



WARP v0.1 Source Reference

Generated by Doxygen 1.8.14

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	ace Namespace Reference	9
5.1.1	Detailed Description	10
5.1.2	Function Documentation	10
5.1.2.1	ascii_to_binary()	10
5.1.2.2	fromstring_split()	10
5.1.2.3	fromstring_token()	11
5.1.3	Variable Documentation	11
5.1.3.1	NP_LE_V15	11
5.1.3.2	table_types	12
5.2	unionize Namespace Reference	12
5.2.1	Variable Documentation	12
5.2.1.1	ace_available	12

6	Class Documentation	13
6.1	ace.AceTable Class Reference	13
6.1.1	Detailed Description	14
6.1.2	Constructor & Destructor Documentation	14
6.1.2.1	__init__()	14
6.1.3	Member Function Documentation	14
6.1.3.1	_read_all()	14
6.1.4	Member Data Documentation	14
6.1.4.1	awr	14
6.1.4.2	name	15
6.1.4.3	temp	15
6.2	cross_section_data Struct Reference	15
6.2.1	Detailed Description	15
6.2.2	Member Data Documentation	15
6.2.2.1	awr	16
6.2.2.2	dist_energy	16
6.2.2.3	dist_scatter	16
6.2.2.4	energy_grid	16
6.2.2.5	energy_grid_len	16
6.2.2.6	n_isotopes	16
6.2.2.7	Q	16
6.2.2.8	rxn_numbers	16
6.2.2.9	rxn_numbers_total	17
6.2.2.10	temp	17
6.2.2.11	total_reaction_channels	17
6.2.2.12	xs	17
6.3	unionize.cross_section_data Class Reference	17
6.3.1	Detailed Description	19
6.3.2	Constructor & Destructor Documentation	19
6.3.2.1	__init__()	19

6.3.3	Member Function Documentation	20
6.3.3.1	<code>_add_isotope()</code>	20
6.3.3.2	<code>_allocate_arrays()</code>	20
6.3.3.3	<code>_get_awr_pointer()</code>	20
6.3.3.4	<code>_get_energy_data()</code>	21
6.3.3.5	<code>_get_length_numbers_pointer()</code>	21
6.3.3.6	<code>_get_main_Egrid_pointer()</code>	22
6.3.3.7	<code>_get_MT_array_pointer()</code>	22
6.3.3.8	<code>_get_MT_numbers_pointer()</code>	22
6.3.3.9	<code>_get_MT_numbers_total_pointer()</code>	23
6.3.3.10	<code>_get_Q_pointer()</code>	23
6.3.3.11	<code>_get_scatter_data()</code>	23
6.3.3.12	<code>_get_temp_pointer()</code>	24
6.3.3.13	<code>_init_from_string()</code>	24
6.3.3.14	<code>_insert_reactions()</code>	24
6.3.3.15	<code>_interpolate()</code>	25
6.3.3.16	<code>_print_isotopes()</code>	25
6.3.3.17	<code>_read_tables()</code>	25
6.3.3.18	<code>_resolve_library()</code>	26
6.3.3.19	<code>_unionize()</code>	26
6.3.4	Member Data Documentation	26
6.3.4.1	<code>awr</code>	26
6.3.4.2	<code>datapath</code>	27
6.3.4.3	<code>isotope_list</code>	27
6.3.4.4	<code>isotropic_tol</code>	27
6.3.4.5	<code>last_loaded</code>	27
6.3.4.6	<code>libraries</code>	27
6.3.4.7	<code>MT_array</code>	27
6.3.4.8	<code>MT_E_grid</code>	28
6.3.4.9	<code>num_isotopes</code>	28

6.3.4.10	num_main_E	28
6.3.4.11	num_reactions	28
6.3.4.12	Q	28
6.3.4.13	reaction_numbers	28
6.3.4.14	reaction_numbers_total	29
6.3.4.15	tables	29
6.3.4.16	temp	29
6.3.4.17	xsdirstring	29
6.4	dist_container Struct Reference	29
6.4.1	Detailed Description	29
6.4.2	Member Data Documentation	30
6.4.2.1	lower	30
6.4.2.2	upper	30
6.5	dist_data Struct Reference	30
6.5.1	Detailed Description	30
6.5.2	Member Data Documentation	30
6.5.2.1	cdf	31
6.5.2.2	erg	31
6.5.2.3	intt	31
6.5.2.4	law	31
6.5.2.5	len	31
6.5.2.6	pdf	31
6.5.2.7	var	31
6.6	ace.DosimetryTable Class Reference	32
6.6.1	Constructor & Destructor Documentation	32
6.6.1.1	__init__()	32
6.6.2	Member Function Documentation	32
6.6.2.1	__repr__()	32
6.7	ace.ElectronTable Class Reference	33
6.7.1	Constructor & Destructor Documentation	33

6.7.1.1	<code>__init__()</code>	33
6.7.2	Member Function Documentation	33
6.7.2.1	<code>__repr__()</code>	33
6.8	<code>ace.EnergyDistribution</code> Class Reference	34
6.8.1	Constructor & Destructor Documentation	34
6.8.1.1	<code>__init__()</code>	34
6.9	<code>geom_data</code> Struct Reference	34
6.9.1	Detailed Description	35
6.9.2	Member Data Documentation	35
6.9.2.1	<code>cellnum</code>	35
6.9.2.2	<code>is_fissile</code>	35
6.9.2.3	<code>loc</code>	35
6.9.2.4	<code>matnum</code>	35
6.9.2.5	<code>max</code>	35
6.9.2.6	<code>min</code>	35
6.9.2.7	<code>talnum</code>	36
6.10	<code>intersection_point</code> Struct Reference	36
6.10.1	Detailed Description	36
6.10.2	Member Data Documentation	36
6.10.2.1	<code>cell</code>	36
6.10.2.2	<code>fiss</code>	37
6.10.2.3	<code>mat</code>	37
6.10.2.4	<code>norm</code>	37
6.10.2.5	<code>sense</code>	37
6.10.2.6	<code>surf_dist</code>	37
6.10.2.7	<code>tally_index</code>	37
6.10.2.8	<code>x</code>	37
6.10.2.9	<code>y</code>	37
6.10.2.10	<code>z</code>	38
6.11	<code>ace.Library</code> Class Reference	38

6.11.1 Detailed Description	38
6.11.2 Constructor & Destructor Documentation	39
6.11.2.1 __init__()	39
6.11.2.2 __del__()	39
6.11.3 Member Function Documentation	39
6.11.3.1 _read_ascii()	39
6.11.3.2 _read_binary()	39
6.11.3.3 find_table()	39
6.11.3.4 read()	40
6.11.4 Member Data Documentation	40
6.11.4.1 binary	40
6.11.4.2 f	40
6.11.4.3 tables	40
6.11.4.4 verbose	40
6.12 material_def Struct Reference	41
6.12.1 Detailed Description	41
6.12.2 Member Data Documentation	41
6.12.2.1 density	41
6.12.2.2 fractions	41
6.12.2.3 id	41
6.12.2.4 is_fissile	42
6.12.2.5 isotopes	42
6.12.2.6 matnum	42
6.12.2.7 num_isotopes	42
6.13 ace.NeutronDiscreteTable Class Reference	42
6.13.1 Constructor & Destructor Documentation	43
6.13.1.1 __init__()	43
6.13.2 Member Function Documentation	43
6.13.2.1 __repr__()	43
6.14 ace.NeutronMGTable Class Reference	43

6.14.1	Constructor & Destructor Documentation	44
6.14.1.1	__init__()	44
6.14.2	Member Function Documentation	44
6.14.2.1	__repr__()	44
6.15	ace.NeutronTable Class Reference	44
6.15.1	Detailed Description	46
6.15.2	Constructor & Destructor Documentation	47
6.15.2.1	__init__()	47
6.15.3	Member Function Documentation	47
6.15.3.1	__iter__()	47
6.15.3.2	__repr__()	47
6.15.3.3	_get_energy_distribution()	47
6.15.3.4	_read_all()	47
6.15.3.5	_read_andp()	48
6.15.3.6	_read_angular_distributions()	48
6.15.3.7	_read_cross_sections()	48
6.15.3.8	_read_energy_distributions()	48
6.15.3.9	_read_fis()	48
6.15.3.10	_read_gpd()	49
6.15.3.11	_read_landp()	49
6.15.3.12	_read_lsigp()	49
6.15.3.13	_read_mtrp()	49
6.15.3.14	_read_nu()	49
6.15.3.15	_read_sigp()	50
6.15.3.16	_read_unr()	50
6.15.3.17	_read_yp()	50
6.15.3.18	find_reaction()	50
6.15.4	Member Data Documentation	50
6.15.4.1	a_dist_energy_in	50
6.15.4.2	a_dist_mu_out	51

6.15.4.3 e_dist_EG	51
6.15.4.4 e_dist_energy_outNE	51
6.15.4.5 e_dist_LP	51
6.15.4.6 heating	51
6.15.4.7 IE_fission	51
6.15.4.8 MT_for_photon_yield	51
6.15.4.9 nu_d_energy	51
6.15.4.10 nu_d_energy_dist	52
6.15.4.11 nu_d_interp_INT	52
6.15.4.12 nu_d_interp_NBT	52
6.15.4.13 nu_d_precursor_const	52
6.15.4.14 nu_d_precursor_energy	52
6.15.4.15 nu_d_precursor_interp_INT	52
6.15.4.16 nu_d_precursor_interp_NBT	52
6.15.4.17 nu_d_precursor_prob	52
6.15.4.18 nu_d_value	53
6.15.4.19 nu_p_energy	53
6.15.4.20 nu_p_interp_INT	53
6.15.4.21 nu_p_interp_NBT	53
6.15.4.22 nu_p_type	53
6.15.4.23 nu_p_value	53
6.15.4.24 nu_t_energy	53
6.15.4.25 nu_t_interp_INT	53
6.15.4.26 nu_t_interp_NBT	54
6.15.4.27 nu_t_type	54
6.15.4.28 nu_t_value	54
6.15.4.29 photon_reactions	54
6.15.4.30 reactions	54
6.15.4.31 sigma	54
6.15.4.32 sigma_f	54

6.15.4.33	sigma_photon	54
6.15.4.34	urr_energy	55
6.15.4.35	urr_table	55
6.16	optix_stuff Class Reference	55
6.16.1	Detailed Description	56
6.16.2	Constructor & Destructor Documentation	56
6.16.2.1	optix_stuff() [1/2]	56
6.16.2.2	optix_stuff() [2/2]	57
6.16.2.3	~optix_stuff()	57
6.16.3	Member Function Documentation	57
6.16.3.1	get_outer_cell()	57
6.16.3.2	get_outer_cell_type()	57
6.16.3.3	get_rand()	58
6.16.3.4	init()	58
6.16.3.5	init_internal()	58
6.16.3.6	make_color()	59
6.16.3.7	make_geom_prim()	59
6.16.3.8	make_geom_xform()	59
6.16.3.9	make_geom_xform_common()	60
6.16.3.10	print()	60
6.16.3.11	set_trace_type()	60
6.16.3.12	trace() [1/3]	61
6.16.3.13	trace() [2/3]	61
6.16.3.14	trace() [3/3]	61
6.16.4	Member Data Documentation	61
6.16.4.1	accel_type	61
6.16.4.2	boundary_condition	62
6.16.4.3	cellnum_ptr	62
6.16.4.4	compute_device	62
6.16.4.5	context	62

6.16.4.6	done_ptr	62
6.16.4.7	GEOM_FLAG	62
6.16.4.8	matnum_ptr	62
6.16.4.9	maxcell	62
6.16.4.10	mincell	63
6.16.4.11	N	63
6.16.4.12	n_materials	63
6.16.4.13	optix_device	63
6.16.4.14	outer_cell	63
6.16.4.15	outer_cell_dims	63
6.16.4.16	outer_cell_type	63
6.16.4.17	positions_ptr	63
6.16.4.18	remap_ptr	64
6.16.4.19	rxn_ptr	64
6.16.4.20	stack_size_multiplier	64
6.16.4.21	talnum_ptr	64
6.16.4.22	traverse_type	64
6.17	particle_data Struct Reference	64
6.17.1	Detailed Description	65
6.17.2	Member Data Documentation	65
6.17.2.1	cellnum	65
6.17.2.2	E	65
6.17.2.3	index	65
6.17.2.4	isonum	66
6.17.2.5	matnum	66
6.17.2.6	Q	66
6.17.2.7	rn_bank	66
6.17.2.8	rxn	66
6.17.2.9	space	66
6.17.2.10	talnum	66

6.17.2.11 weight	66
6.17.2.12 yield	67
6.18 ace.PhotoatomicMGTable Class Reference	67
6.18.1 Constructor & Destructor Documentation	67
6.18.1.1 __init__()	67
6.18.2 Member Function Documentation	67
6.18.2.1 __repr__()	68
6.19 ace.PhotoatomicTable Class Reference	68
6.19.1 Constructor & Destructor Documentation	68
6.19.1.1 __init__()	68
6.19.2 Member Function Documentation	68
6.19.2.1 __repr__()	69
6.20 ace.PhotonuclearTable Class Reference	69
6.20.1 Constructor & Destructor Documentation	69
6.20.1.1 __init__()	69
6.20.2 Member Function Documentation	69
6.20.2.1 __repr__()	70
6.21 primitive Class Reference	70
6.21.1 Detailed Description	70
6.21.2 Constructor & Destructor Documentation	71
6.21.2.1 primitive() [1/2]	71
6.21.2.2 primitive() [2/2]	71
6.21.2.3 ~primitive()	71
6.21.3 Member Function Documentation	71
6.21.3.1 add_transform() [1/3]	72
6.21.3.2 add_transform() [2/3]	72
6.21.3.3 add_transform() [3/3]	72
6.21.3.4 make_hex_array()	73
6.21.3.5 print_transform() [1/2]	73
6.21.3.6 print_transform() [2/2]	73

6.21.4	Member Data Documentation	74
6.21.4.1	location	74
6.21.4.2	material	74
6.21.4.3	max	74
6.21.4.4	min	74
6.21.4.5	n_transforms	74
6.21.4.6	num_primitives	74
6.21.4.7	primitive_id	75
6.21.4.8	transforms	75
6.21.4.9	type	75
6.22	ace.Reaction Class Reference	75
6.22.1	Detailed Description	76
6.22.2	Constructor & Destructor Documentation	76
6.22.2.1	__init__()	77
6.22.3	Member Function Documentation	77
6.22.3.1	__repr__()	77
6.22.3.2	broaden()	77
6.22.3.3	threshold()	77
6.22.4	Member Data Documentation	77
6.22.4.1	IE	77
6.22.4.2	MT	78
6.22.4.3	Q	78
6.22.4.4	sigma	78
6.22.4.5	table	78
6.22.4.6	TY	78
6.23	ace.SabTable Class Reference	78
6.23.1	Detailed Description	79
6.23.2	Constructor & Destructor Documentation	80
6.23.2.1	__init__()	80
6.23.3	Member Function Documentation	80

6.23.3.1	<code>__repr__()</code>	80
6.23.3.2	<code>_read_all()</code>	80
6.23.3.3	<code>_read_itca()</code>	80
6.23.3.4	<code>_read_itce()</code>	81
6.23.3.5	<code>_read_itie()</code>	81
6.23.3.6	<code>_read_itxe()</code>	81
6.23.4	Member Data Documentation	81
6.23.4.1	<code>elastic_e_in</code>	81
6.23.4.2	<code>elastic_mu_out</code>	81
6.23.4.3	<code>elastic_P</code>	81
6.23.4.4	<code>elastic_type</code>	82
6.23.4.5	<code>inelastic_e_in</code>	82
6.23.4.6	<code>inelastic_e_out</code>	82
6.23.4.7	<code>inelastic_mu_out</code>	82
6.23.4.8	<code>inelastic_sigma</code>	82
6.24	<code>spatial_data</code> Struct Reference	82
6.24.1	Detailed Description	83
6.24.2	Member Data Documentation	83
6.24.2.1	<code>enforce_BC</code>	83
6.24.2.2	<code>norm</code>	83
6.24.2.3	<code>surf_dist</code>	83
6.24.2.4	<code>weight</code>	83
6.24.2.5	<code>x</code>	83
6.24.2.6	<code>xhat</code>	83
6.24.2.7	<code>y</code>	84
6.24.2.8	<code>yhat</code>	84
6.24.2.9	<code>z</code>	84
6.24.2.10	<code>zhat</code>	84
6.25	<code>tally_data</code> Struct Reference	84
6.25.1	Detailed Description	85

6.25.2	Member Data Documentation	85
6.25.2.1	cell	85
6.25.2.2	count	85
6.25.2.3	E_max	85
6.25.2.4	E_min	85
6.25.2.5	length	85
6.25.2.6	score	85
6.25.2.7	square	86
6.26	tally_data_host Struct Reference	86
6.26.1	Detailed Description	86
6.26.2	Member Data Documentation	86
6.26.2.1	cell	86
6.26.2.2	count	87
6.26.2.3	count_total	87
6.26.2.4	E_max	87
6.26.2.5	E_min	87
6.26.2.6	length	87
6.26.2.7	score	87
6.26.2.8	score_total	87
6.26.2.9	square	87
6.26.2.10	square_total	88
6.27	wfloat3 Class Reference	88
6.27.1	Detailed Description	89
6.27.2	Constructor & Destructor Documentation	89
6.27.2.1	wfloat3() [1/3]	89
6.27.2.2	wfloat3() [2/3]	89
6.27.2.3	wfloat3() [3/3]	89
6.27.3	Member Function Documentation	90
6.27.3.1	cross()	90
6.27.3.2	dot()	90

6.27.3.3	<code>norm2()</code>	90
6.27.3.4	<code>operator*()</code> [1/2]	91
6.27.3.5	<code>operator*()</code> [2/2]	91
6.27.3.6	<code>operator+()</code> [1/2]	91
6.27.3.7	<code>operator+()</code> [2/2]	92
6.27.3.8	<code>operator-()</code> [1/2]	92
6.27.3.9	<code>operator-()</code> [2/2]	93
6.27.3.10	<code>operator/()</code>	93
6.27.3.11	<code>rodrigues_rotation()</code>	93
6.27.3.12	<code>rotate()</code>	94
6.27.4	Member Data Documentation	94
6.27.4.1	<code>x</code>	94
6.27.4.2	<code>y</code>	94
6.27.4.3	<code>z</code>	94
6.28	wgeometry Class Reference	95
6.28.1	Detailed Description	96
6.28.2	Constructor & Destructor Documentation	96
6.28.2.1	<code>wgeometry()</code>	96
6.28.2.2	<code>~wgeometry()</code>	96
6.28.3	Member Function Documentation	96
6.28.3.1	<code>add_material()</code>	96
6.28.3.2	<code>add_primitive()</code> [1/2]	97
6.28.3.3	<code>add_primitive()</code> [2/2]	97
6.28.3.4	<code>add_tally()</code>	97
6.28.3.5	<code>add_transform()</code> [1/3]	97
6.28.3.6	<code>add_transform()</code> [2/3]	98
6.28.3.7	<code>add_transform()</code> [3/3]	98
6.28.3.8	<code>check()</code>	99
6.28.3.9	<code>check_fissile()</code>	99
6.28.3.10	<code>delete_primitive()</code>	99

6.28.3.11 delete_transform()	100
6.28.3.12 get_boundary_condition()	100
6.28.3.13 get_material_count()	100
6.28.3.14 get_material_table()	100
6.28.3.15 get_maximum_cell()	101
6.28.3.16 get_maximum_material()	101
6.28.3.17 get_minimum_cell()	101
6.28.3.18 get_minimum_material()	101
6.28.3.19 get_outer_cell()	101
6.28.3.20 get_outer_cell_dims()	101
6.28.3.21 get_outer_cell_type()	102
6.28.3.22 get_primitive_count()	102
6.28.3.23 get_transform_count()	102
6.28.3.24 make_hex_array()	102
6.28.3.25 make_material_table()	103
6.28.3.26 print_all()	103
6.28.3.27 print_materials_table()	103
6.28.3.28 print_summary()	103
6.28.3.29 set_datapath()	103
6.28.3.30 set_outer_cell()	104
6.28.3.31 update()	104
6.28.4 Member Data Documentation	104
6.28.4.1 awr_list	104
6.28.4.2 boundary_condition	104
6.28.4.3 cell_num_list	105
6.28.4.4 concentrations_matrix	105
6.28.4.5 datapath	105
6.28.4.6 fissile_flag	105
6.28.4.7 isotope_list	105
6.28.4.8 isotope_list_array	105

6.28.4.9 isotopes	105
6.28.4.10 material_list_array	105
6.28.4.11 material_num_list	106
6.28.4.12 materials	106
6.28.4.13 n_box	106
6.28.4.14 n_cyl	106
6.28.4.15 n_hex	106
6.28.4.16 n_isotopes	106
6.28.4.17 n_materials	106
6.28.4.18 n_primitives	106
6.28.4.19 n_sph	107
6.28.4.20 n_tallies	107
6.28.4.21 n_transforms	107
6.28.4.22 outer_cell	107
6.28.4.23 primitives	107
6.28.4.24 tally_cells	107
6.29 whistory Class Reference	107
6.29.1 Detailed Description	112
6.29.2 Constructor & Destructor Documentation	112
6.29.2.1 whistory()	112
6.29.2.2 ~whistory()	112
6.29.3 Member Function Documentation	112
6.29.3.1 accumulate_keff()	112
6.29.3.2 accumulate_tally()	113
6.29.3.3 bin_fission_points()	113
6.29.3.4 converge()	113
6.29.3.5 copy_data_to_device()	113
6.29.3.6 copy_energy_data()	114
6.29.3.7 copy_python_buffer() [1/4]	114
6.29.3.8 copy_python_buffer() [2/4]	114

6.29.3.9	<code>copy_python_buffer()</code> [3/4]	114
6.29.3.10	<code>copy_python_buffer()</code> [4/4]	115
6.29.3.11	<code>copy_scatter_data()</code>	115
6.29.3.12	<code>device_report()</code>	115
6.29.3.13	<code>get_Py_buffer_dims()</code>	115
6.29.3.14	<code>get_time()</code>	116
6.29.3.15	<code>hot2()</code>	116
6.29.3.16	<code>init()</code>	116
6.29.3.17	<code>init_cross_sections()</code>	116
6.29.3.18	<code>init_CUDPP()</code>	117
6.29.3.19	<code>init_device()</code>	117
6.29.3.20	<code>init_host()</code>	117
6.29.3.21	<code>init_python()</code>	117
6.29.3.22	<code>init_RNG()</code>	117
6.29.3.23	<code>make_color()</code>	117
6.29.3.24	<code>map_active()</code>	118
6.29.3.25	<code>nonzero()</code>	118
6.29.3.26	<code>plot_geom()</code>	118
6.29.3.27	<code>prep_secondaries()</code>	119
6.29.3.28	<code>print_banner()</code>	119
6.29.3.29	<code>print_materials_table()</code>	119
6.29.3.30	<code>print_pointers()</code>	119
6.29.3.31	<code>print_xs_data()</code>	119
6.29.3.32	<code>reduce_done()</code>	120
6.29.3.33	<code>reduce_weight()</code>	120
6.29.3.34	<code>reduce_yield()</code>	120
6.29.3.35	<code>remap_active()</code>	120
6.29.3.36	<code>reset_cycle()</code>	121
6.29.3.37	<code>reset_fixed()</code>	121
6.29.3.38	<code>run()</code>	121

6.29.3.39 sample_fissile_points()	122
6.29.3.40 set_acceration()	122
6.29.3.41 set_device()	122
6.29.3.42 set_dump_level()	122
6.29.3.43 set_filename()	123
6.29.3.44 set_print_level()	123
6.29.3.45 set_run_param()	123
6.29.3.46 set_run_type() [1/2]	123
6.29.3.47 set_run_type() [2/2]	124
6.29.3.48 trace() [1/2]	124
6.29.3.49 trace() [2/2]	124
6.29.3.50 update_RNG()	125
6.29.3.51 write_fission_points()	125
6.29.3.52 write_results()	125
6.29.3.53 write_tally()	125
6.29.3.54 write_to_file() [1/4]	125
6.29.3.55 write_to_file() [2/4]	127
6.29.3.56 write_to_file() [3/4]	127
6.29.3.57 write_to_file() [4/4]	128
6.29.3.58 write_xs_data()	128
6.29.4 Member Data Documentation	128
6.29.4.1 accel_type	128
6.29.4.2 blks	128
6.29.4.3 compact_config	129
6.29.4.4 compactplan	129
6.29.4.5 compute_device	129
6.29.4.6 d_bank_E	129
6.29.4.7 d_bank_space	129
6.29.4.8 d_edges	129
6.29.4.9 d_fissile_energy	129

6.29.4.10 d_fissile_points	129
6.29.4.11 d_isotope_list	130
6.29.4.12 d_material_list	130
6.29.4.13 d_num_active	130
6.29.4.14 d_num_completed	130
6.29.4.15 d_number_density_matrix	130
6.29.4.16 d_particles	130
6.29.4.17 d_reduced_weight	130
6.29.4.18 d_reduced_yields	130
6.29.4.19 d_remap	131
6.29.4.20 d_scanned	131
6.29.4.21 d_tally	131
6.29.4.22 d_valid_N	131
6.29.4.23 d_valid_result	131
6.29.4.24 d_xsdata	131
6.29.4.25 d_zeros	131
6.29.4.26 dh_dist_energy	131
6.29.4.27 dh_dist_scatter	132
6.29.4.28 dh_particles	132
6.29.4.29 dh_tally	132
6.29.4.30 dh_xsdata	132
6.29.4.31 dump_flag	132
6.29.4.32 edges	132
6.29.4.33 filename	132
6.29.4.34 fiss_img	132
6.29.4.35 fones	133
6.29.4.36 h_particles	133
6.29.4.37 h_tally	133
6.29.4.38 h_xsdata	133
6.29.4.39 initial_weight_total	133

6.29.4.40 is_initialized	133
6.29.4.41 isotope_list	133
6.29.4.42 isotopes	133
6.29.4.43 keff2_sum	134
6.29.4.44 keff_err	134
6.29.4.45 keff_sum	134
6.29.4.46 material_list	134
6.29.4.47 mem_free	134
6.29.4.48 mem_total	134
6.29.4.49 mones	134
6.29.4.50 MT_columns	134
6.29.4.51 MT_rows	135
6.29.4.52 N	135
6.29.4.53 n_cycles	135
6.29.4.54 n_edges	135
6.29.4.55 n_isotopes	135
6.29.4.56 n_materials	135
6.29.4.57 n_skip	135
6.29.4.58 n_tallies	135
6.29.4.59 Ndataset	136
6.29.4.60 NUM_THREADS	136
6.29.4.61 number_density_matrix	136
6.29.4.62 ones	136
6.29.4.63 outer_cell	136
6.29.4.64 outer_cell_dims	136
6.29.4.65 outer_cell_type	136
6.29.4.66 print_flag	136
6.29.4.67 problem_geom	137
6.29.4.68 radix_config	137
6.29.4.69 radixplan	137

6.29.4.70	rand_gen	137
6.29.4.71	redu_float_config	137
6.29.4.72	redu_int_config	137
6.29.4.73	reduced_weight	137
6.29.4.74	reduced_weight_total	137
6.29.4.75	reduced_yields	138
6.29.4.76	reduced_yields_total	138
6.29.4.77	reduplan_float	138
6.29.4.78	reduplan_int	138
6.29.4.79	remap	138
6.29.4.80	res	138
6.29.4.81	RUN_FLAG	138
6.29.4.82	scan_int_config	138
6.29.4.83	scanplan_int	139
6.29.4.84	stream	139
6.29.4.85	theCudpp	139
6.29.4.86	total_bytes_energy	139
6.29.4.87	total_bytes_scatter	139
6.29.4.88	xs_isotope_ints	139
6.29.4.89	xs_num_rxns	139
6.29.4.90	xsdatt_instance	139
6.29.4.91	zeros	140
6.30	wtransform Struct Reference	140
6.30.1	Detailed Description	140
6.30.2	Member Data Documentation	140
6.30.2.1	cellmat	140
6.30.2.2	cellnum	141
6.30.2.3	dx	141
6.30.2.4	dy	141
6.30.2.5	dz	141
6.30.2.6	phi	141
6.30.2.7	tally_index	141
6.30.2.8	theta	141

7 File Documentation	143
7.1 ace.pyx File Reference	143
7.2 benchmarks.cpp File Reference	144
7.2.1 Function Documentation	144
7.2.1.1 main()	144
7.3 box_mesh.cu File Reference	144
7.3.1 Function Documentation	145
7.3.1.1 bounds()	145
7.3.1.2 boxnormal()	145
7.3.1.3 intersect()	145
7.3.1.4 rtDeclareVariable() [1/7]	145
7.3.1.5 rtDeclareVariable() [2/7]	145
7.3.1.6 rtDeclareVariable() [3/7]	146
7.3.1.7 rtDeclareVariable() [4/7]	146
7.3.1.8 rtDeclareVariable() [5/7]	146
7.3.1.9 rtDeclareVariable() [6/7]	146
7.3.1.10 rtDeclareVariable() [7/7]	146
7.3.2 Variable Documentation	146
7.3.2.1 dims	147
7.4 camera.cu File Reference	147
7.4.1 Function Documentation	147
7.4.1.1 camera()	147
7.4.1.2 exception()	147
7.4.1.3 rtDeclareVariable() [1/6]	148
7.4.1.4 rtDeclareVariable() [2/6]	148
7.4.1.5 rtDeclareVariable() [3/6]	148
7.4.1.6 rtDeclareVariable() [4/6]	148
7.4.1.7 rtDeclareVariable() [5/6]	148
7.4.1.8 rtDeclareVariable() [6/6]	148
7.4.2 Variable Documentation	149

7.4.2.1	cellnum_buffer	149
7.4.2.2	matnum_buffer	149
7.4.2.3	positions_buffer	149
7.4.2.4	remap_buffer	149
7.4.2.5	rxn_buffer	149
7.4.2.6	talnum_buffer	149
7.5	check_cuda.h File Reference	149
7.5.1	Macro Definition Documentation	150
7.5.1.1	check_cuda	150
7.5.2	Function Documentation	150
7.5.2.1	check_cuda()	150
7.6	check_pointers.cu File Reference	150
7.6.1	Function Documentation	151
7.6.1.1	check_pointers()	151
7.6.1.2	check_pointers_kernel()	151
7.7	copy_points.cu File Reference	151
7.7.1	Function Documentation	152
7.7.1.1	copy_points()	152
7.7.1.2	copy_points_kernel()	152
7.8	cylinder_mesh.cu File Reference	153
7.8.1	Function Documentation	153
7.8.1.1	bounds()	153
7.8.1.2	intersect()	154
7.8.1.3	rtDeclareVariable() [1/7]	154
7.8.1.4	rtDeclareVariable() [2/7]	154
7.8.1.5	rtDeclareVariable() [3/7]	154
7.8.1.6	rtDeclareVariable() [4/7]	154
7.8.1.7	rtDeclareVariable() [5/7]	154
7.8.1.8	rtDeclareVariable() [6/7]	155
7.8.1.9	rtDeclareVariable() [7/7]	155

