_IDEAL_INPUT	130
7.49.2.5	_USE_SHARK_3_0_	130
7.49.2.6	_Write_Result	130
7.49.2.7	_Write_SOLUTION_STATISTICS	130
7.49.2.8	_WT_CONSTRAIN_HARD_	130
7.49.2.9	USAGE_AND_EXIT	130
7.49.3	Function Documentation	131
7.49.3.1	main	131
7.49.4	Variable Documentation	131
7.49.4.1	DEFAULT_MAX_EVALUATIONS	131
7.49.4.2	DROPBAD	131
7.49.4.3	EVALUATIONS	131
7.49.4.4	EXPECTED	131
7.49.4.5	FILENAME	131
7.49.4.6	IDEAL_DATA	131
7.49.4.7	LAMBDA	131
7.49.4.8	MatOffset	131
7.49.4.9	MU	132
7.49.4.10	NO_OF_SOLUTIONS	132
7.49.4.11	PointOffset	132
7.49.4.12	SOLUTION_AMOUNT	132
7.49.4.13	UseNMats	132
7.49.4.14	VARIANT_SW	132

7.50 trunk/test/AntennaConfiguration.h File Reference	132
7.50.1 Define Documentation	132
7.50.1.1 VERSION_MAJOR	132
7.50.1.2 VERSION_MINOR	132
7.50.1.3 VERSION_SUB_MINOR	132

Chapter 1

Todo List

Member `PRPSEvolution::Solve::Process::findSolutionCMA_ES_Mkl ()`
document

Member `PRPSEvolution::Solve::Process::findSolutionCMA_ES_MkII ()`
document

Member `PRPSEvolution::Solve::Process::findSolutionSphere (Solve::ESStrategy strategy)`
document

Member `PRPSEvolution::Solve::Process_MkII::setMaxEvaluations (const int evaluations)`
remove typo

Member `PRPSEvolution::Solve::Ueber9000< T >::evaluate (const ChromosomeT< double > &)`
document

Member `PRPSEvolution::Solve::Ueber9000< T >::evaluateMkl (const - ChromosomeT< double > &)`
document

Member `PRPSEvolution::Solve::Ueber9000< T >::evaluateMkII (const - ChromosomeT< double > &, const ChromosomeT< double > &)`
document

Member `PRPSEvolution::Solve::Ueber9000< T >::evaluateMkIII (const - ChromosomeT< double > &, const ChromosomeT< int > &)`
document

Member `PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkl (const N- Rmatrix< T > &A, const ChromosomeT< double > &x, const NRvector< T > &b)`
documentation

Member `PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkII (const - ChromosomeT< double > &x, const ChromosomeT< int > &n)`
document

Member [**PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkII**](#) (const -
ChromosomeT< double > &x1, const ChromosomeT< double > &x2)
document

Member [**PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkII**](#) (const -
ChromosomeT< double > &x)
document

Member [**PRPSEvolution::WholeTomatoMkl_A::mkl**](#) (const NRmatrix< Doub > &-
A, const SearchPointType &x, const NRvector< Doub > &b) const
documentation

Member [**PRPSEvolution::WholeTomatoMkl_B::mkl**](#) (const NRmatrix< Doub > &-
A, const SearchPointType &x, const NRvector< Doub > &b) const
documentation

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

PRPSError	9
PRPSError::FileIO	9
PRPSEvolution	10
PRPSEvolution::Calibration	13
PRPSEvolution::Exceptions	13
PRPSEvolution::Exceptions::Calibration	13
PRPSEvolution::Exceptions::FileIO	13
PRPSEvolution::Exceptions::General	14
PRPSEvolution::Exceptions::Permutation	14
PRPSEvolution::Exceptions::Solve	14
PRPSEvolution::Models	14
PRPSEvolution::Permute	15
PRPSEvolution::Positioning	16
PRPSEvolution::Solve	16
PRPSEvolution::Support	18

Chapter 3

Class Index

3.1 Class List

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

PRPSEvolution::Permute::AntennaPermutations< N_MAT, T >	19
PRPSEvolution::Constants	20
PRPSEvolution::Positioning::CoordContainer< N, T >	21
PRPSEvolution::Models::EvolutionaryCalibration	23
PRPSEvolution::Exceptions::FileIO::FileNotFoundException	25
PRPSEvolution::Support::FitnessPlaneCalculator< N >	26
PRPSEvolution::Support::FitnessPlaneLimits< N >	28
PRPSEvolution::Exceptions::FileIO::MalformedInput	29
PRPSEvolution::Normalizer< N, T >	29
PRPSEvolution::Exceptions::General::NotImplemented	30
PRPSEvolution::Exceptions::FileIO::OutputFailure	31
PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >	31
PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >	34
PRPSEvolution::Solve::PostProcessing	38
PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >	38
PRPSEvolution::Solve::ProblemDimensions	41
PRPSEvolution::Solve::Process	42
PRPSEvolution::Solve::Process_MkII	48
PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >	58
PRPSEvolution::System	60
PRPSEvolution::Solve::Ueber9000< T >	61
PRPSEvolution::WholeTomatoMkI_A	70
PRPSEvolution::WholeTomatoMkI_B	72
PRPSEvolution::Models::WholeTomatoMkII	74
PRPSEvolution::Models::WholeTomatoMkII_B	77
PRPSEvolution::Models::WholeTomatoMkIII	80
PRPSEvolution::Models::WholeTomatoReduced	84

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

trunk/CMakeFiles/CompilerIdC/ CMakeCCompilerId.c	89
trunk/CMakeFiles/CompilerIdCXX/ CMakeCXXCompilerId.cpp	90
trunk/include/coords.h	91
trunk/include/prps.h	92
trunk/include/PRPSError.h	92
trunk/include/PRPSEvolution.h	93
trunk/include/PRPSEvolutionCalibrationExceptions.h	94
trunk/include/PRPSEvolutionFIOExceptions.h	95
trunk/include/PRPSEvolutionGeneralExceptions.h	96
trunk/include/PRPSEvolutionPermutationExceptions.h	97
trunk/include/PRPSEvolutionSolveExceptions.h	99
trunk/libCalibration/calib.cpp	100
trunk/libCalibration/calib.h	100
trunk/libFitnessPlanCalculator/ FitnessPlaneCalculator.h	101
trunk/libNormalizer/normalizer.cpp	102
trunk/libNormalizer/normalizer.h	103
trunk/libPermutate/permute.cpp	104
trunk/libPermutate/permute.h	104
trunk/libPRPSSystem/prpsevolutionsystem.cpp	106
trunk/libPRPSSystem/prpsevolutionsystem.h	106
trunk/libSolve/ObjectFunctions.cpp	106
trunk/libSolve/ObjectFunctions.h	106
trunk/libSolve/postprocessing.cpp	120
trunk/libSolve/postprocessing.h	121
trunk/libSolve/preprocessing.cpp	122
trunk/libSolve/preprocessing.h	122
trunk/libSolve/process.cpp	123
trunk/libSolve/process.h	123
trunk/libSolve/processMkII.cpp	123

trunk/libSolve/processMkII.h	123
trunk/libSolve/solve.cpp	125
trunk/libSolve/solve.h	126
trunk/libSolve/solveresult.h	127
trunk/libSolve/ueber9000.cpp	127
trunk/libSolve/ueber9000.h	128
trunk/libSolve/Objectivefunctions/EvolutionaryCalibration.cpp	107
trunk/libSolve/Objectivefunctions/EvolutionaryCalibration.h	108
trunk/libSolve/Objectivefunctions/WholeTomatoMkI.cpp	109
trunk/libSolve/Objectivefunctions/WholeTomatoMkI_A.h	109
trunk/libSolve/Objectivefunctions/WholeTomatoMkI_B.h	111
trunk/libSolve/Objectivefunctions/WholeTomatoMkII.cpp	112
trunk/libSolve/Objectivefunctions/WholeTomatoMkII.h	113
trunk/libSolve/Objectivefunctions/WholeTomatoMkII_B.cpp	114
trunk/libSolve/Objectivefunctions/WholeTomatoMkII_B.h	115
trunk/libSolve/Objectivefunctions/WholeTomatoMkIII.cpp	116
trunk/libSolve/Objectivefunctions/WholeTomatoMkIII.h	117
trunk/libSolve/Objectivefunctions/WholeTomatoReduced.cpp	118
trunk/libSolve/Objectivefunctions/WholeTomatoReduced.h	119
trunk/test/AntennaConfiguration.cpp	129
trunk/test/AntennaConfiguration.h	132

Chapter 5

Namespace Documentation

5.1 PRPSError Namespace Reference

Namespaces

- namespace [FileIO](#)

Variables

- const int [okay](#) = 0
- const int [general](#) = -1
- const int [critical](#) = 10

5.1.1 Variable Documentation

5.1.1.1 const int PRPSError::critical = 10

this is devastating

5.1.1.2 const int PRPSError::general = -1

if no other error fits

5.1.1.3 const int PRPSError::okay = 0

this ist no error

5.2 PRPSError::FileIO Namespace Reference

Variables

- const int [okay](#) = 0
- const int [generalError](#) = -1
- const int [fnf](#) = -2
- const int [inputmalformed](#) = -3

5.2.1 Variable Documentation

5.2.1.1 const int PRPSError::FileIO::fnf = -2

file not found error

5.2.1.2 const int PRPSError::FileIO::generalError = -1

if no other error fits

5.2.1.3 const int PRPSError::FileIO::inputmalformed = -3

malformed input

5.2.1.4 const int PRPSError::FileIO::okay = 0

this ist no error

5.3 PRPSEvolution Namespace Reference

Namespaces

- namespace [Calibration](#)
- namespace [Exceptions](#)
- namespace [Models](#)
- namespace [Permute](#)
- namespace [Positioning](#)
- namespace [Solve](#)
- namespace [Support](#)

Classes

- struct [Normalizer](#)
- struct [Constants](#)
- struct [System](#)
- struct [WholeTomatoMkl_A](#)
- struct [WholeTomatoMkl_B](#)

Enumerations

- enum [NormalizationMethods](#) { [Native](#), [Ideal](#), [CMPLX](#), [RND](#), [Complex](#) }

Functions

- std::vector< std::array < double, 3 > > [rPoints \(\)](#)
- std::vector< std::array < double, 3 > > [rCoords \(\)](#)
- template<std::size_t N, typename T>
std::array< T, N > [idealPhaseFromInput](#) (std::vector< std::array < double, 3 > > p, std::vector< std::array < double, 3 > > c, int SELECT_IDEAL_POINT, double [LAMBDA](#))
- [ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) ([WholeTomatoMkl_A](#), shark-
::soo::RealValuedObjectiveFunctionFactory)
- [ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) ([WholeTomatoMkl_B](#), shark-
::soo::RealValuedObjectiveFunctionFactory)

Variables

- const int [ANTENNA_AMOUNT](#) = 8
- const int [EXPECTED_LINES_CALIBRATION_FILE](#) = 4
- const int [EXPECTED_VALUES_CALIBRATION_FILE](#) = [ANTENNA_AMOUNT](#)
- const int [EXPECTED_LINES_COORD_FILE](#) = [ANTENNA_AMOUNT](#)
- const int [EXPECTED_VALUES_COORD_FILE](#) = 3
- const int [EXPECTED_LINES_SYSTEM_INI_FILE](#) = 2
- const int [MAT_ROWS](#) = 3
- const int [MAT_COLS](#) = 10
- const int [CALIBRATION_POINTS_AVAILABLE](#) = 4
- const int [EXPECTED_LINES_MEASUREMENT_FILE](#) = [ANTENNA_AMOUNT](#)
- const int [EXPECTED_VALUES_MEASUREMENT_FILE](#) = 2
- const int [DATA_NV](#) = 65535
- const int [DEFAULT_MIN_GROUP_SIZE](#) = 4
- const double [pi](#) = 3.14159

5.3.1 Detailed Description

This file contains structures and classes belonging to the system itself

5.3.2 Enumeration Type Documentation

5.3.2.1 enum PRPSEvolution::NormalizationMethods

Defines the methods available for normalization

Enumerator:

- Native***
- Ideal***
- CMPLX***
- RND***
- Complex***

5.3.3 Function Documentation

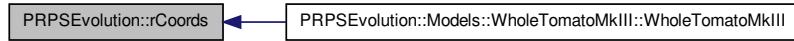
5.3.3.1 **PRPSEvolution::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION (WholeTomatoMkl_A , shark::soo::RealValuedObjectiveFunctionFactory)**

5.3.3.2 **PRPSEvolution::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION (WholeTomatoMkl_B , shark::soo::RealValuedObjectiveFunctionFactory)**

5.3.3.3 **template<std::size_t N, typename T > template std::array< double, 8 > PRPSEvolution::idealPhaseFromInput (std::vector< std::array< double, 3 >> p, std::vector< std::array< double, 3 >> c, int SELECT_IDEAL_POINT, double LAMBDA)**

5.3.3.4 **std::vector<std::array<double,3>> PRPSEvolution::rCoords ()**

Here is the caller graph for this function:



5.3.3.5 **std::vector<std::array<double,3>> PRPSEvolution::rPoints ()**

5.3.4 Variable Documentation

5.3.4.1 **const int PRPSEvolution::ANTENNA_AMOUNT = 8**

5.3.4.2 **const int PRPSEvolution::CALIBRATION_POINTS_AVAILABLE = 4**

5.3.4.3 **const int PRPSEvolution::DATA_NV = 65535**

5.3.4.4 **const int PRPSEvolution::DEFAULT_MIN_GROUP_SIZE = 4**

5.3.4.5 **const int PRPSEvolution::EXPECTED_LINES_CALIBRATION_FILE = 4**

- 5.3.4.6 const int PRPSEvolution::EXPECTED_LINES_COORD_FILE = ANTENNA_AMOUNT
- 5.3.4.7 const int PRPSEvolution::EXPECTED_LINES_MEASUREMENT_FILE = ANTENNA_AMOUNT
- 5.3.4.8 const int PRPSEvolution::EXPECTED_LINES_SYSTEM_INI_FILE = 2
- 5.3.4.9 const int PRPSEvolution::EXPECTED_VALUES_CALIBRATION_FILE = ANTENNA_AMOUNT
- 5.3.4.10 const int PRPSEvolution::EXPECTED_VALUES_COORD_FILE = 3
- 5.3.4.11 const int PRPSEvolution::EXPECTED_VALUES_MEASUREMENT_FILE = 2
- 5.3.4.12 const int PRPSEvolution::MAT_COLS = 10
- 5.3.4.13 const int PRPSEvolution::MAT_ROWS = 3
- 5.3.4.14 const double PRPSEvolution::pi = 3.14159

5.4 PRPSEvolution::Calibration Namespace Reference

Classes

- struct [performCalibration](#)

5.5 PRPSEvolution::Exceptions Namespace Reference

Namespaces

- namespace [Calibration](#)
- namespace [FileIO](#)
- namespace [General](#)
- namespace [Permutation](#)
- namespace [Solve](#)

5.6 PRPSEvolution::Exceptions::Calibration Namespace Reference

5.7 PRPSEvolution::Exceptions::FileIO Namespace Reference

Classes

- struct [FileNotFoundException](#)

- struct [MalformedInput](#)
- struct [OutputFailure](#)

5.8 PRPSEvolution::Exceptions::General Namespace Reference

Classes

- struct [NotImplemented](#)

5.9 PRPSEvolution::Exceptions::Permutation Namespace Reference

5.10 PRPSEvolution::Exceptions::Solve Namespace Reference

5.11 PRPSEvolution::Models Namespace Reference

Classes

- struct [EvolutionaryCalibration](#)
- struct [WholeTomatoMkII](#)
- struct [WholeTomatoMkII_B](#)
- struct [WholeTomatoMkIII](#)
- struct [WholeTomatoReduced](#)

Functions

- [ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) ([EvolutionaryCalibration](#), [soo::RealValuedObjectiveFunctionFactory](#))
- [ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) ([WholeTomatoMkII](#), [soo::RealValuedObjectiveFunctionFactory](#))
- [ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) ([WholeTomatoMkII_B](#), [soo::RealValuedObjectiveFunctionFactory](#))
- [ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) ([WholeTomatoMkIII](#), [soo::RealValuedObjectiveFunctionFactory](#))
- [ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) ([WholeTomatoReduced](#), [soo::RealValuedObjectiveFunctionFactory](#))

Variables

- bool [printYN](#) = false

5.11.1 Function Documentation

5.11.1.1 **PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION**
(EvolutionaryCalibration , soo::RealValuedObjectiveFunctionFactory)

5.11.1.2 **PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION**
(WholeTomatoMkII_B , soo::RealValuedObjectiveFunctionFactory)

5.11.1.3 **PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION**
(WholeTomatoMkIII , soo::RealValuedObjectiveFunctionFactory)

5.11.1.4 **PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION**
(WholeTomatoMkII , soo::RealValuedObjectiveFunctionFactory)

5.11.1.5 **PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION**
(WholeTomatoReduced , soo::RealValuedObjectiveFunctionFactory)

5.11.2 Variable Documentation

5.11.2.1 bool PRPSEvolution::Models::printYN = false

5.12 PRPSEvolution::Permute Namespace Reference

Classes

- struct [AntennaPermutations](#)
- struct [permuteAntennas](#)

Functions

- int [Factorial](#) (int x)
- template<typename Iterator >
bool [next_combination](#) (const Iterator first, Iterator k, const Iterator last)

Variables

- const int [MAX_PERMUTATION_AMOUNT](#) = 35

5.12.1 Function Documentation

5.12.1.1 int PRPSEvolution::Permute::Factorial (int x) [inline]

5.12.1.2 template<typename Iterator > bool PRPSEvolution::Permute::next_combination (const Iterator *first*, Iterator *k*, const Iterator *last*)
[inline]

5.12.2 Variable Documentation

5.12.2.1 const int PRPSEvolution::Permute::MAX_PERMUTATION_AMOUNT = 35

The maximum amount of Permutations for one reference antenna, we need this constant expression for the template

5.13 PRPSEvolution::Positioning Namespace Reference

Classes

- struct [CoordContainer](#)

5.14 PRPSEvolution::Solve Namespace Reference

Classes

- class [PostProcessing](#)
- class [PreProcessing](#)
- class [Process](#)
- class [Process_MkII](#)
- struct [ProblemDimensions](#)
- struct [solveresult_t](#)
- struct [Ueber9000](#)

Enumerations

- enum [SelectBy](#) { ConditionNumber, Random, AllPossible, Best10ByCN, AllFrom4Ant }
- enum [ESStrategy](#) { OnePlusOne, MuPlusLambda, MuCommaLambda, MuCommaLambda_MkII, MuPlusLambda_MkII, CMA_ES_MkI, CMA_ES_MkII }
- enum [Models](#) { WholeTomatoMkI, WholeTomatoMkII, TestSphere }

Functions

- double [meanFromVector](#) (std::vector< double > &res)

Variables

- const int [nConfigsForProcessing](#) = 1
- std::mutex [wMutex](#)
- int [_i](#) = 0

5.14.1 Enumeration Type Documentation

5.14.1.1 enum PRPSEvolution::Solve::ESStrategy

Represents the ES-strategy to find a solution

Enumerator:

OnePlusOne

$$[1 + 1] - ES$$

MuPlusLambda

$$[\mu + \lambda] - ES$$

MuCommaLambda

$$[\mu, \lambda] - ES$$

MuCommaLambda_MkII

MuPlusLambda_MkII

CMA_ES_MkI

CMA_ES_MkII

5.14.1.2 enum PRPSEvolution::Solve::Models

[Models](#) are defined here

Enumerator:

WholeTomatoMkI

WholeTomatoMkII

TestSphere

5.14.1.3 enum PRPSEvolution::Solve::SelectBy

Represents the selection method for the Matrix A that will be used for the solution

Enumerator:

ConditionNumber

Random

AllPossible

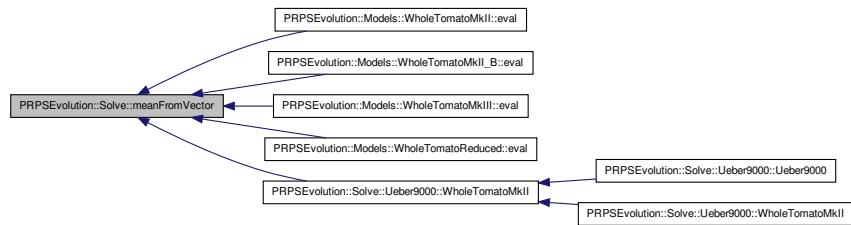
Best10ByCN

AllFrom4Ant

5.14.2 Function Documentation

5.14.2.1 `double PRPSEvolution::Solve::meanFromVector (std::vector< double > & res) [inline]`

Here is the caller graph for this function:



5.14.3 Variable Documentation

5.14.3.1 `int PRPSEvolution::Solve::_i_ = 0`

5.14.3.2 `const int PRPSEvolution::Solve::nConfigsForProcessing = 1`

5.14.3.3 `std::mutex PRPSEvolution::Solve::wMutex`

5.15 PRPSEvolution::Support Namespace Reference

Classes

- struct [FitnessPlaneLimits](#)
- class [FitnessPlaneCalculator](#)

