

WARP

Generated by Doxygen 1.6.1

Fri Oct 16 09:13:52 2015

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	test Namespace Reference	7
4.1.1	Variable Documentation	8
4.1.1.1	assemblyname	8
4.1.1.2	dens_fuel	8
4.1.1.3	dens_water	8
4.1.1.4	filename	8
4.1.1.5	fracs_fuel	8
4.1.1.6	fracs_water	8
4.1.1.7	geom	8
4.1.1.8	hist	8
4.1.1.9	idx	8
4.1.1.10	material	8
4.1.1.11	maxs	8
4.1.1.12	mins	8
4.1.1.13	n_topes	8
4.1.1.14	origin	8

4.1.1.15	prim_id	8
4.1.1.16	tallycell	8
4.1.1.17	tallyname	8
4.1.1.18	topes	8
4.1.1.19	typ	8
4.2	unionize Namespace Reference	9
5	Class Documentation	11
5.1	unionize::cross_section_data Class Reference	11
5.1.1	Detailed Description	13
5.1.2	Member Function Documentation	14
5.1.2.1	__init__	14
5.1.2.2	_add_isotope	14
5.1.2.3	_allocate_arrays	14
5.1.2.4	_get_awr_pointer	14
5.1.2.5	_get_energy_data	15
5.1.2.6	_get_length_numbers_pointer	15
5.1.2.7	_get_main_Egrid_pointer	15
5.1.2.8	_get_MT_array_pointer	15
5.1.2.9	_get_MT_numbers_pointer	16
5.1.2.10	_get_MT_numbers_total_pointer	16
5.1.2.11	_get_Q_pointer	16
5.1.2.12	_get_scattering_data	16
5.1.2.13	_get_temp_pointer	17
5.1.2.14	_init_from_string	17
5.1.2.15	_insert_reactions	17
5.1.2.16	_interpolate	17
5.1.2.17	_print_isotopes	17
5.1.2.18	_read_tables	18
5.1.2.19	_resolve_library	18
5.1.2.20	_unionize	18
5.1.3	Member Data Documentation	18
5.1.3.1	awr	18
5.1.3.2	datapath	18

5.1.3.3	isotope_list	18
5.1.3.4	last_loaded	18
5.1.3.5	libraries	18
5.1.3.6	MT_array	19
5.1.3.7	MT_E_grid	19
5.1.3.8	num_isotopes	19
5.1.3.9	num_main_E	19
5.1.3.10	num_reactions	19
5.1.3.11	Q	19
5.1.3.12	reaction_numbers	19
5.1.3.13	reaction_numbers_total	19
5.1.3.14	tables	19
5.1.3.15	temp	19
5.1.3.16	xsdirstring	20
5.2	geom_data Struct Reference	21
5.2.1	Detailed Description	21
5.2.2	Member Data Documentation	21
5.2.2.1	cellnum	21
5.2.2.2	is_fissile	21
5.2.2.3	loc	21
5.2.2.4	matnum	21
5.2.2.5	max	21
5.2.2.6	min	22
5.3	intersection_point Struct Reference	23
5.3.1	Detailed Description	23
5.3.2	Member Data Documentation	23
5.3.2.1	cell	23
5.3.2.2	fiss	23
5.3.2.3	mat	23
5.3.2.4	norm	23
5.3.2.5	sense	23
5.3.2.6	surf_dist	24
5.3.2.7	x	24
5.3.2.8	y	24

5.3.2.9	<code>z</code>	24
5.4	<code>material_def</code> Struct Reference	25
5.4.1	Detailed Description	25
5.4.2	Member Data Documentation	25
5.4.2.1	<code>density</code>	25
5.4.2.2	<code>fractions</code>	25
5.4.2.3	<code>id</code>	25
5.4.2.4	<code>is_fissile</code>	25
5.4.2.5	<code>isotopes</code>	25
5.4.2.6	<code>matnum</code>	26
5.4.2.7	<code>num_isotopes</code>	26
5.5	<code>optix_stuff</code> Class Reference	27
5.5.1	Detailed Description	28
5.5.2	Constructor & Destructor Documentation	29
5.5.2.1	<code>optix_stuff</code>	29
5.5.2.2	<code>optix_stuff</code>	29
5.5.2.3	<code>~optix_stuff</code>	29
5.5.3	Member Function Documentation	29
5.5.3.1	<code>get_outer_cell</code>	29
5.5.3.2	<code>get_outer_cell_type</code>	29
5.5.3.3	<code>get_rand</code>	29
5.5.3.4	<code>init</code>	29
5.5.3.5	<code>init_internal</code>	30
5.5.3.6	<code>make_color</code>	30
5.5.3.7	<code>make_geom_prim</code>	31
5.5.3.8	<code>make_geom_xform</code>	31
5.5.3.9	<code>make_geom_xform_common</code>	31
5.5.3.10	<code>print</code>	31
5.5.3.11	<code>set_trace_type</code>	32
5.5.3.12	<code>trace</code>	32
5.5.3.13	<code>trace</code>	32
5.5.3.14	<code>trace</code>	32
5.5.4	Member Data Documentation	32
5.5.4.1	<code>accel_type</code>	32

5.5.4.2	boundary_condition	32
5.5.4.3	cellnum_ptr	32
5.5.4.4	compute_device	33
5.5.4.5	context	33
5.5.4.6	done_ptr	33
5.5.4.7	GEOM_FLAG	33
5.5.4.8	matnum_ptr	33
5.5.4.9	maxcell	33
5.5.4.10	mincell	33
5.5.4.11	N	33
5.5.4.12	n_materials	33
5.5.4.13	optix_device	33
5.5.4.14	outer_cell	34
5.5.4.15	outer_cell_dims	34
5.5.4.16	outer_cell_type	34
5.5.4.17	positions_ptr	34
5.5.4.18	remap_ptr	34
5.5.4.19	rxn_ptr	34
5.5.4.20	stack_size_multiplier	34
5.5.4.21	traverse_type	34
5.6	primitive Class Reference	35
5.6.1	Detailed Description	35
5.6.2	Constructor & Destructor Documentation	35
5.6.2.1	primitive	35
5.6.2.2	primitive	36
5.6.2.3	~primitive	36
5.6.3	Member Function Documentation	36
5.6.3.1	add_transform	36
5.6.3.2	add_transform	36
5.6.3.3	add_transform	37
5.6.3.4	make_hex_array	37
5.6.3.5	make_hex_array	37
5.6.3.6	print_transform	38
5.6.3.7	print_transform	38

5.6.4	Member Data Documentation	38
5.6.4.1	location	38
5.6.4.2	material	38
5.6.4.3	max	38
5.6.4.4	min	38
5.6.4.5	n_transforms	38
5.6.4.6	num_primitives	38
5.6.4.7	primitive_id	38
5.6.4.8	transforms	39
5.6.4.9	type	39
5.7	qnode Struct Reference	40
5.7.1	Detailed Description	40
5.7.2	Member Data Documentation	40
5.7.2.1	leaves	40
5.7.2.2	values	40
5.8	qnode_host Struct Reference	41
5.8.1	Detailed Description	41
5.8.2	Member Data Documentation	41
5.8.2.1	cuda_pointer	41
5.8.2.2	node	41
5.9	source_point Struct Reference	42
5.9.1	Detailed Description	42
5.9.2	Member Data Documentation	42
5.9.2.1	enforce_BC	42
5.9.2.2	macro_t	42
5.9.2.3	norm	42
5.9.2.4	surf_dist	42
5.9.2.5	weight	43
5.9.2.6	x	43
5.9.2.7	xhat	43
5.9.2.8	y	43
5.9.2.9	yhat	43
5.9.2.10	z	43
5.9.2.11	zhat	43

5.10 wfloat3 Class Reference	44
5.10.1 Detailed Description	45
5.10.2 Constructor & Destructor Documentation	45
5.10.2.1 wfloat3	45
5.10.2.2 wfloat3	45
5.10.2.3 wfloat3	45
5.10.3 Member Function Documentation	45
5.10.3.1 cross	45
5.10.3.2 dot	46
5.10.3.3 norm2	46
5.10.3.4 operator*	46
5.10.3.5 operator*	46
5.10.3.6 operator+	47
5.10.3.7 operator+	47
5.10.3.8 operator-	47
5.10.3.9 operator-	47
5.10.3.10 operator/	48
5.10.3.11 rodrigues_rotation	48
5.10.3.12 rotate	48
5.10.4 Member Data Documentation	48
5.10.4.1 x	48
5.10.4.2 y	48
5.10.4.3 z	48
5.11 wgeometry Class Reference	49
5.11.1 Detailed Description	50
5.11.2 Constructor & Destructor Documentation	50
5.11.2.1 wgeometry	50
5.11.2.2 ~wgeometry	50
5.11.3 Member Function Documentation	51
5.11.3.1 add_material	51
5.11.3.2 add_primitive	51
5.11.3.3 add_primitive	51
5.11.3.4 add_transform	51
5.11.3.5 add_transform	51

5.11.3.6	add_transform	51
5.11.3.7	check	51
5.11.3.8	check_fissile	51
5.11.3.9	delete_primitive	52
5.11.3.10	delete_transform	52
5.11.3.11	get_boundary_condition	52
5.11.3.12	get_material_count	52
5.11.3.13	get_material_table	52
5.11.3.14	get_maximum_cell	52
5.11.3.15	get_maximum_material	53
5.11.3.16	get_minimum_cell	53
5.11.3.17	get_minimum_material	53
5.11.3.18	get_outer_cell	53
5.11.3.19	get_outer_cell_dims	53
5.11.3.20	get_outer_cell_type	53
5.11.3.21	get_primitive_count	53
5.11.3.22	get_transform_count	54
5.11.3.23	make_hex_array	54
5.11.3.24	make_material_table	54
5.11.3.25	print_all	54
5.11.3.26	print_materials_table	54
5.11.3.27	print_summary	54
5.11.3.28	set_datapath	54
5.11.3.29	set_outer_cell	54
5.11.3.30	update	55
5.11.4	Member Data Documentation	55
5.11.4.1	awr_list	55
5.11.4.2	boundary_condition	55
5.11.4.3	cell_num_list	55
5.11.4.4	concentrations_matrix	55
5.11.4.5	datapath	55
5.11.4.6	fissile_flag	55
5.11.4.7	isotope_list	55
5.11.4.8	isotope_list_array	55

5.11.4.9	isotopes	56
5.11.4.10	material_list_array	56
5.11.4.11	material_num_list	56
5.11.4.12	materials	56
5.11.4.13	n_box	56
5.11.4.14	n_cyl	56
5.11.4.15	n_hex	56
5.11.4.16	n_isotopes	56
5.11.4.17	n_materials	56
5.11.4.18	n_primitives	56
5.11.4.19	n_sph	56
5.11.4.20	n_transforms	57
5.11.4.21	outer_cell	57
5.11.4.22	primitives	57
5.12	whistory Class Reference	58
5.12.1	Detailed Description	65
5.12.2	Constructor & Destructor Documentation	65
5.12.2.1	whistory	65
5.12.2.2	~whistory	65
5.12.3	Member Function Documentation	65
5.12.3.1	accumulate_keff	65
5.12.3.2	accumulate_tally	65
5.12.3.3	bin_fission_points	65
5.12.3.4	converge	66
5.12.3.5	copy_data_to_device	66
5.12.3.6	create_quad_tree	66
5.12.3.7	device_report	66
5.12.3.8	get_time	66
5.12.3.9	hot2	66
5.12.3.10	init	66
5.12.3.11	init_CUDPP	66
5.12.3.12	init_host	67
5.12.3.13	init_RNG	67
5.12.3.14	load_cross_sections	67

5.12.3.15	make_color	67
5.12.3.16	map_active	67
5.12.3.17	nonzero	67
5.12.3.18	plot_geom	68
5.12.3.19	prep_secondaries	68
5.12.3.20	print_materials_table	68
5.12.3.21	print_pointers	68
5.12.3.22	print_xs_data	68
5.12.3.23	reduce_done	68
5.12.3.24	reduce_weight	68
5.12.3.25	reduce_yield	69
5.12.3.26	remap_active	69
5.12.3.27	reset_cycle	69
5.12.3.28	reset_fixed	69
5.12.3.29	run	70
5.12.3.30	sample_fissile_points	70
5.12.3.31	set_acceration	70
5.12.3.32	set_device	70
5.12.3.33	set_dump_level	70
5.12.3.34	set_filename	70
5.12.3.35	set_print_level	71
5.12.3.36	set_run_param	71
5.12.3.37	set_run_type	71
5.12.3.38	set_run_type	71
5.12.3.39	set_tally_cell	71
5.12.3.40	trace	72
5.12.3.41	trace	72
5.12.3.42	update_RNG	72
5.12.3.43	write_fission_points	72
5.12.3.44	write_histories	72
5.12.3.45	write_results	72
5.12.3.46	write_tally	73
5.12.3.47	write_to_file	73
5.12.3.48	write_to_file	73

5.12.3.49	write_to_file	73
5.12.3.50	write_to_file	74
5.12.3.51	write_xs_data	74
5.12.4	Member Data Documentation	74
5.12.4.1	accel_type	74
5.12.4.2	awr_list	74
5.12.4.3	blks	74
5.12.4.4	cellnum	74
5.12.4.5	compact_config	74
5.12.4.6	compactplan	74
5.12.4.7	compute_device	75
5.12.4.8	d_active	75
5.12.4.9	d_awr_list	75
5.12.4.10	d_bank_E	75
5.12.4.11	d_bank_space	75
5.12.4.12	d_cellnum	75
5.12.4.13	d_completed	75
5.12.4.14	d_done	75
5.12.4.15	d_E	75
5.12.4.16	d_edges	75
5.12.4.17	d_fissile_energy	75
5.12.4.18	d_fissile_points	76
5.12.4.19	d_index	76
5.12.4.20	d_isonum	76
5.12.4.21	d_isotope_list	76
5.12.4.22	d_mask	76
5.12.4.23	d_material_list	76
5.12.4.24	d_matnum	76
5.12.4.25	d_num_active	76
5.12.4.26	d_num_completed	76
5.12.4.27	d_number_density_matrix	76
5.12.4.28	d_Q	76
5.12.4.29	d_qnodes_root	77
5.12.4.30	d_reduced_done	77

