

# Global Groundwater Model

Generated by Doxygen 1.8.8

Thu Nov 16 2017 09:25:09



# Contents

<b>1</b>	<b>Deprecated List</b>	<b>1</b>
<b>2</b>	<b>Bug List</b>	<b>3</b>
<b>3</b>	<b>Namespace Index</b>	<b>5</b>
3.1	Namespace List . . . . .	5
<b>4</b>	<b>Hierarchical Index</b>	<b>7</b>
4.1	Class Hierarchy . . . . .	7
<b>5</b>	<b>Class Index</b>	<b>9</b>
5.1	Class List . . . . .	9
<b>6</b>	<b>Namespace Documentation</b>	<b>11</b>
6.1	GlobalFlow Namespace Reference . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.1.2	Enumeration Type Documentation . . . . .	11
6.1.2.1	Field . . . . .	11
<b>7</b>	<b>Class Documentation</b>	<b>13</b>
7.1	A Class Reference . . . . .	13
7.1.1	Detailed Description . . . . .	13
7.2	Abstract Class Reference . . . . .	13
7.2.1	Detailed Description . . . . .	13
7.3	GlobalFlow::Simulation::AbstractStepper Class Reference . . . . .	13
7.3.1	Detailed Description . . . . .	13
7.4	GlobalFlow::Solver::AdaptiveDamping Class Reference . . . . .	14
7.4.1	Detailed Description . . . . .	14
7.5	GlobalFlow::Callback Class Reference . . . . .	14
7.5.1	Detailed Description . . . . .	14
7.6	GlobalFlow::Container Class Reference . . . . .	14
7.6.1	Detailed Description . . . . .	14
7.7	GlobalFlow::DataProcessing::DataOutput::CSVOutput< T > Class Template Reference . . . . .	15
7.7.1	Detailed Description . . . . .	15

7.8	<a href="#">GlobalFlow::DataReader Class Reference</a>	15
7.8.1	<a href="#">Detailed Description</a>	16
7.8.2	<a href="#">Member Function Documentation</a>	16
7.8.2.1	<a href="#">loopFiles</a>	16
7.8.2.2	<a href="#">readData</a>	16
7.8.2.3	<a href="#">readTwoColumns</a>	16
7.8.2.4	<a href="#">readZeroPointFiveToFiveMin</a>	16
7.9	<a href="#">GlobalFlow::Solver::Equation Class Reference</a>	17
7.9.1	<a href="#">Detailed Description</a>	17
7.9.2	<a href="#">Member Function Documentation</a>	17
7.9.2.1	<a href="#">getError</a>	17
7.9.2.2	<a href="#">getItttr</a>	17
7.9.2.3	<a href="#">solve</a>	18
7.9.2.4	<a href="#">toogleSteadyState</a>	18
7.9.2.5	<a href="#">updateStepSize</a>	18
7.9.3	<a href="#">Friends And Related Function Documentation</a>	18
7.9.3.1	<a href="#">operator&lt;&lt;</a>	18
7.10	<a href="#">GlobalFlow::Model::ExternalFlow Class Reference</a>	18
7.10.1	<a href="#">Detailed Description</a>	19
7.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	19
7.10.2.1	<a href="#">ExternalFlow</a>	19
7.10.2.2	<a href="#">ExternalFlow</a>	19
7.10.3	<a href="#">Member Function Documentation</a>	20
7.10.3.1	<a href="#">flowIsHeadDependant</a>	20
7.10.3.2	<a href="#">getP</a>	21
7.10.3.3	<a href="#">getQ</a>	21
7.11	<a href="#">Factory Class Reference</a>	21
7.11.1	<a href="#">Detailed Description</a>	21
7.12	<a href="#">GlobalFlow::DataProcessing::DataOutput::FieldCollector Class Reference</a>	22
7.12.1	<a href="#">Detailed Description</a>	22
7.12.2	<a href="#">Member Function Documentation</a>	22
7.12.2.1	<a href="#">get</a>	22
7.13	<a href="#">GlobalFlow::DataProcessing::DataOutput::FieldFactory Class Reference</a>	22
7.13.1	<a href="#">Detailed Description</a>	23
7.13.2	<a href="#">Member Function Documentation</a>	23
7.13.2.1	<a href="#">build</a>	23
7.14	<a href="#">GlobalFlow::Model::FlowTypeHash Struct Reference</a>	23
7.14.1	<a href="#">Detailed Description</a>	23
7.15	<a href="#">GlobalFlow::Model::FluidMechanics Class Reference</a>	23
7.15.1	<a href="#">Detailed Description</a>	24

7.15.2	Member Function Documentation	24
7.15.2.1	calcDeltaV	24
7.15.2.2	calculateHarmonicMeanConductance	24
7.15.2.3	calculateVerticalConductance	24
7.15.2.4	getDerivate__NWT	25
7.15.2.5	getHCOF	25
7.15.2.6	smoothFunction__NWT	25
7.16	GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T > Class Template Reference	26
7.16.1	Detailed Description	26
7.16.2	Member Function Documentation	26
7.16.2.1	write	26
7.17	GlobalFlow::GW_Interface Class Reference	26
7.17.1	Detailed Description	27
7.17.2	Member Function Documentation	27
7.17.2.1	loadSettings	27
7.17.2.2	setupCallBack	27
7.17.2.3	setupSimulation	27
7.17.2.4	simulate	27
7.17.2.5	writeData	27
7.18	std::hash< GlobalFlow::Model::NeighbourPosition > Struct Template Reference	27
7.18.1	Detailed Description	28
7.19	GlobalFlow::Logging::InfoSinkInterface Class Reference	28
7.19.1	Detailed Description	28
7.20	Interface Interface Reference	28
7.20.1	Detailed Description	28
7.21	GlobalFlow::DataProcessing::DataOutput::InternalTypeHash Struct Reference	29
7.21.1	Detailed Description	29
7.22	Is< T > Struct Template Reference	29
7.22.1	Detailed Description	29
7.23	GlobalFlow::Simulation::Iterator Class Reference	29
7.23.1	Detailed Description	30
7.24	GlobalFlow::Logging::Logger Class Reference	30
7.24.1	Detailed Description	30
7.25	GlobalFlow::Logging::LoggerInterface Class Reference	30
7.25.1	Detailed Description	30
7.26	GlobalFlow::Simulation::MassError Class Reference	30
7.26.1	Detailed Description	31
7.27	GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T > Class Template Reference	31
7.27.1	Detailed Description	31
7.27.2	Member Function Documentation	31

7.27.2.1	write	31
7.28	GlobalFlow::Model::NodeInterface Class Reference	32
7.28.1	Detailed Description	35
7.28.2	Constructor & Destructor Documentation	35
7.28.2.1	NodeInterface	35
7.28.3	Member Function Documentation	36
7.28.3.1	addExternalFlow	36
7.28.3.2	calcLateralFlows	36
7.28.3.3	calculateDewateredFlow	36
7.28.3.4	calculateExternalFlowVolume	36
7.28.3.5	calculateNotHeadDependandFlows	37
7.28.3.6	getConductance	37
7.28.3.7	getCurrentIN	37
7.28.3.8	getCurrentOUT	37
7.28.3.9	getEqFlow	38
7.28.3.10	getExternalFlowByName	38
7.28.3.11	getExternalFlowVolumeByName	38
7.28.3.12	getFlow	38
7.28.3.13	getJacobian	38
7.28.3.14	getK	39
7.28.3.15	getK_vertical	39
7.28.3.16	getLateralFlows	39
7.28.3.17	getNeighbour	39
7.28.3.18	getNonStorageFlow	39
7.28.3.19	getP	40
7.28.3.20	getQ	40
7.28.3.21	getRHS	40
7.28.3.22	getRHS__NWT	40
7.28.3.23	getStorageCapacity	40
7.28.3.24	getStorageCapacity__Primary	40
7.28.3.25	getStorageCapacity__Secondary	41
7.28.3.26	getStorageCapacity__SecondaryNWT	41
7.28.3.27	getTotalStorageFlow	41
7.28.3.28	getVelocity	41
7.28.3.29	getVelocityVector	42
7.28.3.30	hasOcean	42
7.28.3.31	hasRiver	42
7.28.3.32	hasTypeOfExternalFlow	42
7.28.3.33	removeExternalFlow	42
7.28.3.34	resetFloodingHead	43

7.28.3.35 scaleDynamicRivers . . . . .	43
7.28.3.36 scaleRiverConduct . . . . .	43
7.28.3.37 setEfold . . . . .	43
7.28.3.38 setElevation . . . . .	43
7.28.3.39 setEqHead . . . . .	43
7.28.3.40 setK . . . . .	44
7.28.3.41 setK_direct . . . . .	44
7.28.3.42 setNeighbour . . . . .	44
7.28.3.43 setSlope . . . . .	44
7.28.3.44 toggleStadyState . . . . .	44
7.28.3.45 updateExternalFlowConduct . . . . .	45
7.28.3.46 updateHeadChange . . . . .	46
7.28.3.47 updateLakeBottoms . . . . .	46
7.28.3.48 updateUniqueFlow . . . . .	46
7.28.4 Member Data Documentation . . . . .	46
7.28.4.1 cached . . . . .	46
7.29 GlobalFlow::Model::NodeInterface::NodeNotFoundException Class Reference . . . . .	46
7.29.1 Detailed Description . . . . .	47
7.30 GlobalFlow::Simulation::Options Class Reference . . . . .	47
7.30.1 Detailed Description . . . . .	48
7.31 GlobalFlow::DataProcessing::DataOutput::OutputFactory< T > Class Template Reference . . . . .	48
7.31.1 Detailed Description . . . . .	48
7.32 GlobalFlow::DataProcessing::DataOutput::OutputField< T > Class Template Reference . . . . .	49
7.32.1 Detailed Description . . . . .	49
7.33 GlobalFlow::DataProcessing::DataOutput::OutputInterface< T > Class Template Reference . . . . .	49
7.33.1 Detailed Description . . . . .	49
7.33.2 Member Function Documentation . . . . .	49
7.33.2.1 write . . . . .	49
7.34 GlobalFlow::DataProcessing::DataOutput::OutputManager Class Reference . . . . .	50
7.34.1 Detailed Description . . . . .	50
7.35 GlobalFlow::DataProcessing::DataOutput::Outputvisitor Class Reference . . . . .	50
7.35.1 Detailed Description . . . . .	50
7.36 GlobalFlow::Model::PhysicalProperty< Type, Key > Class Template Reference . . . . .	50
7.36.1 Detailed Description . . . . .	51
7.37 Position Class Reference . . . . .	51
7.37.1 Detailed Description . . . . .	51
7.38 PropertiyRepository Class Reference . . . . .	51
7.38.1 Detailed Description . . . . .	51
7.39 GlobalFlow::Model::PropertyRepository< PhysicalProperties > Class Template Reference . . . . .	51
7.39.1 Detailed Description . . . . .	52

7.40	SaticHeadNode Class Reference	52
7.40.1	Detailed Description	52
7.41	GlobalFlow::Simulation::Simulation Class Reference	52
7.41.1	Detailed Description	53
7.41.2	Member Function Documentation	53
7.41.2.1	getCurrentMassError	53
7.41.2.2	getError	53
7.41.2.3	getError	54
7.41.2.4	getFlowByName	54
7.41.2.5	getFlowSumByIDs	54
7.41.2.6	getLossToRivers	54
7.41.2.7	getMassError	55
7.41.2.8	NodeFlowsByID	55
7.41.2.9	NodeInfosByID	55
7.41.2.10	printMassBalances	55
7.41.2.11	readMultipliers	55
7.41.2.12	scaleByFunction	55
7.41.2.13	updateGWRechargeFromWaterGAP	56
7.41.2.14	writeResiduals	56
7.42	GlobalFlow::Simulation::SimulationOutput Class Reference	56
7.42.1	Detailed Description	57
7.43	GlobalFlow::Logging::Singleton< C > Class Template Reference	57
7.43.1	Detailed Description	57
7.44	GlobalFlow::Model::StandardNode Class Reference	57
7.44.1	Detailed Description	57
7.45	GlobalFlow::Model::StaticHeadNode Class Reference	58
7.45.1	Detailed Description	58
7.46	GlobalFlow::Simulation::Stepper Class Reference	58
7.46.1	Detailed Description	58
7.47	Write Class Reference	59
7.47.1	Detailed Description	59
7.48	Writes Class Reference	59
7.48.1	Detailed Description	59
7.49	Writes Class Reference	59
7.49.1	Detailed Description	59
7.50	Writes Interface Reference	59
7.50.1	Detailed Description	59



# Chapter 1

## Deprecated List

### Namespace [GlobalFlow](#)

use methods from DataProcessing

### Member [GlobalFlow::GW\\_Interface::writeData \(\)=0](#)

### Class [GlobalFlow::Logging::InfoSinkInterface](#)

Use methods from DataProcessing



## Chapter 2

# Bug List

Namespace [GlobalFlow](#)

Some conversions make no sense



## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">GlobalFlow</a> . . . . .	11
--------------------------------------	----



## Chapter 4

# Hierarchical Index

### 4.1 Class Hierarchy

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

A . . . . .	13
Abstract . . . . .	13
GlobalFlow::Simulation::AbstractStepper . . . . .	13
GlobalFlow::Simulation::Stepper . . . . .	58
GlobalFlow::Solver::AdaptiveDamping . . . . .	14
basic_sink_backend	
GlobalFlow::Logging::InfoSinkInterface . . . . .	28
GlobalFlow::Callback . . . . .	14
GlobalFlow::Container . . . . .	14
GlobalFlow::DataReader . . . . .	15
GlobalFlow::Solver::Equation . . . . .	17
exception	
GlobalFlow::Model::NodeInterface::NodeNotFoundException . . . . .	46
GlobalFlow::Model::ExternalFlow . . . . .	18
Factory . . . . .	21
GlobalFlow::DataProcessing::DataOutput::FieldCollector . . . . .	22
GlobalFlow::DataProcessing::DataOutput::FieldFactory . . . . .	22
GlobalFlow::Model::FlowTypeHash . . . . .	23
GlobalFlow::Model::FluidMechanics . . . . .	23
GlobalFlow::GW_Interface . . . . .	26
std::hash< GlobalFlow::Model::NeighbourPosition > . . . . .	27
Interface . . . . .	28
GlobalFlow::DataProcessing::DataOutput::InternalTypeHash . . . . .	29
Is< T > . . . . .	29
GlobalFlow::Simulation::Iterator . . . . .	29
GlobalFlow::Simulation::MassError . . . . .	30
GlobalFlow::Model::NodeInterface . . . . .	32
GlobalFlow::Model::StandardNode . . . . .	57
GlobalFlow::Model::StaticHeadNode . . . . .	58
GlobalFlow::Simulation::Options . . . . .	47
GlobalFlow::DataProcessing::DataOutput::OutputFactory< T > . . . . .	48
GlobalFlow::DataProcessing::DataOutput::OutputField< T > . . . . .	49
GlobalFlow::DataProcessing::DataOutput::OutputInterface< T > . . . . .	49
GlobalFlow::DataProcessing::DataOutput::CSVOutput< T > . . . . .	15
GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T > . . . . .	26
GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T > . . . . .	31
GlobalFlow::DataProcessing::DataOutput::OutputManager . . . . .	50

## PhysicalProperties

GlobalFlow::Model::PropertyRepository< PhysicalProperty< long, ID >, PhysicalProperty< long, ArcID >, PhysicalProperty< double, Lat >, PhysicalProperty< double, Lon >, PhysicalProperty< int, Layer >, PhysicalProperty< t_dim, StepModifier >, PhysicalProperty< t_s_meter, Area >, PhysicalProperty< t_meter, VerticalSize >, PhysicalProperty< t_meter, Elevation >, PhysicalProperty< t_meter, TopElevation >, PhysicalProperty< t_dim, Slope >, PhysicalProperty< t_meter, EFolding >, PhysicalProperty< bool, Confinement >, PhysicalProperty< t_vel, K >, PhysicalProperty< t_dim, Anisotropy >, PhysicalProperty< quantity< d_time >, StepSize >, PhysicalProperty< t_c_meter, OUT >, PhysicalProperty< t_c_meter, IN >, PhysicalProperty< t_meter, Head >, PhysicalProperty< t_meter, EQHead >, PhysicalProperty< t_meter, HeadChange >, PhysicalProperty< t_meter, Head_TZero >, PhysicalProperty< t_meter, HeadChange_TZero >, PhysicalProperty< t_dim, SpecificYield >, PhysicalProperty< quantity< perUnit >, SpecificStorage >, PhysicalProperty< t_meter, EdgeLenght >, PhysicalProperty< t_s_meter, SideSurface >, PhysicalProperty< t_c_meter, VolumeOfCell > >	51
GlobalFlow::Model::PropertyRepository< PhysicalProperties >	51
GlobalFlow::Model::PhysicalProperty< Type, Key >	50
Position	51
PropertyRepository	51
SaticHeadNode	52
GlobalFlow::Simulation::Simulation	52
GlobalFlow::Simulation::SimulationOutput	56
GlobalFlow::Logging::Singleton< C >	57
GlobalFlow::Logging::Singleton< LoggerInterface >	57
GlobalFlow::Logging::LoggerInterface	30
GlobalFlow::Logging::Logger	30
static_visitor	
GlobalFlow::DataProcessing::DataOutput::Outputvisitor	50
Write	59
Writes	59
Writes	59
Writes	59



## Chapter 5

# Class Index

### 5.1 Class List

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

A	13
Abstract	13
GlobalFlow::Simulation::AbstractStepper	13
GlobalFlow::Solver::AdaptiveDamping	14
GlobalFlow::Callback	14
GlobalFlow::Container	14
GlobalFlow::DataProcessing::DataOutput::CSVOutput< T >	15
GlobalFlow::DataReader	15
GlobalFlow::Solver::Equation	17
GlobalFlow::Model::ExternalFlow	18
Factory	21
GlobalFlow::DataProcessing::DataOutput::FieldCollector	22
GlobalFlow::DataProcessing::DataOutput::FieldFactory	22
GlobalFlow::Model::FlowTypeHash	23
GlobalFlow::Model::FluidMechanics	23
GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T >	26
GlobalFlow::GW_Interface	26
std::hash< GlobalFlow::Model::NeighbourPosition >	27
GlobalFlow::Logging::InfoSinkInterface	28
Interface	28
GlobalFlow::DataProcessing::DataOutput::InternalTypeHash	29
Is< T >	29
GlobalFlow::Simulation::Iterator	29
GlobalFlow::Logging::Logger	30
GlobalFlow::Logging::LoggerInterface	30
GlobalFlow::Simulation::MassError	30
GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >	31
GlobalFlow::Model::NodeInterface	32
GlobalFlow::Model::NodeInterface::NodeNotFoundException	46
GlobalFlow::Simulation::Options	47
GlobalFlow::DataProcessing::DataOutput::OutputFactory< T >	48
GlobalFlow::DataProcessing::DataOutput::OutputField< T >	49
GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >	49
GlobalFlow::DataProcessing::DataOutput::OutputManager	50
GlobalFlow::DataProcessing::DataOutput::Outputvisitor	50
GlobalFlow::Model::PhysicalProperty< Type, Key >	50
Position	51
PropertyRepository	51

GlobalFlow::Model::PropertyRepository< PhysicalProperties > . . . . .	51
SaticHeadNode . . . . .	52
GlobalFlow::Simulation::Simulation . . . . .	52
GlobalFlow::Simulation::SimulationOutput . . . . .	56
GlobalFlow::Logging::Singleton< C > . . . . .	57
GlobalFlow::Model::StandardNode . . . . .	57
GlobalFlow::Model::StaticHeadNode . . . . .	58
GlobalFlow::Simulation::Stepper . . . . .	58
Write . . . . .	59
Writes . . . . .	59
Writes . . . . .	59
Writes . . . . .	59

## Chapter 6

# Namespace Documentation

### 6.1 GlobalFlow Namespace Reference

#### Classes

- class [Callback](#)
- class [Container](#)
- class [DataReader](#)
- class [GW\\_Interface](#)

#### Typedefs

- using **NodeVector** = std::shared\_ptr< std::vector< std::unique\_ptr< [Model::NodeInterface](#) >>>

#### Enumerations

- enum [Field](#) { **RIVER, LAKE, WETLANDS** }
- enum **severity\_level** {  
    **debug, userinfo, stateinfo, numerics,**  
    **error, critical** }

#### 6.1.1 Detailed Description

Converter Functions intended to be used internally only Currently only needed to allow compilation of different internal types There should be a more beautiful solution to this

**Bug** Some conversions make no sense

**Deprecated** use methods from DataProcessing

#### 6.1.2 Enumeration Type Documentation

##### 6.1.2.1 enum `GlobalFlow::Field` [strong]

[Container](#) for transferring data from GW-modell to WaterGAP Where as get(Field) always returns a map from type <ArcID, value>

Rivers:

Lakes:

Wetlands:

Definition at line 31 of file GW\_Interface.hpp.

## Chapter 7

# Class Documentation

### 7.1 A Class Reference

#### 7.1.1 Detailed Description

to write logs to files

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

- `/home/robert/development/G3M-framework/src/Logging/Logging.hpp`

### 7.2 Abstract Class Reference

#### 7.2.1 Detailed Description

a data output

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

- `/home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp`

### 7.3 GlobalFlow::Simulation::AbstractStepper Class Reference

Inheritance diagram for GlobalFlow::Simulation::AbstractStepper:

#### Public Member Functions

- virtual `Solver::Equation` \* **get** (int col) const =0

#### 7.3.1 Detailed Description

in order to iterate simply over simulation steps Holds a pointer to the equation and the choosen stepsize

Definition at line 34 of file Stepper.hpp.

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

- `/home/robert/development/G3M-framework/src/Simulation/Stepper.hpp`

## 7.4 GlobalFlow::Solver::AdaptiveDamping Class Reference

```
#include <Numerics.hpp>
```

### Public Member Functions

- **AdaptiveDamping** (double Sigma\_Min, double Sigma\_Max, double Change\_Max, vector x\_t0)
- vector **getDamping** (vector &residuals, vector &x, bool apply)
- double **getNorm** ()

#### 7.4.1 Detailed Description

Can be used to dampen the head change per iteration

Definition at line 26 of file Numerics.hpp.

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

- /home/robert/development/G3M-framework/src/Solver/Numerics.hpp

## 7.5 GlobalFlow::Callback Class Reference

### Public Member Functions

- void **call** ([Container](#) container)
- void **operator()** ([Container](#) container)

#### 7.5.1 Detailed Description

Definition at line 57 of file GW\_Interface.hpp.

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

- /home/robert/development/G3M-framework/src/GW\_Interface.hpp

## 7.6 GlobalFlow::Container Class Reference

### Public Member Functions

- **Container** (unordered\_map< int, double > rivers)
- const unordered\_map< int, double > **get** ([Field](#) field)

#### 7.6.1 Detailed Description

Definition at line 39 of file GW\_Interface.hpp.

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

- /home/robert/development/G3M-framework/src/GW\_Interface.hpp

## 7.7 GlobalFlow::DataProcessing::DataOutput::CSVOutput< T > Class Template Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::CSVOutput< T >:

Collaboration diagram for GlobalFlow::DataProcessing::DataOutput::CSVOutput< T >:

### Public Member Functions

- void **write** (path filePath, bool printID, bool printXY, std::vector< std::pair< double, double >> data, pos\_v p)
- void **write** (path filePath, bool printID, bool printXY, std::vector< bool > data, pos\_v p)
- void **write** (path filePath, bool printID, bool printXY, std::vector< double > data, pos\_v p)
- void **write** (path filePath, bool printID, bool printXY, std::vector< std::string > data, pos\_v p)

### 7.7.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::CSVOutput< T >

Definition at line 71 of file OutputFactory.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

## 7.8 GlobalFlow::DataReader Class Reference

### Public Member Functions

- void **initNodes** (NodeVector nodes)
- virtual void **readData** (Simulation::Options op)=0
- template<class Fun >  
void **loopFiles** (std::string path, std::vector< std::string > files, Fun fun)
- int **check** (int globid)
- template<class ProcessDataFunction >  
void **readTwoColumns** (std::string path, ProcessDataFunction processData)
- void **readZeroPointFiveToFiveMin** (std::string path)
- const std::unordered\_map< int, std::vector< int > > & **getArcIDMapping** ()
- const std::unordered\_map< int, int > & **getGlobIDMapping** ()
- std::string **buildDir** (std::string path)

### Public Attributes

- fs::path **data\_dir** {"data"}

### Protected Attributes

- NodeVector **nodes**
- int **stepMod**
- std::unordered\_map< int, int > **lookupglobIDtoID**
- std::unordered\_map< int, std::vector< int > > **lookupZeroPointFivetoFiveMinute**

### 7.8.1 Detailed Description

Definition at line 43 of file DataReader.hpp.

### 7.8.2 Member Function Documentation

**7.8.2.1** `template<class Fun > void GlobalFlow::DataReader::loopFiles ( std::string path, std::vector< std::string > files, Fun fun )` `[inline]`

Gereric method for looping through files inside a directory and apply a gernic function

Parameters

<i>path</i>	
<i>files</i>	
<i>fun</i>	a function

Definition at line 70 of file DataReader.hpp.

**7.8.2.2** `virtual void GlobalFlow::DataReader::readData ( Simulation::Options op )` `[pure virtual]`

Attention

Needs to be implemented!

Note

[Is](#) called by simulation at startup

Parameters

<i>op</i>	Options object
-----------	----------------

**7.8.2.3** `template<class ProcessDataFunction > void GlobalFlow::DataReader::readTwoColumns ( std::string path, ProcessDataFunction processData )` `[inline]`

Read data from a two-column csv file and apply function to data

Parameters

<i>path</i>	
<i>processData</i>	<a href="#">A</a> processing function e.g. upscaling of data

Definition at line 94 of file DataReader.hpp.

**7.8.2.4** `void GlobalFlow::DataReader::readZeroPointFiveToFiveMin ( std::string path )` `[inline]`

Creates a mapping of 0.5° ArcIDs to a list of contained 5' GlobIDs

Parameters

<i>path</i>	
-------------	--

Definition at line 116 of file DataReader.hpp.

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

- `/home/robert/development/G3M-framework/src/DataProcessing/DataReader.hpp`



## 7.9 GlobalFlow::Solver::Equation Class Reference

### Public Types

- typedef Eigen::MatrixX<Scalar> **Scalar**
- typedef Matrix< Scalar, Dynamic, 1 > **VectorType**

### Public Member Functions

- **Equation** (unsigned long numberOfNodes, NodeVector nodes, [Simulation::Options](#) options)
- void [solve](#) ()
- int [getItter](#) ()
- double [getError](#) ()
- **Equation** (const [Equation](#) &)=delete
- [Equation](#) & **operator=** (const [Equation](#) &)=delete
- VectorXd [getResults](#) ()
- bool [toggleSteadyState](#) ()
- void [updateStepSize](#) (size\_t mod)
- VectorType [getResiduals](#) ()
- void [updateClosingCrit](#) (double crit)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, [Equation](#) &eq)

#### 7.9.1 Detailed Description

finite difference equation Should only be accessed through the stepper  
Definition at line 45 of file Equation.hpp.

#### 7.9.2 Member Function Documentation

##### 7.9.2.1 double GlobalFlow::Solver::Equation::getError ( )

###### Returns

The current residual error

Definition at line 328 of file Equation.cpp.

##### 7.9.2.2 int GlobalFlow::Solver::Equation::getItter ( )

###### Returns

The number of iterations

Definition at line 323 of file Equation.cpp.

### 7.9.2.3 void GlobalFlow::Solver::Equation::solve ( )

Solve the current iteration step

Solve [Equation](#) head change convergance

residual norm convergance

Definition at line 188 of file Equation.cpp.

### 7.9.2.4 bool GlobalFlow::Solver::Equation::toggleSteadyState ( ) [inline]

Toogle the steady-state in all nodes

#### Returns

Definition at line 93 of file Equation.hpp.

### 7.9.2.5 void GlobalFlow::Solver::Equation::updateStepSize ( size\_t mod ) [inline]

Set the correct stepsize (default is DAY)

#### Parameters

<i>mod</i>	
------------	--

Definition at line 108 of file Equation.hpp.

## 7.9.3 Friends And Related Function Documentation

### 7.9.3.1 std::ostream& operator<< ( std::ostream & os, Equation & eq ) [friend]

Helper to write out current residuals

#### Parameters

<i>os</i>	
<i>eq</i>	

#### Returns

Definition at line 79 of file Equation.hpp.

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

- /home/robert/development/G3M-framework/src/Solver/Equation.hpp
- /home/robert/development/G3M-framework/src/Solver/Equation.cpp

## 7.10 GlobalFlow::Model::ExternalFlow Class Reference

```
#include <ExternalFlows.hpp>
```

## Public Member Functions

- **ExternalFlow** (int id, FlowType type, t\_meter flowHead, t\_s\_meter\_t cond, t\_meter bottom)
- **ExternalFlow** (int id, t\_vol\_t recharge, FlowType type)
- **ExternalFlow** (int id, t\_meter flowHead, t\_meter bottom, t\_vol\_t evapotrans)  
*Constructor for Evapotranspiration.*
- bool **flowsHeadDependant** (t\_meter head) const noexcept
- t\_s\_meter\_t **getP** (t\_meter head, t\_meter eq\_head, t\_vol\_t recharge, t\_dim slope, t\_vol\_t eqFlow) const noexcept
- t\_vol\_t **getQ** (t\_meter head, t\_meter eq\_head, t\_vol\_t recharge, t\_dim slope, t\_vol\_t eqFlow) const noexcept
- FlowType **getType** () const noexcept
- t\_meter **getBottom** () const noexcept
- t\_vol\_t **getRecharge** () const noexcept
- t\_meter **getFlowHead** () const noexcept
- t\_s\_meter\_t **getDyn** (t\_vol\_t current\_recharge, t\_meter eq\_head, t\_meter head, t\_vol\_t eq\_flow) const noexcept
- t\_meter **getRiverDiff** (t\_meter eqHead) const noexcept
- t\_s\_meter\_t **getConductance** () const noexcept
- int **getID** () const noexcept
- void **setMult** (double mult)

### 7.10.1 Detailed Description

TODO add flow equation here

Definition at line 81 of file ExternalFlows.hpp.

### 7.10.2 Constructor & Destructor Documentation

7.10.2.1 GlobalFlow::Model::ExternalFlow::ExternalFlow ( int *id*, t\_vol\_t *recharge*, FlowType *type* ) [inline]

Only for RECHARGE FAST\_SURFACE\_RUNOFF

Definition at line 94 of file ExternalFlows.hpp.

7.10.2.2 GlobalFlow::Model::ExternalFlow::ExternalFlow ( int *id*, t\_meter *flowHead*, t\_meter *bottom*, t\_vol\_t *evapotrans* ) [inline]

Constructor for Evapotranspiration.

#### Parameters

<i>id</i>	
<i>flowHead</i>	
<i>bottom</i>	
<i>evapotrans</i>	

#### Returns

Definition at line 105 of file ExternalFlows.hpp.

### 7.10.3 Member Function Documentation

7.10.3.1 `bool GlobalFlow::Model::ExternalFlow::flowIsHeadDependant ( t_meter head ) const` `[inline], [noexcept]`

Check if flow can be calculated on the right hand side

## Parameters

<i>head</i>	The current hydraulic head
-------------	----------------------------

## Returns

Bool

Definition at line 113 of file ExternalFlows.hpp.

7.10.3.2 `t_s_meter_t GlobalFlow::Model::ExternalFlow::getP ( t_meter head, t_meter eq_head, t_vol_t recharge, t_dim slope, t_vol_t eqFlow ) const [noexcept]`

The head dependant part of the external flow equation

## Parameters

<i>head</i>	The current hydraulic head
<i>eq_head</i>	The equilibrium head
<i>recharge</i>	The current recharge
<i>slope</i>	
<i>eqFlow</i>	

## Returns

Definition at line 6 of file ExternalFlows.cpp.

7.10.3.3 `t_vol_t GlobalFlow::Model::ExternalFlow::getQ ( t_meter head, t_meter eq_head, t_vol_t recharge, t_dim slope, t_vol_t eqFlow ) const [noexcept]`

The head independant part of the external flow equation

## Parameters

<i>head</i>	
<i>eq_head</i>	
<i>recharge</i>	
<i>slope</i>	
<i>eqFlow</i>	

## Returns

Definition at line 52 of file ExternalFlows.cpp.

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

- /home/robert/development/G3M-framework/src/Model/ExternalFlows.hpp
- /home/robert/development/G3M-framework/src/Model/ExternalFlows.cpp

## 7.11 Factory Class Reference

### 7.11.1 Detailed Description

correct data output file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

## 7.12 GlobalFlow::DataProcessing::DataOutput::FieldCollector Class Reference

```
#include <FieldCollector.hpp>
```

### Public Types

- using **pos\_v** = std::vector< std::pair< double, double >>

### Public Member Functions

- **FieldCollector** (FieldType enumField)
- pos\_v **getPositions** ([Simulation::Simulation](#) const &simulation)
- template<typename T >  
data\_vector< T > **get** ([Simulation::Simulation](#) const &simulation)

#### 7.12.1 Detailed Description

Iterates over internal fields and searches for data to be written out This is currently relatively inefficient  
Definition at line 94 of file FieldCollector.hpp.

#### 7.12.2 Member Function Documentation

7.12.2.1 template<typename T > data\_vector<T> GlobalFlow::DataProcessing::DataOutput::FieldCollector::get (  
**Simulation::Simulation** const & *simulation* ) [inline]

Collects the data from simulation nodes

#### Note

Relatively inefficient currently

#### Parameters

<i>simulation</i>	The simulation
-------------------	----------------

#### Returns

The collected data

Definition at line 130 of file FieldCollector.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/FieldCollector.hpp

## 7.13 GlobalFlow::DataProcessing::DataOutput::FieldFactory Class Reference

### Static Public Member Functions

- static field\_vector **build** (boost::property\_tree::ptree tree, [Simulation::Simulation](#) const &sim)

### 7.13.1 Detailed Description

Definition at line 120 of file OutputManager.hpp.

### 7.13.2 Member Function Documentation

7.13.2.1 static field\_vector GlobalFlow::DataProcessing::DataOutput::FieldFactory::build ( boost::property\_tree::ptree *tree*, Simulation::Simulation const & *sim* ) [inline], [static]

Parameters

<i>tree</i>	
-------------	--

Definition at line 131 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

## 7.14 GlobalFlow::Model::FlowTypeHash Struct Reference

### Public Member Functions

- template<typename T>  
std::size\_t **operator()** (T t) const

### 7.14.1 Detailed Description

Definition at line 69 of file ExternalFlows.hpp.

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

- /home/robert/development/G3M-framework/src/Model/ExternalFlows.hpp

## 7.15 GlobalFlow::Model::FluidMechanics Class Reference

```
#include <FluidMechanics.hpp>
```

### Public Member Functions

- t\_meter **calcDeltaV** (t\_meter head, t\_meter elevation, t\_meter depth) noexcept
- t\_s\_meter\_t **calculateHarmonicMeanConductance** (t\_vel k\_neig, t\_vel k\_self, t\_meter edgeLength\_neig, t\_meter edgeLength\_self, t\_meter head\_neig, t\_meter head\_self, t\_meter ele\_neig, t\_meter ele\_self, t\_meter deltaV\_neig, t\_meter deltaV\_self, bool confined) noexcept  
*Calculates the horizontal flow between two nodes.*
- double **smoothFunction\_\_NWT** (t\_meter elevation, t\_meter verticalSize, t\_meter head)
- t\_s\_meter\_t **getHCOF** (bool steadyState, quantity< Dimensionless > stepModifier, t\_s\_meter storageCapacity, t\_s\_meter\_t P) noexcept
- t\_s\_meter\_t **calculateVerticalConductance** (t\_vel k\_vert\_neig, t\_vel k\_vert\_self, t\_meter verticalSize\_self, t\_meter head\_self, t\_meter elevation\_self, t\_s\_meter area\_self, t\_meter elevation\_neig, t\_meter depth\_neig, t\_meter head\_neig, bool confined) noexcept
- double **getDerivate\_\_NWT** (t\_meter elevation, t\_meter verticalSize, t\_meter head)

### 7.15.1 Detailed Description

Provides helper functions for conductance calulcations

Definition at line 36 of file FluidMechanics.hpp.

### 7.15.2 Member Function Documentation

7.15.2.1 `t_meter GlobalFlow::Model::FluidMechanics::calcDeltaV ( t_meter head, t_meter elevation, t_meter depth )`  
[noexcept]

Used to calculate if a cell is dry

Definition at line 7 of file FluidMechanics.cpp.

7.15.2.2 `quantity< MeterSquaredPerTime > GlobalFlow::Model::FluidMechanics::calculateHarmonicMeanConductance ( t_vel k_neig, t_vel k_self, t_meter edgeLength_neig, t_meter edgeLength_self, t_meter head_neig, t_meter head_self, t_meter ele_neig, t_meter ele_self, t_meter deltaV_neig, t_meter deltaV_self, bool confined )` [noexcept]

Calculates the horizontal flow between two nodes.

Parameters

<i>k_neig</i>	K-value of neighbouring node
<i>edgeLength_↔ neig</i>	The edge size of the neighbouring node (assumes rectangular cells)

Returns

A weighted conductance value for the flow between two nodes

Calculates the harmonic mean conductance between two nodes.  $C = 2 * \text{EdgeLenght}_1 * (\text{TR}_1 * \text{TR}_2) / (\text{TR}_1 * \text{EdgeLenght}_1 + \text{TR}_2 * \text{EdgeLenght}_2)$

Definition at line 18 of file FluidMechanics.cpp.

7.15.2.3 `quantity< MeterSquaredPerTime > GlobalFlow::Model::FluidMechanics::calculateVerticalConductance ( t_vel k_vert_neig, t_vel k_vert_self, t_meter verticalSize_self, t_meter head_self, t_meter elevation_self, t_s_meter area_self, t_meter elevation_neig, t_meter depth_neig, t_meter head_neig, bool confined )` [noexcept]

Calculates the vertical flow between two nodes

Parameters

<i>k_vert_neig</i>	
<i>k_vert_self</i>	
<i>verticalSize_self</i>	
<i>head_self</i>	
<i>elevation_self</i>	
<i>area_self</i>	
<i>elevation_neig</i>	
<i>depth_neig</i>	
<i>head_neig</i>	



<i>confined</i>	
-----------------	--

#### Returns

Definition at line 76 of file FluidMechanics.cpp.

7.15.2.4 `double GlobalFlow::Model::FluidMechanics::getDerivate__NWT ( t_meter elevation, t_meter verticalSize, t_meter head )`

Calculate derivatives for NWT approach

#### Parameters

<i>elevation</i>	
<i>verticalSize</i>	
<i>head</i>	

#### Returns

Definition at line 115 of file FluidMechanics.cpp.

7.15.2.5 `quantity< MeterSquaredPerTime > GlobalFlow::Model::FluidMechanics::getHCOF ( bool steadyState, quantity< Dimensionless > stepModifier, t_s_meter storageCapacity, t_s_meter_t P ) [noexcept]`

Get the coefficients for storage and P components

#### Parameters

<i>steadyState</i>	
<i>stepModifier</i>	
<i>storageCapacity</i>	
<i>P</i>	

#### Returns

Definition at line 103 of file FluidMechanics.cpp.

7.15.2.6 `double GlobalFlow::Model::FluidMechanics::smoothFunction__NWT ( t_meter elevation, t_meter verticalSize, t_meter head )`

Simple smoother function to buffer iteration steps in NWT approach

#### Parameters

<i>elevation</i>	
<i>verticalSize</i>	
<i>head</i>	

## Returns

Definition at line 55 of file FluidMechanics.cpp.

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

- /home/robert/development/G3M-framework/src/Model/FluidMechanics.hpp
- /home/robert/development/G3M-framework/src/Model/FluidMechanics.cpp

## 7.16 GlobalFlow::DataProcessing::DataOutput::GFS\_JSONOutput< T > Class Template Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::GFS\_JSONOutput< T >:

Collaboration diagram for GlobalFlow::DataProcessing::DataOutput::GFS\_JSONOutput< T >:

### Public Member Functions

- void [write](#) (path filePath, bool printID, bool printXY, std::vector< T > data, pos\_v p)

#### 7.16.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::GFS\_JSONOutput< T >

Definition at line 155 of file OutputFactory.hpp.

#### 7.16.2 Member Function Documentation

7.16.2.1 template<typename T> void GlobalFlow::DataProcessing::DataOutput::GFS\_JSONOutput< T >::write ( path filePath, bool printID, bool printXY, std::vector< T > data, pos\_v p ) [inline],[virtual]

Needs to be implemented

#### Parameters

<i>filePath</i>	
<i>printID</i>	Bool
<i>printXY</i>	Bool
<i>data</i>	Data vector
<i>p</i>	<a href="#">Position</a> vector

Implements [GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >](#).

Definition at line 299 of file OutputFactory.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

## 7.17 GlobalFlow::GW\_Interface Class Reference

### Public Member Functions

- virtual void [loadSettings](#) ()=0

- virtual void [setupSimulation](#) ()=0
- virtual void [writeData](#) ()=0
- void [setupCallBack](#) ([Callback](#) callback)
- virtual void [simulate](#) ()=0

### Protected Attributes

- std::function< void([Container](#) container)> **callback**

### 7.17.1 Detailed Description

Definition at line 72 of file GW\_Interface.hpp.

### 7.17.2 Member Function Documentation

7.17.2.1 virtual void GlobalFlow::GW\_Interface::loadSettings ( ) [pure virtual]

Read general simulation settings e.g. Options

7.17.2.2 void GlobalFlow::GW\_Interface::setupCallBack ( [Callback](#) *callback* ) [inline]

Set up the callback for model coupling Could be inefficient

Parameters

<i>callback</i>	
-----------------	--

Definition at line 98 of file GW\_Interface.hpp.

7.17.2.3 virtual void GlobalFlow::GW\_Interface::setupSimulation ( ) [pure virtual]

Do additional work required for a running simulation

7.17.2.4 virtual void GlobalFlow::GW\_Interface::simulate ( ) [pure virtual]

Simulate/Run the model

7.17.2.5 virtual void GlobalFlow::GW\_Interface::writeData ( ) [pure virtual]

How should the data be written out

### Deprecated

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

- /home/robert/development/G3M-framework/src/GW\_Interface.hpp

## 7.18 std::hash< GlobalFlow::Model::NeighbourPosition > Struct Template Reference

### Public Types

- typedef GlobalFlow::Model::NeighbourPosition **argument\_type**

- typedef std::underlying\_type< argument\_type >::type **underlying\_type**
- typedef std::hash< underlying\_type >::result\_type **result\_type**

## Public Member Functions

- result\_type **operator()** (const argument\_type &arg) const

### 7.18.1 Detailed Description

template<> struct std::hash< GlobalFlow::Model::NeighbourPosition >

Definition at line 63 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

## 7.19 GlobalFlow::Logging::InfoSinkInterface Class Reference

```
#include <Sinks.hpp>
```

Inheritance diagram for GlobalFlow::Logging::InfoSinkInterface:

Collaboration diagram for GlobalFlow::Logging::InfoSinkInterface:

## Public Member Functions

- **InfoSinkInterface** (std::string file\_name)
- virtual void **consume** (logboost::record\_view const &rec)=0
- void **flush** ()
- void **write\_data** ()

### 7.19.1 Detailed Description

**Deprecated** Use methods from DataProcessing

Definition at line 39 of file Sinks.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Sinks.hpp

## 7.20 Interface Interface Reference

### 7.20.1 Detailed Description

to be implemented for reading in required data for the model

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataReader.hpp

## 7.21 GlobalFlow::DataProcessing::DataOutput::InternalTypeHash Struct Reference

### Public Member Functions

- `template<typename T>`  
`std::size_t operator() (T t) const`

#### 7.21.1 Detailed Description

Definition at line 87 of file OutputManager.hpp.

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

- `/home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp`

## 7.22 Is< T > Struct Template Reference

```
#include <Helpers.hpp>
```

### Public Member Functions

- `bool in (T a)`
- `template<class Arg , class... Args>`  
`bool in (Arg a, Args...args)`

### Public Attributes

- `T d_`

#### 7.22.1 Detailed Description

```
template<class T>struct Is< T >
```

<http://stackoverflow.com/questions/15181579>

Definition at line 60 of file Helpers.hpp.

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

- `/home/robert/development/G3M-framework/src/Misc/Helpers.hpp`

## 7.23 GlobalFlow::Simulation::Iterator Class Reference

### Public Member Functions

- `Iterator (const AbstractStepper *stepper, TimeFrame time, int pos)`
- `bool operator!= (const Iterator &other) const`
- `step operator* () const`
- `const Iterator & operator++ ()`
- `int getPos ()`

### 7.23.1 Detailed Description

iterator holding the current simulation step

Definition at line 43 of file Stepper.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Stepper.hpp

## 7.24 GlobalFlow::Logging::Logger Class Reference

Inheritance diagram for GlobalFlow::Logging::Logger:

Collaboration diagram for GlobalFlow::Logging::Logger:

### Additional Inherited Members

### 7.24.1 Detailed Description

Definition at line 80 of file Logging.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Logging.hpp

## 7.25 GlobalFlow::Logging::LoggerInterface Class Reference

Inheritance diagram for GlobalFlow::Logging::LoggerInterface:

Collaboration diagram for GlobalFlow::Logging::LoggerInterface:

### Friends

- class **Singleton**< **LoggerInterface** >

### Additional Inherited Members

### 7.25.1 Detailed Description

Definition at line 68 of file Logging.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Logging.hpp

## 7.26 GlobalFlow::Simulation::MassError Class Reference

```
#include <Simulation.hpp>
```

### Public Member Functions

- **MassError** (mpf\_float\_1000 OUT, mpf\_float\_1000 IN, mpf\_float ERR)

## Public Attributes

- `mpf_float_1000 OUT`
- `mpf_float_1000 IN`
- `mpf_float ERR = 0`

### 7.26.1 Detailed Description

Simple container for the mass error calculations

Definition at line 55 of file `Simulation.hpp`.

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

- `/home/robert/development/G3M-framework/src/Simulation/Simulation.hpp`

## 7.27 GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T > Class Template Reference

Inheritance diagram for `GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >`:

Collaboration diagram for `GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >`:

## Public Member Functions

- void `write` (path `filePath`, bool `printID`, bool `printXY`, std::vector< T > `data`, pos\_v `p`)

### 7.27.1 Detailed Description

`template<typename T>class GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >`

Definition at line 112 of file `OutputFactory.hpp`.

### 7.27.2 Member Function Documentation

7.27.2.1 `template<typename T> void GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >::write ( path filePath, bool printID, bool printXY, std::vector< T > data, pos_v p )` `[inline]`, `[virtual]`

Needs to be implemented

#### Parameters

<i>filePath</i>	
<i>printID</i>	Bool
<i>printXY</i>	Bool
<i>data</i>	Data vector
<i>p</i>	Position vector

Implements `GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >`.

Definition at line 114 of file `OutputFactory.hpp`.

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

- `/home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp`

## 7.28 GlobalFlow::Model::NodeInterface Class Reference

#include <Node.hpp>

Inheritance diagram for GlobalFlow::Model::NodeInterface:

Collaboration diagram for GlobalFlow::Model::NodeInterface:

### Classes

- class [NodeNotFoundException](#)

### Public Member Functions

- [NodeInterface](#) (NodeVector nodes, double lat, double lon, t\_s\_meter area, unsigned long ArcID, unsigned long ID, t\_vel K, int stepModifier, double aquiferDepth, double anisotropy, double specificYield, double specificStorage, bool confined)

*Constructor of abstract class [NodeInterface](#).*

- unsigned long [getID](#) ()
- void [setElevation](#) (t\_meter elevation)  
*Set elevation on top layer and propagate to lower layers.*
- void [setSlope](#) (double slope\_percent)  
*Set slope from data on all layers Slope input is in % but is required as absolut thus: slope = sloper\_percent / 100.*
- void [setEfold](#) (double efold)  
*Set e-folding factor from data on all layers.*
- void [setEqHead](#) (t\_meter wtd)  
*Calculated equilibrium groundwater-head from eq\_wtd Assumes that if initialhead = false that the eq\_head is also used as initial head.*
- template<class HeadType >  
t\_vol\_t [calcLateralFlows](#) ()
- t\_vol\_t [getEqFlow](#) () noexcept
- t\_vol\_t [getLateralFlows](#) ()
- bool [resetFloodingHead](#) () noexcept  
*Cuts off all heads above surface elevation.*
- void [scaleRiverConduct](#) ()  
*Scales river conduct by 50%.*
- void [updateHeadChange](#) () noexcept  
*Update the current head change (in comparison to last time step)*
- void [initHead\\_t0](#) () noexcept
- void [setHead\\_direct](#) (double head) noexcept
- t\_vel [getK\\_pure](#) () noexcept
- t\_vel [getK](#) () noexcept  
*Get hydraulic conductivity.*
- t\_vel [getK\\_vertical](#) () noexcept  
*Get hydraulic vertical conductivity.*
- void [setK](#) (t\_vel conduct)  
*Modify hydraulic conductivity (applied to all layers below)*
- void [setK\\_direct](#) (t\_vel conduct)  
*Modify hydraulic conductivity (no e-folding, no layers)*
- t\_c\_meter [getOUT](#) () noexcept  
*Get all outflow since simulation start.*
- t\_c\_meter [getIN](#) () noexcept  
*Get all inflow since simulation start.*



- void **toggleStadyState** (bool onOFF)  
*Toogle steady state simulation.*
- void **updateStepSize** (double mod)
- t\_s\_meter **getStorageCapacity** () noexcept  
*Storage capacity based on yield or specific storage.*
- **ExternalFlow** & **getExternalFlowByName** (FlowType type) throw (out\_of\_range)  
*Get and external flow by its FlowType.*
- t\_vol\_t **getExternalFlowVolumeByName** (FlowType type)  
*Get and external flow volume by its FlowType.*
- t\_vol\_t **getTotalStorageFlow** () noexcept  
*Get flow budget based on head change.*
- t\_vol\_t **calculateExternalFlowVolume** (const **ExternalFlow** &flow)  
*Get flow budget of a specific external flows.*
- t\_vol\_t **calculateDewateredFlow** () noexcept  
*Caluclate dewatered flow.*
- t\_vol\_t **getCurrentIN** () noexcept  
*Get all current IN flow.*
- t\_vol\_t **getCurrentOUT** () noexcept  
*Get all current OUT flow.*
- void **saveMassBalance** () noexcept  
*Tell cell to save its flow budget.*
- void **setNeighbour** (unsigned long ID, NeighbourPosition neighbour)  
*Add a neighbour.*
- int **getNumofNeighbours** ()
- **NodeInterface** \* **getNeighbour** (NeighbourPosition neighbour) noexcept(false)  
*Get a neighbour by position.*
- int **addExternalFlow** (FlowType type, t\_meter flowHead, double cond, t\_meter bottom)  
*At an external flow to the cell.*
- void **removeExternalFlow** (FlowType type)  
*Remove an external flow to the cell by id.*
- bool **hasTypeOfExternalFlow** (FlowType type)  
*Check for an external flow by type.*
- void **updateUniqueFlow** (double amount, FlowType flow=RECHARGE)  
*Updates GW recharge Curently assumes only one recharge as external flow!*
- void **scaleDynamicRivers** (double mult)
- void **updateExternalFlowConduct** (double amount, FlowType type)  
*Update wetlands, lakes.*
- void **updateLakeBottoms** (double amount)  
*Update lake bottoms Used for sensitivity.*
- bool **hasRiver** ()  
*Check for type river.*
- bool **hasOcean** ()  
*Check for type ocean.*
- t\_vol\_t **getQ** () noexcept  
*Get Q part of flow equations.*
- t\_s\_meter\_t **getP** () noexcept  
*Get P part of flow equations.*
- t\_vol\_t **calculateNotHeadDependandFlows** () noexcept  
*Get flow which is not groundwater head dependent.*
- std::unordered\_map< int, t\_s\_meter\_t > **getJacobian** () noexcept  
*The jacobian entry for the cell (NWT approach)*

- `std::unordered_map< int, t_s_meter_t > getConductance ()`  
*The matrix entry for the cell.*
- `t_vol_t getRHS () noexcept`  
*The right hand side of the equation.*
- `double getRHS__NWT () noexcept`  
*The right hand side of the equation (NWT)*
- `void setHead (t_meter head) noexcept`
- `t_meter calcInitialHead (t_meter initialParam) noexcept`
- `bool isStaticNode () noexcept`
- `PhysicalProperties & getProperties ()`
- `void enableNWT ()`
- `template<typename CompareFunction >`  
`t_vol_t getNonStorageFlow (CompareFunction compare) noexcept`  
*Caluclate non storage related in and out flow.*
- `quantity< Velocity > getVelocity (map_itter pos)`
- `std::pair< double, double > getVelocityVector ()`  
*Calculate flow velocity for flow tracking Vx and Vy represent the flow velocity in x and y direction. A negative value represents a flow in the oposite direction.*

## Public Attributes

- `bool cached {false}`
- `t_vol_t eq_flow {0 * si::cubic_meter / day}`

## Protected Types

- using `map_itter` = `std::unordered_map< NeighbourPosition, unsigned long >::const_iterator`

## Protected Member Functions

- `template<typename T, typename F >`  
`T get ()`
- `template<typename T, typename F >`  
`void set (const T &value)`
- `template<typename T, typename F >`  
`T getFrom (NodeInterface *nodeInterface, NeighbourPosition pos)`
- `p_node & at (map_itter pos)`
- `template<typename T, typename F >`  
`T getAt (map_itter pos)`
- `t_s_meter getStorageCapacity__Primary () noexcept`  
*Uses specific storage to caluclate storativity.*
- `t_s_meter getStorageCapacity__Secondary () noexcept`  
*Uses specific yield to caluclate storativity.*
- `t_s_meter getStorageCapacity__SecondaryNWT () noexcept`  
*Uses specific yield with NWT smoother.*
- `template<typename CompareFunction >`  
`t_vol_t getFlow (CompareFunction compare) noexcept`  
*Flow volume of cell.*

## Protected Attributes

- const std::shared\_ptr< std::vector< std::unique\_ptr< [NodeInterface](#) > > > **nodes**
- unordered\_map< NeighbourPosition, unsigned long > **neighbours**
- unordered\_map< FlowType, [ExternalFlow](#), [FlowTypeHash](#) > **externalFlows**
- int **numOfExternalFlows** {0}
- bool **nwt** {false}
- bool **initial\_head** {true}
- bool **simpleDistance** {false}
- bool **simpleK** {false}
- bool **steadyState** {false}
- [FluidMechanics](#) **mechanics**
- [PhysicalProperties](#) **fields**

## Friends

- class **FluidMechanics**

### 7.28.1 Detailed Description

[Interface](#) defining required fields for a node. [A](#) node is the central computational and spatial unit. [A](#) simulated area is separated into a discrete raster of cells or nodes (separate computational units which stay in contact to each other). [Is](#) equal to 'cell'.

Nodes can be of different physical property e.g. different size.

Definition at line 96 of file Node.hpp.

### 7.28.2 Constructor & Destructor Documentation

7.28.2.1 `GlobalFlow::Model::NodeInterface::NodeInterface ( NodeVector nodes, double lat, double lon, t_s_meter area, unsigned long ArcID, unsigned long ID, t_vel K, int stepModifier, double aquiferDepth, double anisotropy, double specificYield, double specificStorage, bool confined )`

Constructor of abstract class [NodeInterface](#).

#### Parameters

<i>nodes</i>	Vector of all other existing nodes
<i>lat</i>	The latitude
<i>lon</i>	The Longitude
<i>area</i>	Area in m <sup>2</sup>
<i>ArcID</i>	Unique ARC-ID specified by Kassel
<i>ID</i>	Internal ID = <a href="#">Position</a> in vector
<i>K</i>	Hydraulic conductivity in meter/day (default)
<i>stepModifier</i>	Modifies default step size of day (default=1)
<i>aquiferDepth</i>	Vertical size of the cell
<i>anisotropy</i>	Modifier for vertical conductivity based on horizontal
<i>specificYield</i>	Yield of storage for dewatered conditions
<i>specificStorage</i>	Specific storage - currently for confined and unconfined

<i>confined</i>	Is node in a confined layer
-----------------	-----------------------------

Definition at line 42 of file Node.cpp.

### 7.28.3 Member Function Documentation

**7.28.3.1** `int GlobalFlow::Model::NodeInterface::addExternalFlow ( FlowType type, t_meter flowHead, double cond, t_meter bottom ) [inline]`

At an external flow to the cell.

#### Parameters

<i>type</i>	The flow type
<i>flowHead</i>	The flow head
<i>cond</i>	The conductance
<i>bottom</i>	The bottom of the flow (e.g river bottom)

#### Returns

Number assigned by cell to flow

Definition at line 711 of file Node.hpp.

**7.28.3.2** `template<class HeadType > t_vol_t GlobalFlow::Model::NodeInterface::calcLateralFlows ( ) [inline]`

Calculate the lateral groundwater flow to the neighbouring nodes Generic function used for calculating equilibrium and current step flow

#### Returns

Definition at line 357 of file Node.hpp.

Here is the call graph for this function:

**7.28.3.3** `t_vol_t GlobalFlow::Model::NodeInterface::calculateDewateredFlow ( ) [inline], [noexcept]`

Caluclate dewatered flow.

#### Returns

Flow volume per time If a cell is dewatered but below a saturated or partly saturated cell: this calculates the needed additional exchange volume

Definition at line 605 of file Node.hpp.

**7.28.3.4** `t_vol_t GlobalFlow::Model::NodeInterface::calculateExternalFlowVolume ( const ExternalFlow & flow ) [inline]`

Get flow budget of a specific external flows.

## Parameters

<i>&amp;flow</i>	<a href="#">A</a> external flow
------------------	---------------------------------

## Returns

Flow volume Note: Water entering storage is treated as an outflow (-), that is a loss of water from the flow system while water released from storage is treated as inflow (+), that is a source of water to the flow system

Definition at line 568 of file Node.hpp.

Here is the call graph for this function:

**7.28.3.5** `t_vol_t GlobalFlow::Model::NodeInterface::calculateNotHeadDependandFlows ( )` `[inline]`, `[noexcept]`

Get flow which is not groundwater head dependent.

## Returns

volume over time Flow can be added to constant flows on right side of the equations If head is above river bottom for example

Definition at line 884 of file Node.hpp.

**7.28.3.6** `std::unordered_map<int, t_s_meter_t> GlobalFlow::Model::NodeInterface::getConductance ( )` `[inline]`

The matrix entry for the cell.

## Returns

map <CellID,Conductance> The left hand side of the equation

Definition at line 958 of file Node.hpp.

**7.28.3.7** `t_vol_t GlobalFlow::Model::NodeInterface::getCurrentIN ( )` `[inline]`, `[noexcept]`

Get all current IN flow.

## Returns

Flow volume

Definition at line 659 of file Node.hpp.

**7.28.3.8** `t_vol_t GlobalFlow::Model::NodeInterface::getCurrentOUT ( )` `[inline]`, `[noexcept]`

Get all current OUT flow.

## Returns

Flow volume

Definition at line 665 of file Node.hpp.

7.28.3.9 `t_vol_t GlobalFlow::Model::NodeInterface::getEqFlow ( ) [inline], [noexcept]`

Calculate the equilibrium lateral flows

Returns

eq lateral flow

Definition at line 391 of file Node.hpp.

7.28.3.10 `ExternalFlow& GlobalFlow::Model::NodeInterface::getExternalFlowByName ( FlowType type ) throw out_of_range`  
`[inline]`

Get and external flow by its FlowType.

Parameters

<i>type</i>	The flow type
-------------	---------------

Returns

Ref to external flow

Exceptions

<i>OutOfRangeException</i>	
----------------------------	--

Definition at line 530 of file Node.hpp.

7.28.3.11 `t_vol_t GlobalFlow::Model::NodeInterface::getExternalFlowVolumeByName ( FlowType type ) [inline]`

Get and external flow volume by its FlowType.

Parameters

<i>type</i>	The flow type
-------------	---------------

Returns

Flow volume

Definition at line 541 of file Node.hpp.

7.28.3.12 `template<typename CompareFunction > t_vol_t GlobalFlow::Model::NodeInterface::getFlow ( CompareFunction compare ) [inline], [protected], [noexcept]`

Flow volume of cell.

Returns

Flow volume

Definition at line 222 of file Node.hpp.

7.28.3.13 `std::unordered_map<int, t_s_meter_t> GlobalFlow::Model::NodeInterface::getJacobian ( ) [inline], [noexcept]`

The jacobian entry for the cell (NWT approach)

**Returns**

map <CellID,Conductance>

Definition at line 911 of file Node.hpp.

#### 7.28.3.14 `t_vel GlobalFlow::Model::NodeInterface::getK ( ) [inline], [noexcept]`

Get hydraulic conductivity.

**Returns**

hydraulic conductivity (scaled by e-folding)

Definition at line 451 of file Node.hpp.

#### 7.28.3.15 `t_vel GlobalFlow::Model::NodeInterface::getK_vertical ( ) [inline], [noexcept]`

Get hydraulic vertical conductivity.

**Returns**

hydraulic conductivity scaled by anisotropy (scaled by e-folding)

Definition at line 468 of file Node.hpp.

#### 7.28.3.16 `t_vol_t GlobalFlow::Model::NodeInterface::getLateralFlows ( ) [inline]`

Get the current lateral flow

**Returns**

Definition at line 405 of file Node.hpp.

#### 7.28.3.17 `NodeInterface* GlobalFlow::Model::NodeInterface::getNeighbour ( NeighbourPosition neighbour ) [inline], [noexcept]`

Get a neighbour by position.

**Parameters**

<i>neighbour</i>	The position relative to the cell
------------------	-----------------------------------

**Returns**

Pointer to cell object

Definition at line 693 of file Node.hpp.

#### 7.28.3.18 `template<typename CompareFunction> t_vol_t GlobalFlow::Model::NodeInterface::getNonStorageFlow ( CompareFunction compare ) [inline], [noexcept]`

Caluclate non storage related in and out flow.

**Returns**

Flow volume

Definition at line 1101 of file Node.hpp.

7.28.3.19 `t_s_meter_t GlobalFlow::Model::NodeInterface::getP ( ) [inline], [noexcept]`

Get P part of flow equations.

#### Returns

volume over time

Definition at line 855 of file Node.hpp.

7.28.3.20 `t_vol_t GlobalFlow::Model::NodeInterface::getQ ( ) [inline], [noexcept]`

Get Q part of flow equations.

#### Returns

volume over time

Definition at line 834 of file Node.hpp.

7.28.3.21 `t_vol_t GlobalFlow::Model::NodeInterface::getRHS ( ) [inline], [noexcept]`

The right hand side of the equation.

#### Returns

volume per time

Definition at line 1045 of file Node.hpp.

7.28.3.22 `double GlobalFlow::Model::NodeInterface::getRHS__NWT ( ) [inline], [noexcept]`

The right hand side of the equation (NWT)

#### Returns

volume per time

Definition at line 1064 of file Node.hpp.

7.28.3.23 `t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity ( ) [inline], [noexcept]`

Storage capacity based on yield or specific storage.

#### Returns

Potential flow budget when multiplied by head change Uses an 0.001m epsilon to determine if a water-table condition is present. If the layer is confined or not in water-table condition returns primary capacity.

Definition at line 512 of file Node.hpp.

7.28.3.24 `t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity__Primary ( ) [inline], [protected], [noexcept]`

Uses specific storage to calculate storativity.



**Returns**

Flow budget for cell depending on head change

Definition at line 191 of file Node.hpp.

**7.28.3.25** `t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity__Secondary ( ) [inline], [protected], [noexcept]`

Uses specific yield to calculate storativity.

**Returns**

Flow budget for cell depending on head change

Definition at line 201 of file Node.hpp.

**7.28.3.26** `t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity__SecondaryNWT ( ) [inline], [protected], [noexcept]`

Uses specific yield with NWT smoother.

**Returns**

Flow budget for cell depending on head change Assumes that smooth function is linear

Definition at line 210 of file Node.hpp.

Here is the call graph for this function:

**7.28.3.27** `t_vol_t GlobalFlow::Model::NodeInterface::getTotalStorageFlow ( ) [inline], [noexcept]`

Get flow budget based on head change.

**Returns**

Flow volume Note: Water entering storage is treated as an outflow (-), that is a loss of water from the flow system while water released from storage is treated as inflow (+), that is a source of water to the flow system

Definition at line 557 of file Node.hpp.

**7.28.3.28** `quantity<Velocity> GlobalFlow::Model::NodeInterface::getVelocity ( map_itter pos ) [inline]`

Calculate the lateral flow velocity

**Parameters**

<i>pos</i>	
------------	--

**Returns**

Definition at line 1122 of file Node.hpp.

**7.28.3.29** `std::pair<double, double> GlobalFlow::Model::NodeInterface::getVelocityVector ( ) [inline]`

Calculate flow velocity for flow tracking Vx and Vy represent the flow velocity in x and y direction. A negative value represents a flow in the opposite direction.

#### Returns

Velocity vector (x,y)

Definition at line 1147 of file Node.hpp.

**7.28.3.30** `bool GlobalFlow::Model::NodeInterface::hasOcean ( ) [inline]`

Check for type ocean.

#### Returns

bool

Definition at line 827 of file Node.hpp.

**7.28.3.31** `bool GlobalFlow::Model::NodeInterface::hasRiver ( ) [inline]`

Check for type river.

#### Returns

bool

Definition at line 821 of file Node.hpp.

**7.28.3.32** `bool GlobalFlow::Model::NodeInterface::hasTypeOfExternalFlow ( FlowType type ) [inline]`

Check for an external flow by type.

#### Parameters

<i>type</i>	The flow type
-------------	---------------

#### Returns

bool

Definition at line 754 of file Node.hpp.

**7.28.3.33** `void GlobalFlow::Model::NodeInterface::removeExternalFlow ( FlowType type ) [inline]`

Remove an external flow to the cell by id.

#### Parameters

<i>ID</i>	The flow id
-----------	-------------

Definition at line 743 of file Node.hpp.

**7.28.3.34** `bool GlobalFlow::Model::NodeInterface::resetFloodingHead ( ) [inline], [noexcept]`

Cuts off all heads above surface elevation.

#### Warning

Should only be used in spinn up phase!

#### Returns

Bool if node was reset

Definition at line 414 of file Node.hpp.

**7.28.3.35** `void GlobalFlow::Model::NodeInterface::scaleDynamicRivers ( double mult ) [inline]`

Scale dyn rivers for sensitivity

#### Parameters

<i>mult</i>	
-------------	--

Definition at line 780 of file Node.hpp.

**7.28.3.36** `void GlobalFlow::Model::NodeInterface::scaleRiverConduct ( ) [inline]`

Scales river conduct by 50%.

#### Warning

Should only be used in spinn up phase

Definition at line 427 of file Node.hpp.

**7.28.3.37** `void GlobalFlow::Model::NodeInterface::setEfold ( double efold ) [inline]`

Set e-folding factor from data on all layers.

#### Parameters

<i>e-fold</i>	
---------------	--

Definition at line 314 of file Node.hpp.

Here is the call graph for this function:

**7.28.3.38** `void GlobalFlow::Model::NodeInterface::setElevation ( t_meter elevation ) [inline]`

Set elevation on top layer and propagate to lower layers.

#### Parameters

<i>elevation</i>	The top elevation (e.g. from DEM)
------------------	-----------------------------------

Definition at line 279 of file Node.hpp.

**7.28.3.39** `void GlobalFlow::Model::NodeInterface::setEqHead ( t_meter wtd ) [inline]`

Calculated equilibrium groundwater-head from eq\_wtd Assumes that if initialhead = false that the eq\_head is also used as initial head.

## Parameters

<i>head</i>	
-------------	--

Definition at line 328 of file Node.hpp.

#### 7.28.3.40 void GlobalFlow::Model::NodeInterface::setK ( t\_vel *conduct* ) [inline]

Modify hydraulic conductivity (applied to all layers below)

## Parameters

<i>New</i>	conductivity (if e-folding enabled scaled on layers)
------------	--

Definition at line 474 of file Node.hpp.

#### 7.28.3.41 void GlobalFlow::Model::NodeInterface::setK\_direct ( t\_vel *conduct* ) [inline]

Modify hydraulic conductivity (no e-folding, no layers)

## Parameters

<i>New</i>	conductivity
------------	--------------

Definition at line 485 of file Node.hpp.

#### 7.28.3.42 void GlobalFlow::Model::NodeInterface::setNeighbour ( unsigned long *ID*, NeighbourPosition *neighbour* ) [inline]

Add a neighbour.

## Parameters

<i>ID</i>	The internal ID and position in vector
<i>neighbour</i>	The position relative to the cell

Definition at line 680 of file Node.hpp.

#### 7.28.3.43 void GlobalFlow::Model::NodeInterface::setSlope ( double *slope\_percent* ) [inline]

Set slope from data on all layers Slope input is in % but is required as absolut thus: slope = sloper\_percent / 100.

## Parameters

<i>slope</i>	
--------------	--

Definition at line 299 of file Node.hpp.

Here is the call graph for this function:

#### 7.28.3.44 void GlobalFlow::Model::NodeInterface::toggleStadyState ( bool *onOFF* ) [inline]

Toggle steady state simulation.

## Parameters

<i>onOFF</i>	true=on Turns all storage equations to zero with no timesteps
--------------	---

Definition at line 502 of file Node.hpp.

7.28.3.45 void GlobalFlow::Model::NodeInterface::updateExternalFlowConduct ( double *amount*, FlowType *type* )  
[inline]

Update wetlands, lakes.

## Parameters

<i>amount</i>	
<i>type</i>	

Definition at line 792 of file Node.hpp.

**7.28.3.46** `void GlobalFlow::Model::NodeInterface::updateHeadChange ( ) [inline], [noexcept]`

Update the current head change (in comparison to last time step)

## Note

Should only be caled at end of timestep

Definition at line 435 of file Node.hpp.

**7.28.3.47** `void GlobalFlow::Model::NodeInterface::updateLakeBottoms ( double amount ) [inline]`

Update lake bottoms Used for sensitivity.

## Parameters

<i>amount</i>	
---------------	--

Definition at line 807 of file Node.hpp.

**7.28.3.48** `void GlobalFlow::Model::NodeInterface::updateUniqueFlow ( double amount, FlowType flow = RECHARGE ) [inline]`

Updates GW recharge Curently assumes only one recharge as external flow!

## Parameters

<i>amount</i>	The new flow amount
---------------	---------------------

Definition at line 763 of file Node.hpp.

## 7.28.4 Member Data Documentation

**7.28.4.1** `bool GlobalFlow::Model::NodeInterface::cached {false}`

Calculated equilibrium flow to neighbouring cells Static thus calculated only once.

Depends on: K in cell and eq\_head in all 6 neighbours

Definition at line 348 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp
- /home/robert/development/G3M-framework/src/Model/Node.cpp

## 7.29 GlobalFlow::Model::NodeInterface::NodeNotFoundException Class Reference

Inheritance diagram for GlobalFlow::Model::NodeInterface::NodeNotFoundException:

Collaboration diagram for GlobalFlow::Model::NodeInterface::NodeNotFoundException:

### 7.29.1 Detailed Description

Definition at line 684 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

## 7.30 GlobalFlow::Simulation::Options Class Reference

```
#include <Options.hpp>
```

### Public Types

- enum **BoundaryCondition** { **CONSTANT\_HEAD\_SEA\_LEVEL**, **CONSTANT\_HEAD\_NEIGHBOUR**, **STATIONARY\_HEAD\_SEA\_LEVEL** }

### Public Member Functions

- void **setClosingCrit** (double crit)
- void **setDamping** (bool set)
- bool **isDampingEnabled** ()
- double **getMinDamp** ()
- double **getMaxDamp** ()
- double **getMaxHeadChange** ()
- bool **isConfined** (int layer)
- vector< bool > **getConfinements** ()
- BoundaryCondition **getBoundaryCondition** ()
- bool **isSensitivity** ()
- bool **isKFromLith** ()
- bool **isKOceanFile** ()
- bool **isSpecificStorageFile** ()
- bool **isSpecificYieldFile** ()
- bool **isKRiverFile** ()
- bool **isAquiferDepthDile** ()
- string **getKDir** ()
- string **getKRiverDir** ()
- string **getKOceanDir** ()
- string **getSSDir** ()
- string **getSYDir** ()
- string **getAQDepthDir** ()
- bool **isRowCol** ()
- int **getInnerItter** ()
- long **getNumberOfNodes** ()
- int **getNumberOfLayers** ()
- int **getMaxIterations** ()
- double **getConvergenceCriteria** ()
- string **getSolverName** ()
- string **getNodesDir** ()
- string **getElevation** ()
- string **getEfolding** ()
- string **getEqWTD** ()
- string **getSlope** ()

- string **getBlue** ()
- vector< string > **getElevation\_A** ()
- vector< string > **getEfoldg\_a** ()
- vector< string > **getEqWTD\_a** ()
- vector< string > **getSlope\_a** ()
- vector< string > **getBlue\_a** ()
- string **getRecharge** ()
- string **getLithology** ()
- string **getRivers** ()
- string **getGlobalLakes** ()
- string **getGlobalWetlands** ()
- string **getLocalLakes** ()
- string **getLocalWetlands** ()
- string **getMapping** ()
- int **getThreads** ()
- const bool **adaptiveStepsizeEnabled** ()
- const int **getStepsizeModifier** () throw (out\_of\_range)
- bool **cacheEnabled** ()
- int **getInitialHead** ()
- double **getInitialK** ()
- double **getOceanConduct** ()
- vector< int > **getAquiferDepth** ()
- double **getAnisotropy** ()
- double **getSpecificYield** ()
- double **getSpecificStorage** ()
- void **load** (const std::string &filename)
- void **save** (const std::string &filename)

### 7.30.1 Detailed Description

Reads simulation options from a JSON file Defines getters and setters for options

Definition at line 37 of file Options.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Options.hpp
- /home/robert/development/G3M-framework/src/Simulation/Options.cpp

## 7.31 GlobalFlow::DataProcessing::DataOutput::OutputFactory< T > Class Template Reference

### Static Public Member Functions

- static [OutputInterface](#)< T > \* **getOutput** (OutputType type)

### 7.31.1 Detailed Description

`template<typename T>class GlobalFlow::DataProcessing::DataOutput::OutputFactory< T >`

Definition at line 312 of file OutputFactory.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp



## 7.32 GlobalFlow::DataProcessing::DataOutput::OutputField< T > Class Template Reference

### Public Member Functions

- **OutputField** (path filePath, bool printID, bool printXY, [Simulation::Simulation](#) const &sim, OutputType outputType, FieldType fieldType)
- void **write** ()

### 7.32.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::OutputField< T >

Definition at line 53 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

## 7.33 GlobalFlow::DataProcessing::DataOutput::OutputInterface< T > Class Template Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >:

### Public Member Functions

- virtual void **write** (path filePath, bool printID, bool printXY, std::vector< T > data, pos\_v p)=0

### 7.33.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >

Definition at line 52 of file OutputFactory.hpp.

### 7.33.2 Member Function Documentation

7.33.2.1 template<typename T> virtual void GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >::write ( path filePath, bool printID, bool printXY, std::vector< T > data, pos\_v p ) [pure virtual]

Needs to be implemented

#### Parameters

<i>filePath</i>	
<i>printID</i>	Bool
<i>printXY</i>	Bool
<i>data</i>	Data vector
<i>p</i>	<a href="#">Position</a> vector

Implemented in [GlobalFlow::DataProcessing::DataOutput::GFS\\_JSONOutput< T >](#), and [GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >](#).

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

## 7.34 GlobalFlow::DataProcessing::DataOutput::OutputManager Class Reference

### Public Member Functions

- **OutputManager** (path output\_spec\_path, [Simulation::Simulation](#) const &sim)
- void [write](#) ()  
*Visits all registered output options and triggers write.*

#### 7.34.1 Detailed Description

Definition at line 170 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

## 7.35 GlobalFlow::DataProcessing::DataOutput::Outputvisitor Class Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::Outputvisitor:

Collaboration diagram for GlobalFlow::DataProcessing::DataOutput::Outputvisitor:

### Public Member Functions

- template<typename T >  
void **operator()** (T &operand) const

#### 7.35.1 Detailed Description

Definition at line 41 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

## 7.36 GlobalFlow::Model::PhysicalProperty< Type, Key > Class Template Reference

```
#include <PhysicalProperties.hpp>
```

### Protected Member Functions

- Type & **\_\_get** () const
- Type **\_\_get** ()
- void **\_\_set** (const Type &value)
- template<class... Args>  
void **\_\_emplace** (Args &&...args)

### 7.36.1 Detailed Description

```
template<class Type, class Key = DefaultProperty>class GlobalFlow::Model::PhysicalProperty< Type, Key >
```

Actual container holding the physical propertie

Definition at line 47 of file PhysicalProperties.hpp.

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

- /home/robert/development/G3M-framework/src/Model/PhysicalProperties.hpp

## 7.37 Position Class Reference

### Public Member Functions

- **Position** (double lat, double lon)

### Public Attributes

- const double **lat** {0}
- const double **lon** {0}

### 7.37.1 Detailed Description

Definition at line 83 of file Helpers.hpp.

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

- /home/robert/development/G3M-framework/src/Misc/Helpers.hpp

## 7.38 PropertyRepository Class Reference

```
#include <PhysicalProperties.hpp>
```

### 7.38.1 Detailed Description

Type repository holding the properties

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

- /home/robert/development/G3M-framework/src/Model/PhysicalProperties.hpp

## 7.39 GlobalFlow::Model::PropertyRepository< PhysicalProperties > Class Template Reference

Inheritance diagram for GlobalFlow::Model::PropertyRepository< PhysicalProperties >:

Collaboration diagram for GlobalFlow::Model::PropertyRepository< PhysicalProperties >:

## Public Member Functions

- `template<class Type , class Key = DefaultProperty>`  
`Type & get () const`
- `template<class Type , class Key = DefaultProperty>`  
`Type get ()`
- `template<class Type , class Key = DefaultProperty>`  
`void set (const Type &value)`
- `template<class Type , class Key = DefaultProperty, class... Args>`  
`void emplace (Args &&...args)`
- `template<class Type , class Key = DefaultProperty>`  
`void addTo (const Type &value)`

### 7.39.1 Detailed Description

`template<class... PhysicalProperties> class GlobalFlow::Model::PropertyRepository< PhysicalProperties >`

Definition at line 88 of file PhysicalProperties.hpp.

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

- `/home/robert/development/G3M-framework/src/Model/PhysicalProperties.hpp`

## 7.40 SaticHeadNode Class Reference

```
#include <Node.hpp>
```

### 7.40.1 Detailed Description

A node without changing head Can be used as boundary condition

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

- `/home/robert/development/G3M-framework/src/Model/Node.hpp`

## 7.41 GlobalFlow::Simulation::Simulation Class Reference

```
#include <Simulation.hpp>
```

## Public Types

- enum **Flows** {  
    **RIVERS** = 1, **DRAINS**, **RIVER\_MM**, **LAKES**,  
    **WETLANDS**, **GLOBAL\_WETLANDS**, **RECHARGE**, **FASTSURFACE**,  
    **NAG**, **STORAGE**, **GENERAL\_HEAD\_BOUNDARY** }

## Public Member Functions

- **Simulation** ([Options](#) op, [DataReader](#) \*reader)
- [Solver::Equation](#) \* **getEquation** ()
- void **save** ()
- std::string [NodeInfosByID](#) (unsigned long nodeID)

- template<int FieldNum>  
std::string [getFlowSumByIDs](#) (std::array< int, FieldNum > ids)
- std::string [NodeFlowsByID](#) (unsigned long nodeID)
- template<class FunOut , class FunIn >  
[MassError](#) [getError](#) (FunOut fun1, FunIn fun2)
- [MassError](#) [getMassError](#) ()
- [MassError](#) [getCurrentMassError](#) ()
- double [getLossToRivers](#) ()
- template<class Fun >  
[MassError](#) [getError](#) (Fun fun)
- string [getFlowByName](#) (Flows flow)
- void [printMassBalances](#) ()
- const double **calcRecharge** (const double recharge, const double &area)
- [DataReader](#) \* **getDataReader** ()
- NodeVector & **getNodes** ()
- void **scaleByIds** (vector< int > ids, string field, double mult)
- template<class Fun , class ChangeFunction >  
void **scaleByFunction** (Fun fun, ChangeFunction apply)
- template<class Fun >  
void [scaleByFunction](#) (Fun fun, string field, double mult)
- void [readMultipliers](#) (string path)
- void [writeResiduals](#) (string path)
- template<typename DataArray >  
void [updateGWRechargeFromWaterGAP](#) (DataArray field, string fieldname, short month, int numberOfGridCells)

## Public Attributes

- NodeVector **nodes**

### 7.41.1 Detailed Description

The simulation class which holds the equation, options and data instance Further contains methods for calculating the mass balance and sensitivity methods TODO: Clean me up!

Definition at line 70 of file Simulation.hpp.

### 7.41.2 Member Function Documentation

#### 7.41.2.1 [MassError](#) GlobalFlow::Simulation::Simulation::getCurrentMassError ( ) [\[inline\]](#)

Get the mass balance for the current step

Returns

Definition at line 222 of file Simulation.hpp.

#### 7.41.2.2 template<class FunOut , class FunIn > [MassError](#) GlobalFlow::Simulation::Simulation::getError ( FunOut *fun1*, FunIn *fun2* ) [\[inline\]](#)

Calculate the mass error

## Parameters

<i>fun1</i>	Function to get OutFlow
<i>fun2</i>	Function to get InFlow

## Returns

Definition at line 190 of file Simulation.hpp.

**7.41.2.3** `template<class Fun > MassError GlobalFlow::Simulation::Simulation::getError ( Fun fun ) [inline]`

Decide if its an In or Outflow

## Parameters

<i>fun</i>	
------------	--

## Returns

Definition at line 265 of file Simulation.hpp.

**7.41.2.4** `string GlobalFlow::Simulation::Simulation::getFlowByName ( Flows flow ) [inline]`

Helper function for printing the mass balance for each flow

## Parameters

<i>flow</i>	
-------------	--

## Returns

Definition at line 281 of file Simulation.hpp.

**7.41.2.5** `template<int FieldNum> std::string GlobalFlow::Simulation::Simulation::getFlowSumByIDs ( std::array< int, FieldNum > ids ) [inline]`

Get budget per node

## Parameters

<i>ids</i>	
------------	--

## Returns

Definition at line 131 of file Simulation.hpp.

**7.41.2.6** `double GlobalFlow::Simulation::Simulation::getLossToRivers ( ) [inline]`

Get the flow lost to external flows

## Returns

Definition at line 235 of file Simulation.hpp.

#### 7.41.2.7 **MassError** GlobalFlow::Simulation::Simulation::getMassError ( ) [inline]

Get the total mass balance

## Returns

Definition at line 209 of file Simulation.hpp.

#### 7.41.2.8 **std::string** GlobalFlow::Simulation::Simulation::NodeFlowsByID ( unsigned long *nodeID* ) [inline]

Return all external flows seperatly

Definition at line 150 of file Simulation.hpp.

Here is the call graph for this function:

#### 7.41.2.9 **std::string** GlobalFlow::Simulation::Simulation::NodeInfosByID ( unsigned long *nodeID* ) [inline]

Get basic node information by its id

## Parameters

<i>nodeID</i>	
---------------	--

## Returns

A string of information

Definition at line 88 of file Simulation.hpp.

Here is the call graph for this function:

#### 7.41.2.10 **void** GlobalFlow::Simulation::Simulation::printMassBalances ( ) [inline]

Prints all mass balances

Definition at line 344 of file Simulation.hpp.

#### 7.41.2.11 **void** GlobalFlow::Simulation::Simulation::readMultipliers ( string *path* ) [inline]

Scale values for sensitivity analysis

## Parameters

<i>path</i>	
-------------	--

Definition at line 497 of file Simulation.hpp.

#### 7.41.2.12 **template**<class Fun > **void** GlobalFlow::Simulation::Simulation::scaleByFunction ( Fun *fun*, string *field*, double *mult* ) [inline]

Helper function for sensitivity

## Parameters

<i>fun</i>	
<i>field</i>	
<i>mult</i>	

Definition at line 415 of file Simulation.hpp.

7.41.2.13 `template<typename DataArray > void GlobalFlow::Simulation::Simulation::updateGWRechargeFromWaterGAP ( DataArray field, string fieldname, short month, int numberOfGridCells ) [inline]`

Coupling function for WaterGAP

## Note

Under development

## Parameters

<i>field</i>	
<i>fieldname</i>	
<i>month</i>	
<i>numberOfGridCells</i>	

Definition at line 582 of file Simulation.hpp.

7.41.2.14 `void GlobalFlow::Simulation::Simulation::writeResiduals ( string path ) [inline]`

Get the residuals of the current iteration

## Parameters

<i>path</i>	
-------------	--

Definition at line 557 of file Simulation.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Simulation.hpp
- /home/robert/development/G3M-framework/src/Simulation/Simulation.cpp

## 7.42 GlobalFlow::Simulation::SimulationOutput Class Reference

### Public Member Functions

- **SimulationOutput** (int field\_count)
- `template<typename Query , class... FieldNames>`  
void **query** (NodeVector nodes, Query query, FieldNames...names)

### Friends

- ostream & **operator**<< (ostream &os, [SimulationOutput](#) const &simulation)



### 7.42.1 Detailed Description

Definition at line 35 of file SimulationOutput.hpp.

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

- /home/robert/development/G3M-framework/src/Misc/SimulationOutput.hpp

## 7.43 GlobalFlow::Logging::Singleton< C > Class Template Reference

### Static Public Member Functions

- static C \* **instance** ()

### 7.43.1 Detailed Description

```
template<typename C>class GlobalFlow::Logging::Singleton< C >
```

Definition at line 46 of file Logging.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Logging.hpp

## 7.44 GlobalFlow::Model::StandardNode Class Reference

```
#include <Node.hpp>
```

Inheritance diagram for GlobalFlow::Model::StandardNode:

Collaboration diagram for GlobalFlow::Model::StandardNode:

### Public Member Functions

- **StandardNode** (std::shared\_ptr< std::vector< std::unique\_ptr< [NodeInterface](#) >>> nodes, double lat, double lon, t\_s\_meter area, unsigned long ArcID, unsigned long ID, t\_vel K, int stepmodifier, double aquiferDepth, double anisotropy, double specificYield, double specificStorage, bool confined)

### Friends

- class **NodeInterface**

### Additional Inherited Members

### 7.44.1 Detailed Description

[A](#) standard groundwater node

Definition at line 1183 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

## 7.45 GlobalFlow::Model::StaticHeadNode Class Reference

Inheritance diagram for GlobalFlow::Model::StaticHeadNode:

Collaboration diagram for GlobalFlow::Model::StaticHeadNode:

### Public Member Functions

- **StaticHeadNode** (std::shared\_ptr< std::vector< std::unique\_ptr< [NodeInterface](#) >>> nodes, unsigned long ID, t\_s\_meter area)

### Friends

- class **NodeInterface**

### Additional Inherited Members

#### 7.45.1 Detailed Description

Definition at line 1261 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

## 7.46 GlobalFlow::Simulation::Stepper Class Reference

Inheritance diagram for GlobalFlow::Simulation::Stepper:

Collaboration diagram for GlobalFlow::Simulation::Stepper:

### Public Member Functions

- **Stepper** ([Solver::Equation](#) \*eq, const TimeFrame time, const size\_t steps)
- virtual [Solver::Equation](#) \* **get** (int col) const
- [Iterator](#) **begin** () const
- [Iterator](#) **end** () const
- const TimeFrame **getStepSize** ()

#### 7.46.1 Detailed Description

holding the simulation iterator

Definition at line 76 of file Stepper.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Stepper.hpp

## 7.47 Write Class Reference

### 7.47.1 Detailed Description

a pseudo GFS json-like file

#### Note

provides additional methods for downscaling

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

## 7.48 Writes Class Reference

### 7.48.1 Detailed Description

a CSV file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

## 7.49 Writes Class Reference

### 7.49.1 Detailed Description

a CSV file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

## 7.50 Writes Interface Reference

### 7.50.1 Detailed Description

a file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp



# Index

- A, [13](#)
- Abstract, [13](#)
- addExternalFlow
  - GlobalFlow::Model::NodeInterface, [36](#)
- build
  - GlobalFlow::DataProcessing::DataOutput::Field↔Factory, [23](#)
- cached
  - GlobalFlow::Model::NodeInterface, [46](#)
- calcDeltaV
  - GlobalFlow::Model::FluidMechanics, [24](#)
- calcLateralFlows
  - GlobalFlow::Model::NodeInterface, [36](#)
- calculateDewateredFlow
  - GlobalFlow::Model::NodeInterface, [36](#)
- calculateExternalFlowVolume
  - GlobalFlow::Model::NodeInterface, [36](#)
- calculateHarmonicMeanConductance
  - GlobalFlow::Model::FluidMechanics, [24](#)
- calculateNotHeadDependantFlows
  - GlobalFlow::Model::NodeInterface, [37](#)
- calculateVerticalConductance
  - GlobalFlow::Model::FluidMechanics, [24](#)
- ExternalFlow
  - GlobalFlow::Model::ExternalFlow, [19](#)
- Factory, [21](#)
- Field
  - GlobalFlow, [11](#)
- flowsIsHeadDependant
  - GlobalFlow::Model::ExternalFlow, [20](#)
- get
  - GlobalFlow::DataProcessing::DataOutput::Field↔Collector, [22](#)
- getConductance
  - GlobalFlow::Model::NodeInterface, [37](#)
- getCurrentIN
  - GlobalFlow::Model::NodeInterface, [37](#)
- getCurrentMassError
  - GlobalFlow::Simulation::Simulation, [53](#)
- getCurrentOUT
  - GlobalFlow::Model::NodeInterface, [37](#)
- getDerivate\_\_NWT
  - GlobalFlow::Model::FluidMechanics, [25](#)
- getEqFlow
  - GlobalFlow::Model::NodeInterface, [37](#)
- getError
  - GlobalFlow::Simulation::Simulation, [53](#), [54](#)
  - GlobalFlow::Solver::Equation, [17](#)
- getExternalFlowByName
  - GlobalFlow::Model::NodeInterface, [38](#)
- getExternalFlowVolumeByName
  - GlobalFlow::Model::NodeInterface, [38](#)
- getFlow
  - GlobalFlow::Model::NodeInterface, [38](#)
- getFlowByName
  - GlobalFlow::Simulation::Simulation, [54](#)
- getFlowSumByIds
  - GlobalFlow::Simulation::Simulation, [54](#)
- getHCOF
  - GlobalFlow::Model::FluidMechanics, [25](#)
- getItter
  - GlobalFlow::Solver::Equation, [17](#)
- getJacobian
  - GlobalFlow::Model::NodeInterface, [38](#)
- getK
  - GlobalFlow::Model::NodeInterface, [39](#)
- getK\_vertical
  - GlobalFlow::Model::NodeInterface, [39](#)
- getLateralFlows
  - GlobalFlow::Model::NodeInterface, [39](#)
- getLossToRivers
  - GlobalFlow::Simulation::Simulation, [54](#)
- getMassError
  - GlobalFlow::Simulation::Simulation, [55](#)
- getNeighbour
  - GlobalFlow::Model::NodeInterface, [39](#)
- getNonStorageFlow
  - GlobalFlow::Model::NodeInterface, [39](#)
- getP
  - GlobalFlow::Model::ExternalFlow, [21](#)
  - GlobalFlow::Model::NodeInterface, [39](#)
- getQ
  - GlobalFlow::Model::ExternalFlow, [21](#)
  - GlobalFlow::Model::NodeInterface, [40](#)
- getRHS
  - GlobalFlow::Model::NodeInterface, [40](#)
- getRHS\_\_NWT
  - GlobalFlow::Model::NodeInterface, [40](#)
- getStorageCapacity
  - GlobalFlow::Model::NodeInterface, [40](#)
- getStorageCapacity\_\_Primary
  - GlobalFlow::Model::NodeInterface, [40](#)
- getStorageCapacity\_\_Secondary
  - GlobalFlow::Model::NodeInterface, [41](#)
- getStorageCapacity\_\_SecondaryNWT

- GlobalFlow::Model::NodeInterface, 41
- getTotalStorageFlow
  - GlobalFlow::Model::NodeInterface, 41
- getVelocity
  - GlobalFlow::Model::NodeInterface, 41
- getVelocityVector
  - GlobalFlow::Model::NodeInterface, 41
- GlobalFlow, 11
  - Field, 11
- GlobalFlow::Callback, 14
- GlobalFlow::Container, 14
- GlobalFlow::DataProcessing::DataOutput::CSV↔
  - Output< T >, 15
- GlobalFlow::DataProcessing::DataOutput::Field↔
  - Collector, 22
  - get, 22
- GlobalFlow::DataProcessing::DataOutput::FieldFactory, 22
  - build, 23
- GlobalFlow::DataProcessing::DataOutput::GFS\_JSO↔
  - NOutput
  - write, 26
- GlobalFlow::DataProcessing::DataOutput::GFS\_JSO↔
  - NOutput< T >, 26
- GlobalFlow::DataProcessing::DataOutput::Internal↔
  - TypeHash, 29
- GlobalFlow::DataProcessing::DataOutput::NETCDF↔
  - Output
  - write, 31
- GlobalFlow::DataProcessing::DataOutput::NETCDF↔
  - Output< T >, 31
- GlobalFlow::DataProcessing::DataOutput::Output↔
  - Factory< T >, 48
- GlobalFlow::DataProcessing::DataOutput::Output↔
  - Field< T >, 49
- GlobalFlow::DataProcessing::DataOutput::Output↔
  - Interface
  - write, 49
- GlobalFlow::DataProcessing::DataOutput::Output↔
  - Interface< T >, 49
- GlobalFlow::DataProcessing::DataOutput::Output↔
  - Manager, 50
- GlobalFlow::DataProcessing::DataOutput::Outputvisitor, 50
- GlobalFlow::DataReader, 15
  - loopFiles, 16
  - readData, 16
  - readTwoColumns, 16
  - readZeroPointFiveToFiveMin, 16
- GlobalFlow::GW\_Interface, 26
  - loadSettings, 27
  - setupCallBack, 27
  - setupSimulation, 27
  - simulate, 27
  - writeData, 27
- GlobalFlow::Logging::InfoSinkInterface, 28
- GlobalFlow::Logging::Logger, 30
- GlobalFlow::Logging::LoggerInterface, 30
- GlobalFlow::Logging::Singleton< C >, 57
- GlobalFlow::Model::ExternalFlow, 18
  - ExternalFlow, 19
  - flowsHeadDependant, 20
  - getP, 21
  - getQ, 21
- GlobalFlow::Model::FlowTypeHash, 23
- GlobalFlow::Model::FluidMechanics, 23
  - calcDeltaV, 24
  - calculateHarmonicMeanConductance, 24
  - calculateVerticalConductance, 24
  - getDerivate\_\_NWT, 25
  - getHCOF, 25
  - smoothFunction\_\_NWT, 25
- GlobalFlow::Model::NodeInterface, 32
  - addExternalFlow, 36
  - cached, 46
  - calcLateralFlows, 36
  - calculateDewateredFlow, 36
  - calculateExternalFlowVolume, 36
  - calculateNotHeadDependantFlows, 37
  - getConductance, 37
  - getCurrentIN, 37
  - getCurrentOUT, 37
  - getEqFlow, 37
  - getExternalFlowByName, 38
  - getExternalFlowVolumeByName, 38
  - getFlow, 38
  - getJacobian, 38
  - getK, 39
  - getK\_vertical, 39
  - getLateralFlows, 39
  - getNeighbour, 39
  - getNonStorageFlow, 39
  - getP, 39
  - getQ, 40
  - getRHS, 40
  - getRHS\_\_NWT, 40
  - getStorageCapacity, 40
  - getStorageCapacity\_\_Primary, 40
  - getStorageCapacity\_\_Secondary, 41
  - getStorageCapacity\_\_SecondaryNWT, 41
  - getTotalStorageFlow, 41
  - getVelocity, 41
  - getVelocityVector, 41
  - hasOcean, 42
  - hasRiver, 42
  - hasTypeOfExternalFlow, 42
  - NodeInterface, 35
  - removeExternalFlow, 42
  - resetFloodingHead, 42
  - scaleDynamicRivers, 43
  - scaleRiverConduct, 43
  - setEfold, 43
  - setElevation, 43
  - setEqHead, 43
  - setK, 44
  - setK\_direct, 44

- setNeighbour, [44](#)
  - setSlope, [44](#)
  - toggleSteadyState, [44](#)
  - updateExternalFlowConduct, [44](#)
  - updateHeadChange, [46](#)
  - updateLakeBottoms, [46](#)
  - updateUniqueFlow, [46](#)
- GlobalFlow::Model::NodeInterface::NodeNotFound↔
  - Exception, [46](#)
- GlobalFlow::Model::PhysicalProperty< Type, Key >, [50](#)
- GlobalFlow::Model::PropertyRepository< Physical↔
  - Properties >, [51](#)
- GlobalFlow::Model::StandardNode, [57](#)
- GlobalFlow::Model::StaticHeadNode, [58](#)
- GlobalFlow::Simulation::AbstractStepper, [13](#)
- GlobalFlow::Simulation::Iterator, [29](#)
- GlobalFlow::Simulation::MassError, [30](#)
- GlobalFlow::Simulation::Options, [47](#)
- GlobalFlow::Simulation::Simulation, [52](#)
  - getCurrentMassError, [53](#)
  - getError, [53, 54](#)
  - getFlowByName, [54](#)
  - getFlowSumByIDs, [54](#)
  - getLossToRivers, [54](#)
  - getMassError, [55](#)
  - NodeFlowsByID, [55](#)
  - NodeInfosByID, [55](#)
  - printMassBalances, [55](#)
  - readMultipliers, [55](#)
  - scaleByFunction, [55](#)
  - updateGWRechargeFromWaterGAP, [56](#)
  - writeResiduals, [56](#)
- GlobalFlow::Simulation::SimulationOutput, [56](#)
- GlobalFlow::Simulation::Stepper, [58](#)
- GlobalFlow::Solver::AdaptiveDamping, [14](#)
- GlobalFlow::Solver::Equation, [17](#)
  - getError, [17](#)
  - getIter, [17](#)
  - operator<<, [18](#)
  - solve, [17](#)
  - toggleSteadyState, [18](#)
  - updateStepSize, [18](#)
- hasOcean
  - GlobalFlow::Model::NodeInterface, [42](#)
- hasRiver
  - GlobalFlow::Model::NodeInterface, [42](#)
- hasTypeOfExternalFlow
  - GlobalFlow::Model::NodeInterface, [42](#)
- Interface, [28](#)
- Is< T >, [29](#)
- loadSettings
  - GlobalFlow::GW\_Interface, [27](#)
- loopFiles
  - GlobalFlow::DataReader, [16](#)
- NodeFlowsByID
  - GlobalFlow::Simulation::Simulation, [55](#)
- NodeInfosByID
  - GlobalFlow::Simulation::Simulation, [55](#)
- NodeInterface
  - GlobalFlow::Model::NodeInterface, [35](#)
- operator<<
  - GlobalFlow::Solver::Equation, [18](#)
- Position, [51](#)
- printMassBalances
  - GlobalFlow::Simulation::Simulation, [55](#)
- PropertyRepository, [51](#)
- readData
  - GlobalFlow::DataReader, [16](#)
- readMultipliers
  - GlobalFlow::Simulation::Simulation, [55](#)
- readTwoColumns
  - GlobalFlow::DataReader, [16](#)
- readZeroPointFiveToFiveMin
  - GlobalFlow::DataReader, [16](#)
- removeExternalFlow
  - GlobalFlow::Model::NodeInterface, [42](#)
- resetFloodingHead
  - GlobalFlow::Model::NodeInterface, [42](#)
- SaticHeadNode, [52](#)
- scaleByFunction
  - GlobalFlow::Simulation::Simulation, [55](#)
- scaleDynamicRivers
  - GlobalFlow::Model::NodeInterface, [43](#)
- scaleRiverConduct
  - GlobalFlow::Model::NodeInterface, [43](#)
- setEfold
  - GlobalFlow::Model::NodeInterface, [43](#)
- setElevation
  - GlobalFlow::Model::NodeInterface, [43](#)
- setEqHead
  - GlobalFlow::Model::NodeInterface, [43](#)
- setK
  - GlobalFlow::Model::NodeInterface, [44](#)
- setK\_direct
  - GlobalFlow::Model::NodeInterface, [44](#)
- setNeighbour
  - GlobalFlow::Model::NodeInterface, [44](#)
- setSlope
  - GlobalFlow::Model::NodeInterface, [44](#)
- setupCallBack
  - GlobalFlow::GW\_Interface, [27](#)
- setupSimulation
  - GlobalFlow::GW\_Interface, [27](#)
- simulate
  - GlobalFlow::GW\_Interface, [27](#)
- smoothFunction\_\_NWT
  - GlobalFlow::Model::FluidMechanics, [25](#)
- solve
  - GlobalFlow::Solver::Equation, [17](#)

std::hash< GlobalFlow::Model::NeighbourPosition >, [27](#)

toggleStadyState

GlobalFlow::Model::NodeInterface, [44](#)

toggleSteadyState

GlobalFlow::Solver::Equation, [18](#)

updateExternalFlowConduct

GlobalFlow::Model::NodeInterface, [44](#)

updateGWRechargeFromWaterGAP

GlobalFlow::Simulation::Simulation, [56](#)

updateHeadChange

GlobalFlow::Model::NodeInterface, [46](#)

updateLakeBottoms

GlobalFlow::Model::NodeInterface, [46](#)

updateStepSize

GlobalFlow::Solver::Equation, [18](#)

updateUniqueFlow

GlobalFlow::Model::NodeInterface, [46](#)

Write, [59](#)

write

GlobalFlow::DataProcessing::DataOutput::GFS\_  
JSONOutput, [26](#)

GlobalFlow::DataProcessing::DataOutput::NETC\_  
DFOOutput, [31](#)

GlobalFlow::DataProcessing::DataOutput::\_  
OutputInterface, [49](#)

writeData

GlobalFlow::GW\_Interface, [27](#)

writeResiduals

GlobalFlow::Simulation::Simulation, [56](#)

Writes, [59](#)