Chapter 6

Class Documentation

6.1 PRPSEvolution::Permute::AntennaPermutations< N_MAT, T > Struct Template Reference

```
#include <permutate.h>
```

Public Member Functions

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

Static Public Member Functions

- [static void dump_matrix \(NRmatrix< T > mat\)](#)
- [static void dump_matrix_2_file \(std::ofstream &f, NRmatrix< T > mat\)](#)

Public Attributes

- [std::array< NRmatrix< T >, N_MAT > mat](#)
- [std::array< std::string, N_MAT > names](#)

```
template<std::size_t N_MAT, typename T> struct PRPSEvolution::Permute::Antenna-  
Permutations< N_MAT, T >
```

6.1.1 Constructor & Destructor Documentation

```
6.1.1.1 template<std::size_t N_MAT, typename T> PRPSEvolution::Permute::-  
AntennaPermutations< N_MAT, T >::AntennaPermutations ( void )  
[inline]
```

6.1.2 Member Function Documentation

6.1.2.1 template<std::size_t N_MAT, typename T> static void PRPSEvolution::Permute::AntennaPermutations< N_MAT, T >::dump_matrix (NRmatrix< T > mat)
[inline, static]

6.1.2.2 template<std::size_t N_MAT, typename T> static void PRPSEvolution::Permute::AntennaPermutations< N_MAT, T >::dump_matrix_2_file (std::ofstream & f, NRmatrix< T > mat) [inline, static]

6.1.3 Member Data Documentation

6.1.3.1 template<std::size_t N_MAT, typename T> std::array< NRmatrix< T >, N_MAT > PRPSEvolution::Permute::AntennaPermutations< N_MAT, T >::mat

6.1.3.2 template<std::size_t N_MAT, typename T> std::array< std::string, N_MAT > PRPSEvolution::Permute::AntennaPermutations< N_MAT, T >::names

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

- [trunk/libPermute/permute.h](#)

6.2 PRPSEvolution::Constants Struct Reference

```
#include <prpsevolutionsystem.h>
```

Public Member Functions

- [Constants \(\)](#)
- [Constants \(const PRPSEvolution::Constants &c\)](#)

Public Attributes

- double [a_1](#)
- double [a_2](#)
- double [lambda](#)
- double [f_mess](#)
- double [c_0](#)

6.2.1 Constructor & Destructor Documentation

6.2.1.1 [PRPSEvolution::Constants::Constants \(\)](#) [inline]

6.2.1.2 PRPSEvolution::Constants::Constants (const PRPSEvolution::Constants & c) [inline]

6.2.2 Member Data Documentation

6.2.2.1 double PRPSEvolution::Constants::a_1

6.2.2.2 double PRPSEvolution::Constants::a_2

6.2.2.3 double PRPSEvolution::Constants::c_0

6.2.2.4 double PRPSEvolution::Constants::f_mess

6.2.2.5 double PRPSEvolution::Constants::lambda

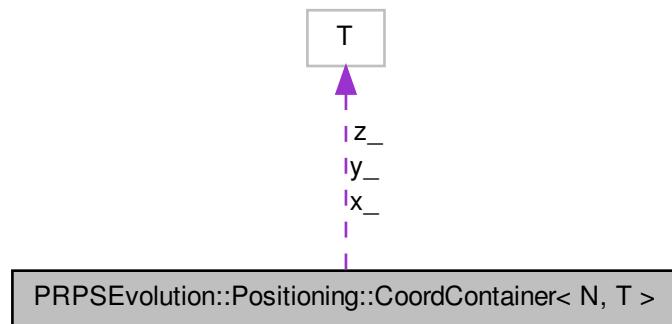
The documentation for this struct was generated from the following file:

- trunk/libPRPSSystem/prpsevolutionsystem.h

6.3 PRPSEvolution::Positioning::CoordContainer< N, T > Struct Template Reference

```
#include <coords.h>
```

Collaboration diagram for PRPSEvolution::Positioning::CoordContainer< N, T >:



Public Types

- `typedef T value_type`

Public Member Functions

- `CoordContainer ()`
- `template<typename T1 >`
`CoordContainer (T1 init)`
- `T & operator[] (std::size_t i)`

Public Attributes

- `T x_[N]`
- `T y_[N]`
- `T z_[N]`

```
template<std::size_t N, typename T> struct PRPSEvolution::Positioning::CoordContainer< N, T >
```

6.3.1 Member Typedef Documentation

6.3.1.1 `template<std::size_t N, typename T> typedef T PRPSEvolution::Positioning::-`
`CoordContainer< N, T >::value_type`

6.3.2 Constructor & Destructor Documentation

6.3.2.1 `template<std::size_t N, typename T > PRPSEvolution::-`
`Positioning::CoordContainer< N, T >::CoordContainer (`
`)`

6.3.2.2 `template<std::size_t N, typename T > template<typename T1 >`
`PRPSEvolution::Positioning::CoordContainer< N, T >::CoordContainer (`
`T1 init)`

6.3.3 Member Function Documentation

6.3.3.1 `template<std::size_t N, typename T > T & PRPSEvolution::-`
`Positioning::CoordContainer< N, T >::operator[] (std::size_t i)`

6.3.4 Member Data Documentation

6.3.4.1 `template<std::size_t N, typename T > T PRPSEvolution::Positioning::Coord-`
`Container< N, T >::x_[N]`

6.3.4.2 template<std::size_t N, typename T> T PRPSEvolution::Positioning::CoordContainer< N, T >::y_[N]

6.3.4.3 template<std::size_t N, typename T> T PRPSEvolution::Positioning::CoordContainer< N, T >::z_[N]

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

- trunk/include/coords.h

6.4 PRPSEvolution::Models::EvolutionaryCalibration Struct - Reference

```
#include <EvolutionaryCalibration.h>
```

Public Types

- typedef AbstractOptimizer < shark::VectorSpace< double > , double, SingleObjectiveResultSet < typename shark::VectorSpace < double >::PointType > > [base_type](#)
- typedef [base_type](#)::ObjectiveFunctionType [ObjectiveFunctionType](#)

Public Member Functions

- [EvolutionaryCalibration \(\)](#)
- std::string [name \(\) const](#)
From INameable: return the class name.
- std::size_t [numberOfVariables \(\) const](#)
- bool [hasScalableDimensionality \(\) const](#)
- void [setNumberOfVariables \(std::size_t numberOfVariables\)](#)
- void [configure \(const PropertyTree &node\)](#)
- void [proposeStartingPoint \(SearchPointType &x\) const](#)
- double [eval \(const SearchPointType &p\) const](#)
- void [setParams \(const NRmatrix< Doub > &M, const NRvector< Doub > &v\)](#)
- void [setMat \(const NRmatrix< Doub > &M\)](#)
- void [setVec \(const NRvector< Doub > &v\)](#)
- double [mkII \(const NRmatrix< Doub > &A, const double *x, const NRvector< Doub > &b\) const](#)

6.4.1 Member Typedef Documentation

6.4.1.1 typedef AbstractOptimizer<shark::VectorSpace< double >,double,SingleObjectiveResultSet<typename shark::VectorSpace< double >::PointType>> PRPSEvolution::Models::EvolutionaryCalibration::[base_type](#)

6.4.1.2 `typedef base_type::ObjectiveFunctionType PRPSEvolution::Models::EvolutionaryCalibration::ObjectiveFunctionType`

6.4.2 Constructor & Destructor Documentation

6.4.2.1 `PRPSEvolution::Models::EvolutionaryCalibration::EvolutionaryCalibration() [inline]`

6.4.3 Member Function Documentation

6.4.3.1 `void PRPSEvolution::Models::EvolutionaryCalibration::configure(const PropertyTree & node) [inline]`

6.4.3.2 `double PRPSEvolution::Models::EvolutionaryCalibration::eval(const SearchPointType & p) const`

6.4.3.3 `bool PRPSEvolution::Models::EvolutionaryCalibration::hasScalableDimensionality() const [inline]`

6.4.3.4 `double PRPSEvolution::Models::EvolutionaryCalibration::mkII(const NRmatrix< Doub > & A, const double * x, const NRvector< Doub > & b) const [inline]`

6.4.3.5 `std::string PRPSEvolution::Models::EvolutionaryCalibration::name() const [inline]`

From INameable: return the class name.

6.4.3.6 `std::size_t PRPSEvolution::Models::EvolutionaryCalibration::numberOfVariables() const [inline]`

6.4.3.7 `void PRPSEvolution::Models::EvolutionaryCalibration::proposeStartingPoint(SearchPointType & x) const [inline]`

Generate a starting value

Parameters

<code>out</code>	<code>x</code>	The suggested search point
------------------	----------------	----------------------------

6.4.3.8 `void PRPSEvolution::Models::EvolutionaryCalibration::setMat(const NRmatrix< Doub > & M) [inline]`

6.4.3.9 void PRPSEvolution::Models::EvolutionaryCalibration-
::setNumberOfVariables (std::size_t *numberOfVariables*)
[inline]

6.4.3.10 void PRPSEvolution::Models::EvolutionaryCalibration::setParams (const
NRmatrix< Doub > & *M*, const NRvector< Doub > & *v*) [inline]

Here is the caller graph for this function:



6.4.3.11 void PRPSEvolution::Models::EvolutionaryCalibration::setVec (const
NRvector< Doub > & *v*) [inline]

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

- trunk/libSolve/Objectivefunctions/[EvolutionaryCalibration.h](#)
- trunk/libSolve/Objectivefunctions/[EvolutionaryCalibration.cpp](#)

6.5 PRPSEvolution::Exceptions::FileIO::FileNotFoundException Struct Reference

```
#include <PRPSEvolutionFIOExceptions.h>
```

Public Member Functions

- const char * [**what \(\) const noexcept**](#)

6.5.1 Member Function Documentation

6.5.1.1 const char* PRPSEvolution::Exceptions::FileIO::FileNotFoundException::what ()
const [inline]

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

- trunk/include/[PRPSEvolutionFIOExceptions.h](#)

6.6 PRPSEvolution::Support::FitnessPlaneCalculator< N > Class Template Reference

```
#include <FitnessPlaneCalculator.h>
```

Public Member Functions

- [FitnessPlaneCalculator](#) (int offset)
- void [rPlaneLimitsFromFile](#) ()
- void [setLimits](#) ([FitnessPlaneLimits< N > Limits](#))
- void [calculate](#) (ObjectiveFunctionType const &model)

Public Attributes

- std::vector < [FitnessPlaneLimits< N >](#) > [limits](#)
- std::string [f_pathBase](#) = "output/fitness/data/plane"
- int [f_count](#) = 0

```
template<std::size_t N> class PRPSEvolution::Support::FitnessPlaneCalculator< N >
```

6.6.1 Constructor & Destructor Documentation

6.6.1.1 template<std::size_t N> PRPSEvolution::Support::FitnessPlaneCalculator< N >::FitnessPlaneCalculator(int offset) [inline]

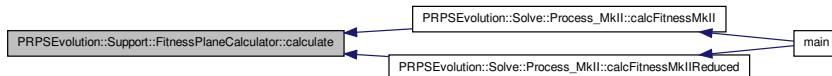
Here is the call graph for this function:



6.6.2 Member Function Documentation

6.6.2.1 template<std::size_t N> void PRPSEvolution::Support::FitnessPlaneCalculator< N >::calculate (ObjectiveFunctionType const & *model*) [inline]

Here is the caller graph for this function:



6.6.2.2 template<std::size_t N> void PRPSEvolution::Support::FitnessPlaneCalculator< N >::rPlaneLimitsFromFile () [inline]

Here is the caller graph for this function:



6.6.2.3 template<std::size_t N> void PRPSEvolution::Support::FitnessPlaneCalculator< N >::setLimits (FitnessPlaneLimits< N > *Limits*) [inline]

6.6.3 Member Data Documentation

6.6.3.1 template<std::size_t N> int PRPSEvolution::Support::FitnessPlaneCalculator< N >::f_count = 0

6.6.3.2 template<std::size_t N> std::string PRPSEvolution::Support::FitnessPlaneCalculator< N >::f_pathBase = "output/fitness/data/plane"

6.6.3.3 template<std::size_t N> std::vector<FitnessPlaneLimits< N >> PRPSEvolution::Support::FitnessPlaneCalculator< N >::limits

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

- trunk/libFitnessPlanCalculator/[FitnessPlaneCalculator.h](#)

6.7 PRPSEvolution::Support::FitnessPlaneLimits< N > Struct - Template Reference

```
#include <FitnessPlaneCalculator.h>
```

Public Member Functions

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

Public Attributes

- std::array< double, N > [upper](#)
- std::array< double, N > [lower](#)
- double [increment](#)
- int [idx1](#)
- int [idx2](#)

```
template<std::size_t N> struct PRPSEvolution::Support::FitnessPlaneLimits< N >
```

6.7.1 Member Function Documentation

6.7.1.1 template<std::size_t N> void PRPSEvolution::Support::FitnessPlaneLimits< N >::[dump \(\)](#) [inline]

6.7.2 Member Data Documentation

6.7.2.1 template<std::size_t N> int PRPSEvolution::Support::FitnessPlaneLimits< N >::[idx1](#)

6.7.2.2 template<std::size_t N> int PRPSEvolution::Support::FitnessPlaneLimits< N >::[idx2](#)

6.7.2.3 template<std::size_t N> double PRPSEvolution::Support::FitnessPlane- Limits< N >::[increment](#)

6.7.2.4 template<std::size_t N> std::array<double,N> PRPSEvolution::Support::- FitnessPlaneLimits< N >::[lower](#)

6.7.2.5 template<std::size_t N> std::array<double,N> PRPSEvolution::Support::- FitnessPlaneLimits< N >::[upper](#)

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

- trunk/libFitnessPlanCalculator/[FitnessPlaneCalculator.h](#)

6.8 PRPSEvolution::Exceptions::FileIO::MalformedInput Struct Reference

```
#include <PRPSEvolutionFIOExceptions.h>
```

Public Member Functions

- const char * [what \(\) const noexcept](#)

6.8.1 Member Function Documentation

6.8.1.1 const char* PRPSEvolution::Exceptions::FileIO::MalformedInput::what () const [inline]

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

- trunk/include/[PRPSEvolutionFIOExceptions.h](#)

6.9 PRPSEvolution::Normalizer< N, T > Struct Template Reference

```
#include <normalizer.h>
```

Public Member Functions

- [Normalizer \(NormalizationMethods method\)](#)
- std::array< T, N > [normalize \(std::array< T, N > phase, std::array< T, N > amp\)](#)
- void [setSelectIdealPoint \(int i\)](#)
- void [setLambda \(double l\)](#)

6.9.1 Detailed Description

template<std::size_t N, typename T>struct PRPSEvolution::Normalizer< N, T >

This class can perform the normalization of the input data

6.9.2 Constructor & Destructor Documentation

6.9.2.1 template<std::size_t N, typename T> PRPSEvolution::Normalizer< N, T >::Normalizer (NormalizationMethods *method*) [inline]

Constructor

Parameters

in	method	Selects the Normalization function
----	--------	------------------------------------

6.9.3 Member Function Documentation

6.9.3.1 template<std::size_t N, typename T > std::array< T, N > PRPSEvolution::Normalizer< N, T >::normalize (std::array< T, N > phase, std::array< T, N > amp)

Calculates the normalizations

Parameters

in	phase	The measured phase data
in	amp	The measured amplitude data

Returns

Array containing the normalized values.

6.9.3.2 template<std::size_t N, typename T > void PRPSEvolution::Normalizer< N, T >::setLambda (double l)

6.9.3.3 template<std::size_t N, typename T > void PRPSEvolution::Normalizer< N, T >::setSelectIdealPoint (int i)

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

- trunk/libNormalizer/normalizer.h
- trunk/libNormalizer/normalizer.cpp

6.10 PRPSEvolution::Exceptions::General::NotImplemented Struct Reference

```
#include <PRPSEvolutionGeneralExceptions.h>
```

Public Member Functions

- const char * [what \(\)](#) const noexcept

6.10.1 Detailed Description

Throw this if a Method is not implemented

6.10.2 Member Function Documentation

6.10.2.1 `const char* PRPSEvolution::Exceptions::General::NotImplemented::what() const [inline]`

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

- trunk/include/PRPSEvolutionGeneralExceptions.h

6.11 PRPSEvolution::Exceptions::FileIO::OutputFailure Struct Reference -

Reference

```
#include <PRPSEvolutionFIOExceptions.h>
```

Public Member Functions

- `const char * what() const noexcept`

6.11.1 Member Function Documentation

6.11.1.1 `const char* PRPSEvolution::Exceptions::FileIO::OutputFailure::what() const [inline]`

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

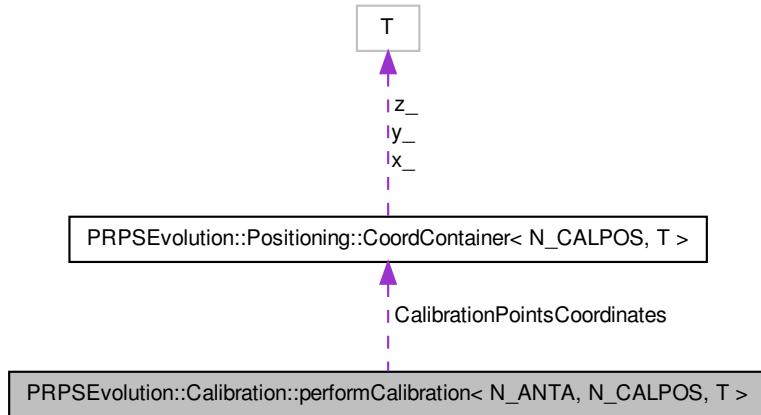
- trunk/include/PRPSEvolutionFIOExceptions.h

6.12 PRPSEvolution::Calibration::performCalibration< N_ANTA, - N_CALPOS, T > Struct Template Reference

```
#include <calib.h>
```

Collaboration diagram for PRPSEvolution::Calibration::performCalibration< N_ANTA, -

`N_CALPOS, T >:`



Public Member Functions

- [performCalibration \(\)](#)

Public Attributes

- `std::array< NRvector< T >, N_ANTA >` [AntennaCoordinates](#)
- `Positioning::CoordContainer < N_CALPOS, T >` [CalibrationPointsCoordinates](#)
- `std::array< std::array< T, N_ANTA >, N_CALPOS >` [distances](#)
- `NRmatrix< T >` [A](#)
- `std::array< NRvector< T >, N_ANTA >` [c_k0](#)

6.12.1 Detailed Description

```
template<std::size_t N_ANTA, std::size_t N_CALPOS, typename T>struct PRPSEvolution::-
Calibration::performCalibration< N_ANTA, N_CALPOS, T >
```

This will perform the calibration stuff

6.12.2 Constructor & Destructor Documentation

6.12.2.1 template<std::size_t N_ANTA, std::size_t N_CALPOS, typename T >
PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >::performCalibration()

6.12.3 Member Data Documentation

6.12.3.1 template<std::size_t N_ANTA, std::size_t N_CALPOS, typename T> NRmatrix< T >
PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >::A

The matrix A represents the geometrical relation of the calibration points

6.12.3.2 template<std::size_t N_ANTA, std::size_t N_CALPOS, typename T> std::array< N-
Rvector< T >, N_ANTA > PRPSEvolution::Calibration::performCalibration<
N_ANTA, N_CALPOS, T >::AntennaCoordinates

This is the array where the result is stored

6.12.3.3 template<std::size_t N_ANTA, std::size_t N_CALPOS, typename T> std::array< N-
Rvector< T >, N_ANTA > PRPSEvolution::Calibration::performCalibration<
N_ANTA, N_CALPOS, T >::c_k0

There are N_ANTA-vectors that are calculated using:

$$c_{k0} = 1/2[r_0^2 - r_k^2 + d_{k0}^2]$$

The Amount of entries depends on the number of Points used for calibration in general
the number of entries id equal to N_CALPOS -1

6.12.3.4 template<std::size_t N_ANTA, std::size_t N_CALPOS, typename
T > Positioning::CoordContainer< N_CALPOS, T >
PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >::CalibrationPointsCoordinates

6.12.3.5 template<std::size_t N_ANTA, std::size_t N_CALPOS, typename
T > std::array< std::array< T , N_ANTA >, N_CALPOS >
PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >::distances

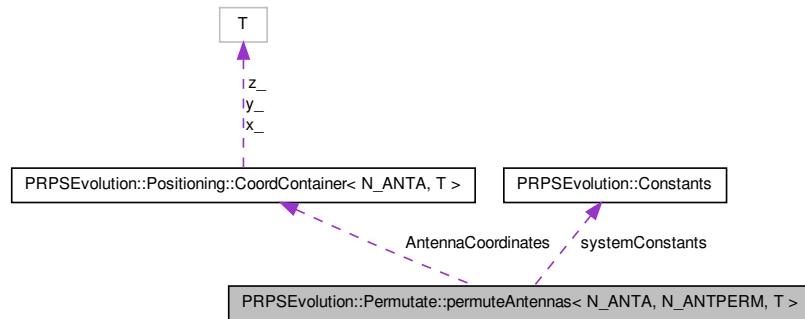
The documentation for this struct was generated from the following file:

- trunk/libCalibration/calib.h

6.13 PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T > Struct Template Reference

```
#include <permute.h>
```

Collaboration diagram for PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >:



Public Member Functions

- `permuteAntennas (const PRPSEvolution::Constants c)`
- `int rCoordFile ()`
- `int computePermutations (const PRPSEvolution::Constants &co)`
- `template<std::size_t NN, std::size_t MM> const NRmatrix< T > computeMatrix (const int ref, const int a1, const int a2, const int a3, const PRPSEvolution::Constants &co)`
- `NRmatrix< T > compute_d_k0_Mat ()`
- `void dumpConfigurationsToFile ()`
- `void dump_matrices_2_file ()`

Public Attributes

- `int ref`
- `PRPSEvolution::Constants systemConstants`
- `Positioning::CoordContainer < N_ANTA, T > AntennaCoordinates`
- `std::array < AntennaPermutations < N_ANTPERM, Doub >, N_ANTA > configurations`
- `NRmatrix< T > d_k0_mat`

6.13.1 Detailed Description

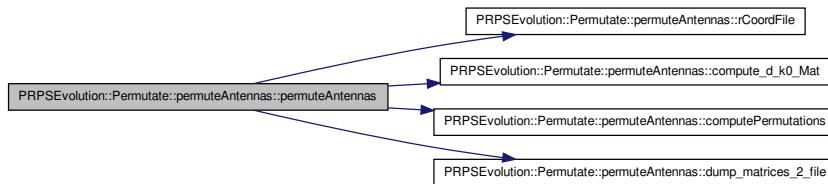
```
template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T>struct PRPSEvolution::  
Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >
```

This will collect some stuff for calculating the permutation of the antennas

6.13.2 Constructor & Destructor Documentation

```
6.13.2.1 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T >  
PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T  
>::permuteAntennas ( const PRPSEvolution::Constants c )
```

Here is the call graph for this function:



6.13.3 Member Function Documentation

```
6.13.3.1 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > NRmatrix< T >  
PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T  
>::compute_d_k0_Mat ( )
```

Here is the caller graph for this function:



```
6.13.3.2 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T
> template<std::size_t NN, std::size_t MM> const NRmatrix< T >
PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T
>::computeMatrix ( const int ref, const int a1, const int a2, const int a3, const
PRPSEvolution::Constants & co )
```

This method will compute all the possible permutations based on the given reference antenna

See also

[ref](#)

Parameters

in	ref	The reference antenna
in	a1	First antenna
in	a2	Second antenna
in	a3	Third antenna

```
6.13.3.3 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > int
PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T
>::computePermutations ( const PRPSEvolution::Constants & co )
```

This method handles the computation of the antenna permutations

Parameters

in	co	Constant structure with the system constants we need
----	----	--

See also

[PRPSEvolution::Constants](#)

Here is the caller graph for this function:



```
6.13.3.4 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > void
PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T
>::dump_matrices_2_file( )
```

This method will dump all the Antennas to an output file

Here is the caller graph for this function:



6.13.3.5 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T> void PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >::dumpConfigurationsToFile()

6.13.3.6 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > int PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >::rCoordFile()

Load the csv-file containing the coordinates and store it into the container.

Here is the caller graph for this function:



6.13.4 Member Data Documentation

6.13.4.1 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T> Positioning::CoordContainer< N_ANTA, T > PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >::AntennaCoordinates

6.13.4.2 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T> std::array< AntennaPermutations< N_ANTPERM, Doub >, N_ANTA > PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >::configurations

6.13.4.3 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T> NRmatrix<T> PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >::d_k0_mat

6.13.4.4 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T> int PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >::ref

6.13.4.5 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T> PRPS-Evolution::Constants PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >::systemConstants

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

- trunk/libPermute/permute.h

6.14 PRPSEvolution::Solve::PostProcessing Class Reference

```
#include <postprocessing.h>
```

Public Member Functions

- [PostProcessing \(\)](#)

6.14.1 Constructor & Destructor Documentation

6.14.1.1 PRPSEvolution::Solve::PostProcessing::PostProcessing ()
[inline]

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

- trunk/libSolve/postprocessing.h

6.15 PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure > Class Template Reference

```
#include <preprocessing.h>
```

Public Member Functions

- [PreProcessing \(const std::array< AntennaPermutations< Permute::MAX_PERMUTATION_AMOUNT, Doub >, N_ANTA > &, const NRmatrix< T > &, const PRPSEvolution::NormalizationMethods method, const int, const int, const double, const int\)](#)
- [int possibleGroupSize \(\)](#)

Public Attributes

- [std::vector< NRmatrix< T > > matrices](#)
- [std::vector< NRvector< T > > vectors](#)

- std::vector< std::string > **names**
- int **antennas**
- std::vector< std::vector < NRmatrix< T > > > **matGroups**
- std::vector< std::vector < NRvector< T > > > **vectorGroups**
- std::vector< std::vector < std::string > > **nameGroups**
- int **antennasPerGroup**

```
template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_Measure> class P-
RPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >
```

6.15.1 Constructor & Destructor Documentation

```
6.15.1.1 template<std::size_t N_ANTA, std::size_t N_Configs, typename T , typename
T_Measure > PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T,
T_Measure >::PreProcessing ( const std::array< AntennaPermutations<
Permute::MAX_PERMUTATION_AMOUNT, Doub >, N_ANTA
> & precalculatedMatrices, const NRmatrix< T > & d_k0s, const
PRPSEvolution::NormalizationMethods method, const int finalAntAmount,
const int offset, const double lambda, const int point )
```

Construct the object and perform necessary [PreProcessing](#) steps.

1. Read out the measurements from the given interface (e.g. a file)
2. Normalize everything
3. Select the matrices for further processing
4. Fill the matrices with the information
5. Precalculate the

c_{k0}

-Vector

6. Store matrices to make them available in the next steps

Parameters

in	<i>precalculatedMatrices</i>	Array containing the precalculated matrices from prior processing steps. This Array contains the static array for all possible permutations of the Antennas
in	<i>d_k0s</i>	This Array contains the d_{k0} , which denotes the euclidean distances between the Antennas
in	<i>finalAntAmount</i>	This field determines the Amount of Matrices we want to use for a calculation
in	<i>offset</i>	
in	<i>lambda</i>	
in	<i>point</i>	

6.15.2 Member Function Documentation

6.15.2.1 template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_Measure> int PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >::possibleGroupSize() [inline]

Returns the possible group size for one antenna. The group size depends on the amount of antennas that provided a decent measurement and can be calculated by:

$$\frac{n!}{k!(n-k)!}$$

Where n is the amount of available-1 and k is the size of the group

6.15.3 Member Data Documentation

6.15.3.1 template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_Measure> int PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >::antennas

Amount of antennas for the solution

6.15.3.2 template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_Measure> int PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >::antennasPerGroup

Amount of antennas for the solution

6.15.3.3 template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_Measure> std::vector<std::vector<NRmatrix<T>>> PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >::matGroups

Will contain the groups of a matrix

6.15.3.4 template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_Measure> std::vector<NRmatrix<T>> PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >::matrices

The precalculated matrices for a solution

```
6.15.3.5 template<std::size_t N_ANTA, std::size_t N_Configs, typename T,
typename T_Measure> std::vector<std::vector<std::string>>
PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure
>::nameGroups
```

Contains the names of a group

```
6.15.3.6 template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_-
Measure> std::vector<std::string> PRPSEvolution::Solve::PreProcessing<
N_ANTA, N_Configs, T, T_Measure >::names
```

The "Names" of the matrices for a solution

```
6.15.3.7 template<std::size_t N_ANTA, std::size_t N_Configs, typename T,
typename T_Measure> std::vector<std::vector<NRvector<T>>>
PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure
>::vectorGroups
```

Contains a group of the b-vectors

```
6.15.3.8 template<std::size_t N_ANTA, std::size_t N_Configs, typename T,
typename T_Measure> std::vector<NRvector<T>>
PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure
>::vectors
```

The b-vectors for the solution

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

- [trunk/libSolve/preprocessing.h](#)

6.16 PRPSEvolution::Solve::ProblemDimensions Struct Reference

```
#include <solve.h>
```

Static Public Attributes

- static const int [WholeTomato](#) = 7
- static const int [WholeTomatoMkl](#) = 10
- static const int [WholeTomatoMkl_A](#) = 10
- static const int [WholeTomatoMkl_B](#) = 7
- static const int [WholeTomatoMkII](#) = 3
- static const int [Sphere](#) = 10
- static const int [Rosenbrock](#) = 15
- static const int [Calibration](#) = 3
- static const int [WholeTomatoReduced](#) = 3

6.16.1 Detailed Description

This gathers the problem dimensions of the defined fitness functions

6.16.2 Member Data Documentation

- 6.16.2.1 `const int PRPSEvolution::Solve::ProblemDimensions::Calibration = 3 [static]`
- 6.16.2.2 `const int PRPSEvolution::Solve::ProblemDimensions::Rosenbrock = 15 [static]`
- 6.16.2.3 `const int PRPSEvolution::Solve::ProblemDimensions::Sphere = 10 [static]`
- 6.16.2.4 `const int PRPSEvolution::Solve::ProblemDimensions::WholeTomato = 7 [static]`
- 6.16.2.5 `const int PRPSEvolution::Solve::ProblemDimensions::WholeTomatoMkl = 10 [static]`
- 6.16.2.6 `const int PRPSEvolution::Solve::ProblemDimensions::WholeTomatoMkl_- A = 10 [static]`
- 6.16.2.7 `const int PRPSEvolution::Solve::ProblemDimensions::WholeTomatoMkl_- B = 7 [static]`
- 6.16.2.8 `const int PRPSEvolution::Solve::ProblemDimensions::WholeTomatoMklII = 3 [static]`

The minimal dimension for this problem, depending on the amount of antennas used this number will increase

- 6.16.2.9 `const int PRPSEvolution::Solve::ProblemDimensions::WholeTomato- Reduced = 3 [static]`

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

- trunk/libSolve/[solve.h](#)

6.17 PRPSEvolution::Solve::Process Class Reference

```
#include <process.h>
```

Public Member Functions

- `Process ()`
- `Process (const Process &p)`
- `double getLastSolutionFitness ()`
- `template<typename T >
T findSolutionSphere (Solve::ESStrategy strategy)`
- `template<typename T >
T findSolutionCMA_ES_Mkl ()`
- `template<typename T >
T findSolutionCMA_ES_MkII ()`
- `template<typename T >
T findSolutionSolveSingle (const NRmatrix< Doub > &A_selected, const NRvector< Doub > &b_selected, const std::vector< std::string > &names_selected, const int ants, const PRPSEvolution::Solve::ESStrategy strategy, const int seed)`
- `template<typename T >
T findSolution (const std::vector< NRmatrix< Doub >> &A_selected, const std::vector< NRvector< Doub >> &b_selected, const std::vector< std::string > &names_selected, const int ants, const PRPSEvolution::Solve::ESStrategy strategy, const int seed)`
- `int sq (int i)`
- `void setMinSolutionFitness (double value)`
- `void setSeed (unsigned int value)`
- `void incrementFileCounter ()`
- `void resetFileCounter ()`

Public Attributes

- `int f_count = 0`

6.17.1 Detailed Description

Find solutions for the possible matrices

6.17.2 Constructor & Destructor Documentation

6.17.2.1 PRPSEvolution::Solve::Process::Process() [inline]

Constructor

6.17.2.2 PRPSEvolution::Solve::Process::Process(const Process & p) [inline]

6.17.3 Member Function Documentation

```
6.17.3.1 template<typename T > T PRPSEvolution::Solve::Process::findSolution
( const std::vector< NRmatrix< Doub >> & A_selected, const std::vector<
NRvector< Doub >> & b_selected, const std::vector< std::string > &
names_selected, const int ants, const PRPSEvolution::Solve::ESStrategy
strategy, const int seed ) [inline]
```

Find a Solution for a given pair of matrices

Parameters

in	<i>A_selected</i>	The matrix A to use in this solution
in	<i>b_selected</i>	The c_{k0}' vector for this solution

Returns

The solution

Here is the caller graph for this function:



```
6.17.3.2 template<typename T > T PRPSEvolution::Solve::Process::findSolutionCM-
A_ES_Mkl( ) [inline]
```

Todo document

Returns

The solution

```
6.17.3.3 template<typename T > T PRPSEvolution::Solve::Process::findSolutionCM-
A_ES_MkII( ) [inline]
```

Todo document

Returns

The solution

Here is the caller graph for this function:



6.17.3.4 template<typename T > T PRPSEvolution::Solve::Process::findSolutionSolveSingle (const NRmatrix< Doub > & A_selected, const NRvector< Doub > & b_selected, const std::vector< std::string > & names_selected, const int ants, const PRPSEvolution::Solve::ESStrategy strategy, const int seed) [inline]

Find a Solution for a given pair of matrices

Parameters

in	A_selected	The matrix A to use in this solution
in	b_selected	The c_k0' vector for this solution

Returns

The solution

6.17.3.5 template<typename T > T PRPSEvolution::Solve::Process::findSolutionSphere (Solve::ESStrategy strategy) [inline]

Set the ES-Strategy

Parameters

in	Strategy	The selected strategy
----	----------	-----------------------

Todo document

Returns

The solution

Here is the caller graph for this function:



6.17.3.6 double PRPSEvolution::Solve::Process::getLastSolutionFitness() [inline]

6.17.3.7 void PRPSEvolution::Solve::Process::incrementFileCounter() [inline]

Here is the caller graph for this function:



6.17.3.8 void PRPSEvolution::Solve::Process::resetFileCounter() [inline]

Here is the caller graph for this function:



6.17.3.9 void PRPSEvolution::Solve::Process::setMinSolutionFitness (double *value*) [inline]

Sets the min. solution fitness we want to achieve.

Parameters

in	<i>value</i>	The new value for the solution fitness
----	--------------	--

Here is the caller graph for this function:



6.17.3.10 void PRPSEvolution::Solve::Process::setSeed (unsigned int *value*) [inline]

Here is the caller graph for this function:



6.17.3.11 int PRPSEvolution::Solve::Process::sq (int *i*) [inline]

6.17.4 Member Data Documentation

6.17.4.1 int PRPSEvolution::Solve::Process::f_count = 0

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

- trunk/libSolve/process.h

6.18 PRPSEvolution::Solve::Process_MkII Class Reference

```
#include <processMkII.h>
```

Public Member Functions

- [Process_MkII \(\)](#)
- [Process_MkII \(NRmatrix< Doub > Mat, NRvector< Doub > Vect, std::string - Name\)](#)
- [Process_MkII \(NRmatrix< Doub > Mat, NRvector< Doub > Vect, std::string - Name, const int mu, const int lambda\)](#)
- [Process_MkII \(std::vector< NRmatrix< Doub >> Mats, std::vector< NRvector< Doub >> Vests, std::vector< std::string > Names\)](#)
- [Process_MkII \(std::vector< NRmatrix< Doub >> Mats, std::vector< NRvector< Doub >> Vests, std::vector< std::string > Names, const int mu, const int lambda\)](#)
- [Process_MkII \(std::vector< NRmatrix< Doub >> Mats, std::vector< NRvector< Doub >> Vests, std::vector< std::string > Names, std::vector< std::vector< int >> IDs, double Epsilon\)](#)
- void [init \(\)](#)
- int [WholeTomatoMkII \(int dimension\)](#)
- int [WholeTomatoReduced \(double lambda\)](#)
- int [cWholeTomatoMkII \(int dimension, int n\)](#)
- int [WholeTomatoMkII_B \(int dimension\)](#)
- int [WholeTomatoMkI_A \(\)](#)
- int [WholeTomatoMkI_B \(\)](#)
- int [Process_MkII_test \(\)](#)
- int [EvolutionaryCalibration \(\)](#)
- void [calcFitnessMkII \(int offset\)](#)
- void [calcFitnessMkIIReduced \(int offset, const double _lambda\)](#)
- void [setEpsilon \(double Value\)](#)
- void [setOutputFilePath \(std::string file\)](#)
- void [setOutputFilePathBase \(std::string file\)](#)
- void [setPrintLastOnly \(void\)](#)
- void [incrementFileCounter \(void\)](#)
- void [setFileCounter \(int count\)](#)
- void [resetFileCounter \(\)](#)
- void [toggleVariant \(\)](#)
- void [setMaxEvaluations \(const int evaluations\)](#)
- void [setAntennaCoords \(std::array< NRvector< Doub >, 8 > coords\)](#)

6.18.1 Constructor & Destructor Documentation

6.18.1.1 PRPSEvolution::Solve::Process_MkII::Process_MkII() [inline]

Set up basic stuff...

Here is the call graph for this function:



6.18.1.2 PRPSEvolution::Solve::Process_MkII::Process_MkII(NRmatrix< Doub > Mat, NRvector< Doub > Vect, std::string Name) [inline]

Here is the call graph for this function:



6.18.1.3 PRPSEvolution::Solve::Process_MkII::Process_MkII(NRmatrix< Doub > Mat, NRvector< Doub > Vect, std::string Name, const int mu, const int lambda) [inline]

Here is the call graph for this function:



6.18.1.4 **PRPSEvolution::Solve::Process_MkII::Process_MkII (std::vector< NRmatrix< Doub >> Mats, std::vector< NRvector< Doub >> Vests, std::vector< std::string > Names) [inline]**

Here is the call graph for this function:



6.18.1.5 **PRPSEvolution::Solve::Process_MkII::Process_MkII (std::vector< NRmatrix< Doub >> Mats, std::vector< NRvector< Doub >> Vests, std::vector< std::string > Names, const int mu, const int lambda) [inline]**

Here is the call graph for this function:



6.18.1.6 **PRPSEvolution::Solve::Process_MkII::Process_MkII (std::vector< NRmatrix< Doub >> Mats, std::vector< NRvector< Doub >> Vests, std::vector< std::string > Names, std::vector< std::vector< int >> IDs, double Epsilon) [inline]**

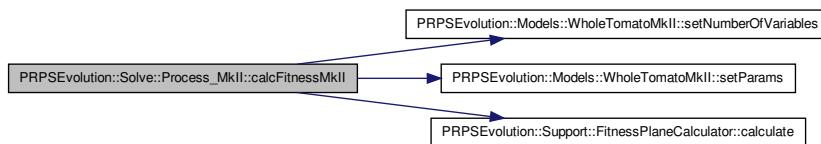
Here is the call graph for this function:



6.18.2 Member Function Documentation

6.18.2.1 void PRPSEvolution::Solve::Process_MkII::calcFitnessMkII (int offset)
[inline]

Here is the call graph for this function:

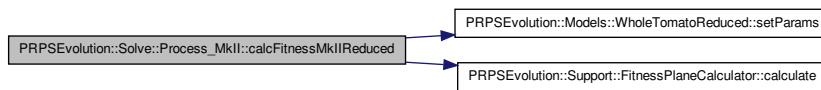


Here is the caller graph for this function:



6.18.2.2 void PRPSEvolution::Solve::Process_MkII::calcFitnessMkIIReduced (int offset, const double _lambda) [inline]

Here is the call graph for this function:



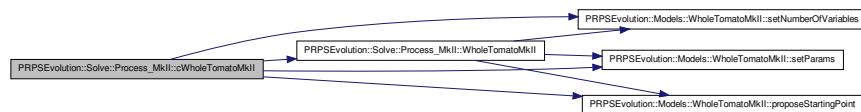
Here is the caller graph for this function:



6.18.2.3 int PRPSEvolution::Solve::Process_MkII::cWholeTomatoMkII (int dimension, int n) [inline]

Concurrent variant of the WholeTomatoMkII approach

Here is the call graph for this function:



Here is the caller graph for this function:



6.18.2.4 int PRPSEvolution::Solve::Process_MkII::EvolutionaryCalibration () [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



6.18.2.5 void PRPSEvolution::Solve::Process_MkII::incrementFileCounter (void) [inline]

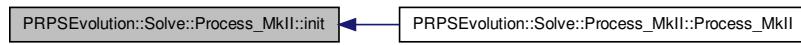
increment the File counter

Here is the caller graph for this function:



6.18.2.6 void PRPSEvolution::Solve::Process_MkII::init () [inline]

Here is the caller graph for this function:



6.18.2.7 int PRPSEvolution::Solve::Process_MkII::Process_MkII_test()
[inline]

Here is the caller graph for this function:



6.18.2.8 void PRPSEvolution::Solve::Process_MkII::resetFileCounter()
[inline]

6.18.2.9 void PRPSEvolution::Solve::Process_MkII::setAntennaCoords(std::array< NRvector< Doub >, 8 > coords) [inline]

Here is the caller graph for this function:



6.18.2.10 void PRPSEvolution::Solve::Process_MkII::setEpsilon(double Value)
[inline]

