

# SlowMC

Generated by Doxygen 1.7.3

Mon Apr 2 2012 09:40:40

# Contents

<b>1 SlowMC: Slowing Down Monte Carlo</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Compiling . . . . .	2
1.3 Running . . . . .	2
<b>2 Directory Hierarchy</b>	<b>2</b>
2.1 Directories . . . . .	2
<b>3 Modules Index</b>	<b>2</b>
3.1 Modules List . . . . .	2
<b>4 Data Type Index</b>	<b>3</b>
4.1 Class List . . . . .	3
<b>5 File Index</b>	<b>4</b>
5.1 File List . . . . .	4
<b>6 Directory Documentation</b>	<b>5</b>
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
<b>7 Module Documentation</b>	<b>7</b>
7.1 global Module Reference . . . . .	7
7.1.1 Detailed Description . . . . .	8
7.1.2 Function/Subroutine Documentation . . . . .	8
7.1.3 Variable Documentation . . . . .	12
7.2 input Module Reference . . . . .	16
7.2.1 Detailed Description . . . . .	17
7.2.2 Function/Subroutine Documentation . . . . .	17
7.3 materials Module Reference . . . . .	18
7.3.1 Detailed Description . . . . .	18
7.3.2 Function/Subroutine Documentation . . . . .	19
7.4 output Module Reference . . . . .	21
7.4.1 Detailed Description . . . . .	21
7.4.2 Function/Subroutine Documentation . . . . .	21
7.5 particle Module Reference . . . . .	23
7.5.1 Detailed Description . . . . .	24
7.5.2 Function/Subroutine Documentation . . . . .	24
7.6 physics Module Reference . . . . .	24
7.6.1 Detailed Description . . . . .	25
7.6.2 Function/Subroutine Documentation . . . . .	25
7.7 read_xml_primitives Module Reference . . . . .	29
7.7.1 Function/Subroutine Documentation . . . . .	30
7.8 tally Module Reference . . . . .	38
7.8.1 Detailed Description . . . . .	39

7.8.2	Function/Subroutine Documentation . . . . .	39
7.9	timing Module Reference . . . . .	42
7.9.1	Function/Subroutine Documentation . . . . .	42
7.9.2	Variable Documentation . . . . .	44
7.10	write_xml_primitives Module Reference . . . . .	44
7.10.1	Function/Subroutine Documentation . . . . .	45
7.11	xml_data_input_t Module Reference . . . . .	51
7.11.1	Function/Subroutine Documentation . . . . .	52
7.11.2	Variable Documentation . . . . .	71
7.12	xmlparse Module Reference . . . . .	72
7.12.1	Function/Subroutine Documentation . . . . .	72
7.12.2	Variable Documentation . . . . .	78
<b>8</b>	<b>Data Type Documentation</b>	<b>79</b>
8.1	datafunc Interface Reference . . . . .	79
8.1.1	Detailed Description . . . . .	79
8.1.2	Constructor & Destructor Documentation . . . . .	80
8.2	endfunc Interface Reference . . . . .	80
8.2.1	Detailed Description . . . . .	80
8.2.2	Constructor & Destructor Documentation . . . . .	80
8.3	xml_data_input_t::material_xml Type Reference . . . . .	81
8.3.1	Detailed Description . . . . .	82
8.3.2	Member Data Documentation . . . . .	82
8.4	xml_data_input_t::materials_xml Type Reference . . . . .	83
8.4.1	Detailed Description . . . . .	83
8.4.2	Member Data Documentation . . . . .	83
8.5	xml_data_input_t::nuclide_xml Type Reference . . . . .	84
8.5.1	Detailed Description . . . . .	84
8.5.2	Member Data Documentation . . . . .	84
8.6	read_xml_primitives::read_from_buffer Interface Reference . . . . .	85
8.6.1	Detailed Description . . . . .	85
8.6.2	Member Function/Subroutine Documentation . . . . .	85
8.7	xml_data_input_t::settings_xml Type Reference . . . . .	86
8.7.1	Detailed Description . . . . .	86
8.7.2	Member Data Documentation . . . . .	86
8.8	startfunc Interface Reference . . . . .	87
8.8.1	Detailed Description . . . . .	88
8.8.2	Constructor & Destructor Documentation . . . . .	88
8.9	xml_data_input_t::tallies_xml Type Reference . . . . .	88
8.9.1	Detailed Description . . . . .	88
8.9.2	Member Data Documentation . . . . .	89
8.10	xml_data_input_t::tally_xml Type Reference . . . . .	89
8.10.1	Detailed Description . . . . .	89
8.10.2	Member Data Documentation . . . . .	89
8.11	timing::Timer Type Reference . . . . .	90
8.11.1	Detailed Description . . . . .	90
8.11.2	Member Data Documentation . . . . .	90
8.12	write_xml_primitives::write_to_xml_line Interface Reference . . . . .	91
8.12.1	Detailed Description . . . . .	91
8.12.2	Member Function/Subroutine Documentation . . . . .	91

---

8.13	write_xml_primitives::write_to_xml_word Interface Reference . . . . .	91
8.13.1	Detailed Description . . . . .	91
8.13.2	Member Function/Subroutine Documentation . . . . .	91
8.14	xmlparse::XML_PARSE Type Reference . . . . .	92
8.14.1	Detailed Description . . . . .	92
8.14.2	Member Data Documentation . . . . .	92
8.15	xmlparse::xml_report_details Interface Reference . . . . .	94
8.15.1	Detailed Description . . . . .	94
8.15.2	Member Function/Subroutine Documentation . . . . .	94
8.16	xmlparse::xml_report_errors Interface Reference . . . . .	94
8.16.1	Detailed Description . . . . .	94
8.16.2	Member Function/Subroutine Documentation . . . . .	95
<b>9</b>	<b>File Documentation</b>	<b>95</b>
9.1	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/global.f90 File Reference . . . . .	95
9.2	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/input.f90 File Reference . . . . .	97
9.3	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/main.f90 File Reference . . . . .	97
9.3.1	Function Documentation . . . . .	97
9.4	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/materials.f90 File Reference . . . . .	101
9.5	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/output.f90 File Reference . . . . .	101
9.6	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/particle.f90 File Reference . . . . .	102
9.7	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/physics.f90 File Reference . . . . .	102
9.8	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/tally.f90 File Reference . . . . .	103
9.9	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/timing.f90 File Reference . . . . .	104
9.10	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/read_xml_primitives.f90 File Reference . . . . .	104
9.11	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/input_t.f90 File Reference . . . . .	105
9.12	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/write_xml_primitives.f90 File Reference . . . . .	107
9.13	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/xmlparse.f90 File Reference . . . . .	108
9.14	/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/xmlreader.f90 File Reference . . . . .	109
9.14.1	Function Documentation . . . . .	110

## 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:

<b>src</b>	<b>5</b>
<b>xml-fortran</b>	<b>6</b>
<b>templates</b>	<b>6</b>

## 3 Modules Index

### 3.1 Modules List

Here is a list of all modules with brief descriptions:

<b>global</b> (Contains all of the global variables )	<b>7</b>
<b>input</b> (Handles reading in the input xml file and initializing global vars )	<b>16</b>
<b>materials</b> (Contains information about the isotopes of problem )	<b>18</b>
<b>output</b> (Contains routines for outputting major info to user )	<b>21</b>
<b>particle</b> (Contains information about the particle that is transporting )	<b>23</b>
<b>physics</b> (Contains routines to model the physics of the problem )	<b>24</b>
<b>read_xml_primitives</b>	<b>29</b>
<b>tally</b> (Contains information about tallying quantities )	<b>38</b>
<b>timing</b>	<b>42</b>
<b>write_xml_primitives</b>	<b>44</b>
<b>xml_data_input_t</b>	<b>51</b>
<b>xmlparse</b>	<b>72</b>

## 4 Data Type Index

### 4.1 Class List

Here are the data types with brief descriptions:

<b>datafunc</b>	<b>79</b>
<b>endfunc</b>	<b>80</b>
<b>xml_data_input_t::material_xml</b>	<b>81</b>
<b>xml_data_input_t::materials_xml</b>	<b>83</b>
<b>xml_data_input_t::nuclide_xml</b>	<b>84</b>
<b>read_xml_primitives::read_from_buffer</b>	<b>85</b>
<b>xml_data_input_t::settings_xml</b>	<b>86</b>
<b>startfunc</b>	<b>87</b>

<a href="#">xml_data_input_t::tallies_xml</a>	<a href="#">88</a>
<a href="#">xml_data_input_t::tally_xml</a>	<a href="#">89</a>
<a href="#">timing::Timer</a>	<a href="#">90</a>
<a href="#">write_xml_primitives::write_to_xml_line</a>	<a href="#">91</a>
<a href="#">write_xml_primitives::write_to_xml_word</a>	<a href="#">91</a>
<a href="#">xmlparse::XML_PARSE</a>	<a href="#">92</a>
<a href="#">xmlparse::xml_report_details</a>	<a href="#">94</a>
<a href="#">xmlparse::xml_report_errors</a>	<a href="#">94</a>

## 5 File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/global.f90</a>	<a href="#">95</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/input.f90</a>	<a href="#">97</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/main.f90</a>	<a href="#">97</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/materials.f90</a>	<a href="#">101</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/output.f90</a>	<a href="#">101</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/particle.f90</a>	<a href="#">102</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/physics.f90</a>	<a href="#">102</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/tally.f90</a>	<a href="#">103</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/timing.f90</a>	<a href="#">104</a>
<a href="#">/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/read_xml_primitives.f90</a>	<a href="#">104</a>

/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/**write\_-  
xml\_primitives.f90** 107

/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/**xmlparse.f90**  
**108**

/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/**xmlreader.f90**  
**109**

/mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/templates/**input\_-  
t.f90** 105

## 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](#)

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

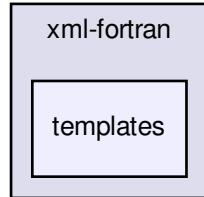
---

**6**

- file [tally.f90](#)
- file [timing.f90](#)

## **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](#)

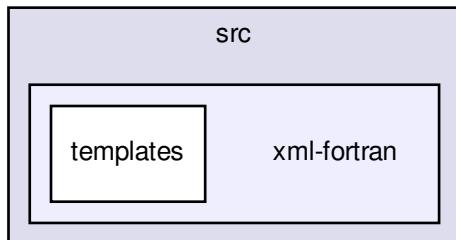
### 6.3

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

7

### 6.3 /mnt/md0/Documents/Documents/Spring2012/211/SlowMC/src/xml-fortran/ Di- rectory 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`.

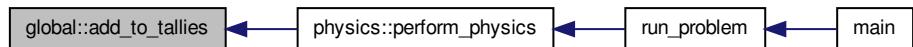
References `tally::add_to_tally()`, `eidx`, `mat`, `n_tallies`, `neut`, `nubar`, and `tal`.

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 222 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\_macroxs(), mat, and n\_materials.

Here is the call graph for this function:



#### 7.1.2.5 subroutine **global::deallocate\_problem( )**

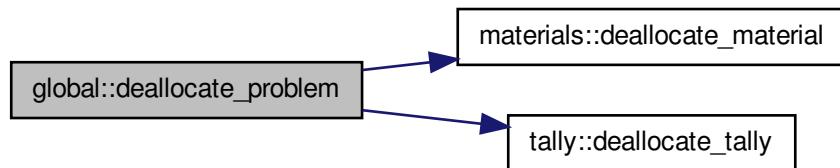
deallocates global variables

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( )**

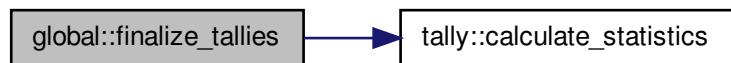
routine that calls another routine to compute tally statistics

Definition at line 244 of file global.f90.

References ana\_kinf\_mean, ana\_kinf\_std, tally::calculate\_statistics(), col\_kinf\_mean, col\_kinf\_std, mat, 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 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.

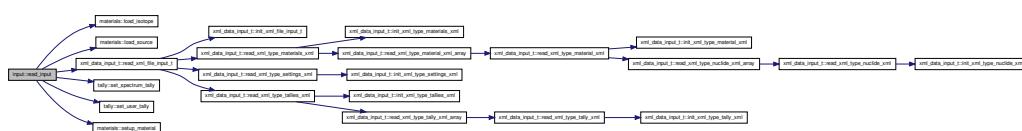
Definition at line 22 of file input.f90.

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\_

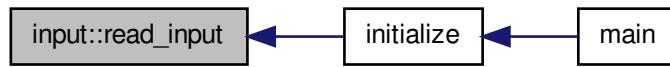
data\_input\_t::settings\_, materials::setup\_material(), global::source\_type, and xml\_data\_-  
input\_t::tallies\_.

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 isotopes 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\_macroxs** (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 isotopes 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 297 of file materials.f90.

Referenced by global::compute\_macro\_cross\_sections().

Here is the caller graph for this function:



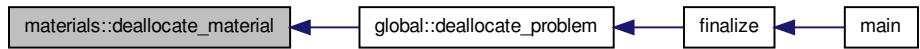
#### 7.3.2.2 subroutine,public materials::deallocate\_material ( type(material\_type) *this* )

routine to deallocate a material

Definition at line 346 of file materials.f90.

Referenced by global::deallocate\_problem().

Here is the caller graph for this function:



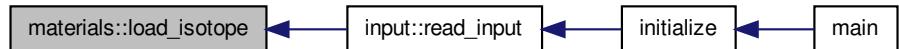
### 7.3.2.3 subroutine,public materials::load\_isotope ( type(material\_type),target this, real(8) N, real(8) A, character(len=255) path, logical thermal, character(len=255) name )

routine that loads isotope properties, xs, etc.  
into memory

Definition at line 108 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 240 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 76 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

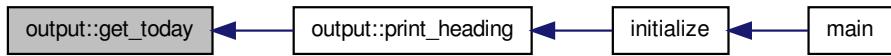
##### 7.4.2.1 subroutine `output::get_today` ( `character(10),intent(out) today_date,` `character(8),intent(out) today_time` )

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` ( )

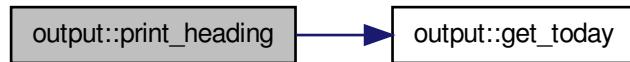
prints the code heading and run information

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 )**

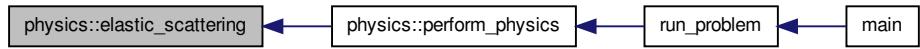
routine to perform thermal/asymptotic elastic scattering physics

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 )**

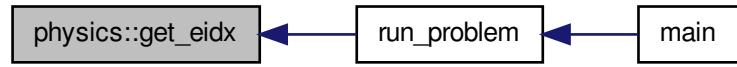
function to compute the index in unionized energy grid

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 ( )**

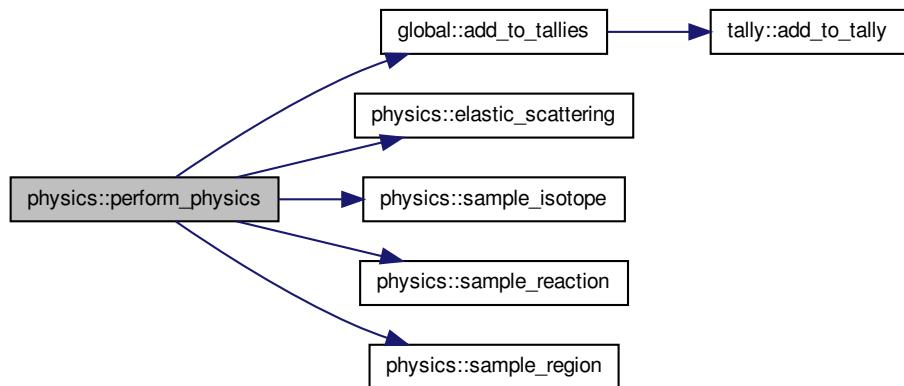
high level routine to perform transport physics

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* )

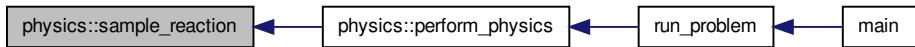
function to sample reaction type

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 ( )

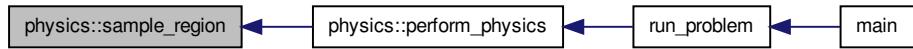
function to sample region where interaction occurs

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:

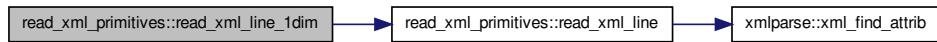


```
7.7.1.8 subroutine read_xml_primitives::read_xml_line_1dim (
    type(XML_PARSE),intent(inout) info,
    character(len=*),intent(in) tag, logical,intent(inout) endtag,
    character(len=*),dimension(:, :, intent(in) attribs, integer,intent(in)
noattribs, character(len=*),dimension(:, intent(in) data,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

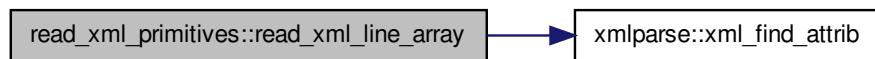


**7.7.1.9 subroutine `read_xml_primitives::read_xml_line_array`** ( `type(XML_PARSE),intent(inout) info,`  
`character(len=*),intent(in) tag, logical,intent(inout) endtag,`  
`character(len=*),dimension(:, :, :),intent(in) attribs, integer,intent(in)`  
`noattribs, character(len=*),dimension(:, :),intent(in)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

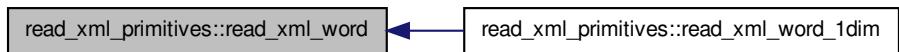
Definition at line 272 of file read\_xml\_primitives.f90.

```
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 )`

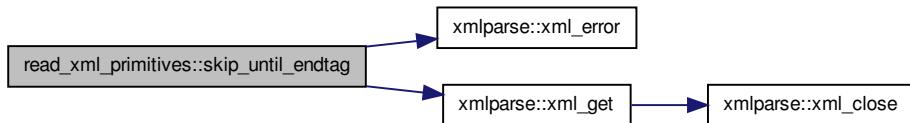
Definition at line 296 of file `read_xml_primitives.f90`.

**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 `xmllib::xml_error()`, and `xmllib::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

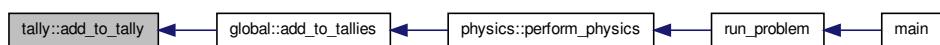
##### 7.8.2.1 subroutine,public tally::add\_to\_tally ( type(tally\_type) this, real(8) fact, real(8) totxs, real(8) E, integer region )

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 217 of file tally.f90.

Referenced by global::bank\_tallies().

Here is the caller graph for this function:

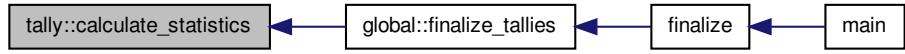
**7.8.2.3 subroutine,public tally::calculate\_statistics ( type(tally\_type) *this*, integer *n* )**

routine to compute mean and standard deviation of tallies

Definition at line 236 of file tally.f90.

Referenced by global::finalize\_tallies().

Here is the caller graph for this function:

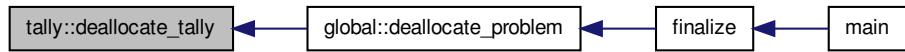
**7.8.2.4 subroutine,public tally::deallocate\_tally ( type(tally\_type) *this* )**

routine to deallocate tally types

Definition at line 255 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.

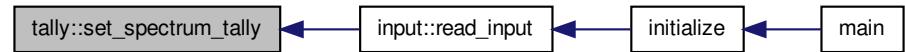
#### 7.8.2.6 subroutine,public tally::set\_spectrum\_tally ( type(tally\_type) this, real(8) emax, real(8) emin, integer n\_materials )

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:



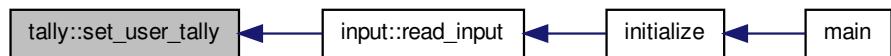
#### 7.8.2.7 subroutine,public tally::set\_user\_tally ( type(tally\_type) this, real(8),dimension(n) Ebins, integer n, integer react\_type, integer isotope, integer region, integer n\_materials, logical dv )

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

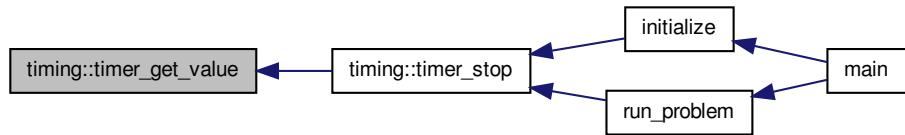
#### 7.9.1 Function/Subroutine Documentation

##### 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`.

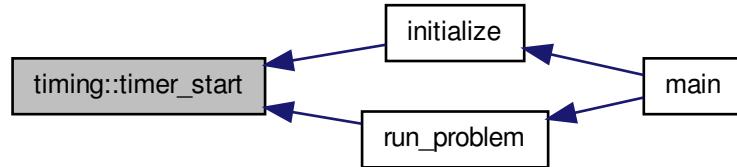
References ZERO.

#### 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:



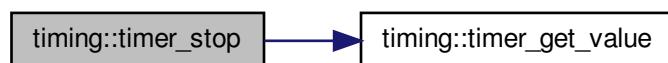
#### 7.9.1.4 subroutine `timing::timer_stop ( type(Timer),intent(inout) self )`

Definition at line 67 of file timing.f90.

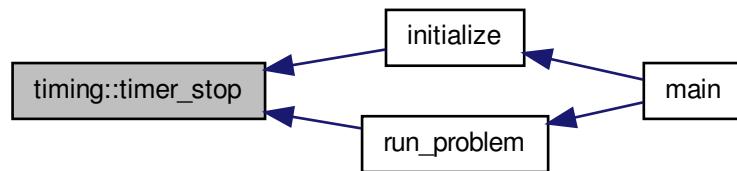
References timer\_get\_value().

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:

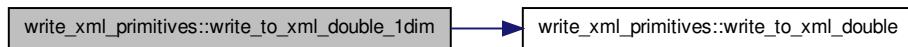


**7.10.1.2 subroutine write\_xml\_primitives::write\_to\_xml\_-**  
`double_1dim ( type(XML_PARSE),intent(in) info,`  
`character(len=*),intent(in) tag, integer,intent(in) indent,`  
`real(kind=kind(1.0d00)),dimension(:),intent(in) values )`

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:

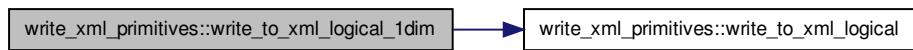


**7.10.1.9 subroutine write\_xml\_primitives::write\_to\_xml\_logical\_1dim (**  
    **type(XML\_PARSE),intent(in) info, character(len=\*),intent(in) tag,**  
    **integer,intent(in) indent, logical,dimension(:),intent(in) values )**

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 )**

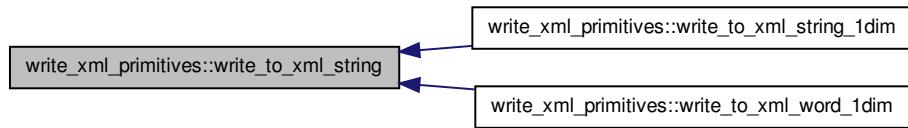
Definition at line 339 of file write\_xml\_primitives.f90.

**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_`

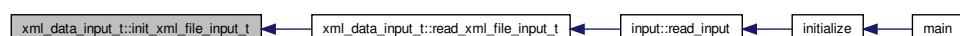
#### 7.11.1 Function/Subroutine Documentation

##### 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 )`

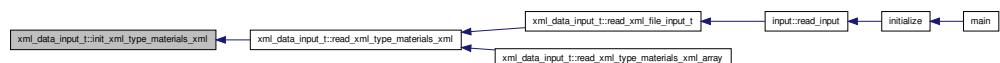
Definition at line 612 of file input\_t.f90.

### 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:



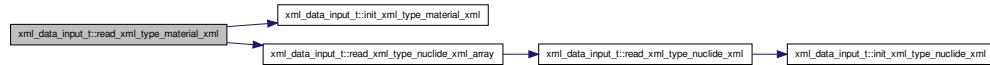
**7.11.1.15 subroutine `xml_data_input_t::read_xml_type_material_xml`** ( **type(XML\_PARSE)** *info*, character(len=\*)**,intent(in)** *starttag*, **logical,intent(inout)** *endtag*, character(len=\*)**,dimension(:,:)**,**,intent(inout)** *attribs*, **integer,intent(inout)** *noattribs*, character(len=\*)**,dimension(:)**,**,intent(inout)** *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:



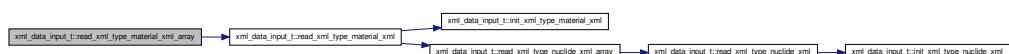
**7.11.1.16 subroutine `xml_data_input_t::read_xml_type_-  
material_xml_array`** ( type(XML\_PARSE) *info*,  
*character(len=\*)*,intent(inout) *tag*, logical,intent(inout)  
*endtag*, *character(len=\*)*,dimension(:,:),intent(inout)  
*attribs*, integer,intent(inout) *noattribs*,  
*character(len=\*)*,dimension(:,),intent(inout) *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.

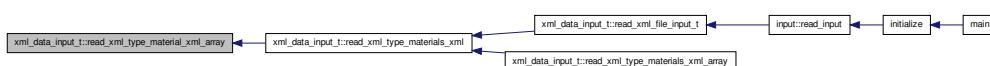
References `read_xml_type_material_xml()`.

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.

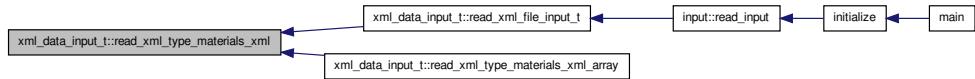
References init\_xml\_type\_materials\_xml(), lurep\_, read\_xml\_type\_material\_xml\_array(), and strict\_.

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:



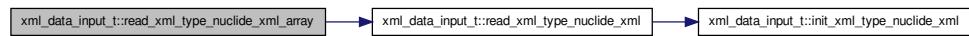
```
7.11.1.20 subroutine xml_data_input_t::read_xml_type_-
nuclide_xml_array ( type(XML_PARSE) info,
character(len=*),intent(inout) tag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, intent(inout))
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, intent(inout) 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.

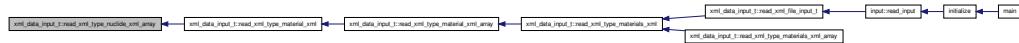
References read\_xml\_type\_nuclide\_xml().

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.

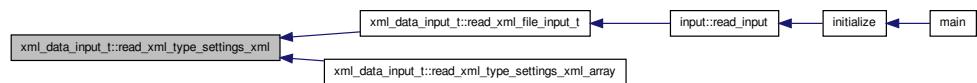
References init\_xml\_type\_settings\_xml(), lurep\_, and strict\_.

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:

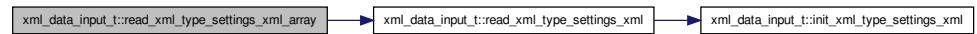


**7.11.1.22 subroutine `xml_data_input_t::read_xml_type_settings_xml_array`** ( `type(XML_PARSE) info`,  
`character(len=*),intent(inout) tag`, `logical,intent(inout) endtag`, `character(len=*),dimension(:, :, intent(inout)) attribs`,  
`integer,intent(inout) noattribs`, `character(len=*),dimension(:, 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.

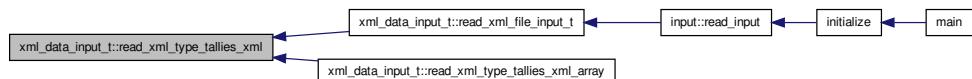
References init\_xml\_type\_tallies\_xml(), lurep\_, read\_xml\_type\_tally\_xml\_array(), and strict\_.

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:

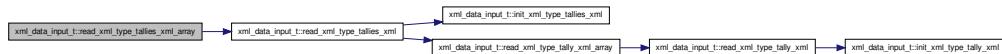


```
7.11.1.24 subroutine xml_data_input_t::read_xml_type_tallies_xml_array ( type(XML_PARSE) info,
character(len=*),intent(inout) tag, logical,intent(inout)
endtag, character(len=*),dimension(:, :, intent(inout))
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:, intent(inout) data,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

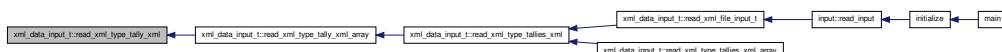
References `init_xml_type_tally_xml()`, `lurep_`, and `strict_`.

Referenced by `read_xml_type_tally_xml_array()`.

Here is the call graph for this function:



Here is the caller graph for this function:



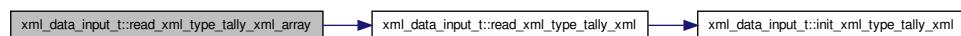
```
7.11.1.26 subroutine xml_data_input_t::read_xml_type_tally_xml_array ( type(XML_PARSE) info,
character(len=*),intent(inout) tag, logical,intent(inout)
endtag, character(len=*),dimension(:, :),intent(inout)
attribs, integer,intent(inout) noattribs,
character(len=*),dimension(:),intent(inout) data,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

Definition at line 803 of file input\_t.f90.

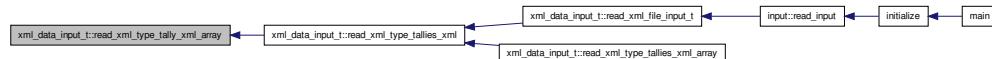
References `read_xml_type_tally_xml()`.

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:



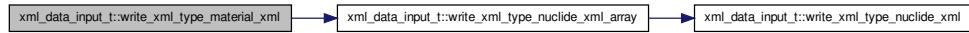
**7.11.1.28 subroutine `xml_data_input_t::write_xml_type_material_xml` (**  
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(material_xml) dvar )`

Definition at line 634 of file input\_t.f90.

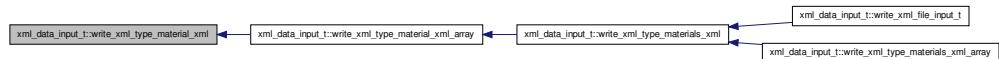
References `write_xml_type_nuclide_xml_array()`.

Referenced by `write_xml_type_material_xml_array()`.

Here is the call graph for this function:



Here is the caller graph for this function:



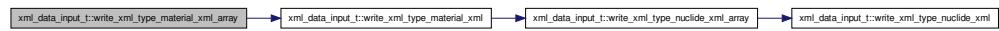
**7.11.1.29 subroutine `xml_data_input_t::write_xml_type_material_xml_array` (**  
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(material_xml),dimension(:) dvar )`

Definition at line 622 of file input\_t.f90.

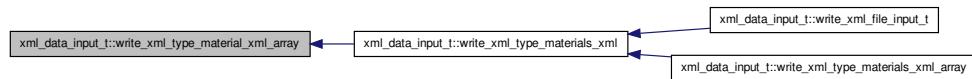
References `write_xml_type_material_xml()`.

Referenced by `write_xml_type_materials_xml()`.

Here is the call graph for this function:



Here is the caller graph for this function:



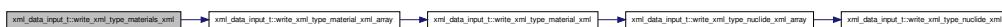
#### 7.11.1.30 subroutine `xml_data_input_t::write_xml_type_materials_xml` ( type(XML\_PARSE) *info*, character(len=\*)*intent(in)* *tag*, integer *indent*, type(materials\_xml) *dvar* )

Definition at line 788 of file input\_t.f90.

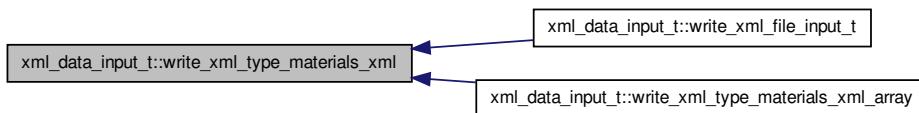
References `write_xml_type_material_xml_array()`.

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:



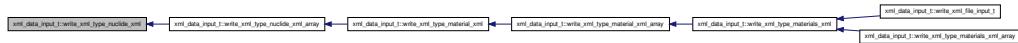
#### 7.11.1.32 subroutine `xml_data_input_t::write_xml_type_nuclide_xml` (

`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` (

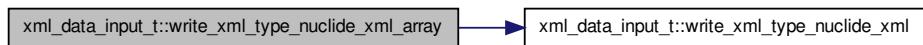
`type(XML_PARSE) info, character(len=*) intent(in) tag, integer indent, type(nuclide_xml), dimension(:) dvar )`

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:

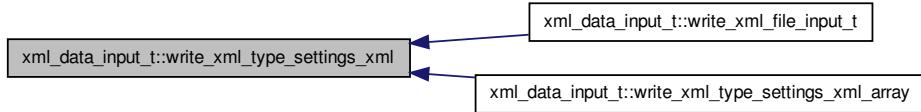


**7.11.1.34 subroutine `xml_data_input_t::write_xml_type_settings_xml`** (  
`type(XML_PARSE) info, character(len=*),intent(in) tag, integer indent, type(settings_xml) dvar` )

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:



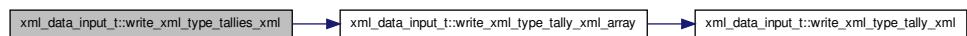
**7.11.1.36 subroutine `xml_data_input_t::write_xml_type_tallies_xml` (**  
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(tallies_xml) dvar )`

Definition at line 1132 of file input\_t.f90.

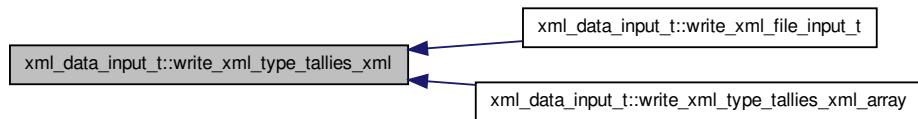
References `write_xml_type_tally_xml_array()`.

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:

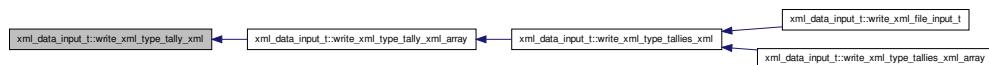


**7.11.1.38 subroutine `xml_data_input_t::write_xml_type_tally_xml` (**  
`type(XML_PARSE) info, character(len=*)intent(in) tag, integer indent, type(tally_xml) dvar )`

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.

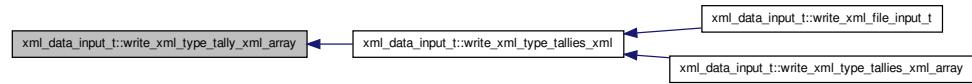
References `write_xml_type_tally_xml()`.

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( (/ '&', '&gt;', '>', '<', '&lt;', '/), (/2,3/))

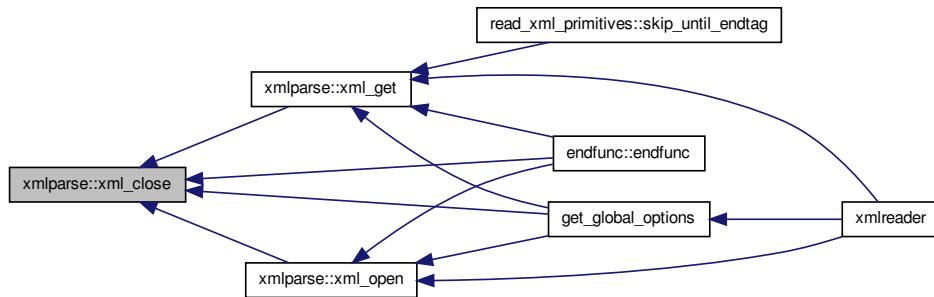
### 7.12.1 Function/Subroutine Documentation

#### 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* )

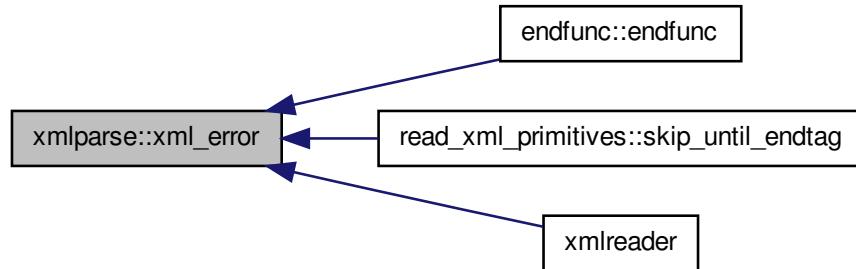
Definition at line 932 of file `xmlparse.f90`.

#### 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:

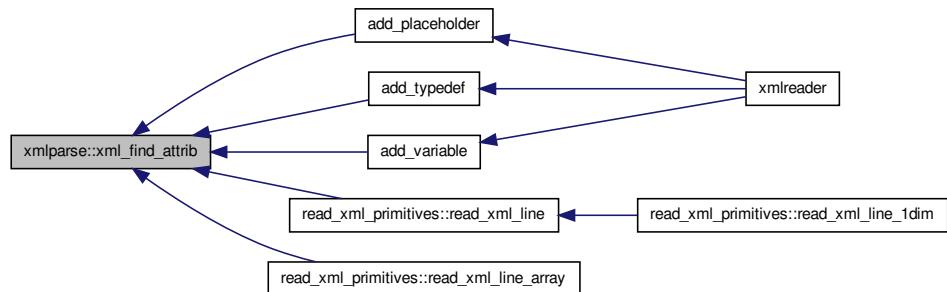


**7.12.1.4 integer `xmlparse::xml_find_attrib` ( `character(len=*)`,  
`dimension(:,::)`  
`attribs`, `integer no_attribs`, `character(len=*) name`, `character(len=*) value` )**

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.

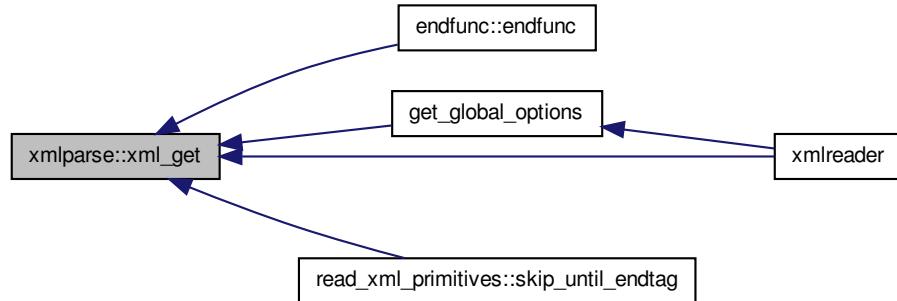
References `xml_close()`.

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:

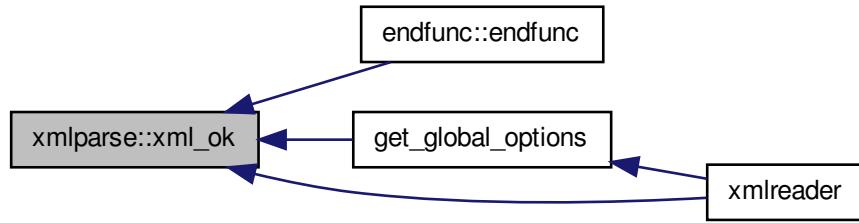


```
7.12.1.6 logical xmlparse::xml_ok ( type(XML_PARSE),intent(in) info )
```

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.

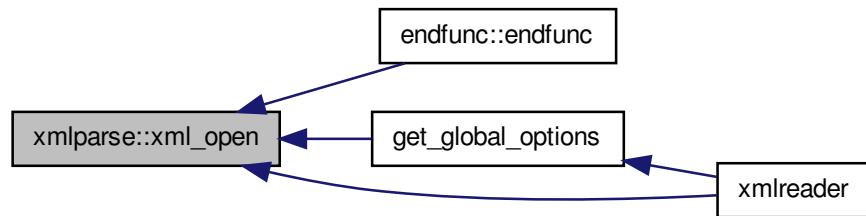
References xml\_close().

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:



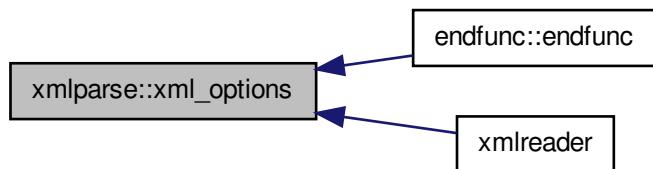
**7.12.1.8 subroutine `xmlparse::xml_options` ( type(XML\_PARSE),intent(inout) info, logical,intent(in),optional ignore\_whitespace, logical,intent(in),optional no\_data\_truncation, integer,intent(in),optional report\_lun, logical,intent(in),optional report\_errors, logical,intent(in),optional report\_details )**

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( (/ '&', '&gt;', '>', '<', '&lt;' /), (/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**

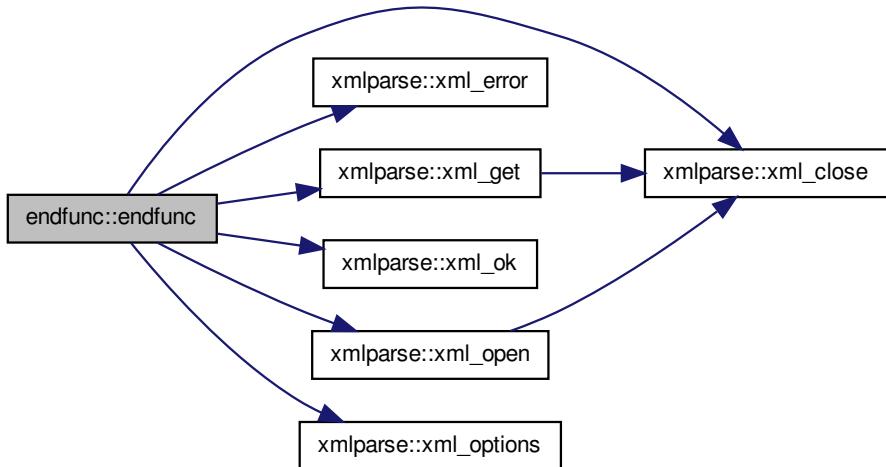
Definition at line 1003 of file xmlparse.f90.

**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:

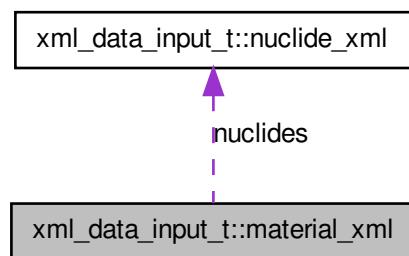


The documentation for this interface was generated from the following file:

- /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**

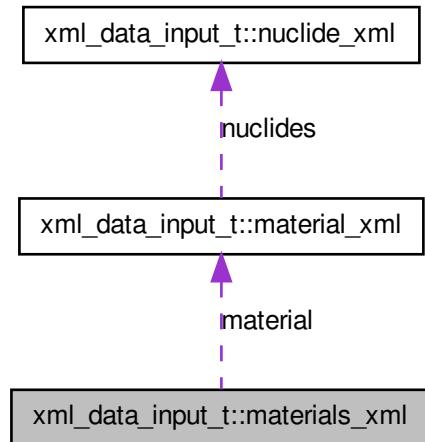
Definition at line 31 of file input\_t.f90.

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 )**

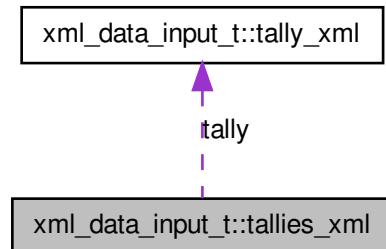
Definition at line 987 of file xmlparse.f90.

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( )

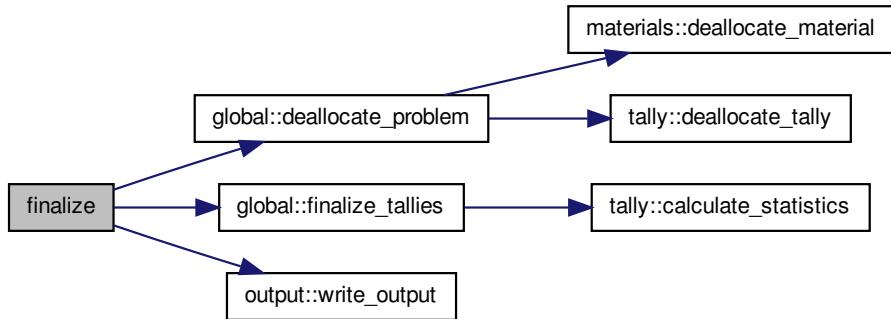
*routine that finalizes the problem*

Definition at line 157 of file main.f90.

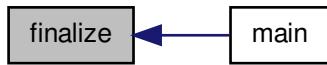
References [global::deallocate\\_problem\(\)](#), [global::finalize\\_tallies\(\)](#), and [output::write\\_output\(\)](#).

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( )

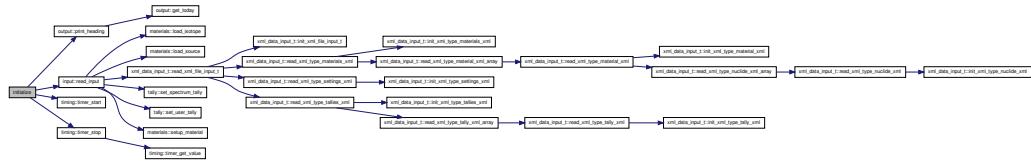
high level routine for initializing problem

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:



### 9.3.1.4 subroutine main::run\_problem( )

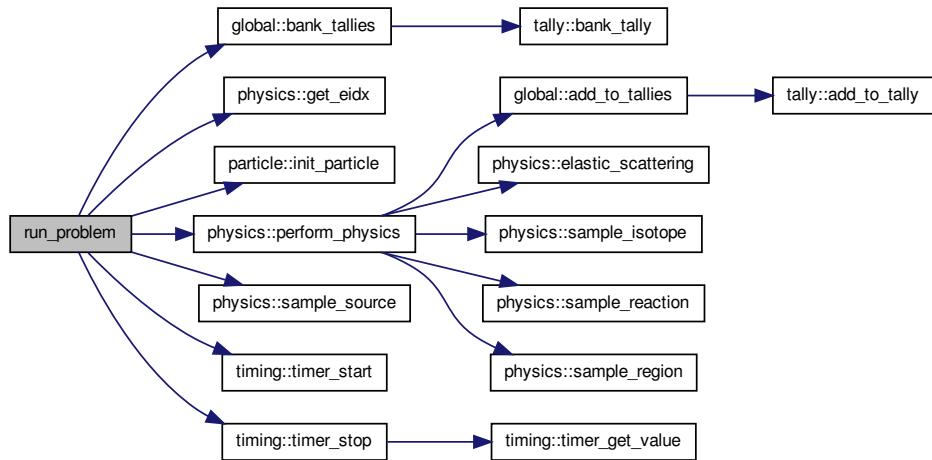
main routine for executing the transport calculation

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( (/ '&', '&amp;', '>', '&gt;', '<', '&lt;' /), (/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`

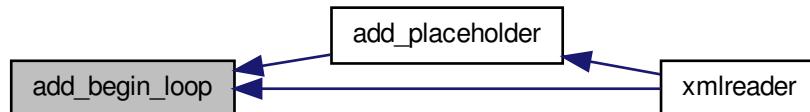
#### **9.14.1 Function Documentation**

##### **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:

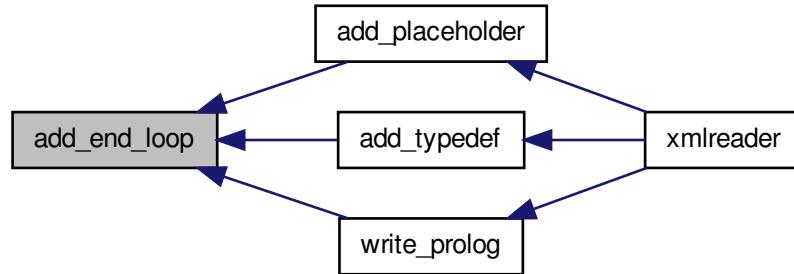


##### **9.14.1.2 subroutine xmlreader::add\_end\_loop ( )**

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:



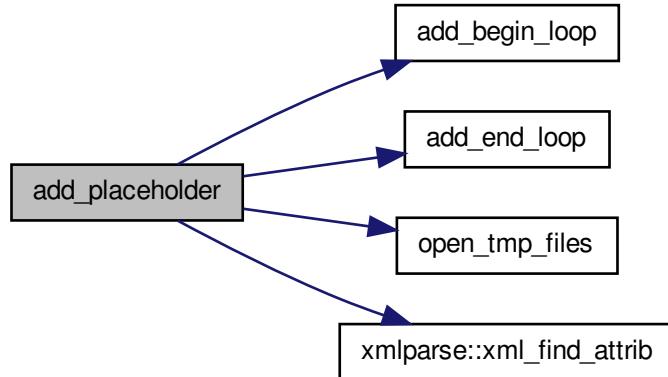
**9.14.1.3 subroutine xmlreader::add\_placeholder ( logical,intent(in) dyn\_strings )**

Definition at line 1131 of file xmlreader.f90.

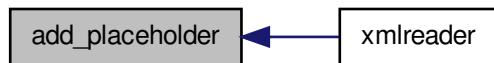
References add\_begin\_loop(), add\_end\_loop(), open\_tmp\_files(), and xmlparse::xml\_find\_attrib().

Referenced by xmlreader().

Here is the call graph for this function:



Here is the caller graph for this function:



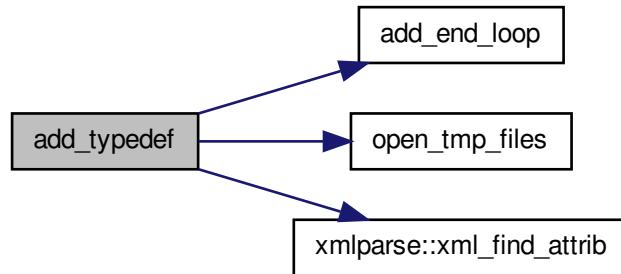
#### **9.14.1.4 subroutine `xmlreader::add_typedef` ( logical,intent(in) *dyn\_strings* )**

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

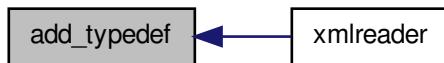
References `add_end_loop()`, `open_tmp_files()`, and `xmlparse::xml_find_attrib()`.

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.

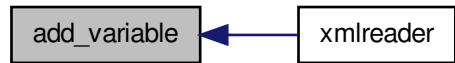
References `xmlparse::xml_find_attrib()`.

Referenced by `xmlreader()`.

Here is the call graph for this function:



Here is the caller graph for this function:

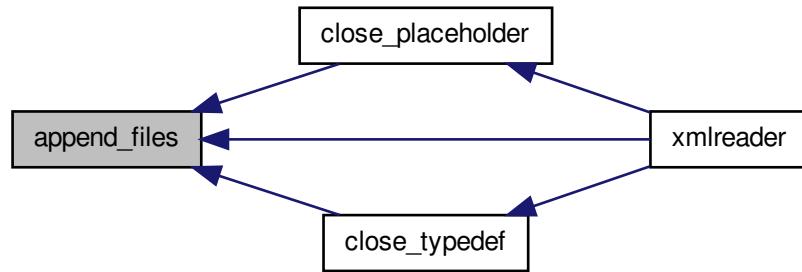


#### **9.14.1.6 subroutine xmlreader::append\_files ( integer,intent(in) *lufirst* )**

Definition at line 467 of file xmlreader.f90.

Referenced by close\_placeholder(), close\_typedef(), and xmlreader().

Here is the caller graph for this function:



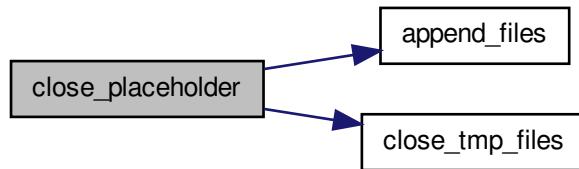
#### 9.14.1.7 subroutine xmlreader::close\_placeholder( )

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:

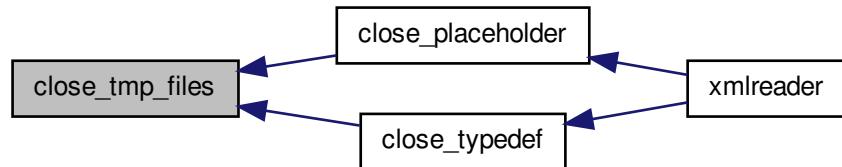


#### **9.14.1.8 subroutine xmlreader::close\_tmp\_files( )**

Definition at line 444 of file xmlreader.f90.

Referenced by close\_placeholder(), and close\_typedef().

Here is the caller graph for this function:



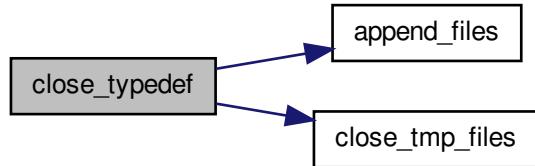
#### **9.14.1.9 subroutine xmlreader::close\_typedef( logical,intent(out) component )**

Definition at line 1105 of file xmlreader.f90.

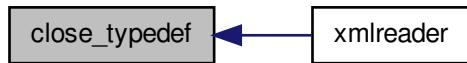
References append\_files(), and close\_tmp\_files().

Referenced by xmlreader().

Here is the call graph for this function:



Here is the caller graph for this function:



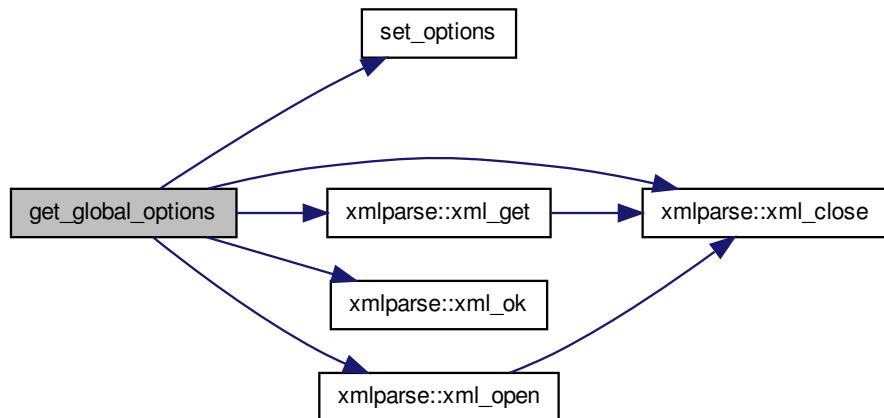
**9.14.1.10 subroutine xmlreader::get\_global\_options (**  
character(len=\*),dimension(:, :) intent(inout) *attribs*,  
integer,intent(inout) *noattribs*, logical,intent(inout) *strict*,  
logical,intent(inout) *global\_type*, character(len=\*),intent(inout)  
*global\_name*, character(len=\*),intent(inout) *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:

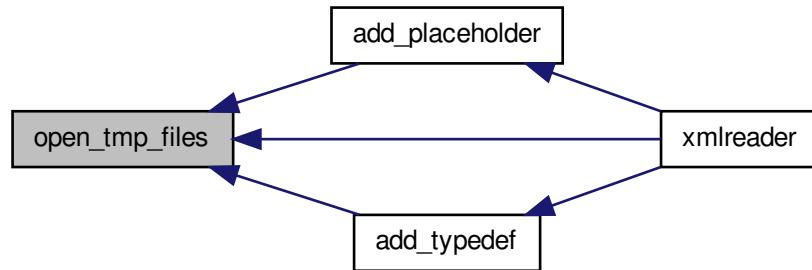


#### **9.14.1.12 subroutine xmlreader::open\_tmp\_files ( integer,intent(in) lufirst )**

Definition at line 420 of file xmlreader.f90.

Referenced by add\_placeholder(), add\_typedef(), and xmlreader().

Here is the caller graph for this function:

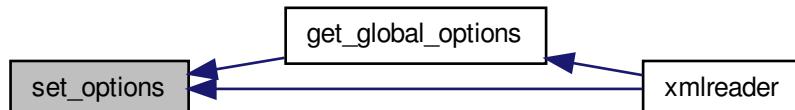


#### **9.14.1.13 subroutine xmlreader::set\_options ( character(len=\*),dimension(:, :) ,intent(in) attrs, integer,intent(in) noattrs, logical,intent(inout) strict, logical,intent(inout) global\_type, character(len=\*),intent(inout) global\_name, character(len=\*),intent(inout) root\_name, logical,intent(inout) 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.

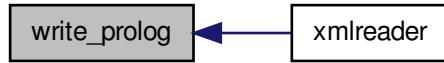
References add\_end\_loop().

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:

