

SlowMC

Generated by Doxygen 1.7.3

Tue Mar 20 2012 18:15:19

Contents

1 SlowMC: Slowing Down Monte Carlo	1
1.1 Overview	1
1.2 Compiling	1
1.3 Running	2
2 Directory Hierarchy	2
2.1 Directories	2
3 Modules Index	2
3.1 Modules List	2
4 Data Type Index	3
4.1 Class List	3
5 File Index	3
5.1 File List	3
6 Directory Documentation	5
6.1 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ Directory Reference	5
6.2 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/ Directory Reference	6
6.3 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/ Directory Reference	6
7 Module Documentation	7
7.1 global Module Reference	7
7.1.1 Detailed Description	8
7.1.2 Function/Subroutine Documentation	8
7.1.3 Variable Documentation	11
7.2 input Module Reference	14
7.2.1 Detailed Description	14
7.2.2 Function/Subroutine Documentation	14
7.3 materials Module Reference	15
7.3.1 Detailed Description	16
7.3.2 Function/Subroutine Documentation	16
7.4 output Module Reference	18
7.4.1 Detailed Description	19
7.4.2 Function/Subroutine Documentation	19
7.5 particle Module Reference	21
7.5.1 Detailed Description	21
7.5.2 Function/Subroutine Documentation	21
7.6 physics Module Reference	21
7.6.1 Detailed Description	22
7.6.2 Function/Subroutine Documentation	22
7.7 read_xml_primitives Module Reference	26
7.7.1 Function/Subroutine Documentation	27
7.8 tally Module Reference	35
7.8.1 Detailed Description	35

7.8.2	Function/Subroutine Documentation	36
7.9	timing Module Reference	38
7.9.1	Function/Subroutine Documentation	39
7.9.2	Variable Documentation	41
7.10	write_xml_primitives Module Reference	41
7.10.1	Function/Subroutine Documentation	42
7.11	xml_data_input_t Module Reference	48
7.11.1	Function/Subroutine Documentation	49
7.11.2	Variable Documentation	65
7.12	xmlparse Module Reference	66
7.12.1	Function/Subroutine Documentation	66
7.12.2	Variable Documentation	72
8	Data Type Documentation	73
8.1	datafunc Interface Reference	73
8.1.1	Detailed Description	73
8.1.2	Constructor & Destructor Documentation	74
8.2	endfunc Interface Reference	74
8.2.1	Detailed Description	74
8.2.2	Constructor & Destructor Documentation	74
8.3	xml_data_input_t::material_xml Type Reference	75
8.3.1	Detailed Description	76
8.3.2	Member Data Documentation	76
8.4	xml_data_input_t::nuclide_xml Type Reference	76
8.4.1	Detailed Description	76
8.4.2	Member Data Documentation	76
8.5	read_xml_primitives::read_from_buffer Interface Reference	77
8.5.1	Detailed Description	77
8.5.2	Member Function/Subroutine Documentation	77
8.6	xml_data_input_t::settings_xml Type Reference	78
8.6.1	Detailed Description	78
8.6.2	Member Data Documentation	78
8.7	startfunc Interface Reference	79
8.7.1	Detailed Description	79
8.7.2	Constructor & Destructor Documentation	79
8.8	xml_data_input_t::tallies_xml Type Reference	80
8.8.1	Detailed Description	80
8.8.2	Member Data Documentation	80
8.9	xml_data_input_t::tally_xml Type Reference	80
8.9.1	Detailed Description	81
8.9.2	Member Data Documentation	81
8.10	timing::Timer Type Reference	81
8.10.1	Detailed Description	81
8.10.2	Member Data Documentation	82
8.11	write_xml_primitives::write_to_xml_line Interface Reference	82
8.11.1	Detailed Description	82
8.11.2	Member Function/Subroutine Documentation	82
8.12	write_xml_primitives::write_to_xml_word Interface Reference	83
8.12.1	Detailed Description	83
8.12.2	Member Function/Subroutine Documentation	83

8.13	xmlparse::XML_PARSE Type Reference	83
8.13.1	Detailed Description	83
8.13.2	Member Data Documentation	84
8.14	xmlparse::xml_report_details Interface Reference	85
8.14.1	Detailed Description	85
8.14.2	Member Function/Subroutine Documentation	85
8.15	xmlparse::xml_report_errors Interface Reference	86
8.15.1	Detailed Description	86
8.15.2	Member Function/Subroutine Documentation	86
9	File Documentation	87
9.1	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/global.f90 File Reference	87
9.2	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/input.f90 File Reference	88
9.3	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/main.f90 File Reference	88
9.3.1	Function Documentation	88
9.4	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/materials.f90 File Reference	92
9.5	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/output.f90 File Reference	93
9.6	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/particle.f90 File Reference	93
9.7	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/physics.f90 File Reference	93
9.8	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/tally.f90 File Reference	94
9.9	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/timing.f90 File Reference	95
9.10	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/read_- xml_primitives.f90 File Reference	95
9.11	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/input_- t.f90 File Reference	96
9.12	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/write_- xml_primitives.f90 File Reference	98
9.13	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/xmlparse.f90 File Reference	99
9.14	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/xmlreader.f90 File Reference	100
9.14.1	Function Documentation	101

1 SlowMC: Slowing Down Monte Carlo

1.1 Overview

This program solves the slowing down neutron transport equation in either infinite medium or effective two-region collision probability theory. It models parts of the

same physics performed by the NJOY data processing code. This code is for strictly academic purposes and allows the user to see the relative impact of physics in the generation of multigroup cross sections and on flux spectra. This code currently uses the following external libraries:

- HDF5 v1.8.#

The package HDF5 can be downloaded from <http://www.hdfgroup.org/HDF5/>

1.2 Compiling

Compiling is as easy as running the Makefile with:

```
make xml-fortran  
make slowmc
```

1.3 Running

To run SlowMC, execute the following:

```
slowmc
```

2 Directory Hierarchy

2.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

src	5
xml-fortran	6
templates	6

3 Modules Index

3.1 Modules List

Here is a list of all modules with brief descriptions:

global (Contains all of the global variables)	7
input (Handles reading in the input xml file and initializing global vars)	14

materials (Contains information about the isotopes of problem)	15
output (Contains routines for outputting major info to user)	18
particle (Contains information about the particle that is transporting)	21
physics (Contains routines to model the physics of the problem)	21
read_xml_primitives	26
tally (Contains information about tallying quantities)	35
timing	38
write_xml_primitives	41
xml_data_input_t	48
xmlparse	66

4 Data Type Index

4.1 Class List

Here are the data types with brief descriptions:

datafunc	73
endfunc	74
xml_data_input_t::material_xml	75
xml_data_input_t::materials_xml	??
xml_data_input_t::nuclide_xml	76
read_xml_primitives::read_from_buffer	77
xml_data_input_t::settings_xml	78
startfunc	79
xml_data_input_t::tallies_xml	80
xml_data_input_t::tally_xml	80
timing::Timer	81
write_xml_primitives::write_to_xml_line	82
write_xml_primitives::write_to_xml_word	83

xmlparse::XML_PARSE	83
xmlparse::xml_report_details	85
xmlparse::xml_report_errors	86

5 File Index

5.1 File List

Here is a list of all files with brief descriptions:

/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ global.f90	87
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ input.f90	88
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ main.f90	88
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ materials.f90	92
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ output.f90	93
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ particle.f90	93
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ physics.f90	93
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ tally.f90	94
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ timing.f90	95
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/ read_xml_primitives.f90	95
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/ write_xml_primitives.f90	98
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/ xmlparse.f90	99
/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/ xmlreader.f90	100

/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/[input_f90](#) 96

6 Directory Documentation

6.1 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/ Directory Reference

Directory dependency graph for /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/:



Directories

- directory [xml-fortran](#)

Files

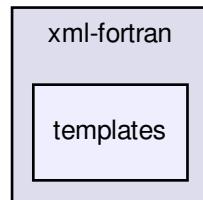
- file [global.f90](#)
- file [input.f90](#)
- file [main.f90](#)
- file [materials.f90](#)
- file [output.f90](#)
- file [particle.f90](#)
- file [physics.f90](#)
- file [tally.f90](#)
- file [timing.f90](#)

6.2 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/ Directory Reference

6

6.2 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/ Directory Reference

Directory dependency graph for /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/:

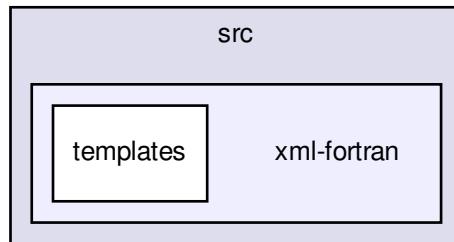


Files

- file [input_t.f90](#)

6.3 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/ Directory Reference

Directory dependency graph for /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/:



Directories

- directory [templates](#)

Files

- file [read_xml_primitives.f90](#)
- file [write_xml_primitives.f90](#)
- file [xmlparse.f90](#)
- file [xmlreader.f90](#)

7 Module Documentation

7.1 global Module Reference

Contains all of the global variables.

Functions/Subroutines

- subroutine [allocate_problem\(\)](#)
allocates global variables for calculation
- subroutine [deallocate_problem\(\)](#)
deallocates global variables
- subroutine [compute_macro_cross_sections\(\)](#)
routine that handles the call to compute macro cross sections
- subroutine [add_to_tallies\(\)](#)
routine that adds temporary value to tallies
- subroutine [bank_tallies\(\)](#)
routine that record temporary history information in tallies
- subroutine [finalize_tallies\(\)](#)
routine that calls another routine to compute tally statistics

Variables

- integer [VERSION_MAJOR](#) = 0
- integer [VERSION_MINOR](#) = 1
- integer [VERSION_RELEASE](#) = 1
- type(particle_type) [neut](#)

- type(material_type), dimension(:), allocatable mat
- type(tally_type), dimension(:), allocatable tal
- integer nhistories
- integer seed
- integer source_type
- integer eidx
- integer n_tallies
- integer n_materials
- integer res_iso
- real(8) Dancoff
- real(8) radius
- real(8) emin = 1e-11_8
- real(8) emax = 20.0_8
- real(8) kT = 8.6173324e-5_8*300*1.0e-6_8
- real(8) nubar = 2.455_8
- type(Timer) time_init
- type(Timer) time_run
- integer n_abs = 0.0_8
- integer n_fiss = 0.0_8
- real(8) ana_kinf_mean = 0.0_8
- real(8) ana_kinf_std = 0.0_8
- real(8) col_kinf_mean = 0.0_8
- real(8) col_kinf_std = 0.0_8

7.1.1 Detailed Description

Contains all of the global variables.

Author

Bryan Herman

7.1.2 Function/Subroutine Documentation

7.1.2.1 subroutine global::add_to_tallies()

routine that adds temporary value to tallies

Definition at line 145 of file global.f90.

References tally::add_to_tally(), eidx, mat, n_tallies, neut, nubar, and tal.

Referenced by physics::perform_physics().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.2.2 subroutine **global::allocate_problem()**

allocates global variables for calculation

Definition at line 71 of file global.f90.

References mat, n_materials, n_tallies, and tal.

7.1.2.3 subroutine **global::bank_tallies()**

routine that record temporary history information in tallies

Definition at line 213 of file global.f90.

References tally::bank_tally(), n_tallies, and tal.

Referenced by run_problem().

Here is the call graph for this function:



Here is the caller graph for this function:



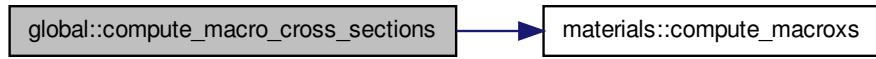
7.1.2.4 subroutine `global::compute_macro_cross_sections()`

routine that handles the call to compute macro cross sections

Definition at line 123 of file `global.f90`.

References `materials::compute_macros()`, `mat`, and `n_materials`.

Here is the call graph for this function:



7.1.2.5 subroutine **global::deallocate_problem()**

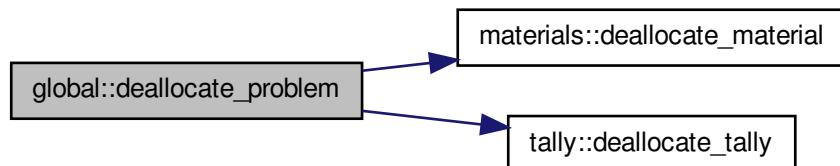
deallocates global variables

Definition at line 86 of file global.f90.

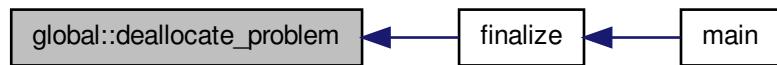
References materials::deallocate_material(), tally::deallocate_tally(), mat, n_materials, n_tallies, and tal.

Referenced by finalize().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.2.6 subroutine **global::finalize_tallies()**

routine that calls another routine to compute tally statistics

Definition at line 235 of file global.f90.

References ana_kinf_mean, ana_kinf_std, tally::calculate_statistics(), col_kinf_mean, col_kinf_std, mat, n_abs, n_fiss, n_materials, n_tallies, nhistories, nubar, and tal.

Referenced by finalize().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3 Variable Documentation

7.1.3.1 real(8) global::ana_kinf_mean = 0.0_8

Definition at line 59 of file global.f90.

Referenced by finalize_tallies(), and output::write_output().

7.1.3.2 real(8) global::ana_kinf_std = 0.0_8

Definition at line 60 of file global.f90.

Referenced by finalize_tallies().

7.1.3.3 real(8) global::col_kinf_mean = 0.0_8

Definition at line 61 of file global.f90.

Referenced by finalize_tallies().

7.1.3.4 real(8) global::col_kinf_std = 0.0_8

Definition at line 62 of file global.f90.

Referenced by finalize_tallies().

7.1.3.5 real(8) global::Dancoff

Definition at line 39 of file global.f90.

Referenced by physics::sample_region().

7.1.3.6 integer global::eidx

Definition at line 35 of file global.f90.

Referenced by add_to_tallies(), run_problem(), physics::sample_isotope(), physics::sample_reaction(), and physics::sample_region().

7.1.3.7 real(8) global::emax = 20.0_8

Definition at line 44 of file global.f90.

Referenced by input::read_input().

7.1.3.8 real(8) global::emin = 1e-11_8

Definition at line 43 of file global.f90.

Referenced by input::read_input(), and run_problem().

7.1.3.9 real(8) global::kT = 8.6173324e-5_8*300*1.0e-6_8

Definition at line 47 of file global.f90.

Referenced by physics::elastic_scattering().

7.1.3.10 type(material_type),dimension(:),allocatable global::mat

Definition at line 26 of file global.f90.

Referenced by add_to_tallies(), allocate_problem(), compute_macro_cross_sections(), deallocate_problem(), physics::elastic_scattering(), finalize_tallies(), physics::get_eidx(), input::read_input(), physics::sample_isotope(), physics::sample_reaction(), physics::sample_region(), and physics::sample_source().

7.1.3.11 integer global::n_abs = 0.0_8

Definition at line 57 of file global.f90.

Referenced by finalize_tallies(), and physics::perform_physics().

7.1.3.12 integer global::n_fiss = 0.0_8

Definition at line 58 of file global.f90.

Referenced by finalize_tallies(), and physics::perform_physics().

7.1.3.13 integer global::n_materials

Definition at line 37 of file global.f90.

Referenced by allocate_problem(), compute_macro_cross_sections(), deallocate_problem(), finalize_tallies(), and physics::sample_region().

7.1.3.14 integer global::n_tallies

Definition at line 36 of file global.f90.

Referenced by add_to_tallies(), allocate_problem(), bank_tallies(), deallocate_problem(), finalize_tallies(), and output::write_output().

7.1.3.15 type(particle_type) global::neut

Definition at line 25 of file global.f90.

Referenced by add_to_tallies(), physics::elastic_scattering(), physics::perform_physics(), run_problem(), physics::sample_region(), and physics::sample_source().

7.1.3.16 integer global::nhistories

Definition at line 30 of file global.f90.

Referenced by finalize_tallies(), input::read_input(), and run_problem().

7.1.3.17 real(8) global::nubar = 2.455_8

Definition at line 50 of file global.f90.

Referenced by add_to_tallies(), and finalize_tallies().

7.1.3.18 real(8) global::radius

Definition at line 40 of file global.f90.

Referenced by physics::sample_region().

7.1.3.19 integer global::res_iso

Definition at line 38 of file global.f90.

7.1.3.20 integer global::seed

Definition at line 31 of file global.f90.

Referenced by initialize(), and input::read_input().

7.1.3.21 integer global::source_type

Definition at line 32 of file global.f90.

Referenced by input::read_input().

7.1.3.22 type(tally_type),dimension(:),allocatable global::tal

Definition at line 27 of file global.f90.

Referenced by add_to_tallies(), allocate_problem(), bank_tallies(), deallocate_problem(), finalize_tallies(), and output::write_output().

7.1.3.23 type(Timer) global::time_init

Definition at line 53 of file global.f90.

Referenced by initialize(), and output::write_output().

7.1.3.24 type(Timer) global::time_run

Definition at line 54 of file global.f90.

Referenced by run_problem(), and output::write_output().

7.1.3.25 integer global::VERSION_MAJOR = 0

Definition at line 20 of file global.f90.

Referenced by output::print_heading().

7.1.3.26 integer global::VERSION_MINOR = 1

Definition at line 21 of file global.f90.

Referenced by output::print_heading().

7.1.3.27 integer global::VERSION_RELEASE = 1

Definition at line 22 of file global.f90.

Referenced by output::print_heading().