5.12.4.31 d_reduced_weight	77
5.12.4.32 d_reduced_yields	77
5.12.4.33 d_remap	77
5.12.4.34 d_rn_bank	77
5.12.4.35 d_rxn	77
5.12.4.36 d_rxn_remap	77
5.12.4.37 d_scanned	77
5.12.4.38 d_space	77
5.12.4.39 d_tally_count	77
5.12.4.40 d_tally_score	78
5.12.4.41 d_tally_square	78
5.12.4.42 d_temp_list	78
5.12.4.43 d_valid_N	78
5.12.4.44 d_valid_result	78
5.12.4.45 d_weight	78
5.12.4.46 d_xs_data_energy	78
5.12.4.47 d_xs_data_main_E_grid	78
5.12.4.48 d_xs_data_MT	78
5.12.4.49 d_xs_data_Q	78
5.12.4.50 d_xs_data_scatter	78
5.12.4.51 d_xs_length_numbers	79
5.12.4.52 d_xs_MT_numbers	79
5.12.4.53 d_xs_MT_numbers_total	79
5.12.4.54 d_yield	79
5.12.4.55 d_zeros	79
5.12.4.56 done	79
5.12.4.57 dump_flag	79
5.12.4.58 E	79
5.12.4.59 edges	79
5.12.4.60 filename	79
5.12.4.61 fiss_hash_table_handle	79
5.12.4.62 fiss_img	80
5.12.4.63 fones	80
5.12.4.64 hash_config	80

5.12.4.65 index	80
5.12.4.66 is_initialized	80
5.12.4.67 isonum	80
5.12.4.68 isotopes	80
5.12.4.69 keff2_sum	80
5.12.4.70 keff_err	80
5.12.4.71 keff_sum	80
5.12.4.72 mate_hash_table_handle	80
5.12.4.73 matnum	80
5.12.4.74 MT_columns	81
5.12.4.75 MT_rows	81
5.12.4.76 N	81
5.12.4.77 n_cycles	81
5.12.4.78 n_edges	81
5.12.4.79 n_isotopes	81
5.12.4.80 n_materials	81
5.12.4.81 n_qnodes	81
5.12.4.82 n_skip	81
5.12.4.83 n_tally	81
5.12.4.84 Ndataset	81
5.12.4.85 NUM_THREADS	82
5.12.4.86 number_density_matrix	82
5.12.4.87 ones	82
5.12.4.88 outer_cell	82
5.12.4.89 outer_cell_dims	82
5.12.4.90 outer_cell_type	82
5.12.4.91 print_flag	82
5.12.4.92 problem_geom	82
5.12.4.93 Q	82
5.12.4.94 qnodes	82
5.12.4.95 qnodes_depth	82
5.12.4.96 qnodes_width	83
5.12.4.97 radix_config	83
5.12.4.98 radixplan	83

5.12.4.99	rand_gen	83
5.12.4.100	redu_float_config	83
5.12.4.101	redu_int_config	83
5.12.4.102	reduced_weight	83
5.12.4.103	reduced_weight_total	83
5.12.4.104	reduced_yields	83
5.12.4.105	reduced_yields_total	83
5.12.4.106	reduplan_float	83
5.12.4.107	reduplan_int	84
5.12.4.108	remap	84
5.12.4.109	res	84
5.12.4.110	rn_bank	84
5.12.4.111	IRNUM_PER_THREAD	84
5.12.4.112	RUN_FLAG	84
5.12.4.113	rxn	84
5.12.4.114	scan_int_config	84
5.12.4.115	scanplan_int	84
5.12.4.116	space	84
5.12.4.117	stream	84
5.12.4.118	tally_cell	85
5.12.4.119	tally_count	85
5.12.4.120	tally_count_total	85
5.12.4.121	tally_score	85
5.12.4.122	tally_score_total	85
5.12.4.123	tally_square	85
5.12.4.124	tally_square_total	85
5.12.4.125	temp_list	85
5.12.4.126	theCudpp	85
5.12.4.127	total_bytes_energy	85
5.12.4.128	total_bytes_scatter	85
5.12.4.129	weight	86
5.12.4.130	xs_data_energy	86
5.12.4.131	xs_data_energy_host	86
5.12.4.132	xs_data_main_E_grid	86

5.12.4.133	xs_data_MT	86
5.12.4.134	xs_data_Q	86
5.12.4.135	xs_data_scatter	86
5.12.4.136	xs_data_scatter_host	86
5.12.4.137	xs_isotope_ints	86
5.12.4.138	xs_length_numbers	86
5.12.4.139	xs_MT_numbers	87
5.12.4.140	xs_MT_numbers_total	87
5.12.4.141	xs_num_rxns	87
5.12.4.142	yield	87
5.12.4.143	zeros	87
5.13	wtransform Struct Reference	88
5.13.1	Detailed Description	88
5.13.2	Member Data Documentation	88
5.13.2.1	cellmat	88
5.13.2.2	cellnum	88
5.13.2.3	dx	88
5.13.2.4	dy	88
5.13.2.5	dz	88
5.13.2.6	phi	89
5.13.2.7	theta	89
6	File Documentation	91
6.1	benchmarks.cpp File Reference	91
6.1.1	Function Documentation	91
6.1.1.1	main	91
6.2	binary_search.h File Reference	92
6.2.1	Function Documentation	92
6.2.1.1	binary_search	92
6.3	datadef.h File Reference	93
6.4	device_copies.h File Reference	94
6.4.1	Function Documentation	94
6.4.1.1	allocate_on_device	94
6.4.1.2	copy_from_device	94

6.4.1.3	copy_to_device	94
6.4.1.4	deallocate_on_device	94
6.5	main.cpp File Reference	95
6.5.1	Function Documentation	95
6.5.1.1	main	95
6.6	mt19937ar.cpp File Reference	96
6.6.1	Define Documentation	97
6.6.1.1	LOWER_MASK	97
6.6.1.2	M	97
6.6.1.3	MATRIX_A	97
6.6.1.4	N	97
6.6.1.5	UPPER_MASK	97
6.6.2	Function Documentation	97
6.6.2.1	genrand_int31	97
6.6.2.2	genrand_int32	97
6.6.2.3	genrand_real1	97
6.6.2.4	genrand_real2	97
6.6.2.5	genrand_real3	97
6.6.2.6	genrand_res53	97
6.6.2.7	init_by_array	97
6.6.2.8	init_genrand	97
6.6.3	Variable Documentation	97
6.6.3.1	mt	97
6.6.3.2	mti	97
6.7	optix_stuff.cpp File Reference	98
6.8	optix_stuff.h File Reference	99
6.9	optixtest.cpp File Reference	100
6.9.1	Function Documentation	100
6.9.1.1	main	100
6.10	primitive.cpp File Reference	101
6.11	primitive.h File Reference	102
6.12	print_banner.cpp File Reference	103
6.12.1	Function Documentation	103
6.12.1.1	print_banner	103

6.13	test.py File Reference	104
6.14	unionize.py File Reference	105
6.15	warp.h File Reference	106
6.16	warp_cuda.h File Reference	107
6.16.1	Function Documentation	110
6.16.1.1	absorb	110
6.16.1.2	check_remap	110
6.16.1.3	copy_points	110
6.16.1.4	cscatter	110
6.16.1.5	escatter	110
6.16.1.6	find_E_grid_index	110
6.16.1.7	find_E_grid_index_quad	110
6.16.1.8	fission	110
6.16.1.9	iscatter	110
6.16.1.10	macroscopic	110
6.16.1.11	microscopic	110
6.16.1.12	pop_secondaries	110
6.16.1.13	pop_source	110
6.16.1.14	print_banner	110
6.16.1.15	print_data	110
6.16.1.16	print_histories	110
6.16.1.17	reaction_edges	110
6.16.1.18	rebase_yield	110
6.16.1.19	sample_fission_spectra	110
6.16.1.20	sample_fixed_source	110
6.16.1.21	set_positions_rand	110
6.16.1.22	tally_spec	110
6.16.1.23	write_to_file	110
6.17	warp_header.h File Reference	111
6.18	wfloat3.h File Reference	112
6.19	wgeometry.cpp File Reference	113
6.20	wgeometry.h File Reference	114
6.21	whistory.cpp File Reference	115
6.21.1	Define Documentation	116

6.21.1.1	check_cuda	116
6.21.2	Function Documentation	116
6.21.2.1	check_cuda	116
6.21.3	Variable Documentation	116
6.21.3.1	optix_obj	116
6.22	whistory.h File Reference	117

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

test	7
unionize	9

Chapter 2

Class Index

2.1 Class List

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

unionize::cross_section_data (Handles cross section data)	11
geom_data (Parameters of a wgeometry)	21
intersection_point (Information pertinent to an intersection point)	23
material_def (Information that defines a material)	25
optix_stuff (OptiX stuff class)	27
primitive (Primitive class)	35
qnode (Quaternary search node)	40
qnode_host (Quaternary search node host container)	41
source_point (Parameters of the neutron source point)	42
wfloat3 (Class definitions for device vector operations)	44
wgeometry (Wgeometry class)	49
whistory (Whistory class)	58
wtransform (Parameters of a wtransform)	88

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

benchmarks.cpp	91
binary_search.h	92
datadef.h	93
device_copies.h	94
main.cpp	95
mt19937ar.cpp	96
optix_stuff.cpp	98
optix_stuff.h	99
optixtest.cpp	100
primitive.cpp	101
primitive.h	102
print_banner.cpp	103
test.py	104
unionize.py	105
warp.h	106
warp_cuda.h	107
warp_header.h	111
wfloat3.h	112
wgeometry.cpp	113
wgeometry.h	114
whistory.cpp	115
whistory.h	117

Chapter 4

Namespace Documentation

4.1 test Namespace Reference

Variables

- tuple `geom` = `warp.wgeometry()`
- int `n_topes` = 4
- int `prim_id` = 0
- string `assemblyname` = "assembly"
- tuple `topes` = `warp.Unsigned(n_topes)`
- tuple `fracs_fuel` = `warp.Float(n_topes)`
- tuple `fracs_water` = `warp.Float(n_topes)`
- tuple `mins` = `warp.Float(3)`
- tuple `maxs` = `warp.Float(3)`
- tuple `origin` = `warp.Float(3)`
- int `dens_fuel` = 15
- int `dens_water` = 3
- int `tallycell` = 316
- `filename` = `assemblyname`
- `tallyname` = `assemblyname`
- int `typ` = 1
- int `material` = 1
- tuple `idx` = `geom.add_transform(prim_id,999,0,0,0,0,0)`
- tuple `hist` = `warp.whistory(100000,geom)`

4.1.1 Variable Documentation

4.1.1.1 `string test::assemblyname = "assembly"`

4.1.1.2 `int test::dens_fuel = 15`

4.1.1.3 `int test::dens_water = 3`

4.1.1.4 `test::filename = assemblyname`

4.1.1.5 `tuple test::fracs_fuel = warp.Float(n_topes)`

4.1.1.6 `tuple test::fracs_water = warp.Float(n_topes)`

4.1.1.7 `tuple test::geom = warp.wgeometry()`

4.1.1.8 `tuple test::hist = warp.whistory(100000,geom)`

4.1.1.9 `tuple test::idx = geom.add_transform(prim_id,999,0,0,0,0)`

4.1.1.10 `int test::material = 1`

4.1.1.11 `tuple test::maxs = warp.Float(3)`

4.1.1.12 `tuple test::mins = warp.Float(3)`

4.1.1.13 `int test::n_topes = 4`

4.1.1.14 `tuple test::origin = warp.Float(3)`

4.1.1.15 `tuple test::prim_id = 0`

4.1.1.16 `int test::tallycell = 316`

4.1.1.17 `string test::tallyname = assemblyname`

4.1.1.18 `tuple test::topes = warp.Unsigned(n_topes)`

4.1.1.19 `int test::typ = 1`

4.2 unionize Namespace Reference

Classes

- class [cross_section_data](#)
handles cross section data

Chapter 5

Class Documentation

5.1 unionize::cross_section_data Class Reference

handles cross section data

Public Member Functions

- [def __init__](#)
initialization function

Public Attributes

- [num_isotopes](#)
number of isotopes
- [isotope_list](#)
isotope list
- [datapath](#)
data path
- [tables](#)
cross section tables
- [libraries](#)
cross section libraries
- [awr](#)
AWR array.

- [temp](#)
temp array
- [Q](#)
Q-value array.
- [num_main_E](#)
main energy
- [reaction_numbers](#)
reaction numbers array
- [reaction_numbers_total](#)
total reaction numbers array
- [num_reactions](#)
number of reactions
- [MT_E_grid](#)
MT energy grid.
- [MT_array](#)
MT number array.
- [last_loaded](#)
Last valid table loaded.
- [xsdirstring](#)

Private Member Functions

- [def _init_from_string](#)
initializes material from isotope list string
- [def _add_isotope](#)
appends the input isotope to the input material's list of isotopes
- [def _read_tables](#)
reads in cross section tables
- [def _resolve_library](#)
- [def _unionize](#)
unionization function
- [def _insert_reactions](#)
insert reactions function

- [def _allocate_arrays](#)
array allocation function
- [def _interpolate](#)
interpolation function
- [def _get_MT_numbers_pointer](#)
gets pointer to MT numbers
- [def _get_awr_pointer](#)
gets pointer to AWR values
- [def _get_temp_pointer](#)
gets pointer to temperature values
- [def _get_Q_pointer](#)
gets pointer to Q-values
- [def _get_MT_array_pointer](#)
gets pointer to MT numbers
- [def _get_main_Egrid_pointer](#)
gets pointer to main energy grid
- [def _get_length_numbers_pointer](#)
creates array of size number of isotopes + main energy grid + number of reactions
- [def _get_MT_numbers_total_pointer](#)
gets pointer to total MT numbers
- [def _print_isotopes](#)
prints list of isotopes in a material
- [def _get_scattering_data](#)
gets table of scattering data
- [def _get_energy_data](#)
gets table of energy data

5.1.1 Detailed Description

handles cross section data

5.1.2 Member Function Documentation

5.1.2.1 `def unionize::cross_section_data::_init_ (self)`

initialization function initializes number of isotopes to zero; isotope list as an empty array; temperature extension as '.03c'; tables, libraries, AWR list, and Q as empty arrays; main energy as zero; reaction numbers and total reaction numbers as empty arrays; number of reactions to zero. sets the MT energy grid # and array as empty.