6.18.2.11 void PRPSEvolution::Solve::Process_MkII::setFileCounter(int count)
[inline]

increment the File counter

6.18.2.12 void PRPSEvolution::Solve::Process_MkII::setMaxEvaluations(const int evaluations) [inline]

Parameters

in	evaluations	The new value for the evaluations
----	-------------	-----------------------------------

Todo remove typo

Here is the caller graph for this function:



6.18.2.13 void PRPSEvolution::Solve::Process_MkII::setOutputFilePath (std::string *file*) [inline]

6.18.2.14 void PRPSEvolution::Solve::Process_MkII::setOutputFilePathBase (std::string *file*) [inline]

Here is the caller graph for this function:



6.18.2.15 void PRPSEvolution::Solve::Process_MkII::setPrintLastOnly (void) [inline]

6.18.2.16 void PRPSEvolution::Solve::Process_MkII::toggleVariant () [inline]

6.18.2.17 int PRPSEvolution::Solve::Process_MkII::WholeTomatoMkl_A()
[inline]

Here is the caller graph for this function:



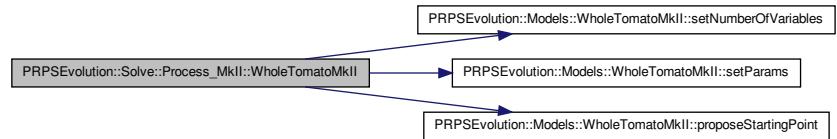
6.18.2.18 int PRPSEvolution::Solve::Process_MkII::WholeTomatoMkl_B()
[inline]

Here is the caller graph for this function:

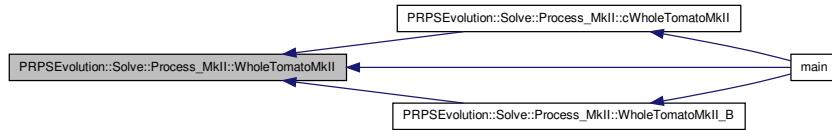


6.18.2.19 int PRPSEvolution::Solve::Process_MkII::WholeTomatoMkII(int
dimension) [inline]

Here is the call graph for this function:

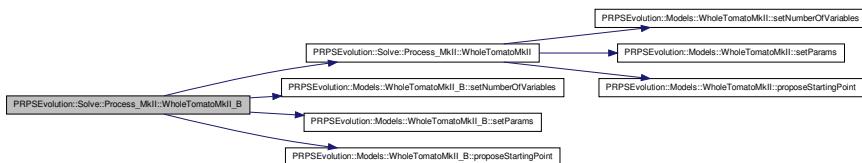


Here is the caller graph for this function:



6.18.2.20 int PRPSEvolution::Solve::Process_MkII::WholeTomatoMkII_B (int dimension) [inline]

Here is the call graph for this function:

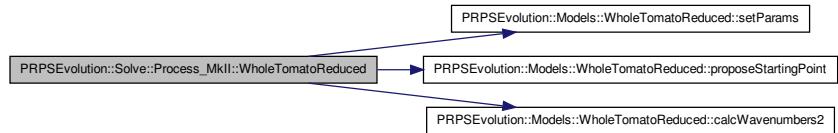


Here is the caller graph for this function:



6.18.2.21 int PRPSEvolution::Solve::Process_MkII::WholeTomatoReduced (double lambda) [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



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

- trunk/libSolve/processMkII.h

6.19 PRPSEvolution::Solve::solveresult_t < T_Store1, T_Store2, T_Return > Struct Template Reference

```
#include <solveresult.h>
```

Public Attributes

- T_Store1 [valCont](#)
- T_Store2 [valDis](#)
- T_Return [fitness](#)
- int [iterations](#)
- int [duration](#)
- bool [converged](#)

6.19.1 Detailed Description

```
template<typename T_Store1, typename T_Store2, typename T_Return>struct PRPSEvolution::-
Solve::solveresult_t< T_Store1, T_Store2, T_Return >
```

Stores the final state of a solution

6.19.2 Member Data Documentation

6.19.2.1 template<typename T_Store1, typename T_Store2, typename T_Return> bool
PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >::converged

Indicates whether the build in convergence criterium was applied, or not

6.19.2.2 template<typename T_Store1, typename T_Store2, typename T_Return> int
PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >::duration

The processing time for this solution

6.19.2.3 template<typename T_Store1, typename T_Store2, typename T_Return> T_Return
PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >::fitness

Whrere the result is stored The fitness value

6.19.2.4 template<typename T_Store1, typename T_Store2, typename T_Return> int
PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >::iterations

The amount of iterations needed for this result

6.19.2.5 template<typename T_Store1, typename T_Store2, typename T_Return> T_Store1
PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >::valCont

6.19.2.6 template<typename T_Store1, typename T_Store2, typename T_Return> T_Store2
PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >::valDis

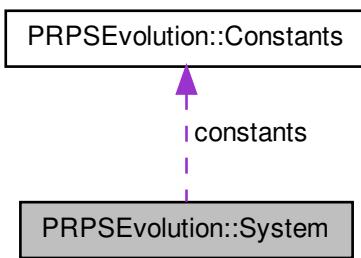
The documentation for this struct was generated from the following file:

- trunk/libSolve/[solveresult.h](#)

6.20 PRPSEvolution::System Struct Reference

```
#include <prpsevolutionsystem.h>
```

Collaboration diagram for PRPSEvolution::System:



Public Member Functions

- [System \(\)](#)
- [System \(const PRPSEvolution::System &s\)](#)
- [int rPRPSIniFile \(\)](#)

Public Attributes

- [PRPSEvolution::Constants constants](#)
- [std::string fn](#)

6.20.1 Constructor & Destructor Documentation

6.20.1.1 PRPSEvolution::System::System () [inline]

Here is the call graph for this function:



6.20.1.2 **PRPSEvolution::System::System (const PRPSEvolution::System & s)**
 [inline]

copy constructor

6.20.2 Member Function Documentation

6.20.2.1 **int PRPSEvolution::System::rPRPSIniFile ()** [inline]

Here is the caller graph for this function:



6.20.3 Member Data Documentation

6.20.3.1 **PRPSEvolution::Constants PRPSEvolution::System::constants**

6.20.3.2 **std::string PRPSEvolution::System::fn**

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

- trunk/libPRPSSystem/prpsevolutionsystem.h

6.21 PRPSEvolution::Solve::Ueber9000< T > Struct Template - Reference

#include <ueber9000.h>

Public Member Functions

- **Ueber9000 ()**
- **Ueber9000 (int i)**
- **Ueber9000 (const Ueber9000 &me)**
- **Ueber9000 (const NRmatrix< T > A_selected, const NRvector< T > b_selected)**
- **Ueber9000 (const std::vector< NRmatrix< T >> As, const std::vector< NRvector< T >> bs, const std::vector< std::string > namess, const int numO-Ants, const int select)**

- std::vector< std::vector< int > > [parseIdxFromNames](#) (const std::vector< std::string > &names)
- double [WholeTomato](#) (const ChromosomeT< double > &x)
- double [WholeTomatoMkII](#) (const ChromosomeT< double > &x)
- double [WholeTomatoMkII](#) (const ChromosomeT< double > &x1, const ChromosomeT< double > &x2)
- double [WholeTomatoMkII](#) (const ChromosomeT< double > &x, const ChromosomeT< int > &n)
- double [WholeTomato](#) (const NRmatrix< T > &A, const ChromosomeT< double > &x, const NRvector< T > &b)
- double [WholeTomatoMkI](#) (const NRmatrix< T > &A, const ChromosomeT< double > &x, const NRvector< T > &b)
- double [WholeTomatoMkII](#) (const NRmatrix< T > &A, const ChromosomeT< double > &x, const NRvector< T > &b)
- double [SuWi_WavenumberVariation](#) (const ChromosomeT< double > &n)
- double [SuWi_PositionVariation](#) (const ChromosomeT< double > &pos)
- double [fitnessSphere](#) (const ChromosomeT< double > &c)
- double [fitnessSphereMkII](#) (const ChromosomeT< double > &c1, const ChromosomeT< double > &c2)
- double [fitnessRosenbrock](#) (const ChromosomeT< double > &c)
- double [fitnessAckley](#) (const std::vector< double > &x)

Public Attributes

- double([Ueber9000](#)< double >::* [evaluate](#))(const ChromosomeT< double > &)
- double([Ueber9000](#)< double >::* [evaluateMkI](#))(const ChromosomeT< double > &)
- double([Ueber9000](#)< double >::* [evaluateMkII](#))(const ChromosomeT< double > &, const ChromosomeT< double > &)
- double([Ueber9000](#)< double >::* [evaluateMkIII](#))(const ChromosomeT< double > &, const ChromosomeT< int > &)
- int [Dimension](#)
- std::vector< NRmatrix< T > > [A](#)
- std::vector< NRvector< T > > [b](#)
- std::vector< std::string > [names](#)
- std::vector< std::vector< int > > [idxs](#)
- int [evaluations](#) = 0

6.21.1 Detailed Description

```
template<typename T>struct PRPSEvolution::Solve::Ueber9000< T >
```

Collect the fitness functions. Make sure they are static so we can function-pointer to them.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 template<typename T> PRPSEvolution::Solve::Ueber9000< T >::Ueber9000() [inline]

Default constructor

6.21.2.2 template<typename T> PRPSEvolution::Solve::Ueber9000< T >::Ueber9000(int i) [inline]

6.21.2.3 template<typename T> PRPSEvolution::Solve::Ueber9000< T >::Ueber9000(const Ueber9000< T > & me) [inline]

Here is the call graph for this function:



6.21.2.4 template<typename T> PRPSEvolution::Solve::Ueber9000< T >::Ueber9000(const NRmatrix< T > A_selected, const NRvector< T > b_selected) [inline]

Construct [Ueber9000](#) to use the WholeTomato as fitness function

Parameters

in	<i>A_selected</i>	The matrix A for this Solution
in	<i>c_k0_- selected</i>	The vector b for this Solution

Here is the call graph for this function:



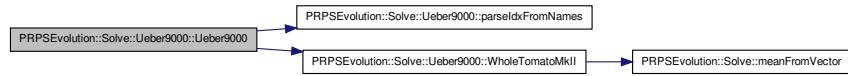
```
6.21.2.5 template<typename T> PRPSEvolution::Solve::Ueber9000< T
>::Ueber9000 ( const std::vector< NRmatrix< T >> As, const std::vector<
NRvector< T >> bs, const std::vector< std::string > namess, const int numOAnts,
const int select ) [inline]
```

Construct [Ueber9000](#) to use the WholeTomato as fitness function

Parameters

in	As	The matrices A to get a solution from
in	bs	The vectors b
in	namess	The Names of the matrices in As
in	numOAnts	The number of antennas used in the matrices in As
in	select	Selects the WholeTomato-Version

Here is the call graph for this function:



6.21.3 Member Function Documentation

```
6.21.3.1 template<typename T> double PRPSEvolution::Solve::Ueber9000< T
>::fitnessAckley ( const std::vector< double > & x ) [inline]
```

The infamous Ackley-function

```
6.21.3.2 template<typename T> double PRPSEvolution::Solve::Ueber9000< T
>::fitnessRosenbrock ( const ChromosomeT< double > & c ) [inline]
```

The Rosenbrock implementation

```
6.21.3.3 template<typename T> double PRPSEvolution::Solve::Ueber9000< T
>::fitnessSphere ( const ChromosomeT< double > & c ) [inline]
```

This ist the fitness function used in the EA algorithm

```
6.21.3.4 template<typename T> double PRPSEvolution::Solve::Ueber9000<
T >::fitnessSphereMkII ( const ChromosomeT< double > & c1, const
ChromosomeT< double > & c2 ) [inline]
```

This ist the fitness function used in the EA algorithm. This implementation uses two input vectors of the same datatype for test purpose of multi chromosome optimization

```
6.21.3.5 template<typename T> std::vector<std::vector<int>>
PRPSEvolution::Solve::Ueber9000< T >::parseIdxFromNames ( const
std::vector< std::string > & names ) [inline]
```

This function will parse the indeces used for a solution

Parameters

in	<i>names</i>	Contains the "Name" of each matrix we want to use in this solution
----	--------------	--

Returns

A two dimensional vector with the indeces of each antenna for each matrix

Here is the caller graph for this function:



```
6.21.3.6 template<typename T> double PRPSEvolution::Solve::Ueber9000< T
>::SuWi_PositionVariation ( const ChromosomeT< double > & pos )
[inline]
```

Approach 3 based on the thoughts of by S. Winter

```
6.21.3.7 template<typename T> double PRPSEvolution::Solve::Ueber9000< T
>::SuWi_WavenumberVariation ( const ChromosomeT< double > & n )
[inline]
```

Approach 2 based on the thoughts of S. Winter. Here we want to optimize the wavenumbers

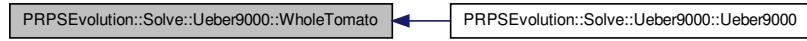
```
6.21.3.8 template<typename T> double PRPSEvolution::Solve::Ueber9000< T
>::WholeTomato ( const ChromosomeT< double > & x ) [inline]
```

This method basically wraps around the real WholeTomato-function. Maps the function so that it can be used with the evaluate-method

Parameters

in	<i>x</i>	The vector x
----	----------	--------------

Here is the caller graph for this function:



6.21.3.9 template<typename T> double PRPSEvolution::Solve::Ueber9000< T >::WholeTomato (const NRmatrix< T > & A, const ChromosomeT< double > & x, const NRvector< T > & b) [inline]

This approach will solve the scene defined by the 10x3 matrix. The approach is described in the Master-Thesis of C.Gnip. Basically solves the linear equation

$$r = \mathbf{Ax} - \mathbf{b}$$

Parameters

in	A	The 10x3 Matrix that is used in this solution
in	x	The vector containing the variables
in	b	Representing the vector b

Returns

The residuum of the equation system representing the "Fitness" of the given - Solution in

See also

x

6.21.3.10 template<typename T> double PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkl (const NRmatrix< T > & A, const ChromosomeT< double > & x, const NRvector< T > & b) [inline]

Todo documentation

Parameters

in	A	The 10x3 Matrix that is used in this solution
in	x	The vector containing the variables
in	b	Representing the vector b

Returns

The residuum of the equation system representing the "Fitness" of the given - Solution in

See also

x

6.21.3.11 template<typename T> double PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkII (const ChromosomeT< double > & x) [inline]

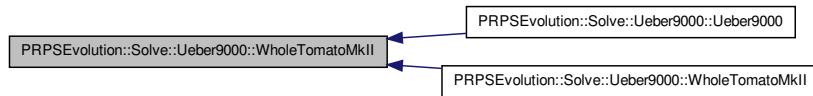
Todo document**Parameters**

in	x	The vector x containing the
----	---	-----------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



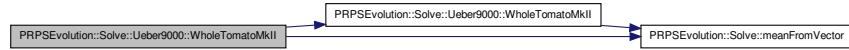
6.21.3.12 template<typename T> double PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkII (const ChromosomeT< double > & x1, const ChromosomeT< double > & x2) [inline]

Todo document

Parameters

in	x	The vector x containing the
----	---	-----------------------------

Here is the call graph for this function:



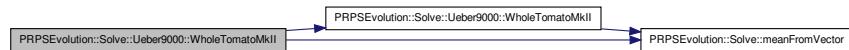
6.21.3.13 template<typename T> double PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkII (const ChromosomeT< double > & x, const ChromosomeT< int > & n) [inline]

Todo document

Parameters

in	x	The vector x containing the
----	---	-----------------------------

Here is the call graph for this function:



6.21.3.14 template<typename T> double PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoMkII (const NRmatrix< T > & A, const ChromosomeT< double > & x, const NRvector< T > & b) [inline]

This function contains the implementation of the whole model. This approach will solve calculate the 10x3 matrix described in the Master-Thesis of C.Gnip Basically solves the linear equation

$$r = Ax - b$$

Parameters

in	A	The 10x3 Matrix that ist used in this solution
in	x	The vector containing the variables
in	b	Representing the vector b

Returns

The residuum of the equation system representing the "Fitness" of the given - Solution in

See also

x

6.21.4 Member Data Documentation

6.21.4.1 template<typename T> std::vector<NRmatrix< T > >
PRPSEvolution::Solve::Ueber9000< T >::A

The Matrices we need to solve the Problem

6.21.4.2 template<typename T> std::vector<NRvector< T > >
PRPSEvolution::Solve::Ueber9000< T >::b

The b-vector needed to find a Solution

6.21.4.3 template<typename T> int **PRPSEvolution::Solve::Ueber9000< T >::Dimension**

The Dimension of the Problem

6.21.4.4 template<typename T> double(Ueber9000<double>::*
PRPSEvolution::Solve::Ueber9000< T >::evaluate)(const ChromosomeT<
double > &)

Todo document

6.21.4.5 template<typename T> double(Ueber9000<double>::*
PRPSEvolution::Solve::Ueber9000< T >::evaluateMkl)(const
ChromosomeT< double > &)

Todo document

6.21.4.6 template<typename T> double(Ueber9000<double>::*
PRPSEvolution::Solve::Ueber9000< T >::evaluateMklII)(const
ChromosomeT< double > &, const ChromosomeT< double > &)

Todo document

6.21.4.7 template<typename T> double(Ueber9000<double>::*
**PRPSEvolution::Solve::Ueber9000< T >::evaluateMkIII)(const
ChromosomeT< double > &, const ChromosomeT< int > &)**

Todo document

6.21.4.8 template<typename T> int **PRPSEvolution::Solve::Ueber9000< T
>::evaluations = 0**

6.21.4.9 template<typename T> std::vector<std::vector<int> >
PRPSEvolution::Solve::Ueber9000< T >::idxs

6.21.4.10 template<typename T> std::vector<std::string>
PRPSEvolution::Solve::Ueber9000< T >::names

The names for the Solution (contains the contributing antennas)

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

- trunk/libSolve/ueber9000.h

6.22 PRPSEvolution::WholeTomatoMkI_A Struct Reference

```
#include <WholeTomatoMkI_A.h>
```

Public Member Functions

- [WholeTomatoMkI_A](#) (unsigned int `numberOfVariables=7`)
- std::string `name () const`
From INameable: return the class name.
- std::size_t `numberOfVariables () const`
- bool `hasScalableDimensionality () const`
- void `setNumberOfVariables (std::size_t numberOfVariables)`
- void `configure (const PropertyTree &node)`
- void `proposeStartingPoint (SearchPointType &x) const`
- double `eval (const SearchPointType &x) const`
- void `setParams (const NRmatrix< Doub > &M, const NRvector< Doub > &v)`
- void `setMat (const NRmatrix< Doub > &M)`
- void `setVec (const NRvector< Doub > &v)`
- double `mkl (const NRmatrix< Doub > &A, const SearchPointType &x, const N-
Rvector< Doub > &b) const`

6.22.1 Constructor & Destructor Documentation

6.22.1.1 `PRPSEvolution::WholeTomatoMkl_A::WholeTomatoMkl_A (unsigned int
 numberOfVariables = 7) [inline]`

6.22.2 Member Function Documentation

6.22.2.1 `void PRPSEvolution::WholeTomatoMkl_A::configure (const PropertyTree &
 node) [inline]`

6.22.2.2 `double PRPSEvolution::WholeTomatoMkl_A::eval (const SearchPointType & x
) const [inline]`

6.22.2.3 `bool PRPSEvolution::WholeTomatoMkl_A::hasScalableDimensionality ()
 const [inline]`

6.22.2.4 `double PRPSEvolution::WholeTomatoMkl_A::mkl (const NRmatrix< Doub >
 & A, const SearchPointType & x, const NRvector< Doub > & b) const [inline]`

Todo documentation

Parameters

in	A	The 10x3 Matrix that ist used in this solution
in	x	The vector containing the variables
in	b	Representing the vector b

Returns

The residuum of the equation system representing the "Fitness" of the given - Solution in

See also

x

6.22.2.5 `std::string PRPSEvolution::WholeTomatoMkl_A::name () const [inline]`

From INameable: return the class name.