7.8.2	Variable Documentation	155
7.8.2.1	dims	155
7.9	datadef.h File Reference	155
7.10	device_copies.cu File Reference	156
7.10.1	Function Documentation	156
7.10.1.1	allocate_on_device()	156
7.10.1.2	copy_from_device()	156
7.10.1.3	copy_to_device()	157
7.10.1.4	deallocate_on_device()	157
7.11	device_copies.h File Reference	157
7.11.1	Function Documentation	158
7.11.1.1	allocate_on_device()	158
7.11.1.2	copy_from_device()	158
7.11.1.3	copy_to_device()	158
7.11.1.4	deallocate_on_device()	159
7.12	find_E_grid_index.cu File Reference	159
7.12.1	Function Documentation	159
7.12.1.1	find_E_grid_index()	160
7.12.1.2	find_E_grid_index_kernel()	160
7.13	fission.cu File Reference	160
7.13.1	Function Documentation	161
7.13.1.1	fission()	161
7.13.1.2	fission_kernel()	161
7.14	hex_mesh.cu File Reference	162
7.14.1	Function Documentation	162
7.14.1.1	accept_point()	162
7.14.1.2	bounds()	162
7.14.1.3	get_t()	163
7.14.1.4	intersect()	163
7.14.1.5	rtDeclareVariable() [1/8]	163

7.14.1.6	rtDeclareVariable() [2/8]	163
7.14.1.7	rtDeclareVariable() [3/8]	163
7.14.1.8	rtDeclareVariable() [4/8]	163
7.14.1.9	rtDeclareVariable() [5/8]	164
7.14.1.10	rtDeclareVariable() [6/8]	164
7.14.1.11	rtDeclareVariable() [7/8]	164
7.14.1.12	rtDeclareVariable() [8/8]	164
7.14.2	Variable Documentation	164
7.14.2.1	dims	164
7.15	hits_mesh.cu File Reference	164
7.15.1	Function Documentation	165
7.15.1.1	closest_hit()	165
7.15.1.2	rtDeclareVariable() [1/9]	165
7.15.1.3	rtDeclareVariable() [2/9]	165
7.15.1.4	rtDeclareVariable() [3/9]	165
7.15.1.5	rtDeclareVariable() [4/9]	166
7.15.1.6	rtDeclareVariable() [5/9]	166
7.15.1.7	rtDeclareVariable() [6/9]	166
7.15.1.8	rtDeclareVariable() [7/9]	166
7.15.1.9	rtDeclareVariable() [8/9]	166
7.15.1.10	rtDeclareVariable() [9/9]	166
7.16	macro_micro.cu File Reference	167
7.16.1	Function Documentation	167
7.16.1.1	macro_micro()	167
7.16.1.2	macro_micro_kernel()	168
7.17	main.cpp File Reference	168
7.17.1	Function Documentation	168
7.17.1.1	main()	168
7.18	miss.cu File Reference	169
7.18.1	Function Documentation	169

7.18.1.1	miss()	169
7.18.1.2	rtDeclareVariable() [1/5]	169
7.18.1.3	rtDeclareVariable() [2/5]	169
7.18.1.4	rtDeclareVariable() [3/5]	170
7.18.1.5	rtDeclareVariable() [4/5]	170
7.18.1.6	rtDeclareVariable() [5/5]	170
7.18.2	Variable Documentation	170
7.18.2.1	cellnum_buffer	170
7.18.2.2	matnum_buffer	170
7.18.2.3	positions_buffer	170
7.18.2.4	remap_buffer	171
7.18.2.5	rxn_buffer	171
7.19	mt19937ar.cpp File Reference	171
7.19.1	Macro Definition Documentation	171
7.19.1.1	LOWER_MASK	172
7.19.1.2	M	172
7.19.1.3	MATRIX_A	172
7.19.1.4	N	172
7.19.1.5	UPPER_MASK	172
7.19.2	Function Documentation	172
7.19.2.1	genrand_int31()	172
7.19.2.2	genrand_int32()	172
7.19.2.3	genrand_real1()	173
7.19.2.4	genrand_real2()	173
7.19.2.5	genrand_real3()	173
7.19.2.6	genrand_res53()	173
7.19.2.7	init_by_array()	173
7.19.2.8	init_genrand()	173
7.19.3	Variable Documentation	173
7.19.3.1	mt	174

7.19.3.2 mti	174
7.20 optix_stuff.cpp File Reference	174
7.21 optix_stuff.h File Reference	174
7.22 optixtest.cpp File Reference	174
7.22.1 Function Documentation	175
7.22.1.1 main()	175
7.23 pop_fission.cu File Reference	175
7.23.1 Function Documentation	175
7.23.1.1 pop_fission()	175
7.23.1.2 pop_fission_kernel()	176
7.24 reaction_edges3.cu File Reference	176
7.24.1 Function Documentation	176
7.24.1.1 reaction_edges()	176
7.24.1.2 reaction_edges_kernel()	177
7.25 rebase_yield.cu File Reference	177
7.25.1 Function Documentation	177
7.25.1.1 rebase_yield()	177
7.25.1.2 rebase_yield_kernel()	178
7.26 safety_check.cu File Reference	178
7.26.1 Function Documentation	178
7.26.1.1 safety_check()	178
7.26.1.2 safety_check_kernel()	179
7.27 sample_fissile_energy.cu File Reference	179
7.27.1 Function Documentation	179
7.27.1.1 sample_fissile_energy()	179
7.27.1.2 sample_fissile_energy_kernel()	180
7.28 sample_fixed_source.cu File Reference	180
7.28.1 Function Documentation	180
7.28.1.1 sample_fixed_source()	180
7.28.1.2 sample_fixed_source_kernel()	181

7.29	scatter_conti.cu File Reference	181
7.29.1	Function Documentation	181
7.29.1.1	scatter_conti()	182
7.29.1.2	scatter_conti_kernel()	182
7.30	scatter_level.cu File Reference	182
7.30.1	Function Documentation	183
7.30.1.1	scatter_level()	183
7.30.1.2	scatter_level_kernel()	183
7.31	scatter_multi.cu File Reference	183
7.31.1	Function Documentation	184
7.31.1.1	scatter_multi()	184
7.31.1.2	scatter_multi_kernel()	184
7.32	set_positions_rand.cu File Reference	185
7.32.1	Function Documentation	185
7.32.1.1	set_positions_rand()	185
7.32.1.2	set_positions_rand_kernel()	185
7.33	sphere_mesh.cu File Reference	186
7.33.1	Function Documentation	186
7.33.1.1	bounds()	186
7.33.1.2	intersect()	187
7.33.1.3	rtDeclareVariable() [1/7]	187
7.33.1.4	rtDeclareVariable() [2/7]	187
7.33.1.5	rtDeclareVariable() [3/7]	187
7.33.1.6	rtDeclareVariable() [4/7]	187
7.33.1.7	rtDeclareVariable() [5/7]	187
7.33.1.8	rtDeclareVariable() [6/7]	188
7.33.1.9	rtDeclareVariable() [7/7]	188
7.33.2	Variable Documentation	188
7.33.2.1	dims	188
7.34	test_function.cu File Reference	188

7.34.1	Function Documentation	188
7.34.1.1	test_function()	189
7.34.1.2	test_kernel()	189
7.35	unionize.py File Reference	189
7.36	warp.h File Reference	189
7.37	warp_cuda.h File Reference	190
7.37.1	Function Documentation	190
7.37.1.1	check_pointers()	191
7.37.1.2	copy_points()	191
7.37.1.3	find_E_grid_index()	192
7.37.1.4	fission()	192
7.37.1.5	macro_micro()	193
7.37.1.6	pop_fission()	193
7.37.1.7	reaction_edges()	194
7.37.1.8	rebase_yield()	194
7.37.1.9	safety_check()	195
7.37.1.10	sample_fissile_energy()	195
7.37.1.11	sample_fixed_source()	196
7.37.1.12	scatter_conti()	196
7.37.1.13	scatter_level()	196
7.37.1.14	scatter_multi()	197
7.37.1.15	set_positions_rand()	197
7.37.1.16	write_to_file()	198
7.38	warp_device.cuh File Reference	198
7.39	wfloat3.h File Reference	198
7.40	wgeometry.cpp File Reference	198
7.41	wgeometry.h File Reference	199
7.42	whistory.cpp File Reference	199
7.42.1	Variable Documentation	199
7.42.1.1	optix_obj	200
7.43	whistory.h File Reference	200
7.44	wprimitive.cpp File Reference	200
7.45	wprimitive.h File Reference	200
7.46	write_to_file.cu File Reference	200
7.46.1	Function Documentation	201
7.46.1.1	write_to_file()	201

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

ace	9
unionize	12

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

cross_section_data	15
unionize.cross_section_data	17
dist_container	29
dist_data	30
geom_data	34
intersection_point	36
material_def	41
object	
ace.AceTable	13
ace.DosimetryTable	32
ace.ElectronTable	33
ace.NeutronDiscreteTable	42
ace.NeutronMGTable	43
ace.NeutronTable	44
ace.PhotoatomicMGTable	67
ace.PhotoatomicTable	68
ace.PhotonuclearTable	69
ace.SabTable	78
ace.EnergyDistribution	34
ace.Library	38
ace.Reaction	75
optix_stuff	55
particle_data	64
primitive	70
spatial_data	82
tally_data	84
tally_data_host	86
wfloat3	88
wgeometry	95
whistory	107
wtransform	140

Chapter 3

Class Index

3.1 Class List

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

ace.AceTable	13
cross_section_data	
Structure that holds the topmost level of cross section data	15
unionize.cross_section_data	
Handles cross section data	17
dist_container	
Container for pointers that map the nearest distributions to the energy grid point where it resides	29
dist_data	
Information that defines an ENDF cross section distribution	30
ace.DosimetryTable	32
ace.ElectronTable	33
ace.EnergyDistribution	34
geom_data	
Parameters of a wgeometry	34
intersection_point	
Information pertinent to an intersection point, used in OptiX	36
ace.Library	38
material_def	
Information that defines a material	41
ace.NeutronDiscreteTable	42
ace.NeutronMGTable	43
ace.NeutronTable	44
optix_stuff	
OptiX stuff class	55
particle_data	
Structure that holds all the arrays that define a particle's state	64
ace.PhotoatomicMGTable	67
ace.PhotoatomicTable	68
ace.PhotonuclearTable	69
primitive	
Primitive class	70
ace.Reaction	75
ace.SabTable	78
spatial_data	
Spatial parameters of the neutron	82

tally_data	Tally data that lives on the device side	84
tally_data_host	Tally data that lives on the host side	86
wfloat3	Class definitions for device vector operations	88
wgeometry	Wgeometry class	95
whistory	Whistory class	107
wtransform	Parameters of a wtransform, a transform used to create a new instance of a wgeometry object . .	140

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

ace.pyx	143
benchmarks.cpp	144
box_mesh.cu	144
camera.cu	147
check_cuda.h	149
check_pointers.cu	150
copy_points.cu	151
cylinder_mesh.cu	153
datadef.h	155
device_copies.cu	156
device_copies.h	157
find_E_grid_index.cu	159
fission.cu	160
hex_mesh.cu	162
hits_mesh.cu	164
macro_micro.cu	167
main.cpp	168
miss.cu	169
mt19937ar.cpp	171
optix_stuff.cpp	174
optix_stuff.h	174
optixtest.cpp	174
pop_fission.cu	175
reaction_edges3.cu	176
rebase_yield.cu	177
safety_check.cu	178
sample_fissile_energy.cu	179
sample_fixed_source.cu	180
scatter_conti.cu	181
scatter_level.cu	182
scatter_multi.cu	183
set_positions_rand.cu	185
sphere_mesh.cu	186
test_function.cu	188
unionize.py	189

warp.h	189
warp_cuda.h	190
warp_device.cuh	198
wfloat3.h	198
wgeometry.cpp	198
wgeometry.h	199
whistory.cpp	199
whistory.h	200
wprimitive.cpp	200
wprimitive.h	200
write_to_file.cu	200

Chapter 5

Namespace Documentation

5.1 ace Namespace Reference

Classes

- class [AceTable](#)
- class [DosimetryTable](#)
- class [ElectronTable](#)
- class [EnergyDistribution](#)
- class [Library](#)
- class [NeutronDiscreteTable](#)
- class [NeutronMGTable](#)
- class [NeutronTable](#)
- class [PhotoatomicMGTable](#)
- class [PhotoatomicTable](#)
- class [PhotonuclearTable](#)
- class [Reaction](#)
- class [SabTable](#)

Functions

- def [fromstring_split](#) (s, sep=None, dtype=float)
- def [fromstring_token](#) (s, sep=" ", bint, inplace=False, int, maxsize=-1)
- def [ascii_to_binary](#) (ascii_file, binary_file)

Variables

- [NP_LE_V15](#)
- dictionary [table_types](#)

5.1.1 Detailed Description

This module is for reading ACE-format cross sections. ACE stands for "A Compact ENDF" format and originated from work on MCNP_. It is used in a number of other Monte Carlo particle transport codes.

ACE-format cross sections are typically generated from ENDF_ files through a cross section processing program like NJOY_. The ENDF data consists of tabulated thermal data, ENDF/B resonance parameters, distribution parameters in the unresolved resonance region, and tabulated data in the fast region. After the ENDF data has been reconstructed and Doppler-broadened, the ACER module generates ACE-format cross sections.

```
.. _MCNP: https://laws.lanl.gov/vhosts/mcnp.lanl.gov/
.. _NJOY: http://t2.lanl.gov/codes.shtml
.. _ENDF: http://www.nndc.bnl.gov/endl
```

```
.. moduleauthor:: Paul Romano <paul.k.romano@gmail.com>, Anthony Scopatz <scopatz@gmail.com>
```

5.1.2 Function Documentation

5.1.2.1 `ascii_to_binary()`

```
def ace.ascii_to_binary (
    ascii_file,
    binary_file )
```

Convert an ACE file in ASCII format (type 1) to binary format (type 2).

```
Parameters
-----
ascii_file : str
    Filename of ASCII ACE file
binary_file : str
    Filename of binary ACE file to be written
```

5.1.2.2 `fromstring_split()`

```
def ace.fromstring_split (
    s,
    sep = None,
    dtype = float )
```

A replacement for `numpy.fromstring()` using the Python `str.split()` and `np.array()`.

```
Parameters
-----
s : str
    String of data.
sep : str or None
    String of separator characters, has the same meaning as in
    str.split().
```

dtype : np.dtype
Numpy dtype to cast elements enough.

Returns

data : ndarray, 1d
Will always return a 1d array of dtype. You must reshape to the appropriate shape.

See Also

fromstring_token : May faster depending on the data.

5.1.2.3 fromstring_token()

```
def ace.fromstring_token (
    s,
    sep = " ",
    bint,
    inplace = False,
    int,
    maxsize = -1 )
```

A replacement for numpy.fromstring() using the C standard library atof() and strtok() functions.

Parameters

s : str
String of data.
sep : str
String of separator characters. Unlike numpy.fromstring(), all characters are separated on independently.
inplace : bool
Whether s should tokenized in-place or whether a copy should be made. If done in-place, the first instance of sep between any tokens will be replaced with the NULL character.
maxsize : int
Specifies the size of the array to pre-allocate. If negative, this will be set to the maximum possible number of elements, ie len(s)/2 + 1.

Returns

data : ndarray, 1d, float64
Will always return a 1d float64 array. You must cast and reshape to the appropriate type and shape.

See Also

fromstring_split : May faster depending on the data.

5.1.3 Variable Documentation

5.1.3.1 NP_LE_V15

ace.NP_LE_V15

5.1.3.2 table_types

dictionary `ace.table_types`

Initial value:

```
1 = {
2     "c": NeutronTable,
3     "t": SabTable,
4     "y": DosimetryTable,
5     "d": NeutronDiscreteTable,
6     "p": PhotoatomicTable,
7     "m": NeutronMGTable,
8     "g": PhotoatomicMGTable,
9     "e": ElectronTable,
10    "u": PhotonuclearTable}
```

5.2 unionize Namespace Reference

Classes

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

Variables

- bool [ace_available](#) = True

5.2.1 Variable Documentation

5.2.1.1 ace_available

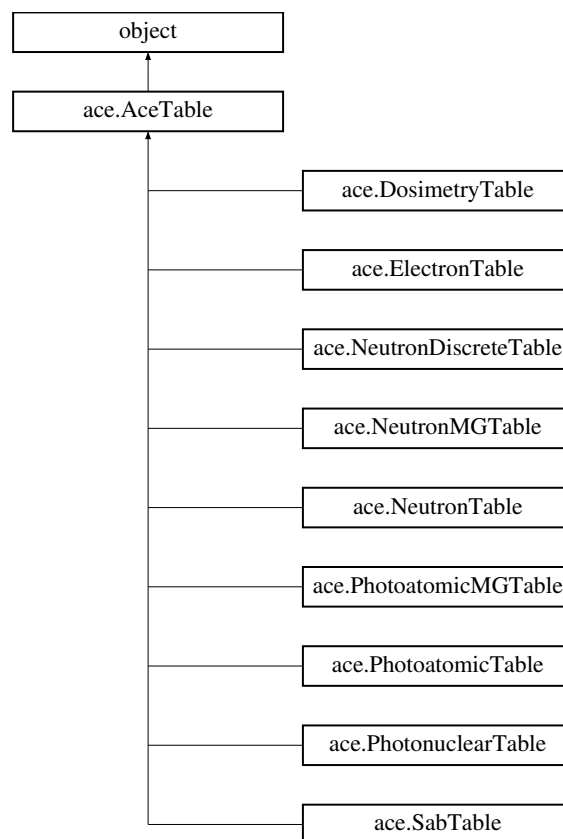
```
bool unionize.ace_available = True
```

Chapter 6

Class Documentation

6.1 ace.AceTable Class Reference

Inheritance diagram for ace.AceTable:



Public Member Functions

- `def __init__(self, name, awr, temp)`

Public Attributes

- [name](#)
- [awr](#)
- [temp](#)

Private Member Functions

- [def _read_all](#) (self)

6.1.1 Detailed Description

Abstract superclass of all other classes for cross section tables.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def ace.AceTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.1.3 Member Function Documentation

6.1.3.1 _read_all()

```
def ace.AceTable._read_all (
    self ) [private]
```

6.1.4 Member Data Documentation

6.1.4.1 awr

```
ace.AceTable.awr
```

6.1.4.2 name

```
ace.AceTable.name
```

6.1.4.3 temp

```
ace.AceTable.temp
```

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

- [ace.pyx](#)

6.2 cross_section_data Struct Reference

structure that holds the topmost level of cross section data

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

Public Attributes

- unsigned [n_isotopes](#)
- unsigned [energy_grid_len](#)
- unsigned [total_reaction_channels](#)
- unsigned * [rxn_numbers](#)
- unsigned * [rxn_numbers_total](#)
- float * [energy_grid](#)
- float * [Q](#)
- float * [xs](#)
- float * [awr](#)
- float * [temp](#)
- [dist_container](#) * [dist_scatter](#)
- [dist_container](#) * [dist_energy](#)

6.2.1 Detailed Description

structure that holds the topmost level of cross section data

contains pointers and parameters to compute any reaction in this requested isotope set - number of isotopes, length of main energy grid, total number of reactions channels, reaction number vector, total reaction channels for each isotope, unionized energy grid vector, reaction Q values, cross section data, isotope atomic weight ratios, isotope temperatures, energy and scattering data distributions

6.2.2 Member Data Documentation

6.2.2.1 awr

`float* cross_section_data::awr`

isotope atomic weight ratio (AWR) list

6.2.2.2 dist_energy

`dist_container* cross_section_data::dist_energy`

energy distribution data redirection matrix

6.2.2.3 dist_scatter

`dist_container* cross_section_data::dist_scatter`

scattering distribution data redirection matrix

6.2.2.4 energy_grid

`float* cross_section_data::energy_grid`

unionized energy grid vector

6.2.2.5 energy_grid_len

`unsigned cross_section_data::energy_grid_len`

length of main energy grid

6.2.2.6 n_isotopes

`unsigned cross_section_data::n_isotopes`

number of isotopes

6.2.2.7 Q

`float* cross_section_data::Q`

reaction Q values

6.2.2.8 rxn_numbers

`unsigned* cross_section_data::rxn_numbers`

reaction number vector

6.2.2.9 rxn_numbers_total

```
unsigned* cross_section_data::rxn_numbers_total
```

total reaction channels for each isotope

6.2.2.10 temp

```
float* cross_section_data::temp
```

isotope temperature list (MeV)

6.2.2.11 total_reaction_channels

```
unsigned cross_section_data::total_reaction_channels
```

total number of reactions channels

6.2.2.12 xs

```
float* cross_section_data::xs
```

cross section data matrix

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

- [datadef.h](#)

6.3 unionize.cross_section_data Class Reference

handles cross section data

Public Member Functions

- `def __init__ (self)`

cross section data class to hold it all together with the functions necessary for WARP

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.
- [isotropic_tol](#)
isotropic tolerance
- [xsdirstring](#)

Private Member Functions

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

- def [__allocate_arrays](#) (self)
array allocation function
- def [__interpolate](#) (self)
interpolation function
- def [__get_MT_numbers_pointer](#) (self)
gets pointer to MT numbers
- def [__get_awr_pointer](#) (self)
gets pointer to AWR values
- def [__get_temp_pointer](#) (self)
gets pointer to temperature values
- def [__get_Q_pointer](#) (self)
gets pointer to Q-values
- def [__get_MT_array_pointer](#) (self)
gets pointer to MT numbers
- def [__get_main_Egrid_pointer](#) (self)
gets pointer to main energy grid
- def [__get_length_numbers_pointer](#) (self)
creates array of size number of isotopes + main energy grid + number of reactions
- def [__get_MT_numbers_total_pointer](#) (self)
gets pointer to total MT numbers
- def [__print_isotopes](#) (self)
prints list of isotopes in a material
- def [__get_scatter_data](#) (self, row, col)
gets table of scattering data
- def [__get_energy_data](#) (self, row, col)
gets table of energy data

6.3.1 Detailed Description

handles cross section data

6.3.2 Constructor & Destructor Documentation

6.3.2.1 `__init__()`

```
def unionize.cross_section_data.__init__ (
    self )
```

cross section data class to hold it all together with the functions necessary for WARP

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

in	<i>self</i>	- this cross_section_data object
----	-------------	--

6.3.3 Member Function Documentation**6.3.3.1 _add_isotope()**

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

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

Parameters

in	<i>self</i>	- this cross_section_data object
in	<i>isotope</i>	- isotope to be appended

6.3.3.2 _allocate_arrays()

```
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

in	<i>self</i>	- this cross_section_data object
----	-------------	--

6.3.3.3 _get_awr_pointer()

```
def unionize.cross_section_data._get_awr_pointer (
    self ) [private]
```

gets pointer to AWR values

Parameters

in	<i>self</i>	- this cross_section_data object
----	-------------	--

Returns

AWR_array - array of AWR values

6.3.3.4 _get_energy_data()

```
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

in	<i>self</i>	- this cross_section_data object
in	<i>row</i>	- point in energy grid
in	<i>col</i>	- MT number

6.3.3.5 _get_length_numbers_pointer()

```
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

in	<i>self</i>	- this cross_section_data object
----	-------------	--

Returns

lengths - lengths array

6.3.3.6 `_get_main_Egrid_pointer()`

```
def unionize.cross_section_data._get_main_Egrid_pointer (
    self ) [private]
```

gets pointer to main energy grid

Parameters

in	<i>self</i>	- this cross_section_data object
----	-------------	--

Returns

E_grid - array of energy grid points

6.3.3.7 `_get_MT_array_pointer()`

```
def unionize.cross_section_data._get_MT_array_pointer (
    self ) [private]
```

gets pointer to MT numbers

Parameters

in	<i>self</i>	- this cross_section_data object
----	-------------	--

Returns

MT_array - array of MT numbers

6.3.3.8 `_get_MT_numbers_pointer()`

```
def unionize.cross_section_data._get_MT_numbers_pointer (
    self ) [private]
```

gets pointer to MT numbers

Parameters

in	<i>self</i>	- this cross_section_data object
----	-------------	--

Returns

MT_num_array - array of MT numbers

6.3.3.9 `_get_MT_numbers_total_pointer()`

```
def unionize.cross_section_data._get_MT_numbers_total_pointer (
    self ) [private]
```

gets pointer to total MT numbers

Parameters

in	self	- this cross_section_data object
----	------	--

Returns

numbers - array of total MT numbers

6.3.3.10 `_get_Q_pointer()`

```
def unionize.cross_section_data._get_Q_pointer (
    self ) [private]
```

gets pointer to Q-values

Parameters

in	self	- this cross_section_data object
----	------	--

Returns

Q_array - array of Q-values

6.3.3.11 `_get_scatter_data()`

```
def unionize.cross_section_data._get_scatter_data (
    self,
    row,
    col ) [private]
```

gets table of scattering data

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

Parameters

in	<i>self</i>	- this cross_section_data object
in	<i>row</i>	- point in energy grid
in	<i>col</i>	- MT number

6.3.3.12 `_get_temp_pointer()`

```
def unionize.cross_section_data._get_temp_pointer (
    self ) [private]
```

gets pointer to temperature values

Parameters

in	<i>self</i>	- this cross_section_data object
----	-------------	--

Returns

`temp_array` - array of temperature values

6.3.3.13 `_init_from_string()`

```
def unionize.cross_section_data._init_from_string (
    self,
    this_string ) [private]
```

initializes material from isotope list string

Parameters

in	<i>self</i>	- this cross_section_data object
in	<i>this_string</i>	- comma-separated isotope list

6.3.3.14 `_insert_reactions()`

```
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

in	<i>self</i>	- this cross_section_data object
----	-------------	--

6.3.3.15 `_interpolate()`

```
def unionize.cross_section_data._interpolate (
    self ) [private]
```

interpolation function

linearly interpolates the cross sections for each isotope in a material

Parameters

in	<i>self</i>	- this cross_section_data object
----	-------------	--

6.3.3.16 `_print_isotopes()`

```
def unionize.cross_section_data._print_isotopes (
    self ) [private]
```

prints list of isotopes in a material

Parameters

in	<i>self</i>	- material for which to print isotope list
----	-------------	--

6.3.3.17 `_read_tables()`

```
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

in	<i>self</i>	- this cross_section_data object
in	<i>datapath</i> ↔	- the input datapath for the xs data
	<i>in</i>	

6.3.3.18 `_resolve_library()`

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

finds path the isotope files

finds the relative path from DATAPATH to a file that contains the specified isotope cross sections. It reads the xsdir file to do this.

Parameters

in	<i>self</i>	- this cross_section_data object
in	<i>tope</i>	- isotope to seach for

6.3.3.19 `_unionize()`

```
def unionize.cross_section_data._unionize (
    self ) [private]
```

unionization function

unionizes MT energy grid.

Parameters

in	<i>self</i>	- this cross_section_data object
----	-------------	--

6.3.4 Member Data Documentation

6.3.4.1 `awr`

```
unionize.cross_section_data.awr
```

AWR array.

6.3.4.2 datapath

`unionize.cross_section_data.datapath`

data path

6.3.4.3 isotope_list

`unionize.cross_section_data.isotope_list`

isotope list

6.3.4.4 isotropic_tol

`unionize.cross_section_data.isotropic_tol`

isotropic tolerance

6.3.4.5 last_loaded

`unionize.cross_section_data.last_loaded`

Last valid table loaded.

6.3.4.6 libraries

`unionize.cross_section_data.libraries`

cross section libraries

6.3.4.7 MT_array

`unionize.cross_section_data.MT_array`

MT number array.

6.3.4.8 MT_E_grid

`unionize.cross_section_data.MT_E_grid`

MT energy grid.

6.3.4.9 num_isotopes

`unionize.cross_section_data.num_isotopes`

number of isotopes

6.3.4.10 num_main_E

`unionize.cross_section_data.num_main_E`

main energy

6.3.4.11 num_reactions

`unionize.cross_section_data.num_reactions`

number of reactions

6.3.4.12 Q

`unionize.cross_section_data.Q`

Q-value array.

6.3.4.13 reaction_numbers

`unionize.cross_section_data.reaction_numbers`

reaction numbers array

6.3.4.14 reaction_numbers_total

`unionize.cross_section_data.reaction_numbers_total`

total reaction numbers array

6.3.4.15 tables

`unionize.cross_section_data.tables`

cross section tables

6.3.4.16 temp

`unionize.cross_section_data.temp`

temp array

6.3.4.17 xsdirstring

`unionize.cross_section_data.xsdirstring`

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

- [unionize.py](#)

6.4 dist_container Struct Reference

container for pointers that map the nearest distributions to the energy grid point where it resides

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

Public Attributes

- [dist_data](#) * [lower](#)
- [dist_data](#) * [upper](#)

6.4.1 Detailed Description

container for pointers that map the nearest distributions to the energy grid point where it resides

pointers to the nearest distribution lower and greater in energy

6.4.2 Member Data Documentation

6.4.2.1 lower

`dist_data* dist_container::lower`

pointer to distribution data of grid point below current energy

6.4.2.2 upper

`dist_data* dist_container::upper`

pointer to distribution data of grid point above current energy

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

- [datadef.h](#)

6.5 dist_data Struct Reference

contains information that defines an ENDF cross section distribution

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

Public Attributes

- float [erg](#)
- unsigned [len](#)
- unsigned [law](#)
- unsigned [intt](#)
- float * [var](#)
- float * [pdf](#)
- float * [cdf](#)

6.5.1 Detailed Description

contains information that defines an ENDF cross section distribution

ernergy, length, law, interpolation type, variable/pdf/cdf arrays

6.5.2 Member Data Documentation

6.5.2.1 cdf

```
float* dist_data::cdf
```

cumulative density function array

6.5.2.2 erg

```
float dist_data::erg
```

energy point of this distribution

6.5.2.3 intt

```
unsigned dist_data::intt
```

interpolation flag of this distribution

6.5.2.4 law

```
unsigned dist_data::law
```

sampling law of this distribution

6.5.2.5 len

```
unsigned dist_data::len
```

length of the arrays in this distribution

6.5.2.6 pdf

```
float* dist_data::pdf
```

probability density function array

6.5.2.7 var

```
float* dist_data::var
```

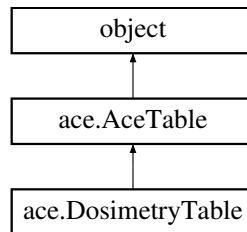
independent variable (mu, E, etc.)

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

- [datadef.h](#)

6.6 ace.DosimetryTable Class Reference

Inheritance diagram for ace.DosimetryTable:



Public Member Functions

- `def __init__(self, name, awr, temp)`
- `def __repr__(self)`

Additional Inherited Members

6.6.1 Constructor & Destructor Documentation

6.6.1.1 __init__()