Parameters:

← *self* - material to do cross section stuff about

5.1.2.2 `def unionize::cross_section_data::_add_isotope (self, isotope)` `[private]`

appends the input isotope the the input material's list of isotopes

Parameters:

← *self* - material to which to add isotope

← *isotope* - isotope to be appended

5.1.2.3 `def unionize::cross_section_data::_allocate_arrays (self)` `[private]`

array allocation function allocates a 2D array of size number of all reactions x number of energy points

Parameters:

← *self* - material to allocate arrays about

5.1.2.4 `def unionize::cross_section_data::_get_awr_pointer (self)` `[private]`

gets pointer to AWR values

Parameters:

← - material

Returns:

AWR_array - array of AWR values

5.1.2.5 def unionize::cross_section_data::_get_energy_data (*self*, *row*, *col*) [private]

gets table of energy data table returned in form of [nextDex, length, mu, cdf]

Parameters:

- ← *self* - isotope
- ← *row* - point in energy grid
- ← *col* - MT number

5.1.2.6 def unionize::cross_section_data::_get_length_numbers_pointer (*self*) [private]

creates array of size number of isotopes + main energy grid + number of reactions

Parameters:

- ← - material

Returns:

lengths - lengths array

5.1.2.7 def unionize::cross_section_data::_get_main_Egrid_pointer (*self*) [private]

gets pointer to main energy grid

Parameters:

- ← - material

Returns:

E_grid - array of energy grid points

5.1.2.8 def unionize::cross_section_data::_get_MT_array_pointer (*self*) [private]

gets pointer to MT numbers

Parameters:

- ← - material

Returns:

MT_array - array of MT numbers

**5.1.2.9 def unionize::cross_section_data::_get_MT_numbers_pointer (*self*)
[private]**

gets pointer to MT numbers

Parameters:

← *self* - material

Returns:

MT_num_array - array of MT numbers

5.1.2.10 def unionize::cross_section_data::_get_MT_numbers_total_pointer (*self*) [private]

gets pointer to total MT numbers

Parameters:

← - isotope

Returns:

numbers - array of total MT numbers

5.1.2.11 def unionize::cross_section_data::_get_Q_pointer (*self*) [private]

gets pointer to Q-values

Parameters:

← - material

Returns:

Q_array - array of Q-values

**5.1.2.12 def unionize::cross_section_data::_get_scattering_data (*self*, *row*, *col*)
[private]**

gets table of scattering data if scattering data exists, table returned in form of [nextDex, length, mu, cdf]

Parameters:

← *self* - isotope

← *row* - point in energy grid

← *col* - MT number

**5.1.2.13 def unionize::cross_section_data::_get_temp_pointer (*self*)
[private]**

gets pointer to temperature values

Parameters:

← - material

Returns:

temp_array - array of temperature values

**5.1.2.14 def unionize::cross_section_data::_init_from_string (*self*, *this_string*)
[private]**

initializes material from isotope list string

Parameters:

← *self* - material to initialize

← *this_string* - comma-separated isotope list

**5.1.2.15 def unionize::cross_section_data::_insert_reactions (*self*)
[private]**

insert reactions function appends ones to the front, appends the isotope's AWR to the table, appends the isotope's total reaction numbers to the table. appends all reaction numbers to the reaction list.

Parameters:

← *self* - isotope for reactions to be inserted

5.1.2.16 def unionize::cross_section_data::_interpolate (*self*) [private]

interpolation function linearly interpolates the cross sections for each isotope in a material

Parameters:

← *self* - material for which to interpolate cross sections

5.1.2.17 def unionize::cross_section_data::_print_isotopes (*self*) [private]

prints list of isotopes in a material

Parameters:

← *self* - material for which to print isotope list

5.1.2.18 `def unionize::cross_section_data::_read_tables (self, datapath_in) [private]`

reads in cross section tables for each isotope in the material, the acefile is appended to the library list, then all of the libraries are read in. the material's number of isotopes is set to how many libraries were retrieved.

Parameters:

← *self* - material to get cross sections for

5.1.2.19 `def unionize::cross_section_data::_resolve_library (self, tope) [private]`

5.1.2.20 `def unionize::cross_section_data::_unionize (self) [private]`

unionization function unionizes MT energy grid and scattering energies in if present.

Parameters:

← *self* - material with attributes to be unionized

5.1.3 Member Data Documentation

5.1.3.1 `unionize::cross_section_data::awr`

AWR array.

5.1.3.2 `unionize::cross_section_data::datapath`

data path

5.1.3.3 `unionize::cross_section_data::isotope_list`

isotope list

5.1.3.4 `unionize::cross_section_data::last_loaded`

Last valid table loaded.

5.1.3.5 `unionize::cross_section_data::libraries`

cross section libraries

5.1.3.6 unionize::cross_section_data::MT_array

MT number array.

5.1.3.7 unionize::cross_section_data::MT_E_grid

MT energy grid.

5.1.3.8 unionize::cross_section_data::num_isotopes

number of isotopes

5.1.3.9 unionize::cross_section_data::num_main_E

main energy

5.1.3.10 unionize::cross_section_data::num_reactions

number of reactions

5.1.3.11 unionize::cross_section_data::Q

Q-value array.

5.1.3.12 unionize::cross_section_data::reaction_numbers

reaction numbers array

5.1.3.13 unionize::cross_section_data::reaction_numbers_total

total reaction numbers array

5.1.3.14 unionize::cross_section_data::tables

cross section tables

5.1.3.15 unionize::cross_section_data::temp

temp array

5.1.3.16 unionize::cross_section_data::xsdirstring

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

- [unionize.py](#)

5.2 geom_data Struct Reference

contains parameters of a [wgeometry](#)

```
#include <datadef.h>
```

Public Attributes

- float [min](#) [3]
- float [max](#) [3]
- float [loc](#) [3]
- int [cellnum](#)
- int [matnum](#)
- int [is_fissile](#)

5.2.1 Detailed Description

contains parameters of a [wgeometry](#) extrema arrays, location array, cell and material numbers, fissile flag

5.2.2 Member Data Documentation

5.2.2.1 int geom_data::cellnum

cell number

5.2.2.2 int geom_data::is_fissile

fissile flag

5.2.2.3 float geom_data::loc[3]

array of coordinate (x,y,z) locations

5.2.2.4 int geom_data::matnum

material number

5.2.2.5 float geom_data::max[3]

array of coordinate (x,y,z) maxima

5.2.2.6 float geom_data::min[3]

array of coordinate (x,y,z) minima

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

- [datadef.h](#)

5.3 intersection_point Struct Reference

contains information pertinent to an intersection point

```
#include <datadef.h>
```

Public Attributes

- float [x](#)
- float [y](#)
- float [z](#)
- float [surf_dist](#)
- int [cell](#)
- int [mat](#)
- int [fiss](#)
- float [norm](#) [3]
- int [sense](#)

5.3.1 Detailed Description

contains information pertinent to an intersection point 3D cartesian coordinates of intersection point, distance to nearest surface, first cell potentially hit, continuity flag, hit buffer, and hit buffer index

5.3.2 Member Data Documentation

5.3.2.1 int intersection_point::cell

cell number

5.3.2.2 int intersection_point::fiss

fissile flag

5.3.2.3 int intersection_point::mat

material number

5.3.2.4 float intersection_point::norm[3]

most recent normal

5.3.2.5 int intersection_point::sense

most recent cell sense

5.3.2.6 float intersection_point::surf_dist

distance to nearest surface

5.3.2.7 float intersection_point::x

x-coordinate

5.3.2.8 float intersection_point::y

y-coordinate

5.3.2.9 float intersection_point::z

z-coordinate

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

- [datadef.h](#)

5.4 material_def Struct Reference

contains information that defines a material

```
#include <datadef.h>
```

Public Attributes

- unsigned [id](#)
- unsigned [matnum](#)
- unsigned [is_fissile](#)
- unsigned [num_isotopes](#)
- float [density](#)
- `std::vector< std::string >` [isotopes](#)
- float * [fractions](#)

5.4.1 Detailed Description

contains information that defines a material material ID, number, fissile flag, number of isotopes, density, isotope list, isotope fraction list

5.4.2 Member Data Documentation

5.4.2.1 float material_def::density

density [g/cc]

5.4.2.2 float* material_def::fractions

isotope fractions

5.4.2.3 unsigned material_def::id

material ID

5.4.2.4 unsigned material_def::is_fissile

fissile flag

5.4.2.5 std::vector<std::string> material_def::isotopes

isotope list

5.4.2.6 unsigned material_def::matnum

material number

5.4.2.7 unsigned material_def::num_isotopes

number of isotopes

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

- [datadef.h](#)

5.5 optix_stuff Class Reference

OptiX stuff class.

```
#include <optix_stuff.h>
```

Public Member Functions

- [optix_stuff](#) (unsigned, unsigned)
constructor
- [optix_stuff](#) ()
"default" constructor
- [~optix_stuff](#) ()
destructor
- void [init](#) ([wgeometry](#), unsigned, std::string)
initialization function
- void [trace](#) ()
creates a trace
- void [trace](#) (unsigned)
creates a trace
- void [trace](#) (unsigned, unsigned)
creates a trace
- void [set_trace_type](#) (unsigned)
sets trace type in the OptiX context,
- void [print](#) ()
prints a summary of OptiX information
- void [make_color](#) (float *, unsigned, unsigned, unsigned)
function to [test](#) tracing
- float [get_rand](#) ()
returns a random float
- unsigned [get_outer_cell](#) ()
returns the outermost cell
- unsigned [get_outer_cell_type](#) ()
returns the outermost cell type

Public Attributes

- CUdeviceptr [positions_ptr](#)
- CUdeviceptr [rxn_ptr](#)
- CUdeviceptr [done_ptr](#)
- CUdeviceptr [cellnum_ptr](#)
- CUdeviceptr [matnum_ptr](#)
- CUdeviceptr [remap_ptr](#)
- unsigned [stack_size_multiplier](#)
- unsigned [N](#)
- float [outer_cell_dims](#) [6]

Private Member Functions

- void [make_geom_xform](#) (wgeometry)
makes a geometry with a transform
- void [make_geom_xform_common](#) (wgeometry)
makes a geometry with a transform
- void [make_geom_prim](#) (wgeometry)
makes a [primitive](#) geometry (no transform)
- void [init_internal](#) (wgeometry, unsigned, std::string)
initializes internal variables needed for OptiX stuff

Private Attributes

- optix::Context [context](#)
- std::string [accel_type](#)
- std::string [traverse_type](#)
- unsigned [mincell](#)
- unsigned [maxcell](#)
- unsigned [outer_cell](#)
- unsigned [boundary_condition](#)
- unsigned [outer_cell_type](#)
- unsigned [n_materials](#)
- unsigned [compute_device](#)
- unsigned [optix_device](#)
- unsigned [GEOM_FLAG](#)

5.5.1 Detailed Description

OptiX stuff class.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 optix_stuff::optix_stuff (unsigned *Nin*, unsigned *mult*)

constructor sets stack size multiplier and number of histories.

Parameters:

← *Nin* - number of histories

← *mult* - stack size multiplier

5.5.2.2 optix_stuff::optix_stuff ()

"default" constructor empty

5.5.2.3 optix_stuff::~~optix_stuff ()

destructor

5.5.3 Member Function Documentation

5.5.3.1 unsigned optix_stuff::get_outer_cell ()

returns the outermost cell

Returns:

outer_cell - number of outermost cell

5.5.3.2 unsigned optix_stuff::get_outer_cell_type ()

returns the outermost cell type

Returns:

outer_cell_type - geometrical [primitive](#) type of the outermost cell

5.5.3.3 float optix_stuff::get_rand ()

returns a random float

5.5.3.4 void optix_stuff::init (wgeometry *problem_geom*, unsigned *compute_device_in*, std::string *accel_type_in*)

initialization function sets minimum and maximum cell numbers, gets material numbers, tries to initialize OptiX and throws an error if not.

Parameters:

- ← *problem_geom* - problem geometry
- ← *compute_device_in* - compute device to use (always 0)
- ← *accel_type_in* - acceleration type

5.5.3.5 void optix_stuff::init_internal (wgeometry *problem_geom*, unsigned *compute_device_in*, std::string *accel_type_in*) [private]

initializes internal variables needed for OptiX stuff sets compute device and acceleration type; sets geometry and image types; creates OptiX context; gets compute device information; sets up scene information; sets stack size; renders the buffers for particles, reactions, done flags, cell numbers, material numbers, and remaps; attaches all buffers to variables; gets CUDA pointers for buffer variables; creates programs for ray generation, exceptions, and misses; sets boundary condition for outer cell; sets trace type; sets the outer cell and gets its dimensions; creates all geometry instances; and validates and compiles the context.

Parameters:

- ← *problem_geom* - problem geometry
- ← *compute_device_in* - compute device to use (always zero)
- ← *accel_type_in* - acceleration type

5.5.3.6 void optix_stuff::make_color (float * *color*, unsigned *x*, unsigned *min*, unsigned *max*)

function to [test](#) tracing gets cell minima and maxima, randomizes starting positions, copies starting positions to a pointer, traces a plane to generate an image, copies the data to a local buffer, creates the images, makes the distribution random, copies the data to a pointer, executes and times the trace, and returns the time it took to do the trace.

Returns:

time_out - time taken to do the trace creates a color map

Parameters:

- ← *color* - color map
- ← *x* - used to check for a miss or normalize the color
- ← *min,max* - values used to normalize the color

5.5.3.7 void optix_stuff::make_geom_prim (wgeometry *problem_geom*) [private]

makes a [primitive](#) geometry (no transform) makes the top level group/acceleration as children of the top level object. for each [primitive](#) in the geometry, creates the geometry types, sets the intersection and bounding box programs, sets the hit programs to the geometry material, sets the program variables for the instance, creates the instances, sets cell-specific variables, makes the geometry group fro the [primitive](#), and puts the geometry instance into its group.

Parameters:

← *problem_geom* - problem geometry

5.5.3.8 void optix_stuff::make_geom_xform (wgeometry *problem_geom*) [private]

makes a geometry with a transform makes the top level group/acceleration as children of the top level object. for each [primitive](#) in the geometry, creates the geometry type, sets the intersection and bounding box programs, sets the hit programs to the geometry material, sets the program variables for the instance, creates the instances, sets cell-specific variables, makes the geometry group for the [primitive](#), puts the geometry instance into its corresponding group, makes any necessary transforms, and attaches to the root node.

Parameters:

← *problem_geom* - problem geometry

5.5.3.9 void optix_stuff::make_geom_xform_common (wgeometry *problem_geom*) [private]

makes a geometry with a transform makes the top level group/acceleration as children of the top level object. for each [primitive](#) in the geometry, creates the geometry type, sets the intersection and bounding box programs, sets the hit programs to the geometry material, sets the program variables for the instance, creates the instances, sets cell-specific variables, makes the geometry group for the [primitive](#), puts the geometry instance into its corresponding group, makes any necessary transforms, and attaches to the root node.

Parameters:

← *problem_geom* - problem geometry

5.5.3.10 void optix_stuff::print ()

prints a summary of OptiX information prints out instancing, image type, compute device, acceleration type, traverse type, stack size, and print buffer size.

5.5.3.11 void optix_stuff::set_trace_type (unsigned *trace_type*)

sets trace type in the OptiX context, sets trace type, 2=transport (finds nearest surface, normal, writes cell number and material number), 3=fissile query(writes fissile flag into material number, writes cell number), 4=geometry plot(same as 2, but misses are squelched, no normals/intersection distances reported)

5.5.3.12 void optix_stuff::trace (unsigned *trace_type*, unsigned *n_active*)

creates a trace sets the trace type, then launches the trace on the compute device with *n_active* histories.

Parameters:

- ← *trace_type* - trace type for OptiX context, 2=transport (finds nearest surface, normal, writes cell number and material number), 3=fissile query(writes fissile flag into material number, writes cell number), 4=geometry plot(same as 2, but misses are squelched, no normals/intersection distances reported)
- ← *n_active* - number of active histories

5.5.3.13 void optix_stuff::trace (unsigned *trace_type*)

creates a trace sets the trace type, then launches the trace on the compute device with *N* histories.

Parameters:

- ← *trace_type* - trace type for OptiX context

5.5.3.14 void optix_stuff::trace ()

creates a trace launches the trace on the compute device with *N* histories.

5.5.4 Member Data Documentation

5.5.4.1 std::string optix_stuff::accel_type [private]

acceleration type

5.5.4.2 unsigned optix_stuff::boundary_condition [private]

boundary condition of outermost cell

5.5.4.3 CUdeviceptr optix_stuff::cellnum_ptr

CUDA cell numbers pointer

5.5.4.4 unsigned optix_stuff::compute_device [private]

compute device number

5.5.4.5 optix::Context optix_stuff::context [private]

OptiX context

5.5.4.6 CUDeviceptr optix_stuff::done_ptr

CUDA done flags pointer

5.5.4.7 unsigned optix_stuff::GEOM_FLAG [private]

geometry flag: 0 = [primitive](#) instancing, 1 = transform instancing, 2 = transform instancing with common primitives

5.5.4.8 CUDeviceptr optix_stuff::matnum_ptr

CUDA material numbers pointer

5.5.4.9 unsigned optix_stuff::maxcell [private]

maximum (usually outermost) cell

5.5.4.10 unsigned optix_stuff::mincell [private]

minimum (usually innermost) cell

5.5.4.11 unsigned optix_stuff::N

number of histories

5.5.4.12 unsigned optix_stuff::n_materials [private]

number of materials

5.5.4.13 unsigned optix_stuff::optix_device [private]

optix device number, always zero since the optix device list should only have the specified cuda device in it

5.5.4.14 unsigned optix_stuff::outer_cell [private]

outermost cell

5.5.4.15 float optix_stuff::outer_cell_dims[6]

outermost cell dimensions

5.5.4.16 unsigned optix_stuff::outer_cell_type [private]

outermost cell type

5.5.4.17 CUdeviceptr optix_stuff::positions_ptr

CUDA positions pointer

5.5.4.18 CUdeviceptr optix_stuff::remap_ptr

CUDA remaps pointer

5.5.4.19 CUdeviceptr optix_stuff::rxn_ptr

CUDA reactions pointer

5.5.4.20 unsigned optix_stuff::stack_size_multiplier

stack size multiplier

5.5.4.21 std::string optix_stuff::traverse_type [private]

traverse type

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

- [optix_stuff.h](#)
- [optix_stuff.cpp](#)

5.6 primitive Class Reference

[primitive](#) class

```
#include <primitive.h>
```

Public Member Functions

- [primitive](#) ()
- [primitive](#) (int, unsigned, std::vector< float >, std::vector< float >, std::vector< float >)
- [~primitive](#) ()
- unsigned [add_transform](#) ()
- unsigned [add_transform](#) (unsigned, float, float, float, float, float)
- unsigned [add_transform](#) (unsigned, unsigned, float, float, float, float, float)
- void [print_transform](#) ()
- void [print_transform](#) (int)
- void [make_hex_array](#) (int, float, float, float, unsigned)
- void [make_hex_array](#) (int, float, float, float, float, unsigned)

Public Attributes

- float [min](#) [3]
- float [max](#) [3]
- float [location](#) [3]
- int [type](#)
- int [primitive_id](#)
- int [n_transforms](#)
- int [material](#)
- std::vector< [wtransform](#) > [transforms](#)

Static Public Attributes

- static int [num_primitives](#) = 0

5.6.1 Detailed Description

[primitive](#) class

5.6.2 Constructor & Destructor Documentation

5.6.2.1 [primitive::primitive](#) ()

box default constructor. sets all coordinate extrema to zero, sets location to origin, sets type and material to zero.

5.6.2.2 `primitive::primitive (int ptype, unsigned cellmat, std::vector< float > mins, std::vector< float > maxs, std::vector< float > locs)`

box valued constructor. sets coordinate extrema to input values, sets location to input values, sets material to input material, sets type to input type, creates a [wtransform](#), adds cell number and material to the [wtransform](#), adds the transform to the transform list.

Parameters:

- ← *p*type - [primitive](#) type
- ← *cellnum* - cell number
- ← *cellmat* - cell material
- ← *xmin,ymin,zmin* - coordinate minima
- ← *xmax,ymax,zmax* - coordinate maxima
- ← *x,y,z* - coordinate points

5.6.2.3 `primitive::~~primitive ()`

[primitive](#) destructor

5.6.3 Member Function Documentation

5.6.3.1 `unsigned primitive::add_transform (unsigned cellnum, unsigned cellmat, float dx, float dy, float dz, float theta, float phi)`

adds a transform. cell number is set to input value, material is set to input material, coordinates are set to input values, angles are set to input values.

Parameters:

- ← *cellnum* - cell number
- ← *cellmat* - cell material
- ← *dx,dy,dz* - transform coordinates
- ← *theta,phi* - transform azimuthal and polar angles, respectively

Returns:

index of the added transform

5.6.3.2 `unsigned primitive::add_transform (unsigned cellnum, float dx, float dy, float dz, float theta, float phi)`

adds a transform, defaults to [primitive](#) material. cell number is set to input value, coordinates are set to input values, angles are set to input values.

Parameters:

- ← *cellnum* - cell number
- ← *dx,dy,dz* - transform coordinates
- ← *theta,phi* - transform azimuthal and polar angles, respectively

Returns:

index of the added transform

5.6.3.3 unsigned primitive::add_transform ()

adds a "default" transform - all coordinate values and angles are zero.

Returns:

index of the added transform

5.6.3.4 void primitive::make_hex_array (int, float, float, float, float, unsigned)

creates a hexagonal array of elements.

Parameters:

- ← *n* - edge length
- ← *offsetx* - coordinate offsets in x
- ← *offsety* - coordinate offsets in y
- ← *PD_ratio* - pitch-to-diameter ratio
- ← *starting_index* - starting index

5.6.3.5 void primitive::make_hex_array (int *n*, float *x*, float *y*, float *PD_ratio*, unsigned *starting_index*)

creates a hexagonal array of elements.

Parameters:

- ← *n* - edge length
- ← *x,y* - coordinates
- ← *PD_ratio* - pitch-to-diameter ratio
- ← *starting_index* - starting index

5.6.3.6 void primitive::print_transform (int *tnum*)

prints out the properties of the input transform.

Parameters:

← *tnum* - transform number

5.6.3.7 void primitive::print_transform ()

prints [primitive](#) ID, coordinate extrema, location, type and material. for each transform, prints the number, cell number, cell material, transform coordinates, and transform angles.

5.6.4 Member Data Documentation**5.6.4.1 float primitive::location[3]**

coordinate location array

5.6.4.2 int primitive::material

material number

5.6.4.3 float primitive::max[3]

coordinate maxima array

5.6.4.4 float primitive::min[3]

coordinate minima array

5.6.4.5 int primitive::n_transforms

number of transforms

5.6.4.6 int primitive::num_primitives = 0 [static]

number of primitives

5.6.4.7 int primitive::primitive_id

[primitive](#) ID number

5.6.4.8 `std::vector<wtransform>` `primitive::transforms`

transform vector

5.6.4.9 `int` `primitive::type`

`primitive` type: 0 = box, 1 = cylinder, 2 = hexagon

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

- [primitive.h](#)
- [primitive.cpp](#)

5.7 qnode Struct Reference

quaternary search node

```
#include <datadef.h>
```

Public Attributes

- float [values](#) [4]
- [qnode](#) * [leaves](#) [4]

5.7.1 Detailed Description

quaternary search node

5.7.2 Member Data Documentation

5.7.2.1 [qnode](#)* [qnode::leaves](#)[4]

array of node leaves

5.7.2.2 float [qnode::values](#)[4]

array of values

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

- [datadef.h](#)

5.8 qnode_host Struct Reference

quaternary search node host container

```
#include <datadef.h>
```

Public Attributes

- [qnode * cuda_pointer](#)
- [qnode node](#)

5.8.1 Detailed Description

quaternary search node host container

5.8.2 Member Data Documentation

5.8.2.1 [qnode* qnode_host::cuda_pointer](#)

CUDA pointer

5.8.2.2 [qnode qnode_host::node](#)

quaternary node

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

- [datadef.h](#)

5.9 source_point Struct Reference

contains the parameters of the neutron source point

```
#include <datadef.h>
```

Public Attributes

- float [x](#)
- float [y](#)
- float [z](#)
- float [xhat](#)
- float [yhat](#)
- float [zhat](#)
- float [surf_dist](#)
- float [macro_t](#)
- float [norm](#) [3]
- unsigned [enforce_BC](#)
- unsigned [weight](#)

5.9.1 Detailed Description

contains the parameters of the neutron source point 3D Cartesian coordinates, shifted coordinates, distance to nearest surface, total macroscopic cross section, boundary condition enforcement flag

5.9.2 Member Data Documentation

5.9.2.1 unsigned source_point::enforce_BC

boundary condition enforcement flag

5.9.2.2 float source_point::macro_t

total macroscopic cross section

5.9.2.3 float source_point::norm[3]

5.9.2.4 float source_point::surf_dist

distance to nearest surface

5.9.2.5 unsigned source_point::weight**5.9.2.6 float source_point::x**

x-coordinate

5.9.2.7 float source_point::xhat

shifted x-coordinate

5.9.2.8 float source_point::y

y-coordinate

5.9.2.9 float source_point::yhat

shifted y-coordinate

5.9.2.10 float source_point::z

z-coordinate

5.9.2.11 float source_point::zhat

shifted z-coordinate

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

- [datadef.h](#)

5.10 wfloat3 Class Reference

class definitions for device vector operations

```
#include <wfloat3.h>
```

Public Member Functions

- `__device__ wfloat3 ()`
sets x,y,z to 0,0,0
- `__device__ wfloat3 (float)`
sets x,y,z to a,a,a
- `__device__ wfloat3 (float, float, float)`
sets x,y,z to a,b,c
- `__device__ wfloat3 operator+ (wfloat3)`
vector addition operator
- `__device__ wfloat3 operator- (wfloat3)`
vector subtraction operator
- `__device__ wfloat3 operator* (wfloat3)`
vector multiplication operator
- `__device__ wfloat3 operator+ (float)`
scalar addition operator
- `__device__ wfloat3 operator- (float)`
scalar subtraction operator
- `__device__ wfloat3 operator* (float)`
scalar multiplication operator
- `__device__ wfloat3 operator/ (float)`
scalar division operator
- `__device__ wfloat3 cross (wfloat3)`
cross product operator
- `__device__ float dot (wfloat3)`
dot product operator
- `__device__ void rodrigues_rotation (wfloat3, float)`
Rodrigues' rotation operator.

- `__device__ wfloat3 rotate (float, float)`
rotation about random cosine
- `__device__ float norm2 ()`
returns square root of sum of squares of coordinates

Public Attributes

- float `x`
- float `y`
- float `z`

5.10.1 Detailed Description

class definitions for device vector operations

5.10.2 Constructor & Destructor Documentation

5.10.2.1 `__device__ wfloat3::wfloat3 () [inline]`

sets x,y,z to 0,0,0

5.10.2.2 `__device__ wfloat3::wfloat3 (float a) [inline]`

sets x,y,z to a,a,a

Parameters:

← *a* - point to set

5.10.2.3 `__device__ wfloat3::wfloat3 (float a, float b, float c) [inline]`

sets x,y,z to a,b,c

Parameters:

← *a,b,c* - points to set

5.10.3 Member Function Documentation

5.10.3.1 `__device__ wfloat3 wfloat3::cross (wfloat3 arg) [inline]`

cross product operator returns the cross product of the vector and arg

Parameters:

← *arg* - vector to cross

Returns:

result - resultant [wfloat3](#)

5.10.3.2 __device__ float wfloat3::dot (wfloat3 *arg*) [inline]

dot product operator returns the dot product of the vector and *arg*

Parameters:

← *arg* - vector to dot

5.10.3.3 __device__ float wfloat3::norm2 () [inline]

returns square root of sum of squares of coordinates

5.10.3.4 __device__ wfloat3 wfloat3::operator* (float *arg*) [inline]

scalar multiplication operator multiplies *x* and *arg*, etc.

Parameters:

← *arg* - number by which to multiply

Returns:

result - resultant [wfloat3](#)

5.10.3.5 __device__ wfloat3 wfloat3::operator* (wfloat3 *arg*) [inline]

vector multiplication operator multiplies *x* and *x*-component of input [wfloat3](#), etc.

Parameters:

← *arg* - [wfloat3](#) coordinates to be multiplied

Returns:

result - resultant [wfloat3](#)

5.10.3.6 `__device__ wfloat3 wfloat3::operator+ (float arg) [inline]`

scalar addition operator adds x and arg, etc.

Parameters:

← *arg* - number to be added

Returns:

result - resultant [wfloat3](#)

5.10.3.7 `__device__ wfloat3 wfloat3::operator+ (wfloat3 arg) [inline]`

vector addition operator adds x and x-component of input [wfloat3](#), etc.

Parameters:

← *arg* - [wfloat3](#) coordinates to be added

Returns:

result - resultant [wfloat3](#)

5.10.3.8 `__device__ wfloat3 wfloat3::operator- (float arg) [inline]`

scalar subtraction operator subtracts arg from x, etc.

Parameters:

← *arg* - number to be subtracted

Returns:

result - resultant [wfloat3](#)

5.10.3.9 `__device__ wfloat3 wfloat3::operator- (wfloat3 arg) [inline]`

vector subtraction operator subtracts x-component of input [wfloat3](#) from x, etc.

Parameters:

← *arg* - [wfloat3](#) coordinates to be subtracted

Returns:

result - resultant [wfloat3](#)

5.10.3.10 `__device__ wfloat3 wfloat3::operator/ (float arg) [inline]`

scalar division operator divides *x* by *arg*, etc.

Parameters:

← *arg* - number by which to divide

Returns:

result - resultant [wfloat3](#)

5.10.3.11 `__device__ void wfloat3::rodrigues_rotation (wfloat3 k, float theta) [inline]`

Rodrigues' rotation operator. rotates a vector in space, given axis and angle of rotation

Parameters:

← *k* - unit vector describing axis of rotation about which to rotate

← *theta* - angle by which to rotate

5.10.3.12 `__device__ wfloat3 wfloat3::rotate (float mu, float rn) [inline]`

rotation about random cosine borrowed from OpenMC

Parameters:

← *mu* - random cos(theta)

← *rn* - random number

5.10.4 Member Data Documentation**5.10.4.1** `float wfloat3::x`

x-coordinate

5.10.4.2 `float wfloat3::y`

y-coordinate

5.10.4.3 `float wfloat3::z`

z-coordinate

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

- [wfloat3.h](#)

5.11 wgeometry Class Reference

wgeometry class

```
#include <wgeometry.h>
```

Public Member Functions

- [wgeometry](#) ()
- [~wgeometry](#) ()
- unsigned [get_minimum_cell](#) ()
- unsigned [get_maximum_cell](#) ()
- unsigned [get_minimum_material](#) ()
- unsigned [get_maximum_material](#) ()
- unsigned [get_primitive_count](#) ()
- unsigned [get_transform_count](#) ()
- unsigned [add_primitive](#) ()
- unsigned [add_primitive](#) (int, unsigned, std::vector< float >, std::vector< float >, std::vector< float >)
- void [update](#) ()
- void [print_summary](#) ()
- void [print_all](#) ()
- void [set_outer_cell](#) (unsigned, unsigned)
- unsigned [get_outer_cell](#) ()
- unsigned [get_boundary_condition](#) ()
- unsigned [get_outer_cell_type](#) ()
- void [add_material](#) (unsigned, unsigned, unsigned, float, std::vector< std::string >, std::vector< float >)
- int [check](#) ()
- unsigned [get_outer_cell_dims](#) (float *)
- unsigned [get_material_count](#) ()
- void [make_material_table](#) ()
- void [get_material_table](#) (unsigned *, unsigned *, float **)
- void [print_materials_table](#) ()
- void [set_datapath](#) (std::string)
- unsigned [check_fissile](#) ()
- unsigned [add_transform](#) (unsigned)
- unsigned [add_transform](#) (unsigned, unsigned, float, float, float, float, float)
- unsigned [add_transform](#) (unsigned, unsigned, unsigned, float, float, float, float, float)
- void [make_hex_array](#) (unsigned, int, float, float, float, unsigned)
- void [delete_primitive](#) (unsigned)
- void [delete_transform](#) (unsigned, unsigned)

Public Attributes

- unsigned [n_box](#)
- unsigned [n_cyl](#)
- unsigned [n_hex](#)
- unsigned [n_sph](#)
- unsigned [n_primitives](#)
- unsigned [n_transforms](#)
- unsigned [outer_cell](#)
- unsigned [n_materials](#)
- unsigned [n_isotopes](#)
- unsigned [fissile_flag](#)
- unsigned [boundary_condition](#)
- unsigned * [material_num_list](#)
- unsigned * [cell_num_list](#)
- std::string [datapath](#)
- std::vector< [primitive](#) > [primitives](#)
- std::vector< [material_def](#) > [materials](#)
- std::vector< std::string > [isotopes](#)
- std::string [isotope_list](#)
- unsigned * [isotope_list_array](#)
- unsigned * [material_list_array](#)
- float * [concentrations_matrix](#)
- float * [awr_list](#)

5.11.1 Detailed Description

[wgeometry](#) class

5.11.2 Constructor & Destructor Documentation

5.11.2.1 [wgeometry::wgeometry \(\)](#)

[wgeometry](#) constructor

5.11.2.2 [wgeometry::~~wgeometry \(\)](#)

[wgeometry](#) destructor

5.11.3 Member Function Documentation

5.11.3.1 `void wgeometry::add_material (unsigned matnum, unsigned is_fissile, unsigned num_topes, float density, std::vector< std::string > isotopes, std::vector< float > fractions)`

adds a material and its properties to the geometry, allocates space for all of the material information.

Parameters:

- ← *matnum* - material number
- ← *is_fissile* - fissile flag
- ← *num_topes* - number of isotopes in material
- ← *density* - density of material
- ← *isotopes* - list of isotopes
- ← *fractions* - fractions of the constituent isotopes

5.11.3.2 `unsigned wgeometry::add_primitive (int p_type, unsigned cellmat, std::vector< float > mins, std::vector< float > maxs, std::vector< float > origin)`

5.11.3.3 `unsigned wgeometry::add_primitive ()`

adds a [primitive](#) to the geometry object

5.11.3.4 `unsigned wgeometry::add_transform (unsigned index, unsigned cellnum, unsigned cellmat, float dx, float dy, float dz, float theta, float phi)`

5.11.3.5 `unsigned wgeometry::add_transform (unsigned index, unsigned cellnum, float dx, float dy, float dz, float theta, float phi)`

5.11.3.6 `unsigned wgeometry::add_transform (unsigned index)`

5.11.3.7 `int wgeometry::check ()`

checks that all cells have unique IDs, checks that there are materials for each number specified in the geometry, checks to make sure that the outer cell exists, checks to see if there are any fissile isotopes.

5.11.3.8 `unsigned wgeometry::check_fissile ()`

sets the data path

Returns:

void

5.11.3.9 void wgeometry::delete_primitive (unsigned *index*)**5.11.3.10 void wgeometry::delete_transform (unsigned *index*, unsigned *element*)****5.11.3.11 unsigned wgeometry::get_boundary_condition ()**

returns the outermost cell type

Returns:

outer_cell_type

5.11.3.12 unsigned wgeometry::get_material_count ()

returns the number of materials.

Returns:

n_materials

5.11.3.13 void wgeometry::get_material_table (unsigned * *n_mat_in*, unsigned * *n_tope_in*, float ** *conc_mat_in*)

creates material and isotope arrays, creates concentration matrix. copies memory for all of those arrays.

Parameters:← *n_mat_in* - number of input materials← *n_tope_in* - number of input isotopes← *material_list_in* - list of input materials← *isotope_list_in* - list of input isotopes← *conc_mat_in* - input concentration matrix**5.11.3.14 unsigned wgeometry::get_maximum_cell ()**

returns the largest cell number, typically the outermost cell

5.11.3.15 unsigned wgeometry::get_maximum_material ()

returns the largest material number

5.11.3.16 unsigned wgeometry::get_minimum_cell ()

returns the smallest cell number, typically the innermost cell

5.11.3.17 unsigned wgeometry::get_minimum_material ()

returns the smallest material number

5.11.3.18 unsigned wgeometry::get_outer_cell ()

returns the outermost cell and its boundary conition

Returns:

outer_cell

5.11.3.19 unsigned wgeometry::get_outer_cell_dims (float * *input_array*)

returns the dimensions of the outermost cell.

Parameters:

\leftarrow *input_array*

Returns:

primitives[k].type

5.11.3.20 unsigned wgeometry::get_outer_cell_type ()

returns the boundary condition flag of outermost cell type

Returns:

boundary_condition

5.11.3.21 unsigned wgeometry::get_primitive_count ()

returns the number of primitives in the geometry object

5.11.3.22 unsigned wgeometry::get_transform_count ()

returns the number of transforms in the geometry object

5.11.3.23 void wgeometry::make_hex_array (unsigned *index*, int *n*, float *x*, float *y*, float *phi*, unsigned *starting_index*)**5.11.3.24 void wgeometry::make_material_table ()**

makes a table of all of the materials.

allocates and copies the isotope and material number lists to their respective arrays, allocates and copies the isotope fractions to the concentration matrix, converts the fractions into number densities, normalizes the fractions, gets the average number density, prints each isotope's material, isotope, and density.

```
memcpy(isotope_list_array,isotopes.data(),n_isotopes*sizeof(unsigned));
```

5.11.3.25 void wgeometry::print_all ()

prints all of the transforms of all the primitives, then prints a geometry summary.

5.11.3.26 void wgeometry::print_materials_table ()

prints out all materials, including each material's constituent isotopes and their number densities.

5.11.3.27 void wgeometry::print_summary ()

prints a summary of the geometry object: numbers of the different kinds of shapes in the geometry, total numbers of primitives and transforms, outer cell, numbers of materials and isotopes, isotope list, properties (density, fissile flag, isotopes) of each material.

5.11.3.28 void wgeometry::set_datapath (std::string *path_in*)

checks whether or not the geometry contains a fissile material.

Returns:

fissile_flag

5.11.3.29 void wgeometry::set_outer_cell (unsigned *ocell*, unsigned *BC*)

goes through all the cells of all of the primitives and checks that the outer cell is set

Parameters:

← *ocell* - the outermost cell

5.11.3.30 void wgeometry::update ()

updates the numbers of all shapes, compiles the list of all isotopes, creates an isotope table.

5.11.4 Member Data Documentation**5.11.4.1 float* wgeometry::awr_list**

atomic weight ratio (AWR) list

5.11.4.2 unsigned wgeometry::boundary_condition

flag for the cell's boundary condition

5.11.4.3 unsigned* wgeometry::cell_num_list

list of cell numbers

5.11.4.4 float* wgeometry::concentrations_matrix

concentrations matrix

5.11.4.5 std::string wgeometry::datapath

path to xsdir and data

5.11.4.6 unsigned wgeometry::fissile_flag

indicates whether or not a material is fissile

5.11.4.7 std::string wgeometry::isotope_list

isotope list

5.11.4.8 unsigned* wgeometry::isotope_list_array

isotope list array

5.11.4.9 `std::vector<std::string> wgeometry::isotopes`

isotopes vector

5.11.4.10 `unsigned* wgeometry::material_list_array`

material list array

5.11.4.11 `unsigned* wgeometry::material_num_list`

list of material numbers

5.11.4.12 `std::vector<material_def> wgeometry::materials`

materials vector

5.11.4.13 `unsigned wgeometry::n_box`

number of boxes

5.11.4.14 `unsigned wgeometry::n_cyl`

number of cylinders

5.11.4.15 `unsigned wgeometry::n_hex`

number of hexagons

5.11.4.16 `unsigned wgeometry::n_isotopes`

number of isotopes

5.11.4.17 `unsigned wgeometry::n_materials`

number of materials

5.11.4.18 `unsigned wgeometry::n_primitives`

number of primitives

5.11.4.19 `unsigned wgeometry::n_sph`

number of spheres

5.11.4.20 unsigned wgeometry::n_transforms

number of transforms

5.11.4.21 unsigned wgeometry::outer_cell

outermost cell (usually used for tallying)

5.11.4.22 std::vector<primitive> wgeometry::primitives

primitives vector

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

- [wgeometry.h](#)
- [wgeometry.cpp](#)

5.12 whistory Class Reference

[whistory](#) class

```
#include <whistory.h>
```

Public Member Functions

- [whistory](#) (unsigned, [wgeometry](#))
constructor
- [~whistory](#) ()
destructor
- void [print_xs_data](#) ()
prints cross section data information
- void [print_pointers](#) ()
prints pointer information
- void [print_materials_table](#) ()
prints table of properties of geometry materials
- void [run](#) ()
runs history
- void [write_xs_data](#) (std::string)
writes cross section data to file
- void [write_tally](#) (unsigned)
writes tally values to file
- void [set_tally_cell](#) (unsigned)
sets tally cell to input value
- void [set_run_type](#) (unsigned)
sets run type
- void [set_run_type](#) (std::string)
sets run type
- void [set_run_param](#) (unsigned, unsigned)
sets number of cycles to skip and number of active cycles
- void [init](#) ()
initialization function

- void [device_report](#) ()
prints out details (model, memory, compute capability, etc.) of all available compute devices
- void [set_device](#) (unsigned)
sets device number to input value
- void [set_acceration](#) (std::string)
does nothing
- void [set_filename](#) (std::string)
sets filename to input string
- void [write_histories](#) (unsigned iteration)
appends history data to file in debug mode.
- void [set_print_level](#) (unsigned level)
sets amount of information printed to stdout
- void [set_dump_level](#) (unsigned level)
sets what types of information are dumped to files
- void [plot_geom](#) (std::string type)
produces png images of the geometry, named filename-[xy,xz,yz].png
- void [make_color](#) (float *, unsigned, unsigned, unsigned)
creates a color map
- void [hot2](#) (float *, long unsigned, long unsigned, long unsigned)
creates a hot2 color map
- void [nonzero](#) (float *, unsigned, unsigned, unsigned)
creates a binary colormap, black iff 0
- void [bin_fission_points](#) ([source_point](#) *, unsigned)
bins and accumulates fission points to grid
- void [write_fission_points](#) ()
writes binned fission point image to a .png

Private Member Functions

- void [init_RNG](#) ()
initializes the random number generator
- void [update_RNG](#) ()
updates the random number
- void [init_CUDPP](#) ()
initializes CUDPP
- void [init_host](#) ()
initializes data on the host device
- void [copy_data_to_device](#) ()
copies data from the host device to the compute device
- void [load_cross_sections](#) ()
loads cross sections
- void [trace](#) (unsigned)
does an OptiX trace
- void [trace](#) (unsigned, unsigned)
does an OptiX trace for a given number of active histories
- unsigned [reduce_done](#) ()
reduces done values
- void [reset_cycle](#) (float)
resets the cycle in criticality mode
- void [reset_fixed](#) ()
resets the cycle in fixed-source mode
- void [converge](#) (unsigned)
not called in [whistory.cpp](#)
- void [sample_fissile_points](#) ()
samples fissile points
- unsigned [reduce_yield](#) ()
reduces yield values
- float [reduce_weight](#) ()
reduces weight values

- void [accumulate_keff](#) (unsigned, unsigned, double *, float *)
accumulates yields into host side values
- void [accumulate_tally](#) ()
raccumulates the flux tally
- void [create_quad_tree](#) ()
builds a quad tree for energy search
- float [get_time](#) ()
returns how long it takes to do something
- void [prep_secondaries](#) ()
prepares for secondary neutrons
- unsigned [map_active](#) ()
maps done histories
- void [remap_active](#) (unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *)
remaps active histories
- void [write_to_file](#) ([source_point](#) *, unsigned, std::string, std::string)
prints the locations of the source points to file
- void [write_to_file](#) ([source_point](#) *, float *, unsigned, std::string, std::string)
prints the locations of the source points to file
- void [write_to_file](#) (unsigned *, unsigned, std::string, std::string)
prints the source points to a file
- void [write_to_file](#) (unsigned *, unsigned *, unsigned, std::string, std::string)
prints the locations of the source points to file
- void [write_results](#) (float, float, std::string)
writes results to file

Private Attributes

- [wgeometry problem_geom](#)
- std::string [accel_type](#)
- CUDPPHandle [theCudpp](#)
- CUDPPHashTableConfig [hash_config](#)
- CUDPPConfiguration [compact_config](#)

- CUDPPConfiguration [scan_int_config](#)
- CUDPPConfiguration [redu_int_config](#)
- CUDPPConfiguration [redu_float_config](#)
- CUDPPConfiguration [radix_config](#)
- CUDPPHandle [mate_hash_table_handle](#)
- CUDPPHandle [fiss_hash_table_handle](#)
- CUDPPHandle [scanplan_int](#)
- CUDPPHandle [reduplan_int](#)
- CUDPPHandle [reduplan_float](#)
- CUDPPHandle [compactplan](#)
- CUDPPHandle [radixplan](#)
- CUDPPResult [res](#)
- unsigned * [d_valid_result](#)
- unsigned * [d_valid_N](#)
- unsigned * [d_remap](#)
- curandGenerator_t [rand_gen](#)
- unsigned [N](#)
- unsigned [Ndataset](#)
- unsigned [RNUM_PER_THREAD](#)
- unsigned [NUM_THREADS](#)
- unsigned [blks](#)
- unsigned [compute_device](#)
- cudaStream_t [stream](#) [5]
- unsigned [RUN_FLAG](#)
- unsigned [qnodes_depth](#)
- unsigned [qnodes_width](#)
- unsigned [outer_cell](#)
- unsigned [outer_cell_type](#)
- unsigned [n_materials](#)
- unsigned [n_isotopes](#)
- unsigned [n_tally](#)
- unsigned [n_qnodes](#)
- unsigned [n_skip](#)
- unsigned [n_cycles](#)
- float [keff_sum](#)
- float [keff2_sum](#)
- float [keff_err](#)
- std::string [filename](#)
- unsigned [is_initialized](#)
- [source_point](#) * [space](#)
- unsigned [print_flag](#)
- unsigned [dump_flag](#)
- unsigned * [xs_length_numbers](#)
cross section length numbers
- unsigned * [xs_MT_numbers_total](#)

- unsigned * [xs_MT_numbers](#)
- float * [xs_data_MT](#)
- float * [xs_data_main_E_grid](#)
- float ** [xs_data_scatter](#)
- float ** [xs_data_energy](#)
- float ** [xs_data_scatter_host](#)
- float ** [xs_data_energy_host](#)
- float * [xs_data_Q](#)
- float * [E](#)
- float * [Q](#)
- unsigned * [rn_bank](#)
- float * [awr_list](#)
- float * [temp_list](#)
- float * [tally_score](#)
- float * [tally_square](#)
- unsigned * [tally_count](#)
- double * [tally_score_total](#)
- double * [tally_square_total](#)
- long unsigned * [tally_count_total](#)
- unsigned [tally_cell](#)
- unsigned * [index](#)
- unsigned * [cellnum](#)
- unsigned * [matnum](#)
- unsigned * [isonum](#)
- unsigned * [rxn](#)
- unsigned * [done](#)
- unsigned * [yield](#)
- float * [weight](#)
- float * [number_density_matrix](#)
- unsigned [reduced_yields](#)
- float [reduced_weight](#)
- unsigned * [remap](#)
- unsigned * [zeros](#)
- unsigned * [ones](#)
- float * [fones](#)
- long unsigned [reduced_yields_total](#)
- double [reduced_weight_total](#)
- [qnode](#) * [qnodes](#)
- [source_point](#) * [d_space](#)
- unsigned * [d_xs_length_numbers](#)
- unsigned * [d_xs_MT_numbers_total](#)
- unsigned * [d_xs_MT_numbers](#)
- float * [d_xs_data_MT](#)
- float * [d_xs_data_main_E_grid](#)
- float ** [d_xs_data_scatter](#)
- float ** [d_xs_data_energy](#)

- float * [d_xs_data_Q](#)
- float * [d_E](#)
- float * [d_Q](#)
- unsigned * [d_rn_bank](#)
- float * [d_awr_list](#)
- float * [d_temp_list](#)
- float * [d_tally_score](#)
- float * [d_tally_square](#)
- unsigned * [d_tally_count](#)
- unsigned * [d_index](#)
- unsigned * [d_cellnum](#)
- unsigned * [d_matnum](#)
- unsigned * [d_isonum](#)
- unsigned * [d_rxn](#)
- unsigned * [d_rxn_remap](#)
- unsigned * [d_done](#)
- unsigned * [d_yield](#)
- float * [d_weight](#)
- unsigned * [d_material_list](#)
- unsigned * [d_isotope_list](#)
- float * [d_number_density_matrix](#)
- unsigned * [d_reduced_yields](#)
- float * [d_reduced_weight](#)
- unsigned * [d_reduced_done](#)
- float * [d_fissile_energy](#)
- [source_point](#) * [d_fissile_points](#)
- unsigned * [d_mask](#)
- [qnode](#) * [d_qnodes_root](#)
- unsigned * [d_completed](#)
- unsigned * [d_scanned](#)
- unsigned * [d_active](#)
- unsigned * [d_num_completed](#)
- unsigned * [d_num_active](#)
- [source_point](#) * [d_bank_space](#)
- float * [d_bank_E](#)
- unsigned * [d_zeros](#)
- unsigned [n_edges](#)
- unsigned * [edges](#)
- unsigned * [d_edges](#)
- std::vector< std::string > [isotopes](#)
- std::vector< unsigned > [xs_num_rxns](#)
- std::vector< unsigned > [xs_isotope_ints](#)
- unsigned [total_bytes_scatter](#)
- unsigned [total_bytes_energy](#)
- unsigned [MT_rows](#)
- unsigned [MT_columns](#)
- float [outer_cell_dims](#) [6]
- long unsigned * [fiss_img](#)

5.12.1 Detailed Description

[whistory](#) class

5.12.2 Constructor & Destructor Documentation

5.12.2.1 whistory::whistory (unsigned *Nin*, *wgeometry problem_geom_in*)

constructor makes geometry, sets tally vector length, creates dataset size, sets compute device and acceleration type, creates CUDA streams.

5.12.2.2 whistory::~~whistory ()

destructor

5.12.3 Member Function Documentation

5.12.3.1 void whistory::accumulate_keff (unsigned *converged*, unsigned *iteration*, double * *keff*, float * *keff_cycle*) [private]

accumulates yields into host side values

Parameters:

- ← *iteration* - the active iteration number (starts at 0)
- ← *keff* - the running cumulative keff
- ← *keff_cycle* - the keff of the last cycle, used to renormalize the source vector

5.12.3.2 void whistory::accumulate_tally () [private]

racumulates the flux tally

Parameters:

- ←

5.12.3.3 void whistory::bin_fission_points (source_point * *d_space*, unsigned *N*)

bins and accumulates fission points to grid

Parameters:

- ← *d_space* - device space points
- ← *N* - dataset size

5.12.3.4 void whistory::converge (unsigned) [private]

not called in [whistory.cpp](#)

5.12.3.5 void whistory::copy_data_to_device () [private]

copies data from the host device to the compute device copies history data, cross section data, and the device pointer array. zeros out the tally arrays.

5.12.3.6 void whistory::create_quad_tree () [private]

builds a quad tree for energy search

5.12.3.7 void whistory::device_report ()

prints out details (model, memory, compute capability, etc.) of all available compute devices

5.12.3.8 float whistory::get_time () [private]

returns how long it takes to do something

5.12.3.9 void whistory::hot2 (float * *color*, long unsigned *val*, long unsigned *min*, long unsigned *max*)

creates a hot2 color map

Parameters:

← *color* - rgb colors, float[3]

← *x* - value

← *min,max* - values used to normalize the color

5.12.3.10 void whistory::init ()

initialization function initializes OptiX stuff and CUDA stuff, allocates device data, creates host data arrays, initializes counters to zero, copies outermost cell dimensions and isotope list, maps edge array, initializes host values, initializes the random number generator and CUDPP, loads cross sections, and copies data to the compute device.

5.12.3.11 void whistory::init_CUDPP () [private]

initializes CUDPP initializes global objects, sorting stuff, int reduction stuff, float reduction stuff, int scan stuff, radix sort stuff.

5.12.3.12 void whistory::init_host () [private]

initializes data on the host device prepares data arrays by filling them with zeros

5.12.3.13 void whistory::init_RNG () [private]

initializes the random number generator

5.12.3.14 void whistory::load_cross_sections () [private]

loads cross sections makes isotope list, initializes the cross section libraries, reads the cross section tables, unionizes the main energy grid across all isotopes, makes the total MT reaction list from all isotopes, allocates the unionized array, inserts and interpolates the cross sections, gets the MT array buffer, gets and copies the unionized MT array, gets the unionized main energy grid buffer, gets the MT number vector, gets the MT number total vector, gets the lengths vector, gets the AWR vector, gets the Q vector. does scattering stuff and energy stuff. passes information to the geometry.

5.12.3.15 void whistory::make_color (float * color, unsigned x, unsigned min, unsigned max)

creates a color map

Parameters:

- ← *color* - rgb colors, float[3]
- ← *x* - value
- ← *min,max* - values used to normalize the color

5.12.3.16 unsigned whistory::map_active () [private]

maps done histories flips done flag, remaps data to active histories, flips done flag back.

Returns:

num_active - number of active histories

5.12.3.17 void whistory::nonzero (float * color, unsigned val, unsigned min, unsigned max)

creates a binary colormap, black iff 0

Parameters:

- ← *color* - rgb colors, float[3]
- ← *x* - value

← *min,max* - values used to normalize the color; unused, only present to keep arguments the same as other colormaps

5.12.3.18 void whistory::plot_geom (std::string *type*)

produces png images of the geometry, named filename-[xy,xz,yz].png

Parameters:

← *type,color* is based 'cell' or 'material'

5.12.3.19 void whistory::prep_secondaries () [private]

prepares for secondary neutrons scans yields to determine where threads write into the done data, compacts done data to know where to write.

5.12.3.20 void whistory::print_materials_table ()

prints table of properties of geometry materials

5.12.3.21 void whistory::print_pointers ()

prints pointer information

5.12.3.22 void whistory::print_xs_data ()

prints cross section data information

5.12.3.23 unsigned whistory::reduce_done () [private]

reduces done values

Returns:

reduced_done - number of done values

5.12.3.24 float whistory::reduce_weight () [private]

reduces weight values

Returns:

total

5.12.3.25 unsigned whistory::reduce_yield () [private]

reduces yield values

Returns:

total

5.12.3.26 void whistory::remap_active (unsigned * *num_active*, unsigned * *escatter_N*, unsigned * *escatter_start*, unsigned * *iscatter_N*, unsigned * *iscatter_start*, unsigned * *cscatter_N*, unsigned * *cscatter_start*, unsigned * *fission_N*, unsigned * *fission_start*) [private]

remaps active histories sorts reaction values, launches edge detection kernels, calculates values for reaction blocks, calculates the total number of active histories, and rezeros the edge vector.

Parameters:

- ← *num_active* - number of active histories
- ← *escatter_N* - number of elastic scatters
- ← *escatter_start* - elastic scattering start
- ← *iscatter_N* - number of inelastic scatters
- ← *iscatter_start* - inelastic scattering start
- ← *cscatter_N* - number of compound scatters
- ← *cscatter_start* - compound scattering start
- ← *fission_N* - number of fissions
- ← *fission_start* - fission start

5.12.3.27 void whistory::reset_cycle (float *keff_cycle*) [private]

resets the cycle in criticality mode rebases the yield so that keff is 1, scans the yield to see where to write, sorts the reaction vector, populates the source, resets the run arrays, and updates the random numbers.

Parameters:

- ← *keff_cycle* - keff value of previous cycle

5.12.3.28 void whistory::reset_fixed () [private]

resets the cycle in fixed-source mode resets the read-in run arrays, samples the fixed source, and updates the random numbers.

5.12.3.29 void whistory::run ()

runs history initializes run variables, clears fissile points file, opens run stats file, records stats. finds the material and nearest surfact distance, finds the main energy grid index, finds interaction length, computes spectra, finds reaction type, remaps threads, does scattering reactions, reduces the yield, resets the cycle, recalculates the running average, and prints the total transport runtime.

5.12.3.30 void whistory::sample_fissile_points () [private]

samples fissile points updates the random numbers, sets uniformly random positions, runs OptiX to get the cell number, compacts data, copies the data back, copies new values in, writes starting positions to file, and updates the random numbers.

5.12.3.31 void whistory::set_acceration (std::string *accel_in*)

does nothing

Parameters:

← *accel_in* - acceleration type

5.12.3.32 void whistory::set_device (unsigned *dev_in*)

sets device number to input value

Parameters:

← *dev_in* - device number

5.12.3.33 void whistory::set_dump_level (unsigned *level*)

sets what types of information are dumped to files

Parameters:

← *level* - dump type flag

5.12.3.34 void whistory::set_filename (std::string *filename_in*)

sets filename to input string

Parameters:

← *filename_in* - filename

5.12.3.35 void whistory::set_print_level (unsigned *level*)

sets amount of information printed to stdout

Parameters:

← *level* - level of verbosity

5.12.3.36 void whistory::set_run_param (unsigned *n_cycles_in*, unsigned *n_skip_in*)

sets number of cycles to skip and number of active cycles

Parameters:

← *n_cycles_in* - number of active cycles

← *n_skip_in* - number of cycles to skip

5.12.3.37 void whistory::set_run_type (std::string *type_in*)

sets run type

Parameters:

← *type_in* - run type

5.12.3.38 void whistory::set_run_type (unsigned *type_in*)

sets run type

Parameters:

← *type_in* - run type

5.12.3.39 void whistory::set_tally_cell (unsigned *cell*)

sets tally cell to input value

Parameters:

← *cell* - tally cell

5.12.3.40 void whistory::trace (unsigned *type*, unsigned *n_active*) [private]

does an OptiX trace for a given number of active histories

Parameters:

- ← *type* - trace type
- ← *n_active* - number of active histories

5.12.3.41 void whistory::trace (unsigned *type*) [private]

does an OptiX trace

Parameters:

- ← *type* - trace type

5.12.3.42 void whistory::update_RNG () [private]

updates the random number

5.12.3.43 void whistory::write_fission_points ()

writes binned fission point image to a .png

5.12.3.44 void whistory::write_histories (unsigned *iteration*)

appends history data to file in debug mode.