6.22.2.6 `std::size_t PRPSEvolution::WholeTomatoMkl_A::numberOfVariables () const [inline]`

6.22.2.7 `void PRPSEvolution::WholeTomatoMkl_A::proposeStartingPoint (SearchPointType & x) const [inline]`

- 6.22.2.8 void PRPSEvolution::WholeTomatoMkl_A::setMat (const NRmatrix< Doub > & M) [inline]
- 6.22.2.9 void PRPSEvolution::WholeTomatoMkl_A::setNumberOfVariables (std::size_t numberOfVariables) [inline]
- 6.22.2.10 void PRPSEvolution::WholeTomatoMkl_A::setParams (const NRmatrix< Doub > & M, const NRvector< Doub > & v) [inline]
- 6.22.2.11 void PRPSEvolution::WholeTomatoMkl_A::setVec (const NRvector< Doub > & v) [inline]

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

- trunk/libSolve/Objectivefunctions/[WholeTomatoMkl_A.h](#)

6.23 PRPSEvolution::WholeTomatoMkl_B Struct Reference

```
#include <WholeTomatoMkl_B.h>
```

Public Member Functions

- [WholeTomatoMkl_B](#) (unsigned int `numberOfVariables`=7)
- std::string `name` () const

From INameable: return the class name.
- std::size_t `numberOfVariables` () const
- bool `hasScalableDimensionality` () const
- void `setNumberOfVariables` (std::size_t `numberOfVariables`)
- void `configure` (const PropertyTree &node)
- void `proposeStartingPoint` (SearchPointType &x) const
- double `eval` (const SearchPointType &x) const
- void `setParams` (const NRmatrix< Doub > &M, const NRvector< Doub > &v)
- void `setMat` (const NRmatrix< Doub > &M)
- void `setVec` (const NRvector< Doub > &v)
- double `mkl` (const NRmatrix< Doub > &A, const SearchPointType &x, const NRvector< Doub > &b) const

6.23.1 Constructor & Destructor Documentation

- 6.23.1.1 PRPSEvolution::WholeTomatoMkl_B::WholeTomatoMkl_B (unsigned int `numberOfVariables` = 7) [inline]

6.23.2 Member Function Documentation

- 6.23.2.1 void PRPSEvolution::WholeTomatoMkl_B::configure (const PropertyTree & node) [inline]
- 6.23.2.2 double PRPSEvolution::WholeTomatoMkl_B::eval (const SearchPointType & x) const [inline]
- 6.23.2.3 bool PRPSEvolution::WholeTomatoMkl_B::hasScalableDimensionality () const [inline]
- 6.23.2.4 double PRPSEvolution::WholeTomatoMkl_B::mkl (const NRmatrix< Doub > & A, const SearchPointType & x, const NRvector< Doub > & b) const [inline]

Todo documentation

Parameters

in	A	The 10x3 Matrix that ist used in this solution
in	x	The vector containing the variables
in	b	Representing the vector b

Returns

The residuum of the equation system representing the "Fitness" of the given - Solution in

See also

x

- 6.23.2.5 std::string PRPSEvolution::WholeTomatoMkl_B::name () const [inline]

From INameable: return the class name.

- 6.23.2.6 std::size_t PRPSEvolution::WholeTomatoMkl_B::numberOfVariables () const [inline]
- 6.23.2.7 void PRPSEvolution::WholeTomatoMkl_B::proposeStartingPoint (SearchPointType & x) const [inline]
- 6.23.2.8 void PRPSEvolution::WholeTomatoMkl_B::setMat (const NRmatrix< Doub > & M) [inline]
- 6.23.2.9 void PRPSEvolution::WholeTomatoMkl_B::setNumberOfVariables (std::size_t numberOfVariables) [inline]

6.23.2.10 void PRPSEvolution::WholeTomatoMkl_B::setParams (const NRmatrix< Doub > & M, const NRvector< Doub > & v) [inline]

6.23.2.11 void PRPSEvolution::WholeTomatoMkl_B::setVec (const NRvector< Doub > & v) [inline]

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

- trunk/libSolve/Objectivefunctions/[WholeTomatoMkl_B.h](#)

6.24 PRPSEvolution::Models::WholeTomatoMkII Struct Reference

```
#include <WholeTomatoMkII.h>
```

Public Types

- `typedef AbstractOptimizer < shark::VectorSpace< double >, double, SingleObjectiveResultSet < typename shark::VectorSpace < double >::PointType > > base_type`
- `typedef base_type::ObjectiveFunctionType ObjectiveFunctionType`

Public Member Functions

- `WholeTomatoMkII (unsigned int numberOfVariables=5)`
- `std::string name () const`
From INameable: return the class name.
- `std::size_t numberOfVariables () const`
- `bool hasScalableDimensionality () const`
- `void setNumberOfVariables (std::size_t numberOfVariables)`
- `void configure (const PropertyTree &node)`
- `void proposeStartingPoint (SearchPointType &x) const`
- `double eval (const SearchPointType &p) const`
- `double mkII (const NRmatrix< Doub > &A, const double *x, const NRvector< Doub > &b) const`
- `bool constrains (const double *x) const`
- `void setParams (const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::string > &n)`
- `void setParams (const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::vector< int >> &i)`

6.24.1 Member Typedef Documentation

6.24.1.1 `typedef AbstractOptimizer<shark::VectorSpace< double >,double,SingleObjectiveResultSet<typename shark::VectorSpace< double >::PointType> > PRPSEvolution::Models::WholeTomatoMkII::base_type`

6.24.1.2 `typedef base_type::ObjectiveFunctionType PRPSEvolution::Models::WholeTomatoMkII::ObjectiveFunctionType`

6.24.2 Constructor & Destructor Documentation

6.24.2.1 `PRPSEvolution::Models::WholeTomatoMkII::WholeTomatoMkII (unsigned int numberOfVariables = 5) [inline]`

6.24.3 Member Function Documentation

6.24.3.1 `void PRPSEvolution::Models::WholeTomatoMkII::configure (const PropertyTree & node) [inline]`

6.24.3.2 `bool PRPSEvolution::Models::WholeTomatoMkII::constraints (const double * x) const [inline]`

Collects the constraints for this model

Parameters

in	<i>x</i>	The vector containing the variables
----	----------	-------------------------------------

Returns

6.24.3.3 `double PRPSEvolution::Models::WholeTomatoMkII::eval (const SearchPointType & p) const`

Here is the call graph for this function:



6.24.3.4 `bool PRPSEvolution::Models::WholeTomatoMkII::hasScalableDimensionality () const [inline]`

6.24.3.5 double PRPSEvolution::Models::WholeTomatoMkII::mkII (const NRmatrix< Doub > & A, const double * x, const NRvector< Doub > & b) const [inline]

The model itself.

Parameters

in	A	The Matrix A. for this Solution
in	x	The x vector, contains the free parameter
in	b	The b vector for this solution

Returns

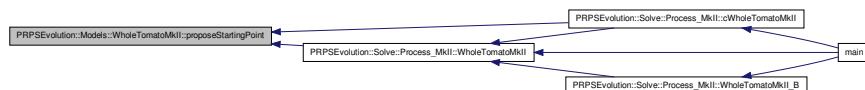
6.24.3.6 std::string PRPSEvolution::Models::WholeTomatoMkII::name () const [inline]

From INameable: return the class name.

6.24.3.7 std::size_t PRPSEvolution::Models::WholeTomatoMkII::numberOfVariables () const [inline]

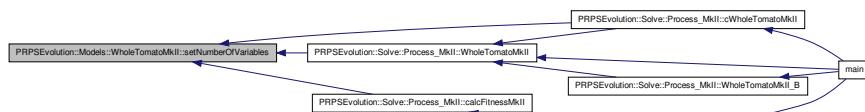
6.24.3.8 void PRPSEvolution::Models::WholeTomatoMkII::proposeStartingPoint (SearchPointType & x) const [inline]

Here is the caller graph for this function:



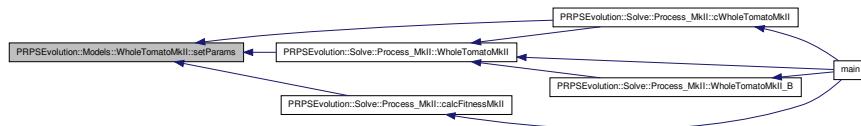
6.24.3.9 void PRPSEvolution::Models::WholeTomatoMkII::setNumberOfVariables (std::size_t *numberOfVariables*) [inline]

Here is the caller graph for this function:



6.24.3.10 void PRPSEvolution::Models::WholeTomatoMkII::setParams (const std::vector< NRmatrix< Doub >> & M, const std::vector< NRvector< Doub >> & v, const std::vector< std::string > & n) [inline]

Here is the caller graph for this function:



6.24.3.11 void PRPSEvolution::Models::WholeTomatoMkII::setParams (const std::vector< NRmatrix< Doub >> & M, const std::vector< NRvector< Doub >> & v, const std::vector< std::vector< int >> & i) [inline]

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

- trunk/libSolve/Objectivefunctions/[WholeTomatoMkII.h](#)
- trunk/libSolve/Objectivefunctions/[WholeTomatoMkII.cpp](#)

6.25 PRPSEvolution::Models::WholeTomatoMkII_B Struct Reference

```
#include <WholeTomatoMkII_B.h>
```

Public Types

- `typedef AbstractOptimizer < shark::VectorSpace< double >, double, SingleObjectiveResultSet < typename shark::VectorSpace < double >::PointType > > base_type`
- `typedef base_type::ObjectiveFunctionType ObjectiveFunctionType`

Public Member Functions

- `WholeTomatoMkII_B (unsigned int numberOfVariables=5)`
- `std::string name () const`
From INameable: return the class name.
- `std::size_t numberOfVariables () const`
- `bool hasScalableDimensionality () const`
- `void setNumberOfVariables (std::size_t numberOfVariables)`
- `void configure (const PropertyTree &node)`

- void `proposeStartingPoint` (SearchPointType &x) const
- double `eval` (const SearchPointType &p) const
- double `mkII` (const NRmatrix< Doub > &A, const double *x, const NRvector< Doub > &b) const
- void `setParams` (const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::string > &n)
- void `setParams` (const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::vector< int >> &i)
- void `setMats` (const std::vector< NRmatrix< Doub >> &M)
- void `setVecs` (const std::vector< NRvector< Doub >> &v)
- void `setNames` (const std::vector< std::string > &n)
- void `setIdx` (const std::vector< std::vector< int >> &i)

6.25.1 Member Typedef Documentation

6.25.1.1 `typedef AbstractOptimizer<shark::VectorSpace< double >,double,Single-ObjectiveResultSet<typename shark::VectorSpace< double >::PointType> >`
`PRPSEvolution::Models::WholeTomatoMkII_B::base_type`

6.25.1.2 `typedef base_type::ObjectiveFunctionType PRPSEvolution::Models::Whole-TomatoMkII_B::ObjectiveFunctionType`

6.25.2 Constructor & Destructor Documentation

6.25.2.1 `PRPSEvolution::Models::WholeTomatoMkII_B::WholeTomatoMkII_B (unsigned int numberOfVariables = 5) [inline]`

6.25.3 Member Function Documentation

6.25.3.1 `void PRPSEvolution::Models::WholeTomatoMkII_B::configure (const PropertyTree & node) [inline]`

6.25.3.2 `double PRPSEvolution::Models::WholeTomatoMkII_B::eval (const SearchPointType & p) const`

Here is the call graph for this function:



6.25.3.3 `bool PRPSEvolution::Models::WholeTomatoMkII_B::hasScalableDimensionality() const [inline]`

6.25.3.4 `double PRPSEvolution::Models::WholeTomatoMkII_B::mkII(const NRmatrix< Doub > & A, const double * x, const NRvector< Doub > & b) const [inline]`

The model itself.

Parameters

in	<i>A</i>	The Matrix A. for this Solution
in	<i>x</i>	The x vector, contains the free parameter
in	<i>b</i>	The b vector for this solution

6.25.3.5 `std::string PRPSEvolution::Models::WholeTomatoMkII_B::name() const [inline]`

From INameable: return the class name.

6.25.3.6 `std::size_t PRPSEvolution::Models::WholeTomatoMkII_B::numberOfVariables() const [inline]`

6.25.3.7 `void PRPSEvolution::Models::WholeTomatoMkII_B::proposeStartingPoint(SearchPointType & x) const [inline]`

Here is the caller graph for this function:



6.25.3.8 `void PRPSEvolution::Models::WholeTomatoMkII_B::setIdx(const std::vector< std::vector< int >> & i) [inline]`

6.25.3.9 `void PRPSEvolution::Models::WholeTomatoMkII_B::setMats(const std::vector< NRmatrix< Doub >> & M) [inline]`

6.25.3.10 `void PRPSEvolution::Models::WholeTomatoMkII_B::setNames(const std::vector< std::string > & n) [inline]`

**6.25.3.11 void PRPSEvolution::Models::WholeTomatoMkII_B-
::setNumberOfVariables (std::size_t *numberOfVariables*)
[inline]**

Here is the caller graph for this function:



**6.25.3.12 void PRPSEvolution::Models::WholeTomatoMkII_B::setParams (const
std::vector< NRmatrix< Doub >> & M, const std::vector< NRvector< Doub >> &
v, const std::vector< std::string > & n) [inline]**

Here is the caller graph for this function:



**6.25.3.13 void PRPSEvolution::Models::WholeTomatoMkII_B::setParams (const
std::vector< NRmatrix< Doub >> & M, const std::vector< NRvector< Doub >> &
v, const std::vector< std::vector< int >> & i) [inline]**

**6.25.3.14 void PRPSEvolution::Models::WholeTomatoMkII_B::setVecs (const
std::vector< NRvector< Doub >> & v) [inline]**

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

- trunk/libSolve/Objectivefunctions/[WholeTomatoMkII_B.h](#)
- trunk/libSolve/Objectivefunctions/[WholeTomatoMkII_B.cpp](#)

6.26 PRPSEvolution::Models::WholeTomatoMkIII Struct Reference

```
#include <WholeTomatoMkIII.h>
```

Public Types

- `typedef AbstractOptimizer < shark::VectorSpace< double >, double, Single-ObjectiveResultSet < typename shark::VectorSpace < double >::PointType >`

- > [base_type](#)
- [typedef base_type::ObjectiveFunctionType ObjectiveFunctionType](#)

Public Member Functions

- [WholeTomatoMkIII \(unsigned int `numberOfVariables`=5\)](#)
- [std::string `name` \(\) const](#)
From INameable: return the class name.
- [std::size_t `numberOfVariables` \(\) const](#)
- [bool `hasScalableDimensionality` \(\) const](#)
- [void `setNumberOfVariables` \(std::size_t `numberOfVariables`\)](#)
- [void `configure` \(const PropertyTree &node\)](#)
- [void `proposeStartingPoint` \(SearchPointType &x\) const](#)
- [double `eval` \(const SearchPointType &p\) const](#)
- [double `mkIII` \(const NRmatrix< Doub > &A, const double *x, const NRvector< Doub > &b\) const](#)
- [bool `constrains` \(const double *x\) const](#)
- [void `setParams` \(const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::string > &n\)](#)
- [void `setParams` \(const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::vector< int >> &i\)](#)
- [void `setMats` \(const std::vector< NRmatrix< Doub >> &M\)](#)
- [void `setVecs` \(const std::vector< NRvector< Doub >> &v\)](#)
- [void `setNames` \(const std::vector< std::string > &n\)](#)
- [void `setIdx` \(const std::vector< std::vector< int >> &i\)](#)

6.26.1 Member Typedef Documentation

6.26.1.1 [typedef AbstractOptimizer<shark::VectorSpace< double >,double,SingleObjectiveResultSet<typename shark::VectorSpace< double >::PointType>> PRPSEvolution::Models::WholeTomatoMkIII::base_type](#)

6.26.1.2 [typedef base_type::ObjectiveFunctionType PRPSEvolution::Models::WholeTomatoMkIII::ObjectiveFunctionType](#)

6.26.2 Constructor & Destructor Documentation

6.26.2.1 PRPSEvolution::Models::WholeTomatoMkIII::WholeTomatoMkIII (
unsigned int *numberOfVariables* = 5) [inline]

Here is the call graph for this function:



6.26.3 Member Function Documentation

**6.26.3.1 void PRPSEvolution::Models::WholeTomatoMkIII::configure (const
 PropertyTree & *node*) [inline]**

**6.26.3.2 bool PRPSEvolution::Models::WholeTomatoMkIII::constraints (const
 double * *x*) const [inline]**

Collects the constraints for this model

Parameters

in	x	The vector containing the variables
----	---	-------------------------------------

Returns

**6.26.3.3 double PRPSEvolution::Models::WholeTomatoMkIII::eval (const
 SearchPointType & *p*) const**

Here is the call graph for this function:



**6.26.3.4 bool PRPSEvolution::Models::WholeTomatoMkIII::hasScalable-
 Dimensionality() const [inline]**

6.26.3.5 double PRPSEvolution::Models::WholeTomatoMkIII::mkII (const NRmatrix< Doub > & A, const double * x, const NRvector< Doub > & b) const [inline]

The model itself.

Parameters

in	A	The Matrix A. for this Solution
in	x	The x vector, contains the free parameter
in	b	The b vector for this solution

Returns

6.26.3.6 std::string PRPSEvolution::Models::WholeTomatoMkIII::name () const [inline]

From INameable: return the class name.

6.26.3.7 std::size_t PRPSEvolution::Models::WholeTomatoMkIII::numberOfVariables () const [inline]

6.26.3.8 void PRPSEvolution::Models::WholeTomatoMkIII::proposeStartingPoint (SearchPointType & x) const [inline]

6.26.3.9 void PRPSEvolution::Models::WholeTomatoMkIII::setIdx (const std::vector< std::vector< int >> & i) [inline]

6.26.3.10 void PRPSEvolution::Models::WholeTomatoMkIII::setMats (const std::vector< NRmatrix< Doub >> & M) [inline]

6.26.3.11 void PRPSEvolution::Models::WholeTomatoMkIII::setNames (const std::vector< std::string > & n) [inline]

6.26.3.12 void PRPSEvolution::Models::WholeTomatoMkIII::setNumberOfVariables (std::size_t *numberOfVariables*) [inline]

6.26.3.13 void PRPSEvolution::Models::WholeTomatoMkIII::setParams (const std::vector< NRmatrix< Doub >> & M, const std::vector< NRvector< Doub >> & v, const std::vector< std::string > & n) [inline]

6.26.3.14 void PRPSEvolution::Models::WholeTomatoMkIII::setParams (const std::vector< NRmatrix< Doub >> & M, const std::vector< NRvector< Doub >> & v, const std::vector< std::vector< int >> & i) [inline]

6.26.3.15 void PRPSEvolution::Models::WholeTomatoMkIII::setVecs (const std::vector< NRvector< Doub >> & v) [inline]

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

- trunk/libSolve/Objectivefunctions/[WholeTomatoMkIII.h](#)
- trunk/libSolve/Objectivefunctions/[WholeTomatoMkIII.cpp](#)

6.27 PRPSEvolution::Models::WholeTomatoReduced Struct - Reference

```
#include <WholeTomatoReduced.h>
```

Public Types

- `typedef AbstractOptimizer < shark::VectorSpace< double >, double, SingleObjectiveResultSet < typename shark::VectorSpace < double >::PointType > > base_type`
- `typedef base_type::ObjectiveFunctionType ObjectiveFunctionType`

Public Member Functions

- `WholeTomatoReduced (unsigned int numberOfVariables=Solve::ProblemDimensions::WholeTomatoReduced)`
- `std::string name () const`
From INameable: return the class name.
- `std::size_t numberOfVariables () const`
- `bool hasScalableDimensionality () const`
- `void setNumberOfVariables (std::size_t numberOfVariables)`
- `void configure (const PropertyTree &node)`
- `void proposeStartingPoint (SearchPointType &x) const`
- `double eval (const SearchPointType &p) const`
- `double reduced (const NRmatrix< Doub > &A, const double *x, const NRvector< Doub > &b) const`
- `bool constrains (const double *x) const`
- `void setParams (const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::string > &n, const std::vector< NRvector< Doub >> &c, const double l)`
- `void setParams (const std::vector< NRmatrix< Doub >> &M, const std::vector< NRvector< Doub >> &v, const std::vector< std::vector< int >> &i, const std::vector< NRvector< Doub >> &c)`
- `std::array< double, 8 > calcWavenumbers2 (double x, double y, double z)`

6.27.1 Member Typedef Documentation

- 6.27.1.1 `typedef AbstractOptimizer<shark::VectorSpace< double >,double,Single-ObjectiveResultSet<typename shark::VectorSpace< double >::PointType> > PRPSEvolution::Models::WholeTomatoReduced::base_type`
- 6.27.1.2 `typedef base_type::ObjectiveFunctionType PRPSEvolution::Models::WholeTomatoReduced::ObjectiveFunctionType`

6.27.2 Constructor & Destructor Documentation

- 6.27.2.1 `PRPSEvolution::Models::WholeTomatoReduced::WholeTomatoReduced (unsigned int numberOfVariables = Solve::ProblemDimensions::WholeTomatoReduced) [inline]`

6.27.3 Member Function Documentation

- 6.27.3.1 `std::array< double, 8 > PRPSEvolution::Models::WholeTomatoReduced::calcWavenumbers2 (double x, double y, double z)`

Here is the caller graph for this function:



- 6.27.3.2 `void PRPSEvolution::Models::WholeTomatoReduced::configure (const PropertyTree & node) [inline]`

- 6.27.3.3 `bool PRPSEvolution::Models::WholeTomatoReduced::constraints (const double * x) const [inline]`

Collects the constraints for this model

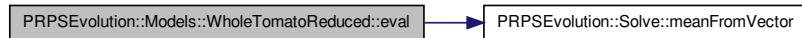
Parameters

in	<i>x</i>	The vector containing the variables
----	----------	-------------------------------------

Returns

6.27.3.4 double PRPSEvolution::Models::WholeTomatoReduced::eval (const SearchPointType & p) const

Here is the call graph for this function:



6.27.3.5 bool PRPSEvolution::Models::WholeTomatoReduced::hasScalable-Dimensionality () const [inline]

6.27.3.6 std::string PRPSEvolution::Models::WholeTomatoReduced::name () const [inline]

From INameable: return the class name.