```
def ace.DosimetryTable.__init__(  
    self,  
    name,  
    awr,  
    temp )
```

6.6.2 Member Function Documentation

6.6.2.1 __repr__()

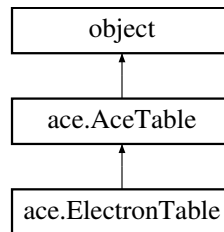
```
def ace.DosimetryTable.__repr__(  
    self )
```

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

- [ace.pyx](#)

6.7 ace.ElectronTable Class Reference

Inheritance diagram for ace.ElectronTable:



Public Member Functions

- `def __init__(self, name, awr, temp)`
- `def __repr__(self)`

Additional Inherited Members

6.7.1 Constructor & Destructor Documentation

6.7.1.1 __init__()

```
def ace.ElectronTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.7.2 Member Function Documentation

6.7.2.1 __repr__()

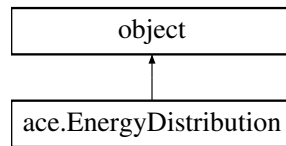
```
def ace.ElectronTable.__repr__ (
    self )
```

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

- [ace.pyx](#)

6.8 ace.EnergyDistribution Class Reference

Inheritance diagram for ace.EnergyDistribution:



Public Member Functions

- def [__init__](#) (self)

6.8.1 Constructor & Destructor Documentation

6.8.1.1 [__init__\(\)](#)

```
def ace.EnergyDistribution.__init__ (
    self )
```

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

- [ace.pyx](#)

6.9 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 [talnum](#)
- int [matnum](#)
- int [is_fissile](#)

6.9.1 Detailed Description

contains parameters of a wgeometry

extrema arrays, location array, cell and material numbers, tally number, fissile flag

6.9.2 Member Data Documentation

6.9.2.1 cellnum

```
int geom_data::cellnum
```

cell number

6.9.2.2 is_fissile

```
int geom_data::is_fissile
```

fissile flag

6.9.2.3 loc

```
float geom_data::loc[3]
```

array of coordinate (x,y,z) locations

6.9.2.4 matnum

```
int geom_data::matnum
```

material number

6.9.2.5 max

```
float geom_data::max[3]
```

array of coordinate (x,y,z) maxima

6.9.2.6 min

```
float geom_data::min[3]
```

array of coordinate (x,y,z) minima

6.9.2.7 talnum

```
int geom_data::talnum
```

tally index

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

- [datadef.h](#)

6.10 intersection_point Struct Reference

contains information pertinent to an intersection point, used in OptiX

```
#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](#)
- int [tally_index](#)

6.10.1 Detailed Description

contains information pertinent to an intersection point, used in OptiX

3D cartesian coordinates of intersection point, distance to nearest surface, first cell potentially hit, material of hit cell, fissile flag of hit cell, normal vector of hit surface, hit cell sense value, index of tally for hit cell

6.10.2 Member Data Documentation

6.10.2.1 cell

```
int intersection_point::cell
```

most recently hit cell number

6.10.2.2 fiss

```
int intersection_point::fiss
```

most recently hit fissile flag

6.10.2.3 mat

```
int intersection_point::mat
```

most recently hit material number

6.10.2.4 norm

```
float intersection_point::norm[3]
```

most recently hit normal

6.10.2.5 sense

```
int intersection_point::sense
```

most recently hit cell sense

6.10.2.6 surf_dist

```
float intersection_point::surf_dist
```

distance to nearest surface

6.10.2.7 tally_index

```
int intersection_point::tally_index
```

tally index of most recently hit cell

6.10.2.8 x

```
float intersection_point::x
```

x-coordinate

6.10.2.9 y

```
float intersection_point::y
```

y-coordinate

6.10.2.10 z

```
float intersection_point::z
```

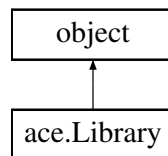
z-coordinate

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

- [datadef.h](#)

6.11 ace.Library Class Reference

Inheritance diagram for ace.Library:



Public Member Functions

- `def __init__ (self, filename)`
- `def read (self, table_names=None)`
- `def find_table (self, name)`
- `def __del__ (self)`

Public Attributes

- [f](#)
- [binary](#)
- [verbose](#)
- [tables](#)

Private Member Functions

- `def _read_binary (self, table_names, recl_length=4096, entries=512)`
- `def _read_ascii (self, table_names)`

6.11.1 Detailed Description

A Library objects represents an ACE-formatted file which may contain multiple tables with data.

Parameters

`filename` : str
Path of the ACE library file to load.

Attributes

`binary` : bool
Identifies Whether the library is in binary format or not

`tables` : dict
Dictionary whose keys are the names of the ACE tables and whose values are the instances of subclasses of AceTable (e.g. NeutronTable)

`verbose` : bool
Determines whether output is printed to the stdout when reading a Library

6.11.2 Constructor & Destructor Documentation

6.11.2.1 `__init__()`

```
def ace.Library.__init__ (
    self,
    filename )
```

6.11.2.2 `__del__()`

```
def ace.Library.__del__ (
    self )
```

6.11.3 Member Function Documentation

6.11.3.1 `_read_ascii()`

```
def ace.Library._read_ascii (
    self,
    table_names ) [private]
```

6.11.3.2 `_read_binary()`

```
def ace.Library._read_binary (
    self,
    table_names,
    recl_length = 4096,
    entries = 512 ) [private]
```

6.11.3.3 `find_table()`

```
def ace.Library.find_table (
    self,
    name )
```

`find_table(name)`

Returns a cross-section table with a given name.

Parameters

`name` : str
Name of the cross-section table, e.g. 92235.70c

6.11.3.4 read()

```
def ace.Library.read (
    self,
    table_names = None )

read(table_names=None)

Read through and parse the ACE-format library.

Parameters
-----
table_names : None, str, or iterable, optional
    Tables from the file to read in.  If None, reads in all of the
    tables.  If str, reads in only the single table of a matching name.
```

6.11.4 Member Data Documentation

6.11.4.1 binary

`ace.Library.binary`

6.11.4.2 f

`ace.Library.f`

6.11.4.3 tables

`ace.Library.tables`

6.11.4.4 verbose

`ace.Library.verbose`

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

- [ace.pyx](#)

6.12 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](#)

6.12.1 Detailed Description

contains information that defines a material

material index, label number, fissile flag, number of isotopes, density, isotope list, isotope atom fraction list

6.12.2 Member Data Documentation

6.12.2.1 density

```
float material_def::density
```

density [g/cc]

6.12.2.2 fractions

```
float* material_def::fractions
```

isotope atom fractions

6.12.2.3 id

```
unsigned material_def::id
```

material index

6.12.2.4 is_fissile

```
unsigned material_def::is_fissile
```

fissile flag

6.12.2.5 isotopes

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

isotope list

6.12.2.6 matnum

```
unsigned material_def::matnum
```

material label number

6.12.2.7 num_isotopes

```
unsigned material_def::num_isotopes
```

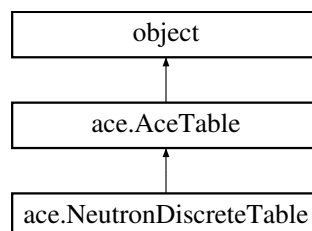
number of isotopes

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

- [datadef.h](#)

6.13 ace.NeutronDiscreteTable Class Reference

Inheritance diagram for ace.NeutronDiscreteTable:



Public Member Functions

- `def __init__(self, name, awr, temp)`
- `def __repr__(self)`

Additional Inherited Members

6.13.1 Constructor & Destructor Documentation

6.13.1.1 `__init__()`

```
def ace.NeutronDiscreteTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.13.2 Member Function Documentation

6.13.2.1 `__repr__()`

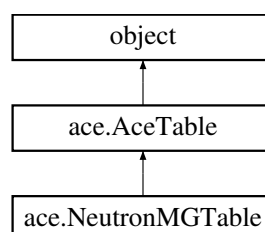
```
def ace.NeutronDiscreteTable.__repr__ (
    self )
```

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

- [ace.pyx](#)

6.14 ace.NeutronMGTable Class Reference

Inheritance diagram for ace.NeutronMGTable:



Public Member Functions

- `def __init__` (self, [name](#), [awr](#), [temp](#))
- `def __repr__` (self)

Additional Inherited Members

6.14.1 Constructor & Destructor Documentation

6.14.1.1 `__init__()`

```
def ace.NeutronMGTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.14.2 Member Function Documentation

6.14.2.1 `__repr__()`

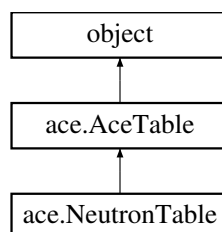
```
def ace.NeutronMGTable.__repr__ (
    self )
```

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

- [ace.pyx](#)

6.15 `ace.NeutronTable` Class Reference

Inheritance diagram for `ace.NeutronTable`:



Public Member Functions

- `def __init__ (self, name, awr, temp)`
- `def __repr__ (self)`
- `def find_reaction (self, mt)`
- `def __iter__ (self)`

Public Attributes

- [reactions](#)
- [photon_reactions](#)
- [heating](#)
- [nu_t_type](#)
- [nu_t_interp_NBT](#)
- [nu_t_interp_INT](#)
- [nu_t_energy](#)
- [nu_t_value](#)
- [nu_p_type](#)
- [nu_p_interp_NBT](#)
- [nu_p_interp_INT](#)
- [nu_p_energy](#)
- [nu_p_value](#)
- [nu_d_interp_NBT](#)
- [nu_d_interp_INT](#)
- [nu_d_energy](#)
- [nu_d_value](#)
- [nu_d_precursor_const](#)
- [nu_d_precursor_energy](#)
- [nu_d_precursor_prob](#)
- [nu_d_precursor_interp_NBT](#)
- [nu_d_precursor_interp_INT](#)
- [nu_d_energy_dist](#)
- [e_dist_energy_outNE](#)
- [e_dist_LP](#)
- [e_dist_EG](#)
- [sigma_photon](#)
- [sigma](#)
- [a_dist_energy_in](#)
- [a_dist_mu_out](#)
- [MT_for_photon_yield](#)
- [IE_fission](#)
- [sigma_f](#)
- [urr_energy](#)
- [urr_table](#)

Private Member Functions

- [def _read_all](#) (self)
- [def _read_cross_sections](#) (self)
- [def _read_nu](#) (self)
- [def _read_angular_distributions](#) (self)
- [def _read_energy_distributions](#) (self)
- [def _get_energy_distribution](#) (self, location_start, delayed_n=False)
- [def _read_gpd](#) (self)
- [def _read_mtrp](#) (self)
- [def _read_lsigg](#) (self)
- [def _read_sigp](#) (self)
- [def _read_landp](#) (self)
- [def _read_andp](#) (self)
- [def _read_yp](#) (self)
- [def _read_fis](#) (self)
- [def _read_unr](#) (self)

6.15.1 Detailed Description

A NeutronTable object contains continuous-energy neutron interaction data read from an ACE-formatted Type I table. These objects are not normally instantiated by the user but rather created when reading data using a Library object and stored within the ``tables`` attribute of a Library object.

Parameters

```
-----
name : str
    ZAID identifier of the table, e.g. '92235.70c'.
awr : float
    Atomic mass ratio of the target nuclide.
temp : float
    Temperature of the target nuclide in eV.
```

Attributes

```
-----
awr : float
    Atomic mass ratio of the target nuclide.

energy : list of floats
    The energy values (MeV) at which reaction cross-sections are tabulated.

name : str
    ZAID identifier of the table, e.g. 92235.70c.

nu_p_energy : list of floats
    Energies in MeV at which the number of prompt neutrons emitted per
    fission is tabulated.

nu_p_type : str
    Indicates how number of prompt neutrons emitted per fission is
    stored. Can be either "polynomial" or "tabular".

nu_p_value : list of floats
    The number of prompt neutrons emitted per fission, if data is stored in
    "tabular" form, or the polynomial coefficients for the "polynomial"
    form.

nu_t_energy : list of floats
    Energies in MeV at which the total number of neutrons emitted per
    fission is tabulated.

nu_t_type : str
    Indicates how total number of neutrons emitted per fission is
    stored. Can be either "polynomial" or "tabular".

nu_t_value : list of floats
    The total number of neutrons emitted per fission, if data is stored in
    "tabular" form, or the polynomial coefficients for the "polynomial"
    form.

reactions : list of Reactions
    A list of Reaction instances containing the cross sections, secondary
    angle and energy distributions, and other associated data for each
    reaction for this nuclide.

sigma_a : list of floats
    The microscopic absorption cross section for each value on the energy
    grid.

sigma_t : list of floats
    The microscopic total cross section for each value on the energy grid.

temp : float
    Temperature of the target nuclide in eV.
```

6.15.2 Constructor & Destructor Documentation

6.15.2.1 `__init__()`

```
def ace.NeutronTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.15.3 Member Function Documentation

6.15.3.1 `__iter__()`

```
def ace.NeutronTable.__iter__ (
    self )
```

6.15.3.2 `__repr__()`

```
def ace.NeutronTable.__repr__ (
    self )
```

6.15.3.3 `_get_energy_distribution()`

```
def ace.NeutronTable._get_energy_distribution (
    self,
    location_start,
    delayed_n = False ) [private]
```

Returns an EnergyDistribution object from data read in starting at location_start.

6.15.3.4 `_read_all()`

```
def ace.NeutronTable._read_all (
    self ) [private]
```

6.15.3.5 `_read_andp()`

```
def ace.NeutronTable._read_andp (
    self ) [private]
```

Find the angular distribution for each photon-producing reaction MT.

6.15.3.6 `_read_angular_distributions()`

```
def ace.NeutronTable._read_angular_distributions (
    self ) [private]
```

Find the angular distribution for each reaction MT

6.15.3.7 `_read_cross_sections()`

```
def ace.NeutronTable._read_cross_sections (
    self ) [private]
```

Reads and parses the ESZ, MTR, LQR, TRY, LSIG, and SIG blocks. These blocks contain the energy grid, all reaction cross sections, the total cross section, average heating numbers, and a list of reactions with their Q-values and multiplicities.

6.15.3.8 `_read_energy_distributions()`

```
def ace.NeutronTable._read_energy_distributions (
    self ) [private]
```

Determine the energy distribution for secondary neutrons for each reaction MT

6.15.3.9 `_read_fis()`

```
def ace.NeutronTable._read_fis (
    self ) [private]
```

Read total fission cross-section data if present. Generally, this table is not provided since it is redundant.

6.15.3.10 `_read_gpd()`

```
def ace.NeutronTable._read_gpd (
    self ) [private]
```

Read total photon production cross section.

6.15.3.11 `_read_landp()`

```
def ace.NeutronTable._read_landp (
    self ) [private]
```

Determine location of angular distribution for each photon-producing reaction MT.

6.15.3.12 `_read_lsigp()`

```
def ace.NeutronTable._read_lsigp (
    self ) [private]
```

Determine location of cross sections for each photon-producing reaction MT.

6.15.3.13 `_read_mtrp()`

```
def ace.NeutronTable._read_mtrp (
    self ) [private]
```

Get the list of reaction MTs for photon-producing reactions for this cross-section table. The MT values are somewhat arbitrary.

6.15.3.14 `_read_nu()`

```
def ace.NeutronTable._read_nu (
    self ) [private]
```

Read the NU block -- this contains information on the prompt and delayed neutron precursor yields, decay constants, etc

6.15.3.15 `_read_sigp()`

```
def ace.NeutronTable._read_sigp (
    self ) [private]
```

Read cross-sections for each photon-producing reaction MT.

6.15.3.16 `_read_unr()`

```
def ace.NeutronTable._read_unr (
    self ) [private]
```

Read the unresolved resonance range probability tables if present.

6.15.3.17 `_read_yp()`

```
def ace.NeutronTable._read_yp (
    self ) [private]
```

Read list of reactions required as photon production yield multipliers.

6.15.3.18 `find_reaction()`

```
def ace.NeutronTable.find_reaction (
    self,
    mt )
```

6.15.4 Member Data Documentation

6.15.4.1 `a_dist_energy_in`

```
ace.NeutronTable.a_dist_energy_in
```

6.15.4.2 a_dist_mu_out

`ace.NeutronTable.a_dist_mu_out`

6.15.4.3 e_dist_EG

`ace.NeutronTable.e_dist_EG`

6.15.4.4 e_dist_energy_outNE

`ace.NeutronTable.e_dist_energy_outNE`

6.15.4.5 e_dist_LP

`ace.NeutronTable.e_dist_LP`

6.15.4.6 heating

`ace.NeutronTable.heating`

6.15.4.7 IE_fission

`ace.NeutronTable.IE_fission`

6.15.4.8 MT_for_photon_yield

`ace.NeutronTable.MT_for_photon_yield`

6.15.4.9 nu_d_energy

`ace.NeutronTable.nu_d_energy`

6.15.4.10 nu_d_energy_dist

```
ace.NeutronTable.nu_d_energy_dist
```

6.15.4.11 nu_d_interp_INT

```
ace.NeutronTable.nu_d_interp_INT
```

6.15.4.12 nu_d_interp_NBT

```
ace.NeutronTable.nu_d_interp_NBT
```

6.15.4.13 nu_d_precursor_const

```
ace.NeutronTable.nu_d_precursor_const
```

6.15.4.14 nu_d_precursor_energy

```
ace.NeutronTable.nu_d_precursor_energy
```

6.15.4.15 nu_d_precursor_interp_INT

```
ace.NeutronTable.nu_d_precursor_interp_INT
```

6.15.4.16 nu_d_precursor_interp_NBT

```
ace.NeutronTable.nu_d_precursor_interp_NBT
```

6.15.4.17 nu_d_precursor_prob

```
ace.NeutronTable.nu_d_precursor_prob
```

6.15.4.18 nu_d_value

`ace.NeutronTable.nu_d_value`

6.15.4.19 nu_p_energy

`ace.NeutronTable.nu_p_energy`

6.15.4.20 nu_p_interp_INT

`ace.NeutronTable.nu_p_interp_INT`

6.15.4.21 nu_p_interp_NBT

`ace.NeutronTable.nu_p_interp_NBT`

6.15.4.22 nu_p_type

`ace.NeutronTable.nu_p_type`

6.15.4.23 nu_p_value

`ace.NeutronTable.nu_p_value`

6.15.4.24 nu_t_energy

`ace.NeutronTable.nu_t_energy`

6.15.4.25 nu_t_interp_INT

`ace.NeutronTable.nu_t_interp_INT`

6.15.4.26 nu_t_interp_NBT

`ace.NeutronTable.nu_t_interp_NBT`

6.15.4.27 nu_t_type

`ace.NeutronTable.nu_t_type`

6.15.4.28 nu_t_value

`ace.NeutronTable.nu_t_value`

6.15.4.29 photon_reactions

`ace.NeutronTable.photon_reactions`

6.15.4.30 reactions

`ace.NeutronTable.reactions`

6.15.4.31 sigma

`ace.NeutronTable.sigma`

6.15.4.32 sigma_f

`ace.NeutronTable.sigma_f`

6.15.4.33 sigma_photon

`ace.NeutronTable.sigma_photon`

6.15.4.34 `urr_energy`

```
ace.NeutronTable.urr_energy
```

6.15.4.35 `urr_table`

```
ace.NeutronTable.urr_table
```

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

- [ace.pyx](#)

6.16 `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 [talnum_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](#)

6.16.1 Detailed Description

OptiX stuff class.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 [optix_stuff\(\)](#) [1/2]

```
optix_stuff::optix_stuff (
    unsigned Nin,
    unsigned mult )
```

constructor

sets stack size multiplier and number of histories.

Parameters

in	<i>Nin</i>	- number of histories
in	<i>mult</i>	- stack size multiplier

6.16.2.2 optix_stuff() [2/2]

```
optix_stuff::optix_stuff ( )
```

"default" constructor

empty

6.16.2.3 ~optix_stuff()

```
optix_stuff::~~optix_stuff ( )
```

destructor

6.16.3 Member Function Documentation

6.16.3.1 get_outer_cell()

```
unsigned optix_stuff::get_outer_cell ( )
```

returns the outermost cell

Returns

outer_cell - number of outermost cell

6.16.3.2 get_outer_cell_type()

```
unsigned optix_stuff::get_outer_cell_type ( )
```

returns the outermost cell type

Returns

outer_cell_type - geometrical primitive type of the outermost cell

6.16.3.3 get_rand()

```
float optix_stuff::get_rand ( )
```

returns a random float

6.16.3.4 init()

```
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

in	<i>problem_geom</i>	- problem geometry
in	<i>compute_device_in</i>	- compute device to use (always 0)
in	<i>accel_type_in</i>	- acceleration type

6.16.3.5 init_internal()

```
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

in	<i>problem_geom</i>	- problem geometry
in	<i>compute_device_in</i>	- compute device to use (always zero)
in	<i>accel_type_in</i>	- acceleration type

6.16.3.6 make_color()

```
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

in	<i>color</i>	- color map
in	<i>x</i>	- used to check for a miss or normalize the color
in	<i>min,max</i>	- values used to normalize the color

6.16.3.7 make_geom_prim()

```
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 from the primitive, and puts the geometry instance into its group.

Parameters

in	<i>problem_geom</i>	- problem geometry
----	---------------------	--------------------

6.16.3.8 make_geom_xform()

```
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

in	<i>problem_geom</i>	- problem geometry
----	---------------------	--------------------

6.16.3.9 make_geom_xform_common()

```
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

in	<i>problem_geom</i>	- problem geometry
----	---------------------	--------------------

6.16.3.10 print()

```
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.

6.16.3.11 set_trace_type()

```
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)

6.16.3.12 trace() [1/3]

```
void optix_stuff::trace ( )
```

creates a trace

launches the trace on the compute device with N histories.

6.16.3.13 trace() [2/3]

```
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

in	<i>trace_type</i>	- trace type for OptiX context
----	-------------------	--------------------------------

6.16.3.14 trace() [3/3]

```
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

in	<i>trace_type</i>	- 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)
in	<i>n_active</i>	- number of active histories

6.16.4 Member Data Documentation**6.16.4.1 accel_type**

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

acceleration type

6.16.4.2 boundary_condition

```
unsigned optix_stuff::boundary_condition [private]
```

boundary condition of outermost cell

6.16.4.3 cellnum_ptr

```
CUdeviceptr optix_stuff::cellnum_ptr
```

CUDA cell numbers pointer

6.16.4.4 compute_device

```
unsigned optix_stuff::compute_device [private]
```

compute device number

6.16.4.5 context

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

OptiX context

6.16.4.6 done_ptr

```
CUdeviceptr optix_stuff::done_ptr
```

CUDA done flags pointer

6.16.4.7 GEOM_FLAG

```
unsigned optix_stuff::GEOM_FLAG [private]
```

geometry flag: 0 = primitive instancing, 1 = transform instancing, 2 = transform instancing with common primitives

6.16.4.8 matnum_ptr

```
CUdeviceptr optix_stuff::matnum_ptr
```

CUDA material numbers pointer

6.16.4.9 maxcell

```
unsigned optix_stuff::maxcell [private]
```

maximum (usually outermost) cell

6.16.4.10 mincell

```
unsigned optix_stuff::mincell [private]
```

minimum (usually innermost) cell

6.16.4.11 N

```
unsigned optix_stuff::N
```

number of histories

6.16.4.12 n_materials

```
unsigned optix_stuff::n_materials [private]
```

number of materials

6.16.4.13 optix_device

```
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

6.16.4.14 outer_cell

```
unsigned optix_stuff::outer_cell [private]
```

outermost cell

6.16.4.15 outer_cell_dims

```
float optix_stuff::outer_cell_dims[6]
```

outermost cell dimensions

6.16.4.16 outer_cell_type

```
unsigned optix_stuff::outer_cell_type [private]
```

outermost cell type

6.16.4.17 positions_ptr

```
CUdeviceptr optix_stuff::positions_ptr
```

CUDA positions pointer

6.16.4.18 remap_ptr

```
CUdeviceptr optix_stuff::remap_ptr
```

CUDA remaps pointer

6.16.4.19 rxn_ptr

```
CUdeviceptr optix_stuff::rxn_ptr
```

CUDA reactions pointer

6.16.4.20 stack_size_multiplier

```
unsigned optix_stuff::stack_size_multiplier
```

stack size multiplier

6.16.4.21 talnum_ptr

```
CUdeviceptr optix_stuff::talnum_ptr
```

CUDA tally numbers pointer

6.16.4.22 traverse_type

```
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](#)

6.17 particle_data Struct Reference

structure that holds all the arrays that define a particle's state

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


Public Attributes

- [spatial_data](#) * [space](#)
- unsigned * [rxn](#)
- float * [E](#)
- float * [Q](#)
- unsigned * [rn_bank](#)
- unsigned * [cellnum](#)
- unsigned * [matnum](#)
- unsigned * [isonum](#)
- int * [talnum](#)
- unsigned * [yield](#)
- float * [weight](#)
- unsigned * [index](#)

6.17.1 Detailed Description

structure that holds all the arrays that define a particle's state

Hold arrays that define a neutron's state and/or need to be passed between kernels. Data locality efficiency dictates that this must be a structure of arrays (SoA) and not be a structure built into arrays (array of structures - AoS). This container structure is passed to almost all kernels so they can access neutron state data.

6.17.2 Member Data Documentation

6.17.2.1 cellnum

```
unsigned* particle_data::cellnum
```

current cell number array

6.17.2.2 E

```
float* particle_data::E
```

energy array

6.17.2.3 index

```
unsigned* particle_data::index
```

current energy grid index array

6.17.2.4 isonum

`unsigned* particle_data::isonum`

current isotope number array

6.17.2.5 matnum

`unsigned* particle_data::matnum`

current material number array

6.17.2.6 Q

`float* particle_data::Q`

current reaction Q value array

6.17.2.7 rn_bank

`unsigned* particle_data::rn_bank`

random number seed array

6.17.2.8 rxn

`unsigned* particle_data::rxn`

current reaction array

6.17.2.9 space

`spatial_data* particle_data::space`

spatial data array

6.17.2.10 talnum

`int* particle_data::talnum`

current tally number array

6.17.2.11 weight

`float* particle_data::weight`

statistical weight array

6.17.2.12 yield

```
unsigned* particle_data::yield
```

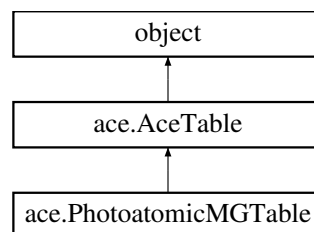
total yield of history array

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

- [datadef.h](#)

6.18 ace.PhotoatomicMGTable Class Reference

Inheritance diagram for ace.PhotoatomicMGTable:



Public Member Functions

- `def __init__(self, name, awr, temp)`
- `def __repr__(self)`

Additional Inherited Members

6.18.1 Constructor & Destructor Documentation

6.18.1.1 __init__()

```
def ace.PhotoatomicMGTable.__init__(
    self,
    name,
    awr,
    temp )
```

6.18.2 Member Function Documentation

6.18.2.1 `__repr__()`

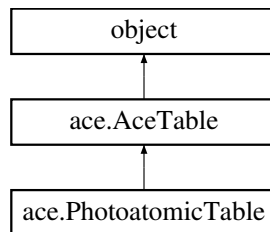
```
def ace.PhotoatomicMGTable.__repr__ (
    self )
```

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

- [ace.pyx](#)

6.19 `ace.PhotoatomicTable` Class Reference

Inheritance diagram for `ace.PhotoatomicTable`:



Public Member Functions

- `def __init__` (self, [name](#), [awr](#), [temp](#))
- `def __repr__` (self)

Additional Inherited Members

6.19.1 Constructor & Destructor Documentation

6.19.1.1 `__init__()`

```
def ace.PhotoatomicTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.19.2 Member Function Documentation

6.19.2.1 `__repr__()`

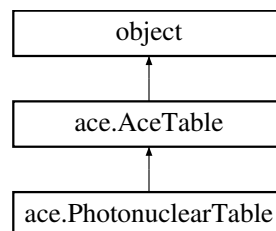
```
def ace.PhotoatomicTable.__repr__ (
    self )
```

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

- [ace.pyx](#)

6.20 ace.PhotonuclearTable Class Reference

Inheritance diagram for ace.PhotonuclearTable:



Public Member Functions

- `def __init__` (self, [name](#), [awr](#), [temp](#))
- `def __repr__` (self)

Additional Inherited Members

6.20.1 Constructor & Destructor Documentation

6.20.1.1 `__init__()`

```
def ace.PhotonuclearTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.20.2 Member Function Documentation

6.20.2.1 `__repr__()`

```
def ace.PhotonuclearTable.__repr__ (
    self )
```

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

- [ace.pyx](#)

6.21 primitive Class Reference

primitive class

```
#include <wprimitive.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)

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

6.21.1 Detailed Description

primitive class

6.21.2 Constructor & Destructor Documentation

6.21.2.1 primitive() [1/2]

```
primitive::primitive ( )
```

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

6.21.2.2 primitive() [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

in	<i>ptype</i>	- primitive type
in	<i>cellmat</i>	- cell material
in	<i>mins</i>	- object minima
in	<i>maxs</i>	- object maxima
in	<i>locs</i>	- object center points

6.21.2.3 ~primitive()

```
primitive::~~primitive ( )
```

primitive destructor

6.21.3 Member Function Documentation

6.21.3.1 add_transform() [1/3]

```
unsigned primitive::add_transform ( )
```

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

Returns

index of the added transform

6.21.3.2 add_transform() [2/3]

```
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

in	<i>cellnum</i>	- cell number
in	<i>dx,dy,dz</i>	- transform coordinates
in	<i>theta,phi</i>	- transform azimuthal and polar angles, respectively

Returns

index of the added transform

6.21.3.3 add_transform() [3/3]

```
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

in	<i>cellnum</i>	- cell number
in	<i>cellmat</i>	- cell material
in	<i>dx</i>	- displacement in x
in	<i>dy</i>	- displacement in y
in	<i>dz</i>	- displacement in z
in	<i>theta</i>	- transform azimuthal and polar angles
in	<i>phi</i>	- transform azimuthal and polar angles respectively

Returns

index of the added transform

6.21.3.4 make_hex_array()

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

creates a hexagonal array of elements.

Parameters

in	<i>n</i>	- edge length
in	<i>x</i>	- x displacement coordinates
in	<i>y</i>	- y displacement coordinates
in	<i>PD_ratio</i>	- pitch-to-diameter ratio
in	<i>starting_index</i>	- starting index

6.21.3.5 print_transform() [1/2]

```
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.

6.21.3.6 print_transform() [2/2]

```
void primitive::print_transform (
    int tnum )
```

prints out the properties of the input transform.

Parameters

in	<i>tnum</i>	- transform number
----	-------------	--------------------

6.21.4 Member Data Documentation**6.21.4.1 location**

```
float primitive::location[3]
```

coordinate location array

6.21.4.2 material

```
int primitive::material
```

material number

6.21.4.3 max

```
float primitive::max[3]
```

coordinate maxima array

6.21.4.4 min

```
float primitive::min[3]
```

coordinate minima array

6.21.4.5 n_transforms

```
int primitive::n_transforms
```

number of transforms

6.21.4.6 num_primitives

```
int primitive::num_primitives =0 [static]
```

number of primitives

6.21.4.7 primitive_id

```
int primitive::primitive_id
```

primitive ID number

6.21.4.8 transforms

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

transform vector

6.21.4.9 type

```
int primitive::type
```

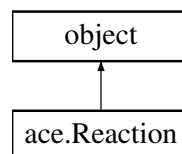
primitive type: 0 = box, 1 = cylinder, 2 = hexagon, 3 = sphere

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

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

6.22 ace.Reaction Class Reference

Inheritance diagram for ace.Reaction:



Public Member Functions

- `def __init__ (self, MT, table=None)`
- `def broaden (self, T_high)`
- `def threshold (self)`
- `def __repr__ (self)`

Public Attributes

- [table](#)
- [MT](#)
- [Q](#)
- [TY](#)
- [IE](#)
- [sigma](#)

6.22.1 Detailed Description

Reaction(MT, table=None)

A Reaction object represents a single reaction channel for a nuclide with an associated cross section and, if present, a secondary angle and energy distribution. These objects are stored within the ``reactions`` attribute on subclasses of AceTable, e.g. NeutronTable.

Parameters

MT : int
The ENDF MT number for this reaction. On occasion, MCNP uses MT numbers that don't correspond exactly to the ENDF specification.

table : AceTable
The ACE table which contains this reaction. This is useful if data on the parent nuclide is needed (for instance, the energy grid at which cross sections are tabulated)

Attributes

ang_energy_in : list of floats
Incoming energies in MeV at which angular distributions are tabulated.

ang_energy_cos : list of floats
Scattering cosines corresponding to each point of the angular distribution functions.

ang_energy_pdf : list of floats
Probability distribution function for angular distribution.

ang_energy_cdf : list of floats
Cumulative distribution function for angular distribution.

e_dist_energy : list of floats
Incoming energies in MeV at which energy distributions are tabulated.

e_dist_law : int
ACE law used for secondary energy distribution.

IE : int
The index on the energy grid corresponding to the threshold of this reaction.

MT : int
The ENDF MT number for this reaction. On occasion, MCNP uses MT numbers that don't correspond exactly to the ENDF specification.

Q : float
The Q-value of this reaction in MeV.

sigma : list of floats
Microscopic cross section for this reaction at each point on the energy grid above the threshold value.

TY : int
An integer whose absolute value is the number of neutrons emitted in this reaction. If negative, it indicates that scattering should be performed in the center-of-mass system. If positive, scattering should be preformed in the laboratory system.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 `__init__()`

```
def ace.Reaction.__init__ (
    self,
    MT,
    table = None )
```

6.22.3 Member Function Documentation

6.22.3.1 `__repr__()`

```
def ace.Reaction.__repr__ (
    self )
```

6.22.3.2 `broaden()`

```
def ace.Reaction.broaden (
    self,
    T_high )
```

6.22.3.3 `threshold()`

```
def ace.Reaction.threshold (
    self )
```

`threshold()`

Return energy threshold for this reaction.

6.22.4 Member Data Documentation

6.22.4.1 IE

`ace.Reaction.IE`

6.22.4.2 MT

`ace.Reaction.MT`

6.22.4.3 Q

`ace.Reaction.Q`

6.22.4.4 sigma

`ace.Reaction.sigma`

6.22.4.5 table

`ace.Reaction.table`

6.22.4.6 TY

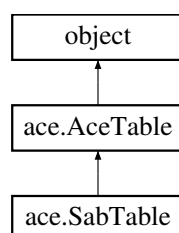
`ace.Reaction.TY`

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

- [ace.pyx](#)

6.23 ace.SabTable Class Reference

Inheritance diagram for `ace.SabTable`:



Public Member Functions

- `def __init__ (self, name, awr, temp)`
- `def __repr__ (self)`

Public Attributes

- `inelastic_e_in`
- `inelastic_sigma`
- `elastic_e_in`
- `elastic_P`
- `elastic_type`
- `inelastic_e_out`
- `inelastic_mu_out`
- `elastic_mu_out`

Private Member Functions

- `def _read_all (self)`
- `def _read_itie (self)`
- `def _read_itce (self)`
- `def _read_itxe (self)`
- `def _read_itca (self)`

6.23.1 Detailed Description

A SabTable object contains thermal scattering data as represented by an S(alpha, beta) table.

Parameters

```
-----
name : str
    ZAID identifier of the table, e.g. lwtr.10t.
awr : float
    Atomic mass ratio of the target nuclide.
temp : float
    Temperature of the target nuclide in eV.
```

Attributes

```
-----
awr : float
    Atomic mass ratio of the target nuclide.

elastic_e_in : list of floats
    Incoming energies in MeV for which the elastic cross section is
    tabulated.

elastic_P : list of floats
    Elastic scattering cross section for data derived in the incoherent
    approximation, or Bragg edge parameters for data derived in the coherent
    approximation.

elastic_type : str
    Describes the behavior of the elastic cross section, i.e. whether it was
    derived in the incoherent or coherent approximation.

inelastic_e_in : list of floats
    Incoming energies in MeV for which the inelastic cross section is
    tabulated.
```

```
inelastic_sigma : list of floats
    Inelastic scattering cross section in barns at each energy.

name : str
    ZAID identifier of the table, e.g. 92235.70c.

temp : float
    Temperature of the target nuclide in eV.
```

6.23.2 Constructor & Destructor Documentation

6.23.2.1 `__init__()`

```
def ace.SabTable.__init__ (
    self,
    name,
    awr,
    temp )
```

6.23.3 Member Function Documentation

6.23.3.1 `__repr__()`

```
def ace.SabTable.__repr__ (
    self )
```

6.23.3.2 `_read_all()`

```
def ace.SabTable._read_all (
    self ) [private]
```

6.23.3.3 `_read_itca()`

```
def ace.SabTable._read_itca (
    self ) [private]
```

Read angular distributions for elastic scattering.

6.23.3.4 `_read_itce()`

```
def ace.SabTable._read_itce (
    self ) [private]
```

Read energy-dependent elastic scattering cross sections.

6.23.3.5 `_read_itie()`

```
def ace.SabTable._read_itie (
    self ) [private]
```

Read energy-dependent inelastic scattering cross sections.

6.23.3.6 `_read_itxe()`

```
def ace.SabTable._read_itxe (
    self ) [private]
```

Read coupled energy/angle distributions for inelastic scattering.

6.23.4 Member Data Documentation

6.23.4.1 `elastic_e_in`

`ace.SabTable.elastic_e_in`

6.23.4.2 `elastic_mu_out`

`ace.SabTable.elastic_mu_out`

6.23.4.3 `elastic_P`

`ace.SabTable.elastic_P`

6.23.4.4 elastic_type

`ace.SabTable.elastic_type`

6.23.4.5 inelastic_e_in

`ace.SabTable.inelastic_e_in`

6.23.4.6 inelastic_e_out

`ace.SabTable.inelastic_e_out`

6.23.4.7 inelastic_mu_out

`ace.SabTable.inelastic_mu_out`

6.23.4.8 inelastic_sigma

`ace.SabTable.inelastic_sigma`

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

- [ace.pyx](#)

6.24 spatial_data Struct Reference

contains the spatial parameters of the neutron

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

Public Attributes

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

6.24.1 Detailed Description

contains the spatial parameters of the neutron

3D Cartesian coordinates, direction vector, distance to nearest surface, total macroscopic cross section, surface normal vector, boundary condition enforcement flag, neutron statistical weight

6.24.2 Member Data Documentation

6.24.2.1 enforce_BC

```
unsigned spatial_data::enforce_BC
```

boundary condition enforcement flag

6.24.2.2 norm

```
float spatial_data::norm[3]
```

normal vector of surface intersection

6.24.2.3 surf_dist

```
float spatial_data::surf_dist
```

distance to nearest surface

6.24.2.4 weight

```
unsigned spatial_data::weight
```

particle statistical weight

6.24.2.5 x

```
float spatial_data::x
```

x-coordinate

6.24.2.6 xhat

```
float spatial_data::xhat
```

direction vector x-component

6.24.2.7 y

```
float spatial_data::y
```

y-coordinate

6.24.2.8 yhat

```
float spatial_data::yhat
```

direction vector y-component

6.24.2.9 z

```
float spatial_data::z
```

z-coordinate

6.24.2.10 zhat

```
float spatial_data::zhat
```

direction vector z-component

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

- [datadef.h](#)

6.25 tally_data Struct Reference

Tally data that lives on the device side.

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

Public Attributes

- float * [score](#)
- float * [square](#)
- unsigned * [count](#)
- unsigned [cell](#)
- unsigned [length](#)
- float [E_min](#)
- float [E_max](#)

6.25.1 Detailed Description

Tally data that lives on the device side.

Tally data that lives on the device side. Everything needed to properly index a tally and the vectors to store the scores and keep track of statistics

6.25.2 Member Data Documentation

6.25.2.1 cell

```
unsigned tally_data::cell
```

tally cell (input)

6.25.2.2 count

```
unsigned* tally_data::count
```

tally count

6.25.2.3 E_max

```
float tally_data::E_max
```

maximum energy (input)

6.25.2.4 E_min

```
float tally_data::E_min
```

minimum energy (input)

6.25.2.5 length

```
unsigned tally_data::length
```

tally length, edges are equi-log (input)

6.25.2.6 score

```
float* tally_data::score
```

tally score

6.25.2.7 square

```
float* tally_data::square
```

tally square

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

- [datadef.h](#)

6.26 tally_data_host Struct Reference

Tally data that lives on the host side.

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

Public Attributes

- float * [score](#)
- float * [square](#)
- unsigned * [count](#)
- double * [score_total](#)
- double * [square_total](#)
- long unsigned * [count_total](#)
- unsigned [cell](#)
- unsigned [length](#)
- float [E_min](#)
- float [E_max](#)

6.26.1 Detailed Description

Tally data that lives on the host side.

Tally data that lives on the host side. Basically holds the same thing as the device array but also contains 64-bit arrays that the tallies are accumulated into to avoid too much roundoff error.

6.26.2 Member Data Documentation

6.26.2.1 cell

```
unsigned tally_data_host::cell
```

tally cell (input)

6.26.2.2 count

```
unsigned* tally_data_host::count
```

tally count

6.26.2.3 count_total

```
long unsigned* tally_data_host::count_total
```

tally count accumulated total

6.26.2.4 E_max

```
float tally_data_host::E_max
```

maximum energy (input)

6.26.2.5 E_min

```
float tally_data_host::E_min
```

minimum energy (input)

6.26.2.6 length

```
unsigned tally_data_host::length
```

tally length, edges are equi-log (input)

6.26.2.7 score

```
float* tally_data_host::score
```

tally score

6.26.2.8 score_total

```
double* tally_data_host::score_total
```

tally score accumulated total

6.26.2.9 square

```
float* tally_data_host::square
```

tally square

6.26.2.10 square_total

```
double* tally_data_host::square_total
```

tally square accumulaed total

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

- [datadef.h](#)

6.27 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 divison 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](#)

6.27.1 Detailed Description

class definitions for device vector operations

6.27.2 Constructor & Destructor Documentation

6.27.2.1 wfloat3() [1/3]

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

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

6.27.2.2 wfloat3() [2/3]

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

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

Parameters

in	<i>a</i>	- point to set
----	----------	----------------

6.27.2.3 wfloat3() [3/3]

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

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

Parameters

in	<i>a,b,c</i>	- points to set
----	--------------	-----------------

6.27.3 Member Function Documentation

6.27.3.1 cross()

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

cross product operator

returns the cross product of the vector and arg

Parameters

in	<i>arg</i>	- vector to cross
----	------------	-------------------

Returns

result - resultant `wfloat3`

6.27.3.2 dot()

```
__device__ float wfloat3::dot (
    wfloat3 arg ) [inline]
```

dot product operator

returns the dot product of the vector and arg

Parameters

in	<i>arg</i>	- vector to dot
----	------------	-----------------

6.27.3.3 norm2()

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

returns square root of sum of squares of coordinates

6.27.3.4 operator*() [1/2]

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

vector multiplication operator

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

Parameters

in	<i>arg</i>	- wfloat3 coordinates to be multiplied
----	------------	--

Returns

result - resultant [wfloat3](#)

6.27.3.5 operator*() [2/2]

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

scalar multiplication operator

multiplies x and arg, etc.

Parameters

in	<i>arg</i>	- number by which to multiply
----	------------	-------------------------------

Returns

result - resultant [wfloat3](#)

6.27.3.6 operator+() [1/2]

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

vector addition operator

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

Parameters

in	<i>arg</i>	- wfloat3 coordinates to be added
----	------------	---

Returns

result - resultant [wfloat3](#)

6.27.3.7 operator+() [2/2]

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

scalar addition operator

adds x and arg, etc.

Parameters

in	<i>arg</i>	- number to be added
----	------------	----------------------

Returns

result - resultant [wfloat3](#)

6.27.3.8 operator-() [1/2]

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

vector subtraction operator

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

Parameters

in	<i>arg</i>	- wfloat3 coordinates to be subtracted
----	------------	--

Returns

result - resultant [wfloat3](#)

6.27.3.9 operator-() [2/2]

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

scalar subtraction operator

subtracts arg from x, etc.

Parameters

in	<i>arg</i>	- number to be subtracted
----	------------	---------------------------

Returns

result - resultant [wfloat3](#)

6.27.3.10 operator/()

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

scalar division operator

divides x by arg, etc.

Parameters

in	<i>arg</i>	- number by which to divide
----	------------	-----------------------------

Returns

result - resultant [wfloat3](#)

6.27.3.11 rodrigues_rotation()

```
__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

in	<i>k</i>	- unit vector describing axis of rotation about which to rotate
in	<i>theta</i>	- angle by which to rotate

6.27.3.12 rotate()

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

rotation about random cosine

borrowed from OpenMC

Parameters

in	<i>mu</i>	- random cos(theta)
in	<i>rn</i>	- random number

6.27.4 Member Data Documentation**6.27.4.1 x**

```
float wfloat3::x
```

x-coordinate

6.27.4.2 y

```
float wfloat3::y
```

y-coordinate

6.27.4.3 z

```
float wfloat3::z
```

z-coordinate

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

- [wfloat3.h](#)

6.28 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 [add_tally](#) (unsigned)
- 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_transform](#) (unsigned, unsigned)
- void [delete_primitive](#) (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 [n_tallies](#)
- unsigned [fissile_flag](#)
- unsigned [boundary_condition](#)
- unsigned * [material_num_list](#)
- unsigned * [cell_num_list](#)
- std::string [datapath](#)
- std::vector< [primitive](#) > [primitives](#)
- std::vector< unsigned > [tally_cells](#)
- 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](#)

6.28.1 Detailed Description

wgeometry class

6.28.2 Constructor & Destructor Documentation

6.28.2.1 wgeometry()

```
wgeometry::wgeometry ( )
```

wgeometry constructor

6.28.2.2 ~wgeometry()

```
wgeometry::~~wgeometry ( )
```

wgeometry destructor

6.28.3 Member Function Documentation

6.28.3.1 add_material()

```
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

in	<i>matnum</i>	- material number
in	<i>is_fissile</i>	- fissile flag
in	<i>num_topes</i>	- number of isotopes in material
in	<i>density</i>	- density of material
in	<i>isotopes</i>	- list of isotopes
in	<i>fractions</i>	- fractions of the constituent isotopes

6.28.3.2 add_primitive() [1/2]

```
unsigned wgeometry::add_primitive ( )
```

adds a primitive to the geometry object

6.28.3.3 add_primitive() [2/2]

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

6.28.3.4 add_tally()

```
void wgeometry::add_tally (
    unsigned cellnum_in )
```

adds a tally for the specified cell to the geometry object

6.28.3.5 add_transform() [1/3]

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

adds a default transform to a primitive

Parameters

in	<i>index</i>	- the index of the primitive to add the transform to
----	--------------	--

Returns

the index of the transform for the specified primitive

6.28.3.6 add_transform() [2/3]

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

adds a transform to a primitive

Parameters

in	<i>index</i>	- the index of the primitive to add the transform to
in	<i>cellnum</i>	- cell number associated with the transform
in	<i>dx</i>	- displacement in x
in	<i>dy</i>	- displacement in y
in	<i>dz</i>	- displacement in z
in	<i>theta</i>	- polar rotation in the xz plane
in	<i>phi</i>	- azimuthal rotation in the xy plane

Returns

the index of the transform for the specified primitive

6.28.3.7 add_transform() [3/3]

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

adds a transform to a primitive

Parameters

in	<i>index</i>	- the index of the primitive to add the transform to
in	<i>cellnum</i>	- cell number associated with the transform
in	<i>cellmat</i>	- material number associated with the transform
in	<i>dx</i>	- displacement in x
in	<i>dy</i>	- displacement in y
in	<i>dz</i>	- displacement in z
in	<i>theta</i>	- polar rotation in the yz plane
in	<i>phi</i>	- azimuthal rotation in the xy plane

Returns

the index of the transform for the specified primitive

6.28.3.8 check()

```
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.

6.28.3.9 check_fissile()

```
unsigned wgeometry::check_fissile ( )
```

checks whether or not the geometry contains a fissile material.

Returns

fissile_flag

6.28.3.10 delete_primitive()

```
void wgeometry::delete_primitive (
    unsigned index )
```

delete a primitive

Parameters

in	<i>index</i>	- primitive index
----	--------------	-------------------

6.28.3.11 delete_transform()

```
void wgeometry::delete_transform (
    unsigned index,
    unsigned element )
```

delete a transform

Parameters

in	<i>index</i>	- primitive index
in	<i>element</i>	- transform index

6.28.3.12 get_boundary_condition()

```
unsigned wgeometry::get_boundary_condition ( )
```

returns the outermost cell type

Returns

outer_cell_type

6.28.3.13 get_material_count()

```
unsigned wgeometry::get_material_count ( )
```

returns the number of materials.

Returns

n_materials

6.28.3.14 get_material_table()

```
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

in	<i>n_mat_in</i>	- number of input materials
in	<i>n_tope_in</i>	- number of input isotopes
in	<i>conc_mat_in</i>	- pointer to pointer to set as location of the now-generated concentration matrix

6.28.3.15 `get_maximum_cell()`

```
unsigned wgeometry::get_maximum_cell ( )
```

returns the largest cell number, typically the outermost cell

6.28.3.16 `get_maximum_material()`

```
unsigned wgeometry::get_maximum_material ( )
```

returns the largest material number

6.28.3.17 `get_minimum_cell()`

```
unsigned wgeometry::get_minimum_cell ( )
```

returns the smallest cell number, typically the innermost cell

6.28.3.18 `get_minimum_material()`

```
unsigned wgeometry::get_minimum_material ( )
```

returns the smallest material number

6.28.3.19 `get_outer_cell()`

```
unsigned wgeometry::get_outer_cell ( )
```

returns the outermost cell and its boundary condition

Returns

`outer_cell`

6.28.3.20 `get_outer_cell_dims()`

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

returns the dimensions of the outermost cell.

Parameters

in	<i>input_array</i>	
----	--------------------	--

Returns

primitives[k].type

6.28.3.21 get_outer_cell_type()

```
unsigned wgeometry::get_outer_cell_type ( )
```

returns the boundary condition flag of outermost cell type

Returns

boundary_condition

6.28.3.22 get_primitive_count()

```
unsigned wgeometry::get_primitive_count ( )
```

returns the number of primitives in the geometry object

6.28.3.23 get_transform_count()

```
unsigned wgeometry::get_transform_count ( )
```

returns the number of transforms in the geometry object

6.28.3.24 make_hex_array()

```
void wgeometry::make_hex_array (
    unsigned index,
    int n,
    float x,
    float y,
    float phi,
    unsigned starting_index )
```

make a bunch of transforms in a hex arrangement

Parameters

in	<i>index</i>	- the index of the primitive to add the transform to
in	<i>n</i>	- the number of elements on an edge
in	<i>x</i>	- x position (not used)
in	<i>y</i>	- y position (not used)
in	<i>phi</i>	- azimuthal xy plane rotation position (not used)
in	<i>starting_index</i>	- the first index of the new transforms added, spans range starting_index to starting_index + n

6.28.3.25 make_material_table()

```
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));`

6.28.3.26 print_all()

```
void wgeometry::print_all ( )
```

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

6.28.3.27 print_materials_table()

```
void wgeometry::print_materials_table ( )
```

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

6.28.3.28 print_summary()

```
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.

6.28.3.29 set_datapath()

```
void wgeometry::set_datapath (
    std::string path_in )
```

sets the data path

Parameters

in	<i>path_in</i>	- input DATAPATH to the xs data
----	----------------	---------------------------------

6.28.3.30 set_outer_cell()

```
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

in	<i>ocell</i>	- the outermost cell
in	<i>BC</i>	- the boundary condition for the outermost cell (1=vacuum 2=mirror)

6.28.3.31 update()

```
void wgeometry::update ( )
```

updates the numbers of all shapes, compiles the list of all isotopes, creates an isotope table, sets tally indicies to cells.

6.28.4 Member Data Documentation**6.28.4.1 awr_list**

```
float* wgeometry::awr_list
```

atomic weight ratio (AWR) list

6.28.4.2 boundary_condition

```
unsigned wgeometry::boundary_condition
```

flag for the cell's boundary condition

6.28.4.3 cell_num_list

```
unsigned* wgeometry::cell_num_list
```

list of cell numbers

6.28.4.4 concentrations_matrix

```
float* wgeometry::concentrations_matrix
```

concentrations matrix

6.28.4.5 datapath

```
std::string wgeometry::datapath
```

path to xsdir and data

6.28.4.6 fissile_flag

```
unsigned wgeometry::fissile_flag
```

indicates whether or not a material is fissile

6.28.4.7 isotope_list

```
std::string wgeometry::isotope_list
```

isotope list

6.28.4.8 isotope_list_array

```
unsigned* wgeometry::isotope_list_array
```

isotope list array

6.28.4.9 isotopes

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

isotopes vector

6.28.4.10 material_list_array

```
unsigned* wgeometry::material_list_array
```

material list array

6.28.4.11 material_num_list

```
unsigned* wgeometry::material_num_list
```

list of material numbers

6.28.4.12 materials

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

materials vector

6.28.4.13 n_box

```
unsigned wgeometry::n_box
```

number of boxes

6.28.4.14 n_cyl

```
unsigned wgeometry::n_cyl
```

number of cylinders

6.28.4.15 n_hex

```
unsigned wgeometry::n_hex
```

number of hexagons

6.28.4.16 n_isotopes

```
unsigned wgeometry::n_isotopes
```

number of isotopes

6.28.4.17 n_materials

```
unsigned wgeometry::n_materials
```

number of materials

6.28.4.18 n_primitives

```
unsigned wgeometry::n_primitives
```

number of primitives

6.28.4.19 n_sph

```
unsigned wgeometry::n_sph
```

number of spheres

6.28.4.20 n_tallies

```
unsigned wgeometry::n_tallies
```

number of tallies

6.28.4.21 n_transforms

```
unsigned wgeometry::n_transforms
```

number of transforms

6.28.4.22 outer_cell

```
unsigned wgeometry::outer_cell
```

outermost cell (usually used for tallying)

6.28.4.23 primitives

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

primitives vector

6.28.4.24 tally_cells

```
std::vector<unsigned> wgeometry::tally_cells
```

primitives vector

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

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

6.29 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](#) ()
writes all tally values to file
- 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 [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](#) ([spatial_data](#) *, unsigned)
bins and accumulates fission points to grid
- void [write_fission_points](#) ()
writes binned fission point image to a .png
- void [print_banner](#) ()

- prints an amazing and beautiful WARP banner to stdout
- void [get_Py_buffer_dims](#) (unsigned *, unsigned *, unsigned *, Py_buffer *)
 does size logic on python buffer and writes values into pointers passed. Made since python interface routines don't set first/second shape value if the array size is 0/1

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
- void [init_device](#) ()
 initializes data on the device
- void [copy_data_to_device](#) ()
 copies data from the host device to the compute device
- void [init_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](#) ()
 accumulates the flux tally
- 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 *)
remaps active histories
- void [write_to_file](#) ([spatial_data](#) *, unsigned, std::string, std::string)
prints the locations of the source points to file
- void [write_to_file](#) ([spatial_data](#) *, 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
- void [copy_python_buffer](#) (float **, float **, std::string)
calls python function, copys returned buffer to C and CUDA pointers, float overload
- void [copy_python_buffer](#) (unsigned **, unsigned **, std::string)
calls python function, copys returned buffer to C and CUDA pointers, unsigned overload
- void [copy_python_buffer](#) (float **, std::string)
calls python function, copys returned buffer to C pointer (no cuda), float overload
- void [copy_python_buffer](#) (unsigned **, std::string)
calls python function, copys returned buffer to C pointer (no cuda), unsigned overload
- int [init_python](#) ()
initialize cross section data object in python
- void [copy_scatter_data](#) ()
- void [copy_energy_data](#) ()

Private Attributes

- [wgeometry](#) [problem_geom](#)
- std::string [accel_type](#)
- CUDPPHandle [theCudpp](#)
- CUDPPConfiguration [compact_config](#)
- CUDPPConfiguration [scan_int_config](#)
- CUDPPConfiguration [redu_int_config](#)
- CUDPPConfiguration [redu_float_config](#)
- CUDPPConfiguration [radix_config](#)
- CUDPPHandle [scanplan_int](#)
- CUDPPHandle [reduplan_int](#)
- CUDPPHandle [reduplan_float](#)
- CUDPPHandle [compactplan](#)
- CUDPPHandle [radixplan](#)
- CUDPPResult [res](#)
- curandGenerator_t [rand_gen](#)
- unsigned [N](#)
- unsigned [Ndataset](#)
- unsigned [NUM_THREADS](#)
- unsigned [blks](#)
- unsigned [compute_device](#)
- cudaStream_t [stream](#) [5]
- [cross_section_data](#) * [d_xsdata](#)
- [cross_section_data](#) [dh_xsdata](#)
- [cross_section_data](#) [h_xsdata](#)

- [particle_data](#) * [d_particles](#)
- [particle_data](#) [dh_particles](#)
- [particle_data](#) [h_particles](#)
- [tally_data](#) * [d_tally](#)
- [tally_data](#) * [dh_tally](#)
- [tally_data_host](#) * [h_tally](#)
- [dist_container](#) * [dh_dist_energy](#)
- [dist_container](#) * [dh_dist_scatter](#)
- unsigned [n_edges](#)
- unsigned * [edges](#)
- unsigned * [d_edges](#)
- unsigned * [reduced_yields](#)
- float * [reduced_weight](#)
- unsigned * [d_reduced_yields](#)
- float * [d_reduced_weight](#)
- long unsigned [reduced_yields_total](#)
- double [reduced_weight_total](#)
- double [initial_weight_total](#)
- unsigned * [material_list](#)
- unsigned * [isotope_list](#)
- float * [number_density_matrix](#)
- unsigned * [d_material_list](#)
- unsigned * [d_isotope_list](#)
- float * [d_number_density_matrix](#)
- unsigned * [remap](#)
- unsigned * [d_remap](#)
- unsigned * [zeros](#)
- unsigned * [d_zeros](#)
- unsigned * [ones](#)
- int * [mones](#)
- float * [fones](#)
- PyObject * [xsdat_instance](#)
- unsigned [RUN_FLAG](#)
- unsigned [outer_cell](#)
- unsigned [outer_cell_type](#)
- unsigned [n_materials](#)
- unsigned [n_isotopes](#)
- unsigned [n_tallies](#)
- unsigned [n_skip](#)
- unsigned [n_cycles](#)
- float [keff_sum](#)
- float [keff2_sum](#)
- float [keff_err](#)
- std::string [filename](#)
- unsigned [is_initialized](#)
- unsigned [print_flag](#)
- unsigned [dump_flag](#)
- size_t [mem_free](#)
- size_t [mem_total](#)
- unsigned * [d_valid_result](#)
- unsigned * [d_valid_N](#)
- float * [d_fissile_energy](#)
- [spatial_data](#) * [d_fissile_points](#)
- unsigned * [d_scanned](#)
- unsigned * [d_num_completed](#)

- unsigned * [d_num_active](#)
- [spatial_data](#) * [d_bank_space](#)
- float * [d_bank_E](#)
- 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](#)

6.29.1 Detailed Description

whistory class

6.29.2 Constructor & Destructor Documentation

6.29.2.1 whistory()

```
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.

6.29.2.2 ~whistory()

```
whistory::~whistory ( )
```

destructor

6.29.3 Member Function Documentation

6.29.3.1 accumulate_keff()

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

accumulates yields into host side values

Parameters

in	<i>converged</i>	- tally accumulation flag
in	<i>iteration</i>	- the active iteration number (starts at 0)
in	<i>keff</i>	- the running cumulative keff
in	<i>keff_cycle</i>	- the keff of the last cycle, used to renormalize the source vector

6.29.3.2 accumulate_tally()

```
void whistory::accumulate_tally ( ) [private]
```

accumulates the flux tally

6.29.3.3 bin_fission_points()

```
void whistory::bin_fission_points (
    spatial_data * d_space,
    unsigned N )
```

bins and accumulates fission points to grid

Parameters

in	<i>d_space</i>	- device space points
in	<i>N</i>	- dataset size

6.29.3.4 converge()

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

not called in [whistory.cpp](#)

6.29.3.5 copy_data_to_device()

```
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.

6.29.3.6 copy_energy_data()

```
void whistory::copy_energy_data ( ) [private]
```

high level function that does the energy distribution data copying

6.29.3.7 copy_python_buffer() [1/4]

```
void whistory::copy_python_buffer (
    float ** device_pointer,
    float ** host_pointer,
    std::string function_name ) [private]
```

calls python function, copys returned buffer to C and CUDA pointers, float overload

Parameters

in	<i>device_pointer</i>	- device pointer
in	<i>host_pointer</i>	- host pointer
in	<i>function_name</i>	- python function name

6.29.3.8 copy_python_buffer() [2/4]

```
void whistory::copy_python_buffer (
    unsigned ** device_pointer,
    unsigned ** host_pointer,
    std::string function_name ) [private]
```

calls python function, copys returned buffer to C and CUDA pointers, unsigned overload

Parameters

in	<i>device_pointer</i>	- device pointer
in	<i>host_pointer</i>	- host pointer
in	<i>function_name</i>	- python function name

6.29.3.9 copy_python_buffer() [3/4]

```
void whistory::copy_python_buffer (
    float ** host_pointer,
    std::string function_name ) [private]
```

calls python function, copys returned buffer to C pointer (no cuda), float overload

Parameters

in	<i>host_pointer</i>	- host pointer
in	<i>function_name</i>	- python function name

6.29.3.10 `copy_python_buffer()` [4/4]

```
void whistory::copy_python_buffer (
    unsigned ** host_pointer,
    std::string function_name ) [private]
```

calls python function, copys returned buffer to C pointer (no cuda), unsigned overload

Parameters

in	<i>host_pointer</i>	- host pointer
in	<i>function_name</i>	- python function name

6.29.3.11 `copy_scatter_data()`

```
void whistory::copy_scatter_data ( ) [private]
```

high level function that does the scatter distribution data copying

6.29.3.12 `device_report()`

```
void whistory::device_report ( )
```

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

6.29.3.13 `get_Py_buffer_dims()`

```
void whistory::get_Py_buffer_dims (
    unsigned * rows,
    unsigned * columns,
    unsigned * bytes,
    Py_buffer * pBuff )
```

does size logic on python buffer and writes values into pointers passed. Made since python interface routines don't set first/second shape value if the array size is 0/1

6.29.3.14 get_time()

```
float whistory::get_time ( ) [private]
```

returns how long it takes to do something

6.29.3.15 hot2()

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

creates a hot2 color map

Parameters

in	<i>color</i>	- rgb colors, float[3]
in	<i>val</i>	- value
in	<i>min,max</i>	- values used to normalize the color

6.29.3.16 init()

```
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.