Parameters:

- ← *iteration* - iteration number

5.12.3.45 void whistory::write_results (float *runtime*, float *keff*, std::string *opentype*) [private]

writes results to file

Parameters:

- ← *runtime* - runtime
- ← *keff* - keff
- ← *opentype* - file extension

5.12.3.46 void whistory::write_tally (unsigned *tallynum*)

writes tally values to file

Parameters:

← *tallynum* - tally number

5.12.3.47 void whistory::write_to_file (unsigned * *array_in*, unsigned * *array_in2*, unsigned *N*, std::string *filename*, std::string *opentype*) [private]

prints the locations of the source points to file

Parameters:

← *array_in* - source point array

← *array_in2* - second array

← *N* - number of histories

← *filename* - filename

← *opentype* - file extension

5.12.3.48 void whistory::write_to_file (unsigned * *array_in*, unsigned *N*, std::string *filename*, std::string *opentype*) [private]

prints the source points to a file

Parameters:

← *array_in* - source point array

← *N* - number of histories

← *filename* - filename

← *opentype* - file extension

5.12.3.49 void whistory::write_to_file (source_point * *array_in*, float * *array_in2*, unsigned *N*, std::string *filename*, std::string *opentype*) [private]

prints the locations of the source points to file

Parameters:

← *array_in* - source point array

← *array_in2* - second array

← *N* - number of histories

← *filename* - filename

← *opentype* - file extension

5.12.3.50 void whistory::write_to_file (source_point * array_in, unsigned N, std::string filename, std::string opentype) [private]

prints the locations of the source points to file

Parameters:

- ← *array_in* - source point array
- ← *N* - number of histories
- ← *filename* - filename
- ← *opentype* - file extension

5.12.3.51 void whistory::write_xs_data (std::string filename)

writes cross section data to file

Parameters:

- ← *filename* - filename

5.12.4 Member Data Documentation

5.12.4.1 std::string whistory::accel_type [private]

acceleration type

5.12.4.2 float* whistory::awr_list [private]

atomic weight ratio (AWR) list

5.12.4.3 unsigned whistory::blks [private]

number of blocks

5.12.4.4 unsigned* whistory::cellnum [private]

cell number

5.12.4.5 CUDPPConfiguration whistory::compact_config [private]

CUDPP compact configuration

5.12.4.6 CUDPPHandle whistory::compactplan [private]

CUDPP compact plan handle

5.12.4.7 unsigned whistory::compute_device [private]

compute device (always 0)

5.12.4.8 unsigned* whistory::d_active [private]

device active pointer

5.12.4.9 float* whistory::d_awr_list [private]

device AWR list

5.12.4.10 float* whistory::d_bank_E [private]

device bank energy

5.12.4.11 source_point* whistory::d_bank_space [private]

device bank space

5.12.4.12 unsigned* whistory::d_cellnum [private]

device cell number

5.12.4.13 unsigned* whistory::d_completed [private]

device completed pointer

5.12.4.14 unsigned* whistory::d_done [private]

device done flag

5.12.4.15 float* whistory::d_E [private]

device energy

5.12.4.16 unsigned* whistory::d_edges [private]

device mapped array of edges

5.12.4.17 float* whistory::d_fissile_energy [private]

device fissile energy

5.12.4.18 source_point* whistory::d_fissile_points [private]

device fissile points

5.12.4.19 unsigned* whistory::d_index [private]

device index

5.12.4.20 unsigned* whistory::d_isonum [private]

device isotope number

5.12.4.21 unsigned* whistory::d_isotope_list [private]

device isotope list

5.12.4.22 unsigned* whistory::d_mask [private]

device mask

5.12.4.23 unsigned* whistory::d_material_list [private]

device material list

5.12.4.24 unsigned* whistory::d_matnum [private]

device material number

5.12.4.25 unsigned* whistory::d_num_active [private]

device number of active histories

5.12.4.26 unsigned* whistory::d_num_completed [private]

device number of completed histories

5.12.4.27 float* whistory::d_number_density_matrix [private]

device isotope number density matrix

5.12.4.28 float* whistory::d_Q [private]

device Q-value

5.12.4.29 `qnode* whistory::d_qnodes_root` `[private]`

device quaternary nodes root

5.12.4.30 `unsigned* whistory::d_reduced_done` `[private]`

device reduced done flags

5.12.4.31 `float* whistory::d_reduced_weight` `[private]`

device reduced weight

5.12.4.32 `unsigned* whistory::d_reduced_yields` `[private]`

device reduced yields

5.12.4.33 `unsigned* whistory::d_remap` `[private]`

remap pointer

5.12.4.34 `unsigned* whistory::d_rn_bank` `[private]`

device reaction bank

5.12.4.35 `unsigned* whistory::d_rxn` `[private]`

device reaction

5.12.4.36 `unsigned* whistory::d_rxn_remap` `[private]`

device reaction remap

5.12.4.37 `unsigned* whistory::d_scanned` `[private]`

device scanned pointer

5.12.4.38 `source_point* whistory::d_space` `[private]`

device spatial source point

5.12.4.39 `unsigned* whistory::d_tally_count` `[private]`

device tally count

5.12.4.40 float* whistory::d_tally_score [private]

device tally score

5.12.4.41 float* whistory::d_tally_square [private]

device tally square

5.12.4.42 float* whistory::d_temp_list [private]

device isotope temperature list

5.12.4.43 unsigned* whistory::d_valid_N [private]

valied number of histories pointer

5.12.4.44 unsigned* whistory::d_valid_result [private]

valid result pointer

5.12.4.45 float* whistory::d_weight [private]

device neutron weight

5.12.4.46 float whistory::d_xs_data_energy [private]**

device energy cross section data

5.12.4.47 float* whistory::d_xs_data_main_E_grid [private]

device cross section main energy grid

5.12.4.48 float* whistory::d_xs_data_MT [private]

device cross section MT numbers

5.12.4.49 float* whistory::d_xs_data_Q [private]

device cross section Q-value data

5.12.4.50 float whistory::d_xs_data_scatter [private]**

device scattering cross section data

5.12.4.51 unsigned* whistory::d_xs_length_numbers [private]

device cross section length numbers

5.12.4.52 unsigned* whistory::d_xs_MT_numbers [private]

device cross section MT numbers

5.12.4.53 unsigned* whistory::d_xs_MT_numbers_total [private]

device cross section total MT numbers

5.12.4.54 unsigned* whistory::d_yield [private]

device yield

5.12.4.55 unsigned* whistory::d_zeros [private]

zeros

5.12.4.56 unsigned* whistory::done [private]

done flag

5.12.4.57 unsigned whistory::dump_flag [private]

dump level

5.12.4.58 float* whistory::E [private]

energy

5.12.4.59 unsigned* whistory::edges [private]

mapped array of edges

5.12.4.60 std::string whistory::filename [private]

file name

5.12.4.61 CUDPPHandle whistory::fiss_hash_table_handle [private]

CUDPP fissile hash table handle

5.12.4.62 long unsigned* whistory::fiss_img [private]

fissile image accumulation

5.12.4.63 float* whistory::fones [private]

float ones

5.12.4.64 CUDPPHashTableConfig whistory::hash_config [private]

CUDPP hash table configuration

5.12.4.65 unsigned* whistory::index [private]

index

5.12.4.66 unsigned whistory::is_initialized [private]

init flag

5.12.4.67 unsigned* whistory::isonum [private]

isotope number

5.12.4.68 std::vector<std::string> whistory::isotopes [private]

cross section isotope string

5.12.4.69 float whistory::keff2_sum [private]**5.12.4.70 float whistory::keff_err [private]****5.12.4.71 float whistory::keff_sum [private]****5.12.4.72 CUDPPHandle whistory::mate_hash_table_handle [private]**

CUDPP material hash table handle

5.12.4.73 unsigned* whistory::matnum [private]

material number

5.12.4.74 unsigned whistory::MT_columns [private]

MT number columns

5.12.4.75 unsigned whistory::MT_rows [private]

MT number rows

5.12.4.76 unsigned whistory::N [private]

number of histories

5.12.4.77 unsigned whistory::n_cycles [private]

number of active cycles

5.12.4.78 unsigned whistory::n_edges [private]

mapped array of number of edges

5.12.4.79 unsigned whistory::n_isotopes [private]

number of isotopes

5.12.4.80 unsigned whistory::n_materials [private]

number of materials

5.12.4.81 unsigned whistory::n_qnodes [private]

number of quaternary nodes

5.12.4.82 unsigned whistory::n_skip [private]

number of cycles to skip

5.12.4.83 unsigned whistory::n_tally [private]

number of tallies

5.12.4.84 unsigned whistory::Ndataset [private]

dataset size for number of histories

5.12.4.85 unsigned whistory::NUM_THREADS [private]

number of threads

5.12.4.86 float* whistory::number_density_matrix [private]

isotope number density matrix

5.12.4.87 unsigned* whistory::ones [private]

ones

5.12.4.88 unsigned whistory::outer_cell [private]

outermost cell

5.12.4.89 float whistory::outer_cell_dims[6] [private]

outer cell minima and maxima

5.12.4.90 unsigned whistory::outer_cell_type [private]

outermost cell type

5.12.4.91 unsigned whistory::print_flag [private]

print verbosity level

5.12.4.92 wgeometry whistory::problem_geom [private]

problem geometry

5.12.4.93 float* whistory::Q [private]

Q-value

5.12.4.94 qnode* whistory::qnodes [private]

quaternary nodes

5.12.4.95 unsigned whistory::qnodes_depth [private]

quaternary node depth

5.12.4.96 unsigned whistory::qnodes_width [private]

quaternary node width

5.12.4.97 CUDPPConfiguration whistory::radix_config [private]

CUDPP radix configuration

5.12.4.98 CUDPPHandle whistory::radixplan [private]

CUDPP radix plan handle

5.12.4.99 curandGenerator_t whistory::rand_gen [private]

random number generator

5.12.4.100 CUDPPConfiguration whistory::redu_float_config [private]

CUDPP reduced float configuration

5.12.4.101 CUDPPConfiguration whistory::redu_int_config [private]

CUDPP reduced int configuration

5.12.4.102 float whistory::reduced_weight [private]

reduced weight

5.12.4.103 double whistory::reduced_weight_total [private]

double for accumulating weight numbers accurately on the host

5.12.4.104 unsigned whistory::reduced_yields [private]

reduced yields

5.12.4.105 long unsigned whistory::reduced_yields_total [private]

long unsigned for accumulating yield numbers accurately on the host

5.12.4.106 CUDPPHandle whistory::reduplan_float [private]

CUDPP reduce plan float handle

5.12.4.107 CUDPPHandle whistory::reduplan_int [private]

CUDPP reduce plan int handle

5.12.4.108 unsigned* whistory::remap [private]

remap

5.12.4.109 CUDPPResult whistory::res [private]

CUDPP result

5.12.4.110 unsigned* whistory::rn_bank [private]

reaction bank

5.12.4.111 unsigned whistory::RNUM_PER_THREAD [private]

random numbers per thread

5.12.4.112 unsigned whistory::RUN_FLAG [private]

run flag

5.12.4.113 unsigned* whistory::rxn [private]

reaction

5.12.4.114 CUDPPConfiguration whistory::scan_int_config [private]

CUDPP scan int configuration

5.12.4.115 CUDPPHandle whistory::scanplan_int [private]

CUDPP scan plan int handle

5.12.4.116 source_point* whistory::space [private]

source point spatial pointer

5.12.4.117 cudaStream_t whistory::stream[5] [private]

CUDA stream

5.12.4.118 unsigned whistory::tally_cell [private]

tally cell

5.12.4.119 unsigned* whistory::tally_count [private]

tally count

5.12.4.120 long unsigned* whistory::tally_count_total [private]

tally count accumulated total

5.12.4.121 float* whistory::tally_score [private]

tally score

5.12.4.122 double* whistory::tally_score_total [private]

tally score accumulated total

5.12.4.123 float* whistory::tally_square [private]

tally square

5.12.4.124 double* whistory::tally_square_total [private]

tally square accumulaed total

5.12.4.125 float* whistory::temp_list [private]

isotope temperature list

5.12.4.126 CUDPPHandle whistory::theCudpp [private]

CUDPP handle

5.12.4.127 unsigned whistory::total_bytes_energy [private]

total size of energy data

5.12.4.128 unsigned whistory::total_bytes_scatter [private]

total size of scattering data

5.12.4.129 `float* whistory::weight` `[private]`

neutron weight

5.12.4.130 `float** whistory::xs_data_energy` `[private]`

energy cross section data

5.12.4.131 `float** whistory::xs_data_energy_host` `[private]`

energy cross section host data

5.12.4.132 `float* whistory::xs_data_main_E_grid` `[private]`

cross section data main energy grid

5.12.4.133 `float* whistory::xs_data_MT` `[private]`

cross section MT data

5.12.4.134 `float* whistory::xs_data_Q` `[private]`

cross section data Q-values

5.12.4.135 `float** whistory::xs_data_scatter` `[private]`

scattering cross section data

5.12.4.136 `float** whistory::xs_data_scatter_host` `[private]`

scattering cross section host data

5.12.4.137 `std::vector<unsigned> whistory::xs_isotope_ints` `[private]`

cross section isotope numbers

5.12.4.138 `unsigned* whistory::xs_length_numbers` `[private]`

cross section length numbers 0 = isotopes, 1 = main menenergy points, 2 = total number of reaction channels, 3 = matrix energy points, 4 = angular cosine points, 5 = outgoing energy points

5.12.4.139 unsigned* whistory::xs_MT_numbers [private]

cross section MT numbers

5.12.4.140 unsigned* whistory::xs_MT_numbers_total [private]

total cross section MT numbers

5.12.4.141 std::vector<unsigned> whistory::xs_num_rxns [private]

cross section number of reactions

5.12.4.142 unsigned* whistory::yield [private]

yield

5.12.4.143 unsigned* whistory::zeros [private]

zeros

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

- [whistory.h](#)
- [whistory.cpp](#)

5.13 wtransform Struct Reference

contains parameters of a [wtransform](#)

```
#include <datadef.h>
```

Public Attributes

- unsigned [cellnum](#)
- unsigned [cellmat](#)
- float [dx](#)
- float [dy](#)
- float [dz](#)
- float [theta](#)
- float [phi](#)

5.13.1 Detailed Description

contains parameters of a [wtransform](#) cell number and material, transform coordinates and angles

5.13.2 Member Data Documentation

5.13.2.1 unsigned wtransform::cellmat

cell material

5.13.2.2 unsigned wtransform::cellnum

cell number

5.13.2.3 float wtransform::dx

dx

5.13.2.4 float wtransform::dy

dy

5.13.2.5 float wtransform::dz

dz

5.13.2.6 float wtransform::phi

azimuthal angle

5.13.2.7 float wtransform::theta

polar angle

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

- [datadef.h](#)

Chapter 6

File Documentation

6.1 benchmarks.cpp File Reference

```
#include "warp.h"
```

Functions

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

6.1.1 Function Documentation

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

6.2 `binary_search.h` File Reference

Functions

- `__forceinline__ __device__ unsigned binary_search` (float *array, float value, unsigned len)

6.2.1 Function Documentation

6.2.1.1 `__forceinline__ __device__ unsigned binary_search` (float * *array*, float *value*, unsigned *len*)

6.3 datadef.h File Reference

```
#include <vector>
```

```
#include <string>
```

Classes

- struct [wtransform](#)
contains parameters of a [wtransform](#)
- struct [geom_data](#)
contains parameters of a [wgeometry](#)
- struct [source_point](#)
contains the parameters of the neutron source point
- struct [qnode](#)
quaternary search node
- struct [qnode_host](#)
quaternary search node host container
- struct [intersection_point](#)
contains information pertinent to an intersection point
- struct [material_def](#)
contains information that defines a material

6.4 device_copies.h File Reference

Functions

- void [copy_to_device](#) (void *, void *, unsigned)
- void [copy_from_device](#) (void *, void *, unsigned)
- void [allocate_on_device](#) (void **, unsigned)
- void [deallocate_on_device](#) (void *)

6.4.1 Function Documentation

6.4.1.1 void [allocate_on_device](#) (void **, unsigned)

6.4.1.2 void [copy_from_device](#) (void *, void *, unsigned)

6.4.1.3 void [copy_to_device](#) (void *, void *, unsigned)

6.4.1.4 void [deallocate_on_device](#) (void *)

6.5 main.cpp File Reference

```
#include "warp.h"
```

Functions

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

6.5.1 Function Documentation

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

6.6 mt19937ar.cpp File Reference

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

Defines

- #define [N](#) 624
- #define [M](#) 397
- #define [MATRIX_A](#) 0x9908b0dfUL
- #define [UPPER_MASK](#) 0x80000000UL
- #define [LOWER_MASK](#) 0x7fffffffUL

Functions

- void [init_genrand](#) (unsigned long s)
- void [init_by_array](#) (unsigned long init_key[], int key_length)
- unsigned long [genrand_int32](#) (void)
- long [genrand_int31](#) (void)
- double [genrand_real1](#) (void)
- double [genrand_real2](#) (void)
- double [genrand_real3](#) (void)
- double [genrand_res53](#) (void)

Variables

- static unsigned long [mt](#) [N]
- static int [mti](#) = N+1

6.6.1 Define Documentation

6.6.1.1 `#define LOWER_MASK 0x7ffffffUL`

6.6.1.2 `#define M 397`

6.6.1.3 `#define MATRIX_A 0x9908b0dfUL`

6.6.1.4 `#define N 624`

6.6.1.5 `#define UPPER_MASK 0x80000000UL`

6.6.2 Function Documentation

6.6.2.1 `long genrand_int31 (void)`

6.6.2.2 `unsigned long genrand_int32 (void)`

6.6.2.3 `double genrand_real1 (void)`

6.6.2.4 `double genrand_real2 (void)`

6.6.2.5 `double genrand_real3 (void)`

6.6.2.6 `double genrand_res53 (void)`

6.6.2.7 `void init_by_array (unsigned long init_key[], int key_length)`

6.6.2.8 `void init_genrand (unsigned long s)`

6.6.3 Variable Documentation

6.6.3.1 `unsigned long mt[N] [static]`

6.6.3.2 `int mti = N+1 [static]`

6.7 optix_stuff.cpp File Reference

```
#include <vector>
#include <iostream>
#include <sstream>
#include <cmath>
#include <assert.h>
#include <time.h>
#include <string.h>
#include <png++/png.hpp>
#include "datadef.h"
#include "primitive.h"
#include "wgeometry.h"
#include "optix_stuff.h"
#include "device_copies.h"
```

6.8 optix_stuff.h File Reference

```
#include <optix_world.h>
```

Classes

- class [optix_stuff](#)
OptiX stuff class.

6.9 optixtest.cpp File Reference

```
#include "warp.h"  
#include "optix_stuff.h"
```

Functions

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

6.9.1 Function Documentation

6.9.1.1 int main (int *argc*, char ** *argv*)

6.10 primitive.cpp File Reference

```
#include <vector>
#include <iostream>
#include <sstream>
#include <cmath>
#include <assert.h>
#include <time.h>
#include "datadef.h"
#include "primitive.h"
```

6.11 primitive.h File Reference

```
#include "datadef.h"
```

Classes

- class [primitive](#)
[primitive](#) class

6.12 print_banner.cpp File Reference

```
#include <iostream>
```

Functions

- void [print_banner](#) ()

6.12.1 Function Documentation

6.12.1.1 void print_banner ()

6.13 test.py File Reference

Namespaces

- namespace `test`

Variables

- tuple `test::geom` = `warp.wgeometry()`
- int `test::n_topes` = 4
- int `test::prim_id` = 0
- string `test::assemblyname` = "assembly"
- tuple `test::topes` = `warp.Unsigned(n_topes)`
- tuple `test::fracs_fuel` = `warp.Float(n_topes)`
- tuple `test::fracs_water` = `warp.Float(n_topes)`
- tuple `test::mins` = `warp.Float(3)`
- tuple `test::maxs` = `warp.Float(3)`
- tuple `test::origin` = `warp.Float(3)`
- int `test::dens_fuel` = 15
- int `test::dens_water` = 3
- int `test::tallycell` = 316
- `test::filename` = `assemblyname`
- `test::tallyname` = `assemblyname`
- int `test::typ` = 1
- int `test::material` = 1
- tuple `test::idx` = `geom.add_transform(prim_id,999,0,0,0,0,0)`
- tuple `test::hist` = `warp.whistory(100000,geom)`

6.14 unionize.py File Reference

Classes

- class [unionize::cross_section_data](#)
handles cross section data

Namespaces

- namespace [unionize](#)

6.15 warp.h File Reference

```
#include <vector>
#include <iostream>
#include <sstream>
#include <cmath>
#include <assert.h>
#include <time.h>
#include <string.h>
#include <cuda.h>
#include <curand.h>
#include < cudpp_hash.h>
#include <Python.h>
#include <png++/png.hpp>
#include "datadef.h"
#include "primitive.h"
#include "wgeometry.h"
#include "warp_cuda.h"
#include "whistory.h"
```

6.16 warp_cuda.h File Reference

```
#include "device_copies.h"
```

Functions

- void [print_banner](#) ()
- void [write_to_file](#) (unsigned *, unsigned, std::string)
- void [set_positions_rand](#) (unsigned, unsigned, unsigned, [source_point](#) *, unsigned *, float *)
- void [copy_points](#) (unsigned, unsigned, unsigned *, unsigned, unsigned *, [source_point](#) *, [source_point](#) *, float *, float *)
- void [sample_fission_spectra](#) (unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, float *, [source_point](#) *, float **)
- void [sample_fixed_source](#) (unsigned, unsigned, unsigned, unsigned *, unsigned *, float *, [source_point](#) *)
- void [macroscopic](#) (unsigned, unsigned, unsigned, unsigned, unsigned, unsigned, unsigned *, [source_point](#) *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, float *, unsigned *, float *, float *, float *, unsigned *)
- void [microscopic](#) (unsigned, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, float *, unsigned *, float *, float *, unsigned *, unsigned *, float *, unsigned *, float *, unsigned *)
- void [tally_spec](#) (unsigned, unsigned, unsigned, unsigned, unsigned *, [source_point](#) *, float *, float *, float *, unsigned *, unsigned *, unsigned *, unsigned *, float *)
- void [escatter](#) (cudaStream_t, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, [source_point](#) *, unsigned *, float *, float *, unsigned *, float **)
- void [iscatter](#) (cudaStream_t, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, [source_point](#) *, unsigned *, float *, float *, unsigned *, float **, float **)
- void [cscatter](#) (cudaStream_t, unsigned, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, [source_point](#) *, unsigned *, float *, float *, unsigned *, float **, float **)
- void [fission](#) (cudaStream_t, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, [source_point](#) *, unsigned *, float *, unsigned *, float *, float **, float **)
- void [absorb](#) (cudaStream_t, unsigned, unsigned, unsigned *, unsigned *, unsigned *)
- void [find_E_grid_index](#) (unsigned, unsigned, unsigned, unsigned *, float *, float *, unsigned *, unsigned *)
- void [find_E_grid_index_quad](#) (unsigned, unsigned, unsigned, unsigned, unsigned *, [qnode](#) *, float *, unsigned *, unsigned *)
- void [print_histories](#) (unsigned, unsigned, unsigned *, unsigned *, [source_point](#) *, float *, unsigned *, unsigned *, unsigned *)
- void [pop_secondaries](#) (unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, [source_point](#) *, float *, unsigned *, float **)

- void [pop_source](#) (unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, [source_point](#) *, float *, unsigned *, float **, float **, [source_point](#) *, float *, float *, float *)

- void [rebase_yield](#) (unsigned, unsigned, float, unsigned *, unsigned *)

- void [reaction_edges](#) (unsigned, unsigned, unsigned *, unsigned *)

- void [check_remap](#) (unsigned, unsigned, unsigned *, unsigned *, unsigned *)

- void [print_data](#) (cudaStream_t, unsigned, unsigned, [source_point](#) *, float *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *)

6.16.1 Function Documentation

- 6.16.1.1 `void absorb (cudaStream_t, unsigned, unsigned, unsigned *, unsigned *, unsigned *)`
- 6.16.1.2 `void check_remap (unsigned, unsigned, unsigned *, unsigned *, unsigned *)`
- 6.16.1.3 `void copy_points (unsigned, unsigned, unsigned *, unsigned, unsigned *, source_point *, source_point *, float *, float *)`
- 6.16.1.4 `void cscatter (cudaStream_t, unsigned, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, source_point *, unsigned *, float *, float *, unsigned *, float **, float **)`
- 6.16.1.5 `void escatter (cudaStream_t, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, source_point *, unsigned *, float *, float *, unsigned *, float **)`
- 6.16.1.6 `void find_E_grid_index (unsigned, unsigned, unsigned, unsigned *, float *, float *, unsigned *, unsigned *)`
- 6.16.1.7 `void find_E_grid_index_quad (unsigned, unsigned, unsigned, unsigned, unsigned *, qnode *, float *, unsigned *, unsigned *)`
- 6.16.1.8 `void fission (cudaStream_t, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, source_point *, unsigned *, float *, unsigned *, float *, float **, float **)`
- 6.16.1.9 `void isscatter (cudaStream_t, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, float *, source_point *, unsigned *, float *, float *, unsigned *, float **, float **)`
- 6.16.1.10 `void macroscopic (unsigned, unsigned, unsigned, unsigned, unsigned, unsigned, unsigned *, source_point *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, float *, unsigned *, float *, float *, float *, unsigned *)`
- 6.16.1.11 `void microscopic (unsigned, unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, float *, unsigned *, float *, float *, unsigned *, unsigned *, float *, unsigned *, float *, unsigned *)`
- 6.16.1.12 `void pop_secondaries (unsigned, unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, source_point *, float *, unsigned *, float **)`
- 6.16.1.13 `void pop_source (unsigned, unsigned, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, source_point *, float *, unsigned *, float **, float **, source_point *, float *, float *, float *)`
- 6.16.1.14 `void print_banner ()`
- 6.16.1.15 `void print_data (cudaStream_t, unsigned, unsigned, source_point *, float *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *, unsigned *)`
- 6.16.1.16 `void print_histories (unsigned, unsigned, unsigned *, unsigned *,`

6.17 warp_header.h File Reference

6.18 wfloat3.h File Reference

Classes

- class [wfloat3](#)
class definitions for device vector operations

6.19 wgeometry.cpp File Reference

```
#include <vector>
#include <iostream>
#include <sstream>
#include <stdio.h>
#include <cmath>
#include <assert.h>
#include <time.h>
#include <string.h>
#include "datadef.h"
#include "primitive.h"
#include "wgeometry.h"
```

6.20 wgeometry.h File Reference

```
#include "primitive.h"
```

Classes

- class [wgeometry](#)
wgeometry class

6.21 whistory.cpp File Reference

```
#include <vector>
#include <iostream>
#include <sstream>
#include <stdio.h>
#include <cmath>
#include <assert.h>
#include <time.h>
#include <string.h>
#include <cuda.h>
#include <curand.h>
#include < cudpp_hash.h>
#include <Python.h>
#include <png++/png.hpp>
#include "datadef.h"
#include "primitive.h"
#include "wgeometry.h"
#include "optix_stuff.h"
#include "warp_cuda.h"
#include "whistory.h"
```

Defines

- #define [check_cuda](#)(ans) { check_cuda((ans), __FILE__, __LINE__); }

Functions

- void [check_cuda](#) (cudaError_t code, const char *file, int line, bool abort=true)

Variables

- [optix_stuff](#) [optix_obj](#)

6.21.1 Define Documentation

6.21.1.1 `#define check_cuda(ans) { check_cuda((ans), __FILE__, __LINE__); }`

6.21.2 Function Documentation

6.21.2.1 `void check_cuda (cudaError_t code, const char *file, int line, bool abort = true) [inline]`

6.21.3 Variable Documentation

6.21.3.1 `optix_stuff optix_obj`

6.22 whistory.h File Reference

Classes

- class [whistory](#)
whistory class