6.27.3.7 std::size_t PRPSEvolution::Models::WholeTomatoReduced::numberOfVariables () const [inline]

6.27.3.8 void PRPSEvolution::Models::WholeTomatoReduced::proposeStartingPoint (SearchPointType & x) const [inline]

Here is the caller graph for this function:



6.27.3.9 double PRPSEvolution::Models::WholeTomatoReduced::reduced (const NRmatrix< Doub > & A, const double * x, const NRvector< Doub > & b) const [inline]

The model itself.

Parameters

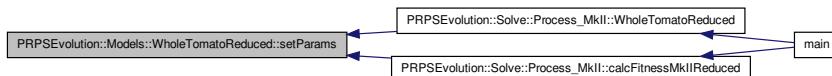
in	A	The Matrix A. for this Solution
in	x	The x vector, contains the free parameter
in	b	The b vector for this solution

Returns

6.27.3.10 void PRPSEvolution::Models::WholeTomatoReduced-
::setNumberOfVariables (std::size_t *numberOfVariables*)
[inline]

6.27.3.11 void PRPSEvolution::Models::WholeTomatoReduced::setParams (const
std::vector< NRmatrix< Doub >> & *M*, const std::vector< NRvector< Doub >> &
v, const std::vector< std::string > & *n*, const std::vector< NRvector< Doub >> &
c, const double *l*) [inline]

Here is the caller graph for this function:



6.27.3.12 void PRPSEvolution::Models::WholeTomatoReduced::setParams (const
std::vector< NRmatrix< Doub >> & *M*, const std::vector< NRvector< Doub >> &
v, const std::vector< std::vector< int >> & *i*, const std::vector< NRvector< Doub >> &
c) [inline]

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

- trunk/libSolve/Objectivefunctions/[WholeTomatoReduced.h](#)
- trunk/libSolve/Objectivefunctions/[WholeTomatoReduced.cpp](#)

Chapter 7

File Documentation

7.1 trunk/CMakeFiles/CompilerIdC/CMakeCCCompilerId.c File - Reference

Defines

- #define COMPILER_ID ""
- #define PLATFORM_ID ""
- #define ARCHITECTURE_ID ""

Functions

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

Variables

- char const * info_compiler = "]"
- char const * info_platform = "]"
- char const * info_arch = "]"

7.1.1 Define Documentation

7.1.1.1 #define ARCHITECTURE_ID """

7.1.1.2 #define COMPILER_ID """

7.1.1.3 #define PLATFORM_ID """

7.1.2 Function Documentation

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

7.1.3 Variable Documentation

7.1.3.1 `char const* info_arch = "]"`

7.1.3.2 `char const* info_compiler = "]"`

7.1.3.3 `char const* info_platform = "]"`

7.2 trunk/CMakeFiles/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

Defines

- `#define COMPILER_ID ""`
- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`

Functions

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

Variables

- `char const * info_compiler = "]"`
- `char const * info_platform = "]"`
- `char const * info_arch = "]"`

7.2.1 Define Documentation

7.2.1.1 `#define ARCHITECTURE_ID ""`

7.2.1.2 `#define COMPILER_ID ""`

7.2.1.3 `#define PLATFORM_ID ""`

7.2.2 Function Documentation

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

7.2.3 Variable Documentation

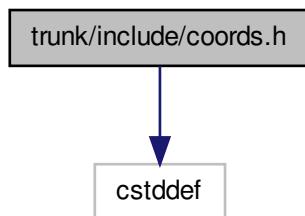
7.2.3.1 `char const* info_arch = "]"`

7.2.3.2 `char const* info_compiler = "]";`

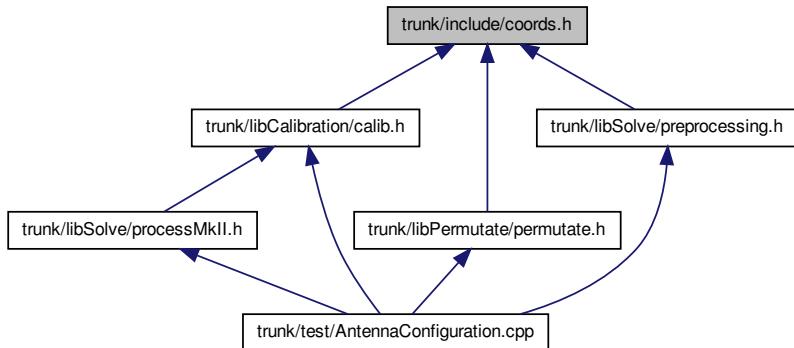
7.2.3.3 `char const* info_platform = "]";`

7.3 trunk/include/coords.h File Reference

#include <cstddef> Include dependency graph for coords.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Positioning::CoordContainer< N, T >](#)

Namespaces

- namespace PRPSEvolution
- namespace PRPSEvolution::Positioning

7.4 trunk/include/prps.h File Reference

Variables

- const int ANTENNA_AMOUNT = 8
- const int EXPECTED_LINES = 10
- const int EXPECTED_VALUES = 10

7.4.1 Variable Documentation

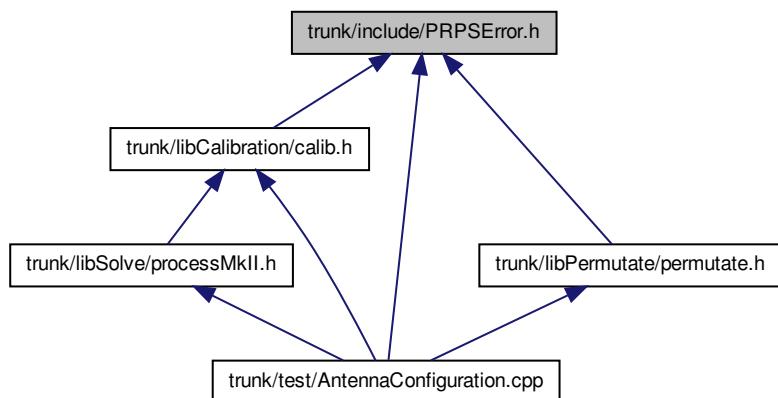
7.4.1.1 const int ANTENNA_AMOUNT = 8

7.4.1.2 const int EXPECTED_LINES = 10

7.4.1.3 const int EXPECTED_VALUES = 10

7.5 trunk/include/PRPSError.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace PRPSError
- namespace PRPSError::FileIO

Variables

- const int PRPSError::FileIO::okay = 0
- const int PRPSError::FileIO::generalError = -1
- const int PRPSError::FileIO::fnf = -2
- const int PRPSError::FileIO::inputmalformed = -3
- const int PRPSError::okay = 0
- const int PRPSError::general = -1
- const int PRPSError::critical = 10

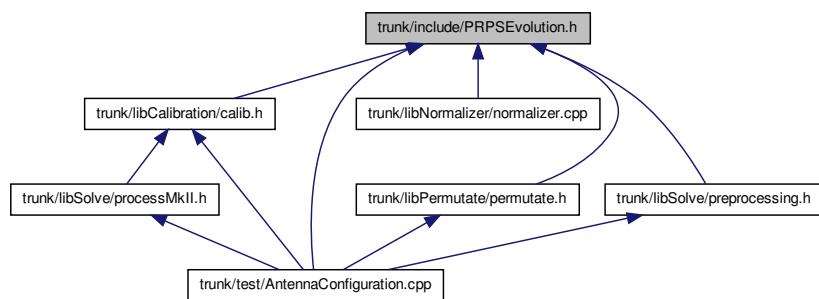
7.5.1 Detailed Description

Date

2013|Jun|18 This file contains definitions belonging to the PRPSError-namespace.
It is split into sub-namespaces for keeping thing nicely small.

7.6 trunk/include/PRPSEvolution.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace PRPSEvolution

Variables

- const int PRPSEvolution::ANTENNA_AMOUNT = 8
- const int PRPSEvolution::EXPECTED_LINES_CALIBRATION_FILE = 4
- const int PRPSEvolution::EXPECTED_VALUES_CALIBRATION_FILE = ANTE-NNA_AMOUNT
- const int PRPSEvolution::EXPECTED_LINES_COORD_FILE = ANTENNA_AMOUNT
- const int PRPSEvolution::EXPECTED_VALUES_COORD_FILE = 3
- const int PRPSEvolution::EXPECTED_LINES_SYSTEM_INI_FILE = 2
- const int PRPSEvolution::MAT_ROWS = 3
- const int PRPSEvolution::MAT_COLS = 10
- const int PRPSEvolution::CALIBRATION_POINTS_AVAILABLE = 4
- const int PRPSEvolution::EXPECTED_LINES_MEASUREMENT_FILE = ANTE-NNA_AMOUNT
- const int PRPSEvolution::EXPECTED_VALUES_MEASUREMENT_FILE = 2
- const int PRPSEvolution::DATA_NV = 65535
- const int PRPSEvolution::DEFAULT_MIN_GROUP_SIZE = 4

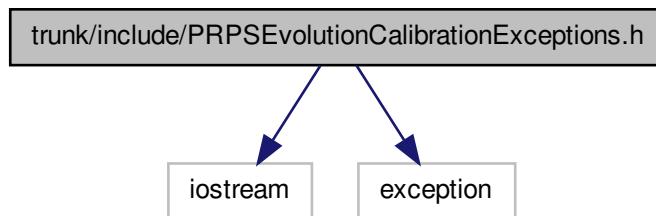
7.6.1 Detailed Description

Date

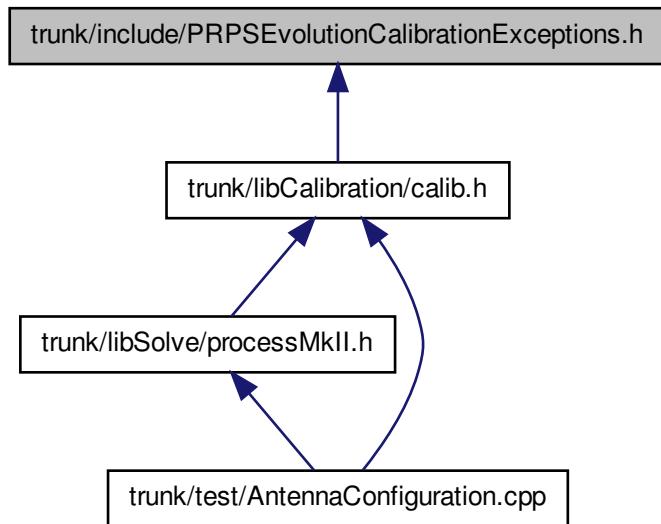
2013|Jun|18 This file collects definitions belonging to the PRPSEvolution-namespace. Especially const. defines.

7.7 trunk/include/PRPSEvolutionCalibrationExceptions.h File - Reference

#include <iostream> #include <exception> Include dependency graph for PRPSEvolutionCalibrationExceptions.h:



This graph shows which files directly or indirectly include this file:



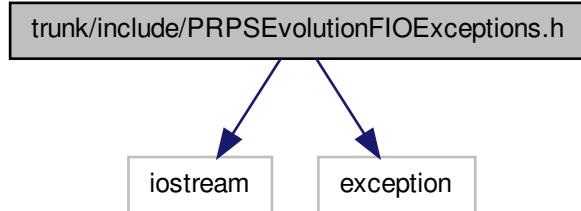
Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Exceptions](#)
- namespace [PRPSEvolution::Exceptions::Calibration](#)

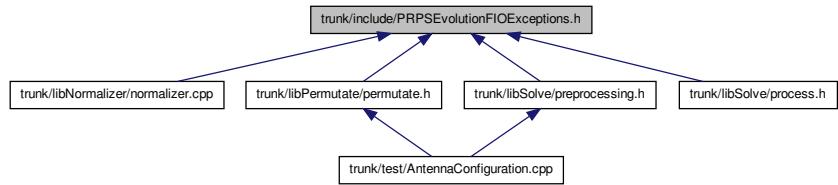
7.8 trunk/include/PRPSEvolutionFIOExceptions.h File Reference

```
#include <iostream> #include <exception> Include dependency
```

graph for PRPSEvolutionFIOExceptions.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Exceptions::FileIO::FileNotFoundException](#)
- struct [PRPSEvolution::Exceptions::FileIO::MalformedInput](#)
- struct [PRPSEvolution::Exceptions::FileIO::OutputFailure](#)

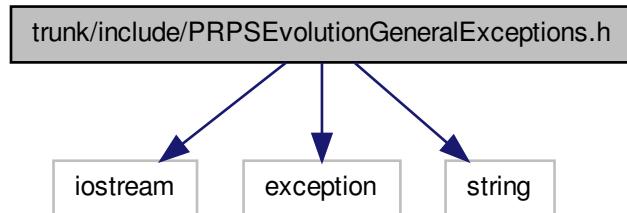
Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Exceptions](#)
- namespace [PRPSEvolution::Exceptions::FileIO](#)

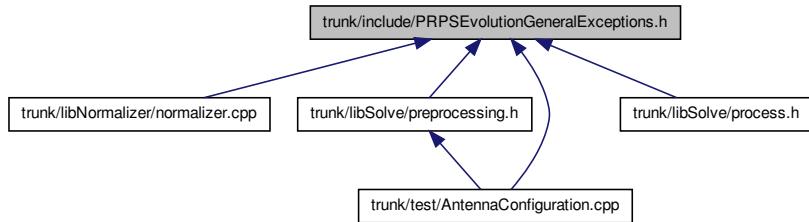
7.9 trunk/include/PRPSEvolutionGeneralExceptions.h File Reference

```
#include <iostream> #include <exception> #include <string> ×
```

Include dependency graph for PRPSEvolutionGeneralExceptions.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Exceptions::General::NotImplemented](#)

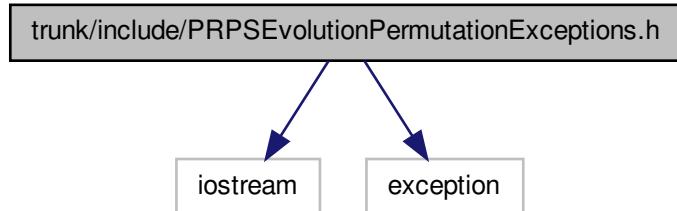
Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Exceptions](#)
- namespace [PRPSEvolution::Exceptions::General](#)

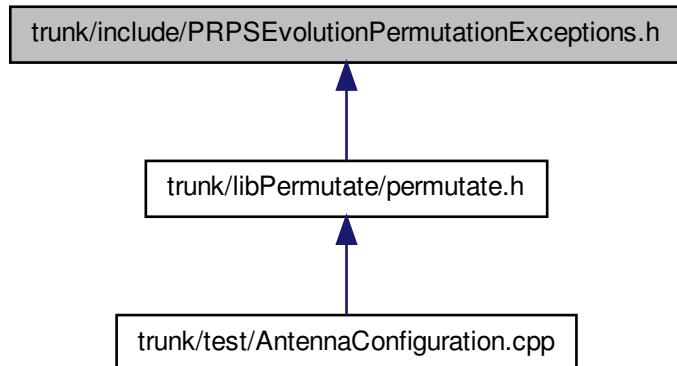
7.10 trunk/include/PRPSEvolutionPermutationExceptions.h File - Reference

```
#include <iostream> #include <exception> Include dependency
```

graph for PRPSEvolutionPermutationExceptions.h:



This graph shows which files directly or indirectly include this file:

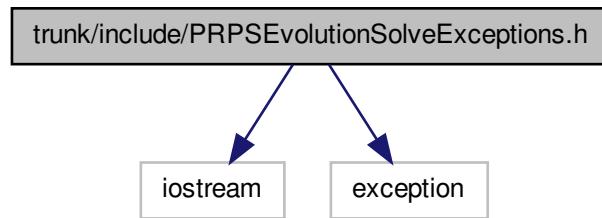


Namespaces

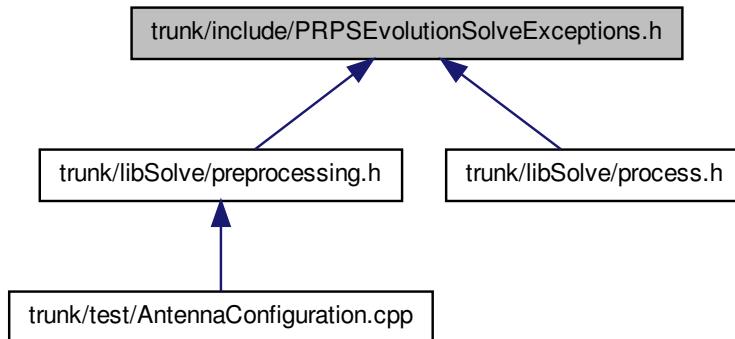
- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Exceptions](#)
- namespace [PRPSEvolution::Exceptions::Permutation](#)

7.11 trunk/include/PRPSEvolutionSolveExceptions.h File Reference

```
#include <iostream> #include <exception> Include dependency  
graph for PRPSEvolutionSolveExceptions.h:
```



This graph shows which files directly or indirectly include this file:



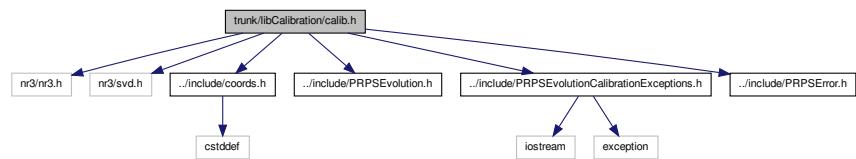
Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Exceptions](#)
- namespace [PRPSEvolution::Exceptions::Solve](#)

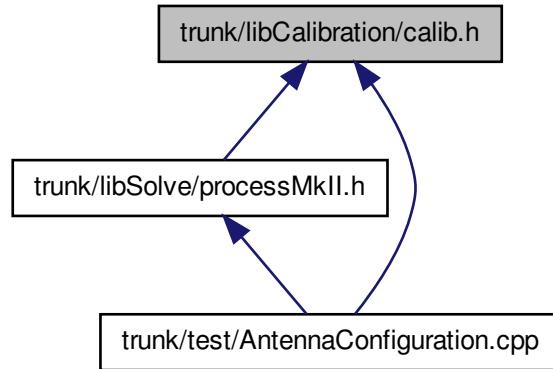
7.12 trunk/libCalibration/calib.cpp File Reference

7.13 trunk/libCalibration/calib.h File Reference

```
#include <nr3/nr3.h> #include <nr3/svd.h> #include "../include/coords.h"
#include "../include/PRPSEvolution.h" #include "../include/-PRPSEvolutionCalibrationExceptions.h" #include "../include/-PRPSError.h" Include dependency graph for calib.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Calibration](#)

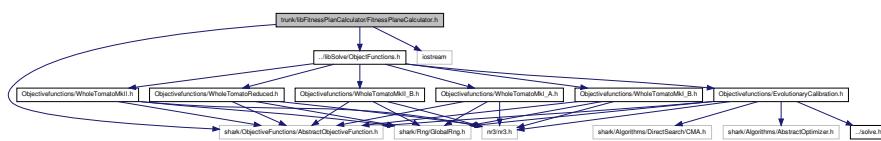
7.13.1 Detailed Description

Date

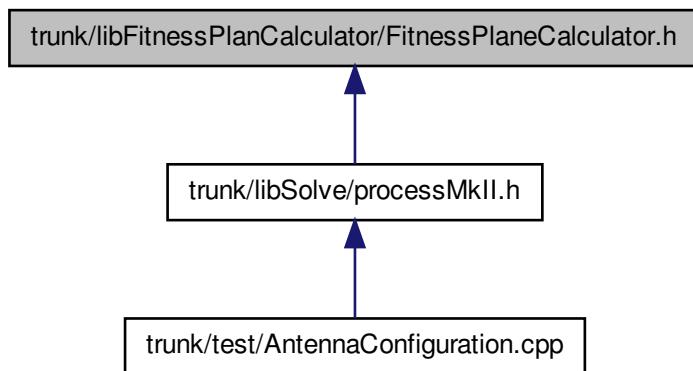
2013|Jun|25

7.14 trunk/libFitnessPlanCalculator/FitnessPlaneCalculator.h File - Reference

```
#include <shark/ObjectiveFunctions/AbstractObjectiveFunction.h>      #include "../libSolve/ObjectFunctions.h" ×
#include <iostream> Include dependency graph for FitnessPlaneCalculator.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Support::FitnessPlaneLimits< N >](#)
- class [PRPSEvolution::Support::FitnessPlaneCalculator< N >](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Support](#)

7.15 trunk/libNormalizer/normalizer.cpp File Reference

```
#include <stdio.h> #include <stdlib.h> #include <iostream> x
#include <fstream>   #include "../include/PRPSEvolution.-h"
#include "../include/PRPSEvolutionGeneralExceptions.-h"   #include "../include/PRPSEvolutionFIOExceptions.h" x
#include <complex>  #include <array>  #include <random> x
#include <algorithm> #include "normalizer.h" Include dependency graph for normalizer.cpp:
```