6.29.3.17 init_cross_sections()

```
void whistory::init_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.

6.29.3.18 init_CUDPP()

```
void whistory::init_CUDPP ( ) [private]
```

initializes CUDPP

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

6.29.3.19 init_device()

```
void whistory::init_device ( ) [private]
```

initializes data on the device

prepares data arrays by filling them with zeros

6.29.3.20 init_host()

```
void whistory::init_host ( ) [private]
```

initializes data on the host

prepares data arrays by filling them with zeros

6.29.3.21 init_python()

```
int whistory::init_python ( ) [private]
```

initialize cross section data object in python

6.29.3.22 init_RNG()

```
void whistory::init_RNG ( ) [private]
```

initializes the random number generator

6.29.3.23 make_color()

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

creates a color map

Parameters

in	<i>color</i>	- rgb colors, float[3]
in	<i>x</i>	- value
in	<i>min,max</i>	- values used to normalize the color

6.29.3.24 map_active()

```
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

6.29.3.25 nonzero()

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

creates a binary colormap, black iff 0

Parameters

in	<i>color</i>	- rgb colors, float[3]
in	<i>val</i>	- value
in	<i>min,max</i>	- values used to normalize the color; unused, only present to keep arguments the same as other colormaps

6.29.3.26 plot_geom()

```
void whistory::plot_geom (
    std::string type )
```

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

Parameters

in	<i>type</i>	- type of plot, ie color is based 'cell' or 'material'
----	-------------	--

6.29.3.27 `prep_secondaries()`

```
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.

6.29.3.28 `print_banner()`

```
void whistory::print_banner ( )
```

prints an amazing and beautiful WARP banner to stdout

6.29.3.29 `print_materials_table()`

```
void whistory::print_materials_table ( )
```

prints table of properties of geometry materials

6.29.3.30 `print_pointers()`

```
void whistory::print_pointers ( )
```

prints pointer information

6.29.3.31 `print_xs_data()`

```
void whistory::print_xs_data ( )
```

prints cross section data information

6.29.3.32 `reduce_done()`

```
unsigned whistory::reduce_done ( ) [private]
```

reduces done values

Returns

reduced_done - number of done values

6.29.3.33 `reduce_weight()`

```
float whistory::reduce_weight ( ) [private]
```

reduces weight values

Returns

total

6.29.3.34 `reduce_yield()`

```
unsigned whistory::reduce_yield ( ) [private]
```

reduces yield values

Returns

total

6.29.3.35 `remap_active()`

```
void whistory::remap_active (
    unsigned * num_active,
    unsigned * lscatter_N,
    unsigned * lscatter_start,
    unsigned * mscatter_N,
    unsigned * mscatter_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

in	<i>num_active</i>	- number of active histories
in	<i>lscatter_N</i>	- number of elastic scatters
in	<i>lscatter_start</i>	- elastic scattering start
in	<i>mscatter_N</i>	- number of inelastic scatters
in	<i>mscatter_start</i>	- inelastic scattering start
in	<i>cscatter_N</i>	- number of compound scatters
in	<i>cscatter_start</i>	- compound scattering start
in	<i>fission_N</i>	- number of fissions
in	<i>fission_start</i>	- fission start

6.29.3.36 reset_cycle()

```
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

in	<i>keff_cycle</i>	- keff value of previous cycle
----	-------------------	--------------------------------

6.29.3.37 reset_fixed()

```
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.

6.29.3.38 run()

```
void whistory::run ( )
```

runs history

initializes run variables, clears fissile points file, opens run stats file, records stats. finds the material and nearest surface 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.

6.29.3.39 sample_fissile_points()

```
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.

6.29.3.40 set_acceration()

```
void whistory::set_acceration (
    std::string accel_in )
```

does nothing

Parameters

in	<i>accel↔ _in</i>	- acceleration type
----	-----------------------	---------------------

6.29.3.41 set_device()

```
void whistory::set_device (
    unsigned dev_in )
```

sets device number to input value

Parameters

in	<i>dev↔ _in</i>	- device number
----	---------------------	-----------------

6.29.3.42 set_dump_level()

```
void whistory::set_dump_level (
    unsigned level )
```

sets what types of information are dumped to files

Parameters

in	<i>/level</i>	- dump type flag
----	---------------	------------------

6.29.3.43 set_filename()

```
void whistory::set_filename (
    std::string filename_in )
```

sets filename to input string

Parameters

in	<i>filename</i> ↔ <i>_in</i>	- filename
----	---------------------------------	------------

6.29.3.44 set_print_level()

```
void whistory::set_print_level (
    unsigned level )
```

sets amount of information printed to stdout

Parameters

in	<i>level</i>	- level of verbosity
----	--------------	----------------------

6.29.3.45 set_run_param()

```
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

in	<i>n_cycles</i> ↔ <i>_in</i>	- number of active cycles
in	<i>n_skip_in</i>	- number of cycles to skip

6.29.3.46 set_run_type() [1/2]

```
void whistory::set_run_type (
```

```
unsigned type_in )
```

sets run type

Parameters

in	<i>type</i> ↔ <i>_in</i>	- run type
----	-----------------------------	------------

6.29.3.47 set_run_type() [2/2]

```
void whistory::set_run_type (
    std::string type_in )
```

sets run type

Parameters

in	<i>type</i> ↔ <i>_in</i>	- run type
----	-----------------------------	------------

6.29.3.48 trace() [1/2]

```
void whistory::trace (
    unsigned type ) [private]
```

does an OptiX trace

Parameters

in	<i>type</i>	- trace type
----	-------------	--------------

6.29.3.49 trace() [2/2]

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

does an OptiX trace for a given number of active histories

Parameters

in	<i>type</i>	- trace type
in	<i>n_active</i>	- number of active histories

6.29.3.50 update_RNG()

```
void whistory::update_RNG ( ) [private]
```

updates the random number

6.29.3.51 write_fission_points()

```
void whistory::write_fission_points ( )
```

writes binned fission point image to a .png

6.29.3.52 write_results()

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

writes results to file

Parameters

in	<i>runtime</i>	- runtime
in	<i>keff</i>	- keff
in	<i>opentype</i>	- file extension

6.29.3.53 write_tally()

```
void whistory::write_tally ( )
```

writes all tally values to file

6.29.3.54 write_to_file() [1/4]

```
void whistory::write_to_file (
    spatial_data * array_in,
```

```
    unsigned N,  
    std::string filename,  
    std::string opentype )    [private]
```

prints the locations of the source points to file

Parameters

in	<i>array_in</i>	- source point array
in	<i>N</i>	- number of histories
in	<i>filename</i>	- filename
in	<i>opentype</i>	- file extension

6.29.3.55 write_to_file() [2/4]

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

prints the locations of the source points to file

Parameters

in	<i>array_in</i>	- source point array
in	<i>array_in2</i>	- second array
in	<i>N</i>	- number of histories
in	<i>filename</i>	- filename
in	<i>opentype</i>	- file extension

6.29.3.56 write_to_file() [3/4]

```
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

in	<i>array_in</i>	- source point array
in	<i>N</i>	- number of histories
in	<i>filename</i>	- filename
in	<i>opentype</i>	- file extension

6.29.3.57 write_to_file() [4/4]

```
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

in	<i>array_in</i>	- source point array
in	<i>array_in2</i>	- second array
in	<i>N</i>	- number of histories
in	<i>filename</i>	- filename
in	<i>opentype</i>	- file extension

6.29.3.58 write_xs_data()

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

writes cross section data to file

Parameters

in	<i>filename</i>	- filename
----	-----------------	------------

6.29.4 Member Data Documentation**6.29.4.1 accel_type**

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

acceleration type

6.29.4.2 blks

```
unsigned whistory::blks [private]
```

number of blocks

6.29.4.3 compact_config

```
CUDPPConfiguration whistory::compact_config [private]
```

CUDPP compact configuration

6.29.4.4 compactplan

```
CUDPPHandle whistory::compactplan [private]
```

CUDPP compact plan handle

6.29.4.5 compute_device

```
unsigned whistory::compute_device [private]
```

compute device

6.29.4.6 d_bank_E

```
float* whistory::d_bank_E [private]
```

device bank energy

6.29.4.7 d_bank_space

```
spatial_data* whistory::d_bank_space [private]
```

device bank space

6.29.4.8 d_edges

```
unsigned* whistory::d_edges [private]
```

device mapped array of edges

6.29.4.9 d_fissile_energy

```
float* whistory::d_fissile_energy [private]
```

device fissile energy

6.29.4.10 d_fissile_points

```
spatial_data* whistory::d_fissile_points [private]
```

device fissile points

6.29.4.11 d_isotope_list

```
unsigned* whistory::d_isotope_list [private]
```

device isotope list

6.29.4.12 d_material_list

```
unsigned* whistory::d_material_list [private]
```

device material list

6.29.4.13 d_num_active

```
unsigned* whistory::d_num_active [private]
```

device number of active histories

6.29.4.14 d_num_completed

```
unsigned* whistory::d_num_completed [private]
```

device number of completed histories

6.29.4.15 d_number_density_matrix

```
float* whistory::d_number_density_matrix [private]
```

device isotope number density matrix

6.29.4.16 d_particles

```
particle_data* whistory::d_particles [private]
```

device particle data structure containing device pointers

6.29.4.17 d_reduced_weight

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

device reduced weight

6.29.4.18 d_reduced_yields

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

device reduced yields

6.29.4.19 d_remap

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

remap pointer

6.29.4.20 d_scanned

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

device scanned pointer

6.29.4.21 d_tally

```
tally_data* whistory::d_tally [private]
```

device tally data structure containing device pointers

6.29.4.22 d_valid_N

```
unsigned* whistory::d_valid_N [private]
```

valied number of histories pointer

6.29.4.23 d_valid_result

```
unsigned* whistory::d_valid_result [private]
```

valid result pointer

6.29.4.24 d_xsdata

```
cross_section_data* whistory::d_xsdata [private]
```

device cross section data structure containing device pointers

6.29.4.25 d_zeros

```
unsigned* whistory::d_zeros [private]
```

device zeros array

6.29.4.26 dh_dist_energy

```
dist_container* whistory::dh_dist_energy [private]
```

6.29.4.27 dh_dist_scatter

```
dist_container* whistory::dh_dist_scatter [private]
```

6.29.4.28 dh_particles

```
particle_data whistory::dh_particles [private]
```

host particle data structure containing device pointers

6.29.4.29 dh_tally

```
tally_data* whistory::dh_tally [private]
```

host tally data structure containing device pointers

6.29.4.30 dh_xsdata

```
cross_section_data whistory::dh_xsdata [private]
```

host cross section data structure containing device pointers

6.29.4.31 dump_flag

```
unsigned whistory::dump_flag [private]
```

dump level

6.29.4.32 edges

```
unsigned* whistory::edges [private]
```

mapped array of edges

6.29.4.33 filename

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

file name

6.29.4.34 fiss_img

```
long unsigned* whistory::fiss_img [private]
```

fissile image accumulation

6.29.4.35 fones

```
float* whistory::fones [private]
```

float ones array

6.29.4.36 h_particles

```
particle_data whistory::h_particles [private]
```

host particle data structure containing host pointers

6.29.4.37 h_tally

```
tally_data_host* whistory::h_tally [private]
```

host tally data structure containing host pointers (has long values for accumulation)

6.29.4.38 h_xsdata

```
cross_section_data whistory::h_xsdata [private]
```

host cross section data structure containing host pointers

6.29.4.39 initial_weight_total

```
double whistory::initial_weight_total [private]
```

double for accumulating the starting weight numbers accurately on the host

6.29.4.40 is_initialized

```
unsigned whistory::is_initialized [private]
```

init flag

6.29.4.41 isotope_list

```
unsigned* whistory::isotope_list [private]
```

isotope list

6.29.4.42 isotopes

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

cross section isotope string

6.29.4.43 keff2_sum

```
float whistory::keff2_sum [private]
```

keff squared sum

6.29.4.44 keff_err

```
float whistory::keff_err [private]
```

keff error

6.29.4.45 keff_sum

```
float whistory::keff_sum [private]
```

keff sum

6.29.4.46 material_list

```
unsigned* whistory::material_list [private]
```

material list

6.29.4.47 mem_free

```
size_t whistory::mem_free [private]
```

device memory free

6.29.4.48 mem_total

```
size_t whistory::mem_total [private]
```

device memory total

6.29.4.49 mones

```
int* whistory::mones [private]
```

int negative ones array

6.29.4.50 MT_columns

```
unsigned whistory::MT_columns [private]
```

MT number columns

6.29.4.51 MT_rows

```
unsigned whistory::MT_rows [private]
```

MT number rows

6.29.4.52 N

```
unsigned whistory::N [private]
```

number of histories

6.29.4.53 n_cycles

```
unsigned whistory::n_cycles [private]
```

number of active cycles

6.29.4.54 n_edges

```
unsigned whistory::n_edges [private]
```

mapped array of number of edges

6.29.4.55 n_isotopes

```
unsigned whistory::n_isotopes [private]
```

number of isotopes

6.29.4.56 n_materials

```
unsigned whistory::n_materials [private]
```

number of materials

6.29.4.57 n_skip

```
unsigned whistory::n_skip [private]
```

number of cycles to skip

6.29.4.58 n_tallies

```
unsigned whistory::n_tallies [private]
```

number of tallies

6.29.4.59 Ndataset

```
unsigned whistory::Ndataset [private]
```

dataset size for number of histories

6.29.4.60 NUM_THREADS

```
unsigned whistory::NUM_THREADS [private]
```

number of threads per block

6.29.4.61 number_density_matrix

```
float* whistory::number_density_matrix [private]
```

isotope number density matrix

6.29.4.62 ones

```
unsigned* whistory::ones [private]
```

int ones array

6.29.4.63 outer_cell

```
unsigned whistory::outer_cell [private]
```

outermost cell

6.29.4.64 outer_cell_dims

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

outer cell minima and maxima

6.29.4.65 outer_cell_type

```
unsigned whistory::outer_cell_type [private]
```

outermost cell type

6.29.4.66 print_flag

```
unsigned whistory::print_flag [private]
```

print verbosity level

6.29.4.67 problem_geom

```
wgeometry whistory::problem_geom [private]
```

problem geometry

6.29.4.68 radix_config

```
CUDPPConfiguration whistory::radix_config [private]
```

CUDPP radix configuration

6.29.4.69 radixplan

```
CUDPPHandle whistory::radixplan [private]
```

CUDPP radix plan handle

6.29.4.70 rand_gen

```
curandGenerator_t whistory::rand_gen [private]
```

random number generator handle

6.29.4.71 redu_float_config

```
CUDPPConfiguration whistory::redu_float_config [private]
```

CUDPP reduced float configuration

6.29.4.72 redu_int_config

```
CUDPPConfiguration whistory::redu_int_config [private]
```

CUDPP reduced int configuration

6.29.4.73 reduced_weight

```
float* whistory::reduced_weight [private]
```

reduced weight

6.29.4.74 reduced_weight_total

```
double whistory::reduced_weight_total [private]
```

double for accumulating weight numbers accurately on the host

6.29.4.75 reduced_yields

```
unsigned* whistory::reduced_yields [private]
```

reduced yields

6.29.4.76 reduced_yields_total

```
long unsigned whistory::reduced_yields_total [private]
```

long unsigned for accumulating yield numbers accurately on the host

6.29.4.77 reduplan_float

```
CUDPPHandle whistory::reduplan_float [private]
```

CUDPP reduce plan float handle

6.29.4.78 reduplan_int

```
CUDPPHandle whistory::reduplan_int [private]
```

CUDPP reduce plan int handle

6.29.4.79 remap

```
unsigned* whistory::remap [private]
```

remap

6.29.4.80 res

```
CUDPPResult whistory::res [private]
```

CUDPP result

6.29.4.81 RUN_FLAG

```
unsigned whistory::RUN_FLAG [private]
```

run flag

6.29.4.82 scan_int_config

```
CUDPPConfiguration whistory::scan_int_config [private]
```

CUDPP scan int configuration

6.29.4.83 scanplan_int

```
CUDPPHandle whistory::scanplan_int [private]
```

CUDPP scan plan int handle

6.29.4.84 stream

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

CUDA streams cor concurrent kernels

6.29.4.85 theCudpp

```
CUDPPHandle whistory::theCudpp [private]
```

CUDPP handle

6.29.4.86 total_bytes_energy

```
unsigned whistory::total_bytes_energy [private]
```

total size of energy data

6.29.4.87 total_bytes_scatter

```
unsigned whistory::total_bytes_scatter [private]
```

total size of scattering data

6.29.4.88 xs_isotope_ints

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

cross section isotope numbers

6.29.4.89 xs_num_rxns

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

cross section number of reactions

6.29.4.90 xsdat_instance

```
PyObject* whistory::xsdat_instance [private]
```

Python object that loads and manipulates the cross section data

6.29.4.91 zeros

```
unsigned* whistory::zeros [private]
```

zeros array

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

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

6.30 wtransform Struct Reference

contains parameters of a wtransform, a transform used to create a new instance of a wgemetry object

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

Public Attributes

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

6.30.1 Detailed Description

contains parameters of a wtransform, a transform used to create a new instance of a wgemetry object

cell number and material, transform coordinates and angles

6.30.2 Member Data Documentation

6.30.2.1 cellmat

```
unsigned wtransform::cellmat
```

cell material

6.30.2.2 cellnum

```
unsigned wtransform::cellnum
```

cell number

6.30.2.3 dx

```
float wtransform::dx
```

displacement in x

6.30.2.4 dy

```
float wtransform::dy
```

displacement in y

6.30.2.5 dz

```
float wtransform::dz
```

displacement in z

6.30.2.6 phi

```
float wtransform::phi
```

rotation in azimuthal angle

6.30.2.7 tally_index

```
int wtransform::tally_index
```

index of tally associated with this instance

6.30.2.8 theta

```
float wtransform::theta
```

rotation in polar angle

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

- [datadef.h](#)

Chapter 7

File Documentation

7.1 ace.pyx File Reference

Classes

- class [ace.Library](#)
- class [ace.AceTable](#)
- class [ace.NeutronTable](#)
- class [ace.EnergyDistribution](#)
- class [ace.SabTable](#)
- class [ace.Reaction](#)
- class [ace.DosimetryTable](#)
- class [ace.NeutronDiscreteTable](#)
- class [ace.NeutronMGTable](#)
- class [ace.PhotoatomicTable](#)
- class [ace.PhotoatomicMGTable](#)
- class [ace.ElectronTable](#)
- class [ace.PhotonuclearTable](#)

Namespaces

- [ace](#)

Functions

- def [ace.fromstring_split](#) (s, sep=None, dtype=float)
- def [ace.fromstring_token](#) (s, sep=" ", bint, inplace=False, int, maxsize=-1)
- def [ace.ascii_to_binary](#) (ascii_file, binary_file)

Variables

- [ace.NP_LE_V15](#)
- dictionary [ace.table_types](#)

7.2 benchmarks.cpp File Reference

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

Functions

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

7.2.1 Function Documentation

7.2.1.1 main()

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

7.3 box_mesh.cu File Reference

```
#include <optix.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <optixu/optixu_aabb_namespace.h>
#include "datadef.h"
```

Functions

- [rtDeclareVariable](#) (optix::Ray, ray, rtCurrentRay,)
- [rtDeclareVariable](#) (unsigned, cellnum, attribute cell_num,)
- [rtDeclareVariable](#) (int, celltal, attribute cell_tal,)
- [rtDeclareVariable](#) (unsigned, cellmat, attribute cell_mat,)
- [rtDeclareVariable](#) (unsigned, cellfissile, attribute cell_fis,)
- [rtDeclareVariable](#) (unsigned, sense, attribute cell_sense,)
- [rtDeclareVariable](#) (float3, normal, attribute normal,)
- static __device__ float3 [boxnormal](#) (float t, float3 t0, float3 t1)
- RT_PROGRAM void [intersect](#) (int object_dex)
- RT_PROGRAM void [bounds](#) (int object_dex, float result[6])

Variables

- rtBuffer< [geom_data](#), 1 > [dims](#)

7.3.1 Function Documentation

7.3.1.1 bounds()

```
RT_PROGRAM void bounds (
    int  object_dex,
    float result[6] )
```

7.3.1.2 boxnormal()

```
static __device__ float3 boxnormal (
    float t,
    float3 t0,
    float3 t1 ) [static]
```

7.3.1.3 intersect()

```
RT_PROGRAM void intersect (
    int  object_dex )
```

7.3.1.4 rtDeclareVariable() [1/7]

```
rtDeclareVariable (
    optix::Ray ,
    ray ,
    rtCurrentRay )
```

7.3.1.5 rtDeclareVariable() [2/7]

```
rtDeclareVariable (
    unsigned ,
    cellnum ,
    attribute cell_num )
```

7.3.1.6 rtDeclareVariable() [3/7]

```
rtDeclareVariable (
    int ,
    celltal ,
    attribute cell_tal )
```

7.3.1.7 rtDeclareVariable() [4/7]

```
rtDeclareVariable (
    unsigned ,
    cellmat ,
    attribute cell_mat )
```

7.3.1.8 rtDeclareVariable() [5/7]

```
rtDeclareVariable (
    unsigned ,
    cellfissile ,
    attribute cell_fis )
```

7.3.1.9 rtDeclareVariable() [6/7]

```
rtDeclareVariable (
    unsigned ,
    sense ,
    attribute cell_sense )
```

7.3.1.10 rtDeclareVariable() [7/7]

```
rtDeclareVariable (
    float3 ,
    normal ,
    attribute normal )
```

7.3.2 Variable Documentation

7.3.2.1 dims

```
rtBuffer<geom_data,1> dims
```

7.4 camera.cu File Reference

```
#include "optix.h"
#include <optix_world.h>
#include "datadef.h"
```

Functions

- [rtDeclareVariable](#) (rtObject, top_object,,)
- [rtDeclareVariable](#) (uint, launch_index_in, rtLaunchIndex,)
- [rtDeclareVariable](#) (uint, launch_dim, rtLaunchDim,)
- [rtDeclareVariable](#) (unsigned, outer_cell,,)
- [rtDeclareVariable](#) (unsigned, trace_type,,)
- [rtDeclareVariable](#) (unsigned, boundary_condition,,)
- RT_PROGRAM void [camera](#) ()
- RT_PROGRAM void [exception](#) ()

Variables

- rtBuffer< [spatial_data](#), 1 > [positions_buffer](#)
- rtBuffer< unsigned, 1 > [rxn_buffer](#)
- rtBuffer< unsigned, 1 > [remap_buffer](#)
- rtBuffer< unsigned, 1 > [cellnum_buffer](#)
- rtBuffer< unsigned, 1 > [matnum_buffer](#)
- rtBuffer< unsigned, 1 > [talnum_buffer](#)

7.4.1 Function Documentation

7.4.1.1 camera()

```
RT_PROGRAM void camera ( )
```

7.4.1.2 exception()

```
RT_PROGRAM void exception ( )
```

7.4.1.3 rtDeclareVariable() [1/6]

```
rtDeclareVariable (
    rtObject ,
    top_object )
```

7.4.1.4 rtDeclareVariable() [2/6]

```
rtDeclareVariable (
    uint ,
    launch_index_in ,
    rtLaunchIndex )
```

7.4.1.5 rtDeclareVariable() [3/6]

```
rtDeclareVariable (
    uint ,
    launch_dim ,
    rtLaunchDim )
```

7.4.1.6 rtDeclareVariable() [4/6]

```
rtDeclareVariable (
    unsigned ,
    outer_cell )
```

7.4.1.7 rtDeclareVariable() [5/6]

```
rtDeclareVariable (
    unsigned ,
    trace_type )
```

7.4.1.8 rtDeclareVariable() [6/6]

```
rtDeclareVariable (
    unsigned ,
    boundary_condition )
```

7.4.2 Variable Documentation

7.4.2.1 cellnum_buffer

`rtBuffer<unsigned,1> cellnum_buffer`

7.4.2.2 matnum_buffer

`rtBuffer<unsigned,1> matnum_buffer`

7.4.2.3 positions_buffer

`rtBuffer<spatial_data,1> positions_buffer`

7.4.2.4 remap_buffer

`rtBuffer<unsigned,1> remap_buffer`

7.4.2.5 rxn_buffer

`rtBuffer<unsigned,1> rxn_buffer`

7.4.2.6 talnum_buffer

`rtBuffer<unsigned,1> talnum_buffer`

7.5 check_cuda.h File Reference

Macros

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

Functions

- `__host__ void check_cuda (cudaError_t code, const char *file, int line, bool abort=true)`
CUDA error check wrapper, host only.

7.5.1 Macro Definition Documentation

7.5.1.1 `check_cuda`

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

7.5.2 Function Documentation

7.5.2.1 `check_cuda()`

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

CUDA error check wrapper, host only.

this inline function prints detailed information about the return code of host-side CUDA functions and where they occur in the code

Parameters

in	<i>code</i>	- CUDA error code output from some host-side CUDA function call
in	<i>file</i>	- file where erroring fuction is, written inline by preprocessor
in	<i>line</i>	- line of file where erroring fuction is, written inline by preprocessor
in	<i>abort</i>	- flag to exit on error, default is true

7.6 `check_pointers.cu` File Reference

```
#include <cuda.h>  
#include <stdio.h>  
#include "datadef.h"  
#include "warp_device.cuh"  
#include "check_cuda.h"
```

```
#include "wfloat3.h"
```

Functions

- `__global__ void check_pointers_kernel` (unsigned `N`, unsigned `dex0`, unsigned `dex1`, `cross_section_data` *`d_xsdata`)
 - `void check_pointers` (unsigned `NUM_THREADS`, unsigned `dex0`, unsigned `dex1`, `cross_section_data` *`d_xsdata`)
- a

7.6.1 Function Documentation

7.6.1.1 check_pointers()

```
void check_pointers (
    unsigned NUM_THREADS,
    unsigned dex0,
    unsigned dex1,
    cross_section_data * d_xsdata )
```

a

b

Parameters

in	<code>NUM_THREADS</code>	- the number of threads to run per thread block
in	<code>dex0</code>	- starting index
in	<code>dex1</code>	- ending index
in	<code>d_xsdata</code>	- device pointer to cross section data pointer array

7.6.1.2 check_pointers_kernel()

```
__global__ void check_pointers_kernel (
    unsigned N,
    unsigned dex0,
    unsigned dex1,
    cross_section_data * d_xsdata )
```

7.7 copy_points.cu File Reference

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

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

Functions

- `__global__ void copy_points_kernel` (unsigned Nout, unsigned *Nvalid, unsigned current_index, unsigned *to_valid, [spatial_data](#) *positions_out, [spatial_data](#) *positions_in, float *E_out, float *E_in)
- void `copy_points` (unsigned NUM_THREADS, unsigned Nout, unsigned *Nvalid, unsigned current_index, unsigned *to_valid, [spatial_data](#) *positions_out, [spatial_data](#) *positions_in, float *E_out, float *E_in)
copy points between two sets of space and energy data buffers, redirected with a mapping array

7.7.1 Function Documentation

7.7.1.1 `copy_points()`

```
void copy_points (
    unsigned NUM_THREADS,
    unsigned Nout,
    unsigned * Nvalid,
    unsigned current_index,
    unsigned * to_valid,
    spatial\_data * positions_out,
    spatial\_data * positions_in,
    float * E_out,
    float * E_in )
```

copy points between two sets of space and energy data buffers, redirected with a mapping array

copy points between two sets of space and energy data buffers, redirected with a mapping array

Parameters

in	<code>NUM_THREADS</code>	- the number of threads to run per thread block
in	<code>Nout</code>	- the total number of threads to launch on the grid
in	<code>Nvalid</code>	- the total number of device elements to copy from
in	<code>current_index</code>	- starting index
in	<code>to_valid</code>	- device pointer to data remapping vector
in	<code>positions_out</code>	- device pointer to spatial data array destination
in	<code>positions_in</code>	- device pointer to spatial data array source
in	<code>E_out</code>	- device pointer to energy data array destination
in	<code>E_in</code>	- device pointer to energy data array source

7.7.1.2 `copy_points_kernel()`

```
__global__ void copy_points_kernel (
```



```

    unsigned Nout,
    unsigned * Nvalid,
    unsigned current_index,
    unsigned * to_valid,
    spatial_data * positions_out,
    spatial_data * positions_in,
    float * E_out,
    float * E_in )

```

7.8 cylinder_mesh.cu File Reference

```

#include <optix.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <optixu/optixu_aabb_namespace.h>
#include "datadef.h"

```

Functions

- [rtDeclareVariable](#) (optix::Ray, ray, rtCurrentRay,)
- [rtDeclareVariable](#) (unsigned, cellnum, attribute cell_num,)
- [rtDeclareVariable](#) (int, celltal, attribute cell_tal,)
- [rtDeclareVariable](#) (unsigned, cellmat, attribute cell_mat,)
- [rtDeclareVariable](#) (unsigned, cellfissile, attribute cell_fis,)
- [rtDeclareVariable](#) (unsigned, sense, attribute cell_sense,)
- [rtDeclareVariable](#) (float3, normal, attribute normal,)
- RT_PROGRAM void [intersect](#) (int object_dex)
- RT_PROGRAM void [bounds](#) (int object_dex, float result[6])

Variables

- rtBuffer< [geom_data](#), 1 > [dims](#)

7.8.1 Function Documentation

7.8.1.1 bounds()

```

RT_PROGRAM void bounds (
    int object_dex,
    float result[6] )

```

7.8.1.2 intersect()

```
RT_PROGRAM void intersect (
    int object_dex )
```

7.8.1.3 rtDeclareVariable() [1/7]

```
rtDeclareVariable (
    optix::Ray ,
    ray ,
    rtCurrentRay )
```

7.8.1.4 rtDeclareVariable() [2/7]

```
rtDeclareVariable (
    unsigned ,
    cellnum ,
    attribute cell_num )
```

7.8.1.5 rtDeclareVariable() [3/7]

```
rtDeclareVariable (
    int ,
    celltal ,
    attribute cell_tal )
```

7.8.1.6 rtDeclareVariable() [4/7]

```
rtDeclareVariable (
    unsigned ,
    cellmat ,
    attribute cell_mat )
```

7.8.1.7 rtDeclareVariable() [5/7]

```
rtDeclareVariable (
    unsigned ,
    cellfissile ,
    attribute cell_fis )
```

7.8.1.8 `rtDeclareVariable()` [6/7]

```
rtDeclareVariable (
    unsigned ,
    sense ,
    attribute cell_sense )
```

7.8.1.9 `rtDeclareVariable()` [7/7]

```
rtDeclareVariable (
    float3 ,
    normal ,
    attribute normal )
```

7.8.2 Variable Documentation

7.8.2.1 `dims`

```
rtBuffer<geom_data,1> dims
```

7.9 `datadef.h` File Reference

```
#include <vector>
#include <string>
```

Classes

- struct [wtransform](#)
contains parameters of a wtransform, a transform used to create a new instance of a wgeometry object
- struct [geom_data](#)
contains parameters of a wgeometry
- struct [spatial_data](#)
contains the spatial parameters of the neutron
- struct [intersection_point](#)
contains information pertinent to an intersection point, used in OptiX
- struct [material_def](#)
contains information that defines a material
- struct [dist_data](#)
contains information that defines an ENDF cross section distribution
- struct [dist_container](#)
container for pointers that map the nearest distributions to the energy grid point where it resides
- struct [cross_section_data](#)
structure that holds the topmost level of cross section data
- struct [particle_data](#)
structure that holds all the arrays that define a particle's state
- struct [tally_data_host](#)
Tally data that lives on the host side.
- struct [tally_data](#)
Tally data that lives on the device side.

7.10 device_copies.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
```

Functions

- void [copy_to_device](#) (void *dest, void *source, unsigned bytes)
function to do a host-to-device copy
- void [copy_from_device](#) (void *dest, void *source, unsigned bytes)
function to do a device-to-host copy
- void [allocate_on_device](#) (void **dest, unsigned bytes)
function to do a device memory allocation
- void [deallocate_on_device](#) (void *dest)
function to do a device memory allocation

7.10.1 Function Documentation

7.10.1.1 [allocate_on_device\(\)](#)

```
void allocate_on_device (
    void ** dest,
    unsigned bytes )
```

function to do a device memory allocation

Parameters

in	<i>dest</i>	- pointer on host
in	<i>bytes</i>	- number of bytes to copy

7.10.1.2 [copy_from_device\(\)](#)

```
void copy_from_device (
    void * dest,
    void * source,
    unsigned bytes )
```

function to do a device-to-host copy

Parameters

in	<i>dest</i>	- pointer on host
in	<i>source</i>	- cuda pointer on device
in	<i>bytes</i>	- number of bytes to copy

7.10.1.3 copy_to_device()

```
void copy_to_device (
    void * dest,
    void * source,
    unsigned bytes )
```

function to do a host-to-device copy

something

Parameters

in	<i>dest</i>	- cuda pointer on device
in	<i>source</i>	- pointer on host
in	<i>bytes</i>	- number of bytes to copy

7.10.1.4 deallocate_on_device()

```
void deallocate_on_device (
    void * dest )
```

function to do a device memory allocation

Parameters

in	<i>dest</i>	- pointer on device to deallocate
----	-------------	-----------------------------------

7.11 device_copies.h File Reference

Functions

- void [copy_to_device](#) (void *, void *, unsigned)
function to do a host-to-device copy
- void [copy_from_device](#) (void *, void *, unsigned)
function to do a device-to-host copy

- void `allocate_on_device` (void **, unsigned)
function to do a device memory allocation
- void `deallocate_on_device` (void *)
function to do a device memory allocation

7.11.1 Function Documentation

7.11.1.1 `allocate_on_device()`

```
void allocate_on_device (
    void ** dest,
    unsigned bytes )
```

function to do a device memory allocation

Parameters

in	<i>dest</i>	- pointer on host
in	<i>bytes</i>	- number of bytes to copy

7.11.1.2 `copy_from_device()`

```
void copy_from_device (
    void * dest,
    void * source,
    unsigned bytes )
```

function to do a device-to-host copy

Parameters

in	<i>dest</i>	- pointer on host
in	<i>source</i>	- cuda pointer on device
in	<i>bytes</i>	- number of bytes to copy

7.11.1.3 `copy_to_device()`

```
void copy_to_device (
    void * dest,
    void * source,
    unsigned bytes )
```

function to do a host-to-device copy

something

Parameters

in	<i>dest</i>	- cuda pointer on device
in	<i>source</i>	- pointer on host
in	<i>bytes</i>	- number of bytes to copy

7.11.1.4 deallocate_on_device()

```
void deallocate_on_device (
    void * dest )
```

function to do a device memory allocation

Parameters

in	<i>dest</i>	- pointer on device to deallocate
----	-------------	-----------------------------------

7.12 find_E_grid_index.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
```

Functions

- `__global__ void find_E_grid_index_kernel` (unsigned `N`, `cross_section_data` *`d_xdata`, unsigned *`remap`, float *`E`, unsigned *`index`, unsigned *`rxn`)
- void `find_E_grid_index` (unsigned `NUM_THREADS`, unsigned `N`, `cross_section_data` *`d_xdata`, unsigned *`d_remap`, float *`d_E`, unsigned *`d_index`, unsigned *`d_rxn`)

a

7.12.1 Function Documentation

7.12.1.1 find_E_grid_index()

```
void find_E_grid_index (
    unsigned NUM_THREADS,
    unsigned N,
    cross_section_data * d_xsdata,
    unsigned * d_remap,
    float * d_E,
    unsigned * d_index,
    unsigned * d_rxn )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector
in	<i>d_E</i>	- device pointer to energy data array
in	<i>d_index</i>	- device pointer to index array (stores the unionized grid index of the current energy)
in	<i>d_rxn</i>	- device pointer of the reaction number array

7.12.1.2 find_E_grid_index_kernel()

```
__global__ void find_E_grid_index_kernel (
    unsigned N,
    cross_section_data * d_xsdata,
    unsigned * remap,
    float * E,
    unsigned * index,
    unsigned * rxn )
```

7.13 fission.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "wfloat3.h"
#include "warp_device.cuh"
#include "check_cuda.h"
```


Functions

- `__global__ void fission_kernel` (unsigned `N`, unsigned `starting_index`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, unsigned *`d_remap`)
- `void fission` (cudaStream_t `stream`, unsigned `NUM_THREADS`, unsigned `N`, unsigned `starting_index`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, unsigned *`d_remap`)

a

7.13.1 Function Documentation

7.13.1.1 fission()

```
void fission (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for fission
in	<i>starting_index</i>	- starting index of the fission block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.13.1.2 fission_kernel()

```
__global__ void fission_kernel (
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

7.14 hex_mesh.cu File Reference

```
#include <optix.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <optixu/optixu_aabb_namespace.h>
#include "datadef.h"
```

Functions

- [rtDeclareVariable](#) (optix::Ray, ray, rtCurrentRay,)
- [rtDeclareVariable](#) (unsigned, cellnum, attribute cell_num,)
- [rtDeclareVariable](#) (int, celltal, attribute cell_tal,)
- [rtDeclareVariable](#) (unsigned, cellmat, attribute cell_mat,)
- [rtDeclareVariable](#) (unsigned, cellfissile, attribute cell_fis,)
- [rtDeclareVariable](#) (unsigned, sense, attribute cell_sense,)
- [rtDeclareVariable](#) (float3, normal, attribute normal,)
- [rtDeclareVariable](#) (uint, launch_index_in, rtLaunchIndex,)
- static __device__ bool [accept_point](#) (float3 pnt, float a, float x1, float x2, float zmin, float zmax)
- static __device__ float [get_t](#) (float3 hat, float3 dir, float3 diff_points)
- RT_PROGRAM void [intersect](#) (int object_dex)
- RT_PROGRAM void [bounds](#) (int object_dex, float result[6])

Variables

- rtBuffer< [geom_data](#), 1 > [dims](#)

7.14.1 Function Documentation

7.14.1.1 [accept_point\(\)](#)

```
static __device__ bool accept_point (
    float3 pnt,
    float a,
    float x1,
    float x2,
    float zmin,
    float zmax ) [static]
```

7.14.1.2 [bounds\(\)](#)

```
RT_PROGRAM void bounds (
    int object_dex,
    float result[6] )
```

7.14.1.3 get_t()

```
static __device__ float get_t (
    float3 hat,
    float3 dir,
    float3 diff_points ) [static]
```

7.14.1.4 intersect()

```
RT_PROGRAM void intersect (
    int object_dex )
```

7.14.1.5 rtDeclareVariable() [1/8]

```
rtDeclareVariable (
    optix::Ray ,
    ray ,
    rtCurrentRay )
```

7.14.1.6 rtDeclareVariable() [2/8]

```
rtDeclareVariable (
    unsigned ,
    cellnum ,
    attribute cell_num )
```

7.14.1.7 rtDeclareVariable() [3/8]

```
rtDeclareVariable (
    int ,
    celltal ,
    attribute cell_tal )
```

7.14.1.8 rtDeclareVariable() [4/8]

```
rtDeclareVariable (
    unsigned ,
    cellmat ,
    attribute cell_mat )
```

7.14.1.9 rtDeclareVariable() [5/8]

```
rtDeclareVariable (
    unsigned ,
    cellfissile ,
    attribute cell_fis )
```

7.14.1.10 rtDeclareVariable() [6/8]

```
rtDeclareVariable (
    unsigned ,
    sense ,
    attribute cell_sense )
```

7.14.1.11 rtDeclareVariable() [7/8]

```
rtDeclareVariable (
    float3 ,
    normal ,
    attribute normal )
```

7.14.1.12 rtDeclareVariable() [8/8]

```
rtDeclareVariable (
    uint ,
    launch_index_in ,
    rtLaunchIndex )
```

7.14.2 Variable Documentation**7.14.2.1 dims**

```
rtBuffer<geom\_data,1> dims
```

7.15 hits_mesh.cu File Reference

```
#include <optix.h>
#include <optixu/optixu_math_namespace.h>
#include "datadef.h"
```

Functions

- [rtDeclareVariable](#) ([intersection_point](#), payload, rtPayload,)
- [rtDeclareVariable](#) (float, int_dist, rtIntersectionDistance,)
- [rtDeclareVariable](#) (optix::Ray, ray, rtCurrentRay,)
- [rtDeclareVariable](#) (unsigned, cellnum, attribute cell_num,)
- [rtDeclareVariable](#) (int, celltal, attribute cell_tal,)
- [rtDeclareVariable](#) (unsigned, cellmat, attribute cell_mat,)
- [rtDeclareVariable](#) (unsigned, cellfissile, attribute cell_fis,)
- [rtDeclareVariable](#) (unsigned, sense, attribute cell_sense,)
- [rtDeclareVariable](#) (float3, normal, attribute normal,)
- RT_PROGRAM void [closest_hit](#) ()

7.15.1 Function Documentation

7.15.1.1 [closest_hit\(\)](#)

```
RT_PROGRAM void closest_hit ( )
```

7.15.1.2 [rtDeclareVariable\(\)](#) [1/9]

```
rtDeclareVariable (
    intersection\_point ,
    payload ,
    rtPayload )
```

7.15.1.3 [rtDeclareVariable\(\)](#) [2/9]

```
rtDeclareVariable (
    float ,
    int_dist ,
    rtIntersectionDistance )
```

7.15.1.4 [rtDeclareVariable\(\)](#) [3/9]

```
rtDeclareVariable (
    optix::Ray ,
    ray ,
    rtCurrentRay )
```

7.15.1.5 rtDeclareVariable() [4/9]

```
rtDeclareVariable (
    unsigned ,
    cellnum ,
    attribute cell_num )
```

7.15.1.6 rtDeclareVariable() [5/9]

```
rtDeclareVariable (
    int ,
    celltal ,
    attribute cell_tal )
```

7.15.1.7 rtDeclareVariable() [6/9]

```
rtDeclareVariable (
    unsigned ,
    cellmat ,
    attribute cell_mat )
```

7.15.1.8 rtDeclareVariable() [7/9]

```
rtDeclareVariable (
    unsigned ,
    cellfissile ,
    attribute cell_fis )
```

7.15.1.9 rtDeclareVariable() [8/9]

```
rtDeclareVariable (
    unsigned ,
    sense ,
    attribute cell_sense )
```

7.15.1.10 rtDeclareVariable() [9/9]

```
rtDeclareVariable (
    float3 ,
    normal ,
    attribute normal )
```

7.16 macro_micro.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "warp_device.cuh"
#include "check_cuda.h"
```

Functions

- `__global__ void macro_micro_kernel` (unsigned `N`, unsigned `converged`, unsigned `n_materials`, unsigned `n_isotopes`, unsigned `n_tallies`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, `tally_data` *`d_tally`, unsigned *`d_remap`, float *`d_number_density_matrix`)
- void `macro_micro` (unsigned `NUM_THREADS`, unsigned `N`, unsigned `converged`, unsigned `n_materials`, unsigned `n_isotopes`, unsigned `n_tallies`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, `tally_data` *`d_tally`, unsigned *`d_remap`, float *`d_number_density_matrix`)

a

7.16.1 Function Documentation

7.16.1.1 macro_micro()

```
void macro_micro (
    unsigned NUM_THREADS,
    unsigned N,
    unsigned converged,
    unsigned n_materials,
    unsigned n_isotopes,
    unsigned n_tallies,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    tally_data * d_tally,
    unsigned * d_remap,
    float * d_number_density_matrix )
```

a

b

Parameters

in	<code>NUM_THREADS</code>	- the number of threads to run per thread block
in	<code>N</code>	- the total number of threads to launch on the grid
in	<code>converged</code>	- flag for tally scoring
in	<code>n_materials</code>	- number of materials
in	<code>n_isotopes</code>	- number of isotopes
in	<code>n_tallies</code>	- number of tallies
in	<code>d_xsdata</code>	- device pointer to cross section data pointer array
in	<code>d_particles</code>	- device pointer to particle data pointer array
in	<code>d_tally</code>	- device pointer to tally array
in	<code>d_remap</code>	- device pointer to data remapping vector
in	<code>d_number_density_matrix</code>	- device pointer to material number density array

7.16.1.2 macro_micro_kernel()

```
__global__ void macro_micro_kernel (
    unsigned N,
    unsigned converged,
    unsigned n_materials,
    unsigned n_tallies,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    tally_data * d_tally,
    unsigned * d_remap,
    float * d_number_density_matrix )
```

7.17 main.cpp File Reference

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

Functions

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

7.17.1 Function Documentation

7.17.1.1 main()

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

a

b

Parameters

in	<i>argc</i>	- argument count
in	<i>argv</i>	- arguments

7.18 miss.cu File Reference

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

Functions

- [rtDeclareVariable](#) (optix::Ray, ray, rtCurrentRay,)
- [rtDeclareVariable](#) (intersection_point, payload, rtPayload,)
- [rtDeclareVariable](#) (uint, launch_index_in, rtLaunchIndex,)
- [rtDeclareVariable](#) (unsigned, trace_type,,)
- [rtDeclareVariable](#) (unsigned, outer_cell,,)
- RT_PROGRAM void [miss](#) ()

Variables

- rtBuffer< unsigned, 1 > [rxn_buffer](#)
- rtBuffer< [spatial_data](#), 1 > [positions_buffer](#)
- rtBuffer< unsigned, 1 > [matnum_buffer](#)
- rtBuffer< unsigned, 1 > [cellnum_buffer](#)
- rtBuffer< unsigned, 1 > [remap_buffer](#)

7.18.1 Function Documentation

7.18.1.1 miss()

```
RT_PROGRAM void miss ( )
```

7.18.1.2 rtDeclareVariable() [1/5]

```
rtDeclareVariable (
    optix::Ray ,
    ray ,
    rtCurrentRay )
```

7.18.1.3 rtDeclareVariable() [2/5]

```
rtDeclareVariable (
    intersection_point ,
    payload ,
    rtPayload )
```

7.18.1.4 `rtDeclareVariable()` [3/5]

```
rtDeclareVariable (
    uint ,
    launch_index_in ,
    rtLaunchIndex )
```

7.18.1.5 `rtDeclareVariable()` [4/5]

```
rtDeclareVariable (
    unsigned ,
    trace_type )
```

7.18.1.6 `rtDeclareVariable()` [5/5]

```
rtDeclareVariable (
    unsigned ,
    outer_cell )
```

7.18.2 Variable Documentation

7.18.2.1 `cellnum_buffer`

```
rtBuffer<unsigned,1> cellnum_buffer
```

7.18.2.2 `matnum_buffer`

```
rtBuffer<unsigned,1> matnum_buffer
```

7.18.2.3 `positions_buffer`

```
rtBuffer<spatial_data,1> positions_buffer
```

7.18.2.4 remap_buffer

```
rtBuffer<unsigned,1> remap_buffer
```

7.18.2.5 rxn_buffer

```
rtBuffer<unsigned,1> rxn_buffer
```

7.19 mt19937ar.cpp File Reference

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

Macros

- `#define N 624`
- `#define M 397`
- `#define MATRIX_A 0x9908b0dfUL /* constant vector a */`
- `#define UPPER_MASK 0x80000000UL /* most significant w-r bits */`
- `#define LOWER_MASK 0x7fffffffUL /* least significant r bits */`

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`

7.19.1 Macro Definition Documentation

7.19.1.1 LOWER_MASK

```
#define LOWER_MASK 0x7fffffffUL /* least significant r bits */
```

7.19.1.2 M

```
#define M 397
```

7.19.1.3 MATRIX_A

```
#define MATRIX_A 0x9908b0dfUL /* constant vector a */
```

7.19.1.4 N

```
#define N 624
```

7.19.1.5 UPPER_MASK

```
#define UPPER_MASK 0x80000000UL /* most significant w-r bits */
```

7.19.2 Function Documentation

7.19.2.1 genrand_int31()

```
long genrand_int31 (  
    void )
```

7.19.2.2 genrand_int32()

```
unsigned long genrand_int32 (  
    void )
```

7.19.2.3 genrand_real1()

```
double genrand_real1 (  
    void )
```

7.19.2.4 genrand_real2()

```
double genrand_real2 (  
    void )
```

7.19.2.5 genrand_real3()

```
double genrand_real3 (  
    void )
```

7.19.2.6 genrand_res53()

```
double genrand_res53 (  
    void )
```

7.19.2.7 init_by_array()

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

7.19.2.8 init_genrand()

```
void init_genrand (  
    unsigned long s )
```

7.19.3 Variable Documentation

7.19.3.1 mt

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

7.19.3.2 mti

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

7.20 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 "wprimitive.h"
#include "wgeometry.h"
#include "optix_stuff.h"
#include "device_copies.h"
```

7.21 optix_stuff.h File Reference

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

Classes

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

7.22 optixtest.cpp File Reference

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

Functions

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

7.22.1 Function Documentation

7.22.1.1 main()

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

7.23 pop_fission.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "warp_device.cuh"
#include "check_cuda.h"
```

Functions

- `__global__ void pop_fission_kernel` (unsigned `N`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, unsigned *`d_scanned`, `spatial_data` *`fission_particles`, float *`fission_energy`)
- void `pop_fission` (unsigned `NUM_THREADS`, unsigned `N`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_`↵
_`particles`, unsigned *`d_scanned`, `spatial_data` *`fission_particles`, float *`fission_energy`)

a

7.23.1 Function Documentation

7.23.1.1 pop_fission()

```
void pop_fission (
    unsigned NUM_THREADS,
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_scanned,
    spatial_data * fission_particles,
    float * fission_energy )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_scanned</i>	- device pointer to array of the cumulative sum (scan) of the yield array, used to find final index where new particles will be written
in	<i>fission_particles</i>	- device pointer to intermediate spatial data array where popped values will be written
in	<i>fission_energy</i>	- device pointer to intermediate energy data array where popped values will be written

7.23.1.2 pop_fission_kernel()

```
__global__ void pop_fission_kernel (
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_scanned,
    spatial_data * fission_particles,
    float * fission_energy )
```

7.24 reaction_edges3.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "check_cuda.h"
```

Functions

- `__global__ void reaction_edges_kernel` (unsigned *N*, unsigned *edges, unsigned *rxn)
 - void `reaction_edges` (unsigned NUM_THREADS, unsigned *N*, unsigned *d_edges, unsigned *d_rxn)
- a*

7.24.1 Function Documentation

7.24.1.1 reaction_edges()

```
void reaction_edges (
    unsigned NUM_THREADS,
    unsigned N,
    unsigned * d_edges,
    unsigned * d_rxn )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>d_edges</i>	- device pointer to the edges array (stores to indices of edges the reaction blocks)
in	<i>d_rxn</i>	- device pointer of the reaction number array

7.24.1.2 reaction_edges_kernel()

```
__global__ void reaction_edges_kernel (
    unsigned N,
    unsigned * edges,
    unsigned * rxn )
```

7.25 rebase_yield.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "warp_device.cuh"
```

Functions

- `__global__ void rebase_yield_kernel` (unsigned *N*, float *keff*, [particle_data](#) **d_particles*)
- void [rebase_yield](#) (unsigned *NUM_THREADS*, unsigned *N*, float *keff*, [particle_data](#) **d_particles*)

7.25.1 Function Documentation

7.25.1.1 rebase_yield()

```
void rebase_yield (
    unsigned NUM_THREADS,
    unsigned N,
    float keff,
    particle\_data * d_particles )
```

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>keff</i>	- k-effective of the current cycle
in	<i>d_particles</i>	- device pointer to particle data pointer array

7.25.1.2 rebase_yield_kernel()

```
__global__ void rebase_yield_kernel (
    unsigned N,
    float keff,
    particle_data * d_particles )
```

7.26 safety_check.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "warp_device.cuh"
#include "check_cuda.h"
#include "wfloat3.h"
```

Functions

- `__global__ void safety_check_kernel` (unsigned `N`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, unsigned *`d_remap`)
- void `safety_check` (unsigned `NUM_THREADS`, unsigned `N`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, unsigned *`d_remap`)

a

7.26.1 Function Documentation

7.26.1.1 safety_check()

```
void safety_check (
    unsigned NUM_THREADS,
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<code>NUM_THREADS</code>	- the number of threads to run per thread block
in	<code>N</code>	- the total number of threads to launch on the grid
in	<code>d_xsdata</code>	- device pointer to cross section data pointer array
in	<code>d_particles</code>	- device pointer to particle data pointer array
in	<code>d_remap</code>	- device pointer to data remapping vector

7.26.1.2 safety_check_kernel()

```
__global__ void safety_check_kernel (
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

7.27 sample_fissile_energy.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "warp_device.cuh"
#include "check_cuda.h"
```

Functions

- `__global__ void sample_fissile_energy_kernel` (unsigned `N`, float `a`, float `b`, unsigned `*rn_bank`, float `*E`)
 - void `sample_fissile_energy` (unsigned `NUM_THREADS`, unsigned `N`, float `a`, float `b`, unsigned `*rn_bank`, float `*E`)
- `a`

7.27.1 Function Documentation

7.27.1.1 sample_fissile_energy()

```
void sample_fissile_energy (
    unsigned NUM_THREADS,
    unsigned N,
    float a,
    float b,
    unsigned * rn_bank,
    float * E )
```

`a`

`b`

Parameters

in	<code>NUM_THREADS</code>	- the number of threads to run per thread block
in	<code>N</code>	- the total number of threads to launch on the grid
in	<code>a</code>	- Watt spectrum, parameter a
in	<code>b</code>	- Watt spectrum, parameter b
in	<code>rn_bank</code>	- device pointer to random number array
in	<code>E</code>	- device pointer to energy data array

7.27.1.2 sample_fissile_energy_kernel()

```
__global__ void sample_fissile_energy_kernel (
    unsigned N,
    float a,
    float b,
    unsigned * rn_bank,
    float * E )
```

7.28 sample_fixed_source.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "LCRNG.cuh"
#include "check_cuda.h"
```

Functions

- `__global__ void sample_fixed_source_kernel` (unsigned [N](#), unsigned *active, unsigned *rn_bank, float *E, source_point *space)
- void [sample_fixed_source](#) (unsigned NUM_THREADS, unsigned [N](#), unsigned *active, unsigned *rn_bank, float *E, source_point *space)

a

7.28.1 Function Documentation

7.28.1.1 sample_fixed_source()

```
void sample_fixed_source (
    unsigned NUM_THREADS,
    unsigned N,
    unsigned * active,
    unsigned * rn_bank,
    float * E,
    source_point * space )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>active</i>	- device pointer to remapping vector
in	<i>rn_bank</i>	- device pointer to random number array
in	<i>E</i>	- device pointer to energy data array
in	<i>space</i>	- device pointer to spatial data array

7.28.1.2 sample_fixed_source_kernel()

```
__global__ void sample_fixed_source_kernel (
    unsigned N,
    unsigned * active,
    unsigned * rn_bank,
    float * E,
    source_point * space )
```

7.29 scatter_conti.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "wfloat3.h"
#include "warp_device.cuh"
#include "check_cuda.h"
```

Functions

- `__global__ void scatter_conti_kernel` (unsigned [N](#), unsigned starting_index, [cross_section_data](#) *d_xsdata, [particle_data](#) *d_particles, unsigned *d_remap)
- `void scatter_conti` (cudaStream_t stream, unsigned NUM_THREADS, unsigned [N](#), unsigned starting_index, [cross_section_data](#) *d_xsdata, [particle_data](#) *d_particles, unsigned *d_remap)

a

7.29.1 Function Documentation

7.29.1.1 scatter_conti()

```
void scatter_conti (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for continuum scattering
in	<i>starting_index</i>	- starting index of the continuum scatter block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.29.1.2 scatter_conti_kernel()

```
__global__ void scatter_conti_kernel (
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

7.30 scatter_level.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "wfloat3.h"
#include "warp_device.cuh"
#include "check_cuda.h"
```

Functions

- `__global__ void scatter_level_kernel` (unsigned *N*, unsigned *starting_index*, `cross_section_data` **d_xsdata*, `particle_data` **d_particles*, unsigned **d_remap*)
- `void scatter_level` (cudaStream_t *stream*, unsigned *NUM_THREADS*, unsigned *N*, unsigned *starting_index*, `cross_section_data` **d_xsdata*, `particle_data` **d_particles*, unsigned **d_remap*)

a

7.30.1 Function Documentation

7.30.1.1 scatter_level()

```
void scatter_level (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for level scattering
in	<i>starting_index</i>	- starting index of the level scatter block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.30.1.2 scatter_level_kernel()

```
__global__ void scatter_level_kernel (
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

7.31 scatter_multi.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "wfloat3.h"
#include "warp_device.cuh"
#include "check_cuda.h"
```

Functions

- `__global__ void scatter_multi_kernel` (unsigned `N`, unsigned `starting_index`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, unsigned *`d_remap`)
- `void scatter_multi` (`cudaStream_t` `stream`, unsigned `NUM_THREADS`, unsigned `N`, unsigned `starting_index`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, unsigned *`d_remap`)

a

7.31.1 Function Documentation

7.31.1.1 scatter_multi()

```
void scatter_multi (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for multiplicity scattering
in	<i>starting_index</i>	- starting index of the multiplicity scatter block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.31.1.2 scatter_multi_kernel()

```
__global__ void scatter_multi_kernel (
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```


7.32 set_positions_rand.cu File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
#include "warp_device.cuh"
```

Functions

- `__global__ void set_positions_rand_kernel` (unsigned `N`, unsigned `outer_cell_type`, `spatial_data` *`positions`, `ptr`, unsigned *`rn_bank`, float `x_min`, float `y_min`, float `z_min`, float `x_max`, float `y_max`, float `z_max`)
- void `set_positions_rand` (unsigned `NUM_THREADS`, unsigned `N`, unsigned `outer_cell_type`, `spatial_data` *`d_space`, unsigned *`d_rn_bank`, float *`outer_cell_dims`)
sets starting cycle points uniformly random

7.32.1 Function Documentation

7.32.1.1 set_positions_rand()

```
void set_positions_rand (
    unsigned NUM_THREADS,
    unsigned N,
    unsigned outer_cell_type,
    spatial_data * d_space,
    unsigned * d_rn_bank,
    float * outer_cell_dims )
```

sets starting cycle points uniformly random

sets starting cycle points uniformly random in the specified volume

Parameters

in	<code>NUM_THREADS</code>	- the number of threads to run per thread block
in	<code>N</code>	- the total number of threads to launch on the grid
in	<code>outer_cell_type</code>	- the outer cell type, sets the shape of the sampling
in	<code>d_space</code>	- device pointer to spatial data array
in	<code>d_rn_bank</code>	- device pointer to random number array
in	<code>outer_cell_dims</code>	- host pointer to array of outer cell extrema

7.32.1.2 set_positions_rand_kernel()

```
__global__ void set_positions_rand_kernel (
    unsigned N,
```

```

    unsigned outer_cell_type,
    spatial_data * positions_ptr,
    unsigned * rn_bank,
    float x_min,
    float y_min,
    float z_min,
    float x_max,
    float y_max,
    float z_max )

```

7.33 sphere_mesh.cu File Reference

```

#include <optix.h>
#include <optixu/optixu_math_namespace.h>
#include <optixu/optixu_matrix_namespace.h>
#include <optixu/optixu_aabb_namespace.h>
#include "datadef.h"

```

Functions

- [rtDeclareVariable](#) (optix::Ray, ray, rtCurrentRay,)
- [rtDeclareVariable](#) (unsigned, cellnum, attribute cell_num,)
- [rtDeclareVariable](#) (int, celltal, attribute cell_tal,)
- [rtDeclareVariable](#) (unsigned, cellmat, attribute cell_mat,)
- [rtDeclareVariable](#) (unsigned, cellfissile, attribute cell_fis,)
- [rtDeclareVariable](#) (unsigned, sense, attribute cell_sense,)
- [rtDeclareVariable](#) (float3, normal, attribute normal,)
- RT_PROGRAM void [intersect](#) (int object_dex)
- RT_PROGRAM void [bounds](#) (int object_dex, float result[6])

Variables

- rtBuffer< [geom_data](#), 1 > [dims](#)

7.33.1 Function Documentation

7.33.1.1 bounds()

```

RT_PROGRAM void bounds (
    int object_dex,
    float result[6] )

```

7.33.1.2 intersect()

```
RT_PROGRAM void intersect (
    int object_dex )
```

7.33.1.3 rtDeclareVariable() [1/7]

```
rtDeclareVariable (
    optix::Ray ,
    ray ,
    rtCurrentRay )
```

7.33.1.4 rtDeclareVariable() [2/7]

```
rtDeclareVariable (
    unsigned ,
    cellnum ,
    attribute cell_num )
```

7.33.1.5 rtDeclareVariable() [3/7]

```
rtDeclareVariable (
    int ,
    celltal ,
    attribute cell_tal )
```

7.33.1.6 rtDeclareVariable() [4/7]

```
rtDeclareVariable (
    unsigned ,
    cellmat ,
    attribute cell_mat )
```

7.33.1.7 rtDeclareVariable() [5/7]

```
rtDeclareVariable (
    unsigned ,
    cellfissile ,
    attribute cell_fis )
```

7.33.1.8 `rtDeclareVariable()` [6/7]

```
rtDeclareVariable (
    unsigned ,
    sense ,
    attribute cell_sense )
```

7.33.1.9 `rtDeclareVariable()` [7/7]

```
rtDeclareVariable (
    float3 ,
    normal ,
    attribute normal )
```

7.33.2 Variable Documentation

7.33.2.1 `dims`

```
rtBuffer<geom_data,1> dims
```

7.34 `test_function.cu` File Reference

```
#include <cuda.h>
#include <stdio.h>
#include "datadef.h"
```

Functions

- `__global__ void test_kernel` (unsigned `N`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, `tally_data` *`d_tally`, unsigned *`d_remap`)
- `void test_function` (unsigned `NUM_THREADS`, unsigned `N`, `cross_section_data` *`d_xsdata`, `particle_data` *`d_particles`, `tally_data` *`d_tally`, unsigned *`d_remap`)

7.34.1 Function Documentation

7.34.1.1 test_function()

```
void test_function (
    unsigned NUM_THREADS,
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    tally_data * d_tally,
    unsigned * d_remap )
```

7.34.1.2 test_kernel()

```
__global__ void test_kernel (
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    tally_data * d_tally,
    unsigned * d_remap )
```

7.35 unionize.py File Reference

Classes

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

Namespaces

- [unionize](#)

Variables

- bool [unionize.ace_available](#) = True

7.36 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 "wprimitive.h"
#include "wgeometry.h"
#include "warp_cuda.h"
#include "whistory.h"
```

7.37 warp_cuda.h File Reference

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

Functions

- void [write_to_file](#) (unsigned *, unsigned, std::string)
writes CUDA array to a text file
- void [set_positions_rand](#) (unsigned, unsigned, unsigned, [spatial_data](#) *, unsigned *, float *)
sets starting cycle points uniformly random
- void [copy_points](#) (unsigned, unsigned, unsigned *, unsigned, unsigned *, [spatial_data](#) *, [spatial_data](#) *, float *, float *)
copy points between two sets of space and energy data buffers, redirected with a mapping array
- void [sample_fixed_source](#) (unsigned, unsigned, unsigned *, unsigned *, float *, [spatial_data](#) *)
- void [macro_micro](#) (unsigned, unsigned, unsigned, unsigned, unsigned, unsigned, [cross_section_data](#) *, [particle_data](#) *, [tally_data](#) *, unsigned *, float *)
a
- void [scatter_level](#) (cudaStream_t, unsigned, unsigned, unsigned, [cross_section_data](#) *, [particle_data](#) *, unsigned *)
a
- void [scatter_conti](#) (cudaStream_t, unsigned, unsigned, unsigned, [cross_section_data](#) *, [particle_data](#) *, unsigned *)
a
- void [scatter_multi](#) (cudaStream_t, unsigned, unsigned, unsigned, [cross_section_data](#) *, [particle_data](#) *, unsigned *)
a
- void [fission](#) (cudaStream_t, unsigned, unsigned, unsigned, [cross_section_data](#) *, [particle_data](#) *, unsigned *)
a
- void [find_E_grid_index](#) (unsigned, unsigned, [cross_section_data](#) *, unsigned *, float *, unsigned *, unsigned *)
a
- void [pop_fission](#) (unsigned, unsigned, [cross_section_data](#) *, [particle_data](#) *, unsigned *, [spatial_data](#) *, float *)
a
- void [rebase_yield](#) (unsigned, unsigned, float, [particle_data](#) *)
- void [reaction_edges](#) (unsigned, unsigned, unsigned *, unsigned *)
a
- void [sample_fissile_energy](#) (unsigned, unsigned, float, float, unsigned *, float *)
a
- void [safety_check](#) (unsigned, unsigned, [cross_section_data](#) *, [particle_data](#) *, unsigned *)
a
- void [check_pointers](#) (unsigned, unsigned, unsigned, [cross_section_data](#) *)
a

7.37.1 Function Documentation

7.37.1.1 check_pointers()

```
void check_pointers (
    unsigned NUM_THREADS,
    unsigned dex0,
    unsigned dex1,
    cross_section_data * d_xsdata )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>dex0</i>	- starting index
in	<i>dex1</i>	- ending index
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array

7.37.1.2 copy_points()

```
void copy_points (
    unsigned NUM_THREADS,
    unsigned Nout,
    unsigned * Nvalid,
    unsigned current_index,
    unsigned * to_valid,
    spatial_data * positions_out,
    spatial_data * positions_in,
    float * E_out,
    float * E_in )
```

copy points between two sets of space and energy data buffers, redirected with a mapping array

copy points between two sets of space and energy data buffers, redirected with a mapping array

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>Nout</i>	- the total number of threads to launch on the grid
in	<i>Nvalid</i>	- the total number of device elements to copy from
in	<i>current_index</i>	- starting index
in	<i>to_valid</i>	- device pointer to data remapping vector
in	<i>positions_out</i>	- device pointer to spatial data array destination
in	<i>positions_in</i>	- device pointer to spatial data array source
in	<i>E_out</i>	- device pointer to energy data array destination
in	<i>E_in</i>	- device pointer to energy data array source

7.37.1.3 find_E_grid_index()

```
void find_E_grid_index (
    unsigned NUM_THREADS,
    unsigned N,
    cross_section_data * d_xsdata,
    unsigned * d_remap,
    float * d_E,
    unsigned * d_index,
    unsigned * d_rxn )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector
in	<i>d_E</i>	- device pointer to energy data array
in	<i>d_index</i>	- device pointer to index array (stores the unionized grid index of the current energy)
in	<i>d_rxn</i>	- device pointer of the reaction number array

7.37.1.4 fission()

```
void fission (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for fission
in	<i>starting_index</i>	- starting index of the fission block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.37.1.5 macro_micro()

```

void macro_micro (
    unsigned NUM_THREADS,
    unsigned N,
    unsigned converged,
    unsigned n_materials,
    unsigned n_isotopes,
    unsigned n_tallies,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    tally_data * d_tally,
    unsigned * d_remap,
    float * d_number_density_matrix )

```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>converged</i>	- flag for tally scoring
in	<i>n_materials</i>	- number of materials
in	<i>n_isotopes</i>	- number of isotopes
in	<i>n_tallies</i>	- number of tallies
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_tally</i>	- device pointer to tally array
in	<i>d_remap</i>	- device pointer to data remapping vector
in	<i>d_number_density_matrix</i>	- device pointer to material number density array

7.37.1.6 pop_fission()

```

void pop_fission (
    unsigned NUM_THREADS,
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_scanned,
    spatial_data * fission_particles,
    float * fission_energy )

```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_scanned</i>	- device pointer to array of the cumulative sum (scan) of the yield array, used to find final index where new particles will be written
in	<i>fission_particles</i>	- device pointer to intermediate spatial data array where popped values will be written
in	<i>fission_energy</i>	- device pointer to intermediate energy data array where popped values will be written

7.37.1.7 reaction_edges()

```
void reaction_edges (
    unsigned NUM_THREADS,
    unsigned N,
    unsigned * d_edges,
    unsigned * d_rxn )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>d_edges</i>	- device pointer to the edges array (stores to indices of edges the reaction blocks)
in	<i>d_rxn</i>	- device pointer of the reaction number array

7.37.1.8 rebase_yield()

```
void rebase_yield (
    unsigned NUM_THREADS,
    unsigned N,
    float keff,
    particle_data * d_particles )
```

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>keff</i>	- k-effective of the current cycle
in	<i>d_particles</i>	- device pointer to particle data pointer array

7.37.1.9 safety_check()

```
void safety_check (
    unsigned NUM_THREADS,
    unsigned N,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.37.1.10 sample_fissile_energy()

```
void sample_fissile_energy (
    unsigned NUM_THREADS,
    unsigned N,
    float a,
    float b,
    unsigned * rn_bank,
    float * E )
```

a

b

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>a</i>	- Watt spectrum, parameter a
in	<i>b</i>	- Watt spectrum, parameter b
in	<i>rn_bank</i>	- device pointer to random number array
in	<i>E</i>	- device pointer to energy data array

7.37.1.11 sample_fixed_source()

```
void sample_fixed_source (
    unsigned ,
    unsigned ,
    unsigned * ,
    unsigned * ,
    float * ,
    spatial_data * )
```

7.37.1.12 scatter_conti()

```
void scatter_conti (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for continuum scattering
in	<i>starting_index</i>	- starting index of the continuum scatter block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.37.1.13 scatter_level()

```
void scatter_level (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for level scattering
in	<i>starting_index</i>	- starting index of the level scatter block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.37.1.14 scatter_multi()

```
void scatter_multi (
    cudaStream_t stream,
    unsigned NUM_THREADS,
    unsigned N,
    unsigned starting_index,
    cross_section_data * d_xsdata,
    particle_data * d_particles,
    unsigned * d_remap )
```

a

b

Parameters

in	<i>stream</i>	- CUDA stream to launch the kernel on
in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid for multiplicity scattering
in	<i>starting_index</i>	- starting index of the multiplicity scatter block in the remap vector
in	<i>d_xsdata</i>	- device pointer to cross section data pointer array
in	<i>d_particles</i>	- device pointer to particle data pointer array
in	<i>d_remap</i>	- device pointer to data remapping vector

7.37.1.15 set_positions_rand()

```
void set_positions_rand (
    unsigned NUM_THREADS,
    unsigned N,
    unsigned outer_cell_type,
    spatial_data * d_space,
    unsigned * d_rn_bank,
    float * outer_cell_dims )
```

sets starting cycle points uniformly random

sets starting cycle points uniformly random in the specified volume

Parameters

in	<i>NUM_THREADS</i>	- the number of threads to run per thread block
in	<i>N</i>	- the total number of threads to launch on the grid
in	<i>outer_cell_type</i>	- the outer cell type, sets the shape of the sampling
in	<i>d_space</i>	- device pointer to spatial data array
in	<i>d_rn_bank</i>	- device pointer to random number array
in	<i>outer_cell_dims</i>	- host pointer to array of outer cell extrema

7.37.1.16 write_to_file()

```
void write_to_file (
    unsigned * array_in,
    unsigned N,
    std::string filename )
```

writes CUDA array to a text file

copies the cuda array to a local buffer, writes to buffer to a new file, then frees the local memory

Parameters

in	<i>array_in</i>	- device pointer to array to write
in	<i>N</i>	- number of elements to write
in	<i>filename</i>	- name for the file

7.38 warp_device.cuh File Reference**7.39 wfloat3.h File Reference****Classes**

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

7.40 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 "wprimitive.h"
#include "wgeometry.h"
```

7.41 wgeometry.h File Reference

```
#include "wprimitive.h"
```

Classes

- class [wgeometry](#)
wgeometry class

7.42 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.h>
#include <Python.h>
#include <png++/png.hpp>
#include "datadef.h"
#include "wprimitive.h"
#include "wgeometry.h"
#include "optix_stuff.h"
#include "warp_cuda.h"
#include "whistory.h"
#include "check_cuda.h"
```

Variables

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

7.42.1 Variable Documentation

7.42.1.1 `optix_obj`

`optix_stuff` `optix_obj`

7.43 `whistory.h` File Reference

Classes

- class `whistory`
whistory class

7.44 `wprimitive.cpp` File Reference

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

7.45 `wprimitive.h` File Reference

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

Classes

- class `primitive`
primitive class

7.46 `write_to_file.cu` File Reference

```
#include <iostream>
#include <sstream>
#include <stdio.h>
#include <cuda.h>
```

Functions

- void `write_to_file` (unsigned *array_in, unsigned N, std::string filename)
writes CUDA array to a text file

7.46.1 Function Documentation

7.46.1.1 write_to_file()

```
void write_to_file (
    unsigned * array_in,
    unsigned N,
    std::string filename )
```

writes CUDA array to a text file

copies the cuda array to a local buffer, writes to buffer to a new file, then frees the local memory

Parameters

in	<i>array_{in}</i>	- device pointer to array to write
in	<i>N</i>	- number of elements to write
in	<i>filename</i>	- name for the file

Index

- `__del__`
 - `ace::Library`, [39](#)
- `__init__`
 - `ace::AceTable`, [14](#)
 - `ace::DosimetryTable`, [32](#)
 - `ace::ElectronTable`, [33](#)
 - `ace::EnergyDistribution`, [34](#)
 - `ace::Library`, [39](#)
 - `ace::NeutronDiscreteTable`, [43](#)
 - `ace::NeutronMGTable`, [44](#)
 - `ace::NeutronTable`, [47](#)
 - `ace::PhotoatomicMGTable`, [67](#)
 - `ace::PhotoatomicTable`, [68](#)
 - `ace::PhotonuclearTable`, [69](#)
 - `ace::Reaction`, [76](#)
 - `ace::SabTable`, [80](#)
 - `unionize::cross_section_data`, [19](#)
- `__iter__`
 - `ace::NeutronTable`, [47](#)
- `__repr__`
 - `ace::DosimetryTable`, [32](#)
 - `ace::ElectronTable`, [33](#)
 - `ace::NeutronDiscreteTable`, [43](#)
 - `ace::NeutronMGTable`, [44](#)
 - `ace::NeutronTable`, [47](#)
 - `ace::PhotoatomicMGTable`, [67](#)
 - `ace::PhotoatomicTable`, [68](#)
 - `ace::PhotonuclearTable`, [69](#)
 - `ace::Reaction`, [77](#)
 - `ace::SabTable`, [80](#)
- `_add_isotope`
 - `unionize::cross_section_data`, [20](#)
- `_allocate_arrays`
 - `unionize::cross_section_data`, [20](#)
- `_get_MT_array_pointer`
 - `unionize::cross_section_data`, [22](#)
- `_get_MT_numbers_pointer`
 - `unionize::cross_section_data`, [22](#)
- `_get_MT_numbers_total_pointer`
 - `unionize::cross_section_data`, [23](#)
- `_get_Q_pointer`
 - `unionize::cross_section_data`, [23](#)
- `_get_awr_pointer`
 - `unionize::cross_section_data`, [20](#)
- `_get_energy_data`
 - `unionize::cross_section_data`, [21](#)
- `_get_energy_distribution`
 - `ace::NeutronTable`, [47](#)
- `_get_length_numbers_pointer`
 - `unionize::cross_section_data`, [21](#)
- `_get_main_Egrid_pointer`
 - `unionize::cross_section_data`, [21](#)
- `_get_scatter_data`
 - `unionize::cross_section_data`, [23](#)
- `_get_temp_pointer`
 - `unionize::cross_section_data`, [24](#)
- `_init_from_string`
 - `unionize::cross_section_data`, [24](#)
- `_insert_reactions`
 - `unionize::cross_section_data`, [24](#)
- `_interpolate`
 - `unionize::cross_section_data`, [25](#)
- `_print_isotopes`
 - `unionize::cross_section_data`, [25](#)
- `_read_all`
 - `ace::AceTable`, [14](#)
 - `ace::NeutronTable`, [47](#)
 - `ace::SabTable`, [80](#)
- `_read_andp`
 - `ace::NeutronTable`, [47](#)
- `_read_angular_distributions`
 - `ace::NeutronTable`, [48](#)
- `_read_ascii`
 - `ace::Library`, [39](#)
- `_read_binary`
 - `ace::Library`, [39](#)
- `_read_cross_sections`
 - `ace::NeutronTable`, [48](#)
- `_read_energy_distributions`
 - `ace::NeutronTable`, [48](#)
- `_read_fis`
 - `ace::NeutronTable`, [48](#)
- `_read_gpd`
 - `ace::NeutronTable`, [48](#)
- `_read_itca`
 - `ace::SabTable`, [80](#)
- `_read_itce`
 - `ace::SabTable`, [80](#)
- `_read_itie`
 - `ace::SabTable`, [81](#)
- `_read_itxe`
 - `ace::SabTable`, [81](#)
- `_read_landp`
 - `ace::NeutronTable`, [49](#)
- `_read_lsigp`
 - `ace::NeutronTable`, [49](#)
- `_read_mtrp`
 - `ace::NeutronTable`, [49](#)

- `_read_nu`
 - `ace::NeutronTable`, 49
- `_read_sigp`
 - `ace::NeutronTable`, 49
- `_read_tables`
 - `unionize::cross_section_data`, 25
- `_read_unr`
 - `ace::NeutronTable`, 50
- `_read_yp`
 - `ace::NeutronTable`, 50
- `_resolve_library`
 - `unionize::cross_section_data`, 26
- `_unionize`
 - `unionize::cross_section_data`, 26
- `~optix_stuff`
 - `optix_stuff`, 57
- `~primitive`
 - `primitive`, 71
- `~wgeometry`
 - `wgeometry`, 96
- `~whistory`
 - `whistory`, 112
- `a_dist_energy_in`
 - `ace::NeutronTable`, 50
- `a_dist_mu_out`
 - `ace::NeutronTable`, 50
- `accel_type`
 - `optix_stuff`, 61
 - `whistory`, 128
- `accept_point`
 - `hex_mesh.cu`, 162
- `accumulate_keff`
 - `whistory`, 112
- `accumulate_tally`
 - `whistory`, 113
- `ace`, 9
 - `ascii_to_binary`, 10
 - `fromstring_split`, 10
 - `fromstring_token`, 11
 - `NP_LE_V15`, 11
 - `table_types`, 11
- `ace.AceTable`, 13
- `ace.DosimetryTable`, 32
- `ace.ElectronTable`, 33
- `ace.EnergyDistribution`, 34
- `ace.Library`, 38
- `ace.NeutronDiscreteTable`, 42
- `ace.NeutronMGTable`, 43
- `ace.NeutronTable`, 44
- `ace.PhotoatomicMGTable`, 67
- `ace.PhotoatomicTable`, 68
- `ace.PhotonuclearTable`, 69
- `ace.pyx`, 143
- `ace.Reaction`, 75
- `ace.SabTable`, 78
- `ace::AceTable`
 - `__init__`, 14
 - `_read_all`, 14
- `awr`, 14
 - `name`, 14
 - `temp`, 15
- `ace::DosimetryTable`
 - `__init__`, 32
 - `__repr__`, 32
- `ace::ElectronTable`
 - `__init__`, 33
 - `__repr__`, 33
- `ace::EnergyDistribution`
 - `__init__`, 34
- `ace::Library`
 - `__del__`, 39
 - `__init__`, 39
 - `_read_ascii`, 39
 - `_read_binary`, 39
 - `binary`, 40
 - `f`, 40
 - `find_table`, 39
 - `read`, 39
 - `tables`, 40
 - `verbose`, 40
- `ace::NeutronDiscreteTable`
 - `__init__`, 43
 - `__repr__`, 43
- `ace::NeutronMGTable`
 - `__init__`, 44
 - `__repr__`, 44
- `ace::NeutronTable`
 - `__init__`, 47
 - `__iter__`, 47
 - `__repr__`, 47
 - `_get_energy_distribution`, 47
 - `_read_all`, 47
 - `_read_andp`, 47
 - `_read_angular_distributions`, 48
 - `_read_cross_sections`, 48
 - `_read_energy_distributions`, 48
 - `_read_fis`, 48
 - `_read_gpd`, 48
 - `_read_landp`, 49
 - `_read_lsipg`, 49
 - `_read_mtrp`, 49
 - `_read_nu`, 49
 - `_read_sigp`, 49
 - `_read_unr`, 50
 - `_read_yp`, 50
 - `a_dist_energy_in`, 50
 - `a_dist_mu_out`, 50
 - `e_dist_EG`, 51
 - `e_dist_LP`, 51
 - `e_dist_energy_outNE`, 51
 - `find_reaction`, 50
 - `heating`, 51
 - `IE_fission`, 51
 - `MT_for_photon_yield`, 51
 - `nu_d_energy`, 51
 - `nu_d_energy_dist`, 51

- nu_d_interp_INT, [52](#)
- nu_d_interp_NBT, [52](#)
- nu_d_precursor_const, [52](#)
- nu_d_precursor_energy, [52](#)
- nu_d_precursor_interp_INT, [52](#)
- nu_d_precursor_interp_NBT, [52](#)
- nu_d_precursor_prob, [52](#)
- nu_d_value, [52](#)
- nu_p_energy, [53](#)
- nu_p_interp_INT, [53](#)
- nu_p_interp_NBT, [53](#)
- nu_p_type, [53](#)
- nu_p_value, [53](#)
- nu_t_energy, [53](#)
- nu_t_interp_INT, [53](#)
- nu_t_interp_NBT, [53](#)
- nu_t_type, [54](#)
- nu_t_value, [54](#)
- photon_reactions, [54](#)
- reactions, [54](#)
- sigma, [54](#)
- sigma_f, [54](#)
- sigma_photon, [54](#)
- urr_energy, [54](#)
- urr_table, [55](#)
- ace::PhotoatomicMGTable
 - __init__, [67](#)
 - __repr__, [67](#)
- ace::PhotoatomicTable
 - __init__, [68](#)
 - __repr__, [68](#)
- ace::PhotonuclearTable
 - __init__, [69](#)
 - __repr__, [69](#)
- ace::Reaction
 - __init__, [76](#)
 - __repr__, [77](#)
 - broaden, [77](#)
 - IE, [77](#)
 - MT, [77](#)
 - Q, [78](#)
 - sigma, [78](#)
 - table, [78](#)
 - threshold, [77](#)
 - TY, [78](#)
- ace::SabTable
 - __init__, [80](#)
 - __repr__, [80](#)
 - _read_all, [80](#)
 - _read_itca, [80](#)
 - _read_itce, [80](#)
 - _read_itie, [81](#)
 - _read_itxe, [81](#)
 - elastic_e_in, [81](#)
 - elastic_mu_out, [81](#)
 - elastic_P, [81](#)
 - elastic_type, [81](#)
 - inelastic_e_in, [82](#)
 - inelastic_e_out, [82](#)
 - inelastic_mu_out, [82](#)
 - inelastic_sigma, [82](#)
- ace_available
 - unionize, [12](#)
- add_material
 - wgeometry, [96](#)
- add_primitive
 - wgeometry, [97](#)
- add_tally
 - wgeometry, [97](#)
- add_transform
 - primitive, [71](#), [72](#)
 - wgeometry, [97](#), [98](#)
- allocate_on_device
 - device_copies.cu, [156](#)
 - device_copies.h, [158](#)
- ascii_to_binary
 - ace, [10](#)
- awr
 - ace::AceTable, [14](#)
 - cross_section_data, [15](#)
 - unionize::cross_section_data, [26](#)
- awr_list
 - wgeometry, [104](#)
- benchmarks.cpp, [144](#)
 - main, [144](#)
- bin_fission_points
 - whistory, [113](#)
- binary
 - ace::Library, [40](#)
- blks
 - whistory, [128](#)
- boundary_condition
 - optix_stuff, [61](#)
 - wgeometry, [104](#)
- bounds
 - box_mesh.cu, [145](#)
 - cylinder_mesh.cu, [153](#)
 - hex_mesh.cu, [162](#)
 - sphere_mesh.cu, [186](#)
- box_mesh.cu, [144](#)
 - bounds, [145](#)
 - boxnormal, [145](#)
 - dims, [146](#)
 - intersect, [145](#)
 - rtDeclareVariable, [145](#), [146](#)
- boxnormal
 - box_mesh.cu, [145](#)
- broaden
 - ace::Reaction, [77](#)
- camera
 - camera.cu, [147](#)
- camera.cu, [147](#)
 - camera, [147](#)
 - cellnum_buffer, [149](#)
 - exception, [147](#)

- matnum_buffer, 149
- positions_buffer, 149
- remap_buffer, 149
- rtDeclareVariable, 147, 148
- rxn_buffer, 149
- talnum_buffer, 149
- cdf
 - dist_data, 30
- cell
 - intersection_point, 36
 - tally_data, 85
 - tally_data_host, 86
- cell_num_list
 - wgeometry, 104
- cellmat
 - wtransform, 140
- cellnum
 - geom_data, 35
 - particle_data, 65
 - wtransform, 140
- cellnum_buffer
 - camera.cu, 149
 - miss.cu, 170
- cellnum_ptr
 - optix_stuff, 62
- check
 - wgeometry, 99
- check_cuda
 - check_cuda.h, 150
- check_cuda.h, 149
 - check_cuda, 150
- check_fissile
 - wgeometry, 99
- check_pointers
 - check_pointers.cu, 151
 - warp_cuda.h, 190
- check_pointers.cu, 150
 - check_pointers, 151
 - check_pointers_kernel, 151
- check_pointers_kernel
 - check_pointers.cu, 151
- closest_hit
 - hits_mesh.cu, 165
- compact_config
 - whistory, 128
- compactplan
 - whistory, 129
- compute_device
 - optix_stuff, 62
 - whistory, 129
- concentrations_matrix
 - wgeometry, 105
- context
 - optix_stuff, 62
- converge
 - whistory, 113
- copy_data_to_device
 - whistory, 113
- copy_energy_data
 - whistory, 113
- copy_from_device
 - device_copies.cu, 156
 - device_copies.h, 158
- copy_points
 - copy_points.cu, 152
 - warp_cuda.h, 191
- copy_points.cu, 151
 - copy_points, 152
 - copy_points_kernel, 152
- copy_points_kernel
 - copy_points.cu, 152
- copy_python_buffer
 - whistory, 114, 115
- copy_scatter_data
 - whistory, 115
- copy_to_device
 - device_copies.cu, 157
 - device_copies.h, 158
- count
 - tally_data, 85
 - tally_data_host, 86
- count_total
 - tally_data_host, 87
- cross
 - wfloat3, 90
- cross_section_data, 15
 - awr, 15
 - dist_energy, 16
 - dist_scatter, 16
 - energy_grid, 16
 - energy_grid_len, 16
 - n_isotopes, 16
 - Q, 16
 - rxn_numbers, 16
 - rxn_numbers_total, 16
 - temp, 17
 - total_reaction_channels, 17
 - xs, 17
- cylinder_mesh.cu, 153
 - bounds, 153
 - dims, 155
 - intersect, 153
 - rtDeclareVariable, 154, 155
- d_bank_E
 - whistory, 129
- d_bank_space
 - whistory, 129
- d_edges
 - whistory, 129
- d_fissile_energy
 - whistory, 129
- d_fissile_points
 - whistory, 129
- d_isotope_list
 - whistory, 129
- d_material_list

- whistory, [130](#)
- d_num_active
 - whistory, [130](#)
- d_num_completed
 - whistory, [130](#)
- d_number_density_matrix
 - whistory, [130](#)
- d_particles
 - whistory, [130](#)
- d_reduced_weight
 - whistory, [130](#)
- d_reduced_yields
 - whistory, [130](#)
- d_remap
 - whistory, [130](#)
- d_scanned
 - whistory, [131](#)
- d_tally
 - whistory, [131](#)
- d_valid_N
 - whistory, [131](#)
- d_valid_result
 - whistory, [131](#)
- d_xsdata
 - whistory, [131](#)
- d_zeros
 - whistory, [131](#)
- datadef.h, [155](#)
- datapath
 - unionize::cross_section_data, [26](#)
 - wgeometry, [105](#)
- deallocate_on_device
 - device_copies.cu, [157](#)
 - device_copies.h, [159](#)
- delete_primitive
 - wgeometry, [99](#)
- delete_transform
 - wgeometry, [100](#)
- density
 - material_def, [41](#)
- device_copies.cu, [156](#)
 - allocate_on_device, [156](#)
 - copy_from_device, [156](#)
 - copy_to_device, [157](#)
 - deallocate_on_device, [157](#)
- device_copies.h, [157](#)
 - allocate_on_device, [158](#)
 - copy_from_device, [158](#)
 - copy_to_device, [158](#)
 - deallocate_on_device, [159](#)
- device_report
 - whistory, [115](#)
- dh_dist_energy
 - whistory, [131](#)
- dh_dist_scatter
 - whistory, [131](#)
- dh_particles
 - whistory, [132](#)
- dh_tally
 - whistory, [132](#)
- dh_xsdata
 - whistory, [132](#)
- dims
 - box_mesh.cu, [146](#)
 - cylinder_mesh.cu, [155](#)
 - hex_mesh.cu, [164](#)
 - sphere_mesh.cu, [188](#)
- dist_container, [29](#)
 - lower, [30](#)
 - upper, [30](#)
- dist_data, [30](#)
 - cdf, [30](#)
 - erg, [31](#)
 - intt, [31](#)
 - law, [31](#)
 - len, [31](#)
 - pdf, [31](#)
 - var, [31](#)
- dist_energy
 - cross_section_data, [16](#)
- dist_scatter
 - cross_section_data, [16](#)
- done_ptr
 - optix_stuff, [62](#)
- dot
 - wfloat3, [90](#)
- dump_flag
 - whistory, [132](#)
- dx
 - wtransform, [141](#)
- dy
 - wtransform, [141](#)
- dz
 - wtransform, [141](#)
- E
 - particle_data, [65](#)
- e_dist_EG
 - ace::NeutronTable, [51](#)
- e_dist_LP
 - ace::NeutronTable, [51](#)
- e_dist_energy_outNE
 - ace::NeutronTable, [51](#)
- E_max
 - tally_data, [85](#)
 - tally_data_host, [87](#)
- E_min
 - tally_data, [85](#)
 - tally_data_host, [87](#)
- edges
 - whistory, [132](#)
- elastic_e_in
 - ace::SabTable, [81](#)
- elastic_mu_out
 - ace::SabTable, [81](#)
- elastic_P
 - ace::SabTable, [81](#)

- elastic_type
 - ace::SabTable, [81](#)
- energy_grid
 - cross_section_data, [16](#)
- energy_grid_len
 - cross_section_data, [16](#)
- enforce_BC
 - spatial_data, [83](#)
- erg
 - dist_data, [31](#)
- exception
 - camera.cu, [147](#)
- f
 - ace::Library, [40](#)
- filename
 - whistory, [132](#)
- find_E_grid_index
 - find_E_grid_index.cu, [159](#)
 - warp_cuda.h, [191](#)
- find_E_grid_index.cu, [159](#)
 - find_E_grid_index, [159](#)
 - find_E_grid_index_kernel, [160](#)
- find_E_grid_index_kernel
 - find_E_grid_index.cu, [160](#)
- find_reaction
 - ace::NeutronTable, [50](#)
- find_table
 - ace::Library, [39](#)
- fiss
 - intersection_point, [36](#)
- fiss_img
 - whistory, [132](#)
- fissile_flag
 - wgeometry, [105](#)
- fission
 - fission.cu, [161](#)
 - warp_cuda.h, [192](#)
- fission.cu, [160](#)
 - fission, [161](#)
 - fission_kernel, [161](#)
- fission_kernel
 - fission.cu, [161](#)
- foncs
 - whistory, [132](#)
- fractions
 - material_def, [41](#)
- fromstring_split
 - ace, [10](#)
- fromstring_token
 - ace, [11](#)
- GEOM_FLAG
 - optix_stuff, [62](#)
- genrand_int31
 - mt19937ar.cpp, [172](#)
- genrand_int32
 - mt19937ar.cpp, [172](#)
- genrand_real1
 - mt19937ar.cpp, [172](#)
- genrand_real2
 - mt19937ar.cpp, [173](#)
- genrand_real3
 - mt19937ar.cpp, [173](#)
- genrand_res53
 - mt19937ar.cpp, [173](#)
- geom_data, [34](#)
 - cellnum, [35](#)
 - is_fissile, [35](#)
 - loc, [35](#)
 - matnum, [35](#)
 - max, [35](#)
 - min, [35](#)
 - talnum, [35](#)
- get_Py_buffer_dims
 - whistory, [115](#)
- get_boundary_condition
 - wgeometry, [100](#)
- get_material_count
 - wgeometry, [100](#)
- get_material_table
 - wgeometry, [100](#)
- get_maximum_cell
 - wgeometry, [101](#)
- get_maximum_material
 - wgeometry, [101](#)
- get_minimum_cell
 - wgeometry, [101](#)
- get_minimum_material
 - wgeometry, [101](#)
- get_outer_cell
 - optix_stuff, [57](#)
 - wgeometry, [101](#)
- get_outer_cell_dims
 - wgeometry, [101](#)
- get_outer_cell_type
 - optix_stuff, [57](#)
 - wgeometry, [102](#)
- get_primitive_count
 - wgeometry, [102](#)
- get_rand
 - optix_stuff, [57](#)
- get_t
 - hex_mesh.cu, [162](#)
- get_time
 - whistory, [115](#)
- get_transform_count
 - wgeometry, [102](#)
- h_particles
 - whistory, [133](#)
- h_tally
 - whistory, [133](#)
- h_xsdata
 - whistory, [133](#)
- heating
 - ace::NeutronTable, [51](#)
- hex_mesh.cu, [162](#)

- accept_point, [162](#)
- bounds, [162](#)
- dims, [164](#)
- get_t, [162](#)
- intersect, [163](#)
- rtDeclareVariable, [163](#), [164](#)
- hits_mesh.cu, [164](#)
 - closest_hit, [165](#)
 - rtDeclareVariable, [165](#), [166](#)
- hot2
 - whistory, [116](#)
- IE_fission
 - ace::NeutronTable, [51](#)
- id
 - material_def, [41](#)
- IE
 - ace::Reaction, [77](#)
- index
 - particle_data, [65](#)
- inelastic_e_in
 - ace::SabTable, [82](#)
- inelastic_e_out
 - ace::SabTable, [82](#)
- inelastic_mu_out
 - ace::SabTable, [82](#)
- inelastic_sigma
 - ace::SabTable, [82](#)
- init
 - optix_stuff, [58](#)
 - whistory, [116](#)
- init_CUDPP
 - whistory, [116](#)
- init_RNG
 - whistory, [117](#)
- init_by_array
 - mt19937ar.cpp, [173](#)
- init_cross_sections
 - whistory, [116](#)
- init_device
 - whistory, [117](#)
- init_genrand
 - mt19937ar.cpp, [173](#)
- init_host
 - whistory, [117](#)
- init_internal
 - optix_stuff, [58](#)
- init_python
 - whistory, [117](#)
- initial_weight_total
 - whistory, [133](#)
- intersect
 - box_mesh.cu, [145](#)
 - cylinder_mesh.cu, [153](#)
 - hex_mesh.cu, [163](#)
 - sphere_mesh.cu, [186](#)
- intersection_point, [36](#)
 - cell, [36](#)
 - fiss, [36](#)
 - mat, [37](#)
 - norm, [37](#)
 - sense, [37](#)
 - surf_dist, [37](#)
 - tally_index, [37](#)
 - x, [37](#)
 - y, [37](#)
 - z, [37](#)
- intt
 - dist_data, [31](#)
- is_fissile
 - geom_data, [35](#)
 - material_def, [41](#)
- is_initialized
 - whistory, [133](#)
- isonum
 - particle_data, [65](#)
- isotope_list
 - unionize::cross_section_data, [27](#)
 - wgeometry, [105](#)
 - whistory, [133](#)
- isotope_list_array
 - wgeometry, [105](#)
- isotopes
 - material_def, [42](#)
 - wgeometry, [105](#)
 - whistory, [133](#)
- isotropic_tol
 - unionize::cross_section_data, [27](#)
- keff2_sum
 - whistory, [133](#)
- keff_err
 - whistory, [134](#)
- keff_sum
 - whistory, [134](#)
- LOWER_MASK
 - mt19937ar.cpp, [171](#)
- last_loaded
 - unionize::cross_section_data, [27](#)
- law
 - dist_data, [31](#)
- len
 - dist_data, [31](#)
- length
 - tally_data, [85](#)
 - tally_data_host, [87](#)
- libraries
 - unionize::cross_section_data, [27](#)
- loc
 - geom_data, [35](#)
- location
 - primitive, [74](#)
- lower
 - dist_container, [30](#)
- M
 - mt19937ar.cpp, [172](#)

- MATRIX_A
 - mt19937ar.cpp, 172
- MT_E_grid
 - unionize::cross_section_data, 27
- MT_array
 - unionize::cross_section_data, 27
- MT_columns
 - whistory, 134
- MT_for_photon_yield
 - ace::NeutronTable, 51
- MT_rows
 - whistory, 134
- macro_micro
 - macro_micro.cu, 167
 - warp_cuda.h, 193
- macro_micro.cu, 167
 - macro_micro, 167
 - macro_micro_kernel, 168
- macro_micro_kernel
 - macro_micro.cu, 168
- main
 - benchmarks.cpp, 144
 - main.cpp, 168
 - optixtest.cpp, 175
- main.cpp, 168
 - main, 168
- make_color
 - optix_stuff, 59
 - whistory, 117
- make_geom_prim
 - optix_stuff, 59
- make_geom_xform
 - optix_stuff, 59
- make_geom_xform_common
 - optix_stuff, 60
- make_hex_array
 - primitive, 73
 - wgeometry, 102
- make_material_table
 - wgeometry, 103
- map_active
 - whistory, 118
- mat
 - intersection_point, 37
- material
 - primitive, 74
- material_def, 41
 - density, 41
 - fractions, 41
 - id, 41
 - is_fissile, 41
 - isotopes, 42
 - matnum, 42
 - num_isotopes, 42
- material_list
 - whistory, 134
- material_list_array
 - wgeometry, 105
- material_num_list
 - wgeometry, 105
- materials
 - wgeometry, 106
- matnum
 - geom_data, 35
 - material_def, 42
 - particle_data, 66
- matnum_buffer
 - camera.cu, 149
 - miss.cu, 170
- matnum_ptr
 - optix_stuff, 62
- max
 - geom_data, 35
 - primitive, 74
- maxcell
 - optix_stuff, 62
- mem_free
 - whistory, 134
- mem_total
 - whistory, 134
- min
 - geom_data, 35
 - primitive, 74
- mincell
 - optix_stuff, 62
- miss
 - miss.cu, 169
- miss.cu, 169
 - cellnum_buffer, 170
 - matnum_buffer, 170
 - miss, 169
 - positions_buffer, 170
 - remap_buffer, 170
 - rtDeclareVariable, 169, 170
 - rxn_buffer, 171
- mones
 - whistory, 134
- MT
 - ace::Reaction, 77
- mt
 - mt19937ar.cpp, 173
- mt19937ar.cpp, 171
 - genrand_int31, 172
 - genrand_int32, 172
 - genrand_real1, 172
 - genrand_real2, 173
 - genrand_real3, 173
 - genrand_res53, 173
 - init_by_array, 173
 - init_genrand, 173
 - LOWER_MASK, 171
 - M, 172
 - MATRIX_A, 172
 - mt, 173
 - mti, 174
 - N, 172

- UPPER_MASK, [172](#)
- mti
 - mt19937ar.cpp, [174](#)
- N
 - mt19937ar.cpp, [172](#)
 - optix_stuff, [63](#)
 - whistory, [135](#)
- n_box
 - wgeometry, [106](#)
- n_cycles
 - whistory, [135](#)
- n_cyl
 - wgeometry, [106](#)
- n_edges
 - whistory, [135](#)
- n_hex
 - wgeometry, [106](#)
- n_isotopes
 - cross_section_data, [16](#)
 - wgeometry, [106](#)
 - whistory, [135](#)
- n_materials
 - optix_stuff, [63](#)
 - wgeometry, [106](#)
 - whistory, [135](#)
- n_primitives
 - wgeometry, [106](#)
- n_skip
 - whistory, [135](#)
- n_sph
 - wgeometry, [106](#)
- n_tallies
 - wgeometry, [107](#)
 - whistory, [135](#)
- n_transforms
 - primitive, [74](#)
 - wgeometry, [107](#)
- NP_LE_V15
 - ace, [11](#)
- NUM_THREADS
 - whistory, [136](#)
- name
 - ace::AceTable, [14](#)
- Ndataset
 - whistory, [135](#)
- nonzero
 - whistory, [118](#)
- norm
 - intersection_point, [37](#)
 - spatial_data, [83](#)
- norm2
 - wfloat3, [90](#)
- nu_d_energy
 - ace::NeutronTable, [51](#)
- nu_d_energy_dist
 - ace::NeutronTable, [51](#)
- nu_d_interp_INT
 - ace::NeutronTable, [52](#)
- nu_d_interp_NBT
 - ace::NeutronTable, [52](#)
- nu_d_precursor_const
 - ace::NeutronTable, [52](#)
- nu_d_precursor_energy
 - ace::NeutronTable, [52](#)
- nu_d_precursor_interp_INT
 - ace::NeutronTable, [52](#)
- nu_d_precursor_interp_NBT
 - ace::NeutronTable, [52](#)
- nu_d_precursor_prob
 - ace::NeutronTable, [52](#)
- nu_d_value
 - ace::NeutronTable, [52](#)
- nu_p_energy
 - ace::NeutronTable, [53](#)
- nu_p_interp_INT
 - ace::NeutronTable, [53](#)
- nu_p_interp_NBT
 - ace::NeutronTable, [53](#)
- nu_p_type
 - ace::NeutronTable, [53](#)
- nu_p_value
 - ace::NeutronTable, [53](#)
- nu_t_energy
 - ace::NeutronTable, [53](#)
- nu_t_interp_INT
 - ace::NeutronTable, [53](#)
- nu_t_interp_NBT
 - ace::NeutronTable, [53](#)
- nu_t_type
 - ace::NeutronTable, [54](#)
- nu_t_value
 - ace::NeutronTable, [54](#)
- num_isotopes
 - material_def, [42](#)
 - unionize::cross_section_data, [28](#)
- num_main_E
 - unionize::cross_section_data, [28](#)
- num_primitives
 - primitive, [74](#)
- num_reactions
 - unionize::cross_section_data, [28](#)
- number_density_matrix
 - whistory, [136](#)
- ones
 - whistory, [136](#)
- operator*
 - wfloat3, [90, 91](#)
- operator+
 - wfloat3, [91, 92](#)
- operator-
 - wfloat3, [92](#)
- operator/
 - wfloat3, [93](#)
- optix_device
 - optix_stuff, [63](#)
- optix_obj

- whistory.cpp, 199
- optix_stuff, 55
 - ~optix_stuff, 57
 - accel_type, 61
 - boundary_condition, 61
 - cellnum_ptr, 62
 - compute_device, 62
 - context, 62
 - done_ptr, 62
 - GEOM_FLAG, 62
 - get_outer_cell, 57
 - get_outer_cell_type, 57
 - get_rand, 57
 - init, 58
 - init_internal, 58
 - make_color, 59
 - make_geom_prim, 59
 - make_geom_xform, 59
 - make_geom_xform_common, 60
 - matnum_ptr, 62
 - maxcell, 62
 - mincell, 62
 - N, 63
 - n_materials, 63
 - optix_device, 63
 - optix_stuff, 56, 57
 - outer_cell, 63
 - outer_cell_dims, 63
 - outer_cell_type, 63
 - positions_ptr, 63
 - print, 60
 - remap_ptr, 63
 - rxn_ptr, 64
 - set_trace_type, 60
 - stack_size_multiplier, 64
 - talnum_ptr, 64
 - trace, 60, 61
 - traverse_type, 64
- optix_stuff.cpp, 174
- optix_stuff.h, 174
- optixtest.cpp, 174
 - main, 175
- outer_cell
 - optix_stuff, 63
 - wgeometry, 107
 - whistory, 136
- outer_cell_dims
 - optix_stuff, 63
 - whistory, 136
- outer_cell_type
 - optix_stuff, 63
 - whistory, 136
- particle_data, 64
 - cellnum, 65
 - E, 65
 - index, 65
 - isonum, 65
 - matnum, 66
- Q, 66
 - rn_bank, 66
 - rxn, 66
 - space, 66
 - talnum, 66
 - weight, 66
 - yield, 66
- pdf
 - dist_data, 31
- phi
 - wtransform, 141
- photon_reactions
 - ace::NeutronTable, 54
- plot_geom
 - whistory, 118
- pop_fission
 - pop_fission.cu, 175
 - warp_cuda.h, 193
- pop_fission.cu, 175
 - pop_fission, 175
 - pop_fission_kernel, 176
- pop_fission_kernel
 - pop_fission.cu, 176
- positions_buffer
 - camera.cu, 149
 - miss.cu, 170
- positions_ptr
 - optix_stuff, 63
- prep_secondaries
 - whistory, 119
- primitive, 70
 - ~primitive, 71
 - add_transform, 71, 72
 - location, 74
 - make_hex_array, 73
 - material, 74
 - max, 74
 - min, 74
 - n_transforms, 74
 - num_primitives, 74
 - primitive, 71
 - primitive_id, 74
 - print_transform, 73
 - transforms, 75
 - type, 75
- primitive_id
 - primitive, 74
- primitives
 - wgeometry, 107
- print
 - optix_stuff, 60
- print_all
 - wgeometry, 103
- print_banner
 - whistory, 119
- print_flag
 - whistory, 136
- print_materials_table

- wgeometry, 103
- whistory, 119
- print_pointers
 - whistory, 119
- print_summary
 - wgeometry, 103
- print_transform
 - primitive, 73
- print_xs_data
 - whistory, 119
- problem_geom
 - whistory, 136
- Q
 - ace::Reaction, 78
 - cross_section_data, 16
 - particle_data, 66
 - unionize::cross_section_data, 28
- RUN_FLAG
 - whistory, 138
- radix_config
 - whistory, 137
- radixplan
 - whistory, 137
- rand_gen
 - whistory, 137
- reaction_edges
 - reaction_edges3.cu, 176
 - warp_cuda.h, 194
- reaction_edges3.cu, 176
 - reaction_edges, 176
 - reaction_edges_kernel, 177
- reaction_edges_kernel
 - reaction_edges3.cu, 177
- reaction_numbers
 - unionize::cross_section_data, 28
- reaction_numbers_total
 - unionize::cross_section_data, 28
- reactions
 - ace::NeutronTable, 54
- read
 - ace::Library, 39
- rebase_yield
 - rebase_yield.cu, 177
 - warp_cuda.h, 194
- rebase_yield.cu, 177
 - rebase_yield, 177
 - rebase_yield_kernel, 178
- rebase_yield_kernel
 - rebase_yield.cu, 178
- redu_float_config
 - whistory, 137
- redu_int_config
 - whistory, 137
- reduce_done
 - whistory, 119
- reduce_weight
 - whistory, 120
- reduce_yield
 - whistory, 120
- reduced_weight
 - whistory, 137
- reduced_weight_total
 - whistory, 137
- reduced_yields
 - whistory, 137
- reduced_yields_total
 - whistory, 138
- reduplan_float
 - whistory, 138
- reduplan_int
 - whistory, 138
- remap
 - whistory, 138
- remap_active
 - whistory, 120
- remap_buffer
 - camera.cu, 149
 - miss.cu, 170
- remap_ptr
 - optix_stuff, 63
- res
 - whistory, 138
- reset_cycle
 - whistory, 121
- reset_fixed
 - whistory, 121
- rn_bank
 - particle_data, 66
- rodrigues_rotation
 - wfloat3, 93
- rotate
 - wfloat3, 94
- rtDeclareVariable
 - box_mesh.cu, 145, 146
 - camera.cu, 147, 148
 - cylinder_mesh.cu, 154, 155
 - hex_mesh.cu, 163, 164
 - hits_mesh.cu, 165, 166
 - miss.cu, 169, 170
 - sphere_mesh.cu, 187, 188
- run
 - whistory, 121
- rxn
 - particle_data, 66
- rxn_buffer
 - camera.cu, 149
 - miss.cu, 171
- rxn_numbers
 - cross_section_data, 16
- rxn_numbers_total
 - cross_section_data, 16
- rxn_ptr
 - optix_stuff, 64
- safety_check
 - safety_check.cu, 178

- warp_cuda.h, 195
- safety_check.cu, 178
 - safety_check, 178
 - safety_check_kernel, 179
- safety_check_kernel
 - safety_check.cu, 179
- sample_fissile_energy
 - sample_fissile_energy.cu, 179
 - warp_cuda.h, 195
- sample_fissile_energy.cu, 179
 - sample_fissile_energy, 179
 - sample_fissile_energy_kernel, 180
- sample_fissile_energy_kernel
 - sample_fissile_energy.cu, 180
- sample_fissile_points
 - whistory, 121
- sample_fixed_source
 - sample_fixed_source.cu, 180
 - warp_cuda.h, 195
- sample_fixed_source.cu, 180
 - sample_fixed_source, 180
 - sample_fixed_source_kernel, 181
- sample_fixed_source_kernel
 - sample_fixed_source.cu, 181
- scan_int_config
 - whistory, 138
- scanplan_int
 - whistory, 138
- scatter_conti
 - scatter_conti.cu, 181
 - warp_cuda.h, 196
- scatter_conti.cu, 181
 - scatter_conti, 181
 - scatter_conti_kernel, 182
- scatter_conti_kernel
 - scatter_conti.cu, 182
- scatter_level
 - scatter_level.cu, 183
 - warp_cuda.h, 196
- scatter_level.cu, 182
 - scatter_level, 183
 - scatter_level_kernel, 183
- scatter_level_kernel
 - scatter_level.cu, 183
- scatter_multi
 - scatter_multi.cu, 184
 - warp_cuda.h, 197
- scatter_multi.cu, 183
 - scatter_multi, 184
 - scatter_multi_kernel, 184
- scatter_multi_kernel
 - scatter_multi.cu, 184
- score
 - tally_data, 85
 - tally_data_host, 87
- score_total
 - tally_data_host, 87
- sense
 - intersection_point, 37
- set_acceration
 - whistory, 122
- set_datapath
 - wgeometry, 103
- set_device
 - whistory, 122
- set_dump_level
 - whistory, 122
- set_filename
 - whistory, 123
- set_outer_cell
 - wgeometry, 104
- set_positions_rand
 - set_positions_rand.cu, 185
 - warp_cuda.h, 197
- set_positions_rand.cu, 185
 - set_positions_rand, 185
 - set_positions_rand_kernel, 185
- set_positions_rand_kernel
 - set_positions_rand.cu, 185
- set_print_level
 - whistory, 123
- set_run_param
 - whistory, 123
- set_run_type
 - whistory, 123, 124
- set_trace_type
 - optix_stuff, 60
- sigma
 - ace::NeutronTable, 54
 - ace::Reaction, 78
- sigma_f
 - ace::NeutronTable, 54
- sigma_photon
 - ace::NeutronTable, 54
- space
 - particle_data, 66
- spatial_data, 82
 - enforce_BC, 83
 - norm, 83
 - surf_dist, 83
 - weight, 83
 - x, 83
 - xhat, 83
 - y, 83
 - yhat, 84
 - z, 84
 - zhat, 84
- sphere_mesh.cu, 186
 - bounds, 186
 - dims, 188
 - intersect, 186
 - rtDeclareVariable, 187, 188
- square
 - tally_data, 85
 - tally_data_host, 87
- square_total

- tally_data_host, [87](#)
- stack_size_multiplier
 - optix_stuff, [64](#)
- stream
 - whistory, [139](#)
- surf_dist
 - intersection_point, [37](#)
 - spatial_data, [83](#)
- table
 - ace::Reaction, [78](#)
- table_types
 - ace, [11](#)
- tables
 - ace::Library, [40](#)
 - unionize::cross_section_data, [29](#)
- tally_cells
 - wgeometry, [107](#)
- tally_data, [84](#)
 - cell, [85](#)
 - count, [85](#)
 - E_max, [85](#)
 - E_min, [85](#)
 - length, [85](#)
 - score, [85](#)
 - square, [85](#)
- tally_data_host, [86](#)
 - cell, [86](#)
 - count, [86](#)
 - count_total, [87](#)
 - E_max, [87](#)
 - E_min, [87](#)
 - length, [87](#)
 - score, [87](#)
 - score_total, [87](#)
 - square, [87](#)
 - square_total, [87](#)
- tally_index
 - intersection_point, [37](#)
 - wtransform, [141](#)
- talnum
 - geom_data, [35](#)
 - particle_data, [66](#)
- talnum_buffer
 - camera.cu, [149](#)
- talnum_ptr
 - optix_stuff, [64](#)
- temp
 - ace::AceTable, [15](#)
 - cross_section_data, [17](#)
 - unionize::cross_section_data, [29](#)
- test_function
 - test_function.cu, [188](#)
- test_function.cu, [188](#)
 - test_function, [188](#)
 - test_kernel, [189](#)
- test_kernel
 - test_function.cu, [189](#)
- theCudpp
 - whistory, [139](#)
- theta
 - wtransform, [141](#)
- threshold
 - ace::Reaction, [77](#)
- total_bytes_energy
 - whistory, [139](#)
- total_bytes_scatter
 - whistory, [139](#)
- total_reaction_channels
 - cross_section_data, [17](#)
- trace
 - optix_stuff, [60, 61](#)
 - whistory, [124](#)
- transforms
 - primitive, [75](#)
- traverse_type
 - optix_stuff, [64](#)
- TY
 - ace::Reaction, [78](#)
- type
 - primitive, [75](#)
- UPPER_MASK
 - mt19937ar.cpp, [172](#)
- unionize, [12](#)
 - ace_available, [12](#)
- unionize.cross_section_data, [17](#)
- unionize.py, [189](#)
- unionize::cross_section_data
 - __init__, [19](#)
 - _add_isotope, [20](#)
 - _allocate_arrays, [20](#)
 - _get_MT_array_pointer, [22](#)
 - _get_MT_numbers_pointer, [22](#)
 - _get_MT_numbers_total_pointer, [23](#)
 - _get_Q_pointer, [23](#)
 - _get_awr_pointer, [20](#)
 - _get_energy_data, [21](#)
 - _get_length_numbers_pointer, [21](#)
 - _get_main_Egrid_pointer, [21](#)
 - _get_scatter_data, [23](#)
 - _get_temp_pointer, [24](#)
 - _init_from_string, [24](#)
 - _insert_reactions, [24](#)
 - _interpolate, [25](#)
 - _print_isotopes, [25](#)
 - _read_tables, [25](#)
 - _resolve_library, [26](#)
 - _unionize, [26](#)
 - awr, [26](#)
 - datapath, [26](#)
 - isotope_list, [27](#)
 - isotropic_tol, [27](#)
 - last_loaded, [27](#)
 - libraries, [27](#)
 - MT_E_grid, [27](#)
 - MT_array, [27](#)
 - num_isotopes, [28](#)

- num_main_E, 28
- num_reactions, 28
- Q, 28
- reaction_numbers, 28
- reaction_numbers_total, 28
- tables, 29
- temp, 29
- xsdirstring, 29
- update
 - wgeometry, 104
- update_RNG
 - whistory, 125
- upper
 - dist_container, 30
- urr_energy
 - ace::NeutronTable, 54
- urr_table
 - ace::NeutronTable, 55
- var
 - dist_data, 31
- verbose
 - ace::Library, 40
- warp.h, 189
- warp_cuda.h, 190
 - check_pointers, 190
 - copy_points, 191
 - find_E_grid_index, 191
 - fission, 192
 - macro_micro, 193
 - pop_fission, 193
 - reaction_edges, 194
 - rebase_yield, 194
 - safety_check, 195
 - sample_fissile_energy, 195
 - sample_fixed_source, 195
 - scatter_conti, 196
 - scatter_level, 196
 - scatter_multi, 197
 - set_positions_rand, 197
 - write_to_file, 198
- warp_device.cuh, 198
- weight
 - particle_data, 66
 - spatial_data, 83
- wfloat3, 88
 - cross, 90
 - dot, 90
 - norm2, 90
 - operator*, 90, 91
 - operator+, 91, 92
 - operator-, 92
 - operator/, 93
 - rodrigues_rotation, 93
 - rotate, 94
 - wfloat3, 89
 - x, 94
 - y, 94
 - z, 94
- wfloat3.h, 198
- wgeometry, 95
 - ~wgeometry, 96
 - add_material, 96
 - add_primitive, 97
 - add_tally, 97
 - add_transform, 97, 98
 - awr_list, 104
 - boundary_condition, 104
 - cell_num_list, 104
 - check, 99
 - check_fissile, 99
 - concentrations_matrix, 105
 - datapath, 105
 - delete_primitive, 99
 - delete_transform, 100
 - fissile_flag, 105
 - get_boundary_condition, 100
 - get_material_count, 100
 - get_material_table, 100
 - get_maximum_cell, 101
 - get_maximum_material, 101
 - get_minimum_cell, 101
 - get_minimum_material, 101
 - get_outer_cell, 101
 - get_outer_cell_dims, 101
 - get_outer_cell_type, 102
 - get_primitive_count, 102
 - get_transform_count, 102
 - isotope_list, 105
 - isotope_list_array, 105
 - isotopes, 105
 - make_hex_array, 102
 - make_material_table, 103
 - material_list_array, 105
 - material_num_list, 105
 - materials, 106
 - n_box, 106
 - n_cyl, 106
 - n_hex, 106
 - n_isotopes, 106
 - n_materials, 106
 - n_primitives, 106
 - n_sph, 106
 - n_tallies, 107
 - n_transforms, 107
 - outer_cell, 107
 - primitives, 107
 - print_all, 103
 - print_materials_table, 103
 - print_summary, 103
 - set_datapath, 103
 - set_outer_cell, 104
 - tally_cells, 107
 - update, 104
 - wgeometry, 96
- wgeometry.cpp, 198

wgeometry.h, 199
whistory, 107
 ~whistory, 112
 accel_type, 128
 accumulate_keff, 112
 accumulate_tally, 113
 bin_fission_points, 113
 blks, 128
 compact_config, 128
 compactplan, 129
 compute_device, 129
 converge, 113
 copy_data_to_device, 113
 copy_energy_data, 113
 copy_python_buffer, 114, 115
 copy_scatter_data, 115
 d_bank_E, 129
 d_bank_space, 129
 d_edges, 129
 d_fissile_energy, 129
 d_fissile_points, 129
 d_isotope_list, 129
 d_material_list, 130
 d_num_active, 130
 d_num_completed, 130
 d_number_density_matrix, 130
 d_particles, 130
 d_reduced_weight, 130
 d_reduced_yields, 130
 d_remap, 130
 d_scanned, 131
 d_tally, 131
 d_valid_N, 131
 d_valid_result, 131
 d_xsdata, 131
 d_zeros, 131
 device_report, 115
 dh_dist_energy, 131
 dh_dist_scatter, 131
 dh_particles, 132
 dh_tally, 132
 dh_xsdata, 132
 dump_flag, 132
 edges, 132
 filename, 132
 fiss_img, 132
 fones, 132
 get_Py_buffer_dims, 115
 get_time, 115
 h_particles, 133
 h_tally, 133
 h_xsdata, 133
 hot2, 116
 init, 116
 init_CUDPP, 116
 init_RNG, 117
 init_cross_sections, 116
 init_device, 117
 init_host, 117
 init_python, 117
 initial_weight_total, 133
 is_initialized, 133
 isotope_list, 133
 isotopes, 133
 keff2_sum, 133
 keff_err, 134
 keff_sum, 134
 MT_columns, 134
 MT_rows, 134
 make_color, 117
 map_active, 118
 material_list, 134
 mem_free, 134
 mem_total, 134
 mones, 134
 N, 135
 n_cycles, 135
 n_edges, 135
 n_isotopes, 135
 n_materials, 135
 n_skip, 135
 n_tallies, 135
 NUM_THREADS, 136
 Ndataset, 135
 nonzero, 118
 number_density_matrix, 136
 ones, 136
 outer_cell, 136
 outer_cell_dims, 136
 outer_cell_type, 136
 plot_geom, 118
 prep_secondaries, 119
 print_banner, 119
 print_flag, 136
 print_materials_table, 119
 print_pointers, 119
 print_xs_data, 119
 problem_geom, 136
 RUN_FLAG, 138
 radix_config, 137
 radixplan, 137
 rand_gen, 137
 redu_float_config, 137
 redu_int_config, 137
 reduce_done, 119
 reduce_weight, 120
 reduce_yield, 120
 reduced_weight, 137
 reduced_weight_total, 137
 reduced_yields, 137
 reduced_yields_total, 138
 reduplan_float, 138
 reduplan_int, 138
 remap, 138
 remap_active, 120
 res, 138

- reset_cycle, [121](#)
- reset_fixed, [121](#)
- run, [121](#)
- sample_fissile_points, [121](#)
- scan_int_config, [138](#)
- scanplan_int, [138](#)
- set_acceration, [122](#)
- set_device, [122](#)
- set_dump_level, [122](#)
- set_filename, [123](#)
- set_print_level, [123](#)
- set_run_param, [123](#)
- set_run_type, [123](#), [124](#)
- stream, [139](#)
- theCudpp, [139](#)
- total_bytes_energy, [139](#)
- total_bytes_scatter, [139](#)
- trace, [124](#)
- update_RNG, [125](#)
- whistory, [112](#)
- write_fission_points, [125](#)
- write_results, [125](#)
- write_tally, [125](#)
- write_to_file, [125](#), [127](#)
- write_xs_data, [128](#)
- xs_isotope_ints, [139](#)
- xs_num_rxns, [139](#)
- xsdats_instance, [139](#)
- zeros, [139](#)
- whistory.cpp, [199](#)
 - optix_obj, [199](#)
- whistory.h, [200](#)
- wprimitive.cpp, [200](#)
- wprimitive.h, [200](#)
- write_fission_points
 - whistory, [125](#)
- write_results
 - whistory, [125](#)
- write_tally
 - whistory, [125](#)
- write_to_file
 - warp_cuda.h, [198](#)
 - whistory, [125](#), [127](#)
 - write_to_file.cu, [201](#)
- write_to_file.cu, [200](#)
 - write_to_file, [201](#)
- write_xs_data
 - whistory, [128](#)
- wtransform, [140](#)
 - cellmat, [140](#)
 - cellnum, [140](#)
 - dx, [141](#)
 - dy, [141](#)
 - dz, [141](#)
 - phi, [141](#)
 - tally_index, [141](#)
 - theta, [141](#)
- intersection_point, [37](#)
- spatial_data, [83](#)
- wfloat3, [94](#)
- xhat
 - spatial_data, [83](#)
- xs
 - cross_section_data, [17](#)
- xs_isotope_ints
 - whistory, [139](#)
- xs_num_rxns
 - whistory, [139](#)
- xsdats_instance
 - whistory, [139](#)
- xsdirstring
 - unionize::cross_section_data, [29](#)
- y
 - intersection_point, [37](#)
 - spatial_data, [83](#)
 - wfloat3, [94](#)
- yhat
 - spatial_data, [84](#)
- yield
 - particle_data, [66](#)
- z
 - intersection_point, [37](#)
 - spatial_data, [84](#)
 - wfloat3, [94](#)
- zeros
 - whistory, [139](#)
- zhat
 - spatial_data, [84](#)