MOHIDLagrangian

0.01

Generated by Doxygen 1.8.13

Contents

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

Chapter 2

Data Type Index

2.1 Class Hierarchy

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

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Module Documentation

5.1 about Module Reference

Module to print version, licence, preambles.

Functions/Subroutines

- subroutine, public printlicpreamble

  Birjukovs Canelas - MARETEC

Variables

- type(string) version
- type(string) author
- type(string) date

5.1.1 Detailed Description

Module to print version, licence, preambles.

Author

Ricardo Birjukovs Canelas

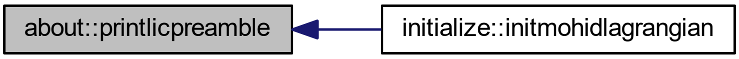
5.1.2 Function/Subroutine Documentation

5.1.2.1 printlicpreamble()

```
subroutine, public about::printlicpreamble ( )
```

Birjukovs Canelas - MARETEC

Public licence and preamble printer routine. Here is the caller graph for this function:



5.1.3 Variable Documentation

5.1.3.1 author

```
type(string) about::author  [private]
```

5.1.3.2 date

```
type(string) about::date  [private]
```

5.1.3.3 version

```
type(string) about::version  [private]
```

5.2 commom_modules Module Reference

Module to hold all of the commonly used base modules.

5.2.1 Detailed Description

Module to hold all of the commonly used base modules.

Author

Ricardo Birjukovs Canelas

5.3 finalize Module Reference

Module with the simulation closing related definitions and methods.

Functions/Subroutines

- subroutine simulation_end

  Birjukovs Canelas - MARETEC
- subroutine deallocate_simulation

  Birjukovs Canelas - MARETEC
- subroutine closelog

  Birjukovs Canelas - MARETEC
- subroutine, public finalizemohidlagrangian

  Birjukovs Canelas - MARETEC

5.3.1 Detailed Description

Module with the simulation closing related definitions and methods.

Author

Ricardo Birjukovs Canelas

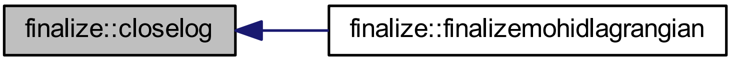
5.3.2 Function/Subroutine Documentation

5.3.2.1 closelog()

```
subroutine finalize::closelog ( )  [private]
```

Birjukovs Canelas - MARETEC

Private logfile closing routine. Here is the caller graph for this function:

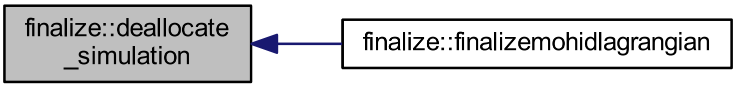


5.3.2.2 deallocate_simulation()

```
subroutine finalize::deallocate_simulation ( )  [private]
```

Birjukovs Canelas - MARETEC

Private globals deallocation routine. Here is the caller graph for this function:

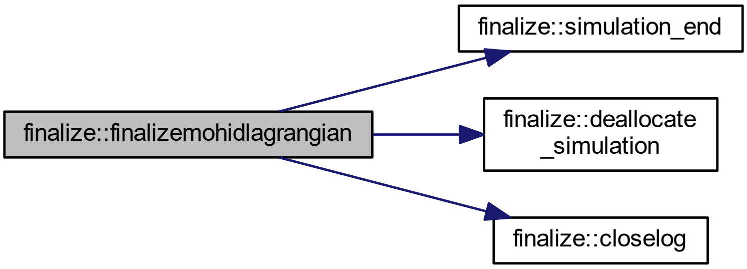


5.3.2.3 finalizemohidlagrangian()

```
subroutine, public finalize::finalizemohidlagrangian ( )
```

Birjukovs Canelas - MARETEC

Public finalization routine. Destroys, deallocates and closes the simulation space Here is the call graph for this function:

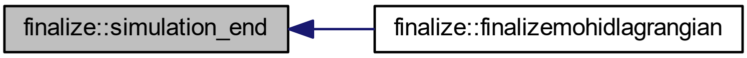


5.3.2.4 simulation_end()

```
subroutine finalize::simulation_end ( )  [private]
```

Birjukovs Canelas - MARETEC

Private closing statement routine. Here is the caller graph for this function:



5.4 geometry Module Reference

Module that defines geometry classes and related methods.

Data Types

- type box

  Type - point class.
- type line

  Type - line class.
- type point

  Type - point class.
- type shape

  Type - extendable shape class.
- type sphere

  Type - sphere class.

Functions/Subroutines

- subroutine, public allocategeomlist

  Birjukovs Canelas - MARETEC
- logical function, public isvalidgeom (geomname)

  Birjukovs Canelas - MARETEC
- subroutine getnp (self, np, dp)

  Birjukovs Canelas - MARETEC
- subroutine getpointdistribution (self, np, dp, ptlist)

  Birjukovs Canelas - MARETEC
- subroutine sphere_np_count (dp, r, np)

  Birjukovs Canelas - MARETEC
- subroutine sphere_grid (dp, r, np, ptlist)

  Birjukovs Canelas - MARETEC
- subroutine box_grid (dp, size, np, ptlist)

  Birjukovs Canelas - MARETEC
- subroutine line_grid (dp, dist, np, ptlist)

  Birjukovs Canelas - MARETEC

Variables

- type(string), dimension(:), allocatable, public geomlist

    String list (array) with the name of possible geometry types.

5.4.1 Detailed Description

Module that defines geometry classes and related methods.

Author

Ricardo Birjukovs Canelas

5.4.2 Function/Subroutine Documentation

5.4.2.1 allocategeomlist()

```
subroutine, public geometry::allocategeomlist ( )
```

Birjukovs Canelas - MARETEC

Public routine to allocate the possible geometry name list

5.4.2.2 box_grid()

```
subroutine geometry::box_grid (
             real(prec), intent(in) dp,
             type(vector), intent(in) size,
             integer, intent(in) np,
             type(vector), dimension(np), intent(out) ptlist )  [private]
```

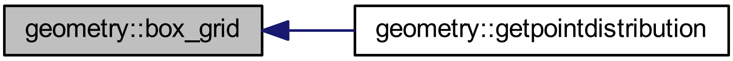
Birjukovs Canelas - MARETEC

private routine that returns the points distributed on a grid with spacing dp inside a box

Parameters

in	*dp,size,np,ptlist*	

Here is the caller graph for this function:



5.4.2.3 getnp()

```
subroutine geometry::getnp (
            class(shape) self,
            integer, intent(out) np,
            real(prec), intent(in) dp )  [private]
```

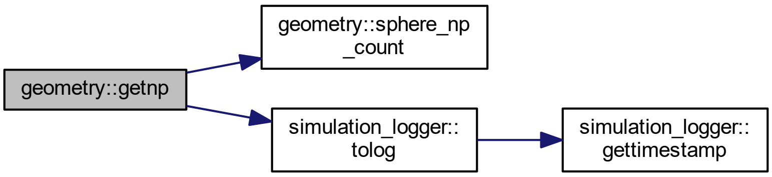
Birjukovs Canelas - MARETEC

method to get the number of points that fill a given geometry

Parameters

in	*self,np*	

Here is the call graph for this function:



5.4.2.4 getpointdistribution()

```
subroutine geometry::getpointdistribution (
            class(shape) self,
            integer, intent(in) np,
            real(prec), intent(in) dp,
            type(vector), dimension(np), intent(inout) ptlist )  [private]
```

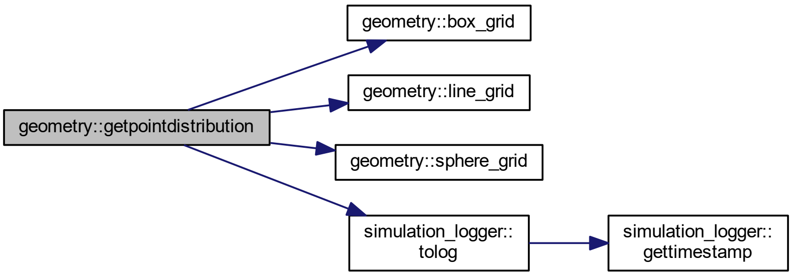
Birjukovs Canelas - MARETEC

method to get the number of points that fill a given geometry

Parameters

in	*self,np*	

Here is the call graph for this function:



5.4.2.5 isvalidgeom()

```
logical function, public geometry::isvalidgeom (
            type(string), intent(in) geomname )
```

Birjukovs Canelas - MARETEC

Public function that returns a logical if the input geometry name is valid

Parameters

in	*geomname*	

5.4.2.6 line_grid()

```
subroutine geometry::line_grid (
            real(prec), intent(in) dp,
            type(vector), intent(in) dist,
            integer, intent(in) np,
            type(vector), dimension(np), intent(out) ptlist )  [private]
```

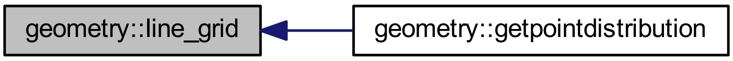
Birjukovs Canelas - MARETEC

private routine that returns the points distributed on a grid with spacing dp along a line

Parameters

in	*dp,size,np,ptlist*	

Here is the caller graph for this function:



5.4.2.7 sphere_grid()

```
subroutine geometry::sphere_grid (
            real(prec), intent(in) dp,
            real(prec), intent(in) r,
            integer, intent(in) np,
            type(vector), dimension(np), intent(out) ptlist )  [private]
```

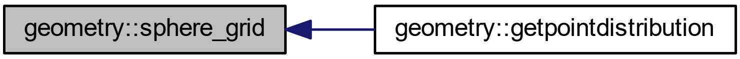
Birjukovs Canelas - MARETEC

private routine that returns the points distributed on a grid with spacing dp inside a sphere

Parameters

in	*dp,r,np,ptlist*	

Here is the caller graph for this function:



5.4.2.8 sphere_np_count()

```
subroutine geometry::sphere_np_count (
            real(prec), intent(in) dp,
            real(prec), intent(in) r,
            integer, intent(out) np )  [private]
```

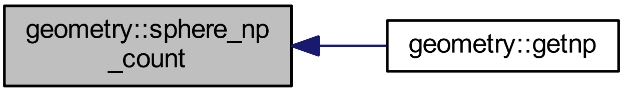
Birjukovs Canelas - MARETEC

private routine that returns the number of points distributed on a grid with spacing dp inside a sphere

Parameters

in	*dp,r,np*	

Here is the caller graph for this function:



5.4.3 Variable Documentation

5.4.3.1 geomlist

```
type(string), dimension(:), allocatable, public geometry::geomlist
```

String list (array) with the name of possible geometry types.

5.5 initialize Module Reference

Module with the simulation initialization related definitions and methods. Has one public access routine that is incharge of building the simulation space from input files.

Functions/Subroutines

- subroutine linkpropertysources (linksNode)

  Birjukovs Canelas - MARETEC
- subroutine init_properties (case_node)

  Birjukovs Canelas - MARETEC
- subroutine read_xml_geometry (source, source_detail, geometry)

  Birjukovs Canelas - MARETEC
- subroutine init_sources (case_node)

  Birjukovs Canelas - MARETEC
- subroutine init_simdefs (case_node)

  Birjukovs Canelas - MARETEC
- subroutine init_caseconstants (case_node)

  Birjukovs Canelas - MARETEC
- subroutine init_parameters (execution_node)

  Birjukovs Canelas - MARETEC
- subroutine, public initmohidlagrangian (xmlfilename)

  Birjukovs Canelas - MARETEC

5.5.1 Detailed Description

Module with the simulation initialization related definitions and methods. Has one public access routine that is incharge of building the simulation space from input files.

Author

Ricardo Birjukovs Canelas

5.5.2 Function/Subroutine Documentation

5.5.2.1 init_caseconstants()

```
subroutine initialize::init_caseconstants (
            type(node), intent(in), pointer case_node )  [private]
```

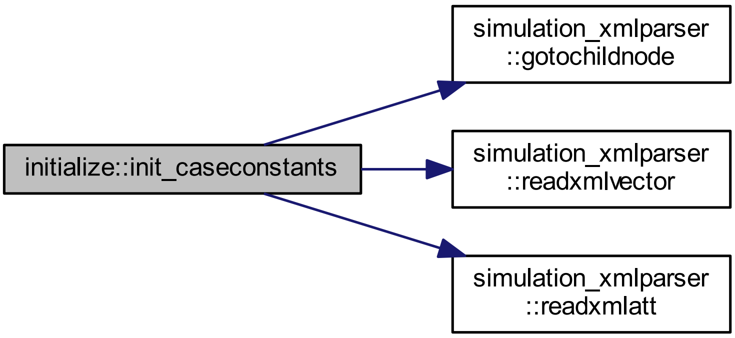
Birjukovs Canelas - MARETEC

Private case constant parser routine. Builds the simulation parametric space from the input xml case file.

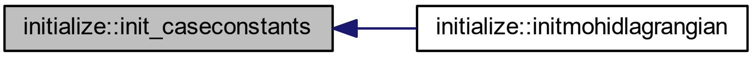
Parameters

in	*parsedxml*	

Here is the call graph for this function:


initialize::init_caseconstants
simulation_xmlparser ::gotochildnode
simulation_xmlparser ::readxmlvector
simulation_xmlparser ::readxmlatt

Here is the caller graph for this function:


initialize::init_caseconstants
initialize::initmohidlagrangian

5.5.2.2 init_parameters()

```
subroutine initialize::init_parameters (
            type(node), intent(in), pointer execution_node )  [private]
```

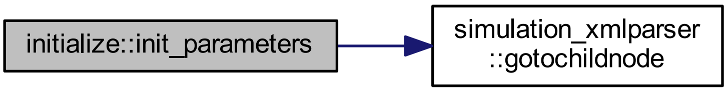
Birjukovs Canelas - MARETEC

Private parameter parser routine. Builds the simulation parametric space from the input xml case file.

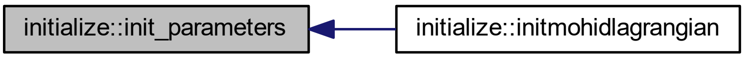
Parameters

in	*parsedxml*	

Here is the call graph for this function:


initialize::init_parameters
simulation_xmlparser ::gotochildnode

Here is the caller graph for this function:



5.5.2.3 init_properties()

```
subroutine initialize::init_properties (
            type(node), intent(in), pointer case_node )  [private]
```

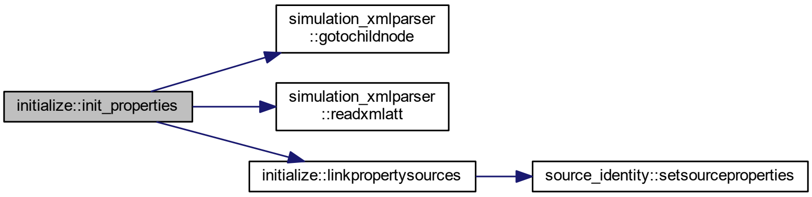
Birjukovs Canelas - MARETEC

Private property xml parser routine. Reads the properties tab from the xml file and links these to the corresponding source

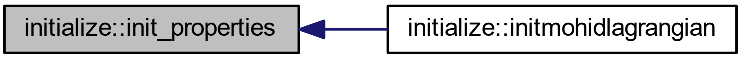
Parameters

in	*parsedxml*	

Here is the call graph for this function:


initialize::init_properties
simulation_xmlparser ::gotochildnode
simulation_xmlparser ::readxmlatt
initialize::linkpropertysources
source_identity::setsourceproperties

Here is the caller graph for this function:



5.5.2.4 init_simdefs()

```
subroutine initialize::init_simdefs (
            type(node), intent(in), pointer case_node )  [private]
```

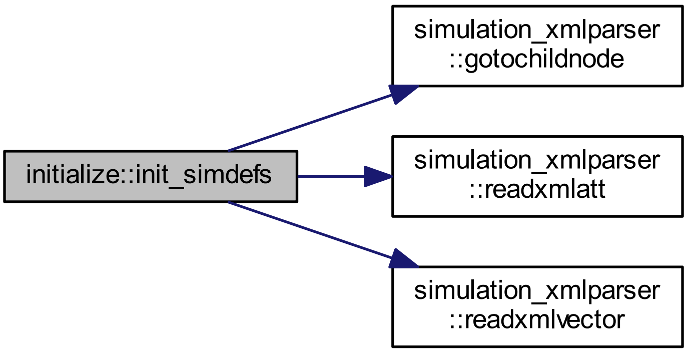
Birjukovs Canelas - MARETEC

Private simulation definitions parser routine. Builds the simulation geometric space from the input xml case file.

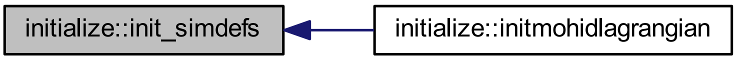
Parameters

in	*parsedxml*	

Here is the call graph for this function:


initialize::init_simdefs
simulation_xmlparser ::gotochildnode
simulation_xmlparser ::readxmlatt
simulation_xmlparser ::readxmlvector

Here is the caller graph for this function:



5.5.2.5 init_sources()

```
subroutine initialize::init_sources (
            type(node), intent(in), pointer case_node )  [private]
```

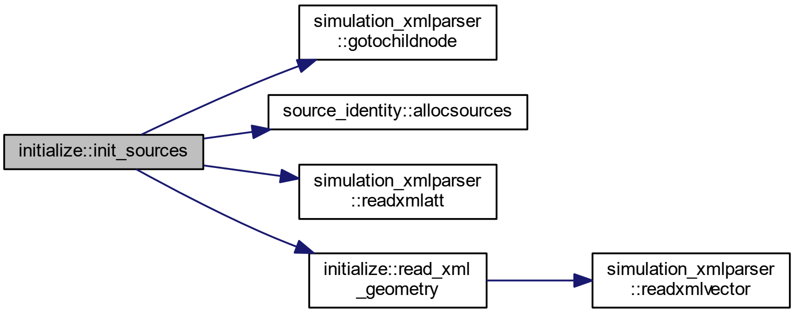
Birjukovs Canelas - MARETEC

Private source definitions parser routine. Builds the tracer sources from the input xml case file.

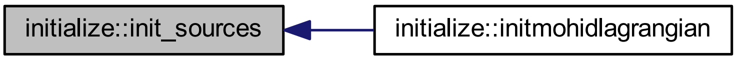
Parameters

in	*parsedxml*	

Here is the call graph for this function:


simulation_xmlparser ::gotochildnode
source_identity::allocsources
initialize::init_sources
simulation_xmlparser ::readxmlatt
initialize::read_xml _geometry
simulation_xmlparser ::readxmlvector

Here is the caller graph for this function:



5.5.2.6 initmohidlagrangian()

```
subroutine, public initialize::initmohidlagrangian (
            type(string), intent(in) xmlfilename )
```

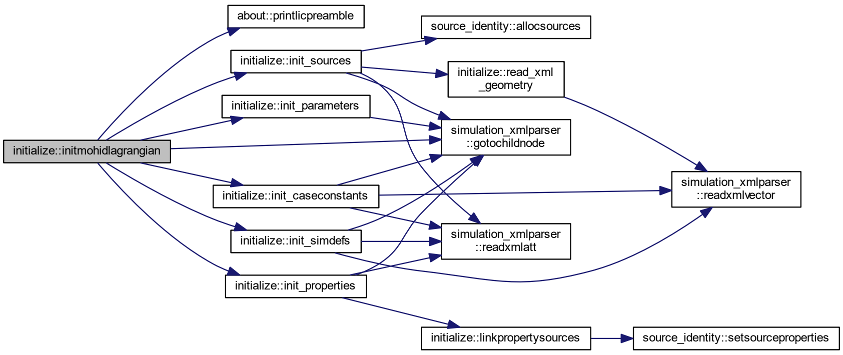
Birjukovs Canelas - MARETEC

Public xml parser routine. Builds the simulation space from the input xml case file.

Parameters

in	*xmlfilename*	
in	*xmlfilename*	.xml file name

Here is the call graph for this function:


about::printlicpreamble
source_identity::allocsources
initialize::init_sources
initialize::read_xml _geometry
initialize::init_parameters
simulation_xmlparser ::gotochildnode
initialize::initmohidlagrangian
initialize::init_caseconstants
simulation_xmlparser ::readxmlvector
initialize::init_simdefs
simulation_xmlparser ::readxmlatt
initialize::init_properties
initialize::linkpropertysources
source_identity::setsourceproperties

5.5.2.7 linkpropertysources()

```
subroutine initialize::linkpropertysources (
            type(node), intent(in), pointer linksNode ) [private]
```

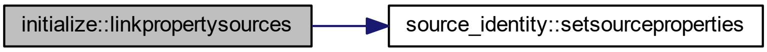
Birjukovs Canelas - MARETEC

Private property xml parser routine. Reads the properties tab from the xml file and links these to the corresponding source

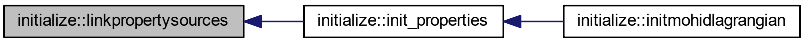
Parameters

in	*parsedxml*	

Here is the call graph for this function:


initialize::linkpropertysources
source_identity::setsourceproperties

Here is the caller graph for this function:


initialize::linkpropertysources
initialize::init_properties
initialize::initmohidlagrangian

5.5.2.8 read_xml_geometry()

```
subroutine initialize::read_xml_geometry (
            type(node), intent(in), pointer source,
            type(node), intent(in), pointer source_detail,
            class(shape), intent(inout) geometry )  [private]
```

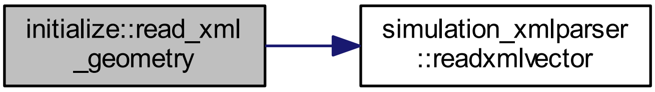
Birjukovs Canelas - MARETEC

Private geometry xml parser routine. Reads a geometry from the xml depending on the geometry type of the node

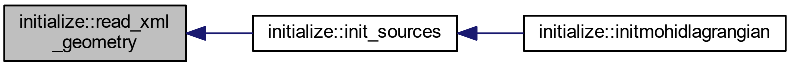
Parameters

in	*source,geometry*	
in	*source*	Working xml node
in	*source_detail*	Working xml node details
in,out	*geometry*	Geometrical object to fill

Here is the call graph for this function:


initialize::read_xml_geometry
simulation_xmlparser::readxmlvector

Here is the caller graph for this function:


initialize::read_xml_geometry
initialize::init_sources
initialize::initmohidlagrangian

5.6 simulation_globals Module Reference

Module to hold the simulation global parameter classes and their methods.

Data Types

- type constants_t

  Case Constants class.
- type filenames_t

  File names class.
- type parameters_t
- type simdefs_t

  Simulation definitions class.

Functions/Subroutines

- subroutine setparameter (self, parmkey, parmvalue)

  Birjukovs Canelas - MARETEC
- subroutine check (self)

  Birjukovs Canelas - MARETEC
- subroutine printsimparameters (self)

  Birjukovs Canelas - MARETEC
- subroutine getintegratorname (name, code)

  Birjukovs Canelas - MARETEC
- subroutine setgravity (self, grav)

  Birjukovs Canelas - MARETEC

- subroutine setrho (self, read_rho)

  Birjukovs Canelas - MARETEC
- subroutine setdp (self, read_dp)

  Birjukovs Canelas - MARETEC
- subroutine setdt (self, read_dt)

  Birjukovs Canelas - MARETEC
- subroutine setboundingbox (self, point_, coords)

  Birjukovs Canelas - MARETEC
- subroutine printsimdefs (self)

  Birjukovs Canelas - MARETEC

Variables

- real(prec_time), public simtime
- type(parameters_t), public parameters
- type(simdefs_t), public simdefs
- type(constants_t), public constants
- type(filenames_t), public filenames

5.6.1 Detailed Description

Module to hold the simulation global parameter classes and their methods.

Author

Ricardo Birjukovs Canelas

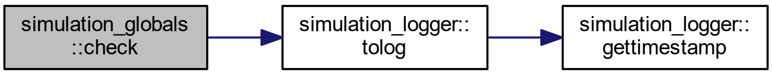
5.6.2 Function/Subroutine Documentation

5.6.2.1 check()

```
subroutine simulation_globals::check (
             class(parameters_t), intent(inout) self )  [private]
```

Birjukovs Canelas - MARETEC

Private parameter checking method. Checks if mandatory parameters were set Here is the call graph for this function:

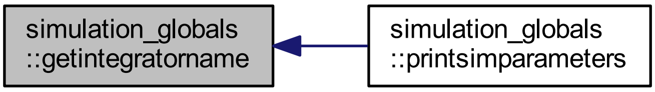

simulation_globals ::check
simulation_logger:: tolog
simulation_logger:: gettimestamp

5.6.2.2 getintegratorname()

```
subroutine simulation_globals::getintegratorname (
            type(string), intent(inout) name,
            integer, intent(in) code )  [private]
```

Birjukovs Canelas - MARETEC

private routine to get integrator scheme name Here is the caller graph for this function:

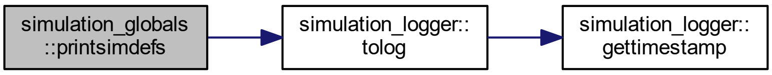

simulation_globals ::getintegratorname
simulation_globals ::printsimparameters

5.6.2.3 printsimdefs()

```
subroutine simulation_globals::printsimdefs (
            class(simdefs_t), intent(in) self )  [private]
```

Birjukovs Canelas - MARETEC

Public simulation definitions printing routine. Here is the call graph for this function:

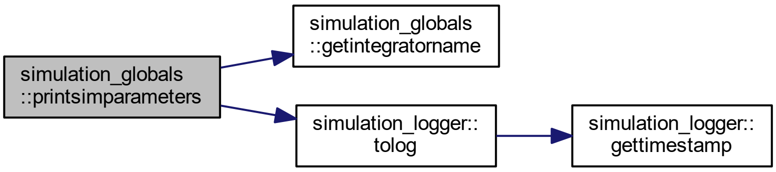

simulation_globals ::printsimdefs
simulation_logger:: tolog
simulation_logger:: gettimestamp

5.6.2.4 printsimparameters()

```
subroutine simulation_globals::printsimparameters (
            class(parameters_t), intent(inout) self )  [private]
```

Birjukovs Canelas - MARETEC

Private parameter printing method. Here is the call graph for this function:



5.6.2.5 setboundingbox()

```
subroutine simulation_globals::setboundingbox (
            class(simdefs_t), intent(inout) self,
            type(string), intent(in) point_,
            type(vector) coords )  [private]
```

Birjukovs Canelas - MARETEC

Private bounding box setting routine.

Parameters

in	*point_,coords*	

5.6.2.6 setdp()

```
subroutine simulation_globals::setdp (
            class(simdefs_t), intent(inout) self,
            type(string), intent(in) read_dp )  [private]
```

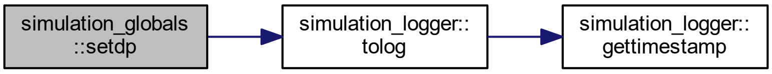
Birjukovs Canelas - MARETEC

Private dp setting routine.

Parameters

in	*read_dp*	

Here is the call graph for this function:


simulation_globals ::setdp
simulation_logger:: tolog
simulation_logger:: gettimestamp

5.6.2.7 setdt()

```
subroutine simulation_globals::setdt (
            class(simdefs_t), intent(inout) self,
            type(string), intent(in) read_dt )  [private]
```

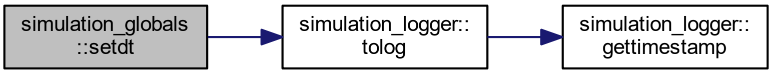
Birjukovs Canelas - MARETEC

Private dt setting routine.

Parameters

in	*read↩ _dt*	

Here is the call graph for this function:


simulation_globals ::setdt
simulation_logger:: tolog
simulation_logger:: gettimestamp

5.6.2.8 setgravity()

```
subroutine simulation_globals::setgravity (
            class(constants_t), intent(inout) self,
            type(vector) grav )  [private]
```

Birjukovs Canelas - MARETEC

Public Gravity setting routine.

Parameters

in	*grav*	

5.6.2.9 setparameter()

```
subroutine simulation_globals::setparameter (
            class(parameters_t), intent(inout) self,
            type(string), intent(in) parmkey,
            type(string), intent(in) parmvalue )  [private]
```

Birjukovs Canelas - MARETEC

Private parameter setting method. Builds the simulation parametric space from the input case file.

Parameters

in	*parmkey,parmvalue*	

5.6.2.10 setrho()

```
subroutine simulation_globals::setrho (
            class(constants_t), intent(inout) self,
            type(string), intent(in) read_rho )  [private]
```

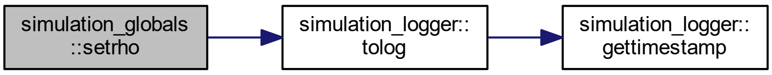
Birjukovs Canelas - MARETEC

Provate Rho_Ref setting routine.

Parameters

in	*read_rho*	

Here is the call graph for this function:


simulation_globals ::setrho
simulation_logger:: tolog
simulation_logger:: gettimestamp

5.6.3 Variable Documentation

5.6.3.1 constants

`type(constants_t), public simulation_globals::constants`

5.6.3.2 filenames

`type(filenames_t), public simulation_globals::filenames`

5.6.3.3 parameters

`type(parameters_t), public simulation_globals::parameters`

5.6.3.4 simdefs

`type(simdefs_t), public simulation_globals::simdefs`

5.6.3.5 simtime

`real(prec_time), public simulation_globals::simtime`

5.7 simulation_logger Module Reference

Module to hold all the simulation logger related definitions and methods.

Functions/Subroutines

- subroutine, public gettimestamp (timestamp)
  Birjukovs Canelas - MARETEC
- subroutine, public tolog (tologstr, timeoption)
  Birjukovs Canelas - MARETEC
- subroutine, public initmohidlagrangianlog (outpath)
  Birjukovs Canelas - MARETEC

Variables

- integer, public log_unit = -1
  'Number' of log file

5.7.1 Detailed Description

Module to hold all the simulation logger related definitions and methods.

Author

Ricardo Birjukovs Canelas

5.7.2 Function/Subroutine Documentation

5.7.2.1 gettimestamp()

```
subroutine, public simulation_logger::gettimestamp (
            type(string), intent(out) timestamp )
```

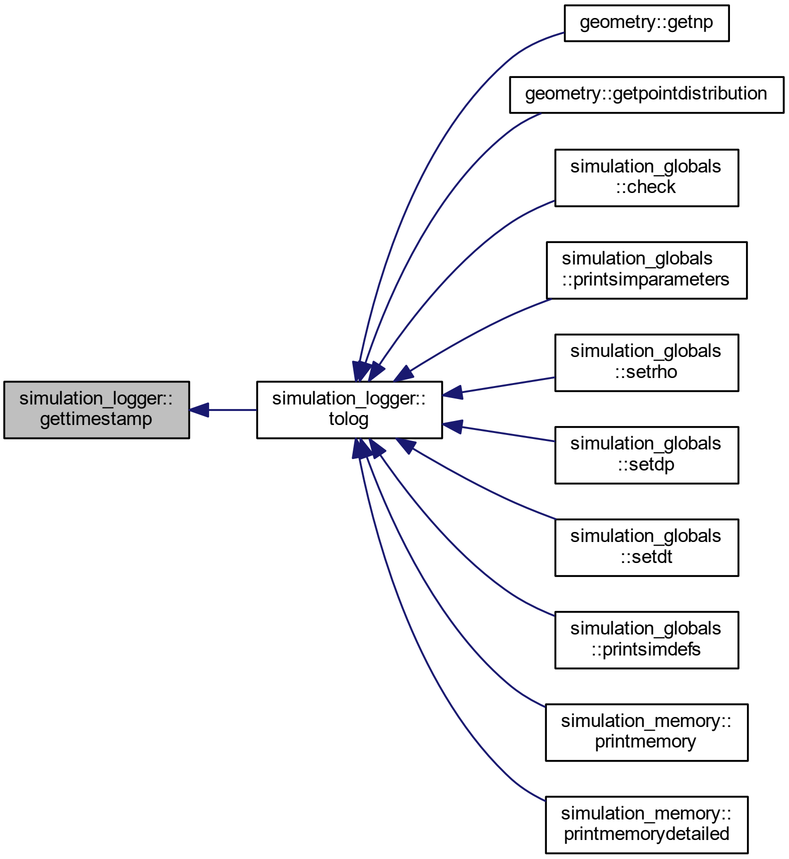
Birjukovs Canelas - MARETEC

Public timestamp builder

Parameters

in	*timestamp*	

Here is the caller graph for this function:



5.7.2.2 initmohidlagrangianlog()

```
subroutine, public simulation_logger::initmohidlagrangianlog (
            type(string), intent(in) outpath )
```

Birjukovs Canelas - MARETEC

Public log file initizalization routine.

Parameters

in	*outpath*	

5.7.2.3 tolog()

```
subroutine, public simulation_logger::tolog (
             type(string), intent(in) tologstr,
             logical, intent(in), optional timeoption )
```

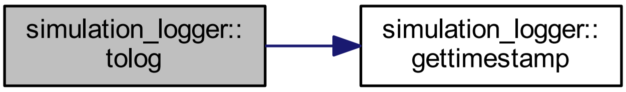
Birjukovs Canelas - MARETEC

Public log serialization routine

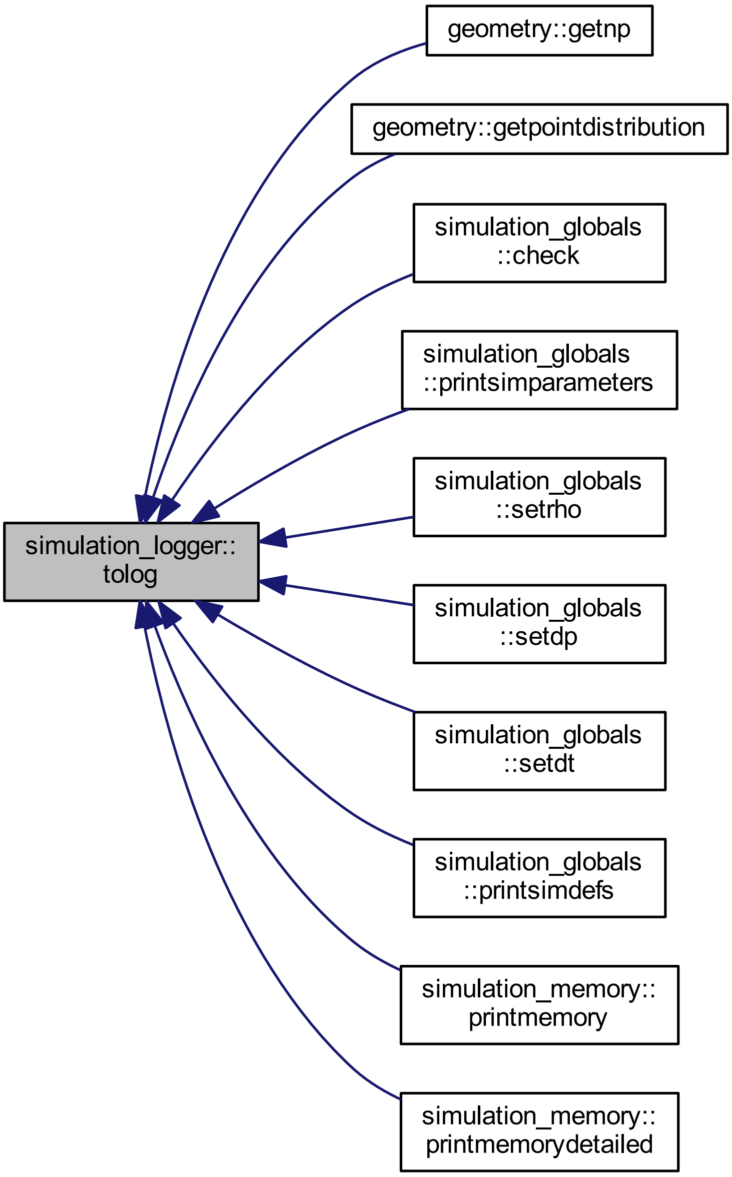
Parameters

in	*outpath*	

Here is the call graph for this function:



Here is the caller graph for this function:


geometry::getnp
geometry::getpointdistribution
simulation_globals ::check
simulation_globals ::printsimparameters
simulation_globals ::setrho
simulation_logger:: tolog
simulation_globals ::setdp
simulation_globals ::setdt
simulation_globals ::printsimdefs
simulation_memory:: printmemory
simulation_memory:: printmemorydetailed

5.7.3 Variable Documentation

5.7.3.1 log_unit

```
integer, public simulation_logger::log_unit = -1
```

'Number' of log file

5.8 simulation_memory Module Reference

Module to hold the simulation memory managment class and its methods.

Data Types

- type memory_t

Functions/Subroutines

- subroutine initialize (self)

  Birjukovs Canelas - MARETEC
- subroutine getotal (self, size)

  Birjukovs Canelas - MARETEC
- subroutine addsource (self, size)

  Birjukovs Canelas - MARETEC
- subroutine addtracer (self, size)

  Birjukovs Canelas - MARETEC
- subroutine adddef (self, size)

  Birjukovs Canelas - MARETEC
- subroutine printmemory (self)

  Birjukovs Canelas - MARETEC
- subroutine printmemorydetailed (self)

  Birjukovs Canelas - MARETEC

Variables

- type(memory_t), public simmemory

5.8.1 Detailed Description

Module to hold the simulation memory managment class and its methods.

Author

Ricardo Birjukovs Canelas

5.8.2 Function/Subroutine Documentation

5.8.2.1 adddef()

```
subroutine simulation_memory::adddef (
            class(memory_t), intent(inout) self,
            integer, intent(in) size )  [private]
```

Birjukovs Canelas - MARETEC

Private method to add the size of a definition to the memory log.

5.8.2.2 addsource()

```
subroutine simulation_memory::addsource (
            class(memory_t), intent(inout) self,
            integer, intent(in) size )  [private]
```

Birjukovs Canelas - MARETEC

Private method to add the size of a Source to the memory log.

5.8.2.3 addtracer()

```
subroutine simulation_memory::addtracer (
            class(memory_t), intent(inout) self,
            integer, intent(in) size )  [private]
```

Birjukovs Canelas - MARETEC

Private method to add the size of a Tracer to the memory log.

5.8.2.4 getotal()

```
subroutine simulation_memory::getotal (
            class(memory_t), intent(inout) self,
            integer, intent(out) size )  [private]
```

Birjukovs Canelas - MARETEC

Private method to retreive the total size of the allocated memory.

5.8.2.5 initialize()

```
subroutine simulation_memory::initialize (
            class(memory_t), intent(inout) self )  [private]
```

Birjukovs Canelas - MARETEC

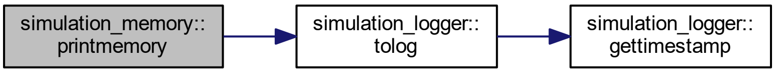
Private memory logger initialization method.

5.8.2.6 printmemory()

```
subroutine simulation_memory::printmemory (
            class(memory_t), intent(inout) self )  [private]
```

Birjukovs Canelas - MARETEC

Private method to print the allocated memory. Here is the call graph for this function:

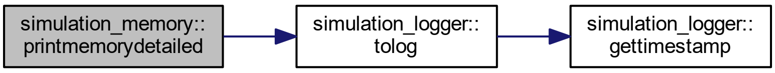


5.8.2.7 printmemorydetailed()

```
subroutine simulation_memory::printmemorydetailed (
            class(memory_t), intent(inout) self )  [private]
```

Birjukovs Canelas - MARETEC

Private method to print the allocated memory. Here is the call graph for this function:



5.8.3 Variable Documentation

5.8.3.1 simmemory

```
type(memory_t), public simulation_memory::simmemory
```

5.9 simulation_precision Module Reference

Module to control the precision of the variables trough the project.

Variables

- integer, parameter sp = kind(1._R4P)

  Simple precision definition switch.
- integer, parameter dp = kind(1._R8P)

  Double precision definition switch.
- integer, parameter, public prec = sp
- integer, parameter, public prec_time = sp
- integer, parameter, public prec_wrt = sp
- real(prec), parameter, public missing_value_default = -9999.0_dp
- real(prec), parameter, public mv = MISSING_VALUE_DEFAULT
- real(prec), parameter, public mv_int = int(MISSING_VALUE_DEFAULT)
- real(prec), parameter, public err_dist = 1E8_dp
- integer, parameter, public err_ind = -1

5.9.1 Detailed Description

Module to control the precision of the variables trough the project.

Author

Ricardo Birjukovs Canelas

5.9.2 Variable Documentation

5.9.2.1 dp

```
integer, parameter simulation_precision::dp = kind(1._R8P)  [private]
```

Double precision definition switch.

5.9.2.2 err_dist

```
real(prec), parameter, public simulation_precision::err_dist = 1E8_dp
```

5.9.2.3 err_ind

```
integer, parameter, public simulation_precision::err_ind = -1
```

5.9.2.4 missing_value_default

```
real(prec), parameter, public simulation_precision::missing_value_default = -9999.0_dp
```

5.9.2.5 mv

```
real(prec), parameter, public simulation_precision::mv = MISSING_VALUE_DEFAULT
```

5.9.2.6 mv_int

```
real(prec), parameter, public simulation_precision::mv_int = int(MISSING_VALUE_DEFAULT)
```

5.9.2.7 prec

```
integer, parameter, public simulation_precision::prec = sp
```

5.9.2.8 prec_time

```
integer, parameter, public simulation_precision::prec_time = sp
```

5.9.2.9 prec_wrt

```
integer, parameter, public simulation_precision::prec_wrt = sp
```

5.9.2.10 sp

```
integer, parameter simulation_precision::sp = kind(1._R4P)  [private]
```

Simple precision definition switch.

5.10 simulation_xmlparser Module Reference

Module with the simulation xml parsing related definitions and routines.

Functions/Subroutines

- subroutine, public readxmlatt (xmlnode, tag, att_name, att_value, mandatory)

  Birjukovs Canelas - MARETEC
- subroutine, public readxmlvector (xmlnode, tag, vec, mandatory)

  Birjukovs Canelas - MARETEC
- subroutine, public gotochildnode (currentNode, targetNode, targetNodeName, mandatory)

  Birjukovs Canelas - MARETEC

5.10.1 Detailed Description

Module with the simulation xml parsing related definitions and routines.

Author

Ricardo Birjukovs Canelas

5.10.2 Function/Subroutine Documentation

5.10.2.1 gotochildnode()

```
subroutine, public simulation_xmlparser::gotochildnode (
             type(node), intent(in), pointer currentNode,
             type(node), intent(out), pointer targetNode,
             type(string), intent(in) targetNodeName,
             logical, intent(in), optional mandatory )
```

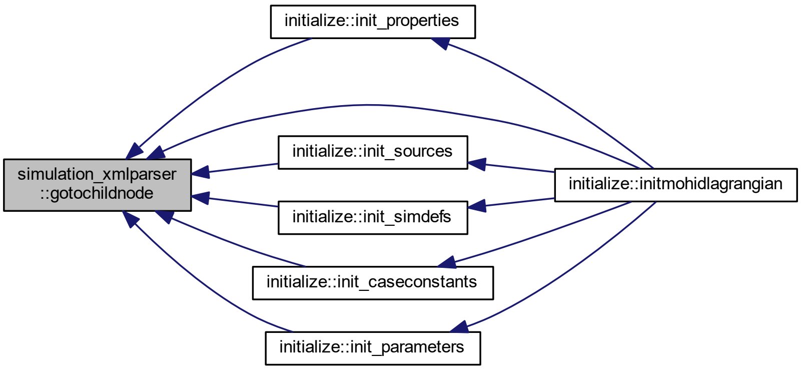
Birjukovs Canelas - MARETEC

Private routine to retrieve a node within a node. Returns a nullifyed pointer if not found, stops if mandatory.

Parameters

in	*currentNode,targetNode,targetNodeName,mandatory*	
in	*mandatory*	Swich for optional or mandatory tags

Here is the caller graph for this function:


initialize::init_properties
simulation_xmlparser ::gotochildnode
initialize::init_sources
initialize::initmohidlagrangian
initialize::init_simdefs
initialize::init_caseconstants
initialize::init_parameters

5.10.2.2 readxmlatt()

```
subroutine, public simulation_xmlparser::readxmlatt (
            type(node), intent(in), pointer xmlnode,
            type(string), intent(in) tag,
            type(string), intent(in) att_name,
            type(string), intent(out) att_value,
            logical, intent(in), optional mandatory )
```

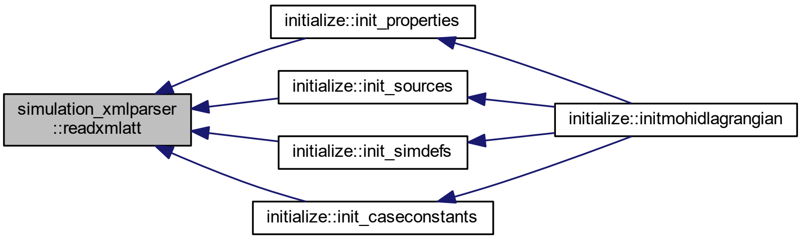
Birjukovs Canelas - MARETEC

Private attribute xml parser routine. In the format <Tag att_name="att_value"

Parameters

in	*xmlnode,tag,vec,mandatory*	
in	*xmlnode*	Working xml node
in	*tag*	Tag to search in xml node
in	*att_name*	Atribute name to collect from tag
out	*att_value*	Attribute value
in	*mandatory*	Swich for optional or mandatory tags

Here is the caller graph for this function:


initialize::init_properties
initialize::init_sources
simulation_xmlparser ::readxmlatt
initialize::initmohidlagrangian
initialize::init_simdefs
initialize::init_caseconstants

5.10.2.3 readxmlvector()

```
subroutine, public simulation_xmlparser::readxmlvector (
            type(node), intent(in), pointer xmlnode,
            type(string), intent(in) tag,
            type(vector), intent(out) vec,
            logical, intent(in), optional mandatory )
```

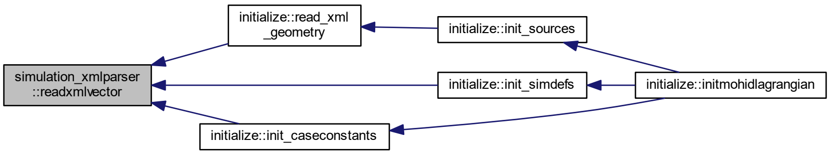
Birjukovs Canelas - MARETEC

Private vector xml parser routine. Vector must be in format <Tag x="vec%x" y="vec%y" z="vec%z">

Parameters

in	*xmlnode,tag,vec,mandatory*	
in	*xmlnode*	Working xml node
in	*tag*	Tag to search in xml node
out	*vec*	Vector to fill with read contents
in	*mandatory*	Swich for optional or mandatory tags

Here is the caller graph for this function:


initialize::read_xml _geometry
initialize::init_sources
simulation_xmlparser ::readxmlvector
initialize::init_simdefs
initialize::initmohidlagrangian
initialize::init_caseconstants

5.11 source_emitter Module Reference

Module that defines an emitter class and related methods. This module is responsible for building a potential tracer list based on the availble sources and calling their initializers.

Data Types

- type emitter_t

Functions/Subroutines

- subroutine initracers (self, srcs)

  Birjukovs Canelas - MARETEC
- subroutine alloctracers (self, srcs)

  Birjukovs Canelas - MARETEC
- subroutine initialize (self, srcs)

  Birjukovs Canelas - MARETEC
- subroutine setotalnp (src)

  Birjukovs Canelas - MARETEC

Variables

- type(emitter_t), public emitter

5.11.1 Detailed Description

Module that defines an emitter class and related methods. This module is responsible for building a potential tracer list based on the availble sources and calling their initializers.

Author

Ricardo Birjukovs Canelas

5.11.2 Function/Subroutine Documentation

5.11.2.1 alloctracers()

```
subroutine source_emitter::alloctracers (
            class(emitter_t), intent(inout) self,
            class(source_class), dimension(:), intent(inout) srcs )  [private]
```

Birjukovs Canelas - MARETEC

method that initializes an emmiter class object. Computes the total emittable particles this emmiter will allocate and sets other variables

Parameters

in	*self,src*	

5.11.2.2 initialize()

```
subroutine source_emitter::initialize (
            class(emitter_t), intent(inout) self,
            class(source_class), dimension(:), intent(inout) srcs )  [private]
```

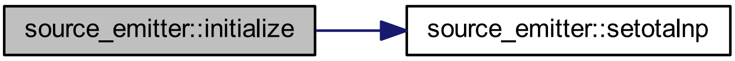
Birjukovs Canelas - MARETEC

method that initializes an emmiter class object. Computes the total emittable particles this emmiter will allocate and sets other variables

Parameters

in	*self,src*	

Here is the call graph for this function:



5.11.2.3 initracers()

```
subroutine source_emitter::initracers (
            class(emitter_t), intent(inout) self,
            class(source_class), dimension(:), intent(inout) srcs )  [private]
```

Birjukovs Canelas - MARETEC

method that calls the tracer initialization from the emmiter object

Parameters

in	*self,src*	

5.11.2.4 setotalnp()

```
subroutine source_emitter::setotalnp (
            class(source_class), intent(inout) src )  [private]
```

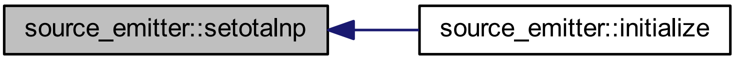
Birjukovs Canelas - MARETEC

private routine that returns the total number of tracers an input source will potentially create

Parameters

in	*src*	

$NP_{total}^{source-i} = (T_{end}^{source-i} - T_{start}^{source-i}) * Rate^{source-i} * NP_{emission}^{source-i}$ Here is the caller graph for this function:



5.11.3 Variable Documentation

5.11.3.1 emitter

```
type(emitter_t), public source_emitter::emitter
```

5.12 source_identity Module Reference

Module that defines a source class and related methods.

Data Types

- type source_class

  Type - The source class.
- type source_par
- type source_state

  Type - state variables of a source object.
- type source_stats

  Type - statistical variables of a source object.
- type source_stencil

  Type - holder for the tracer creation stencil of the source.

Functions/Subroutines

- subroutine, public allocsources (nsources)

  Birjukovs Canelas - MARETEC
- subroutine, public setsourceproperties (srcid_str, ptype, pname)

  Birjukovs Canelas - MARETEC
- subroutine initialize (src, id, name, emitting_rate, start, finish, source_geometry, geometry)

  Birjukovs Canelas - MARETEC
- subroutine linkproperty (src, ptype, pname)

  Birjukovs Canelas - MARETEC
- subroutine printout (src)

  Birjukovs Canelas - MARETEC

Variables

- type(source_class), dimension(:), allocatable, public source

5.12.1 Detailed Description

Module that defines a source class and related methods.

Author

Ricardo Birjukovs Canelas

5.12.2 Function/Subroutine Documentation

5.12.2.1 allocsources()

```
subroutine, public source_identity::allocsources (
             integer, intent(in) nsources )
```

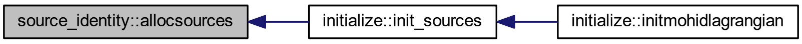
Birjukovs Canelas - MARETEC

source allocation routine - allocates the sources objects

Parameters

in	*nsources*	

Here is the caller graph for this function:


source_identity::allocsources
initialize::init_sources
initialize::initmohidlagrangian

5.12.2.2 initialize()

```
subroutine source_identity::initialize (
             class(source_class) src,
             integer, intent(in) id,
             type(string), intent(in) name,
             real(prec), intent(in) emitting_rate,
             real(prec), intent(in) start,
             real(prec), intent(in) finish,
             type(string), intent(in) source_geometry,
             class(shape), intent(in) geometry )  [private]
```

Birjukovs Canelas - MARETEC

source inititialization proceadure - initializes Source variables

Parameters

in	*src,id,name,emitting_rate,source_geometry*	

5.12.2.3 linkproperty()

```
subroutine source_identity::linkproperty (
            class(source_class) src,
            type(string), intent(in) ptype,
            type(string), intent(in) pname )  [private]
```

Birjukovs Canelas - MARETEC

source property setting proceadure - initializes Source variables

Parameters

in	*src,ptype,pname*	

5.12.2.4 printout()

```
subroutine source_identity::printout (
            class(source_class) src )  [private]
```

Birjukovs Canelas - MARETEC

source print routine - prints a source info on console/log

Parameters

in	*src*	

5.12.2.5 setsourceproperties()

```
subroutine, public source_identity::setsourceproperties (
            type(string), intent(in) srcid_str,
            type(string), intent(in) ptype,
            type(string), intent(in) pname )
```

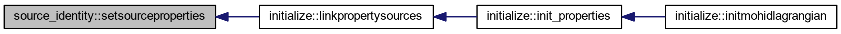
Birjukovs Canelas - MARETEC

source property setting routine, calls source by id to set its properties

Parameters

in	*srcid,ptype,pname*	
in	*srcid_str*	Source id tag
in	*ptype*	Property type to set
in	*pname*	Property name to set

Here is the caller graph for this function:


source_identity::setsourceproperties
initialize::linkpropertysources
initialize::init_properties
initialize::initmohidlagrangian

5.12.3 Variable Documentation

5.12.3.1 source

```
type(source_class), dimension(:), allocatable, public source_identity::source
```

5.13 sources Module Reference

Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods.

5.13.1 Detailed Description

Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods.

Author

Ricardo Birjukovs Canelas

5.14 tracer_base Module Reference

Module that defines a pure Lagrangian tracer class and related methods.

Data Types

- type tracer_class

  Type - The pure Lagrangian tracer class.
- type tracer_par_class
- type tracer_state_class

  Type - state variables of a pure Lagrangian tracer object.
- type tracer_stats_class

  Type - statistical variables of a pure Lagrangian tracer object.

Functions/Subroutines

- subroutine initialize (trc, id, id_source, time, pt)

  Birjukovs Canelas - MARETEC

Variables

- type(tracer_class), dimension(:), allocatable, public tracer

5.14.1 Detailed Description

Module that defines a pure Lagrangian tracer class and related methods.

Author

Ricardo Birjukovs Canelas

5.14.2 Function/Subroutine Documentation

5.14.2.1 initialize()

```
subroutine tracer_base::initialize (
             class(tracer_class) trc,
             integer, intent(in) id,
             integer, intent(in) id_source,
             real(prec_time), intent(in) time,
             type(vector), intent(in) pt )  [private]
```

Birjukovs Canelas - MARETEC

Tracer initialization method

Parameters

in		

5.14.3 Variable Documentation

5.14.3.1 tracer

```
type(tracer_class), dimension(:), allocatable, public tracer_base::tracer
```

5.15 tracer_interp Module Reference

5.16 tracer_paper Module Reference

Module that defines a Lagrangian tracer class for paper modelling and related methods. The type is defined as a derived type from the pule Lagrangian tracer, and hence inherits all of it's data and methods.

Data Types

- type paper_class

  Type - The plastic material Lagrangian tracer class.
- type paper_par_class
- type paper_state_class

  Type - State variables of a tracer object representing a paper material.

Functions/Subroutines

- subroutine paper_initialize (trc, id, id_source, time, pt)

  Birjukovs Canelas - MARETEC

5.16.1 Detailed Description

Module that defines a Lagrangian tracer class for paper modelling and related methods. The type is defined as a derived type from the pule Lagrangian tracer, and hence inherits all of it's data and methods.

Author

Ricardo Birjukovs Canelas

5.16.2 Function/Subroutine Documentation

5.16.2.1 paper_initialize()

```
subroutine tracer_paper::paper_initialize (
            class(paper_class) trc,
            integer, intent(in) id,
            integer, intent(in) id_source,
            real(prec_time), intent(in) time,
            type(vector), intent(in) pt )  [private]
```

Birjukovs Canelas - MARETEC

Tracer initialization method

Parameters

in		

5.17 tracer_plastic Module Reference

Module that defines a Lagrangian tracer class for plastic modelling and related methods. The type is defined as a derived type from the pule Lagrangian tracer, and hence inherits all of it's data and methods.

Data Types

- type plastic_class

  Type - The plastic material Lagrangian tracer class.
- type plastic_par_class
- type plastic_state_class

  Type - State variables of a tracer object representing a plastic material.

Functions/Subroutines

- subroutine plastic_initialize (trc, id, id_source, time, pt)

  Birjukovs Canelas - MARETEC

5.17.1 Detailed Description

Module that defines a Lagrangian tracer class for plastic modelling and related methods. The type is defined as a derived type from the pule Lagrangian tracer, and hence inherits all of it's data and methods.

Author

Ricardo Birjukovs Canelas

5.17.2 Function/Subroutine Documentation

5.17.2.1 plastic_initialize()

```
subroutine tracer_plastic::plastic_initialize (
            class(plastic_class) trc,
            integer, intent(in) id,
            integer, intent(in) id_source,
            real(prec_time), intent(in) time,
            type(vector), intent(in) pt )  [private]
```

Birjukovs Canelas - MARETEC

Tracer initialization method

Parameters

in		

5.18 tracers Module Reference

Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer block. This is intended to serve as the base class for every type of tracer class needed, that should be built as derived of this class, with the necessary modifiers to model the desired behaviour. Basic tracer data (parameters, variables) are implemented. Tracer methods such as I/O, integration and interpolation routines are implemented.

5.18.1 Detailed Description

Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer block. This is intended to serve as the base class for every type of tracer class needed, that should be built as derived of this class, with the necessary modifiers to model the desired behaviour. Basic tracer data (parameters, variables) are implemented. Tracer methods such as I/O, integration and interpolation routines are implemented.

Author

Ricardo Birjukovs Canelas

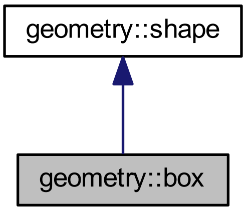
Chapter 6

Data Type Documentation

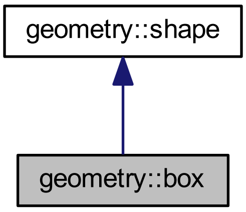
6.1 geometry::box Type Reference

Type - point class.

Inheritance diagram for geometry::box:



Collaboration diagram for geometry::box:



Private Attributes

- type(vector) size

  Box size.

6.1.1 Detailed Description

Type - point class.

6.1.2 Member Data Documentation

6.1.2.1 size

```
type(vector) geometry::box::size  [private]
```

Box size.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90

6.2 simulation_globals::constants_t Type Reference

Case Constants class.

Private Member Functions

- procedure setgravity
- procedure setrho

Private Attributes

- type(vector) gravity

  Gravitational acceleration vector (default=(0 0 -9.81)) (m s-2)
- real(prec) rho_ref = 1000.0

  Reference density of the medium (default=1000.0) (kg m-3)

6.2.1 Detailed Description

Case Constants class.

6.2.2 Member Function/Subroutine Documentation

6.2.2.1 setgravity()

```
procedure simulation_globals::constants_t::setgravity ( )  [private]
```

6.2.2.2 setrho()

```
procedure simulation_globals::constants_t::setrho ( )  [private]
```

6.2.3 Member Data Documentation

6.2.3.1 gravity

```
type(vector) simulation_globals::constants_t::gravity  [private]
```

Gravitational acceleration vector (default=(0 0 -9.81)) (m s-2)

6.2.3.2 rho_ref

```
real(prec) simulation_globals::constants_t::rho_ref = 1000.0  [private]
```

Reference density of the medium (default=1000.0) (kg m-3)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90

6.3 source_emitter::emitter_t Type Reference

Private Member Functions

- procedure initialize
- procedure alloctracers
- procedure initracers

Private Attributes

- integer emitted
- integer emittable

6.3.1 Member Function/Subroutine Documentation

6.3.1.1 alloctracers()

```
procedure source_emitter::emitter_t::alloctracers ( )  [private]
```

6.3.1.2 initialize()

```
procedure source_emitter::emitter_t::initialize ( )  [private]
```

6.3.1.3 initracers()

```
procedure source_emitter::emitter_t::initracers ( )  [private]
```

6.3.2 Member Data Documentation

6.3.2.1 emittable

```
integer source_emitter::emitter_t::emittable  [private]
```

6.3.2.2 emitted

```
integer source_emitter::emitter_t::emitted  [private]
```

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_emitter.f90

6.4 simulation_globals::filenames_t Type Reference

File names class.

Private Attributes

- type(string) mainxmlfilename

  Input .xml file name.
- type(string) propsxmlfilename

  Properties .xml file name.

6.4.1 Detailed Description

File names class.

6.4.2 Member Data Documentation

6.4.2.1 mainxmlfilename

```
type(string) simulation_globals::filenames_t::mainxmlfilename  [private]
```

Input .xml file name.

6.4.2.2 propsxmlfilename

```
type(string) simulation_globals::filenames_t::propsxmlfilename  [private]
```

Properties .xml file name.

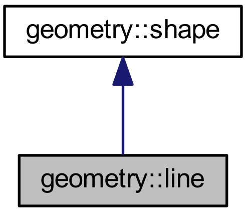
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90

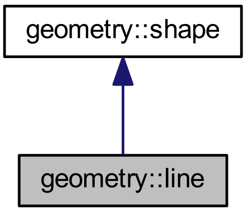
6.5 geometry::line Type Reference

Type - line class.

Inheritance diagram for geometry::line:



Collaboration diagram for geometry::line:



Private Attributes

- type(vector) last

  Coordinates of the end point.

6.5.1 Detailed Description

Type - line class.

6.5.2 Member Data Documentation

6.5.2.1 last

```
type(vector) geometry::line::last  [private]
```

Coordinates of the end point.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90

6.6 simulation_memory::memory_t Type Reference

Private Member Functions

- procedure initialize
- procedure addsource
- procedure addtracer
- procedure adddef
- procedure getotal
- procedure printout => printmemory
- procedure detailedprintout => printmemorydetailed

Private Attributes

- integer size_of_sources

    Size of the sources in memory (bytes)
- integer size_of_tracers

    Size of the tracers in memory (bytes)
- integer size_of_defs

    Size of the parameters and definitions in memory (bytes)

6.6.1 Member Function/Subroutine Documentation

6.6.1.1 adddef()

```
procedure simulation_memory::memory_t::adddef ( )  [private]
```

6.6.1.2 addsource()

```
procedure simulation_memory::memory_t::addsource ( )  [private]
```

6.6.1.3 addtracer()

```
procedure simulation_memory::memory_t::addtracer ( )  [private]
```

6.6.1.4 detailedprintout()

```
procedure simulation_memory::memory_t::detailedprintout ( )  [private]
```

6.6.1.5 getotal()

```
procedure simulation_memory::memory_t::getotal ( )  [private]
```

6.6.1.6 initialize()

```
procedure simulation_memory::memory_t::initialize ( )  [private]
```

6.6.1.7 printout()

```
procedure simulation_memory::memory_t::printout ( )  [private]
```

6.6.2 Member Data Documentation

6.6.2.1 size_of_defs

```
integer simulation_memory::memory_t::size_of_defs  [private]
```

Size of the parameters and definitions in memory (bytes)

6.6.2.2 size_of_sources

```
integer simulation_memory::memory_t::size_of_sources  [private]
```

Size of the sources in memory (bytes)

6.6.2.3 size_of_tracers

```
integer simulation_memory::memory_t::size_of_tracers  [private]
```

Size of the tracers in memory (bytes)

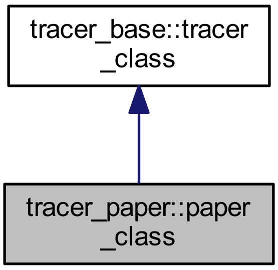
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_memory.f90

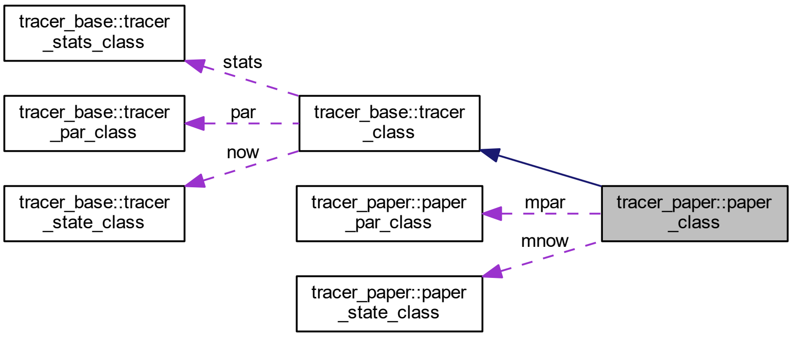
6.7 tracer_paper::paper_class Type Reference

Type - The plastic material Lagrangian tracer class.

Inheritance diagram for tracer_paper::paper_class:



Collaboration diagram for tracer_paper::paper_class:


tracer_base::tracer_stats_class
stats
tracer_base::tracer_par_class
par
tracer_base::tracer_class
now
tracer_base::tracer_state_class
tracer_paper::paper_par_class
mpar
tracer_paper::paper_class
mnow
tracer_paper::paper_state_class

Private Member Functions

- procedure initialize => paper_initialize

Private Attributes

- type(paper_par_class) mpar

  To access material parameters.
- type(paper_state_class) mnow

  To access material state variables.

6.7.1 Detailed Description

Type - The plastic material Lagrangian tracer class.

6.7.2 Member Function/Subroutine Documentation

6.7.2.1 initialize()

```
procedure tracer_paper::paper_class::initialize ( )  [private]
```

6.7.3 Member Data Documentation

6.7.3.1 mnow

```
type(paper_state_class) tracer_paper::paper_class::mnow  [private]
```

To access material state variables.

6.7.3.2 mpar

```
type(paper_par_class) tracer_paper::paper_class::mpar  [private]
```

To access material parameters.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_paper.f90

6.8 tracer_paper::paper_par_class Type Reference

Private Attributes

- real(prec) density

  density of the material
- real(prec) degradation_rate

  degradation rate of the material
- logical particulate

  flag to indicate if the material is a particle (false) or a collection of particles (true)
- real(prec) size

  Size (radius) of the particles (equals to the tracer radius if particulate==false)

6.8.1 Member Data Documentation

6.8.1.1 degradation_rate

```
real(prec) tracer_paper::paper_par_class::degradation_rate  [private]
```

degradation rate of the material

6.8.1.2 density

```
real(prec) tracer_paper::paper_par_class::density  [private]
```

density of the material

6.8.1.3 particulate

```
logical tracer_paper::paper_par_class::particulate  [private]
```

flag to indicate if the material is a particle (false) or a collection of particles (true)

6.8.1.4 size

```
real(prec) tracer_paper::paper_par_class::size  [private]
```

Size (radius) of the particles (equals to the tracer radius if particulate==false)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_paper.f90

6.9 tracer_paper::paper_state_class Type Reference

Type - State variables of a tracer object representing a paper material.

Private Attributes

- real(prec) radius

    Tracer radius (m)
- real(prec) condition

    Material condition (1-0)
- real(prec) concentration

    Particle concentration.

6.9.1 Detailed Description

Type - State variables of a tracer object representing a paper material.

6.9.2 Member Data Documentation

6.9.2.1 concentration

```
real(prec) tracer_paper::paper_state_class::concentration  [private]
```

Particle concentration.

6.9.2.2 condition

```
real(prec) tracer_paper::paper_state_class::condition  [private]
```

Material condition (1-0)

6.9.2.3 radius

```
real(prec) tracer_paper::paper_state_class::radius  [private]
```

Tracer radius (m)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_paper.f90

6.10 simulation_globals::parameters_t Type Reference

Private Member Functions

- procedure setparameter
- procedure check
- procedure printout => printsimparameters

Private Attributes

- integer integrator = 1

  Integration Algorithm 1:Verlet, 2:Symplectic, 3:RK4 (default=1)
- real(prec) cfl = 0.5

  Courant Friedrichs Lewy condition number.
- real(prec) warmuptime = 0.0

  Time to freeze the tracers at simulation start (warmup) (s) (default=0.0)
- real(prec) timemax = MV

  Simulation duration (s)
- real(prec) timeout = MV

  Time out data (1/Hz)

6.10.1 Member Function/Subroutine Documentation

6.10.1.1 check()

```
procedure simulation_globals::parameters_t::check ( )  [private]
```

6.10.1.2 printout()

```
procedure simulation_globals::parameters_t::printout ( )  [private]
```

6.10.1.3 setparameter()

```
procedure simulation_globals::parameters_t::setparameter ( )  [private]
```

6.10.2 Member Data Documentation

6.10.2.1 cfl

```
real(prec) simulation_globals::parameters_t::cfl = 0.5  [private]
```

Courant Friedrichs Lewy condition number.

6.10.2.2 integrator

```
integer simulation_globals::parameters_t::integrator = 1  [private]
```

Integration Algorithm 1:Verlet, 2:Symplectic, 3:RK4 (default=1)

6.10.2.3 timemax

```
real(prec) simulation_globals::parameters_t::timemax = MV  [private]
```

Simulation duration (s)

6.10.2.4 timeout

```
real(prec) simulation_globals::parameters_t::timeout = MV  [private]
```

Time out data (1/Hz)

6.10.2.5 warmuptime

```
real(prec) simulation_globals::parameters_t::warmuptime = 0.0  [private]
```

Time to freeze the tracers at simulation start (warmup) (s) (default=0.0)

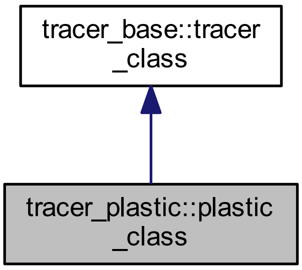
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90

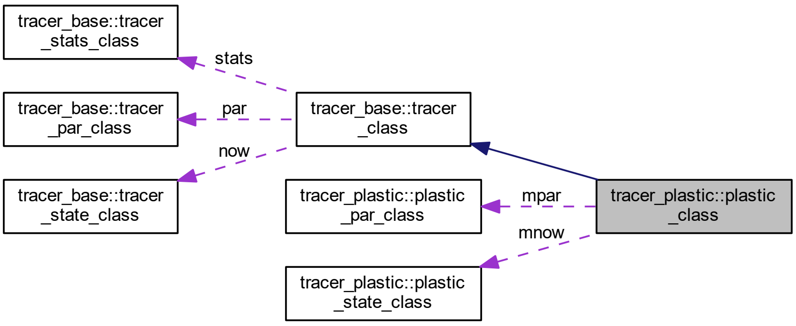
6.11 tracer_plastic::plastic_class Type Reference

Type - The plastic material Lagrangian tracer class.

Inheritance diagram for tracer_plastic::plastic_class:



Collaboration diagram for tracer_plastic::plastic_class:



Private Member Functions

- procedure initialize => plastic_initialize

Private Attributes

- type(plastic_par_class) mpar

    To access material parameters.
- type(plastic_state_class) mnow

    To access material state variables.

6.11.1 Detailed Description

Type - The plastic material Lagrangian tracer class.

6.11.2 Member Function/Subroutine Documentation

6.11.2.1 initialize()

```
procedure tracer_plastic::plastic_class::initialize ( )  [private]
```

6.11.3 Member Data Documentation

6.11.3.1 mnow

```
type(plastic_state_class) tracer_plastic::plastic_class::mnow  [private]
```

To access material state variables.

6.11.3.2 mpar

```
type(plastic_par_class) tracer_plastic::plastic_class::mpar  [private]
```

To access material parameters.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_plastic.f90

6.12 tracer_plastic::plastic_par_class Type Reference

Private Attributes

- real(prec) density

  density of the material
- real(prec) degradation_rate

  degradation rate of the material
- logical particulate

  flag to indicate if the material is a particle (false) or a collection of particles (true)
- real(prec) size

  Size (radius) of the particles (equals to the tracer radius if particulate==false)

6.12.1 Member Data Documentation

6.12.1.1 degradation_rate

```
real(prec) tracer_plastic::plastic_par_class::degradation_rate  [private]
```

degradation rate of the material

6.12.1.2 density

```
real(prec) tracer_plastic::plastic_par_class::density  [private]
```

density of the material

6.12.1.3 particulate

```
logical tracer_plastic::plastic_par_class::particulate  [private]
```

flag to indicate if the material is a particle (false) or a collection of particles (true)

6.12.1.4 size

```
real(prec) tracer_plastic::plastic_par_class::size  [private]
```

Size (radius) of the particles (equals to the tracer radius if particulate==false)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_plastic.f90

6.13 tracer_plastic::plastic_state_class Type Reference

Type - State variables of a tracer object representing a plastic material.

Private Attributes

- real(prec) radius

  Tracer radius (m)
- real(prec) condition

  Material condition (1-0)
- real(prec) concentration

  Particle concentration.

6.13.1 Detailed Description

Type - State variables of a tracer object representing a plastic material.

6.13.2 Member Data Documentation

6.13.2.1 concentration

```
real(prec) tracer_plastic::plastic_state_class::concentration  [private]
```

Particle concentration.

6.13.2.2 condition

```
real(prec) tracer_plastic::plastic_state_class::condition  [private]
```

Material condition (1-0)

6.13.2.3 radius

```
real(prec) tracer_plastic::plastic_state_class::radius  [private]
```

Tracer radius (m)

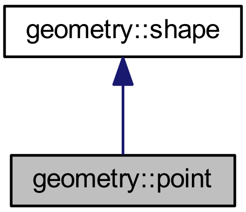
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_plastic.f90

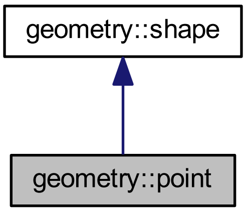
6.14 geometry::point Type Reference

Type - point class.

Inheritance diagram for geometry::point:



Collaboration diagram for geometry::point:



6.14.1 Detailed Description

Type - point class.

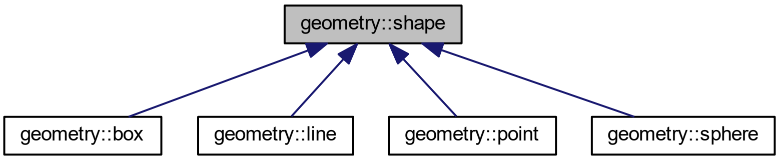
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90

6.15 geometry::shape Type Reference

Type - extendable shape class.

Inheritance diagram for geometry::shape:



Private Member Functions

- procedure getnp

  Gets the number of points that define that geometry (based on GLOBALS::dp)
- procedure getpointdistribution

  Gets the actual list of points always referant to the origin (based on GLOBALS::dp)

Private Attributes

- type(vector) pt

  Coordinates of a point.

6.15.1 Detailed Description

Type - extendable shape class.

6.15.2 Member Function/Subroutine Documentation

6.15.2.1 getnp()

```
procedure geometry::shape::getnp ( )  [private]
```

Gets the number of points that define that geometry (based on GLOBALS::dp)

6.15.2.2 getpointdistribution()

```
procedure geometry::shape::getpointdistribution ( )  [private]
```

Gets the actual list of points always referant to the origin (based on GLOBALS::dp)

6.15.3 Member Data Documentation

6.15.3.1 pt

```
type(vector) geometry::shape::pt  [private]
```

Coordinates of a point.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90

6.16 simulation_globals::simdefs_t Type Reference

Simulation definitions class.

Private Member Functions

- procedure setdp
- procedure setdt
- procedure setboundingbox
- procedure printout => printsimdefs

Private Attributes

- real(prec) dp = MV

  Initial particle spacing at source generation.
- real(prec_time) dt

  Timestep for fixed step integrators (s)
- type(vector) pointmin

  Point that defines the lowest corner of the simulation bounding box.
- type(vector) pointmax

  Point that defines the upper corner of the simulation bounding box.

6.16.1 Detailed Description

Simulation definitions class.

6.16.2 Member Function/Subroutine Documentation

6.16.2.1 printout()

```
procedure simulation_globals::simdefs_t::printout ( )  [private]
```

6.16.2.2 setboundingbox()

```
procedure simulation_globals::simdefs_t::setboundingbox ( )  [private]
```

6.16.2.3 setdp()

```
procedure simulation_globals::simdefs_t::setdp ( )  [private]
```

6.16.2.4 setdt()

```
procedure simulation_globals::simdefs_t::setdt ( )  [private]
```

6.16.3 Member Data Documentation

6.16.3.1 dp

```
real(prec) simulation_globals::simdefs_t::dp = MV  [private]
```

Initial particle spacing at source generation.

6.16.3.2 dt

```
real(prec_time) simulation_globals::simdefs_t::dt  [private]
```

Timestep for fixed step integrators (s)

6.16.3.3 pointmax

```
type(vector) simulation_globals::simdefs_t::pointmax  [private]
```

Point that defines the upper corner of the simulation bounding box.

6.16.3.4 pointmin

```
type(vector) simulation_globals::simdefs_t::pointmin  [private]
```

Point that defines the lowest corner of the simulation bounding box.

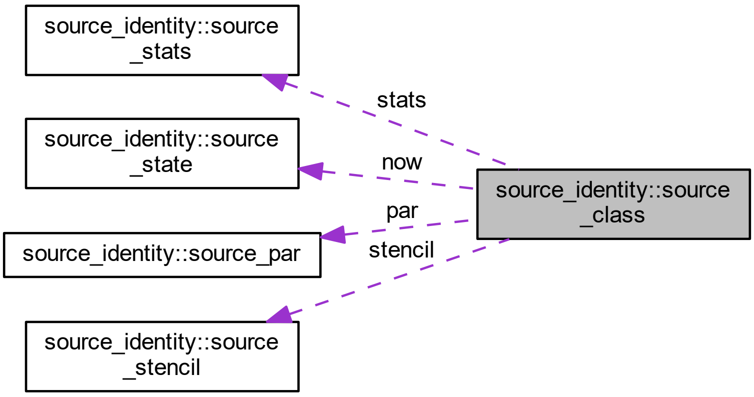
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90

6.17 source_identity::source_class Type Reference

Type - The source class.

Collaboration diagram for source_identity::source_class:


source_identity::source_stats
source_identity::source_state
source_identity::source_par
source_identity::source_stencil
stats
now
par
stencil
source_identity::source_class

Private Member Functions

- procedure initialize
- procedure linkproperty
- procedure printout

Private Attributes

- type(source_par) par

  To access parameters.
- type(source_state) now

  To access state variables.
- type(source_stencil) stencil

  To acess stencil variables.
- type(source_stats) stats

  To access statistics.

6.17.1 Detailed Description

Type - The source class.

6.17.2 Member Function/Subroutine Documentation

6.17.2.1 initialize()

```
procedure source_identity::source_class::initialize ( )  [private]
```

6.17.2.2 linkproperty()

```
procedure source_identity::source_class::linkproperty ( )  [private]
```

6.17.2.3 printout()

```
procedure source_identity::source_class::printout ( )  [private]
```

6.17.3 Member Data Documentation

6.17.3.1 now

```
type(source_state) source_identity::source_class::now  [private]
```

To access state variables.

6.17.3.2 par

```
type(source_par) source_identity::source_class::par  [private]
```

To access parameters.

6.17.3.3 stats

```
type(source_stats) source_identity::source_class::stats  [private]
```

To access statistics.

6.17.3.4 stencil

```
type(source_stencil) source_identity::source_class::stencil  [private]
```

To acess stencil variables.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90

6.18 source_identity::source_par Type Reference

Private Attributes

- integer id

  unique source identification (integer)
- real(prec_time) emitting_rate

  Emitting rate of the source (Hz)
- real(prec_time) startime

  time to start emitting tracers
- real(prec_time) stoptime

  time to stop emitting tracers
- type(string) name

  source name
- type(string) property_type

  source property type (plastic, paper, fish, etc)
- type(string) property_name

  source property name
- type(string) source_geometry

  Source type : 'point', 'line', 'sphere', 'box'.
- class(shape), allocatable geometry

  Source geometry.

6.18.1 Member Data Documentation

6.18.1.1 emitting_rate

```
real(prec_time) source_identity::source_par::emitting_rate  [private]
```

Emitting rate of the source (Hz)

6.18.1.2 geometry

```
class(shape), allocatable source_identity::source_par::geometry  [private]
```

Source geometry.

6.18.1.3 id

```
integer source_identity::source_par::id  [private]
```

unique source identification (integer)

6.18.1.4 name

```
type(string) source_identity::source_par::name  [private]
```

source name

6.18.1.5 property_name

```
type(string) source_identity::source_par::property_name  [private]
```

source property name

6.18.1.6 property_type

```
type(string) source_identity::source_par::property_type  [private]
```

source property type (plastic, paper, fish, etc)

6.18.1.7 source_geometry

```
type(string) source_identity::source_par::source_geometry  [private]
```

Source type : 'point', 'line', 'sphere', 'box'.

6.18.1.8 startime

```
real(prec_time) source_identity::source_par::startime  [private]
```

time to start emitting tracers

6.18.1.9 stoptime

```
real(prec_time) source_identity::source_par::stoptime  [private]
```

time to stop emitting tracers

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90

6.19 source_identity::source_state Type Reference

Type - state variables of a source object.

Private Attributes

- real(prec_time) age
- logical active

  active switch
- type(vector) pos

  Position of the source baricenter (m)
- type(vector) vel

  Velocity of the source (m s-1)
- real(prec) depth

  Depth of the source baricenter (m)
- real(prec) t

  Temperature of the source (Celcius)

6.19.1 Detailed Description

Type - state variables of a source object.

6.19.2 Member Data Documentation

6.19.2.1 active

```
logical source_identity::source_state::active  [private]
```

active switch

6.19.2.2 age

```
real(prec_time) source_identity::source_state::age  [private]
```

6.19.2.3 depth

```
real(prec) source_identity::source_state::depth  [private]
```

Depth of the source baricenter (m)

6.19.2.4 pos

```
type(vector) source_identity::source_state::pos  [private]
```

Position of the source baricenter (m)

6.19.2.5 t

```
real(prec) source_identity::source_state::t  [private]
```

Temperature of the source (Celcius)

6.19.2.6 vel

```
type(vector) source_identity::source_state::vel  [private]
```

Velocity of the source (m s-1)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90

6.20 source_identity::source_stats Type Reference

Type - statistical variables of a source object.

Private Attributes

- integer particles_emitted

  Number of emitted particles by this source.
- real(prec_wrt) acc_t

  Accumulated temperature of the tracer (Celcius)
- integer ns

  Number of sampling steps.

6.20.1 Detailed Description

Type - statistical variables of a source object.

6.20.2 Member Data Documentation

6.20.2.1 acc_t

```
real(prec_wrt) source_identity::source_stats::acc_t  [private]
```

Accumulated temperature of the tracer (Celcius)

6.20.2.2 ns

```
integer source_identity::source_stats::ns  [private]
```

Number of sampling steps.

6.20.2.3 particles_emitted

```
integer source_identity::source_stats::particles_emitted  [private]
```

Number of emitted particles by this source.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90

6.21 source_identity::source_stencil Type Reference

Type - holder for the tracer creation stencil of the source.

Private Attributes

- integer np

  Number of tracers by emission.
- integer total_np

  Total number of tracers that this source will generate.
- type(vector), dimension(:), allocatable ptlist

  list of points (coordinates), relative to the source geometry point, to be generated at every emission

6.21.1 Detailed Description

Type - holder for the tracer creation stencil of the source.

6.21.2 Member Data Documentation

6.21.2.1 np

```
integer source_identity::source_stencil::np  [private]
```

Number of tracers by emission.

6.21.2.2 ptlist

```
type(vector), dimension(:), allocatable source_identity::source_stencil::ptlist  [private]
```

list of points (coordinates), relative to the source geometry point, to be generated at every emission

6.21.2.3 total_np

```
integer source_identity::source_stencil::total_np  [private]
```

Total number of tracers that this source will generate.

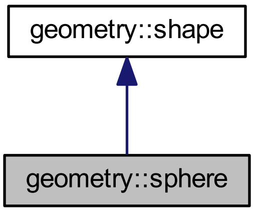
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90

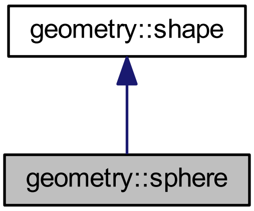
6.22 geometry::sphere Type Reference

Type - sphere class.

Inheritance diagram for geometry::sphere:



Collaboration diagram for geometry::sphere:



Private Attributes

- real(prec) radius

  Sphere radius.

6.22.1 Detailed Description

Type - sphere class.

6.22.2 Member Data Documentation

6.22.2.1 radius

```
real(prec) geometry::sphere::radius [private]
```

Sphere radius.

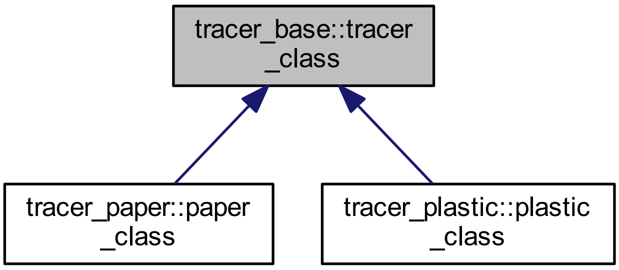
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90

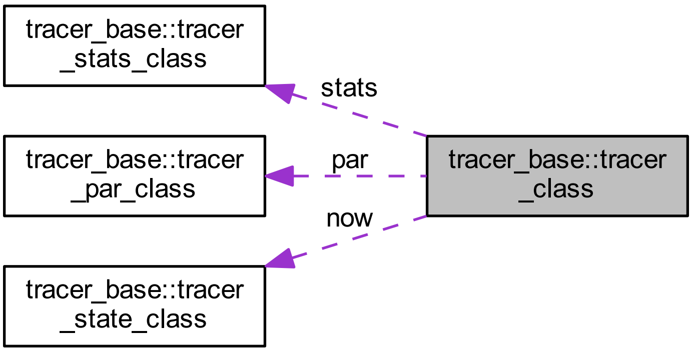
6.23 tracer_base::tracer_class Type Reference

Type - The pure Lagrangian tracer class.

Inheritance diagram for tracer_base::tracer_class:


tracer_base::tracer _class
tracer_paper::paper _class
tracer_plastic::plastic _class

Collaboration diagram for tracer_base::tracer_class:


tracer_base::tracer_stats_class
tracer_base::tracer_par_class
tracer_base::tracer_state_class
tracer_base::tracer_class
stats
par
now

Private Member Functions

- procedure initialize

Private Attributes

- type(tracer_par_class) par

  To access parameters.
- type(tracer_state_class) now

  To access state variables.
- type(tracer_stats_class) stats

  To access statistics.

6.23.1 Detailed Description

Type - The pure Lagrangian tracer class.

6.23.2 Member Function/Subroutine Documentation

6.23.2.1 initialize()

```
procedure tracer_base::tracer_class::initialize ( )  [private]
```

6.23.3 Member Data Documentation

6.23.3.1 now

```
type(tracer_state_class) tracer_base::tracer_class::now [private]
```

To access state variables.

6.23.3.2 par

```
type(tracer_par_class) tracer_base::tracer_class::par [private]
```

To access parameters.

6.23.3.3 stats

```
type(tracer_stats_class) tracer_base::tracer_class::stats [private]
```

To access statistics.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_base.f90

6.24 tracer_base::tracer_par_class Type Reference

Private Attributes

- integer id

  unique tracer identification
- integer idsource

  Source to which the tracer belongs.
- real(prec) velmax

  Maximum velocity of tracer to track (m/s)
- logical noise
- type(string) interp_method

  interpolation method this tracer calls

6.24.1 Member Data Documentation

6.24.1.1 id

```
integer tracer_base::tracer_par_class::id  [private]
```

unique tracer identification

6.24.1.2 idsource

```
integer tracer_base::tracer_par_class::idsource  [private]
```

Source to which the tracer belongs.

6.24.1.3 interp_method

```
type(string) tracer_base::tracer_par_class::interp_method  [private]
```

interpolation method this tracer calls

6.24.1.4 noise

```
logical tracer_base::tracer_par_class::noise  [private]
```

6.24.1.5 velmax

```
real(prec) tracer_base::tracer_par_class::velmax  [private]
```

Maximum velocity of tracer to track (m/s)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_base.f90

6.25 tracer_base::tracer_state_class Type Reference

Type - state variables of a pure Lagrangian tracer object.

Private Attributes

- real(prec_time) age
- logical active

  active switch
- type(vector) pos

  Position of the tracer (m)
- type(vector) vel

  Velocity of the tracer (m s-1)
- type(vector) acc

  Acceleration of the tracer (m s-2)
- real(prec) depth

  Depth of the tracer (m)

6.25.1 Detailed Description

Type - state variables of a pure Lagrangian tracer object.

6.25.2 Member Data Documentation

6.25.2.1 acc

```
type(vector) tracer_base::tracer_state_class::acc  [private]
```

Acceleration of the tracer (m s-2)

6.25.2.2 active

```
logical tracer_base::tracer_state_class::active  [private]
```

active switch

6.25.2.3 age

```
real(prec_time) tracer_base::tracer_state_class::age  [private]
```

6.25.2.4 depth

```
real(prec) tracer_base::tracer_state_class::depth  [private]
```

Depth of the tracer (m)

6.25.2.5 pos

```
type(vector) tracer_base::tracer_state_class::pos  [private]
```

Position of the tracer (m)

6.25.2.6 vel

```
type(vector) tracer_base::tracer_state_class::vel  [private]
```

Velocity of the tracer (m s-1)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_base.f90

6.26 tracer_base::tracer_stats_class Type Reference

Type - statistical variables of a pure Lagrangian tracer object.

Private Attributes

- type(vector) acc_pos

  Accumulated position of the tracer (m)
- type(vector) acc_vel

  Accumulated velocity of the tracer (m s-1)
- real(prec_wrt) acc_depth

  Accumulated depth of the tracer (m)
- integer ns

  Number of sampling steps.

6.26.1 Detailed Description

Type - statistical variables of a pure Lagrangian tracer object.

6.26.2 Member Data Documentation

6.26.2.1 acc_depth

```
real(prec_wrt) tracer_base::tracer_stats_class::acc_depth  [private]
```

Accumulated depth of the tracer (m)

6.26.2.2 acc_pos

```
type(vector) tracer_base::tracer_stats_class::acc_pos  [private]
```

Accumulated position of the tracer (m)

6.26.2.3 acc_vel

```
type(vector) tracer_base::tracer_stats_class::acc_vel  [private]
```

Accumulated velocity of the tracer (m s-1)

6.26.2.4 ns

```
integer tracer_base::tracer_stats_class::ns  [private]
```

Number of sampling steps.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_base.f90

Chapter 7

File Documentation

7.1 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/app/main.f90 File Reference

7.2 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/about.f90 File Reference

Modules

- module about

  Module to print version, licence, preambles.

Functions/Subroutines

- subroutine, public about::printlicpreamble

  Birjukovs Canelas - MARETEC

Variables

- type(string) about::version
- type(string) about::author
- type(string) about::date

7.3 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_↩ modules.f90 File Reference

Modules

- module commom_modules

  Module to hold all of the commonly used base modules.

7.4 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/finalize.f90 File Reference

Modules

- module finalize

  Module with the simulation closing related definitions and methods.

Functions/Subroutines

- subroutine finalize::simulation_end

  Birjukovs Canelas - MARETEC
- subroutine finalize::deallocate_simulation

  Birjukovs Canelas - MARETEC
- subroutine finalize::closelog

  Birjukovs Canelas - MARETEC
- subroutine, public finalize::finalizemohidlagrangian

  Birjukovs Canelas - MARETEC

7.5 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90 File Reference

Data Types

- type geometry::shape

  Type - extendable shape class.
- type geometry::point

  Type - point class.
- type geometry::line

  Type - line class.
- type geometry::sphere

  Type - sphere class.
- type geometry::box

  Type - point class.

Modules

- module geometry

  Module that defines geometry classes and related methods.

Functions/Subroutines

- subroutine, public geometry::allocategeomlist

  Birjukovs Canelas - MARETEC
- logical function, public geometry::isvalidgeom (geomname)

  Birjukovs Canelas - MARETEC
- subroutine geometry::getnp (self, np, dp)

  Birjukovs Canelas - MARETEC
- subroutine geometry::getpointdistribution (self, np, dp, ptlist)

  Birjukovs Canelas - MARETEC
- subroutine geometry::sphere_np_count (dp, r, np)

  Birjukovs Canelas - MARETEC
- subroutine geometry::sphere_grid (dp, r, np, ptlist)

  Birjukovs Canelas - MARETEC
- subroutine geometry::box_grid (dp, size, np, ptlist)

  Birjukovs Canelas - MARETEC
- subroutine geometry::line_grid (dp, dist, np, ptlist)

  Birjukovs Canelas - MARETEC

Variables

- type(string), dimension(:), allocatable, public geometry::geomlist

  String list (array) with the name of possible geometry types.

7.6 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90 File Reference

Modules

- module initialize

  Module with the simulation initialization related definitions and methods. Has one public access routine that is in-charge of building the simulation space from input files.

Functions/Subroutines

- subroutine initialize::linkpropertysources (linksNode)

  Birjukovs Canelas - MARETEC
- subroutine initialize::init_properties (case_node)

  Birjukovs Canelas - MARETEC
- subroutine initialize::read_xml_geometry (source, source_detail, geometry)

  Birjukovs Canelas - MARETEC
- subroutine initialize::init_sources (case_node)

  Birjukovs Canelas - MARETEC
- subroutine initialize::init_simdefs (case_node)

  Birjukovs Canelas - MARETEC
- subroutine initialize::init_caseconstants (case_node)

  Birjukovs Canelas - MARETEC
- subroutine initialize::init_parameters (execution_node)

  Birjukovs Canelas - MARETEC
- subroutine, public initialize::initmohidlagrangian (xmlfilename)

  Birjukovs Canelas - MARETEC

7.7 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_↩ globals.f90 File Reference

Data Types

- type simulation_globals::parameters_t
- type simulation_globals::simdefs_t

  Simulation definitions class.
- type simulation_globals::constants_t

  Case Constants class.
- type simulation_globals::filenames_t

  File names class.

Modules

- module simulation_globals

  Module to hold the simulation global parameter classes and their methods.

Functions/Subroutines

- subroutine simulation_globals::setparameter (self, parmkey, parmvalue)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::check (self)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::printsimparameters (self)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::getintegratorname (name, code)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::setgravity (self, grav)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::setrho (self, read_rho)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::setdp (self, read_dp)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::setdt (self, read_dt)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::setboundingbox (self, point_, coords)

  Birjukovs Canelas - MARETEC
- subroutine simulation_globals::printsimdefs (self)

  Birjukovs Canelas - MARETEC

Variables

- real(prec_time), public simulation_globals::simtime
- type(parameters_t), public simulation_globals::parameters
- type(simdefs_t), public simulation_globals::simdefs
- type(constants_t), public simulation_globals::constants
- type(filenames_t), public simulation_globals::filenames

7.8 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_↩ logger.f90 File Reference

Modules

- module simulation_logger

  Module to hold all the simulation logger related definitions and methods.

Functions/Subroutines

- subroutine, public simulation_logger::gettimestamp (timestamp)

  Birjukovs Canelas - MARETEC
- subroutine, public simulation_logger::tolog (tologstr, timeoption)

  Birjukovs Canelas - MARETEC
- subroutine, public simulation_logger::initmohidlagrangianlog (outpath)

  Birjukovs Canelas - MARETEC

Variables

- integer, public simulation_logger::log_unit = -1

  'Number' of log file

7.9 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_↩ memory.f90 File Reference

Data Types

- type simulation_memory::memory_t

Modules

- module simulation_memory

  Module to hold the simulation memory managment class and its methods.

Functions/Subroutines

- subroutine simulation_memory::initialize (self)

  Birjukovs Canelas - MARETEC
- subroutine simulation_memory::getotal (self, size)

  Birjukovs Canelas - MARETEC
- subroutine simulation_memory::addsource (self, size)

  Birjukovs Canelas - MARETEC
- subroutine simulation_memory::addtracer (self, size)

  Birjukovs Canelas - MARETEC
- subroutine simulation_memory::adddef (self, size)

  Birjukovs Canelas - MARETEC
- subroutine simulation_memory::printmemory (self)

  Birjukovs Canelas - MARETEC
- subroutine simulation_memory::printmemorydetailed (self)

  Birjukovs Canelas - MARETEC

Variables

- type(memory_t), public simulation_memory::simmemory

7.10 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_↩ precision.f90 File Reference

Modules

- module simulation_precision

  Module to control the precision of the variables trough the project.

Variables

- integer, parameter simulation_precision::sp = kind(1._R4P)

  Simple precision definition switch.
- integer, parameter simulation_precision::dp = kind(1._R8P)

  Double precision definition switch.
- integer, parameter, public simulation_precision::prec = sp
- integer, parameter, public simulation_precision::prec_time = sp
- integer, parameter, public simulation_precision::prec_wrt = sp
- real(prec), parameter, public simulation_precision::missing_value_default = -9999.0_dp
- real(prec), parameter, public simulation_precision::mv = MISSING_VALUE_DEFAULT
- real(prec), parameter, public simulation_precision::mv_int = int(MISSING_VALUE_DEFAULT)
- real(prec), parameter, public simulation_precision::err_dist = 1E8_dp
- integer, parameter, public simulation_precision::err_ind = -1

7.11 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_↩ xmlparser.f90 File Reference

Modules

- module simulation_xmlparser

  Module with the simulation xml parsing related definitions and routines.

Functions/Subroutines

- subroutine, public simulation_xmlparser::readxmlatt (xmlnode, tag, att_name, att_value, mandatory)

  Birjukovs Canelas - MARETEC
- subroutine, public simulation_xmlparser::readxmlvector (xmlnode, tag, vec, mandatory)

  Birjukovs Canelas - MARETEC
- subroutine, public simulation_xmlparser::gotochildnode (currentNode, targetNode, targetNodeName, mandatory)

  Birjukovs Canelas - MARETEC

7.12 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_↩ emitter.f90 File Reference

Data Types

- type source_emitter::emitter_t

Modules

- module source_emitter

  Module that defines an emitter class and related methods. This module is responsible for building a potential tracer list based on the availble sources and calling their initializers.

Functions/Subroutines

- subroutine source_emitter::initracers (self, srcs)

  Birjukovs Canelas - MARETEC
- subroutine source_emitter::alloctracers (self, srcs)

  Birjukovs Canelas - MARETEC
- subroutine source_emitter::initialize (self, srcs)

  Birjukovs Canelas - MARETEC
- subroutine source_emitter::setotalnp (src)

  Birjukovs Canelas - MARETEC

Variables

- type(emitter_t), public source_emitter::emitter

7.13 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_↩ identity.f90 File Reference

Data Types

- type source_identity::source_par
- type source_identity::source_state

  Type - state variables of a source object.
- type source_identity::source_stats

  Type - statistical variables of a source object.
- type source_identity::source_stencil

  Type - holder for the tracer creation stencil of the source.
- type source_identity::source_class

  Type - The source class.

Modules

- module source_identity

  Module that defines a source class and related methods.

Functions/Subroutines

- subroutine, public source_identity::allocsources (nsources)

  Birjukovs Canelas - MARETEC
- subroutine, public source_identity::setsourceproperties (srcid_str, ptype, pname)

  Birjukovs Canelas - MARETEC
- subroutine source_identity::initialize (src, id, name, emitting_rate, start, finish, source_geometry, geometry)

  Birjukovs Canelas - MARETEC
- subroutine source_identity::linkproperty (src, ptype, pname)

  Birjukovs Canelas - MARETEC
- subroutine source_identity::printout (src)

  Birjukovs Canelas - MARETEC

Variables

- type(source_class), dimension(:), allocatable, public source_identity::source

7.14 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/sources.f90 File Reference

Modules

- module sources

  Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods.

7.15 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_↩ base.f90 File Reference

Data Types

- type tracer_base::tracer_par_class
- type tracer_base::tracer_state_class

  Type - state variables of a pure Lagrangian tracer object.
- type tracer_base::tracer_stats_class

  Type - statistical variables of a pure Lagrangian tracer object.
- type tracer_base::tracer_class

  Type - The pure Lagrangian tracer class.

Modules

- module tracer_base

  Module that defines a pure Lagrangian tracer class and related methods.

Functions/Subroutines

- subroutine tracer_base::initialize (trc, id, id_source, time, pt)

  Birjukovs Canelas - MARETEC

Variables

- type(tracer_class), dimension(:), allocatable, public tracer_base::tracer

7.16 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_↩ interp.f90 File Reference

Modules

- module tracer_interp

7.17 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_↩ paper.f90 File Reference

Data Types

- type tracer_paper::paper_par_class
- type tracer_paper::paper_state_class

  Type - State variables of a tracer object representing a paper material.
- type tracer_paper::paper_class

  Type - The plastic material Lagrangian tracer class.

Modules

- module tracer_paper

  Module that defines a Lagrangian tracer class for paper modelling and related methods. The type is defined as a derived type from the pule Lagrangian tracer, and hence inherits all of it's data and methods.

Functions/Subroutines

- subroutine tracer_paper::paper_initialize (trc, id, id_source, time, pt)

  Birjukovs Canelas - MARETEC

7.18 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_↩ plastic.f90 File Reference

Data Types

- type tracer_plastic::plastic_par_class
- type tracer_plastic::plastic_state_class

  Type - State variables of a tracer object representing a plastic material.
- type tracer_plastic::plastic_class

  Type - The plastic material Lagrangian tracer class.

Modules

- module tracer_plastic

    Module that defines a Lagrangian tracer class for plastic modelling and related methods. The type is defined as a derived type from the pule Lagrangian tracer, and hence inherits all of it's data and methods.

Functions/Subroutines

- subroutine tracer_plastic::plastic_initialize (trc, id, id_source, time, pt)

    Birjukovs Canelas - MARETEC

7.19 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracers.f90 File Reference

Modules

- module tracers

    Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer block. This is intended to serve as the base class for every type of tracer class needed, that should be built as derived of this class, with the necessary modifiers to model the desired behaviour. Basic tracer data (parameters, variables) are implemented. Tracer methods such as I/O, integration and interpolation routines are implemented.