Namespaces

- namespace [PRPSEvolution](#)

Functions

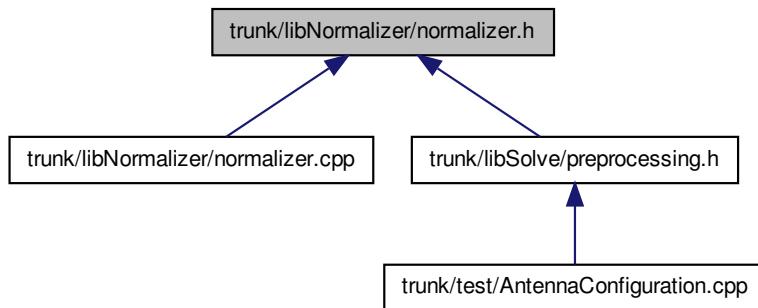
- `std::vector< std::array < double, 3 > > PRPSEvolution::rPoints ()`
- `std::vector< std::array < double, 3 > > PRPSEvolution::rCoords ()`
- template<std::size_t N, typename T>
`std::array< T, N > PRPSEvolution::idealPhaseFromInput (std::vector< std::array< double, 3 >> p, std::vector< std::array< double, 3 >> c, int SELECT_IDEAL_POINT, double LAMBDA)`

Variables

- const double [PRPSEvolution::pi](#) = 3.14159

7.16 trunk/libNormalizer/normalizer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Normalizer< N, T >](#)

Namespaces

- namespace [PRPSEvolution](#)

Enumerations

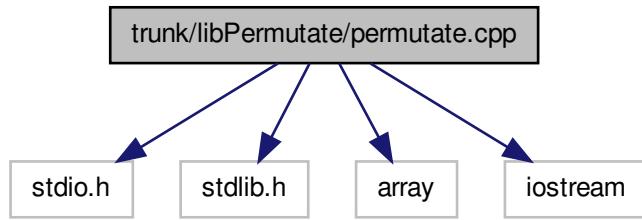
- enum [PRPSEvolution::NormalizationMethods](#) { [PRPSEvolution::Native](#), [PRPSEvolution::Ideal](#), [PRPSEvolution::CMPLX](#), [PRPSEvolution::RND](#), [PRPSEvolution::Complex](#) }

7.16.1 Detailed Description

Contains the 'Normalizer' class that is able to perform the normalization of input data. Also implements different methods of normalization, such as: -Random -Complex -Ideal -Native

7.17 trunk/libPermute/permute.cpp File Reference

```
#include <stdio.h> #include <stdlib.h> #include <array>
#include <iostream> Include dependency graph for permute.cpp:
```



Functions

- void [test2 \(\)](#)

7.17.1 Function Documentation

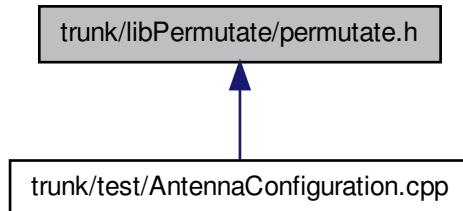
7.17.1.1 void test2()

7.18 trunk/libPermute/permute.h File Reference

```
#include <stdio.h> #include <stdlib.h> #include <iterator>
#include <iostream> #include <algorithm> #include <array>
#include <string> #include "../include/coords.h" #include
"../include/PRPSEvolution.h" #include "../include/PRPS-
EvolutionPermutationExceptions.h" #include "../include/-
PRPSEvolutionFIOExceptions.h" #include "../include/PRPS-
Error.h" #include "../libPRPSSystem/prpsevolutionsystem.-
h" #include "nr3/nr3.h" #include "nr3/svd.h" Include dependency
graph for permute.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct `PRPSEvolution::Permute::AntennaPermutations< N_MAT, T >`
- struct `PRPSEvolution::Permute::permuteAntennas< N_ANTA, N_ANTPERM, T >`

Namespaces

- namespace `PRPSEvolution`
- namespace `PRPSEvolution::Permute`

Functions

- int `PRPSEvolution::Permute::Factorial` (int x)
- template<typename Iterator>
bool `PRPSEvolution::Permute::next_combination` (const Iterator first, Iterator k, const Iterator last)

Variables

- const int `PRPSEvolution::Permute::MAX_PERMUTATION_AMOUNT` = 35

7.18.1 Detailed Description

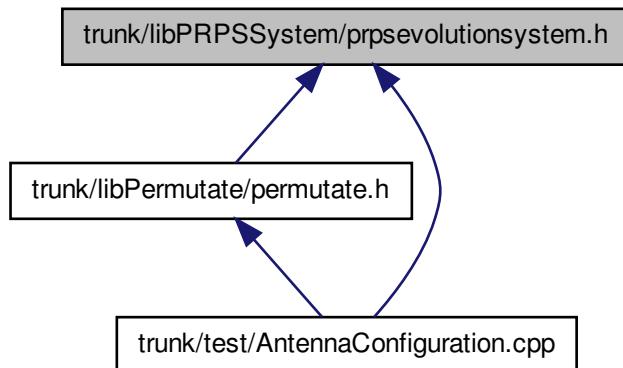
Date

2013|Jun|25

7.19 trunk/libPRPSSystem/prpsevolutionsystem.cpp File Reference

7.20 trunk/libPRPSSystem/prpsevolutionsystem.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Constants](#)
- struct [PRPSEvolution::System](#)

Namespaces

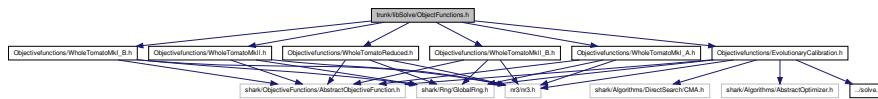
- namespace [PRPSEvolution](#)

7.21 trunk/libSolve/ObjectFunctions.cpp File Reference

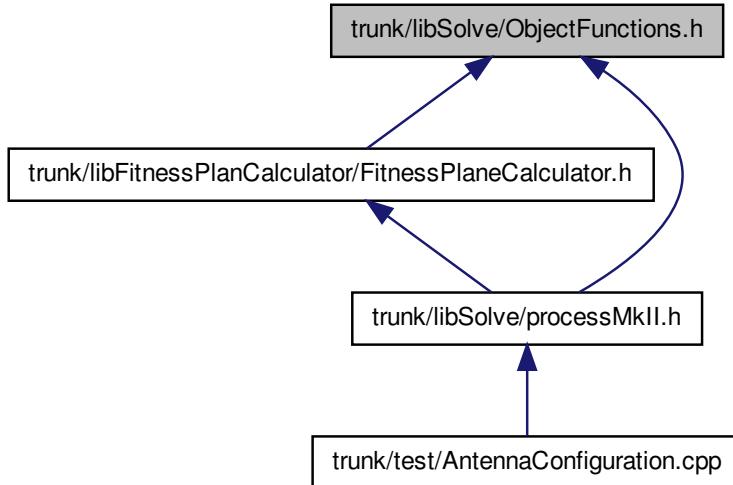
7.22 trunk/libSolve/ObjectFunctions.h File Reference

```
#include "Objectivefunctions/EvolutionaryCalibration.h" x
#include "Objectivefunctions/WholeTomatoMkII.h" #include
"Objectivefunctions/WholeTomatoReduced.h" #include "-"
Objectivefunctions/WholeTomatoMkII_B.h" #include "Objectivefunctions/-
```

```
WholeTomatoMkI_A.h" #include "Objectivefunctions/Whole-
TomatoMkI_B.h" Include dependency graph for ObjectFunctions.h:
```



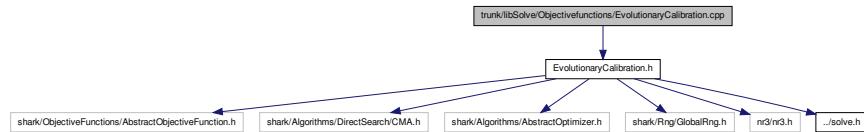
This graph shows which files directly or indirectly include this file:



7.23 trunk/libSolve/Objectivefunctions/EvolutionaryCalibration.cpp File Reference

```
#include "EvolutionaryCalibration.h" Include dependency graph for -
```

EvolutionaryCalibration.cpp:

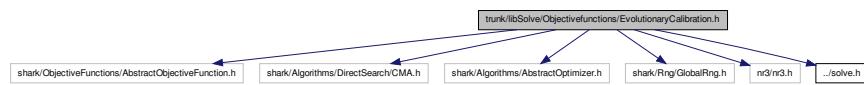


Namespaces

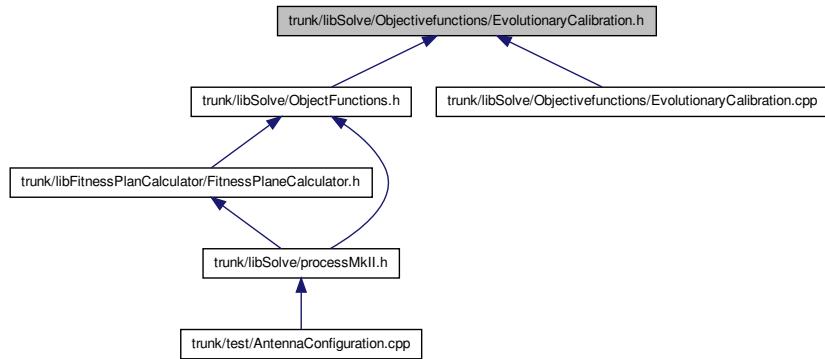
- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

7.24 trunk/libSolve/Objectivefunctions/EvolutionaryCalibration.h File Reference

```
#include <shark/ObjectiveFunctions/AbstractObjective-
Function.h>      #include <shark/Algorithms/DirectSearch/C-
MA.h>      #include <shark/Algorithms/AbstractOptimizer.h> ×
#include <shark/Rng/GlobalRng.h>      #include <nr3/nr3.h> ×
#include "../solve.h" Include dependency graph for EvolutionaryCalibration.-h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Models::EvolutionaryCalibration](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

Functions

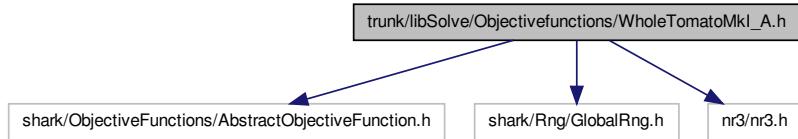
- [PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) (-
EvolutionaryCalibration, soo::RealValuedObjectiveFunctionFactory)

7.25 trunk/libSolve/Objectivefunctions/WholeTomatoMkl.cpp File - Reference

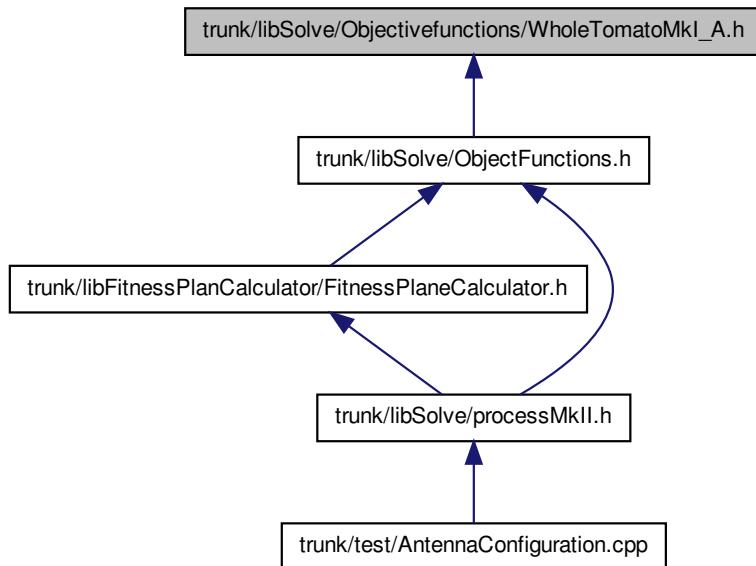
7.26 trunk/libSolve/Objectivefunctions/WholeTomatoMkl_A.h File - Reference

```
#include <shark/ObjectiveFunctions/AbstractObjective-
Function.h>    #include <shark/Rng/GlobalRng.h>    #include
```

<nr3/nr3.h> Include dependency graph for WholeTomatoMkl_A.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::WholeTomatoMkl_A](#)

Namespaces

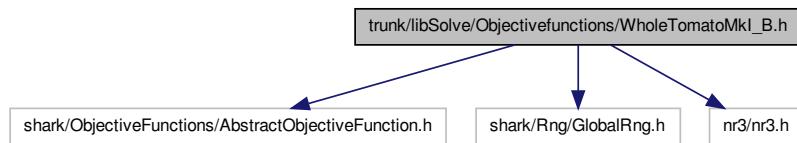
- namespace [PRPSEvolution](#)

Functions

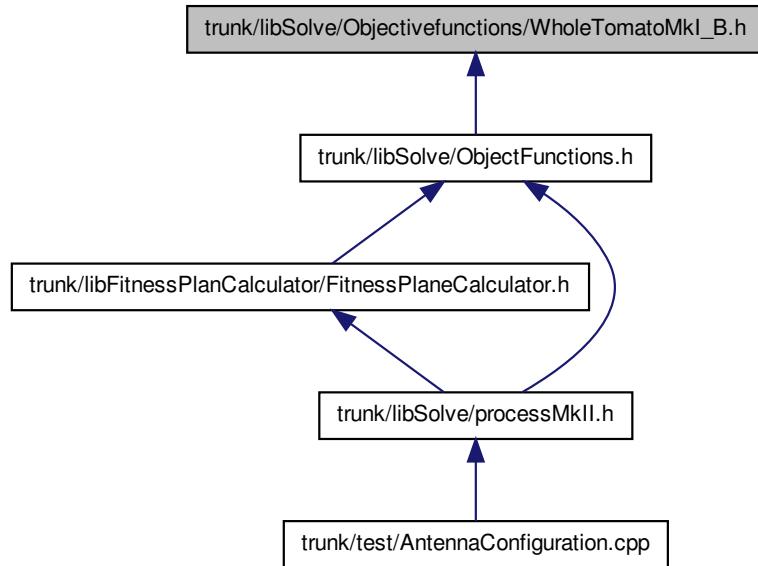
- PRPSEvolution::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION (WholeTomatoMkl_A, shark::soo::RealValuedObjectiveFunctionFactory)

7.27 trunk/libSolve/Objectivefunctions/WholeTomatoMkl_B.h File - Reference

```
#include <shark/ObjectiveFunctions/AbstractObjectiveFunction.h> #include <shark/Rng/GlobalRng.h> #include <nr3/nr3.h> Include dependency graph for WholeTomatoMkl_B.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::WholeTomatoMkI_B](#)

Namespaces

- namespace [PRPSEvolution](#)

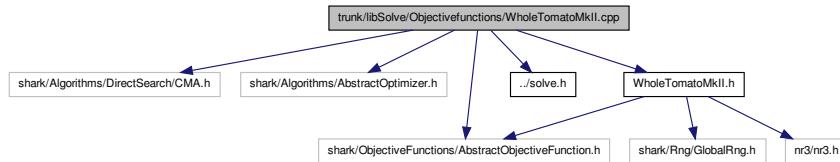
Functions

- [PRPSEvolution::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) (WholeTomatoMkI_B, shark::soo::RealValuedObjectiveFunctionFactory)

7.28 trunk/libSolve/Objectivefunctions/WholeTomatoMkII.cpp File - Reference

```
#include <shark/Algorithms/DirectSearch/CMA.h>      #include
<shark/Algorithms/AbstractOptimizer.h>  #include <shark/-
```

```
ObjectiveFunctions/AbstractObjectiveFunction.h> #include
"../solve.h" #include "WholeTomatoMkII.h" Include dependency
graph for WholeTomatoMkII.cpp:
```

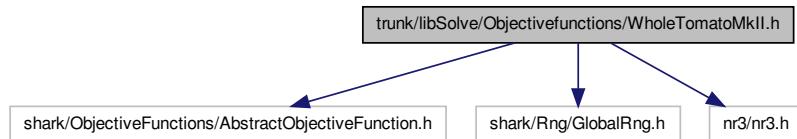


Namespaces

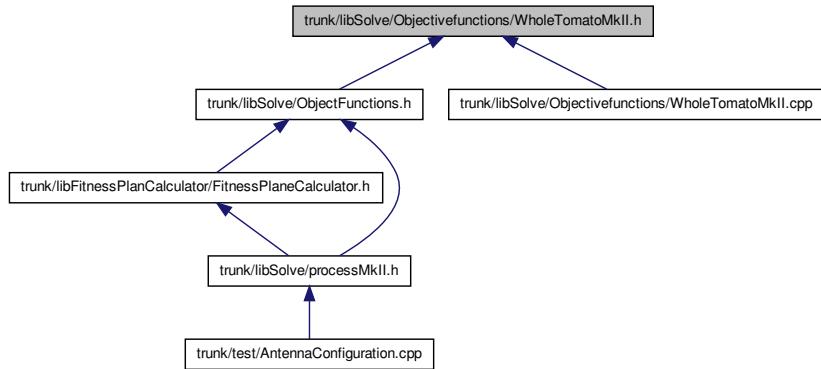
- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

7.29 trunk/libSolve/Objectivefunctions/WholeTomatoMkII.h File - Reference

```
#include <shark/ObjectiveFunctions/AbstractObjective-
Function.h> #include <shark/Rng/GlobalRng.h> #include
<nr3/nr3.h> Include dependency graph for WholeTomatoMkII.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Models::WholeTomatoMkII](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

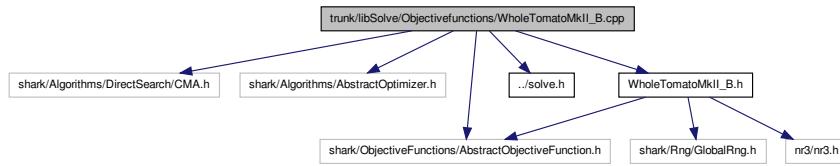
Functions

- [PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) (-
WholeTomatoMkII, sooo::RealValuedObjectiveFunctionFactory)

7.30 trunk/libSolve/Objectivefunctions/WholeTomatoMkII_B.cpp - File Reference

```
#include <shark/Algorithms/DirectSearch/CMA.h> #include
<shark/Algorithms/AbstractOptimizer.h> #include <shark/-
ObjectiveFunctions/AbstractObjectiveFunction.h> #include
"../solve.h" #include "WholeTomatoMkII_B.h" Include dependency
```

graph for WholeTomatoMkII_B.cpp:

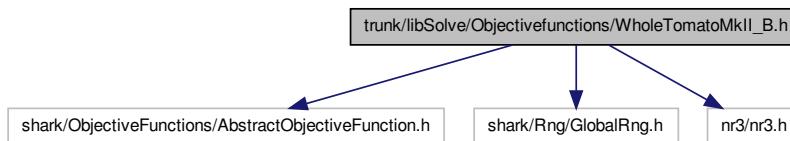


Namespaces

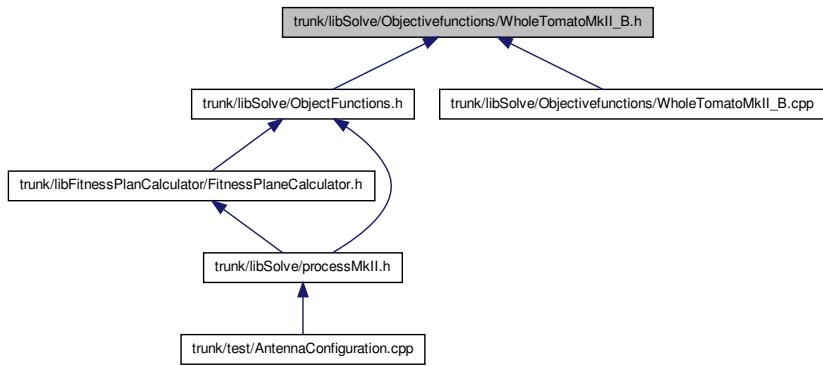
- namespace PRPSEvolution
- namespace PRPSEvolution::Models

7.31 trunk/libSolve/Objectivefunctions/WholeTomatoMkII_B.h File - Reference

```
#include <shark/ObjectiveFunctions/AbstractObjective-
Function.h>    #include <shark/Rng/GlobalRng.h>    #include
<nr3/nr3.h> Include dependency graph for WholeTomatoMkII_B.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Models::WholeTomatoMkII_B](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

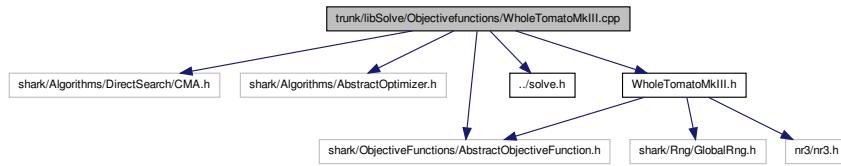
Functions

- [PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) (-
WholeTomatoMkII_B, soo::RealValuedObjectiveFunctionFactory)

7.32 trunk/libSolve/Objectivefunctions/WholeTomatoMkIII.cpp File Reference

```
#include <shark/Algorithms/DirectSearch/CMA.h> #include
<shark/Algorithms/AbstractOptimizer.h> #include <shark/-
ObjectiveFunctions/AbstractObjectiveFunction.h> #include
"../solve.h" #include "WholeTomatoMkIII.h" Include dependency
```

graph for WholeTomatoMkIII.cpp:

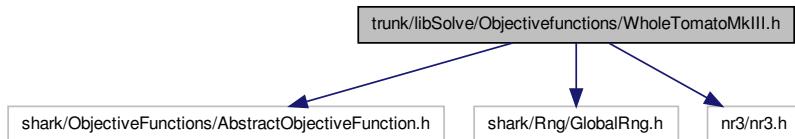


Namespaces

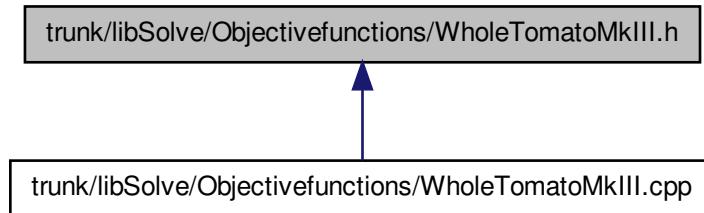
- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

7.33 trunk/libSolve/Objectivefunctions/WholeTomatoMkIII.h File - Reference

```
#include <shark/ObjectiveFunctions/AbstractObjective-
Function.h> #include <shark/Rng/GlobalRng.h> #include
<nr3/nr3.h> Include dependency graph for WholeTomatoMkIII.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Models::WholeTomatoMkIII](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

Functions

- [PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) (-
WholeTomatoMkIII, sooo::RealValuedObjectiveFunctionFactory)

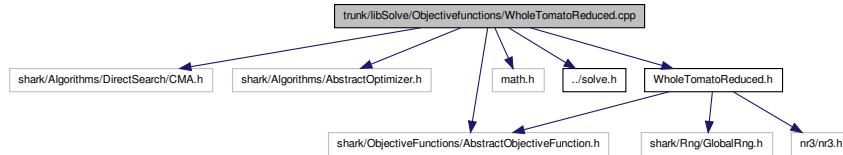
7.34 trunk/libSolve/Objectivefunctions/WholeTomatoReduced.cpp

File Reference

```
#include <shark/Algorithms/DirectSearch/CMA.h>    #include
<shark/Algorithms/AbstractOptimizer.h>  #include <shark/-
ObjectiveFunctions/AbstractObjectiveFunction.h>  #include
<math.h>      #include "../solve.h"    #include "WholeTomato-
```

7.35 trunk/libSolve/Objectivefunctions/WholeTomatoReduced.h File Reference

Reduced.h" Include dependency graph for WholeTomatoReduced.cpp:



Namespaces

- namespace PRPSEvolution
- namespace PRPSEvolution::Models

Defines

- #define _WT_CONSTRAIN_HARD_

Variables

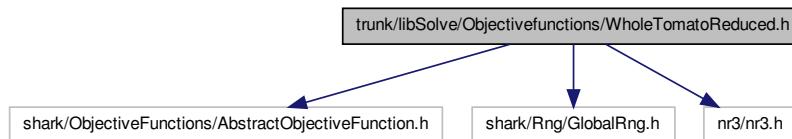
- bool PRPSEvolution::Models::printYN = false

7.34.1 Define Documentation

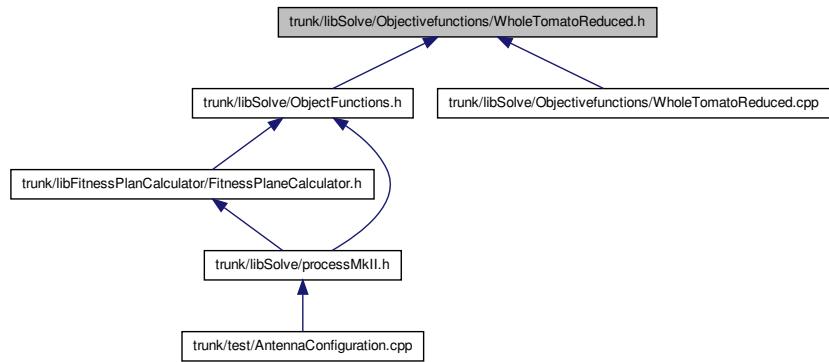
7.34.1.1 #define _WT_CONSTRAIN_HARD_

7.35 trunk/libSolve/Objectivefunctions/WholeTomatoReduced.h - File Reference

```
#include <shark/ObjectiveFunctions/AbstractObjective-
Function.h>  #include <shark/Rng/GlobalRng.h>  #include
<nr3/nr3.h> Include dependency graph for WholeTomatoReduced.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Models::WholeTomatoReduced](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Models](#)

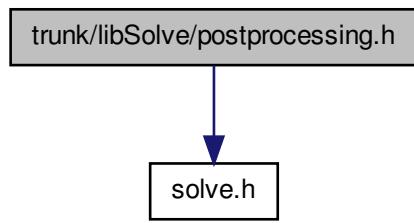
Functions

- [PRPSEvolution::Models::ANNOUNCE_SINGLE_OBJECTIVE_FUNCTION](#) (-
WholeTomatoReduced, soo::RealValuedObjectiveFunctionFactory)

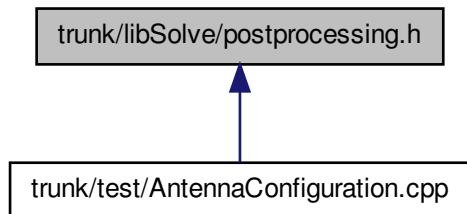
7.36 trunk/libSolve/postprocessing.cpp File Reference

7.37 trunk/libSolve/postprocessing.h File Reference

#include "solve.h" Include dependency graph for postprocessing.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [PRPSEvolution::Solve::PostProcessing](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Solve](#)

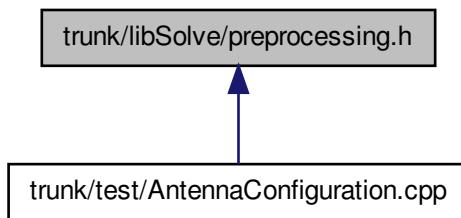
7.38 trunk/libSolve/preprocessing.cpp File Reference

7.39 trunk/libSolve/preprocessing.h File Reference

```
#include <iostream> #include <sstream> #include <string>
#include <stdio.h> #include <stdlib.h> #include <iterator>
#include <algorithm> #include <array> #include "../lib-
Normalizer/normalizer.h" #include "../include/coords.h"
#include "../include/PRPSEvolution.h" #include "../include/-_
PRPSEvolutionSolveExceptions.h" #include "../include/-_
PRPSEvolutionFIOExceptions.h" #include "../include/PR-
PSEvolutionGeneralExceptions.h" #include "nr3/nr3.h"
#include "solve.h" Include dependency graph for preprocessing.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_-Measure >`

Namespaces

- namespace `PRPSEvolution`
- namespace `PRPSEvolution::Solve`

7.40 trunk/libSolve/process.cpp File Reference

7.41 trunk/libSolve/process.h File Reference

```
#include <iostream> #include <string> #include <random>
#include <stdio.h> #include <chrono> #include <stdlib.h>
#include <iterator> #include <algorithm> #include <array>
#include "nr3/nr3.h" #include <Shark2.3/EALib/-ChromosomeCMA.h> #include <Shark2.3/SharkDefs.h> #include <Shark2.3/EALib/PopulationT.h> #include <Shark2.3/EALib/-ObjectiveFunction.h> #include <Shark2.3/EALib/Population.h> #include <Shark2.3/EALib/CMA.h> #include "../include/-PRPSEvolutionSolveExceptions.h" #include "../include/P-PRPSEvolutionFIOExceptions.h" #include "../include/PRPS-EvolutionGeneralExceptions.h" #include "solveresult.h" ×
#include "solve.h" #include "ueber9000.h" Include dependency graph for process.h:
```



Classes

- class [PRPSEvolution::Solve::Process](#)

Namespaces

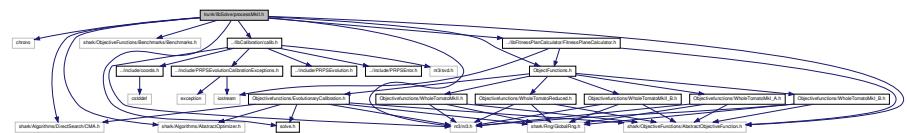
- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Solve](#)

7.42 trunk/libSolve/processMkII.cpp File Reference

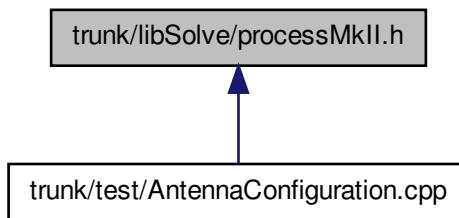
7.43 trunk/libSolve/processMkII.h File Reference

```
#include <chrono>      #include <shark/Algorithms/Direct-Search/CMA.h> #include <shark/Algorithms/AbstractOptimizer.h> #include <shark/ObjectiveFunctions/Benchmarks/Benchmarks.h> #include <shark/ObjectiveFunctions/AbstractObjective-Function.h> #include <nr3/nr3.h> #include "solve.h" ×
#include "../libCalibration/calib.h" #include "Object-
```

Functions.h" #include "../libFitnessPlanCalculator/FitnessPlaneCalculator.h" Include dependency graph for processMkII.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [PRPSEvolution::Solve::Process_MkII](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Solve](#)

Defines

- #define [STUFF](#)(Function, Vars)
- #define [SOLVE](#)(MODEL)
- #define [SOLVE_AND_WRITE](#)(MODEL)

7.43.1 Define Documentation

7.43.1.1 #define SOLVE(MODEL)

Value:

```
shark::CMA cma;
 \
    cma.init( MODEL );
 \
    do { cma.step( MODEL ); } while(cma.solution().value > epsilon );
 \

```

7.43.1.2 #define SOLVE_AND_WRITE(MODEL)

Value:

```
shark::CMA cma;
 \
    cma.init( MODEL );
 \
    do {
 \
        cma.step( MODEL );
 \
        f << model.evaluationCounter() << " "
 \
                                << cma.solution().value << " "
 \
                                << cma.solution().point << " "
 \
                                << cma.sigma()
 \
                                << std::endl;
 \
    } while(cma.solution().value > epsilon
 \
           && model.evaluationCounter() < maxEvaluations);
 \

```

7.43.1.3 #define STUFF(Function, Vars)

Value:

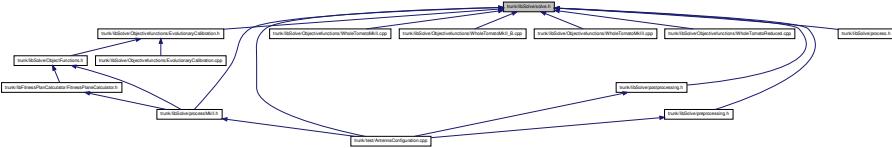
```
Function model(Vars);
 \
    model.setNumberOfVariables(Vars);
 \
    shark::CMA cma;
 \
    cma.init( model );
 \
    do { cma.step( model ); } while(cma.solution().value > epsilon );
 \

```

7.44 trunk/libSolve/solve.cpp File Reference

7.45 trunk/libSolve/solve.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct PRPSEvolution::Solve::ProblemDimensions

Namespaces

- namespace PRPSEvolution
 - namespace PRPSEvolution::Solve

Enumerations

- enum PRPSEvolution::Solve::SelectBy { PRPSEvolution::Solve::ConditionNumber, PRPSEvolution::Solve::Random, PRPSEvolution::Solve::AllPossible, PRPSEvolution::Solve::Best10ByCN, PRPSEvolution::Solve::AllFrom4Ant }
 - enum PRPSEvolution::Solve::ESStrategy { PRPSEvolution::Solve::OnePlusOne, PRPSEvolution::Solve::MuPlusLambda, PRPSEvolution::Solve::MuCommaLambda, PRPSEvolution::Solve::MuCommaLambda_MkII, PRPSEvolution::Solve::MuPlusLambda_MkII, PRPSEvolution::Solve::CMA_ES_MkI, PRPSEvolution::Solve::CMA_ES_MkII }
 - enum PRPSEvolution::Solve::Models { PRPSEvolution::Solve::WholeTomatoMkI, PRPSEvolution::Solve::WholeTomatoMkII, PRPSEvolution::Solve::TestSphere }

Functions

- double PRPSEvolution::Solve::meanFromVector (std::vector< double > &res)

Variables

- const int PRPSEvolution::Solve::nConfigsForProcessing = 1

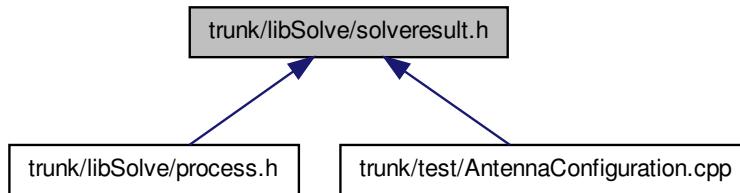
7.45.1 Detailed Description

Date

2013|Jun|25

7.46 trunk/libSolve/solveresult.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Solve::solveresult_t< T_Store1, T_Store2, T_Return >](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Solve](#)

7.46.1 Detailed Description

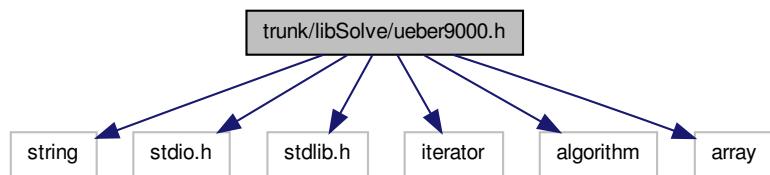
Date

2013|Jul|5

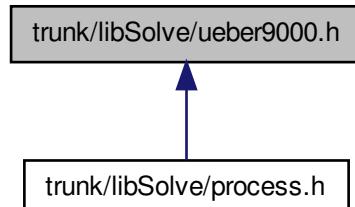
7.47 trunk/libSolve/ueber9000.cpp File Reference

7.48 trunk/libSolve/ueber9000.h File Reference

```
#include <string> #include <stdio.h> #include <stdlib.h>
#include <iterator> #include <algorithm> #include <array>
Include dependency graph for ueber9000.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [PRPSEvolution::Solve::Ueber9000< T >](#)

Namespaces

- namespace [PRPSEvolution](#)
- namespace [PRPSEvolution::Solve](#)

Variables

- std::mutex PRPSEvolution::Solve::wMutex
- int PRPSEvolution::Solve::_i_ = 0

7.49 trunk/test/AntennaConfiguration.cpp File Reference

```
#include <stdio.h> #include <stdlib.h> #include <math.h>
#include <array> #include <iostream> #include <exception>
#include <fstream> #include <sstream> #include <string>
#include <chrono> #include <future> #include <thread>
#include <vector> #include "../libSolve/processMkII.h"
#include "../include/PRPSEvolution.h" #include "../include/-/
PRPSError.h" #include "../include/PRPSEvolutionGeneral-
Exceptions.h" #include "AntennaConfiguration.h" #include
"../libPermute/permute.h" #include "../libPRPSSystem/prpsevolutionsystem.-
h" #include "../libCalibration/calib.h" #include "../lib-
Solve/solve.h" #include "../libSolve/solveresult.h" #include
"../libSolve/preprocessing.h" #include "../libSolve/postprocessing.-
h" Include dependency graph for AntennaConfiguration.cpp:
```



Defines

- #define _USE_SHARK_3_0_
- #define _Write_Result
- #define _DROP_BAD_
- #define _REFINE_SELECTION
- #define _PP_FORM_GROUPS
- #define _USE_IDEAL_INPUT
- #define _Write_SOLUTION_STATISTICS
- #define _WT_CONSTRAIN_HARD_
- #define USAGE_AND_EXIT

Functions

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

Variables

- const int EXPECTED = 9

- const int SOLUTION_AMOUNT = 1
- const int DEFAULT_MAX_EVALUATIONS = 4500
- int VARIANT_SW
- int NO_OF_SOLUTIONS
- int MU = 0
- int LAMBDA = 0
- int UseNMats = 1
- int EVALUATIONS = DEFAULT_MAX_EVALUATIONS
- int MatOffset = 0
- int PointOffset = 0
- bool DROPBAD = false
- bool IDEAL_DATA = false
- std::string FILENAME = ""

7.49.1 Detailed Description

This File contains the `main()` of the AntennaApp-Project

7.49.2 Define Documentation

7.49.2.1 `#define _DROP_BAD_`

7.49.2.2 `#define _PP_FORM_GROUPS`

7.49.2.3 `#define _REFINE_SELECTION`

7.49.2.4 `#define _USE_IDEAL_INPUT`

7.49.2.5 `#define _USE_SHARK_3_0_`

7.49.2.6 `#define _Write_Result`

7.49.2.7 `#define _Write SOLUTION_STATISTICS`

7.49.2.8 `#define _WT_CONSTRAIN_HARD_`

7.49.2.9 `#define USAGE_AND_EXIT`

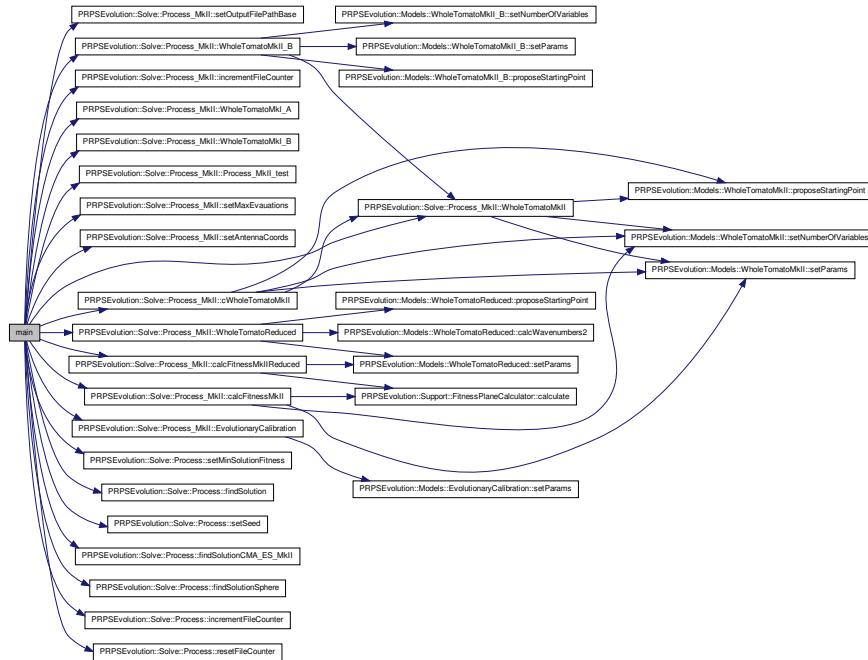
Value:

```
{
  \
    std::cout << "USAGE: " << std::endl << "\t " << argv[0]
  \
    << " [VARIANT_SW] [NO_OF_SOLUTIONS] [DROPBAD] [FILENAME] [MU] [Lambda]"
    [UseNMats] [DEFAULT_MAX_EVALUATIONS]"<< std::endl; \
    exit(-1);
  \
}
```

7.49.3 Function Documentation

7.49.3.1 int main (int argc, char * argv[])

Here is the call graph for this function:



7.49.4 Variable Documentation

7.49.4.1 const int DEFAULT_MAX_EVALUATIONS = 4500

7.49.4.2 bool DROPBAD = false

7.49.4.3 int EVALUATIONS = DEFAULT_MAX_EVALUATIONS

7.49.4.4 const int EXPECTED = 9

7.49.4.5 std::string FILENAME = ""

7.49.4.6 bool IDEAL_DATA = false

7.49.4.7 int LAMBDA = 0

7.49.4.8 int MatOffset = 0

7.49.4.9 int MU = 0

7.49.4.10 int NO_OF_SOLUTIONS

7.49.4.11 int PointOffset = 0

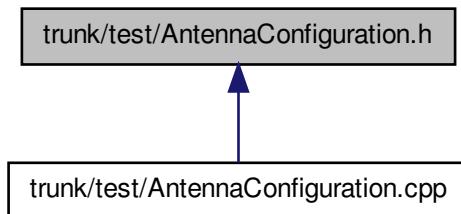
7.49.4.12 const int SOLUTION_AMOUNT = 1

7.49.4.13 int UseNMats = 1

7.49.4.14 int VARIANT_SW

7.50 trunk/test/AntennaConfiguration.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define VERSION_MAJOR 0
- #define VERSION_MINOR 1
- #define VERSION_SUB_MINOR 1

7.50.1 Define Documentation

7.50.1.1 #define VERSION_MAJOR 0

7.50.1.2 #define VERSION_MINOR 1

7.50.1.3 #define VERSION_SUB_MINOR 1