7.2 input Module Reference

Handles reading in the input xml file and initializing global vars.

Functions/Subroutines

- subroutine, public `read_input`

Reads the input xml file and sets global variables.

7.2.1 Detailed Description

Handles reading in the input xml file and initializing global vars.

Author

Bryan Herman

7.2.2 Function/Subroutine Documentation

7.2.2.1 subroutine,public input::read_input()

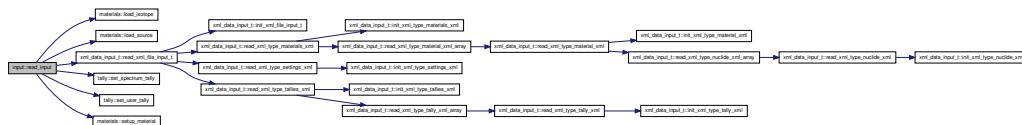
Reads the input xml file and sets global variables.

Definition at line 22 of file input.f90.

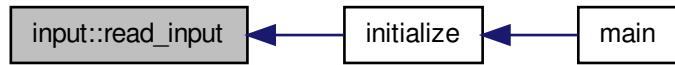
References global::emax, global::emin, materials::load_isotope(), materials::load_source(), global::mat, xml_data_input_t::materials_, global::nhistories, xml_data_input_t::read_xml_file_input_t(), global::seed, tally::set_spectrum_tally(), tally::set_user_tally(), xml_data_input_t::settings_, materials::setup_material(), global::source_type, and xml_data_input_t::tallies_.

Referenced by initialize().

Here is the call graph for this function:



Here is the caller graph for this function:



7.3 materials Module Reference

Contains information about the isotopics of problem.

Data Types

- type **source_type**
- type **thermal_type**
- type **iso_type**
- type **material_type**

Functions/Subroutines

- subroutine, public **setup_material** (this, emin, emax, nisotopes, vol)
routine that initializes the materials
- subroutine, public **load_isotope** (this, N, A, path, thermal, name)
routine that loads isotope properties, xs, etc.
- subroutine, public **load_source** (this, source_type, source_path)
routine to load fission source into memory
- subroutine, public **compute_macros** (this)
routine to pre-compute macroscopic cross sections
- subroutine, public **deallocate_material** (this)
routine to deallocate a material

7.3.1 Detailed Description

Contains information about the isotopics of problem.

Author

Bryan Herman

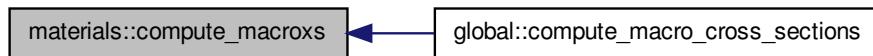
7.3.2 Function/Subroutine Documentation**7.3.2.1 subroutine,public materials::compute_macroxs (type(material_type),target *this*)**

routine to pre-compute macroscopic cross sections

Definition at line 294 of file materials.f90.

Referenced by global::compute_macro_cross_sections().

Here is the caller graph for this function:

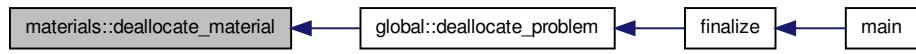
**7.3.2.2 subroutine,public materials::deallocate_material (type(material_type) *this*)**

routine to deallocate a material

Definition at line 341 of file materials.f90.

Referenced by global::deallocate_problem().

Here is the caller graph for this function:



7.3.2.3 subroutine,public materials::load_isotope (type(material_type) *this*, real(8) *N*, real(8) *A*, character(len=255) *path*, logical *thermal*, character(len=255) *name*)

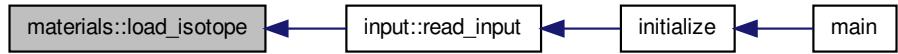
routine that loads isotope properties, xs, etc.

into memory

Definition at line 106 of file materials.f90.

Referenced by input::read_input().

Here is the caller graph for this function:



7.3.2.4 subroutine,public materials::load_source (type(material_type) *this*, integer *source_type*, character(len=255) *source_path*)

routine to load fission source into memory

Definition at line 237 of file materials.f90.

Referenced by input::read_input().

Here is the caller graph for this function:



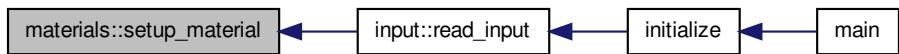
7.3.2.5 subroutine,public materials::setup_material (type(material_type) *this*, real(8) *emin*, real(8) *emax*, integer *nisotopes*, real(8) *vol*)

routine that initializes the materials

Definition at line 74 of file materials.f90.

Referenced by input::read_input().

Here is the caller graph for this function:



7.4 output Module Reference

Contains routines for outputting major info to user.

Functions/Subroutines

- subroutine, public [print_heading \(\)](#)
prints the code heading and run information
- subroutine, public [write_output \(\)](#)
routine that writes timing info and hdf5 file
- subroutine [get_today](#) (today_date, today_time)
calculates information about date/time of run

7.4.1 Detailed Description

Contains routines for outputting major info to user.

Author

Bryan Herman

7.4.2 Function/Subroutine Documentation

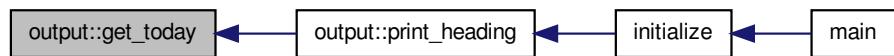
7.4.2.1 subroutine [output::get_today](#) (character(10),intent(out) today_date, character(8),intent(out) today_time)

calculates information about date/time of run

Definition at line 160 of file output.f90.

Referenced by print_heading().

Here is the caller graph for this function:



7.4.2.2 subroutine,public output::print_heading()

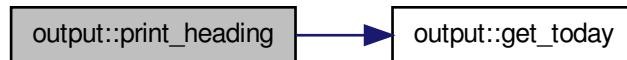
prints the code heading and run information

Definition at line 22 of file output.f90.

References get_today(), global::VERSION_MAJOR, global::VERSION_MINOR, and global::VERSION_RELEASE.

Referenced by initialize().

Here is the call graph for this function:



Here is the caller graph for this function:



7.4.2.3 subroutine,public output::write_output()

routine that writes timing info and hdf5 file

Definition at line 64 of file output.f90.

References global::ana_kinf_mean, global::n_tallies, global::tal, global::time_init, and global::time_run.

Referenced by finalize().

Here is the caller graph for this function:



7.5 particle Module Reference

Contains information about the particle that is transporting.

Data Types

- type **particle_type**

Functions/Subroutines

- subroutine, public **init_particle** (this)

routine to initialize a particle

7.5.1 Detailed Description

Contains information about the particle that is transporting.

Author

Bryan Herman

7.5.2 Function/Subroutine Documentation

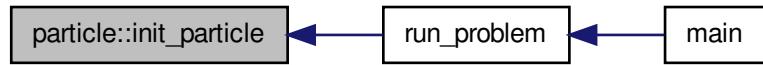
7.5.2.1 subroutine,public particle::init_particle (type(particle_type) this)

routine to initialize a particle

Definition at line 32 of file particle.f90.

Referenced by run_problem().

Here is the caller graph for this function:



7.6 physics Module Reference

Contains routines to model the physics of the problem.

Functions/Subroutines

- subroutine, public [sample_source \(\)](#)
routine to sample source from cdf
- subroutine, public [perform_physics \(\)](#)
high level routine to perform transport physics
- integer, public [get_eidx \(E\)](#)

function to compute the index in unionized energy grid

- **integer sample_region ()**
function to sample region where interaction occurs
- **integer sample_isotope (region)**
function to sample interaction isotope
- **integer sample_reaction (region, isoidx)**
function to sample reaction type
- **subroutine elastic_scattering (region, isoidx)**
routine to perform thermal/asymptotic elastic scattering physics

7.6.1 Detailed Description

Contains routines to model the physics of the problem.

Author

Bryan Herman

7.6.2 Function/Subroutine Documentation

7.6.2.1 subroutine physics::elastic_scattering (**integer region, integer isoidx**)

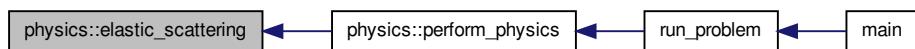
routine to perform thermal/asymptotic elastic scattering physics

Definition at line 303 of file physics.f90.

References global::kT, global::mat, and global::neut.

Referenced by perform_physics().

Here is the caller graph for this function:



7.6.2.2 integer,public physics::get_eidx (real(8) *E*)

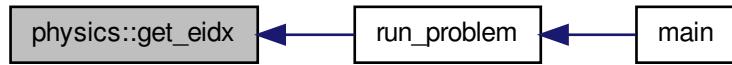
function to compute the index in unionized energy grid

Definition at line 89 of file physics.f90.

References global::mat.

Referenced by run_problem().

Here is the caller graph for this function:

**7.6.2.3 subroutine,public physics::perform_physics ()**

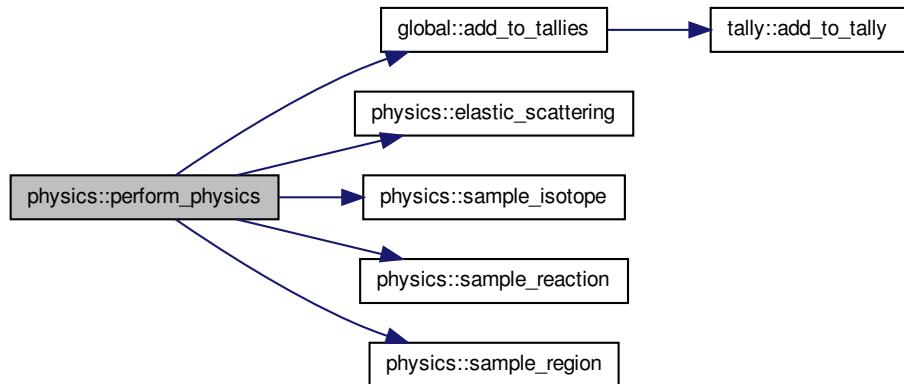
high level routine to perform transport physics

Definition at line 54 of file physics.f90.

References global::add_to_tallies(), elastic_scattering(), global::n_abs, global::n_fiss, global::neut, sample_isotope(), sample_reaction(), and sample_region().

Referenced by run_problem().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.2.4 integer physics::sample_isotope (integer *region*)

function to sample interaction isotope

Definition at line 199 of file physics.f90.

References `global::eidx`, and `global::mat`.

Referenced by `perform_physics()`.

Here is the caller graph for this function:



7.6.2.5 integer physics::sample_reaction (integer *region*, integer *isoidx*)

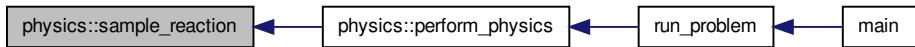
function to sample reaction type

Definition at line 257 of file physics.f90.

References global::eidx, and global::mat.

Referenced by perform_physics().

Here is the caller graph for this function:



7.6.2.6 integer physics::sample_region ()

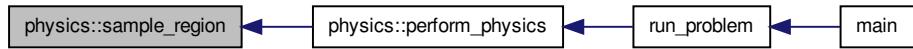
function to sample region where interaction occurs

Definition at line 115 of file physics.f90.

References global::Dancoff, global::eidx, global::mat, global::n_materials, global::neut, and global::radius.

Referenced by perform_physics().

Here is the caller graph for this function:



7.6.2.7 subroutine,public physics::sample_source()

routine to sample source from cdf

Definition at line 22 of file physics.f90.

References global::mat, and global::neut.

Referenced by run_problem().

Here is the caller graph for this function:



7.7 read_xml_primitives Module Reference

Data Types

- interface [read_from_buffer](#)

Functions/Subroutines

- subroutine [skip_until_endtag](#) (info, tag, attrs, data, error)
- subroutine [read_xml_integer](#) (info, tag, endtag, attrs, noattrs, data, nodata, var, has_var)
- subroutine [read_xml_line](#) (info, tag, endtag, attrs, noattrs, data, nodata, var, has_var)

- subroutine `read_xml_real` (info, tag, endtag, attrs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_double` (info, tag, endtag, attrs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_logical` (info, tag, endtag, attrs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_word` (info, tag, endtag, attrs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_integer_array` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_line_array` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_real_array` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_double_array` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_logical_array` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_word_array` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_integer_1dim` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_real_1dim` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_double_1dim` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_logical_1dim` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_word_1dim` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)
- subroutine `read_xml_line_1dim` (info, tag, endtag, attrs, noattribs, data, no-data, var, has_var)

7.7.1 Function/Subroutine Documentation

**7.7.1.1 subroutine `read_xml_primitives::read_xml_double` (info
, tag, endtag, attrs, noattribs, data, nodata,
real(kind=kind(1.0d00)),intent(inout) var, has_var)**

Definition at line 160 of file `read_xml_primitives.f90`.

Referenced by `read_xml_double_1dim()`.

Here is the caller graph for this function:



7.7.1.2 subroutine `read_xml_primitives::read_xml_double_1dim` (`type(XML_PARSE),intent(inout) info,`
`character(len=*),intent(in) tag, logical,intent(inout) endtag,`
`character(len=*),dimension(:, :, intent(in) attribs, integer,intent(in) noattribs, character(len=*),dimension(:, :, intent(in) data,`
`integer,intent(in) nodata, real(kind=kind(1.0d00)),dimension(:),pointer var, logical,intent(inout) has_var)`

Definition at line 414 of file `read_xml_primitives.f90`.

References `read_xml_double()`.

Here is the call graph for this function:



7.7.1.3 subroutine `read_xml_primitives::read_xml_double_array` (`info , tag , endtag , attribs , noattribs , data , nodata ,`
`real(kind=kind(1.0d00)),dimension(:,pointer var, has_var)`

Definition at line 280 of file `read_xml_primitives.f90`.

7.7.1.4 subroutine `read_xml_primitives::read_xml_integer` (`info , tag , endtag ,`
`attribs , noattribs , data , nodata , integer,intent(inout) var, has_var)`

Definition at line 91 of file read_xml_primitives.f90.

Referenced by read_xml_integer_1dim().

Here is the caller graph for this function:



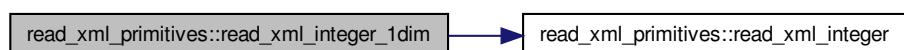
7.7.1.5 subroutine read_xml_primitives::read_xml_integer_1dim

```
1dim ( type(XML_PARSE),intent(inout) info,
character(len=*),intent(in) tag, logical,intent(inout) endtag,
character(len=*),dimension(:, :, intent(in) attribs, integer,intent(in)
noattribs, character(len=*),dimension(:, intent(in) data,
integer,intent(in) nodata, integer,dimension(:,pointer var,
logical,intent(inout) has_var )
```

Definition at line 358 of file read_xml_primitives.f90.

References read_xml_integer().

Here is the call graph for this function:



7.7.1.6 subroutine read_xml_primitives::read_xml_integer_array

```
(info , tag , endtag , attribs , noattribs , data , nodata ,
integer,dimension(:,pointer var, has_var )
```

Definition at line 199 of file read_xml_primitives.f90.

```
7.7.1.7 subroutine read_xml_primitives::read_xml_line ( type(XML_PARSE),intent(inout) info,
character(len=*),intent(in) tag, logical,intent(inout) endtag,
character(len=*),dimension(:, :, intent(in) attribs, integer,intent(in)
noattribs, character(len=*),dimension(:, :, intent(in) data,
integer,intent(in) nodata, character(len=*),intent(inout) var,
logical,intent(inout) has_var )
```

Definition at line 113 of file read_xml_primitives.f90.

References xmlparse::xml_find_attrib().

Referenced by read_xml_line_1dim().

Here is the call graph for this function:



Here is the caller graph for this function:

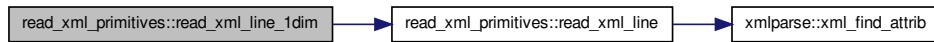


```
7.7.1.8 subroutine read_xml_primitives::read_xml_line_1dim ( type(XML_PARSE),intent(inout) info,
character(len=*),intent(in) tag, logical,intent(inout) endtag,
character(len=*),dimension(:, :, intent(in) attribs, integer,intent(in)
noattribs, character(len=*),dimension(:, :, intent(in) data,
integer,intent(in) nodata, character(len=*),dimension(:, :, pointer var,
logical,intent(inout) has_var )
```

Definition at line 498 of file read_xml_primitives.f90.

References `read_xml_line()`.

Here is the call graph for this function:

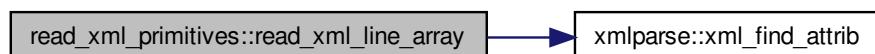


7.7.1.9 subroutine `read_xml_primitives::read_xml_line_array` (`type(XML_PARSE),intent(inout) info,`
`character(len=*),intent(in) tag, logical,intent(inout) endtag,`
`character(len=*),dimension(:, :, :),intent(in) attribs, integer,intent(in)`
`noattribs, character(len=*),dimension(:, :),intent(in) data,`
`integer,intent(in) nodata, character(len=*),dimension(:, :),pointer var,`
`logical,intent(inout) has_var`)

Definition at line 220 of file `read_xml_primitives.f90`.

References `xmlparse::xml_find_attrib()`.

Here is the call graph for this function:



7.7.1.10 subroutine `read_xml_primitives::read_xml_logical` (`info, tag,`
`endtag, attribs, noattribs, data, nodata, logical,intent(inout) var,`
`has_var`)

Definition at line 168 of file `read_xml_primitives.f90`.

Referenced by `read_xml_logical_1dim()`.

Here is the caller graph for this function:



```
7.7.1.11 subroutine read_xml_primitives::read_xml_logical_-
1dim ( type(XML_PARSE),intent(inout) info,
character(len=*),intent(in) tag, logical,intent(inout) endtag,
character(len=*),dimension(:,:),intent(in) attribs, integer,intent(in)
noattribs, character(len=*),dimension(:,),intent(in) data,
integer,intent(in) nodata, logical,dimension(:,pointer var,
logical,intent(inout) has_var )
```

Definition at line 442 of file read_xml_primitives.f90.

References read_xml_logical().

Here is the call graph for this function:



```
7.7.1.12 subroutine read_xml_primitives::read_xml_logical_array (
info , tag , endtag , attribs , noattribs , data , nodata ,
logical,dimension(:,pointer var, has_var )
```

Definition at line 288 of file read_xml_primitives.f90.

```
7.7.1.13 subroutine read_xml_primitives::read_xml_real ( info , tag , endtag ,
attribs , noattribs , data , nodata , real,intent(inout) var, has_var )
```

Definition at line 152 of file read_xml_primitives.f90.

Referenced by read_xml_real_1dim().

Here is the caller graph for this function:



7.7.1.14 subroutine read_xml_primitives::read_xml_real_1dim (*type(XML_PARSE),intent(inout) info,*
character(len=),intent(in) tag, logical,intent(inout) endtag,*
character(len=),dimension(:, :, :),intent(in) attribs, integer,intent(in)*
noattribs, character(len=),dimension(:, :),intent(in) data,*
integer,intent(in) nodata, real,dimension(:, :),pointer var,
logical,intent(inout) has_var)

Definition at line 386 of file read_xml_primitives.f90.

References read_xml_real().

Here is the call graph for this function:



7.7.1.15 subroutine read_xml_primitives::read_xml_real_array (*info, tag,*
endtag, attribs, noattribs, data, nodata, real,dimension(:, :),pointer
var, has_var)

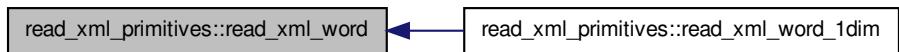
Definition at line 272 of file read_xml_primitives.f90.

```
7.7.1.16 subroutine read_xml_primitives::read_xml_word( info, tag, endtag
, attribs, noattribs, data, nodata, character(len=*),intent(inout)
var, has_var )
```

Definition at line 176 of file read_xml_primitives.f90.

Referenced by read_xml_word_1dim().

Here is the caller graph for this function:

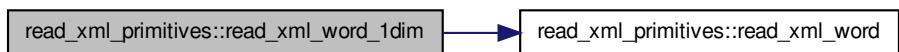


```
7.7.1.17 subroutine read_xml_primitives::read_xml_word_-
1dim( type(XML_PARSE),intent(inout) info,
character(len=*),intent(in) tag, logical,intent(inout) endtag,
character(len=*),dimension(:, :),intent(in) attribs, integer,intent(in)
noattribs, character(len=*),dimension(:, :),intent(in) data,
integer,intent(in) nodata, character(len=*),dimension(:, :),pointer var,
logical,intent(inout) has_var )
```

Definition at line 470 of file read_xml_primitives.f90.

References read_xml_word().

Here is the call graph for this function:



7.7.1.18 subroutine `read_xml_primitives::read_xml_word_array` (
`info, tag, endtag, attribs, noattribs, data, nodata,`
`character(len=*),dimension(:),pointer var, has_var)`

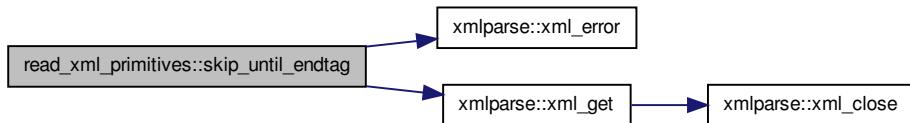
Definition at line 296 of file `read_xml_primitives.f90`.

7.7.1.19 subroutine `read_xml_primitives::skip_until_endtag` (
`type(XML_PARSE),intent(inout) info, character(len=*),intent(in)`
`tag, character(len=*),dimension(:, :),intent(inout) attribs,`
`character(len=*),dimension(:, :),intent(inout) data, logical,intent(out)`
`error)`

Definition at line 50 of file `read_xml_primitives.f90`.

References `xmlparse::xml_error()`, and `xmlparse::xml_get()`.

Here is the call graph for this function:



7.8 tally Module Reference

Contains information about tallying quantities.

Data Types

- type `tally_type`

Functions/Subroutines

- subroutine, public `set_user_tally` (this, Ebins, n, react_type, isotope, region, n_materials, dv)
routine to initialize user-defined tallies
- subroutine, public `set_spectrum_tally` (this, emax, emin, n_materials)
routine to initialize all tallies

- subroutine, public `set_kinf_tally` (this, emax, emin, n_materials)
routine to initialize kinf nu-fission tally
- subroutine, public `add_to_tally` (this, fact, totxs, E, region)
routine to add quantities during transport of a particle
- subroutine, public `bank_tally` (this)
routine to bank a histories tallies
- subroutine, public `calculate_statistics` (this, n)
routine to compute mean and standard deviation of tallies
- subroutine, public `deallocate_tally` (this)
routine to deallocate tally types

7.8.1 Detailed Description

Contains information about tallying quantities.

Author

Bryan Herman

7.8.2 Function/Subroutine Documentation

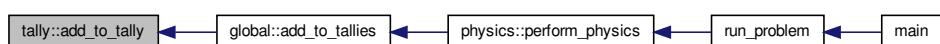
7.8.2.1 subroutine,public tally::add_to_tally (type(tally_type) this, real(8) fact, real(8) totxs, real(8) E, integer region)

routine to add quantities during transport of a particle

Definition at line 170 of file tally.f90.

Referenced by `global::add_to_tallies()`.

Here is the caller graph for this function:



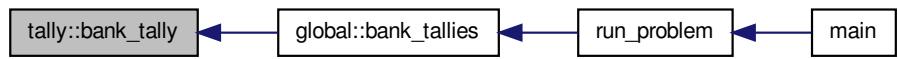
7.8.2.2 subroutine,public tally::bank_tally (type(tally_type) *this*)

routine to bank a histories tallies

Definition at line 214 of file tally.f90.

Referenced by global::bank_tallies().

Here is the caller graph for this function:

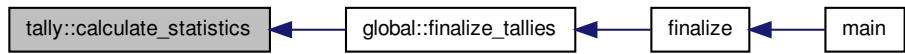
**7.8.2.3 subroutine,public tally::calculate_statistics (type(tally_type) *this*, integer *n*)**

routine to compute mean and standard deviation of tallies

Definition at line 233 of file tally.f90.

Referenced by global::finalize_tallies().

Here is the caller graph for this function:

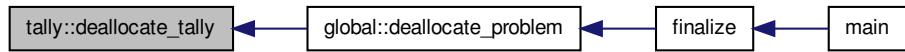
**7.8.2.4 subroutine,public tally::deallocate_tally (type(tally_type) *this*)**

routine to deallocate tally types

Definition at line 252 of file tally.f90.

Referenced by global::deallocate_problem().

Here is the caller graph for this function:



7.8.2.5 subroutine,public tally::set_kinf_tally (type(tally_type) this, real(8) emax, real(8) emin, integer n_materials)

routine to initiaize kinf nu-fission tally

Definition at line 131 of file tally.f90.

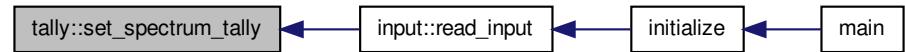
7.8.2.6 subroutine,public tally::set_spectrum_tally (type(tally_type) this, real(8) emax, real(8) emin, integer n_materials)

routine to initialize all tallies

Definition at line 95 of file tally.f90.

Referenced by input::read_input().

Here is the caller graph for this function:



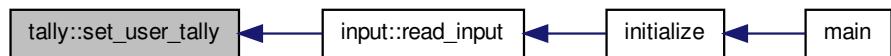
7.8.2.7 subroutine,public tally::set_user_tally (type(tally_type) this, real(8),dimension(n) Ebins, integer n, integer react_type, integer isotope, integer region, integer n_materials, logical dv)

routine to intialize user-defined tallies

Definition at line 43 of file tally.f90.

Referenced by `input::read_input()`.

Here is the caller graph for this function:



7.9 timing Module Reference

Data Types

- type `Timer`

Functions/Subroutines

- subroutine `timer_start` (`self`)
- real(8) `timer_get_value` (`self`)
- subroutine `timer_stop` (`self`)
- subroutine `timer_reset` (`self`)

Variables

- real(8) `ZERO` = 1.0_8

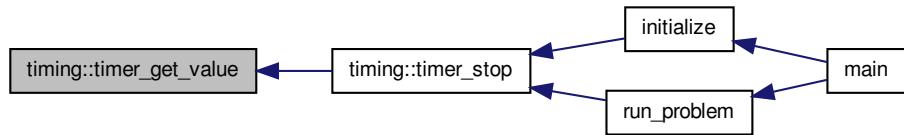
7.9.1 Function/Subroutine Documentation

7.9.1.1 real(8) `timing::timer_get_value` (`type(Timer),intent(in) self`)

Definition at line 44 of file timing.f90.

Referenced by `timer_stop()`.

Here is the caller graph for this function:



7.9.1.2 subroutine `timing::timer_reset (type(Timer),intent(inout) self)`

Definition at line 84 of file `timing.f90`.

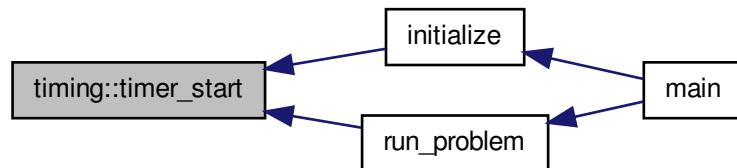
References ZERO.

7.9.1.3 subroutine `timing::timer_start (type(Timer),intent(inout) self)`

Definition at line 30 of file `timing.f90`.

Referenced by `initialize()`, and `run_problem()`.

Here is the caller graph for this function:



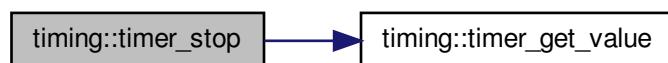
7.9.1.4 subroutine `timing::timer_stop (type(Timer),intent(inout) self)`

Definition at line 67 of file timing.f90.

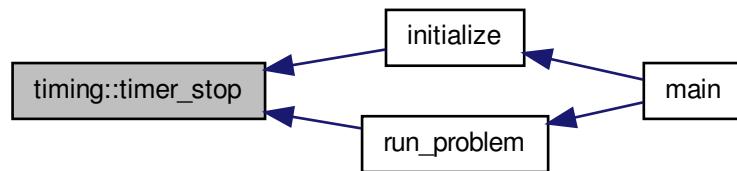
References timer_get_value().

Referenced by initialize(), and run_problem().

Here is the call graph for this function:



Here is the caller graph for this function:



7.9.2 Variable Documentation

7.9.2.1 real(8) timing::ZERO = 1.0_8

Definition at line 22 of file timing.f90.

Referenced by timer_reset().

7.10 write_xml_primitives Module Reference

Data Types

- interface [write_to_xml_word](#)
- interface [write_to_xml_line](#)

Functions/Subroutines

- subroutine [write_to_xml_integer](#) (info, tag, indent, value)
- subroutine [write_to_xml_integer_1dim](#) (info, tag, indent, values)
- subroutine [write_to_xml_real](#) (info, tag, indent, value)
- subroutine [write_to_xml_real_1dim](#) (info, tag, indent, values)
- subroutine [write_to_xml_double](#) (info, tag, indent, value)
- subroutine [write_to_xml_double_1dim](#) (info, tag, indent, values)
- subroutine [write_to_xml_string](#) (info, tag, indent, value)
- subroutine [write_to_xml_word_1dim](#) (info, tag, indent, values)
- subroutine [write_to_xml_string_1dim](#) (info, tag, indent, values)
- subroutine [write_to_xml_logical](#) (info, tag, indent, value)
- subroutine [write_to_xml_logical_1dim](#) (info, tag, indent, values)
- subroutine [write_to_xml_integer_array](#) (info, tag, indent, array)
- subroutine [write_to_xml_real_array](#) (info, tag, indent, array)
- subroutine [write_to_xml_double_array](#) (info, tag, indent, array)
- subroutine [write_to_xml_logical_array](#) (info, tag, indent, array)
- subroutine [write_to_xml_word_array](#) (info, tag, indent, array)
- subroutine [write_to_xml_line_array](#) (info, tag, indent, array)

7.10.1 Function/Subroutine Documentation

7.10.1.1 subroutine write_xml_primitives::write_to_xml_double (
`type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,`
`integer,intent(in) indent, real(kind=kind(1.0d0)),intent(in) value)`

Definition at line 136 of file write_xml_primitives.f90.

Referenced by [write_to_xml_double_1dim\(\)](#).

Here is the caller graph for this function:



7.10.1.2 subroutine write_xml_primitives::write_to_xml_-
`double_1dim (type(XML_PARSE),intent(in) info,`
`character(len=*),intent(in) tag, integer,intent(in) indent,`
`real(kind=kind(1.0d00)),dimension(:),intent(in) values)`

Definition at line 161 of file write_xml_primitives.f90.

References write_to_xml_double().

Here is the call graph for this function:



7.10.1.3 subroutine write_xml_primitives::write_to_xml_double_array (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, real(kind=kind(1.0d0)),dimension(:,intent(in) array))

Definition at line 371 of file write_xml_primitives.f90.

7.10.1.4 subroutine write_xml_primitives::write_to_xml_integer (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, integer,intent(in) value)

Definition at line 42 of file write_xml_primitives.f90.

Referenced by write_to_xml_integer_1dim().

Here is the caller graph for this function:



7.10.1.5 subroutine write_xml_primitives::write_to_xml_integer_1dim (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, integer,dimension(:,intent(in) values))

Definition at line 65 of file write_xml_primitives.f90.

References write_to_xml_integer().

Here is the call graph for this function:



7.10.1.6 subroutine write_xml_primitives::write_to_xml_integer_array (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, integer,dimension(:,),intent(in) array)

Definition at line 307 of file write_xml_primitives.f90.

7.10.1.7 subroutine write_xml_primitives::write_to_xml_line_array (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, logical,dimension(:,),intent(in) array)

Definition at line 467 of file write_xml_primitives.f90.

7.10.1.8 subroutine write_xml_primitives::write_to_xml_logical (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, logical,intent(in) value)

Definition at line 256 of file write_xml_primitives.f90.

Referenced by write_to_xml_logical_1dim().

Here is the caller graph for this function:

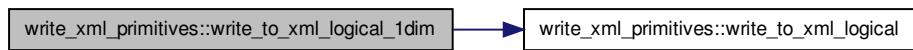


7.10.1.9 subroutine write_xml_primitives::write_to_xml_logical_1dim (
 type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,
 integer,intent(in) indent, logical,dimension(:),intent(in) values)

Definition at line 284 of file write_xml_primitives.f90.

References write_to_xml_logical().

Here is the call graph for this function:



7.10.1.10 subroutine write_xml_primitives::write_to_xml_logical_array (
 type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,
 integer,intent(in) indent, logical,dimension(:),intent(in) array)

Definition at line 403 of file write_xml_primitives.f90.

7.10.1.11 subroutine write_xml_primitives::write_to_xml_real (
 type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,
 integer,intent(in) indent, real,intent(in) value)

Definition at line 88 of file write_xml_primitives.f90.

Referenced by write_to_xml_real_1dim().

Here is the caller graph for this function:



7.10.1.12 subroutine write_xml_primitives::write_to_xml_real_1dim (
 type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,
 integer,intent(in) indent, real,dimension(:),intent(in) values)

Definition at line 113 of file write_xml_primitives.f90.

References write_to_xml_real().

Here is the call graph for this function:



7.10.1.13 subroutine write_xml_primitives::write_to_xml_real_array (
 type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,
 integer,intent(in) indent, real,dimension(:),intent(in) array)

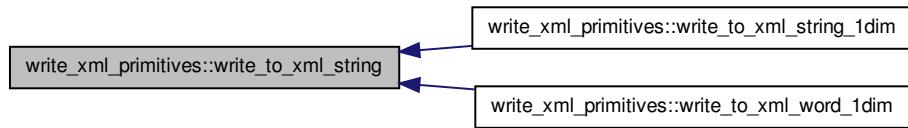
Definition at line 339 of file write_xml_primitives.f90.

7.10.1.14 subroutine write_xml_primitives::write_to_xml_string (
 type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,
 integer,intent(in) indent, character(len=*),intent(in) value)

Definition at line 184 of file write_xml_primitives.f90.

Referenced by write_to_xml_string_1dim(), and write_to_xml_word_1dim().

Here is the caller graph for this function:



```
7.10.1.15 subroutine write_xml_primitives::write_to_xml_string_1dim (  
    type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,  
    integer,intent(in) indent, character(len=*),dimension(:,),intent(in)  
    values )
```

Definition at line 233 of file write_xml_primitives.f90.

References write_to_xml_string().

Here is the call graph for this function:



```
7.10.1.16 subroutine write_xml_primitives::write_to_xml_word_1dim (   
    type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,  
    integer,intent(in) indent, character(len=*),dimension(:,),intent(in)  
    values )
```

Definition at line 211 of file write_xml_primitives.f90.

References write_to_xml_string().

Here is the call graph for this function:



```
7.10.1.17 subroutine write_xml_primitives::write_to_xml_word_array (   
    type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag,  
    integer,intent(in) indent, character(len=*),dimension(:,),intent(in)  
    array )
```

Definition at line 435 of file write_xml_primitives.f90.

7.11 xml_data_input_t Module Reference

Data Types

- type settings_xml
- type nuclide_xml
- type material_xml
- type materials_xml
- type tally_xml
- type tallies_xml

Functions/Subroutines

- subroutine `read_xml_type_settings_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `read_xml_type_settings_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `init_xml_type_settings_xml_array` (dvar)
- subroutine `init_xml_type_settings_xml` (dvar)
- subroutine `write_xml_type_settings_xml_array` (info, tag, indent, dvar)
- subroutine `write_xml_type_settings_xml` (info, tag, indent, dvar)
- subroutine `read_xml_type_nuclide_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `read_xml_type_nuclide_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `init_xml_type_nuclide_xml_array` (dvar)
- subroutine `init_xml_type_nuclide_xml` (dvar)
- subroutine `write_xml_type_nuclide_xml_array` (info, tag, indent, dvar)
- subroutine `write_xml_type_nuclide_xml` (info, tag, indent, dvar)
- subroutine `read_xml_type_material_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `read_xml_type_material_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `init_xml_type_material_xml_array` (dvar)
- subroutine `init_xml_type_material_xml` (dvar)
- subroutine `write_xml_type_material_xml_array` (info, tag, indent, dvar)
- subroutine `write_xml_type_material_xml` (info, tag, indent, dvar)
- subroutine `read_xml_type_materials_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `read_xml_type_materials_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `init_xml_type_materials_xml_array` (dvar)
- subroutine `init_xml_type_materials_xml` (dvar)
- subroutine `write_xml_type_materials_xml_array` (info, tag, indent, dvar)

- subroutine `write_xml_type_materials_xml` (info, tag, indent, dvar)
- subroutine `read_xml_type_tally_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `read_xml_type_tally_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `init_xml_type_tally_xml_array` (dvar)
- subroutine `init_xml_type_tally_xml` (dvar)
- subroutine `write_xml_type_tally_xml_array` (info, tag, indent, dvar)
- subroutine `write_xml_type_tally_xml` (info, tag, indent, dvar)
- subroutine `read_xml_type_tallies_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `read_xml_type_tallies_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `init_xml_type_tallies_xml_array` (dvar)
- subroutine `init_xml_type_tallies_xml` (dvar)
- subroutine `write_xml_type_tallies_xml_array` (info, tag, indent, dvar)
- subroutine `write_xml_type_tallies_xml` (info, tag, indent, dvar)
- subroutine `read_xml_file_input_t` (fname, lurep, errout)
- subroutine `write_xml_file_input_t` (fname, lurep)
- subroutine `init_xml_file_input_t`

Variables

- integer, private `lurep_`
- logical, private `strict_`
- type(`settings_xml`) `settings_`
- type(`materials_xml`) `materials_`
- type(`tallies_xml`) `tallies_`

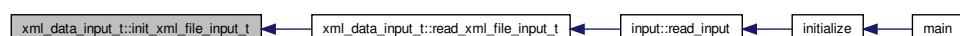
7.11.1 Function/Subroutine Documentation

7.11.1.1 subroutine `xml_data_input_t::init_xml_file_input_t()`

Definition at line 1269 of file `input_t.f90`.

Referenced by `read_xml_file_input_t()`.

Here is the caller graph for this function:



7.11.1.2 subroutine `xml_data_input_t::init_xml_type_material_xml (type(material_xml) dvar)`

Definition at line 619 of file input_t.f90.

Referenced by `read_xml_type_material_xml()`.

Here is the caller graph for this function:



7.11.1.3 subroutine `xml_data_input_t::init_xml_type_material_xml_array (type(material_xml),dimension(:),pointer dvar)`

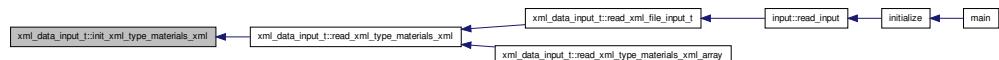
Definition at line 612 of file input_t.f90.

7.11.1.4 subroutine `xml_data_input_t::init_xml_type_materials_xml (type(materials_xml) dvar)`

Definition at line 773 of file input_t.f90.

Referenced by `read_xml_type_materials_xml()`.

Here is the caller graph for this function:



7.11.1.5 subroutine `xml_data_input_t::init_xml_type_materials_xml_array (type(materials_xml),dimension(:),pointer dvar)`

Definition at line 766 of file input_t.f90.

7.11.1.6 subroutine `xml_data_input_t::init_xml_type_nuclide_xml` (type(`nuclide_xml`) `dvar`)

Definition at line 442 of file input_t.f90.

Referenced by `read_xml_type_nuclide_xml()`.

Here is the caller graph for this function:



7.11.1.7 subroutine xml_data_input_t::init_xml_type_nuclide_xml_array (
type(nuclide_xml),dimension(:),pointer dvar)

Definition at line 435 of file input_t.f90.

7.11.1.8 subroutine xml_data_input_t::init_xml_type_settings_xml (
type(settings_xml) dvar)

Definition at line 235 of file input_t.f90.

Referenced by `read_xml_type_settings_xml()`.

Here is the caller graph for this function:



7.11.1.9 subroutine xml_data_input_t::init_xml_type_settings_xml_array (
type(settings_xml),dimension(:),pointer dvar)

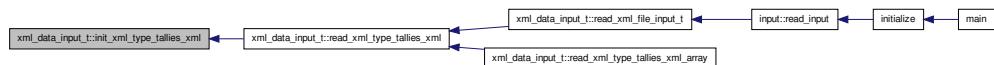
Definition at line 228 of file input_t.f90.

7.11.1.10 subroutine `xml_data_input_t::init_xml_type_tallies_xml` (
`type(tallies_xml) dvar` **)**

Definition at line 1117 of file input_t.f90.

Referenced by `read_xml_type_tallies_xml()`.

Here is the caller graph for this function:



7.11.1.11 subroutine `xml_data_input_t::init_xml_type_tallies_xml_array` (
`type(tallies_xml),dimension(:),pointer dvar` **)**

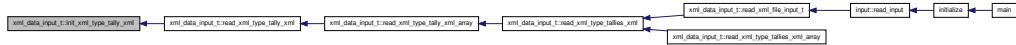
Definition at line 1110 of file input_t.f90.

7.11.1.12 subroutine `xml_data_input_t::init_xml_type_tally_xml` (
`type(tally_xml) dvar` **)**

Definition at line 960 of file input_t.f90.

Referenced by `read_xml_type_tally_xml()`.

Here is the caller graph for this function:



7.11.1.13 subroutine `xml_data_input_t::init_xml_type_tally_xml_array` (
`type(tally_xml),dimension(:),pointer dvar` **)**

Definition at line 953 of file input_t.f90.

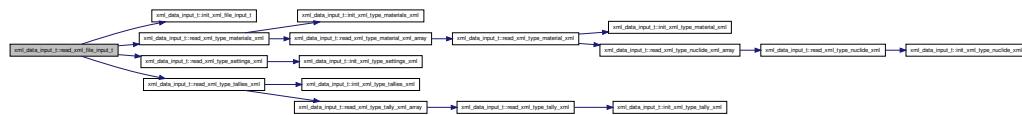
7.11.1.14 subroutine `xml_data_input_t::read_xml_file_input_t` (character(len=*)**,intent(in)** *fname*, **integer,intent(in),optional** *lurep*, **logical,intent(out),optional** *errout*)

Definition at line 1147 of file input_t.f90.

References `init_xml_file_input_t()`, `lurep_`, `materials_`, `read_xml_type_materials_xml()`, `read_xml_type_settings_xml()`, `read_xml_type_tallies_xml()`, `settings_`, `strict_`, and `tallies_-`.

Referenced by `input::read_input()`.

Here is the call graph for this function:



Here is the caller graph for this function:



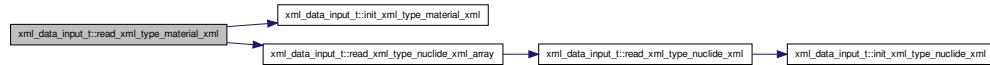
7.11.1.15 subroutine `xml_data_input_t::read_xml_type_material_xml` (**type(XML_PARSE)** *info*, character(len=*)**,intent(in)** *starttag*, **logical,intent(inout)** *endtag*, character(len=*)**,dimension(:,:)**,**intent(inout)** *attribs*, **integer,intent(inout)** *noattribs*, character(len=*)**,dimension(:)**,**intent(inout)** *data*, **integer,intent(inout)** *nodata*, **type(material_xml)**,**intent(inout)** *dvar*, **logical,intent(inout)** *has_dvar*)

Definition at line 503 of file input_t.f90.

References `init_xml_type_material_xml()`, `lurep_`, `read_xml_type_nuclide_xml_array()`, and `strict_`.

Referenced by `read_xml_type_material_xml_array()`.

Here is the call graph for this function:



Here is the caller graph for this function:



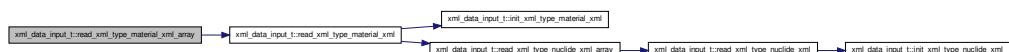
7.11.1.16 subroutine `xml_data_input_t::read_xml_type_material_xml_array` (type(XML_PARSE) *info*,
 character(len=*)*intent(inout)* *tag*, logical*intent(inout)*
endtag, character(len=*)*dimension(:, :)**intent(inout)*
attribs, integer*intent(inout)* *noattribs*,
 character(len=*)*dimension(:)**intent(inout)* *data*, integer*intent(inout)*
nodata, type(material_xml)*dimension(:)**pointer* *dvar*,
 logical*intent(inout)* *has_dvar*)

Definition at line 477 of file input_t.f90.

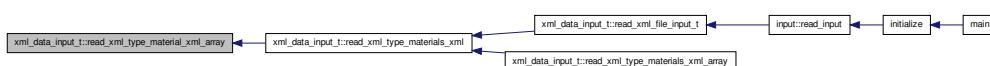
References `read_xml_type_material_xml()`.

Referenced by `read_xml_type_materials_xml()`.

Here is the call graph for this function:



Here is the caller graph for this function:



```
7.11.1.17 subroutine xml_data_input_t::read_xml_type_-
materials_xml ( type(XML_PARSE) info,
character(len=*),intent(in) starttag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, ),intent(inout)
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, ),intent(inout) data, integer,intent(inout)
nodata, type(materials_xml),intent(inout) dvar, logical,intent(inout)
has_dvar )
```

Definition at line 677 of file input_t.f90.

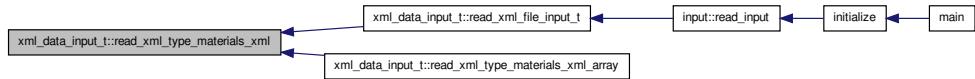
References init_xml_type_materials_xml(), lurep_, read_xml_type_material_xml_array(), and strict_.

Referenced by read_xml_file_input_t(), and read_xml_type_materials_xml_array().

Here is the call graph for this function:



Here is the caller graph for this function:



```
7.11.1.18 subroutine xml_data_input_t::read_xml_type_-
materials_xml_array ( type(XML_PARSE) info,
character(len=*),intent(inout) tag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, ),intent(inout)
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, ),intent(inout) data, integer,intent(inout)
nodata, type(materials_xml),dimension(:, ),pointer dvar,
logical,intent(inout) has_dvar )
```

Definition at line 651 of file input_t.f90.

References read_xml_type_materials_xml().

Here is the call graph for this function:



```
7.11.1.19 subroutine xml_data_input_t::read_xml_-
type_nuclide_xml ( type(XML_PARSE) info,
character(len=*),intent(in) starttag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, :),intent(inout)
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, :),intent(inout) data, integer,intent(inout)
nodata, type(nuclide_xml),intent(inout) dvar, logical,intent(inout)
has_dvar )
```

Definition at line 297 of file input_t.f90.

References init_xml_type_nuclide_xml(), lurep_, and strict_.

Referenced by `read_xml_type_nuclide_xml_array()`.

Here is the call graph for this function:



Here is the caller graph for this function:



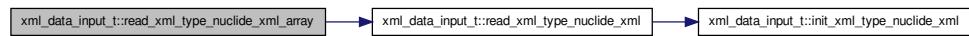
```
7.11.1.20 subroutine xml_data_input_t::read_xml_type_-
nuclide_xml_array ( type(XML_PARSE) info,
character(len=*),intent(inout) tag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, intent(inout))
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, intent(inout) data, integer,intent(inout)
nodata, type(nuclide_xml),dimension(:, pointer dvar,
logical,intent(inout) has_dvar )
```

Definition at line 271 of file input_t.f90.

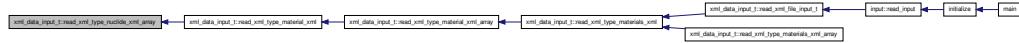
References read_xml_type_nuclide_xml().

Referenced by read_xml_type_material_xml().

Here is the call graph for this function:



Here is the caller graph for this function:



```
7.11.1.21 subroutine xml_data_input_t::read_xml_-
type_settings_xml ( type(XML_PARSE) info,
character(len=*),intent(in) starttag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, intent(inout))
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, intent(inout) data, integer,intent(inout)
nodata, type(settings_xml),intent(inout) dvar, logical,intent(inout)
has_dvar )
```

Definition at line 80 of file input_t.f90.

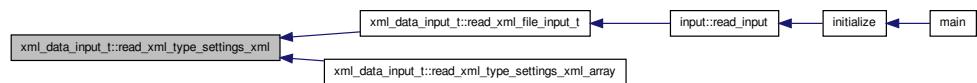
References init_xml_type_settings_xml(), lurep_, and strict_.

Referenced by read_xml_file_input_t(), and read_xml_type_settings_xml_array().

Here is the call graph for this function:



Here is the caller graph for this function:



7.11.1.22 subroutine `xml_data_input_t::read_xml_type_settings_xml_array` (`type(XML_PARSE) info`,
`character(len=*),intent(inout) tag`, `logical,intent(inout) endtag`, `character(len=*),dimension(:, :, :),intent(inout) attribs`,
`integer,intent(inout) noattribs`,
`character(len=*),dimension(:, :),intent(inout) data`, `integer,intent(inout) nodata`,
`type(settings_xml),dimension(:, :),pointer dvar`,
`logical,intent(inout) has_dvar`)

Definition at line 54 of file `input_t.f90`.

References `read_xml_type_settings_xml()`.

Here is the call graph for this function:



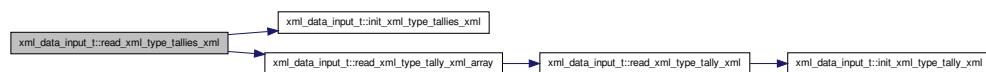
```
7.11.1.23 subroutine xml_data_input_t::read_xml_type_tallies_xml ( type(XML_PARSE) info,
character(len=*),intent(in) starttag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, intent(inout))
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, intent(inout) data, integer,intent(inout)
nodata, type(tallies_xml),intent(inout) dvar, logical,intent(inout)
has_dvar )
```

Definition at line 1021 of file input_t.f90.

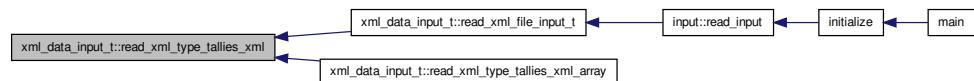
References init_xml_type_tallies_xml(), lurep_, read_xml_type_tally_xml_array(), and strict_.

Referenced by read_xml_file_input_t(), and read_xml_type_tallies_xml_array().

Here is the call graph for this function:



Here is the caller graph for this function:

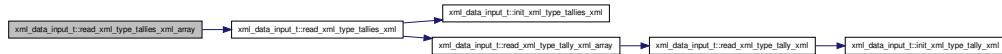


```
7.11.1.24 subroutine xml_data_input_t::read_xml_type_tallies_xml_array ( type(XML_PARSE) info,
character(len=*),intent(inout) tag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, intent(inout))
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, intent(inout) data, integer,intent(inout)
nodata, type(tallies_xml),dimension(:, pointer dvar,
logical,intent(inout) has_dvar )
```

Definition at line 995 of file input_t.f90.

References read_xml_type_tallies_xml().

Here is the call graph for this function:



7.11.1.25 subroutine `xml_data_input_t::read_xml_type_tally_xml`

```

subroutine xml_data_input_t::read_xml_type_tally_xml ( type(XML_PARSE) info,
character(len=*),intent(in) starttag, logical,intent(inout)
endtag, character(len=*),dimension(:,:),intent(inout)
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:,),intent(inout) data, integer,intent(inout)
nodata, type(tally_xml),intent(inout) dvar, logical,intent(inout)
has_dvar )
  
```

Definition at line 829 of file input_t.f90.

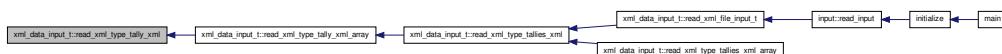
References init_xml_type_tally_xml(), lurep_, and strict_.

Referenced by read_xml_type_tally_xml_array().

Here is the call graph for this function:



Here is the caller graph for this function:



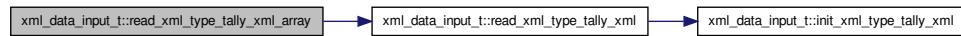
```
7.11.1.26 subroutine xml_data_input_t::read_xml_type_tally_xml_array ( type(XML_PARSE) info,
character(len=*),intent(inout) tag, logical,intent(inout)
endtag, character(len=*),dimension(:, :),intent(inout)
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:),intent(inout) data, integer,intent(inout)
nodata, type(tally_xml),dimension(:),pointer dvar,
logical,intent(inout) has_dvar )
```

Definition at line 803 of file input_t.f90.

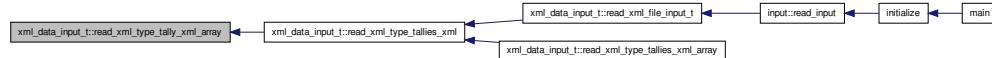
References `read_xml_type_tally_xml()`.

Referenced by `read_xml_type_tallies_xml()`.

Here is the call graph for this function:



Here is the caller graph for this function:



```
7.11.1.27 subroutine xml_data_input_t::write_xml_file_input_t (
character(len=*),intent(in) fname, integer,intent(in),optional lurep )
```

Definition at line 1248 of file input_t.f90.

References `materials_`, `settings_`, `tallies_`, `write_xml_type_materials_xml()`, `write_xml_type_settings_xml()`, and `write_xml_type_tallies_xml()`.

Here is the call graph for this function:



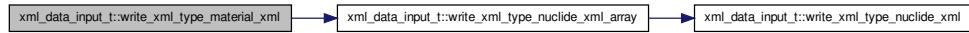
7.11.1.28 subroutine `xml_data_input_t::write_xml_type_material_xml` (
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(material_xml) dvar)`

Definition at line 634 of file input_t.f90.

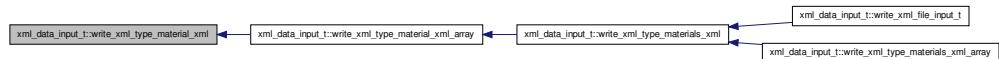
References `write_xml_type_nuclide_xml_array()`.

Referenced by `write_xml_type_material_xml_array()`.

Here is the call graph for this function:



Here is the caller graph for this function:



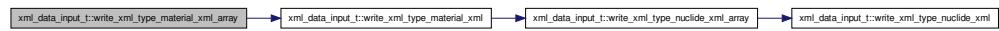
7.11.1.29 subroutine `xml_data_input_t::write_xml_type_material_xml_array` (
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(material_xml),dimension(:) dvar)`

Definition at line 622 of file input_t.f90.

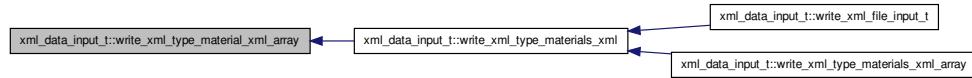
References `write_xml_type_material_xml()`.

Referenced by `write_xml_type_materials_xml()`.

Here is the call graph for this function:



Here is the caller graph for this function:



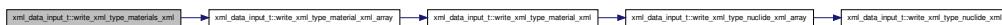
7.11.1.30 subroutine `xml_data_input_t::write_xml_type_materials_xml` (type(XML_PARSE) *info*, character(len=*)*intent(in)* *tag*, integer *indent*, type(materials_xml) *dvar*)

Definition at line 788 of file input_t.f90.

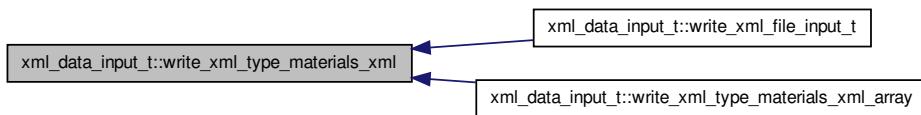
References `write_xml_type_material_xml_array()`.

Referenced by `write_xml_file_input_t()`, and `write_xml_type_materials_xml_array()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.11.1.31 subroutine `xml_data_input_t::write_xml_type_materials_xml_array` (type(XML_PARSE) *info*, character(len=*)*intent(in)* *tag*, integer *indent*, type(materials_xml),dimension(:) *dvar*)

Definition at line 776 of file input_t.f90.

References `write_xml_type_materials_xml()`.

Here is the call graph for this function:



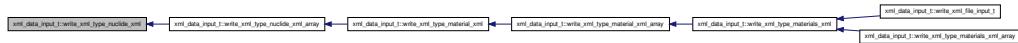
7.11.1.32 subroutine `xml_data_input_t::write_xml_type_nuclide_xml` (

`type(XML_PARSE) info, character(len=*) intent(in) tag, integer indent, type(nuclide_xml) dvar)`

Definition at line 457 of file input_t.f90.

Referenced by `write_xml_type_nuclide_xml_array()`.

Here is the caller graph for this function:



7.11.1.33 subroutine `xml_data_input_t::write_xml_type_nuclide_xml_array` (

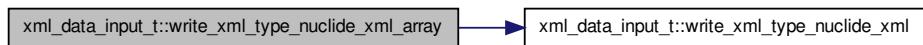
`type(XML_PARSE) info, character(len=*) intent(in) tag, integer indent, type(nuclide_xml), dimension(:) dvar)`

Definition at line 445 of file input_t.f90.

References `write_xml_type_nuclide_xml()`.

Referenced by `write_xml_type_material_xml()`.

Here is the call graph for this function:



Here is the caller graph for this function:

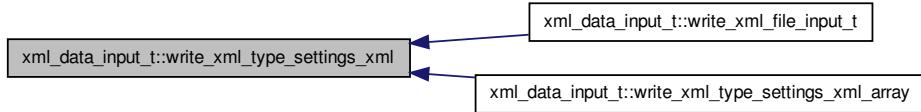


7.11.1.34 subroutine `xml_data_input_t::write_xml_type_settings_xml` (
`type(XML_PARSE) info, character(len=*),intent(in) tag, integer indent, type(settings_xml) dvar)`

Definition at line 250 of file input_t.f90.

Referenced by `write_xml_file_input_t()`, and `write_xml_type_settings_xml_array()`.

Here is the caller graph for this function:

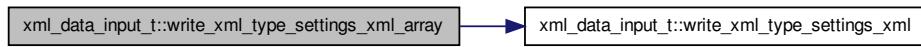


7.11.1.35 subroutine `xml_data_input_t::write_xml_type_settings_xml_array` (
`type(XML_PARSE) info, character(len=*),intent(in) tag, integer indent, type(settings_xml),dimension(:) dvar)`

Definition at line 238 of file input_t.f90.

References `write_xml_type_settings_xml()`.

Here is the call graph for this function:



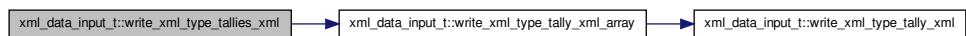
7.11.1.36 subroutine `xml_data_input_t::write_xml_type_tallies_xml` (
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(tallies_xml) dvar)`

Definition at line 1132 of file input_t.f90.

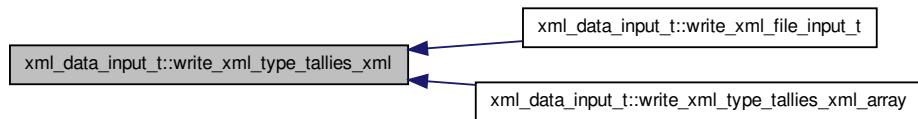
References `write_xml_type_tally_xml_array()`.

Referenced by `write_xml_file_input_t()`, and `write_xml_type_tallies_xml_array()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.11.1.37 subroutine `xml_data_input_t::write_xml_type_tallies_xml_array` (
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(tallies_xml),dimension(:) dvar)`

Definition at line 1120 of file input_t.f90.

References `write_xml_type_tallies_xml()`.

Here is the call graph for this function:

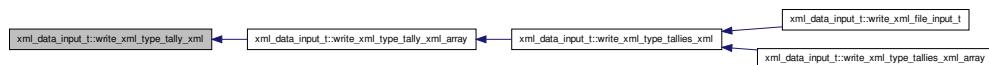


7.11.1.38 subroutine `xml_data_input_t::write_xml_type_tally_xml` (
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(tally_xml) dvar)`

Definition at line 976 of file input_t.f90.

Referenced by `write_xml_type_tally_xml_array()`.

Here is the caller graph for this function:



7.11.1.39 subroutine `xml_data_input_t::write_xml_type_tally_xml_array` (
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(tally_xml),dimension(:) dvar)`

Definition at line 964 of file input_t.f90.

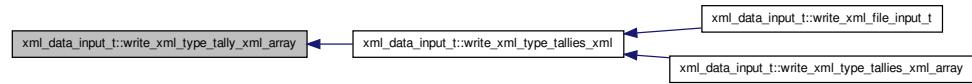
References `write_xml_type_tally_xml()`.

Referenced by `write_xml_type_tallies_xml()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.11.2 Variable Documentation**7.11.2.1 integer,private xml_data_input_t::lurep_**

Definition at line 7 of file input_t.f90.

Referenced by read_xml_file_input_t(), read_xml_type_material_xml(), read_xml_type_materials_xml(), read_xml_type_nuclide_xml(), read_xml_type_settings_xml(), read_xml_type_tallies_xml(), and read_xml_type_tally_xml().

7.11.2.2 type(materials_xml) xml_data_input_t::materials_

Definition at line 51 of file input_t.f90.

Referenced by input::read_input(), read_xml_file_input_t(), and write_xml_file_input_t().

7.11.2.3 type(settings_xml) xml_data_input_t::settings_

Definition at line 50 of file input_t.f90.

Referenced by input::read_input(), read_xml_file_input_t(), and write_xml_file_input_t().

7.11.2.4 logical,private xml_data_input_t::strict_

Definition at line 8 of file input_t.f90.

Referenced by read_xml_file_input_t(), read_xml_type_material_xml(), read_xml_type_materials_xml(), read_xml_type_nuclide_xml(), read_xml_type_settings_xml(), read_xml_type_tallies_xml(), and read_xml_type_tally_xml().

7.11.2.5 type(tallies_xml) xml_data_input_t::tallies_

Definition at line 52 of file input_t.f90.

Referenced by input::read_input(), read_xml_file_input_t(), and write_xml_file_input_t().

7.12 xmlparse Module Reference

Data Types

- type [XML_PARSE](#)
- interface [xml_report_details](#)
- interface [xml_report_errors](#)

Functions/Subroutines

- subroutine [xml_report_errors_extern_](#) (info, text)
- subroutine [xml_open](#) (info, fname, mustread)
- subroutine [xml_close](#) (info)
- subroutine [xml_get](#) (info, tag, endtag, attribs, no_attribs, data, no_data)
- subroutine [xml_put](#) (info, tag, attribs, no_attribs, data, no_data, type)
- subroutine [xml_options](#) (info, ignore_whitespace, no_data_truncation, report_lun, report_errors, report_details)
- logical [xml_ok](#) (info)
- logical [xml_error](#) (info)
- logical [xml_data_trunc](#) (info)
- integer [xml_find_attrib](#) (attribs, no_attribs, name, value)
- recursive subroutine [xml_process](#) (filename, attribs, data, [startfunc](#), [datafunc](#), [endfunc](#), lunrep, error)

Variables

- integer, parameter [XML_BUFFER_LENGTH](#) = 10000
- integer, parameter [XML_STDOUT](#) = -1
- integer, private [report_lun_](#) = [XML_STDOUT](#)
- logical, private [report_errors_](#) = .false.
- logical, private [report_details_](#) = .false.
- character(len=10), dimension(2, 3), save, private [entities](#) = reshape((/ '&', '>', '>', '<', '<', '/), (/2,3/))

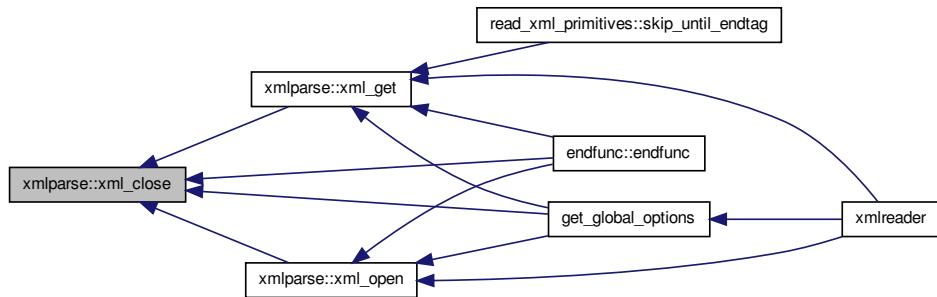
7.12.1 Function/Subroutine Documentation

7.12.1.1 subroutine [xmlparse::xml_close](#) (type(XML_PARSE), intent(inout) info)

Definition at line 332 of file `xmlparse.f90`.

Referenced by `endfunc::endfunc()`, `get_global_options()`, `xml_get()`, and `xml_open()`.

Here is the caller graph for this function:



7.12.1.2 logical `xmlparse::xml_data_trunc` (type(XML_PARSE),intent(in) *info*)

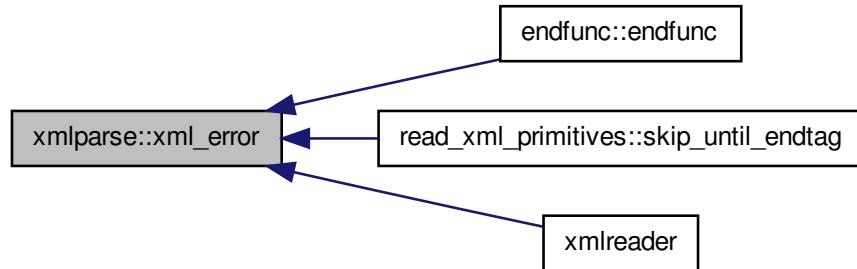
Definition at line 932 of file `xmlparse.f90`.

7.12.1.3 logical `xmlparse::xml_error` (type(XML_PARSE),intent(in) *info*)

Definition at line 915 of file `xmlparse.f90`.

Referenced by `endfunc::endfunc()`, `read_xml_primitives::skip_until_endtag()`, and `xmlreader()`.

Here is the caller graph for this function:

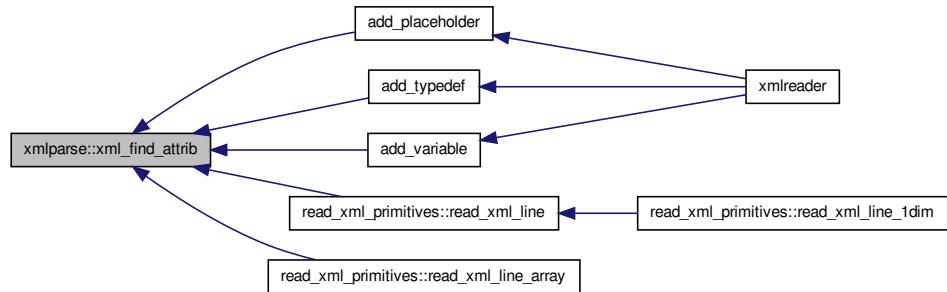


**7.12.1.4 integer `xmlparse::xml_find_attrib` (`character(len=*)`,
`dimension(:,::)` *attribs*, `integer no_attribs`, `character(len=*) name`, `character(len=*) value`)**

Definition at line 942 of file `xmlparse.f90`.

Referenced by `add_placeholder()`, `add_typeDefinition()`, `add_variable()`, `read_xml_primitives::read_xml_line()`, and `read_xml_primitives::read_xml_line_array()`.

Here is the caller graph for this function:



```
7.12.1.5 subroutine xmlparse::xml_get ( type(XML_PARSE),intent(inout)
  info, character(len=*),intent(out) tag, logical,intent(out) endtag,
  character(len=*),dimension(:, :, :),intent(out) attrs, integer,intent(out)
  no_attrs, character(len=*),dimension(:, :),intent(out) data,
  integer,intent(out) no_data )
```

Definition at line 358 of file xmlparse.f90.

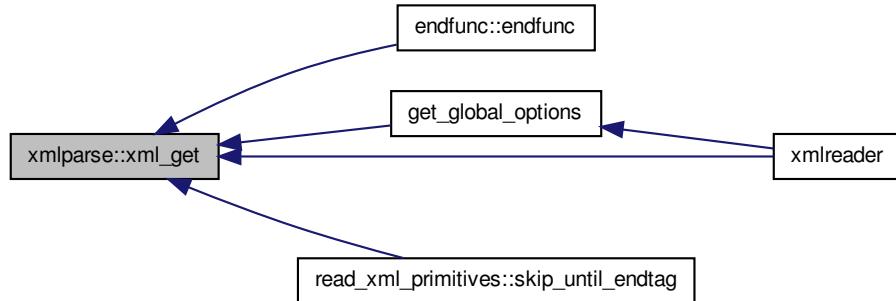
References `xml_close()`.

Referenced by `endfunc::endfunc()`, `get_global_options()`, `read_xml_primitives::skip_until_endtag()`, and `xmlreader()`.

Here is the call graph for this function:



Here is the caller graph for this function:

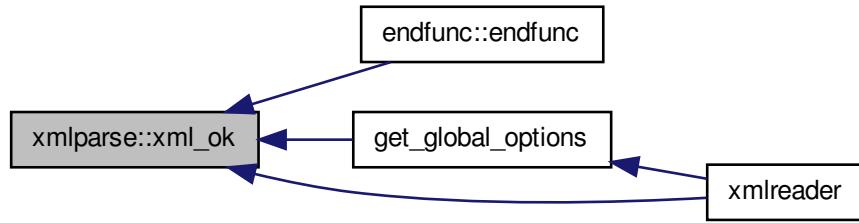


```
7.12.1.6 logical xmlparse::xml_ok ( type(XML_PARSE),intent(in) info )
```

Definition at line 897 of file xmlparse.f90.

Referenced by endfunc::endfunc(), get_global_options(), and xmlreader().

Here is the caller graph for this function:



7.12.1.7 subroutine xmlparse::xml_open (type(XML_PARSE),intent(out) info, character(len=*),intent(in) fname, logical,intent(in) mustread)

Definition at line 252 of file xmlparse.f90.

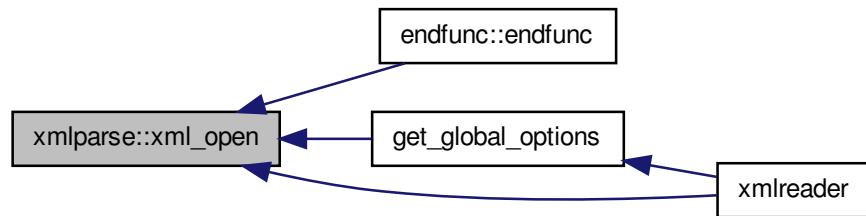
References xml_close().

Referenced by endfunc::endfunc(), get_global_options(), and xmlreader().

Here is the call graph for this function:



Here is the caller graph for this function:



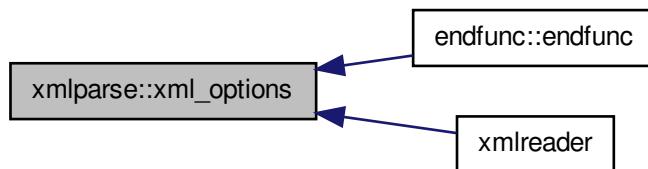
7.12.1.8 subroutine `xmlparse::xml_options` (`type(XML_PARSE),intent(inout)`
`info, logical,intent(in),optional ignore_whitespace,`
`logical,intent(in),optional no_data_truncation,`
`integer,intent(in),optional report_lun, logical,intent(in),optional`
`report_errors, logical,intent(in),optional report_details)`

Definition at line 860 of file `xmlparse.f90`.

References `report_details_`, `report_errors_`, and `report_lun_`.

Referenced by `enfunc::endfunc()`, and `xmlreader()`.

Here is the caller graph for this function:



**7.12.1.9 recursive subroutine xmlparse::xml_process (character(len=*)
filename, character(len=*),dimension(:, :) attribs,
character(len=*),dimension(:) data, startfunc startfunc, datafunc
datafunc, endfunc endfunc, integer lunrep, logical error)**

Definition at line 978 of file xmlparse.f90.

**7.12.1.10 subroutine xmlparse::xml_put (type(XML_-
PARSE),intent(inout) info, character(len=*),intent(in) tag,
character(len=*),dimension(:, :),intent(in) attribs, integer,intent(in)
no_attribs, character(len=*),dimension(:,),intent(in) data,
integer,intent(in) no_data, character(len=*) type)**

Definition at line 614 of file xmlparse.f90.

References entities.

7.12.1.11 subroutine xmlparse::xml_report_errors_extern_ (
type(XML_PARSE),intent(in) info, character(len=*),intent(in) text)

Definition at line 229 of file xmlparse.f90.

References report_details_, report_errors_, report_lun_, and XML_STDOUT.

7.12.2 Variable Documentation

**7.12.2.1 character(len=10),dimension(2,3),save,private xmlparse::entities =
reshape((/ '&', '>', '>', '<', '<', '/), (/2,3/))**

Definition at line 81 of file xmlparse.f90.

Referenced by xml_put().

7.12.2.2 logical,private xmlparse::report_details_ = .false.

Definition at line 75 of file xmlparse.f90.

Referenced by xml_options(), xmlparse::xml_report_details::xml_report_details_int_()
, xmlparse::xml_report_details::xml_report_details_string_(), xml_report_errors_extern_()
, xmlparse::xml_report_errors::xml_report_errors_int_(), and xmlparse::xml_report_-
errors::xml_report_errors_string_().

7.12.2.3 logical,private xmlparse::report_errors_ = .false.

Definition at line 74 of file xmlparse.f90.

Referenced by xml_options(), xml_report_errors_extern_(), xmlparse::xml_report_errors::xml_report_errors_int_(), and xmlparse::xml_report_errors::xml_report_errors_string_().

7.12.2.4 integer,private xmlparse::report_lun_ = XML_STDOUT

Definition at line 73 of file xmlparse.f90.

Referenced by xml_options(), xmlparse::xml_report_details::xml_report_details_int_(), xmlparse::xml_report_details::xml_report_details_string_(), xml_report_errors_extern_(), xmlparse::xml_report_errors::xml_report_errors_int_(), and xmlparse::xml_report_errors::xml_report_errors_string_().

7.12.2.5 integer,parameter xmlparse::XML_BUFFER_LENGTH = 10000

Definition at line 49 of file xmlparse.f90.

7.12.2.6 integer,parameter xmlparse::XML_STDOUT = -1

Definition at line 72 of file xmlparse.f90.

Referenced by xmlparse::xml_report_details::xml_report_details_int_(), xmlparse::xml_report_details::xml_report_details_string_(), xml_report_errors_extern_(), xmlparse::xml_report_errors::xml_report_errors_int_(), and xmlparse::xml_report_errors::xml_report_errors_string_().

8 Data Type Documentation

8.1 datafunc Interface Reference

Public Member Functions

- recursive subroutine [datafunc](#) (tag, data, error)

8.1.1 Detailed Description

Definition at line 995 of file xmlparse.f90.

8.1.2 Constructor & Destructor Documentation**8.1.2.1 recursive subroutine datafunc::datafunc (character(len=*) tag,
character(len=*),dimension(:) data, logical error)**

Definition at line 995 of file xmlparse.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[xmlparse.f90](#)

8.2 endfunc Interface Reference**Public Member Functions**

- recursive subroutine [endfunc](#) (tag, error)

8.2.1 Detailed Description

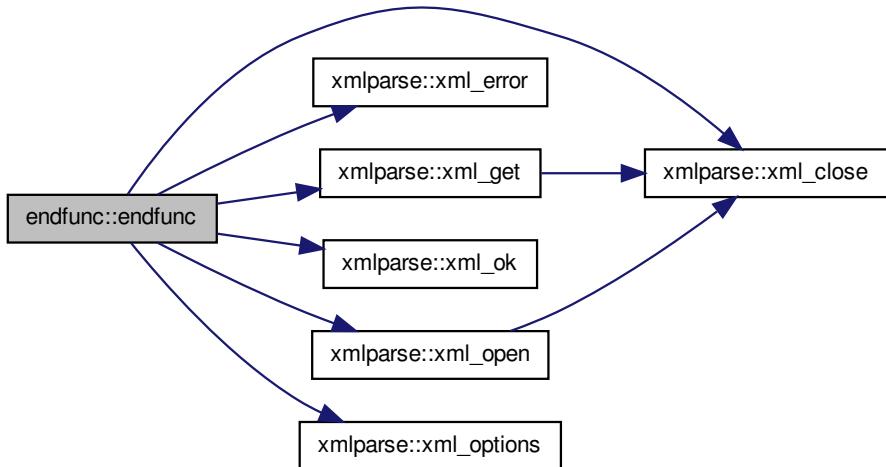
Definition at line 1003 of file xmlparse.f90.

8.2.2 Constructor & Destructor Documentation**8.2.2.1 recursive subroutine endfunc::endfunc (character(len=*) tag, logical
error)**

Definition at line 1003 of file xmlparse.f90.

References `xmlparse::xml_close()`, `xmlparse::xml_error()`, `xmlparse::xml_get()`, `xmlparse::xml_ok()`, `xmlparse::xml_open()`, and `xmlparse::xml_options()`.

Here is the call graph for this function:

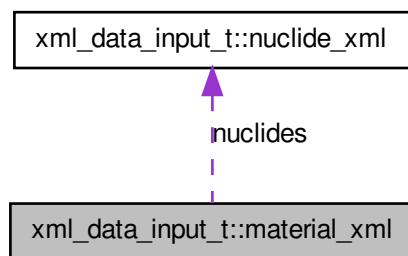


The documentation for this interface was generated from the following file:

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[xmlparse.f90](#)

8.3 `xml_data_input_t::material_xml` Type Reference

Collaboration diagram for `xml_data_input_t::material_xml`:



Public Attributes

- character(len=255) **type**
- real(kind=kind(1.0d0)) **V**
- **type(nuclide_xml)**, dimension(:), pointer **nuclides** = > null()

8.3.1 Detailed Description

Definition at line 29 of file input_t.f90.

8.3.2 Member Data Documentation**8.3.2.1 type(nuclide_xml),dimension(:),pointer xml_data_input_t::material_-
xml::nuclides = > null()**

Definition at line 32 of file input_t.f90.

8.3.2.2 character(len=255) xml_data_input_t::material_xml::type

Definition at line 30 of file input_t.f90.

8.3.2.3 real(kind=kind(1.0d0)) xml_data_input_t::material_xml::V

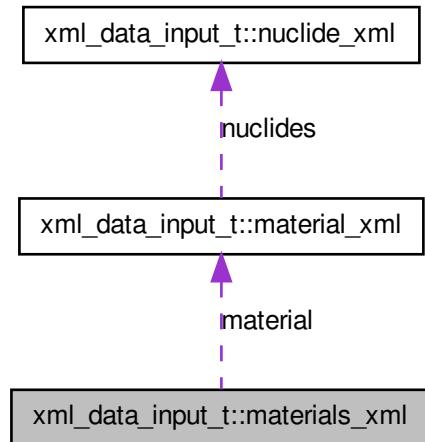
Definition at line 31 of file input_t.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/[input_t.f90](#)

8.4 xml_data_input_t::materials_xml Type Reference

Collaboration diagram for xml_data_input_t::materials_xml:



Public Attributes

- type(material_xml), dimension(:), pointer material => null()

8.4.1 Detailed Description

Definition at line 35 of file input_t.f90.

8.4.2 Member Data Documentation

8.4.2.1 type(material_xml),dimension(:),pointer xml_data_input_t::materials_- xml::material = > null()

Definition at line 36 of file input_t.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/[input_t.f90](#)

8.5 xml_data_input_t::nuclide_xml Type Reference

Public Attributes

- real(kind=kind(1.0d0)) **N**
- real(kind=kind(1.0d0)) **A**
- real(kind=kind(1.0d0)) **V**
- character(len=255) **path**
- logical **thermal**
- character(len=255) **name**

8.5.1 Detailed Description

Definition at line 20 of file input_t.f90.

8.5.2 Member Data Documentation

8.5.2.1 real(kind=kind(1.0d0)) xml_data_input_t::nuclide_xml::A

Definition at line 22 of file input_t.f90.

8.5.2.2 real(kind=kind(1.0d0)) xml_data_input_t::nuclide_xml::N

Definition at line 21 of file input_t.f90.

8.5.2.3 character(len=255) xml_data_input_t::nuclide_xml::name

Definition at line 26 of file input_t.f90.

8.5.2.4 character(len=255) xml_data_input_t::nuclide_xml::path

Definition at line 24 of file input_t.f90.

8.5.2.5 logical xml_data_input_t::nuclide_xml::thermal

Definition at line 25 of file input_t.f90.

8.5.2.6 `real(kind=kind(1.0d0)) xml_data_input_t::nuclide_xml::V`

Definition at line 23 of file `input_t.f90`.

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

- `/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/input_t.f90`

8.6 `read_xml_primitives::read_from_buffer` Interface Reference**Public Member Functions**

- subroutine `read_from_buffer_integers` (`buffer, var, ierror`)
- subroutine `read_from_buffer_reals` (`buffer, var, ierror`)
- subroutine `read_from_buffer_doubles` (`buffer, var, ierror`)
- subroutine `read_from_buffer_logicals` (`buffer, var, ierror`)
- subroutine `read_from_buffer_words` (`buffer, var, ierror`)

8.6.1 Detailed Description

Definition at line 30 of file `read_xml_primitives.f90`.

8.6.2 Member Function/Subroutine Documentation**8.6.2.1 subroutine `read_xml_primitives::read_from_buffer::read_from_buffer_doubles` (`buffer , real(kind=kind(1.0d00)),dimension(:),pointer var, ierror`)**

Definition at line 331 of file `read_xml_primitives.f90`.

8.6.2.2 subroutine `read_xml_primitives::read_from_buffer::read_from_buffer_integers` (`buffer , integer,dimension(:),pointer var, ierror`)

Definition at line 312 of file `read_xml_primitives.f90`.

8.6.2.3 subroutine `read_xml_primitives::read_from_buffer::read_from_buffer_logicals` (`buffer , logical,dimension(:),pointer var, ierror`)

Definition at line 339 of file read_xml_primitives.f90.

8.6.2.4 subroutine read_xml_primitives::read_from_buffer::read_from_buffer_reals (buffer , real,dimension(:),pointer var, ierror)

Definition at line 323 of file read_xml_primitives.f90.

8.6.2.5 subroutine read_xml_primitives::read_from_buffer::read_from_buffer_words (buffer , character(len=*),dimension(:),pointer var, ierror)

Definition at line 347 of file read_xml_primitives.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[read_xml_primitives.f90](#)

8.7 xml_data_input_t::settings_xml Type Reference

Public Attributes

- integer histories
- integer seed
- integer source_type
- character(len=255) source_path
- real(kind=kind(1.0d0)) Dancoff
- character(len=255) res_iso
- real(kind=kind(1.0d0)) radius

8.7.1 Detailed Description

Definition at line 10 of file input_t.f90.

8.7.2 Member Data Documentation

8.7.2.1 real(kind=kind(1.0d0)) xml_data_input_t::settings_xml::Dancoff

Definition at line 15 of file input_t.f90.

8.7.2.2 integer xml_data_input_t::settings_xml::histories

Definition at line 11 of file input_t.f90.

8.7.2.3 real(kind=kind(1.0d0)) xml_data_input_t::settings_xml::radius

Definition at line 17 of file input_t.f90.

8.7.2.4 character(len=255) xml_data_input_t::settings_xml::res_iso

Definition at line 16 of file input_t.f90.

8.7.2.5 integer xml_data_input_t::settings_xml::seed

Definition at line 12 of file input_t.f90.

8.7.2.6 character(len=255) xml_data_input_t::settings_xml::source_path

Definition at line 14 of file input_t.f90.

8.7.2.7 integer xml_data_input_t::settings_xml::source_type

Definition at line 13 of file input_t.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/[input_t.f90](#)

8.8 startfunc Interface Reference**Public Member Functions**

- recursive subroutine [startfunc](#) (tag, attrs, error)

8.8.1 Detailed Description

Definition at line 987 of file xmlparse.f90.

8.8.2 Constructor & Destructor Documentation

**8.8.2.1 recursive subroutine startfunc::startfunc (character(len=*) tag,
character(len=*),dimension(:,:) attrs, logical error)**

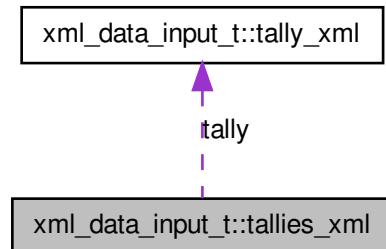
Definition at line 987 of file xmlparse.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[xmlparse.f90](#)

8.9 xml_data_input_t::tallies_xml Type Reference

Collaboration diagram for xml_data_input_t::tallies_xml:



Public Attributes

- type(tally_xml), dimension(:), pointer tally = > null()

8.9.1 Detailed Description

Definition at line 47 of file input_t.f90.

8.9.2 Member Data Documentation**8.9.2.1 type(tally_xml),dimension(:),pointer xml_data_input_t::tallies_-
xml::tally = > null()**

Definition at line 48 of file input_t.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/[input_t.f90](#)

8.10 xml_data_input_t::tally_xml Type Reference**Public Attributes**

- logical [dv](#)
- real(kind=kind(1.0d0)), dimension(:), pointer [Ebins](#) = > null()
- character(len=255) [type](#)
- integer [isotope](#)
- integer [region](#)

8.10.1 Detailed Description

Definition at line 39 of file input_t.f90.

8.10.2 Member Data Documentation**8.10.2.1 logical xml_data_input_t::tally_xml::dv**

Definition at line 40 of file input_t.f90.

**8.10.2.2 real(kind=kind(1.0d0)),dimension(:),pointer
xml_data_input_t::tally_xml::Ebins = > null()**

Definition at line 41 of file input_t.f90.

8.10.2.3 integer xml_data_input_t::tally_xml::isotope

Definition at line 43 of file input_t.f90.

8.10.2.4 integer xml_data_input_t::tally_xml::region

Definition at line 44 of file input_t.f90.

8.10.2.5 character(len=255) xml_data_input_t::tally_xml::type

Definition at line 42 of file input_t.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/[input_t.f90](#)

8.11 timing::Timer Type Reference**Public Attributes**

- logical [running](#) = .false.
- integer [start_counts](#) = 0
- real(8) [elapsed](#) = 0.

8.11.1 Detailed Description

Definition at line 11 of file timing.f90.

8.11.2 Member Data Documentation**8.11.2.1 real(8) timing::Timer::elapsed = 0.**

Definition at line 14 of file timing.f90.

8.11.2.2 logical timing::Timer::running = .false.

Definition at line 12 of file timing.f90.

8.11.2.3 integer timing::Timer::start_counts = 0

Definition at line 13 of file timing.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/[timing.f90](#)

8.12 write_xml_primitives::write_to_xml_line Interface Reference

Public Member Functions

- subroutine [write_to_xml_string](#) (info, tag, indent, value)

8.12.1 Detailed Description

Definition at line 27 of file write_xml_primitives.f90.

8.12.2 Member Function/Subroutine Documentation

8.12.2.1 subroutine write_xml_primitives::write_to_xml_line::write_to_xml_string (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, character(len=*),intent(in) value)

Definition at line 184 of file write_xml_primitives.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[write_xml_primitives.f90](#)

8.13 write_xml_primitives::write_to_xml_word Interface Reference

Public Member Functions

- subroutine [write_to_xml_string](#) (info, tag, indent, value)

8.13.1 Detailed Description

Definition at line 24 of file write_xml_primitives.f90.

8.13.2 Member Function/Subroutine Documentation

8.13.2.1 subroutine write_xml_primitives::write_to_xml_word::write_to_xml_string (type(XML_PARSE),intent(in) info, character(len=*),intent(in) tag, integer,intent(in) indent, character(len=*),intent(in) value)

Definition at line 184 of file write_xml_primitives.f90.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[write_-xml_primitives.f90](#)

8.14 xmlparse::XML_PARSE Type Reference

Public Attributes

- integer [lun](#)
- integer [level](#)
- integer [lineno](#)
- logical [ignore_whitespace](#)
- logical [no_data_truncation](#)
- logical [too_many_attribs](#)
- logical [too_many_data](#)
- logical [eof](#)
- logical [error](#)
- character(len=[XML_BUFFER_LENGTH](#)) [line](#)

8.14.1 Detailed Description

Definition at line 55 of file xmlparse.f90.

8.14.2 Member Data Documentation

8.14.2.1 logical xmlparse::XML_PARSE::eof

Definition at line 63 of file xmlparse.f90.

8.14.2.2 logical xmlparse::XML_PARSE::error

Definition at line 64 of file xmlparse.f90.

8.14.2.3 logical xmlparse::XML_PARSE::ignore_whitespace

Definition at line 59 of file xmlparse.f90.

8.14.2.4 integer `xmlparse::XML_PARSE::level`

Definition at line 57 of file `xmlparse.f90`.

**8.14.2.5 character(len=XML_BUFFER_LENGTH)
`xmlparse::XML_PARSE::line`**

Definition at line 65 of file `xmlparse.f90`.

8.14.2.6 integer `xmlparse::XML_PARSE::lineno`

Definition at line 58 of file `xmlparse.f90`.

8.14.2.7 integer `xmlparse::XML_PARSE::lun`

Definition at line 56 of file `xmlparse.f90`.

8.14.2.8 logical `xmlparse::XML_PARSE::no_data_truncation`

Definition at line 60 of file `xmlparse.f90`.

8.14.2.9 logical `xmlparse::XML_PARSE::too_many_attribs`

Definition at line 61 of file `xmlparse.f90`.

8.14.2.10 logical `xmlparse::XML_PARSE::too_many_data`

Definition at line 62 of file `xmlparse.f90`.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[xmlparse.f90](#)

8.15 `xmlparse::xml_report_details` Interface Reference

Public Member Functions

- subroutine `xml_report_details_int_` (`text, int`)
- subroutine `xml_report_details_string_` (`text, string`)

8.15.1 Detailed Description

Definition at line 106 of file `xmlparse.f90`.

8.15.2 Member Function/Subroutine Documentation

8.15.2.1 subroutine `xmlparse::xml_report_details::xml_report_details_int_` (`character(len=*)intent(in) text, integer,intent(in) int`)

Definition at line 126 of file `xmlparse.f90`.

References `xmlparse::report_details_`, `xmlparse::report_lun_`, and `xmlparse::XML_-STDOUT`.

8.15.2.2 subroutine `xmlparse::xml_report_details::xml_report_details_string_` (`character(len=*)intent(in) text, character(len=*)intent(in) string`)

Definition at line 147 of file `xmlparse.f90`.

References `xmlparse::report_details_`, `xmlparse::report_lun_`, and `xmlparse::XML_-STDOUT`.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[xmlparse.f90](#)

8.16 `xmlparse::xml_report_errors` Interface Reference

Public Member Functions

- subroutine `xml_report_errors_int_` (`text, int, lineno`)
- subroutine `xml_report_errors_string_` (`text, string, lineno`)
- subroutine `xml_report_errors_extern_` (`info, text`)

8.16.1 Detailed Description

Definition at line 110 of file `xmlparse.f90`.

8.16.2 Member Function/Subroutine Documentation**8.16.2.1 subroutine xmlparse::xml_report_errors::xml_report_errors_extern_**
(**type(XML_PARSE),intent(in) info, character(len=*)intent(in) text**)

Definition at line 229 of file xmlparse.f90.

8.16.2.2 subroutine xmlparse::xml_report_errors::xml_report_errors_int_
(**character(len=*)intent(in) text, integer,intent(in) int,**
integer,intent(in),optional lineno)

Definition at line 169 of file xmlparse.f90.

References `xmlparse::report_details_`, `xmlparse::report_errors_`, `xmlparse::report_lun_-`, and `xmlparse::XML_STDOUT`.

8.16.2.3 subroutine xmlparse::xml_report_errors::xml_report_errors_string_
(**character(len=*)intent(in) text, character(len=*)intent(in) string,**
integer,intent(in),optional lineno)

Definition at line 198 of file xmlparse.f90.

References `xmlparse::report_details_`, `xmlparse::report_errors_`, `xmlparse::report_lun_-`, and `xmlparse::XML_STDOUT`.

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

- /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/[xmlparse.f90](#)

9 File Documentation

9.1 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/global.f90 File Reference

Modules

- module [global](#)
Contains all of the global variables.

Functions/Subroutines

- subroutine [global::allocate_problem](#) ()

allocates global variables for calculation

- subroutine `global::deallocate_problem()`
deallocates global variables
- subroutine `global::compute_macro_cross_sections()`
routine that handles the call to compute macro cross sections
- subroutine `global::add_to_tallies()`
routine that adds temporary value to tallies
- subroutine `global::bank_tallies()`
routine that record temporary history information in tallies
- subroutine `global::finalize_tallies()`
routine that calls another routine to compute tally statistics

Variables

- integer `global::VERSION_MAJOR = 0`
- integer `global::VERSION_MINOR = 1`
- integer `global::VERSION_RELEASE = 1`
- type(particle_type) `global::neut`
- type(material_type), dimension(:), allocatable `global::mat`
- type(tally_type), dimension(:), allocatable `global::tal`
- integer `global::nhistories`
- integer `global::seed`
- integer `global::source_type`
- integer `global::eidx`
- integer `global::n_tallies`
- integer `global::n_materials`
- integer `global::res_iso`
- real(8) `global::Dancoff`
- real(8) `global::radius`
- real(8) `global::emin = 1e-11_8`
- real(8) `global::emax = 20.0_8`
- real(8) `global::kT = 8.6173324e-5_8*300*1.0e-6_8`
- real(8) `global::nubar = 2.455_8`
- type(Timer) `global::time_init`
- type(Timer) `global::time_run`
- integer `global::n_abs = 0.0_8`
- integer `global::n_fiss = 0.0_8`
- real(8) `global::ana_kinf_mean = 0.0_8`
- real(8) `global::ana_kinf_std = 0.0_8`
- real(8) `global::col_kinf_mean = 0.0_8`
- real(8) `global::col_kinf_std = 0.0_8`

9.2 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/input.f90 File Reference

Modules

- module [input](#)

Handles reading in the input xml file and initializing global vars.

Functions/Subroutines

- subroutine, public [input::read_input](#)

Reads the input xml file and sets global variables.

9.3 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/main.f90 File Reference

Functions/Subroutines

- program [main](#)

- subroutine [initialize](#) ()

high level routine for initializing problem

- subroutine [run_problem](#) ()

main routine for executing the transport calculation

- subroutine [finalize](#) ()

routine that finalizes the problem

9.3.1 Function Documentation

9.3.1.1 subroutine main::finalize()

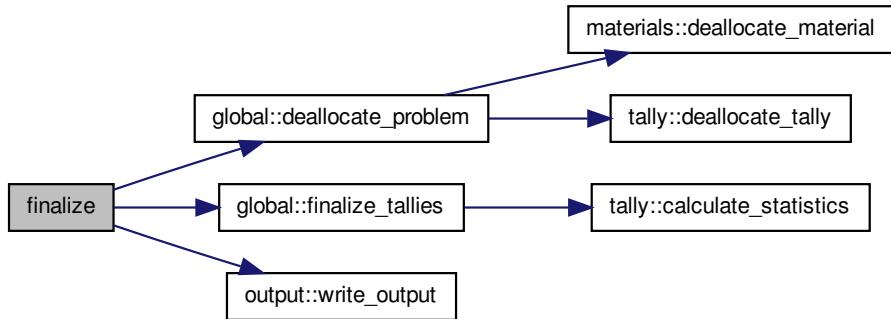
routine that finalizes the problem

Definition at line 157 of file main.f90.

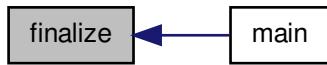
References [global::deallocate_problem\(\)](#), [global::finalize_tallies\(\)](#), and [output::write_output\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



9.3.1.2 subroutine main::initialize()

high level routine for initializing problem

Definition at line 58 of file main.f90.

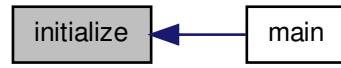
References output::print_heading(), input::read_input(), global::seed, global::time_init, timing::timer_start(), and timing::timer_stop().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

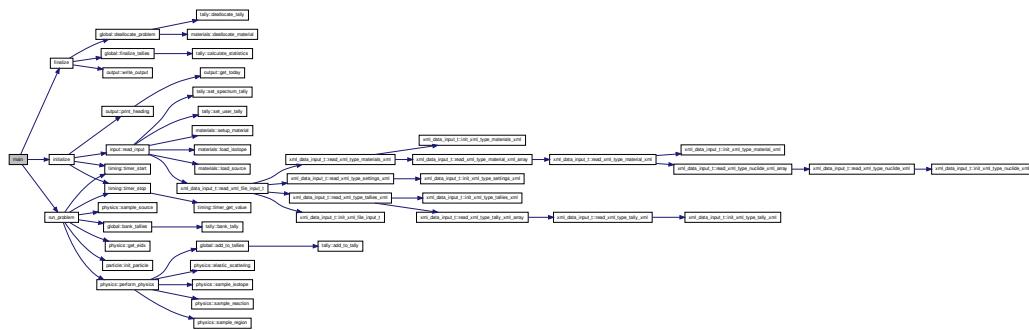


9.3.1.3 program main ()

Definition at line 1 of file main.f90.

References finalize(), initialize(), and run_problem().

Here is the call graph for this function:



9.3.1.4 subroutine main::run_problem()

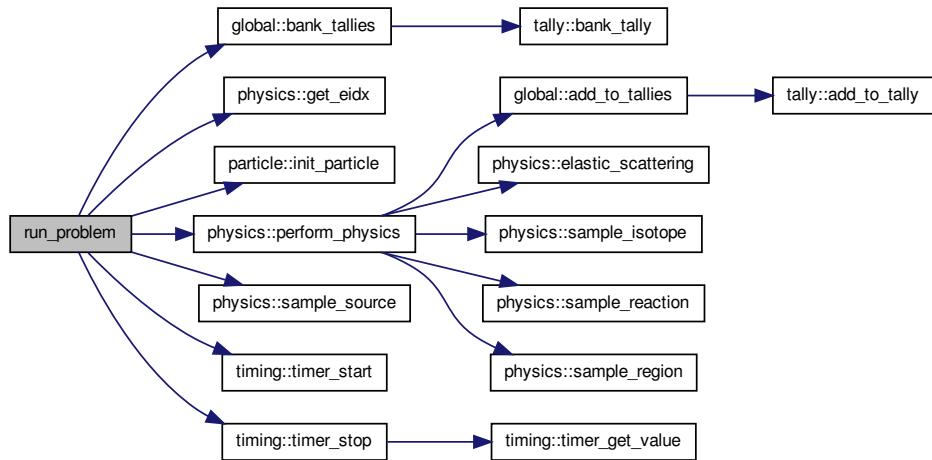
main routine for executing the transport calculation

Definition at line 100 of file main.f90.

References global::bank_tallies(), global::eidx, global::emin, physics::get_eidx(), particle::init_particle(), global::neut, global::nhistories, physics::perform_physics(), physics::sample_source(), global::time_run, timing::timer_start(), and timing::timer_stop().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



9.4

/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/materials.f90 File
Reference

101

9.4 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/materials.f90 File Reference

Data Types

- type **materials::source_type**
- type **materials::thermal_type**
- type **materials::iso_type**
- type **materials::material_type**

Modules

- module **materials**

Contains information about the isotopes of problem.

Functions/Subroutines

- subroutine, public **materials::setup_material** (this, emin, emax, nisotopes, vol)
routine that initializes the materials
- subroutine, public **materials::load_isotope** (this, N, A, path, thermal, name)
routine that loads isotope properties, xs, etc.
- subroutine, public **materials::load_source** (this, source_type, source_path)
routine to load fission source into memory
- subroutine, public **materials::compute_macros** (this)
routine to pre-compute macroscopic cross sections
- subroutine, public **materials::deallocate_material** (this)
routine to deallocate a material

9.5 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/output.f90 File Reference

Modules

- module **output**

Contains routines for outputting major info to user.

Functions/Subroutines

- subroutine, public **output::print_heading ()**
prints the code heading and run information
- subroutine, public **output::write_output ()**
routine that writes timing info and hdf5 file
- subroutine **output::get_today (today_date, today_time)**
calculates information about date/time of run

9.6 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/particle.f90 File Reference

Data Types

- type **particle::particle_type**

Modules

- module **particle**
Contains information about the particle that is transporting.

Functions/Subroutines

- subroutine, public **particle::init_particle (this)**
routine to initialize a particle

9.7 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/physics.f90 File Reference

Modules

- module **physics**
Contains routines to model the physics of the problem.

Functions/Subroutines

- subroutine, public **physics::sample_source ()**
routine to sample source from cdf

- subroutine, public `physics::perform_physics ()`
high level routine to perform transport physics
- integer, public `physics::get_eidx (E)`
function to compute the index in unionized energy grid
- integer `physics::sample_region ()`
function to sample region where interaction occurs
- integer `physics::sample_isotope (region)`
function to sample interaction isotope
- integer `physics::sample_reaction (region, isoidx)`
function to sample reaction type
- subroutine `physics::elastic_scattering (region, isoidx)`
routine to perform thermal/asymptotic elastic scattering physics

9.8 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/tally.f90 File Reference

Data Types

- type `tally::tally_type`

Modules

- module `tally`
Contains information about tallying quantities.

Functions/Subroutines

- subroutine, public `tally::set_user_tally (this, Ebins, n, react_type, isotope, region, n_materials, dv)`
routine to initialize user-defined tallies
- subroutine, public `tally::set_spectrum_tally (this, emax, emin, n_materials)`
routine to initialize all tallies
- subroutine, public `tally::set_kinf_tally (this, emax, emin, n_materials)`
routine to initialize kinf nu-fission tally
- subroutine, public `tally::add_to_tally (this, fact, totxs, E, region)`

routine to add quantities during transport of a particle

- subroutine, public [tally::bank_tally](#) (this)
routine to bank a histories tallies
- subroutine, public [tally::calculate_statistics](#) (this, n)
routine to compute mean and standard deviation of tallies
- subroutine, public [tally::deallocate_tally](#) (this)
routine to deallocate tally types

9.9 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/timing.f90 File Reference

Data Types

- type [timing::Timer](#)

Modules

- module [timing](#)

Functions/Subroutines

- subroutine [timing::timer_start](#) (self)
- real(8) [timing::timer_get_value](#) (self)
- subroutine [timing::timer_stop](#) (self)
- subroutine [timing::timer_reset](#) (self)

Variables

- real(8) [timing::ZERO](#) = 1.0_8

9.10 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/read_xml_primitives.f90 File Reference

Data Types

- interface [read_xml_primitives::read_from_buffer](#)

Modules

- module [read_xml_primitives](#)

Functions/Subroutines

- subroutine `read_xml_primitives::skip_until_endtag` (info, tag, attribs, data, error)
- subroutine `read_xml_primitives::read_xml_integer` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_line` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_real` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_double` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_logical` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_word` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_integer_array` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_line_array` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_real_array` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_double_array` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_logical_array` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_word_array` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_integer_1dim` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_real_1dim` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_double_1dim` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_logical_1dim` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_word_1dim` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)
- subroutine `read_xml_primitives::read_xml_line_1dim` (info, tag, endtag, attribs, noattribs, data, nodata, var, has_var)

9.11 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/input_t.f90 File Reference

Data Types

- type `xml_data_input_t::settings_xml`

- type `xml_data_input_t::nuclide_xml`
- type `xml_data_input_t::material_xml`
- type `xml_data_input_t::materials_xml`
- type `xml_data_input_t::tally_xml`
- type `xml_data_input_t::tallies_xml`

Modules

- module `xml_data_input_t`

Functions/Subroutines

- subroutine `xml_data_input_t::read_xml_type_settings_xml_array` (info, tag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::read_xml_type_settings_xml` (info, starttag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::init_xml_type_settings_xml_array` (dvar)
- subroutine `xml_data_input_t::init_xml_type_settings_xml` (dvar)
- subroutine `xml_data_input_t::write_xml_type_settings_xml_array` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::write_xml_type_settings_xml` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::read_xml_type_nuclide_xml_array` (info, tag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::read_xml_type_nuclide_xml` (info, starttag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::init_xml_type_nuclide_xml_array` (dvar)
- subroutine `xml_data_input_t::init_xml_type_nuclide_xml` (dvar)
- subroutine `xml_data_input_t::write_xml_type_nuclide_xml_array` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::write_xml_type_nuclide_xml` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::read_xml_type_material_xml_array` (info, tag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::read_xml_type_material_xml` (info, starttag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::init_xml_type_material_xml_array` (dvar)
- subroutine `xml_data_input_t::init_xml_type_material_xml` (dvar)
- subroutine `xml_data_input_t::write_xml_type_material_xml_array` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::write_xml_type_material_xml` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::read_xml_type_materials_xml_array` (info, tag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::read_xml_type_materials_xml` (info, starttag, end-tag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::init_xml_type_materials_xml_array` (dvar)
- subroutine `xml_data_input_t::init_xml_type_materials_xml` (dvar)

- subroutine `xml_data_input_t::write_xml_type_materials_xml_array` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::write_xml_type_materials_xml` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::read_xml_type_tally_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::read_xml_type_tally_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::init_xml_type_tally_xml_array` (dvar)
- subroutine `xml_data_input_t::init_xml_type_tally_xml` (dvar)
- subroutine `xml_data_input_t::write_xml_type_tally_xml_array` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::write_xml_type_tally_xml` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::read_xml_type_tallies_xml_array` (info, tag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::read_xml_type_tallies_xml` (info, starttag, endtag, attribs, noattribs, data, nodata, dvar, has_dvar)
- subroutine `xml_data_input_t::init_xml_type_tallies_xml_array` (dvar)
- subroutine `xml_data_input_t::init_xml_type_tallies_xml` (dvar)
- subroutine `xml_data_input_t::write_xml_type_tallies_xml_array` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::write_xml_type_tallies_xml` (info, tag, indent, dvar)
- subroutine `xml_data_input_t::read_xml_file_input_t` (fname, lurep, errout)
- subroutine `xml_data_input_t::write_xml_file_input_t` (fname, lurep)
- subroutine `xml_data_input_t::init_xml_file_input_t`

Variables

- integer, private `xml_data_input_t::lurep_`
- logical, private `xml_data_input_t::strict_`
- type(settings_xml) `xml_data_input_t::settings_`
- type(materials_xml) `xml_data_input_t::materials_`
- type(tallies_xml) `xml_data_input_t::tallies_`

9.12 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/write_xml_primitives.f90 File Reference

Data Types

- interface `write_xml_primitives::write_to_xml_word`
- interface `write_xml_primitives::write_to_xml_line`

Modules

- module `write_xml_primitives`

Functions/Subroutines

- subroutine `write_xml_primitives::write_to_xml_integer` (info, tag, indent, value)
- subroutine `write_xml_primitives::write_to_xml_integer_1dim` (info, tag, indent, values)
- subroutine `write_xml_primitives::write_to_xml_real` (info, tag, indent, value)
- subroutine `write_xml_primitives::write_to_xml_real_1dim` (info, tag, indent, values)
- subroutine `write_xml_primitives::write_to_xml_double` (info, tag, indent, value)
- subroutine `write_xml_primitives::write_to_xml_double_1dim` (info, tag, indent, values)
- subroutine `write_xml_primitives::write_to_xml_string` (info, tag, indent, value)
- subroutine `write_xml_primitives::write_to_xml_word_1dim` (info, tag, indent, values)
- subroutine `write_xml_primitives::write_to_xml_string_1dim` (info, tag, indent, values)
- subroutine `write_xml_primitives::write_to_xml_logical` (info, tag, indent, value)
- subroutine `write_xml_primitives::write_to_xml_logical_1dim` (info, tag, indent, values)
- subroutine `write_xml_primitives::write_to_xml_integer_array` (info, tag, indent, array)
- subroutine `write_xml_primitives::write_to_xml_real_array` (info, tag, indent, array)
- subroutine `write_xml_primitives::write_to_xml_double_array` (info, tag, indent, array)
- subroutine `write_xml_primitives::write_to_xml_logical_array` (info, tag, indent, array)
- subroutine `write_xml_primitives::write_to_xml_word_array` (info, tag, indent, array)
- subroutine `write_xml_primitives::write_to_xml_line_array` (info, tag, indent, array)

9.13 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/xmlparse.f90 File Reference

Data Types

- type `xmlparse::XML_PARSE`
- interface `xmlparse::xml_report_details`
- interface `xmlparse::xml_report_errors`
- interface `startfunc`
- interface `datafunc`
- interface `endfunc`

Modules

- module `xmlparse`

Functions/Subroutines

- subroutine `xmlparse::xml_report_errors_extern_` (info, text)
- subroutine `xmlparse::xml_open` (info, fname, mustread)
- subroutine `xmlparse::xml_close` (info)
- subroutine `xmlparse::xml_get` (info, tag, endtag, attribs, no_attribs, data, no_data)
- subroutine `xmlparse::xml_put` (info, tag, attribs, data, no_data, type)
- subroutine `xmlparse::xml_options` (info, ignore_whitespace, no_data_truncation, report_lun, report_errors, report_details)
- logical `xmlparse::xml_ok` (info)
- logical `xmlparse::xml_error` (info)
- logical `xmlparse::xml_data_trunc` (info)
- integer `xmlparse::xml_find_attrib` (attribs, no_attribs, name, value)
- recursive subroutine `xmlparse::xml_process` (filename, attribs, data, `startfunc`, `datafunc`, `endfunc`, lunrep, error)

Variables

- integer, parameter `xmlparse::XML_BUFFER_LENGTH` = 10000
- integer, parameter `xmlparse::XML_STDOUT` = -1
- integer, private `xmlparse::report_lun_` = XML_STDOUT
- logical, private `xmlparse::report_errors_` = .false.
- logical, private `xmlparse::report_details_` = .false.
- character(len=10), dimension(2, 3), save, private `xmlparse::entities` = reshape((/ '&', '&', '>', '>', '<', '<' /), (/2,3/))

Functions/Subroutines

- program `xmlreader`
- subroutine `get_global_options` (attribs, noattribs, strict, global_type, global_name, root_name, dyn_strings)
- subroutine `set_options` (attribs, noattribs, strict, global_type, global_name, root_name, dyn_strings)
- subroutine `open_tmp_files` (lufirst)
- subroutine `close_tmp_files`
- subroutine `append_files` (lufirst)
- subroutine `merge_files`
- subroutine `write_prolog`
- subroutine `add_begin_loop` (checktag, component)
- subroutine `add_end_loop`
- subroutine `add_variable` (component)
- subroutine `add_typeDefinition` (dyn_strings)

- subroutine `close_typedef` (component)
- subroutine `add_placeholder` (dyn_strings)
- subroutine `close_placeholder`

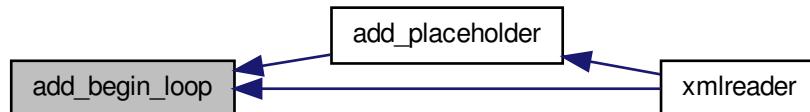
9.14.1 Function Documentation

9.14.1.1 subroutine xmlreader::add_begin_loop (logical checktag, logical component)

Definition at line 632 of file xmlreader.f90.

Referenced by `add_placeholder()`, and `xmlreader()`.

Here is the caller graph for this function:

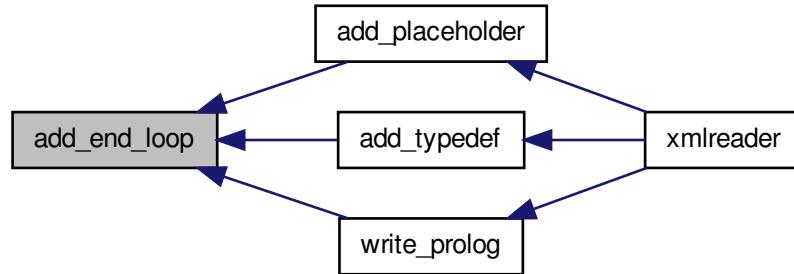


9.14.1.2 subroutine xmlreader::add_end_loop ()

Definition at line 716 of file xmlreader.f90.

Referenced by `add_placeholder()`, `add_typedef()`, and `write_prolog()`.

Here is the caller graph for this function:



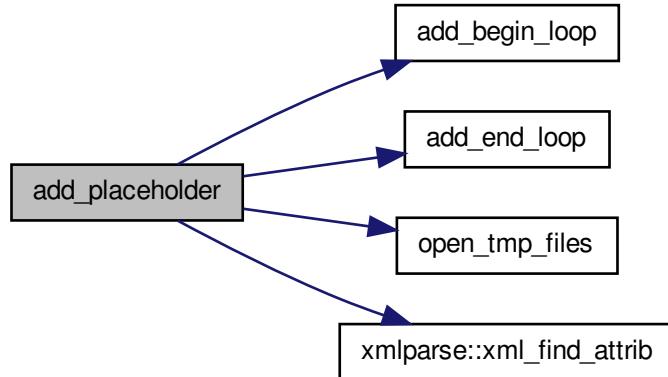
9.14.1.3 subroutine xmlreader::add_placeholder (logical,intent(in) dyn_strings)

Definition at line 1131 of file xmlreader.f90.

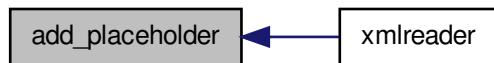
References add_begin_loop(), add_end_loop(), open_tmp_files(), and xmlparse::xml_find_attrib().

Referenced by xmlreader().

Here is the call graph for this function:



Here is the caller graph for this function:



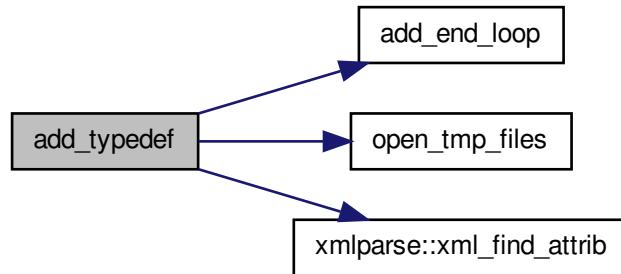
9.14.1.4 subroutine `xmlreader::add_typedef` (logical,intent(in) *dyn_strings*)

Definition at line 945 of file `xmlreader.f90`.

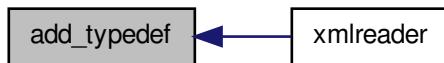
References `add_end_loop()`, `open_tmp_files()`, and `xmlparse::xml_find_attrib()`.

Referenced by `xmlreader()`.

Here is the call graph for this function:



Here is the caller graph for this function:



9.14.1.5 subroutine xmlreader::add_variable (logical component)

Definition at line 744 of file xmlreader.f90.

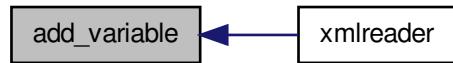
References `xmlparse::xml_find_attrib()`.

Referenced by `xmlreader()`.

Here is the call graph for this function:



Here is the caller graph for this function:

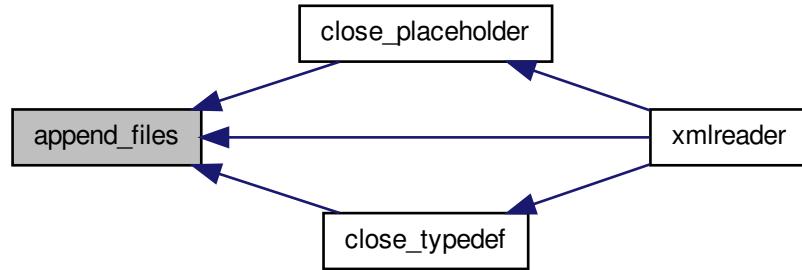


9.14.1.6 subroutine xmlreader::append_files (integer,intent(in) *lufirst*)

Definition at line 467 of file xmlreader.f90.

Referenced by close_placeholder(), close_typedef(), and xmlreader().

Here is the caller graph for this function:



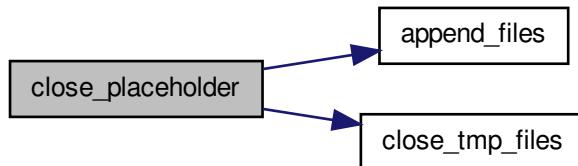
9.14.1.7 subroutine xmlreader::close_placeholder()

Definition at line 1223 of file xmlreader.f90.

References `append_files()`, and `close_tmp_files()`.

Referenced by `xmlreader()`.

Here is the call graph for this function:



Here is the caller graph for this function:

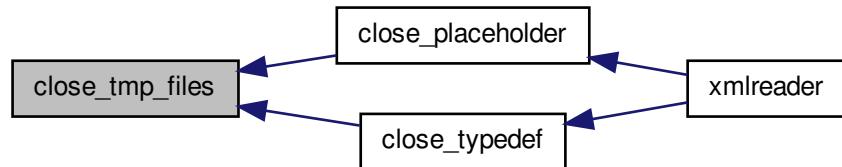


9.14.1.8 subroutine xmlreader::close_tmp_files()

Definition at line 444 of file xmlreader.f90.

Referenced by close_placeholder(), and close_typedef().

Here is the caller graph for this function:



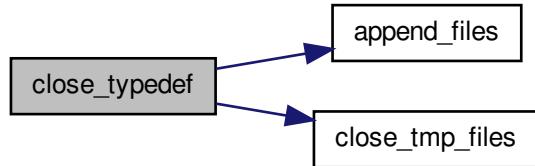
9.14.1.9 subroutine xmlreader::close_typedef(logical,intent(out) component)

Definition at line 1105 of file xmlreader.f90.

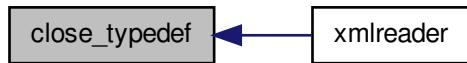
References append_files(), and close_tmp_files().

Referenced by xmlreader().

Here is the call graph for this function:



Here is the caller graph for this function:



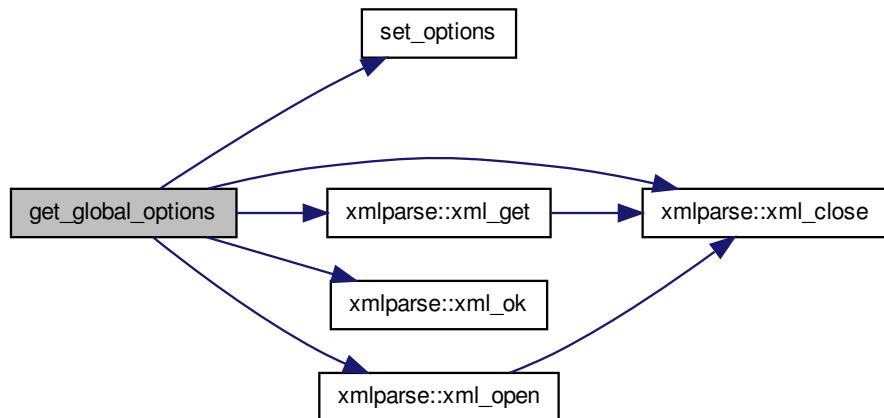
9.14.1.10 subroutine xmlreader::get_global_options (
character(len=*),dimension(:, :) intent(inout) *attribs*,
integer,intent(inout) *noattribs*, logical,intent(inout) *strict*,
logical,intent(inout) *global_type*, character(len=*),intent(inout)
global_name, character(len=*),intent(inout) *root_name*,
logical,intent(inout) *dyn_strings*)

Definition at line 328 of file xmlreader.f90.

References set_options(), xmlparse::xml_close(), xmlparse::xml_get(), xmlparse::xml_ok(), and xmlparse::xml_open().

Referenced by xmlreader().

Here is the call graph for this function:



Here is the caller graph for this function:



9.14.1.11 subroutine xmlreader::merge_files()

Definition at line 506 of file `xmlreader.f90`.

Referenced by `xmlreader()`.

Here is the caller graph for this function:

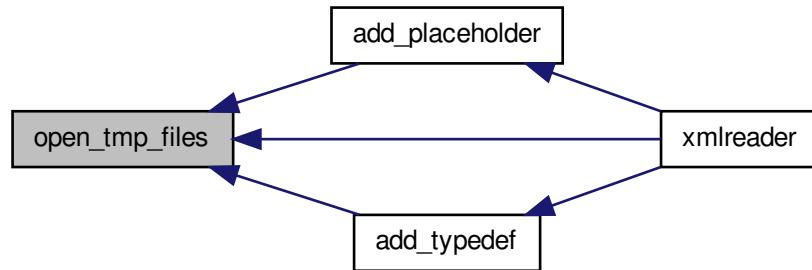


9.14.1.12 subroutine xmlreader::open_tmp_files (integer,intent(in) lufirst)

Definition at line 420 of file xmlreader.f90.

Referenced by add_placeholder(), add_typedef(), and xmlreader().

Here is the caller graph for this function:

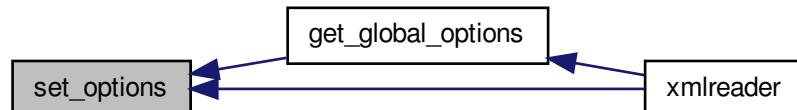


9.14.1.13 subroutine xmlreader::set_options (character(len=*),dimension(:, :) ,intent(in) attrs, integer,intent(in) noattrs, logical,intent(inout) strict, logical,intent(inout) global_type, character(len=*),intent(inout) global_name, character(len=*),intent(inout) root_name, logical,intent(inout) dyn_strings)

Definition at line 372 of file xmlreader.f90.

Referenced by get_global_options(), and xmlreader().

Here is the caller graph for this function:



9.14.1.14 subroutine xmlreader::write_prolog()

Definition at line 549 of file xmlreader.f90.

References add_end_loop().

Referenced by xmlreader().

Here is the call graph for this function:



Here is the caller graph for this function:



9.14.1.15 program xmlreader()

Definition at line 12 of file xmlreader.f90.

References add_begin_loop(), add_placeholder(), add_typedef(), add_variable(), append_files(), close_placeholder(), close_typedef(), get_global_options(), merge_files(), open_tmp_files(), set_options(), write_prolog(), xmlparse::xml_error(), xmlparse::xml_get(), xmlparse::xml_ok(), xmlparse::xml_open(), and xmlparse::xml_options().

Here is the call graph for this function:

