

MOHIDLagrangian

0.01

Generated by Doxygen 1.8.14

Contents

1	Modules Index	1
1.1	Modules List	1
2	Data Type Index	3
2.1	Class Hierarchy	3
3	Data Type Index	5
3.1	Data Types List	5
4	File Index	7
4.1	File List	7
5	Module Documentation	9
5.1	commom_modules Module Reference	9
5.1.1	Detailed Description	9
5.2	geometry Module Reference	9
5.2.1	Detailed Description	10
5.3	initialize Module Reference	10
5.3.1	Detailed Description	10
5.3.2	Function/Subroutine Documentation	10
5.3.2.1	init_caseconstants()	10
5.3.2.2	init_parameters()	11
5.3.2.3	init_simdefs()	12
5.3.2.4	init_sources()	13
5.3.2.5	initmohidlagrangian()	13

5.3.2.6	read_xml_geometry()	14
5.3.2.7	readxmlatt()	15
5.3.2.8	readxmlvector()	15
5.4	simulation_globals Module Reference	16
5.4.1	Detailed Description	17
5.4.2	Function/Subroutine Documentation	17
5.4.2.1	getintegratorname()	17
5.4.2.2	printsimdefs()	18
5.4.2.3	printsimparameters()	18
5.4.2.4	setboundingbox()	18
5.4.2.5	setdp()	18
5.4.2.6	setgravity()	19
5.4.2.7	setparameter()	19
5.4.2.8	setrho()	19
5.4.3	Variable Documentation	20
5.4.3.1	constants	20
5.4.3.2	parameters	20
5.4.3.3	simdefs	20
5.5	simulation_precision Module Reference	20
5.5.1	Detailed Description	21
5.5.2	Variable Documentation	21
5.5.2.1	dp	21
5.5.2.2	err_dist	21
5.5.2.3	err_ind	21
5.5.2.4	missing_value_default	21
5.5.2.5	mv	21
5.5.2.6	mv_int	22
5.5.2.7	prec	22
5.5.2.8	prec_time	22
5.5.2.9	prec_wrt	22

5.5.2.10	sp	22
5.6	source Module Reference	22
5.6.1	Detailed Description	22
5.7	source_emitter Module Reference	22
5.7.1	Detailed Description	23
5.8	source_identity Module Reference	23
5.8.1	Detailed Description	23
5.8.2	Function/Subroutine Documentation	23
5.8.2.1	allocsources()	23
5.8.2.2	initialize()	24
5.8.2.3	printout()	24
5.8.3	Variable Documentation	25
5.8.3.1	source	25
5.9	tracer Module Reference	25
5.9.1	Detailed Description	25
5.10	tracer2d Module Reference	25
5.10.1	Detailed Description	25
5.11	tracer3d Module Reference	25
5.11.1	Detailed Description	26
5.11.2	Function/Subroutine Documentation	26
5.11.2.1	tracer_init()	26
5.12	tracer_interp Module Reference	27

6	Data Type Documentation	29
6.1	geometry::box Type Reference	29
6.1.1	Detailed Description	30
6.1.2	Member Data Documentation	30
6.1.2.1	size	30
6.2	simulation_globals::constants_t Type Reference	30
6.2.1	Detailed Description	30
6.2.2	Member Function/Subroutine Documentation	31
6.2.2.1	setgravity()	31
6.2.2.2	setrho()	31
6.2.3	Member Data Documentation	31
6.2.3.1	gravity	31
6.2.3.2	rho_ref	31
6.3	geometry::line Type Reference	32
6.3.1	Detailed Description	32
6.3.2	Member Data Documentation	32
6.3.2.1	last	33
6.4	simulation_globals::parameters_t Type Reference	33
6.4.1	Member Function/Subroutine Documentation	33
6.4.1.1	printout()	33
6.4.1.2	setparameter()	33
6.4.2	Member Data Documentation	34
6.4.2.1	cfl	34
6.4.2.2	integrator	34
6.4.2.3	timemax	34
6.4.2.4	timeout	34
6.4.2.5	warmuptime	34
6.5	geometry::point Type Reference	35
6.5.1	Detailed Description	35
6.6	geometry::shape Type Reference	36

6.6.1	Detailed Description	36
6.6.2	Member Data Documentation	36
6.6.2.1	pt	36
6.7	simulation_globals::simdefs_t Type Reference	36
6.7.1	Detailed Description	37
6.7.2	Member Function/Subroutine Documentation	37
6.7.2.1	printout()	37
6.7.2.2	setboundingbox()	37
6.7.2.3	setdp()	37
6.7.3	Member Data Documentation	37
6.7.3.1	dp	38
6.7.3.2	pointmax	38
6.7.3.3	pointmin	38
6.8	source_identity::source_class Type Reference	38
6.8.1	Detailed Description	39
6.8.2	Member Function/Subroutine Documentation	39
6.8.2.1	initialize()	39
6.8.2.2	printout()	39
6.8.3	Member Data Documentation	39
6.8.3.1	now	39
6.8.3.2	par	40
6.8.3.3	stats	40
6.9	source_identity::source_par Type Reference	40
6.9.1	Member Data Documentation	40
6.9.1.1	emitting_rate	40
6.9.1.2	geometry	41
6.9.1.3	id	41
6.9.1.4	name	41
6.9.1.5	property_name	41
6.9.1.6	source_geometry	41

6.10	source_identity::source_state Type Reference	41
6.10.1	Detailed Description	42
6.10.2	Member Data Documentation	42
6.10.2.1	active	42
6.10.2.2	age	42
6.10.2.3	depth	42
6.10.2.4	pos	43
6.10.2.5	t	43
6.10.2.6	vel	43
6.11	source_identity::source_stats Type Reference	43
6.11.1	Detailed Description	43
6.11.2	Member Data Documentation	43
6.11.2.1	acc_t	44
6.11.2.2	ns	44
6.11.2.3	particles_emitted	44
6.12	geometry::sphere Type Reference	44
6.12.1	Detailed Description	45
6.12.2	Member Data Documentation	45
6.12.2.1	radius	45
6.13	tracer3d::tracer_class Type Reference	46
6.13.1	Detailed Description	46
6.13.2	Member Data Documentation	46
6.13.2.1	now	46
6.13.2.2	par	47
6.13.2.3	stats	47
6.14	tracer3d::tracer_par_class Type Reference	47
6.14.1	Detailed Description	48
6.14.2	Member Data Documentation	48
6.14.2.1	group	48
6.14.2.2	id	48

6.14.2.3	interp_method	48
6.14.2.4	noise	48
6.14.2.5	tpar	48
6.14.2.6	vel_max	49
6.15	tracer3d::tracer_par_trans_class Type Reference	49
6.15.1	Member Data Documentation	49
6.15.1.1	par_trans_file	49
6.15.1.2	use_par_trans	49
6.16	tracer3d::tracer_state_class Type Reference	49
6.16.1	Detailed Description	50
6.16.2	Member Data Documentation	50
6.16.2.1	acc	50
6.16.2.2	active	50
6.16.2.3	age	50
6.16.2.4	depth	51
6.16.2.5	pos	51
6.16.2.6	t	51
6.16.2.7	vel	51
6.17	tracer3d::tracer_stats_class Type Reference	51
6.17.1	Detailed Description	52
6.17.2	Member Data Documentation	52
6.17.2.1	acc_depth	52
6.17.2.2	acc_pos	52
6.17.2.3	acc_t	52
6.17.2.4	acc_vel	52
6.17.2.5	ns	52

7	File Documentation	53
7.1	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/app/main.f90 File Reference . .	53
7.2	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_modules.f90 File Reference	53
7.3	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90 File Reference .	53
7.4	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90 File Reference .	54
7.5	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90 File Reference	54
7.6	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_precision.f90 File Reference	55
7.7	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source.f90 File Reference . .	56
7.8	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_emitter.f90 File Reference	56
7.9	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90 File Reference	56
7.10	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer.f90 File Reference . .	57
7.11	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90 File Reference .	57
7.12	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90 File Reference .	57
7.13	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_interp.f90 File Reference	58

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

commom_modules	Module to hold all of the commonly used base modules	9
geometry	Module that defines geometry classes and related methods	9
initialize	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	10
simulation_globals	Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics	16
simulation_precision	Module to control the precision of the variables trough the project	20
source	Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods	22
source_emitter	Module that defines an emitter class and related methods	22
source_identity	Module that defines a source class and related methods	23
tracer	Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. 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	25
tracer2d	Module that holds the functions for 2D tracer models - typically these are called to cancel the vertical component computed for a general tracer	25
tracer3d	Module that defines a pure Lagrangian tracer class and related methods	25
tracer_interp	27

Chapter 2

Data Type Index

2.1 Class Hierarchy

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

simulation_globals::constants_t	30
simulation_globals::parameters_t	33
geometry::shape	36
geometry::box	29
geometry::line	32
geometry::point	35
geometry::sphere	44
simulation_globals::simdefs_t	36
source_identity::source_class	38
source_identity::source_par	40
source_identity::source_state	41
source_identity::source_stats	43
tracer3d::tracer_class	46
tracer3d::tracer_par_class	47
tracer3d::tracer_par_trans_class	49
tracer3d::tracer_state_class	49
tracer3d::tracer_stats_class	51

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

geometry::box	
Type - point class	29
simulation_globals::constants_t	
Case Constants class	30
geometry::line	
Type - line class	32
simulation_globals::parameters_t	
geometry::point	
Type - point class	35
geometry::shape	
Type - extendable shape class	36
simulation_globals::simdefs_t	
Simulation definitions class	36
source_identity::source_class	
Type - The source class	38
source_identity::source_par	
source_identity::source_state	
Type - state variables of a source object	41
source_identity::source_stats	
Type - statistical variables of a source object	43
geometry::sphere	
Type - sphere class	44
tracer3d::tracer_class	
Type - The pure Lagrangian tracer class	46
tracer3d::tracer_par_class	
Type - parameters of a pure Lagrangian tracer object	47
tracer3d::tracer_par_trans_class	
tracer3d::tracer_state_class	
Type - state variables of a pure Lagrangian tracer object	49
tracer3d::tracer_stats_class	
Type - statistical variables of a pure Lagrangian tracer object	51

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/app/main.f90	53
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_modules.f90	53
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90	53
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90	54
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90	54
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_precision.f90	55
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source.f90	56
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_emitter.f90	56
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90	56
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer.f90	57
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90	57
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90	57
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_interp.f90	58

Chapter 5

Module Documentation

5.1 commom_modules Module Reference

Module to hold all of the commonly used base modules.

5.1.1 Detailed Description

Module to hold all of the commonly used base modules.

Author

Ricardo Birjukovs Canelas

5.2 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.

5.2.1 Detailed Description

Module that defines geometry classes and related methods.

Author

Ricardo Birjukovs Canelas

5.3 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 [readxmlatt](#) (xmlnode, tag, att_name, att_value)
Birjukovs Canelas - MARETEC
- subroutine [readxmlvector](#) (xmlnode, tag, vec)
Birjukovs Canelas - MARETEC
- subroutine [read_xml_geometry](#) (source, source_detail, geometry)
Birjukovs Canelas - MARETEC
- subroutine [init_sources](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [init_simdefs](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [init_caseconstants](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [init_parameters](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine, public [initmohidlagrangian](#) (xmlfilename)
Birjukovs Canelas - MARETEC

5.3.1 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

5.3.2 Function/Subroutine Documentation

5.3.2.1 [init_caseconstants\(\)](#)

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

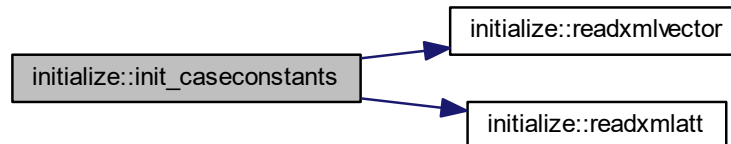
Birjukovs Canelas - MARETEC

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

Parameters

in	<i>parsedxml</i>	
in	<i>parsedxml</i>	.xml file handle

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.2 init_parameters()

```

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

Birjukovs Canelas - MARETEC

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

Parameters

in	<i>parsedxml</i>	
in	<i>parsedxml</i>	.xml file handle

Here is the caller graph for this function:



5.3.2.3 init_simdefs()

```

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

```

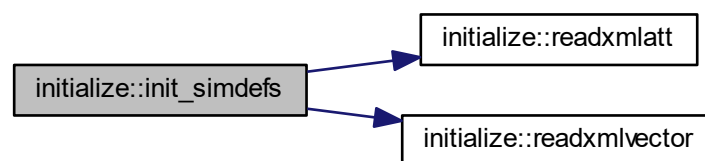
Birjukovs Canelas - MARETEC

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

Parameters

in	<i>parsedxml</i>	
in	<i>parsedxml</i>	.xml file handle

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.4 init_sources()

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

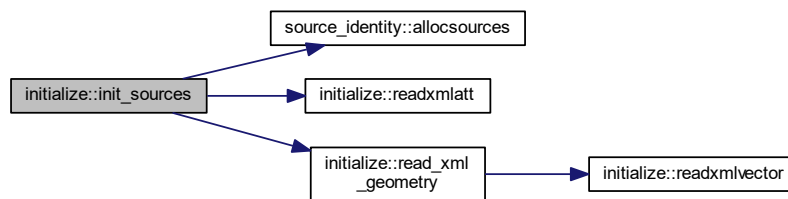
Birjukovs Canelas - MARETEC

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

Parameters

in	<i>parsedxml</i>	
in	<i>parsedxml</i>	.xml file handle

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.5 initmohidlagrangian()

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

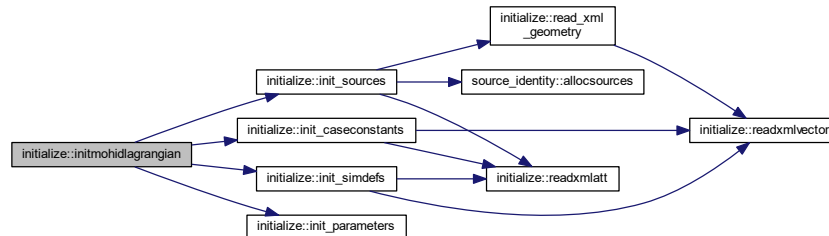
Birjukovs Canelas - MARETEC

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

Parameters

in	<i>xmlfilename</i>	
in	<i>xmlfilename</i>	.xml file name

Here is the call graph for this function:

**5.3.2.6 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]

```

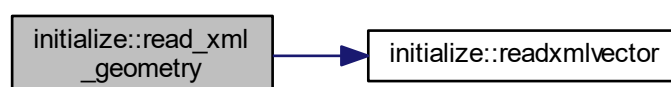
Birjukovs Canelas - MARETEC

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

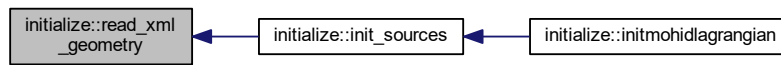
Parameters

in	<i>source,geometry</i>	
in	<i>source</i>	Working xml node
in	<i>source_detail</i>	Working xml node details

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.7 readxmlatt()

```

subroutine initialize::readxmlatt (
    type(node), intent(in), pointer xmlnode,
    type(string), intent(in) tag,
    type(string), intent(in) att_name,
    type(string), intent(out) att_value ) [private]
  
```

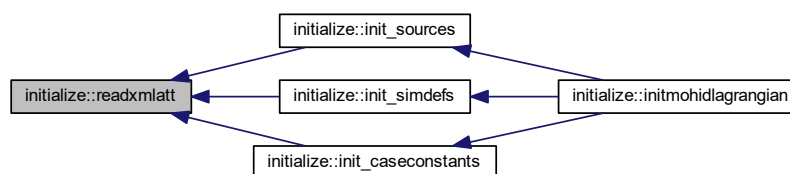
Birjukovs Canelas - MARETEC

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

Parameters

in	<i>xmlnode,tag,vec</i>	
in	<i>xmlnode</i>	Working xml node
in	<i>tag</i>	Tag to search in xml node
in	<i>att_name</i>	Atribute name to collect from tag
out	<i>att_value</i>	Attribute value

Here is the caller graph for this function:



5.3.2.8 readxmlvector()

```

subroutine initialize::readxmlvector (
    type(node), intent(in), pointer xmlnode,
  
```

```

type(string), intent(in) tag,
type(vector), intent(out) vec ) [private]

```

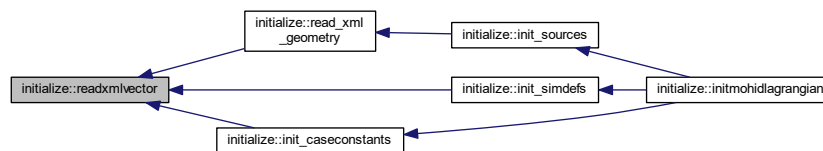
Birjukovs Canelas - MARETEC

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

Parameters

in	<i>xmlnode,tag,vec</i>	
in	<i>xmlnode</i>	Working xml node
in	<i>tag</i>	Tag to search in xml node
out	<i>vec</i>	Vector to fill with read contents

Here is the caller graph for this function:



5.4 simulation_globals Module Reference

Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics.

Data Types

- type [constants_t](#)
Case Constants class.
- type [parameters_t](#)
- type [simdefs_t](#)
Simulation definitions class.

Functions/Subroutines

- subroutine [setparameter](#) (parm, parmkey, parmvalue)
Birjukovs Canelas - MARETEC
- subroutine [printsimparameters](#) (parm)
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 [setboundingbox](#) (self, point_, coords)
Birjukovs Canelas - MARETEC
- subroutine [printsimdefs](#) (self)
Birjukovs Canelas - MARETEC

Variables

- type([parameters_t](#)), public [parameters](#)
- type([simdefs_t](#)), public [simdefs](#)
- type([constants_t](#)), public [constants](#)

5.4.1 Detailed Description

Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics.

Author

Ricardo Birjukovs Canelas

5.4.2 Function/Subroutine Documentation

5.4.2.1 [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:



5.4.2.2 printsimdefs()

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

Birjukovs Canelas - MARETEC

Public simulation definitions printing routine.

5.4.2.3 printsimparameters()

```
subroutine simulation_globals::printsimparameters (
    class(parameters_t) parm ) [private]
```

Birjukovs Canelas - MARETEC

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



5.4.2.4 setboundingbox()

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

Birjukovs Canelas - MARETEC

Public bounding box setting routine.

Parameters

in	<i>point_,coords</i>	
----	----------------------	--

5.4.2.5 setdp()

```
subroutine simulation_globals::setdp (
```

```
class(simdefs_t) self,
type(string), intent(in) read_dp ) [private]
```

Birjukovs Canelas - MARETEC

Public dp setting routine.

Parameters

in	<i>read_dp</i>	
----	----------------	--

5.4.2.6 setgravity()

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

Birjukovs Canelas - MARETEC

Public Gravity setting routine.

Parameters

in	<i>grav</i>	
----	-------------	--

5.4.2.7 setparameter()

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

Birjukovs Canelas - MARETEC

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

Parameters

in	<i>parmkey,parmvalue</i>	
----	--------------------------	--

5.4.2.8 setrho()

```
subroutine simulation_globals::setrho (
```

```
class(constants_t) self,
type(string), intent(in) read_rho ) [private]
```

Birjukovs Canelas - MARETEC

Public Rho_Ref setting routine.

Parameters

in	read_rho	
----	----------	--

5.4.3 Variable Documentation

5.4.3.1 constants

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

5.4.3.2 parameters

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

5.4.3.3 simdefs

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

5.5 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

5.5.1 Detailed Description

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

Author

Ricardo Birjukovs Canelas

5.5.2 Variable Documentation

5.5.2.1 dp

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

Double precision definition switch.

5.5.2.2 err_dist

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

5.5.2.3 err_ind

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

5.5.2.4 missing_value_default

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

5.5.2.5 mv

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

5.5.2.6 mv_int

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

5.5.2.7 prec

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

5.5.2.8 prec_time

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

5.5.2.9 prec_wrt

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

5.5.2.10 sp

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

Simple precision definition switch.

5.6 source Module Reference

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

5.6.1 Detailed Description

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

Author

Ricardo Birjukovs Canelas

5.7 source_emitter Module Reference

Module that defines an emitter class and related methods.

5.7.1 Detailed Description

Module that defines an emitter class and related methods.

Author

Ricardo Birjukovs Canelas

5.8 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.

Functions/Subroutines

- subroutine, public [allocsources](#) (nsources)
Birjukovs Canelas - MARETEC
- subroutine [initialize](#) (src, id, name, emitting_rate, source_geometry, geometry)
Birjukovs Canelas - MARETEC
- subroutine [printout](#) (src)
Birjukovs Canelas - MARETEC

Variables

- type([source_class](#)), dimension(:), allocatable, public [source](#)

5.8.1 Detailed Description

Module that defines a source class and related methods.

Author

Ricardo Birjukovs Canelas

5.8.2 Function/Subroutine Documentation

5.8.2.1 allocsources()

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

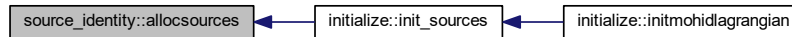
Birjukovs Canelas - MARETEC

source allocation routine - allocates the sources objects

Parameters

in	<i>nsources</i>	
----	-----------------	--

Here is the caller graph for this function:

**5.8.2.2 initialize()**

```

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

Birjukovs Canelas - MARETEC

source initialization procedure - initializes Source variables

Parameters

in	<i>src,id,name,emitting_rate,source_geometry</i>	
----	--	--

5.8.2.3 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	<i>src</i>	
----	------------	--

5.8.3 Variable Documentation

5.8.3.1 source

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

5.9 tracer Module Reference

Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. 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.

5.9.1 Detailed Description

Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. 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

5.10 tracer2d Module Reference

Module that holds the functions for 2D tracer models - typically these are called to cancel the vertical component computed for a general tracer.

5.10.1 Detailed Description

Module that holds the functions for 2D tracer models - typically these are called to cancel the vertical component computed for a general tracer.

Author

Ricardo Birjukovs Canelas

5.11 tracer3d 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 - parameters of a pure Lagrangian tracer object.
- type `tracer_par_trans_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, public `tracer_init` (trc, id, time, x, y, z)
Birjukovs Canelas - MARETEC

5.11.1 Detailed Description

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

Author

Ricardo Birjukovs Canelas

5.11.2 Function/Subroutine Documentation

5.11.2.1 `tracer_init()`

```
subroutine, public tracer3d::tracer_init (
    type(tracer_class), intent(inout) trc,
    integer, intent(in) id,
    real(prec_time), intent(in) time,
    real(prec), intent(in) x,
    real(prec), intent(in) y,
    real(prec), intent(in) z )
```

Birjukovs Canelas - MARETEC

Tracer initialization routine - Generates a tracer and initializes its variables

Parameters

out	<i>trc</i>	
in	<i>filename</i>	

5.12 tracer_interp Module Reference

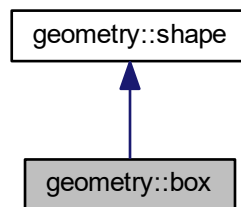
Chapter 6

Data Type Documentation

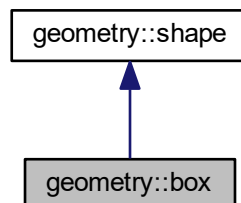
6.1 geometry::box Type Reference

Type - point class.

Inheritance diagram for geometry::box:



Collaboration diagram for geometry::box:



Public 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
```

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)

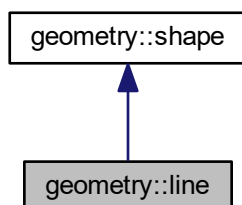
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[simulation_globals.f90](#)

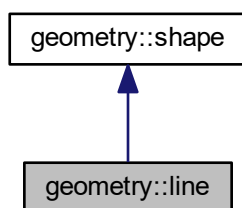
6.3 geometry::line Type Reference

Type - line class.

Inheritance diagram for geometry::line:



Collaboration diagram for geometry::line:



Public Attributes

- type(vector) [last](#)
Coordinates of the end point.

6.3.1 Detailed Description

Type - line class.

6.3.2 Member Data Documentation

6.3.2.1 last

```
type(vector) geometry::line::last
```

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](#)

6.4 simulation_globals::parameters_t Type Reference

Private Member Functions

- procedure [setparameter](#)
- procedure [printout](#) => [printsimpparameters](#)

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)

6.4.1 Member Function/Subroutine Documentation

6.4.1.1 printout()

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

6.4.1.2 setparameter()

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

6.4.2 Member Data Documentation

6.4.2.1 cfl

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

Courant Friedrichs Lewy condition number.

6.4.2.2 integrator

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

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

6.4.2.3 timemax

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

Simulation duration (s)

6.4.2.4 timeout

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

Time out data (1/Hz)

6.4.2.5 warmuptime

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

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

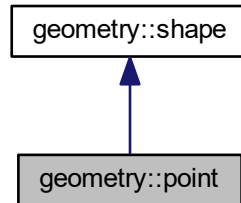
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[simulation_globals.f90](#)

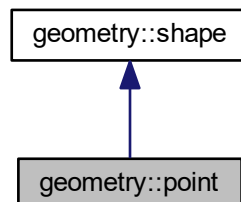
6.5 geometry::point Type Reference

Type - point class.

Inheritance diagram for geometry::point:



Collaboration diagram for geometry::point:



Additional Inherited Members

6.5.1 Detailed Description

Type - point class.

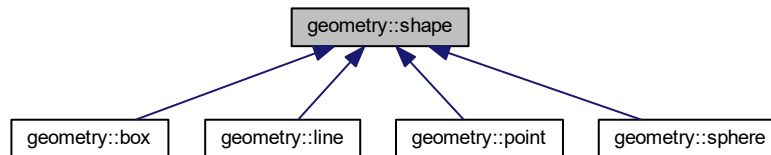
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[geometry.f90](#)

6.6 geometry::shape Type Reference

Type - extendable shape class.

Inheritance diagram for geometry::shape:



Public Attributes

- type(vector) [pt](#)
Coordinates of a point.

6.6.1 Detailed Description

Type - extendable shape class.

6.6.2 Member Data Documentation

6.6.2.1 pt

type(vector) geometry::shape::pt

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](#)

6.7 simulation_globals::simdefs_t Type Reference

Simulation definitions class.

Private Member Functions

- procedure [setdp](#)
- procedure [setboundingbox](#)
- procedure [printout](#) => [printsimdefs](#)

Private Attributes

- real(prec) [dp](#) = MV
Initial particle spacing at source generation.
- 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.

6.7.1 Detailed Description

Simulation definitions class.

6.7.2 Member Function/Subroutine Documentation

6.7.2.1 [printout\(\)](#)

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

6.7.2.2 [setboundingbox\(\)](#)

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

6.7.2.3 [setdp\(\)](#)

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

6.7.3 Member Data Documentation

6.7.3.1 dp

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

Initial particle spacing at source generation.

6.7.3.2 pointmax

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

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

6.7.3.3 pointmin

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

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

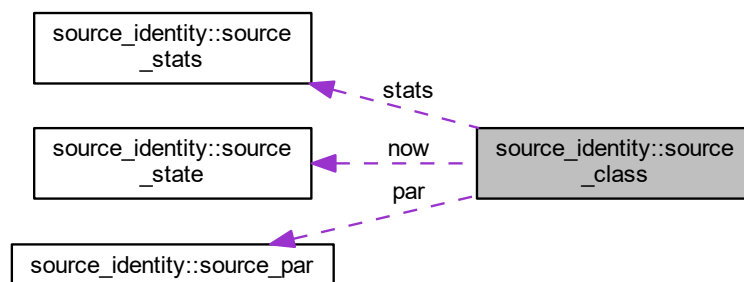
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[simulation_globals.f90](#)

6.8 source_identity::source_class Type Reference

Type - The source class.

Collaboration diagram for source_identity::source_class:



Private Member Functions

- procedure [initialize](#)
- procedure [printout](#)

Private Attributes

- type([source_par](#)) [par](#)
To access parameters.
- type([source_state](#)) [now](#)
To access state variables.
- type([source_stats](#)) [stats](#)
To access statistics.

6.8.1 Detailed Description

Type - The source class.

6.8.2 Member Function/Subroutine Documentation

6.8.2.1 initialize()

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

6.8.2.2 printout()

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

6.8.3 Member Data Documentation

6.8.3.1 now

```
type(source\_state) source_identity::source_class::now [private]
```

To access state variables.

6.8.3.2 par

```
type(source\_par) source_identity::source_class::par [private]
```

To access parameters.

6.8.3.3 stats

```
type(source\_stats) source_identity::source_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/[source_identity.f90](#)

6.9 source_identity::source_par Type Reference

Private Attributes

- integer [id](#)
unique source identification (integer)
- real(prec) [emitting_rate](#)
Emitting rate of the source (Hz)
- type(string) [name](#)
source name
- type(string) [property_name](#)
source property name
- type(string) [source_geometry](#)
Source type : 'point', 'line', 'sphere', 'box'.
- class([shape](#)), allocatable [geometry](#)
Source geometry.

6.9.1 Member Data Documentation

6.9.1.1 emitting_rate

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

Emitting rate of the source (Hz)

6.9.1.2 geometry

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

Source geometry.

6.9.1.3 id

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

unique source identification (integer)

6.9.1.4 name

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

source name

6.9.1.5 property_name

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

source property name

6.9.1.6 source_geometry

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

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

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[source_identity.f90](#)

6.10 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)

6.10.1 Detailed Description

Type - state variables of a source object.

6.10.2 Member Data Documentation

6.10.2.1 [active](#)

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

[active](#) switch

6.10.2.2 [age](#)

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

6.10.2.3 [depth](#)

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

[Depth](#) of the source baricenter (m)

6.10.2.4 pos

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

Position of the source baricenter (m)

6.10.2.5 t

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

Temperature of the source (Celcius)

6.10.2.6 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](#)

6.11 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.

6.11.1 Detailed Description

Type - statistical variables of a source object.

6.11.2 Member Data Documentation

6.11.2.1 `acc_t`

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

Accumulated temperature of the tracer (Celcius)

6.11.2.2 `ns`

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

Number of sampling steps.

6.11.2.3 `particles_emitted`

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

Number of emitted particles by this source.

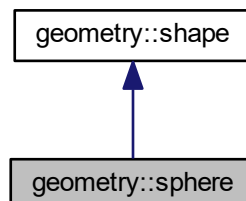
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[source_identity.f90](#)

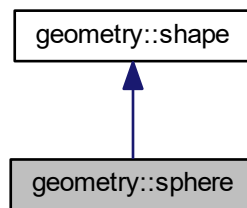
6.12 `geometry::sphere` Type Reference

Type - sphere class.

Inheritance diagram for `geometry::sphere`:



Collaboration diagram for geometry::sphere:



Public Attributes

- `real(prec) radius`
Sphere radius.

6.12.1 Detailed Description

Type - sphere class.

6.12.2 Member Data Documentation

6.12.2.1 radius

```
real(prec) geometry::sphere::radius
```

Sphere radius.

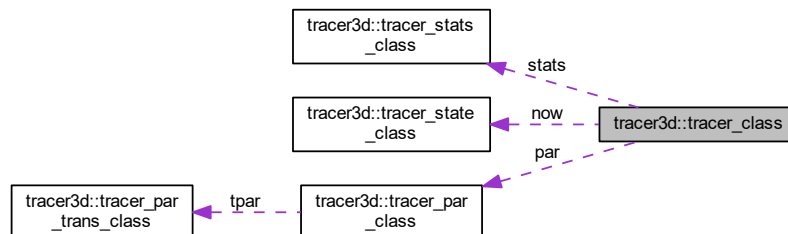
The documentation for this type was generated from the following file:

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

6.13 tracer3d::tracer_class Type Reference

Type - The pure Lagrangian tracer class.

Collaboration diagram for tracer3d::tracer_class:



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.

6.13.1 Detailed Description

Type - The pure Lagrangian tracer class.

6.13.2 Member Data Documentation

6.13.2.1 now

```
type(tracer_state_class) tracer3d::tracer_class::now [private]
```

To access state variables.

6.13.2.2 par

```
type(tracer_par_class) tracer3d::tracer_class::par [private]
```

To access parameters.

6.13.2.3 stats

```
type(tracer_stats_class) tracer3d::tracer_class::stats [private]
```

To access statistics.

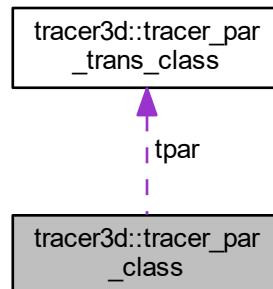
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90

6.14 tracer3d::tracer_par_class Type Reference

Type - parameters of a pure Lagrangian tracer object.

Collaboration diagram for tracer3d::tracer_par_class:



Private Attributes

- integer `id`
unique tracer identification (integer)
- integer `group`
Group to which the tracer belongs (usually by source)
- real(prec) `vel_max`
Maximum velocity of tracer to track (m/s)
- logical `noise`
- character(len=56) `interp_method`
interpolation method this tracer calls
- type(tracer_par_trans_class) `tpar`
access to the transient parameters is done through this

6.14.1 Detailed Description

Type - parameters of a pure Lagrangian tracer object.

6.14.2 Member Data Documentation

6.14.2.1 group

```
integer tracer3d::tracer_par_class::group [private]
```

Group to which the tracer belongs (usually by source)

6.14.2.2 id

```
integer tracer3d::tracer_par_class::id [private]
```

unique tracer identification (integer)

6.14.2.3 interp_method

```
character(len=56) tracer3d::tracer_par_class::interp_method [private]
```

interpolation method this tracer calls

6.14.2.4 noise

```
logical tracer3d::tracer_par_class::noise [private]
```

6.14.2.5 tpar

```
type(tracer_par_trans_class) tracer3d::tracer_par_class::tpar [private]
```

access to the transient parameters is done through this

6.14.2.6 vel_max

```
real(prec) tracer3d::tracer_par_class::vel_max [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/[tracer3D.f90](#)

6.15 tracer3d::tracer_par_trans_class Type Reference

Private Attributes

- character(len=512) [par_trans_file](#)
- logical [use_par_trans](#)

6.15.1 Member Data Documentation

6.15.1.1 par_trans_file

```
character(len=512) tracer3d::tracer_par_trans_class::par_trans_file [private]
```

6.15.1.2 use_par_trans

```
logical tracer3d::tracer_par_trans_class::use_par_trans [private]
```

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[tracer3D.f90](#)

6.16 tracer3d::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)
- `real(prec)` [t](#)
Temperature of the tracer (Celcius)

6.16.1 Detailed Description

Type - state variables of a pure Lagrangian tracer object.

6.16.2 Member Data Documentation

6.16.2.1 `acc`

```
type(vector) tracer3d::tracer_state_class::acc [private]
```

Acceleration of the tracer (m s-2)

6.16.2.2 `active`

```
logical tracer3d::tracer_state_class::active [private]
```

active switch

6.16.2.3 `age`

```
real(prec_time) tracer3d::tracer_state_class::age [private]
```

6.16.2.4 depth

```
real(prec) tracer3d::tracer_state_class::depth [private]
```

Depth of the tracer (m)

6.16.2.5 pos

```
type(vector) tracer3d::tracer_state_class::pos [private]
```

Position of the tracer (m)

6.16.2.6 t

```
real(prec) tracer3d::tracer_state_class::t [private]
```

Temperature of the tracer (Celcius)

6.16.2.7 vel

```
type(vector) tracer3d::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/[tracer3D.f90](#)

6.17 tracer3d::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)
- real(prec_wrt) [acc_t](#)
Accumulated temperature of the tracer (Celcius)
- integer [ns](#)
Number of sampling steps.

6.17.1 Detailed Description

Type - statistical variables of a pure Lagrangian tracer object.

6.17.2 Member Data Documentation

6.17.2.1 `acc_depth`

```
real(prec_wrt) tracer3d::tracer_stats_class::acc_depth [private]
```

Accumulated depth of the tracer (m)

6.17.2.2 `acc_pos`

```
type(vector) tracer3d::tracer_stats_class::acc_pos [private]
```

Accumulated position of the tracer (m)

6.17.2.3 `acc_t`

```
real(prec_wrt) tracer3d::tracer_stats_class::acc_t [private]
```

Accumulated temperature of the tracer (Celcius)

6.17.2.4 `acc_vel`

```
type(vector) tracer3d::tracer_stats_class::acc_vel [private]
```

Accumulated velocity of the tracer (m s-1)

6.17.2.5 `ns`

```
integer tracer3d::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/[tracer3D.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/common_↵ modules.f90 File Reference

Modules

- module [common_modules](#)
Module to hold all of the commonly used base modules.

7.3 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.

7.4 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::readxmlatt](#) (xmlnode, tag, att_name, att_value)
Birjukovs Canelas - MARETEC
- subroutine [initialize::readxmlvector](#) (xmlnode, tag, vec)
Birjukovs Canelas - MARETEC
- subroutine [initialize::read_xml_geometry](#) (source, source_detail, geometry)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_sources](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_simdefs](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_caseconstants](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_parameters](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine, public [initialize::initmohidlagrangian](#) (xmlfilename)
Birjukovs Canelas - MARETEC

7.5 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.

Modules

- module [simulation_globals](#)

Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics.

Functions/Subroutines

- subroutine [simulation_globals::setparameter](#) (parm, parmkey, parmvalue)
Birjukovs Canelas - MARETEC
- subroutine [simulation_globals::printsimpparameters](#) (parm)
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::setboundingbox](#) (self, point_, coords)
Birjukovs Canelas - MARETEC
- subroutine [simulation_globals::printsimdefs](#) (self)
Birjukovs Canelas - MARETEC

Variables

- type(parameters_t), public [simulation_globals::parameters](#)
- type(simdefs_t), public [simulation_globals::simdefs](#)
- type(constants_t), public [simulation_globals::constants](#)

7.6 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

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

Modules

- module [source](#)

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

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

Modules

- module [source_emitter](#)

Module that defines an emitter class and related methods.

7.9 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_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 [source_identity::initialize](#) (src, id, name, emitting_rate, source_geometry, geometry)
Birjukovs Canelas - MARETEC
- subroutine [source_identity::printout](#) (src)
Birjukovs Canelas - MARETEC

Variables

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

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

Modules

- module [tracer](#)

Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. 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.

7.11 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90 File Reference

Modules

- module [tracer2d](#)

Module that holds the functions for 2D tracer models - typically these are called to cancel the vertical component computed for a general tracer.

7.12 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90 File Reference

Data Types

- type [tracer3d::tracer_par_trans_class](#)
- type [tracer3d::tracer_par_class](#)
Type - parameters of a pure Lagrangian tracer object.
- type [tracer3d::tracer_state_class](#)
Type - state variables of a pure Lagrangian tracer object.
- type [tracer3d::tracer_stats_class](#)
Type - statistical variables of a pure Lagrangian tracer object.
- type [tracer3d::tracer_class](#)
Type - The pure Lagrangian tracer class.

Modules

- module [tracer3d](#)

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

Functions/Subroutines

- subroutine, public [tracer3d::tracer_init](#) (trc, id, time, x, y, z)
Birjukovs Canelas - MARETEC

7.13 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_[↩](#) interp.f90 File Reference

Modules

- module [tracer_interp](#)