

DropletSimLibrary

Generated by Doxygen 1.8.3.1

Mon May 13 2013 02:33:50

Contents

1	DropletSimLibrary	1
1.1	Introduction	1
1.2	Dependencies	1
1.3	Installation	1
1.3.1	Step 1: Obtaining Source Code	1
1.3.2	Step 2: Building DropletSimLibrary	1
1.4	How to contribute	2
1.4.1	Issue Tracker	2
1.4.2	Contacting us	2
2	Doxygen Guide	3
2.1	Dependencies	3
2.2	Running Doxygen	3
2.3	Building a .PDF Manual	3
3	Supported Features	5
3.1	Droplets	5
3.2	Simulator	5
4	Deprecated List	7
5	Class Index	9
5.1	Class List	9
6	File Index	11
6.1	File List	11
7	Class Documentation	13
7.1	_Droplet_Actuator_Data Struct Reference	13
7.1.1	Detailed Description	13
7.1.2	Member Data Documentation	13
7.1.2.1	_oscillator	13
7.1.2.2	bOut	13
7.1.2.3	currMoveDir	13

7.1.2.4	currTurnDir	14
7.1.2.5	gOut	14
7.1.2.6	moveStepRemaining	14
7.1.2.7	moveTimeRemaining	14
7.1.2.8	rotateStepRemaining	14
7.1.2.9	rotateTimeRemaining	14
7.1.2.10	rOut	14
7.2	_Droplet_Communication_Data Struct Reference	14
7.2.1	Detailed Description	15
7.2.2	Member Data Documentation	15
7.2.2.1	commChannels	15
7.2.2.2	sendActive	15
7.3	_Droplet_Component_Data Struct Reference	15
7.3.1	Detailed Description	15
7.3.2	Member Data Documentation	15
7.3.2.1	capacitorPower	15
7.3.2.2	dropletID	15
7.3.2.3	leg1Power	16
7.3.2.4	leg2Power	16
7.3.2.5	leg3Power	16
7.4	_Droplet_Localization_Data Struct Reference	16
7.4.1	Detailed Description	16
7.4.2	Member Data Documentation	16
7.4.2.1	lastRelPosUpdate	16
7.4.2.2	movedSinceLastUpdate	16
7.4.2.3	posX	16
7.4.2.4	posY	17
7.4.2.5	posZ	17
7.4.2.6	rotA	17
7.4.2.7	rotX	17
7.4.2.8	rotY	17
7.4.2.9	rotZ	17
7.5	_Droplet_Sensor_Data Struct Reference	17
7.5.1	Detailed Description	17
7.5.2	Member Data Documentation	17
7.5.2.1	bIn	17
7.5.2.2	gIn	18
7.5.2.3	rIn	18
7.6	_Droplet_Sim_Comm_Channel_Data Struct Reference	18
7.6.1	Detailed Description	18

7.6.2	Member Data Documentation	18
7.6.2.1	inBuf	18
7.6.2.2	inMsgLength	18
7.6.2.3	lastMsgInTimestamp	18
7.6.2.4	lastMsgOutTimestamp	18
7.6.2.5	outBuf	19
7.6.2.6	outMsgLength	19
7.7	_Droplet_Timing_Data Struct Reference	19
7.7.1	Detailed Description	19
7.7.2	Member Data Documentation	19
7.7.2.1	timer	19
7.7.2.2	trigger	19
7.8	_Object_Physics_Data Struct Reference	19
7.8.1	Detailed Description	20
7.8.2	Member Data Documentation	20
7.8.2.1	_worldID	20
7.8.2.2	colShapeIndex	20
7.8.2.3	friction	20
7.8.2.4	localInertia	20
7.8.2.5	mass	20
7.9	_Simulator_Physics_Data Struct Reference	20
7.9.1	Detailed Description	21
7.9.2	Member Data Documentation	21
7.9.2.1	_colShapeIDCounter	21
7.9.2.2	_dynObjCollisionBM	21
7.9.2.3	_physicsWorldObjCounter	21
7.9.2.4	_staticObjCollisionBM	21
7.9.2.5	broadphase	21
7.9.2.6	collisionConfig	21
7.9.2.7	collisionDispatch	21
7.9.2.8	collisionShapes	21
7.9.2.9	constraintSolver	21
7.9.2.10	dynWorld	21
7.10	Droplet_Motion_Direction_Data Struct Reference	22
7.10.1	Detailed Description	22
7.10.2	Member Data Documentation	22
7.10.2.1	currMoveDir	22
7.10.2.2	currTurnDir	22
7.11	DropletSim Class Reference	22
7.11.1	Detailed Description	23

7.11.2	Constructor & Destructor Documentation	23
7.11.2.1	DropletSim	23
7.11.2.2	~DropletSim	23
7.11.3	Member Function Documentation	24
7.11.3.1	AddCollisionShape	24
7.11.3.2	AddDroplet	24
7.11.3.3	AddPhysicalObject	24
7.11.3.4	AddPhysicalObject	25
7.11.3.5	Cleanup	25
7.11.3.6	CreateFloor	26
7.11.3.7	Init	26
7.11.3.8	SetUpProjector	26
7.11.3.9	SetUpProjector	27
7.11.3.10	Step	28
7.11.4	Friends And Related Function Documentation	28
7.11.4.1	DropletSimInfo	28
7.11.5	Member Data Documentation	28
7.11.5.1	dropletPositions	28
7.11.5.2	dropletRelPos	28
7.11.5.3	droplets	28
7.11.5.4	firstRun	28
7.11.5.5	goodRand	29
7.11.5.6	objectPositions	29
7.11.5.7	physicalObjects	29
7.11.5.8	projector	29
7.11.5.9	projSet	29
7.11.5.10	timer	29
7.12	DropletSimInfo Class Reference	29
7.12.1	Detailed Description	30
7.12.2	Member Function Documentation	30
7.12.2.1	GetActuationData	30
7.12.2.2	GetCommData	31
7.12.2.3	GetCompData	31
7.12.2.4	GetDropletColors	31
7.12.2.5	GetDropletPositions	31
7.12.2.6	GetMotionDirections	32
7.12.2.7	GetObjectPositions	32
7.12.2.8	GetPhysData	32
7.12.2.9	GetRemainingMotionTimes	33
7.12.2.10	GetSensorColors	33

7.12.2.11	GetStepRT	33
7.12.2.12	GetTimeRatio	34
7.12.2.13	GetTotalDiff	34
7.12.2.14	GetTotalRT	34
7.12.2.15	GetTotalST	35
7.12.3	Friends And Related Function Documentation	35
7.12.3.1	DropletSim	35
7.13	DropletTimeControl Class Reference	36
7.13.1	Detailed Description	36
7.13.2	Member Function Documentation	37
7.13.2.1	getStepRT	37
7.13.2.2	getTimeRatio	37
7.13.2.3	getTotalDiff	38
7.13.2.4	getTotalRT	39
7.13.2.5	getTotalST	39
7.13.2.6	initTimer	40
7.13.2.7	printAll	41
7.13.2.8	printStepRT	41
7.13.2.9	printTimeRatio	41
7.13.2.10	printTotalDiff	42
7.13.2.11	printTotalRT	42
7.13.2.12	printTotalST	42
7.13.2.13	printVars	43
7.13.2.14	resetTimer	43
7.13.2.15	updateTimer	43
7.13.2.16	updateTimer	44
7.14	DSimPhysicalObject Class Reference	44
7.14.1	Detailed Description	45
7.14.2	Constructor & Destructor Documentation	45
7.14.2.1	DSimPhysicalObject	45
7.14.3	Member Function Documentation	45
7.14.3.1	_InitPhysics	45
7.14.3.2	_InitPhysics	46
7.14.4	Member Data Documentation	46
7.14.4.1	objPhysics	46
7.15	IDroplet Class Reference	46
7.15.1	Detailed Description	47
7.15.2	Constructor & Destructor Documentation	48
7.15.2.1	IDroplet	48
7.15.2.2	~IDroplet	48

7.15.3	Member Function Documentation	48
7.15.3.1	_InitPhysics	48
7.15.3.2	cancel_move	48
7.15.3.3	cancel_rotate	49
7.15.3.4	check_for_new_messages	49
7.15.3.5	check_timer	49
7.15.3.6	DropletInit	49
7.15.3.7	DropletMainLoop	49
7.15.3.8	get_droplet_id	50
7.15.3.9	get_rgb	50
7.15.3.10	ir_send	50
7.15.3.11	is_moving	50
7.15.3.12	is_rotating	50
7.15.3.13	leg1_status	51
7.15.3.14	leg2_status	51
7.15.3.15	leg3_status	51
7.15.3.16	move_duration	51
7.15.3.17	move_steps	51
7.15.3.18	rand_byte	51
7.15.3.19	reset_all_systems	52
7.15.3.20	reset_ir_sensor	52
7.15.3.21	reset_motors	52
7.15.3.22	reset_rgb_led	53
7.15.3.23	reset_rgb_sensor	53
7.15.3.24	reset_timers	53
7.15.3.25	rotate_duration	54
7.15.3.26	rotate_steps	54
7.15.3.27	set_blue_led	54
7.15.3.28	set_green_led	54
7.15.3.29	set_red_led	55
7.15.3.30	set_rgb	55
7.15.3.31	set_timer	55
7.15.4	Friends And Related Function Documentation	55
7.15.4.1	AccessActuatorData	55
7.15.4.2	AccessCommData	55
7.15.4.3	AccessCompData	56
7.15.4.4	AccessPhysicsData	56
7.15.4.5	AccessSensorData	56
7.15.4.6	AccessTimeData	56
7.15.5	Member Data Documentation	56

7.15.5.1	buf	56
7.15.5.2	data_len	57
7.15.5.3	global_rx_buffer	57
7.15.5.4	message_time	57
7.15.5.5	msg_return_order	57
7.15.5.6	printed_read_prompt	57
7.15.5.7	read	57
7.15.5.8	receivers_used	57
7.15.5.9	sender_ID	57
7.15.5.10	size	57
7.16	IDropletProjector Class Reference	57
7.16.1	Detailed Description	58
7.16.2	Constructor & Destructor Documentation	59
7.16.2.1	IDropletProjector	59
7.16.2.2	~IDropletProjector	59
7.16.3	Member Function Documentation	59
7.16.3.1	GetPixel	59
7.16.3.2	GetPixels	59
7.16.3.3	GetPixels	59
7.16.3.4	LoadFile	60
7.16.3.5	SetDirectory	60
7.16.4	Member Data Documentation	61
7.16.4.1	dataSet	61
7.16.4.2	fileDir	61
7.16.4.3	fileFormat	61
7.16.4.4	fileName	61
7.16.4.5	floorLength	61
7.16.4.6	floorWidth	61
7.16.4.7	imgData	61
7.16.4.8	imgLength	62
7.16.4.9	imgWidth	62
7.16.4.10	projLength	62
7.16.4.11	projPixelLength	62
7.16.4.12	projPixelWidth	62
7.16.4.13	projWidth	62
7.17	IDropletSimInterface Class Reference	62
7.17.1	Detailed Description	63
7.17.2	Constructor & Destructor Documentation	63
7.17.2.1	IDropletSimInterface	63
7.17.2.2	~IDropletSimInterface	63

7.17.3	Member Function Documentation	63
7.17.3.1	CreateDroplet	63
7.17.3.2	InitializeSim	63
7.17.3.3	SetDropletCollisionShape	63
7.17.4	Member Data Documentation	63
7.17.4.1	sim	63
7.18	Ran Struct Reference	63
7.18.1	Detailed Description	64
7.18.2	Constructor & Destructor Documentation	64
7.18.2.1	Ran	64
7.18.3	Member Function Documentation	64
7.18.3.1	doub	64
7.18.3.2	int64	64
7.18.4	Member Data Documentation	65
7.18.4.1	u	65
7.18.4.2	v	65
7.18.4.3	w	65
7.19	SimSetupData Class Reference	65
7.19.1	Detailed Description	65
7.19.2	Constructor & Destructor Documentation	66
7.19.2.1	SimSetupData	66
7.19.2.2	SimSetupData	66
7.19.3	Friends And Related Function Documentation	66
7.19.3.1	DropletSim	66
7.20	TrigArray Class Reference	66
7.20.1	Detailed Description	67
7.20.2	Constructor & Destructor Documentation	67
7.20.2.1	TrigArray	67
7.20.2.2	~TrigArray	67
7.20.3	Member Function Documentation	67
7.20.3.1	AddData	67
7.20.3.2	GetAngle	67
7.20.3.3	GetData	67
7.20.3.4	GetDistance	67
7.20.3.5	RemoveData	67
8	File Documentation	69
8.1	doxygen.dox File Reference	69
8.2	DropletDataStructs.h File Reference	69
8.2.1	Detailed Description	71

8.2.2	Macro Definition Documentation	71
8.2.2.1	_DROPLET_DATA_STRUCTS	71
8.2.3	Typedef Documentation	71
8.2.3.1	DropletActuatorData	71
8.2.3.2	DropletCommChannelData	71
8.2.3.3	DropletCommData	71
8.2.3.4	DropletCompData	71
8.2.3.5	DropletSensorData	71
8.2.3.6	DropletTimeData	71
8.2.3.7	GPSInfo	71
8.2.3.8	ObjectPhysicsData	71
8.2.3.9	SimPhysicsData	71
8.3	DropletSim.cpp File Reference	72
8.4	DropletSim.h File Reference	72
8.4.1	Macro Definition Documentation	73
8.4.1.1	_DROPLET_SIM	73
8.5	DropletSimGlobals.h File Reference	73
8.5.1	Detailed Description	75
8.5.2	Macro Definition Documentation	75
8.5.2.1	_DROPLET_SIM_GLOBALS	75
8.5.2.2	BROADCAST_THRESHOLD	75
8.5.2.3	DROPLET_ANGULAR_DAMPING	75
8.5.2.4	DROPLET_ID_START	75
8.5.2.5	DROPLET_LINEAR_DAMPING	75
8.5.2.6	DROPLET_NUM_TIMERS	75
8.5.2.7	DROPLET_REL_POS_UPDATE_TIME	76
8.5.2.8	DS_ERROR	76
8.5.2.9	DS_FATAL	76
8.5.2.10	DS_SUCCESS	76
8.5.2.11	DS_WARNING	76
8.5.2.12	FLOOR_FRICTION	76
8.5.2.13	IMPULSE_SCALING	76
8.5.2.14	IR_BUFFER_SIZE	76
8.5.2.15	IR_MAX_DATA_SIZE	76
8.5.2.16	IR_MSG_HEADER	76
8.5.2.17	IR_RX_STATUS_BUSY	76
8.5.2.18	MOTOR_POS_SCALING	76
8.5.2.19	MOVE_OFF	77
8.5.2.20	MOVE_TIME	77
8.5.2.21	NEWEST_MSG_FIRST	77

8.5.2.22	NORTH	77
8.5.2.23	NORTH_EAST	77
8.5.2.24	NORTH_WEST	77
8.5.2.25	OBJECT_ANGULAR_DAMPING	77
8.5.2.26	OBJECT_LINEAR_DAMPING	77
8.5.2.27	OLDEST_MSG_FIRST	77
8.5.2.28	PHYSICS_GRAVITY	77
8.5.2.29	SAFE_DELETE	77
8.5.2.30	SOUTH	77
8.5.2.31	SOUTH_EAST	78
8.5.2.32	SOUTH_WEST	78
8.5.2.33	STAGGERED_START	78
8.5.2.34	STEP_TIME	78
8.5.2.35	TURN_CLOCKWISE	78
8.5.2.36	TURN_COUNTERCLOCKWISE	78
8.5.2.37	TURN_OFF	78
8.5.2.38	WALK_STEP_TIME	78
8.5.3	Typedef Documentation	78
8.5.3.1	droplet_id_type	78
8.5.3.2	DS_RESULT	78
8.5.3.3	move_direction	78
8.5.3.4	msg_order	79
8.5.3.5	turn_direction	79
8.6	DropletSimInfo.cpp File Reference	79
8.7	DropletSimInfo.h File Reference	79
8.7.1	Detailed Description	80
8.7.2	Typedef Documentation	80
8.7.2.1	DirInfo	80
8.8	DropletTimeControl.cpp File Reference	80
8.8.1	Function Documentation	81
8.8.1.1	tvToDouble	81
8.9	DropletTimeControl.h File Reference	81
8.9.1	Detailed Description	82
8.9.2	Macro Definition Documentation	83
8.9.2.1	_DROPLET_TIME_CONTROL	83
8.10	DropletUtil.cpp File Reference	83
8.11	DropletUtil.h File Reference	83
8.11.1	Macro Definition Documentation	84
8.11.1.1	_DROPLET_UTIL	85
8.12	DSimPhysicalObject.cpp File Reference	85

8.13	DSimPhysicalObject.h File Reference	85
8.13.1	Macro Definition Documentation	86
8.13.1.1	_DSIM_PHYSICAL_OBJECT	86
8.14	features.dox File Reference	86
8.15	IDroplet.cpp File Reference	87
8.16	IDroplet.h File Reference	87
8.16.1	Macro Definition Documentation	88
8.16.1.1	_I_DROPLET	88
8.17	IDropletProjector.cpp File Reference	88
8.18	IDropletProjector.h File Reference	89
8.18.1	Macro Definition Documentation	89
8.18.1.1	_DROPLET_PROJECTOR	90
8.19	IDropletSimInterface.cpp File Reference	90
8.20	IDropletSimInterface.h File Reference	90
8.20.1	Macro Definition Documentation	91
8.20.1.1	_I_DROPLET_SIM_INTERFACE	91
8.20.2	Enumeration Type Documentation	91
8.20.2.1	BasicObjectShapes	91
8.21	inttypes.h File Reference	92
8.21.1	Macro Definition Documentation	92
8.21.1.1	__INTTYPES_H_	92
8.21.2	Typedef Documentation	92
8.21.2.1	int16_t	92
8.21.2.2	int32_t	93
8.21.2.3	int64_t	93
8.21.2.4	int8_t	93
8.21.2.5	intptr_t	93
8.21.2.6	uint16_t	93
8.21.2.7	uint32_t	93
8.21.2.8	uint64_t	93
8.21.2.9	uint8_t	93
8.21.2.10	uintptr_t	93
8.22	main.dox File Reference	93

Chapter 1

DropletSimLibrary

1.1 Introduction

DropletSimLibrary is an example client for the Droplets simulation library.

1.2 Dependencies

- Visual Studio 2010 or higher
- Bullet 2.80+
- EasyBMP 1.06

1.3 Installation

Please consult the build guide for a more detailed of how to build DropletSimLibrary.

1.3.1 Step 1: Obtaining Source Code

Source code for this project can be attained from the [cu-droplet Google Code page](#). To download it you will need to have installed a [git client](#).

1.3.2 Step 2: Building DropletSimLibrary

Building DropletSimLibrary is fairly straight-forward. To do this, you simply perform the following:

1. Navigate to the project folder found in `DropletSimulator/DropletSimLibrary/vs2010/`
2. Open the Visual Studio solution file `DropletSimLibrary.sln`
3. Select one of the build configurations. See below for an explanation of each.
4. Under the Build menu, select Build Solution. On a clean checkout this will force it to build all dependent libraries and may take several minutes.

This builds a binary version of the library that you can link against in your own projections.

1.4 How to contribute

TODO: Add policies on contributing.

1.4.1 Issue Tracker

Current known issues with the Droplets project can be found at the [cu-droplet issues tracker on Google Code](#).

1.4.2 Contacting us.

TODO: Add primary contact information for the project.

Chapter 2

Doxygen Guide

This contains instructions for how to use Doxygen to build the documentation.

2.1 Dependencies

- [Doxygen](#)
- [GraphViz](#)
- LaTeX (or MiKTeX) to build PDF guides

2.2 Running Doxygen

Building documentation is straight forward. Assuming you have correctly installed Doxygen and GraphViz, you simply do the following:

1. Open DoxyWizard
2. Under the "File" menu, select "Open" and navigate to `DropletSimulator/DropletSimLibrary/docs/Doxyfile`
3. Select the "Run" tab
4. Click "Run Doxygen"

This will create two directories inside `DropletSimulator/DropletSimLibrary/docs/` - one named `html/` that contains a set of HTML documents that can be uploaded to a server and one titled `latex/` that contains a LaTeX-formatted manual.

2.3 Building a .PDF Manual

Once the above is done it is possible to generate a PDF version of the LaTeX manual by navigating to `DropletSimulator/DropletSimDemos/DropletGUI/docs/latex` and running `make` (under the Linux command line) or `make.bat` (under Windows). This will generate a PDF-formatted version of the manual in that folder named `refman.pdf` that can be safely copied elsewhere.

Chapter 3

Supported Features

3.1 Droplets

- Six directions of linear movement
- In-place rotation
- Communication with customizable range
- Range and bearing from communications
- True color illumination
- RGB color sensing
- Self-righting

3.2 Simulator

- Realistic physics simulation using the Bullet physics library
- Cross-compiling programs between simulator and hardware
- Load projection images onto the arena
- Supports custom arenas
- Add spheres and cubes into arena
- Run heterogeneous programs on Droplets
- Add Droplets during simulation
- Track leg power status of Droplets
- Set of demo programs and blank program templates for the user
- Console version which can be compiled on UNIX systems

Chapter 4

Deprecated List

Member [DropletSim::CreateFloor](#) (int floorShapeIndex, int wallXShapeIndex=-1, int wallYShapeIndex=-1)
Creates a floor.

Chapter 5

Class Index

5.1 Class List

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

_Droplet_Actuator_Data	13
_Droplet_Communication_Data	14
_Droplet_Component_Data	15
_Droplet_Localization_Data	16
_Droplet_Sensor_Data	17
_Droplet_Sim_Comm_Channel_Data	18
_Droplet_Timing_Data	19
_Object_Physics_Data	19
_Simulator_Physics_Data	20
Droplet_Motion_Direction_Data	22
DropletSim	22
DropletSimInfo	
Helper class used for retrieving information from the simulator. Retrieves any interesting info stored in DropletDataStructs	29
DropletTimeControl	
Class used for calculating timing information for the Droplet Simulator. In order to access timing information from outside the simulator, use DropletSimInfo	36
DSimPhysicalObject	44
IDroplet	46
IDropletProjector	57
IDropletSimInterface	62
Ran	63
SimSetupData	65
TrigArray	66

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

DropletDataStructs.h	
This file contains private data structures to be used by the simulator only! Public data structures used for returning information to an external caller are defined in DropletSimInfo.h	69
DropletSim.cpp	72
DropletSim.h	72
DropletSimGlobals.h	
This file contains global variables	73
DropletSimInfo.cpp	79
DropletSimInfo.h	
Declares the droplet simulation information class that gives access to info stored by the structs declared in DropletDataStructs and timing info calculated by DropletTimeControl . Detailed info on the data structs can be found in: cu-droplet\DropletSimulator\DropletSimLibrary\include\- DropletDataStructs.h Details on Droplet Simulator return codes (DS_RESULT) can be found in: cu-droplet\DropletSimulator\DropletSimLibrary\include\DropletSimGlobals.h	79
DropletTimeControl.cpp	80
DropletTimeControl.h	
Declares the droplet time control class that gathers timing info. Times are returned as doubles in seconds. The UNIX compilation calculates time with microsecond accuracy and the windows compilation calculates time with millisecond accuracy but both are only displayed to millisecond accuracy	81
DropletUtil.cpp	83
DropletUtil.h	83
DSimPhysicalObject.cpp	85
DSimPhysicalObject.h	85
IDroplet.cpp	87
IDroplet.h	87
IDropletProjector.cpp	88
IDropletProjector.h	89
IDropletSimInterface.cpp	90
IDropletSimInterface.h	90
inttypes.h	92

Chapter 7

Class Documentation

7.1 `_Droplet_Actuator_Data` Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- `bool _oscillator`
- `uint8_t bOut`
- `move_direction currMoveDir`
- `turn_direction currTurnDir`
- `uint8_t gOut`
- `float moveStepRemaining`
- `float moveTimeRemaining`
- `float rotateStepRemaining`
- `float rotateTimeRemaining`
- `uint8_t rOut`

7.1.1 Detailed Description

Defines an alias representing information describing the droplet actuator.

Definition at line 74 of file `DropletDataStructs.h`.

7.1.2 Member Data Documentation

7.1.2.1 `bool _Droplet_Actuator_Data::_oscillator`

Definition at line 79 of file `DropletDataStructs.h`.

7.1.2.2 `uint8_t _Droplet_Actuator_Data::bOut`

Definition at line 76 of file `DropletDataStructs.h`.

7.1.2.3 `move_direction _Droplet_Actuator_Data::currMoveDir`

Definition at line 80 of file `DropletDataStructs.h`.

7.1.2.4 `turn_direction` `_Droplet_Actuator_Data::currTurnDir`

Definition at line 81 of file `DropletDataStructs.h`.

7.1.2.5 `uint8_t` `_Droplet_Actuator_Data::gOut`

Definition at line 76 of file `DropletDataStructs.h`.

7.1.2.6 `float` `_Droplet_Actuator_Data::moveStepRemaining`

Definition at line 78 of file `DropletDataStructs.h`.

7.1.2.7 `float` `_Droplet_Actuator_Data::moveTimeRemaining`

Definition at line 77 of file `DropletDataStructs.h`.

7.1.2.8 `float` `_Droplet_Actuator_Data::rotateStepRemaining`

Definition at line 78 of file `DropletDataStructs.h`.

7.1.2.9 `float` `_Droplet_Actuator_Data::rotateTimeRemaining`

Definition at line 77 of file `DropletDataStructs.h`.

7.1.2.10 `uint8_t` `_Droplet_Actuator_Data::rOut`

Definition at line 76 of file `DropletDataStructs.h`.

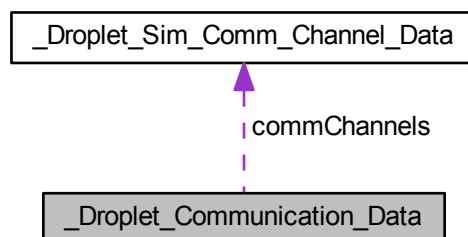
The documentation for this struct was generated from the following file:

- [DropletDataStructs.h](#)

7.2 `_Droplet_Communication_Data` Struct Reference

```
#include <DropletDataStructs.h>
```

Collaboration diagram for `_Droplet_Communication_Data`:



Public Attributes

- [DropletCommChannelData commChannels](#) [6]
- bool [sendActive](#)

7.2.1 Detailed Description

Defines an alias representing information describing the droplet communication.

Definition at line 110 of file DropletDataStructs.h.

7.2.2 Member Data Documentation

7.2.2.1 DropletCommChannelData _Droplet_Communication_Data::commChannels[6]

Definition at line 113 of file DropletDataStructs.h.

7.2.2.2 bool _Droplet_Communication_Data::sendActive

Definition at line 112 of file DropletDataStructs.h.

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

- [DropletDataStructs.h](#)

7.3 _Droplet_Component_Data Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- [uint8_t capacitorPower](#)
- [droplet_id_type dropletID](#)
- [int8_t leg1Power](#)
- [int8_t leg2Power](#)
- [int8_t leg3Power](#)

7.3.1 Detailed Description

Defines an alias representing information describing the droplet component.

Definition at line 121 of file DropletDataStructs.h.

7.3.2 Member Data Documentation

7.3.2.1 uint8_t _Droplet_Component_Data::capacitorPower

Definition at line 124 of file DropletDataStructs.h.

7.3.2.2 droplet_id_type _Droplet_Component_Data::dropletID

Definition at line 125 of file DropletDataStructs.h.

7.3.2.3 `int8_t _Droplet_Component_Data::leg1Power`

Definition at line 123 of file `DropletDataStructs.h`.

7.3.2.4 `int8_t _Droplet_Component_Data::leg2Power`

Definition at line 123 of file `DropletDataStructs.h`.

7.3.2.5 `int8_t _Droplet_Component_Data::leg3Power`

Definition at line 123 of file `DropletDataStructs.h`.

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

- [DropletDataStructs.h](#)

7.4 `_Droplet_Localization_Data` Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- double [lastRelPosUpdate](#)
- bool [movedSinceLastUpdate](#)
- float [posX](#)
- float [posY](#)
- float [posZ](#)
- float [rotA](#)
- float [rotX](#)
- float [rotY](#)
- float [rotZ](#)

7.4.1 Detailed Description

Stores localization information for droplets.

Definition at line 61 of file `DropletDataStructs.h`.

7.4.2 Member Data Documentation

7.4.2.1 `double _Droplet_Localization_Data::lastRelPosUpdate`

Definition at line 66 of file `DropletDataStructs.h`.

7.4.2.2 `bool _Droplet_Localization_Data::movedSinceLastUpdate`

Definition at line 65 of file `DropletDataStructs.h`.

7.4.2.3 `float _Droplet_Localization_Data::posX`

Definition at line 64 of file `DropletDataStructs.h`.

7.4.2.4 `float _Droplet_Localization_Data::posY`

Definition at line 64 of file `DropletDataStructs.h`.

7.4.2.5 `float _Droplet_Localization_Data::posZ`

Definition at line 64 of file `DropletDataStructs.h`.

7.4.2.6 `float _Droplet_Localization_Data::rotA`

Definition at line 63 of file `DropletDataStructs.h`.

7.4.2.7 `float _Droplet_Localization_Data::rotX`

Definition at line 63 of file `DropletDataStructs.h`.

7.4.2.8 `float _Droplet_Localization_Data::rotY`

Definition at line 63 of file `DropletDataStructs.h`.

7.4.2.9 `float _Droplet_Localization_Data::rotZ`

Definition at line 63 of file `DropletDataStructs.h`.

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

- [DropletDataStructs.h](#)

7.5 `_Droplet_Sensor_Data` Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- [uint8_t bln](#)
- [uint8_t gln](#)
- [uint8_t rln](#)

7.5.1 Detailed Description

Defines an alias representing information describing the droplet sensor.

Definition at line 89 of file `DropletDataStructs.h`.

7.5.2 Member Data Documentation

7.5.2.1 `uint8_t _Droplet_Sensor_Data::bln`

Definition at line 91 of file `DropletDataStructs.h`.

7.5.2.2 uint8_t _Droplet_Sensor_Data::gln

Definition at line 91 of file DropletDataStructs.h.

7.5.2.3 uint8_t _Droplet_Sensor_Data::rln

Definition at line 91 of file DropletDataStructs.h.

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

- [DropletDataStructs.h](#)

7.6 _Droplet_Sim_Comm_Channel_Data Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- unsigned char [inBuf](#) [[IR_BUFFER_SIZE](#)]
- [uint8_t](#) [inMsgLength](#)
- [uint16_t](#) [lastMsgInTimestamp](#)
- [uint16_t](#) [lastMsgOutTimestamp](#)
- unsigned char [outBuf](#) [[IR_BUFFER_SIZE](#)]
- [uint8_t](#) [outMsgLength](#)

7.6.1 Detailed Description

Defines an internally used struct to store message information per channel.

Definition at line 99 of file DropletDataStructs.h.

7.6.2 Member Data Documentation

7.6.2.1 unsigned char _Droplet_Sim_Comm_Channel_Data::inBuf[IR_BUFFER_SIZE]

Definition at line 101 of file DropletDataStructs.h.

7.6.2.2 uint8_t _Droplet_Sim_Comm_Channel_Data::inMsgLength

Definition at line 104 of file DropletDataStructs.h.

7.6.2.3 uint16_t _Droplet_Sim_Comm_Channel_Data::lastMsgInTimestamp

Definition at line 103 of file DropletDataStructs.h.

7.6.2.4 uint16_t _Droplet_Sim_Comm_Channel_Data::lastMsgOutTimestamp

Definition at line 103 of file DropletDataStructs.h.

7.6.2.5 unsigned char _Droplet_Sim_Comm_Channel_Data::outBuf[IR_BUFFER_SIZE]

Definition at line 102 of file DropletDataStructs.h.

7.6.2.6 uint8_t _Droplet_Sim_Comm_Channel_Data::outMsgLength

Definition at line 104 of file DropletDataStructs.h.

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

- [DropletDataStructs.h](#)

7.7 _Droplet_Timing_Data Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- float [timer](#) [DROPLET_NUM_TIMERS]
- uint8_t [trigger](#) [DROPLET_NUM_TIMERS]

7.7.1 Detailed Description

Definition at line 129 of file DropletDataStructs.h.

7.7.2 Member Data Documentation

7.7.2.1 float _Droplet_Timing_Data::timer[DROPLET_NUM_TIMERS]

Definition at line 131 of file DropletDataStructs.h.

7.7.2.2 uint8_t _Droplet_Timing_Data::trigger[DROPLET_NUM_TIMERS]

Definition at line 132 of file DropletDataStructs.h.

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

- [DropletDataStructs.h](#)

7.8 _Object_Physics_Data Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- unsigned int [_worldID](#)
- int [colShapeIndex](#)
- btScalar [friction](#)
- btVector3 [localInertia](#)
- btScalar [mass](#)

7.8.1 Detailed Description

Defines an alias representing information describing the object physics.

Definition at line 48 of file DropletDataStructs.h.

7.8.2 Member Data Documentation

7.8.2.1 unsigned int _Object_Physics_Data::_worldID

Definition at line 53 of file DropletDataStructs.h.

7.8.2.2 int _Object_Physics_Data::colShapeIndex

Definition at line 52 of file DropletDataStructs.h.

7.8.2.3 btScalar _Object_Physics_Data::friction

Definition at line 50 of file DropletDataStructs.h.

7.8.2.4 btVector3 _Object_Physics_Data::localInertia

Definition at line 51 of file DropletDataStructs.h.

7.8.2.5 btScalar _Object_Physics_Data::mass

Definition at line 50 of file DropletDataStructs.h.

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

- [DropletDataStructs.h](#)

7.9 _Simulator_Physics_Data Struct Reference

```
#include <DropletDataStructs.h>
```

Public Attributes

- int [_colShapeIDCounter](#)
- int [_physicsWorldObjCounter](#)
- btBroadphaseInterface * [broadphase](#)
- btDefaultCollisionConfiguration * [collisionConfig](#)
- btCollisionDispatcher * [collisionDispatch](#)
- btAlignedObjectArray
 < btCollisionShape * > * [collisionShapes](#)
- btConstraintSolver * [constraintSolver](#)
- btDiscreteDynamicsWorld * [dynWorld](#)

Static Public Attributes

- static const int [_dynObjCollisionBM](#) = 2
- static const int [_staticObjCollisionBM](#) = 1

7.9.1 Detailed Description

Defines an alias representing information describing the simulator physics.

Definition at line 27 of file DropletDataStructs.h.

7.9.2 Member Data Documentation

7.9.2.1 `int _Simulator_Physics_Data::_colShapelDCounter`

Definition at line 40 of file DropletDataStructs.h.

7.9.2.2 `const int _Simulator_Physics_Data::_dynObjCollisionBM = 2` `[static]`

Definition at line 37 of file DropletDataStructs.h.

7.9.2.3 `int _Simulator_Physics_Data::_physicsWorldObjCounter`

Definition at line 40 of file DropletDataStructs.h.

7.9.2.4 `const int _Simulator_Physics_Data::_staticObjCollisionBM = 1` `[static]`

Definition at line 36 of file DropletDataStructs.h.

7.9.2.5 `btBroadphaseInterface* _Simulator_Physics_Data::broadphase`

Definition at line 32 of file DropletDataStructs.h.

7.9.2.6 `btDefaultCollisionConfiguration* _Simulator_Physics_Data::collisionConfig`

Definition at line 30 of file DropletDataStructs.h.

7.9.2.7 `btCollisionDispatcher* _Simulator_Physics_Data::collisionDispatch`

Definition at line 31 of file DropletDataStructs.h.

7.9.2.8 `btAlignedObjectArray<btCollisionShape*>* _Simulator_Physics_Data::collisionShapes`

Definition at line 39 of file DropletDataStructs.h.

7.9.2.9 `btConstraintSolver* _Simulator_Physics_Data::constraintSolver`

Definition at line 33 of file DropletDataStructs.h.

7.9.2.10 `btDiscreteDynamicsWorld* _Simulator_Physics_Data::dynWorld`

Definition at line 34 of file DropletDataStructs.h.

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

- [DropletDataStructs.h](#)

7.10 Droplet_Motion_Direction_Data Struct Reference

```
#include <DropletSimInfo.h>
```

Public Attributes

- [move_direction](#) currMoveDir
- [turn_direction](#) currTurnDir

7.10.1 Detailed Description

Definition at line 31 of file DropletSimInfo.h.

7.10.2 Member Data Documentation

7.10.2.1 move_direction Droplet_Motion_Direction_Data::currMoveDir

Definition at line 33 of file DropletSimInfo.h.

7.10.2.2 turn_direction Droplet_Motion_Direction_Data::currTurnDir

Definition at line 34 of file DropletSimInfo.h.

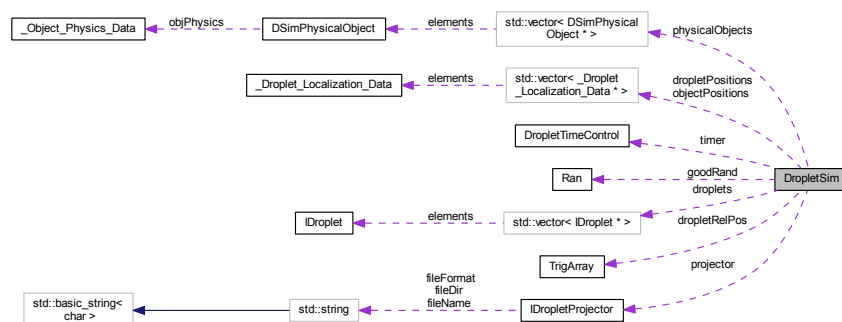
The documentation for this struct was generated from the following file:

- [DropletSimInfo.h](#)

7.11 DropletSim Class Reference

```
#include <DropletSim.h>
```

Collaboration diagram for DropletSim:



Public Member Functions

- [DropletSim](#) (void)
- [~DropletSim](#) ()
- [DS_RESULT AddCollisionShape](#) (btCollisionShape *colShape, int *colShapeIndex)

- [DS_RESULT AddDroplet](#) ([IDroplet](#) *pDroplet, [std::pair](#)< float, float > startPos, float startAngle)
- [DS_RESULT AddPhysicalObject](#) ([DSimPhysicalObject](#) *pObject, [std::pair](#)< float, float > startPos, float startAngle)
- [DS_RESULT AddPhysicalObject](#) ([DSimPhysicalObject](#) *pObject, [std::pair](#)< float, float > startPos, float startHeight, float startAngle)
- [DS_RESULT Cleanup](#) (void)
- [DS_RESULT CreateFloor](#) (int floorShapeIndex, int wallXShapeIndex=-1, int wallYShapeIndex=-1)
- [DS_RESULT Init](#) (const [SimSetupData](#) &setupData)
- [DS_RESULT SetUpProjector](#) ([std::string](#) imgDir, [std::string](#) imgName)
- [DS_RESULT SetUpProjector](#) ([std::string](#) imgDir, [std::string](#) imgName, int projWidth, int projLength)
- [DS_RESULT Step](#) (void)

Protected Attributes

- [std::vector](#)< [GPSInfo](#) * > [dropletPositions](#)
- [TrigArray](#) * [dropletRelPos](#)
- [std::vector](#)< [IDroplet](#) * > [droplets](#)
- bool [firstRun](#)
- [Ran](#) * [goodRand](#)
- [std::vector](#)< [GPSInfo](#) * > [objectPositions](#)
- [std::vector](#)< [DSimPhysicalObject](#) * > [physicalObjects](#)
- [IDropletProjector](#) * [projector](#)
- bool [projSet](#)
- [DropletTimeControl](#) [timer](#)

Friends

- class [DropletSimInfo](#)

7.11.1 Detailed Description

Droplet simulator.

Definition at line 28 of file [DropletSim.h](#).

7.11.2 Constructor & Destructor Documentation

7.11.2.1 [DropletSim::DropletSim \(void \)](#)

Default constructor.

Definition at line 3 of file [DropletSim.cpp](#).

7.11.2.2 [DropletSim::~DropletSim \(\)](#)

Destructor.

Definition at line 17 of file [DropletSim.cpp](#).

7.11.3 Member Function Documentation

7.11.3.1 DS_RESULT DropletSim::AddCollisionShape (btCollisionShape * *colShape*, int * *colShapeIndex*)

Adds a collision shape to 'colShapeIndex'.

Parameters

in, out	<i>colShape</i>	If non-null, the col shape.
in, out	<i>colShapeIndex</i>	If non-null, zero-based index of the col shape.

Returns

.

Definition at line 24 of file DropletSim.cpp.

7.11.3.2 DS_RESULT DropletSim::AddDroplet (IDroplet * *pDroplet*, std::pair< float, float > *startPos*, float *startAngle*)

Adds a droplet.

Parameters

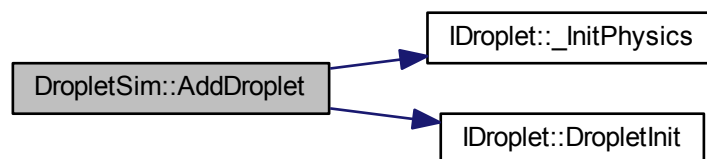
in, out	<i>pDroplet</i>	If non-null, the droplet.
	<i>startPos</i>	The starting (x, y) pos of the droplet on the arena.
	<i>startAngle</i>	The start angle.

Returns

.

Definition at line 79 of file DropletSim.cpp.

Here is the call graph for this function:



7.11.3.3 DS_RESULT DropletSim::AddPhysicalObject (DSimPhysicalObject * *pObject*, std::pair< float, float > *startPos*, float *startAngle*)

Adds a physical object.

Parameters

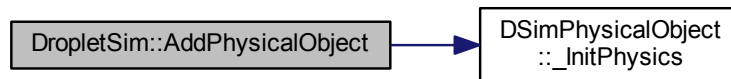
in, out	<i>pObject</i>	If non-null, the object.
	<i>startPos</i>	The starting (x, y) pos of the object on the arena.
	<i>startAngle</i>	The start angle.

Returns

.

Definition at line 108 of file DropletSim.cpp.

Here is the call graph for this function:



7.11.3.4 DS_RESULT DropletSim::AddPhysicalObject (DSimPhysicalObject * *pObject*, std::pair< float, float > *startPos*, float *startHeight*, float *startAngle*)

Adds a physical object.

Parameters

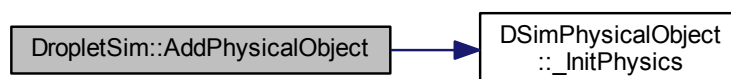
<i>in, out</i>	<i>pObject</i>	If non-null, the object.
	<i>startPos</i>	The starting (x, y) pos of the object on the arena.
	<i>startHeight</i>	The starting z pos of the object on the arena.
	<i>startAngle</i>	The start angle.

Returns

.

Definition at line 140 of file DropletSim.cpp.

Here is the call graph for this function:



7.11.3.5 DS_RESULT DropletSim::Cleanup (void)

Cleans up and frees up memory used by the physics engine.

Returns

.

Definition at line 330 of file DropletSim.cpp.

7.11.3.6 DS_RESULT DropletSim::CreateFloor (int *floorShapeIndex*, int *wallXShapeIndex* = -1, int *wallYShapeIndex* = -1)

Deprecated Creates a floor.

Parameters

<i>floorShapeIndex</i>	Zero-based index of the floor shape.
<i>wallXShapeIndex</i>	(optional) zero-based index of the wall x coordinate shape.
<i>wallYShapeIndex</i>	(optional) zero-based index of the wall y coordinate shape.

Returns

The new floor.

Definition at line 32 of file DropletSim.cpp.

7.11.3.7 DS_RESULT DropletSim::Init (const SimSetupData & *setupData*)

Initialises this object.

Parameters

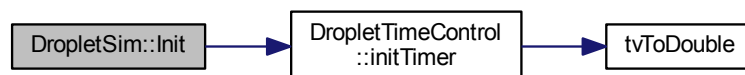
<i>setupData</i>	Information describing the setup.
------------------	-----------------------------------

Returns

.

Definition at line 223 of file DropletSim.cpp.

Here is the call graph for this function:



7.11.3.8 DS_RESULT DropletSim::SetUpProjector (std::string *imgDir*, std::string *imgName*)

Sets up the projector.

Parameters

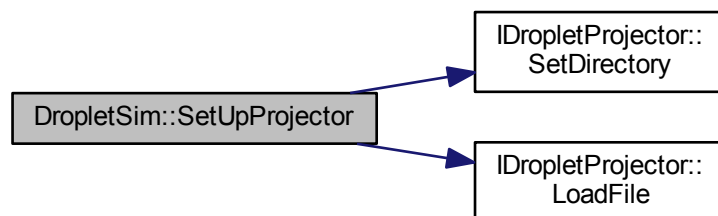
<i>imgDir</i>	The image dir.
<i>imgName</i>	Name of the image.

Returns

.

Definition at line 173 of file DropletSim.cpp.

Here is the call graph for this function:



7.11.3.9 DS_RESULT DropletSim::SetUpProjector (std::string *imgDir*, std::string *imgName*, int *projWidth*, int *projLength*)

Sets up the projector.

Parameters

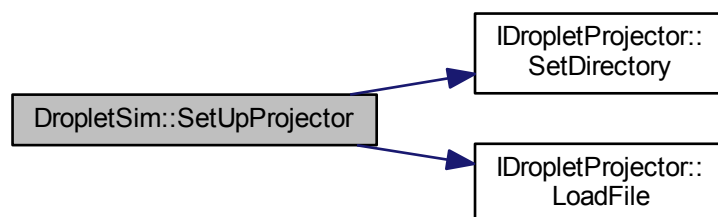
<i>imgDir</i>	The image dir.
<i>imgName</i>	Name of the image.
<i>projWidth</i>	Width of the project.
<i>projLength</i>	Length of the project.

Returns

.

Definition at line 197 of file DropletSim.cpp.

Here is the call graph for this function:



7.11.3.10 DS_RESULT DropletSim::Step (void)

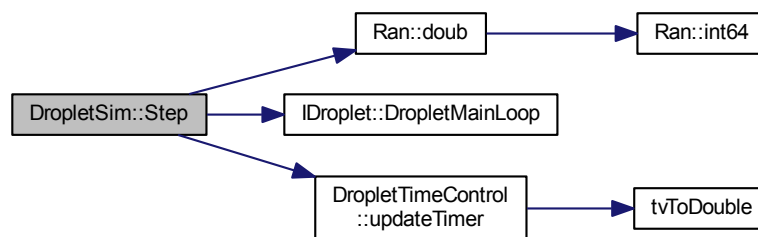
Executes a step in the simulation.

Returns

.

Definition at line 264 of file DropletSim.cpp.

Here is the call graph for this function:



7.11.4 Friends And Related Function Documentation

7.11.4.1 friend class DropletSimInfo [friend]

Definition at line 30 of file DropletSim.h.

7.11.5 Member Data Documentation

7.11.5.1 std::vector<GPSInfo *> DropletSim::dropletPositions [protected]

Vector containing droplet positions.

Definition at line 134 of file DropletSim.h.

7.11.5.2 TrigArray* DropletSim::dropletRelPos [protected]

Definition at line 135 of file DropletSim.h.

7.11.5.3 std::vector<IDroplet *> DropletSim::droplets [protected]

Vector containing droplet information.

Definition at line 122 of file DropletSim.h.

7.11.5.4 bool DropletSim::firstRun [protected]

Definition at line 109 of file DropletSim.h.

7.11.5.5 `Ran* DropletSim::goodRand` `[protected]`

Definition at line 110 of file DropletSim.h.

7.11.5.6 `std::vector<GPSInfo *> DropletSim::objectPositions` `[protected]`

Vector containing physical object positions (other than droplets).

Definition at line 141 of file DropletSim.h.

7.11.5.7 `std::vector<DSimPhysicalObject *> DropletSim::physicalObjects` `[protected]`

Vector containing physical object information (other than droplets).

Definition at line 128 of file DropletSim.h.

7.11.5.8 `IDropletProjector* DropletSim::projector` `[protected]`

The projector.

Definition at line 116 of file DropletSim.h.

7.11.5.9 `bool DropletSim::projSet` `[protected]`

Definition at line 109 of file DropletSim.h.

7.11.5.10 `DropletTimeControl DropletSim::timer` `[protected]`

Definition at line 144 of file DropletSim.h.

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

- [DropletSim.h](#)
- [DropletSim.cpp](#)

7.12 DropletSimInfo Class Reference

Helper class used for retrieving information from the simulator. Retrieves any interesting info stored in DropletDataStructs.

```
#include <DropletSimInfo.h>
```

Public Member Functions

- [DS_RESULT GetActuationData](#) (std::vector< [DropletActuatorData](#) * > *act, [DropletSim](#) &simulator)
puts droplet actuator info into a supplied vector.
- [DS_RESULT GetCommData](#) (std::vector< [DropletCommData](#) * > *comm, [DropletSim](#) &simulator)
puts droplet communication info into a supplied vector.
- [DS_RESULT GetCompData](#) (std::vector< [DropletCompData](#) * > *comp, [DropletSim](#) &simulator)
puts droplet component info into a supplied vector.
- [DS_RESULT GetDropletColors](#) (std::vector< [uint8_t](#) * > *colors, [DropletSim](#) &simulator)
puts droplet LED color info into a supplied vector. color data is RGB ranging from 0 to 255
- [DS_RESULT GetDropletPositions](#) (std::vector< [GPSInfo](#) * > *outPosData, [DropletSim](#) &simulator)

- puts droplet position info into a supplied vector.*
- [DS_RESULT GetMotionDirections](#) (std::vector< [DirInfo](#) * > *directions, [DropletSim](#) &simulator)
puts droplet motion (move and rotate) direction info into a supplied vector.
- [DS_RESULT GetObjectPositions](#) (std::vector< [GPSInfo](#) * > *outPosData, [DropletSim](#) &simulator)
puts object position info into a supplied vector.
- [DS_RESULT GetPhysData](#) (std::vector< [ObjectPhysicsData](#) * > *phys, [DropletSim](#) &simulator)
puts object physics info into a supplied vector.
- [DS_RESULT GetRemainingMotionTimes](#) (std::vector< float * > *times, [DropletSim](#) &simulator)
puts remaining droplet motion (move and rotate) time info into a supplied vector.
- [DS_RESULT GetSensorColors](#) (std::vector< [uint8_t](#) * > *colors, [DropletSim](#) &simulator)
puts droplet RGB sensor color info into a supplied vector. colors range from 0 to 255.
- double [GetStepRT](#) ([DropletSim](#) &simulator)
gets the real time elapsed since the simulator calculated the last step from the [DropletTimeControl](#) class.
- double [GetTimeRatio](#) ([DropletSim](#) &simulator)
gets the ratio of time simulated in one step to real time elapsed since the last step from the [DropletTimeControl](#) class.
- double [GetTotalDiff](#) ([DropletSim](#) &simulator)
gets the difference between the total real time and simulator time elapsed from the [DropletTimeControl](#) class.
- double [GetTotalRT](#) ([DropletSim](#) &simulator)
gets the total real time elapsed since the simulator has started from the [DropletTimeControl](#) class.
- double [GetTotalST](#) ([DropletSim](#) &simulator)
gets the total time simulated since the simulator has started from the [DropletTimeControl](#) class.

Friends

- class [DropletSim](#)

7.12.1 Detailed Description

Helper class used for retrieving information from the simulator. Retrieves any interesting info stored in DropletData-Structs.

Definition at line 46 of file DropletSimInfo.h.

7.12.2 Member Function Documentation

7.12.2.1 DS_RESULT DropletSimInfo::GetActuationData (std::vector< [DropletActuatorData](#) * > * act, [DropletSim](#) & simulator)

puts droplet actuator info into a supplied vector.

Parameters

in, out	<i>act</i>	If non-null, information describing actuator data.
in, out	<i>simulator</i>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 240 of file DropletSimInfo.cpp.

7.12.2.2 DS_RESULT DropletSimInfo::GetCommData (std::vector< DropletCommData * > * *comm*, DropletSim & *simulator*)

puts droplet communication info into a supplied vector.

Parameters

<i>in, out</i>	<i>comm</i>	If non-null, information describing the communications.
<i>in, out</i>	<i>simulator</i>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 177 of file DropletSimInfo.cpp.

7.12.2.3 DS_RESULT DropletSimInfo::GetCompData (std::vector< DropletCompData * > * *comp*, DropletSim & *simulator*)

puts droplet component info into a supplied vector.

Parameters

<i>in, out</i>	<i>comp</i>	If non-null, information describing the component data.
<i>in, out</i>	<i>simulator</i>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 214 of file DropletSimInfo.cpp.

7.12.2.4 DS_RESULT DropletSimInfo::GetDropletColors (std::vector< uint8_t * > * *colors*, DropletSim & *simulator*)

puts droplet LED color info into a supplied vector. color data is RGB ranging from 0 to 255

Parameters

<i>in, out</i>	<i>colors</i>	If non-null, information describing the droplet colors.
<i>in, out</i>	<i>simulator</i>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 57 of file DropletSimInfo.cpp.

7.12.2.5 DS_RESULT DropletSimInfo::GetDropletPositions (std::vector< GPSInfo * > * *outPosData*, DropletSim & *simulator*)

puts droplet position info into a supplied vector.

Parameters

<i>in, out</i>	<i>outPosData</i>	If non-null, information describing the droplet positions.
<i>in, out</i>	<i>simulator</i>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 5 of file DropletSimInfo.cpp.

7.12.2.6 **DS_RESULT** DropletSimInfo::GetMotionDirections (`std::vector< DirInfo * > * directions`, `DropletSim & simulator`)

puts droplet motion (move and rotate) direction info into a supplied vector.

Parameters

<code>in, out</code>	<code>directions</code>	If non-null, information describing the move directions.
<code>in, out</code>	<code>simulator</code>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 104 of file DropletSimInfo.cpp.

7.12.2.7 **DS_RESULT** DropletSimInfo::GetObjectPositions (`std::vector< GPSInfo * > * outPosData`, `DropletSim & simulator`)

puts object position info into a supplied vector.

Parameters

<code>in, out</code>	<code>outPosData</code>	If non-null, information describing the object positions.
<code>in, out</code>	<code>simulator</code>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 31 of file DropletSimInfo.cpp.

7.12.2.8 **DS_RESULT** DropletSimInfo::GetPhysData (`std::vector< ObjectPhysicsData * > * phys`, `DropletSim & simulator`)

puts object physics info into a supplied vector.

Parameters

<code>in, out</code>	<code>phys</code>	If non-null, information describing the physics data.
<code>in, out</code>	<code>simulator</code>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 151 of file DropletSimInfo.cpp.

7.12.2.9 DS_RESULT DropletSimInfo::GetRemainingMotionTimes (std::vector< float * > * *times*, DropletSim & *simulator*)

puts remaining droplet motion (move and rotate) time info into a supplied vector.

Parameters

in, out	<i>times</i>	If non-null, information describing the remaining times.
in, out	<i>simulator</i>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 81 of file DropletSimInfo.cpp.

7.12.2.10 DS_RESULT DropletSimInfo::GetSensorColors (std::vector< uint8_t * > * *colors*, DropletSim & *simulator*)

puts droplet RGB sensor color info into a supplied vector. colors range from 0 to 255.

Parameters

in, out	<i>colors</i>	If non-null, information describing the colors.
in, out	<i>simulator</i>	The simulator.

Returns

A Droplet Simulator error code.

Definition at line 127 of file DropletSimInfo.cpp.

7.12.2.11 double DropletSimInfo::GetStepRT (DropletSim & *simulator*)

gets the real time elapsed since the simulator calculated the last step from the [DropletTimeControl](#) class.

Parameters

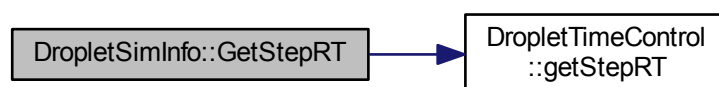
in, out	<i>simulator</i>	The simulator.
---------	------------------	----------------

Returns

The real time since the last step.

Definition at line 276 of file DropletSimInfo.cpp.

Here is the call graph for this function:



7.12.2.12 double DropletSimInfo::GetTimeRatio (DropletSim & *simulator*)

gets the ratio of time simulated in one step to real time elapsed since the last step from the [DropletTimeControl](#) class.

Parameters

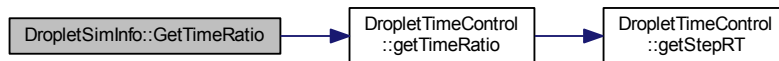
<i>in, out</i>	<i>simulator</i>	The simulator.
----------------	------------------	----------------

Returns

The total real time elapsed.

Definition at line 284 of file DropletSimInfo.cpp.

Here is the call graph for this function:



7.12.2.13 double DropletSimInfo::GetTotalDiff (DropletSim & *simulator*)

gets the difference between the total real time and simulator time elapsed from the [DropletTimeControl](#) class.

Parameters

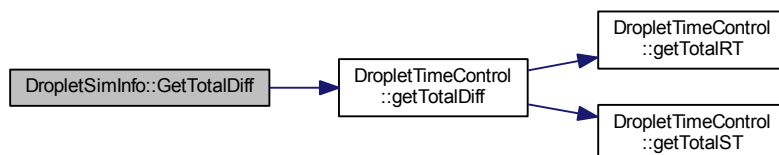
<i>in, out</i>	<i>simulator</i>	The simulator.
----------------	------------------	----------------

Returns

The difference between total real time and simulator time.

Definition at line 280 of file DropletSimInfo.cpp.

Here is the call graph for this function:



7.12.2.14 double DropletSimInfo::GetTotalRT (DropletSim & *simulator*)

gets the total real time elapsed since the simulator has started from the [DropletTimeControl](#) class.

Parameters

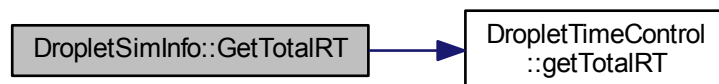
<i>in, out</i>	<i>simulator</i>	The simulator.
----------------	------------------	----------------

Returns

The total real time elapsed.

Definition at line 268 of file DropletSimInfo.cpp.

Here is the call graph for this function:



7.12.2.15 double DropletSimInfo::GetTotalST (DropletSim & *simulator*)

gets the total time simulated since the simulator has started from the [DropletTimeControl](#) class.

Parameters

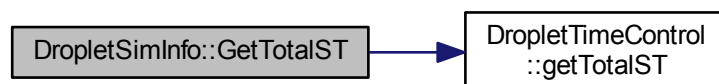
<i>in, out</i>	<i>simulator</i>	The simulator.
----------------	------------------	----------------

Returns

The total time simulated.

Definition at line 272 of file DropletSimInfo.cpp.

Here is the call graph for this function:



7.12.3 Friends And Related Function Documentation

7.12.3.1 friend class DropletSim [friend]

Definition at line 48 of file DropletSimInfo.h.

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

- [DropletSimInfo.h](#)
- [DropletSimInfo.cpp](#)

7.13 DropletTimeControl Class Reference

class used for calculating timing information for the Droplet Simulator. In order to access timing information from outside the simulator, use [DropletSimInfo](#).

```
#include <DropletTimeControl.h>
```

Public Member Functions

- double [getStepRT](#) ()
gets the real time elapsed since the last step.
- double [getTimeRatio](#) ()
gets the ratio between the time simulated in one step and real time elapsed since the last step.
- double [getTotalDiff](#) ()
gets the difference between total real time and sim time.
- double [getTotalIRT](#) ()
get total real time elapsed since the simulator started.
- double [getTotalST](#) ()
get total time simulated so far.
- void [initTimer](#) (double sss)
initializes the timer with a specified sim step size.
- void [printAll](#) ()
prints all calculated time values on a single line to the console.
- void [printStepRT](#) ()
prints real time elapsed since the last step to the console.
- void [printTimeRatio](#) ()
prints the ratio between the sim time and real tim of the last step to the console.
- void [printTotalDiff](#) ()
prints the difference between total real time and simulated time to the console.
- void [printTotalIRT](#) ()
prints total real time to the console.
- void [printTotalST](#) ()
prints total simulator time to the console.
- void [printVars](#) ()
prints the stored timestamps: init time, current time, and time of the last step.
- void [resetTimer](#) ()
resets the timer.
- void [updateTimer](#) (double sss)
updates relevant variables and should be called each step. can be used to update the sim step size.
- void [updateTimer](#) ()

7.13.1 Detailed Description

class used for calculating timing information for the Droplet Simulator. In order to access timing information from outside the simulator, use [DropletSimInfo](#).

Class that controls droplet time properties.

Definition at line 48 of file DropletTimeControl.h.

7.13.2 Member Function Documentation

7.13.2.1 double DropletTimeControl::getStepRT ()

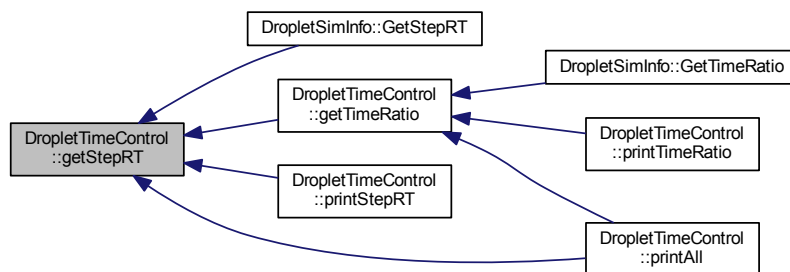
gets the real time elapsed since the last step.

Returns

The real time elapsed since the last step.

Definition at line 89 of file DropletTimeControl.cpp.

Here is the caller graph for this function:



7.13.2.2 double DropletTimeControl::getTimeRatio ()

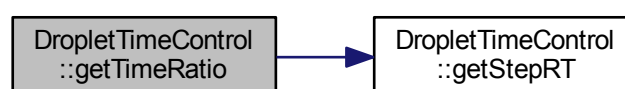
gets the ratio between the time simulated in one step and real time elapsed since the last step.

Returns

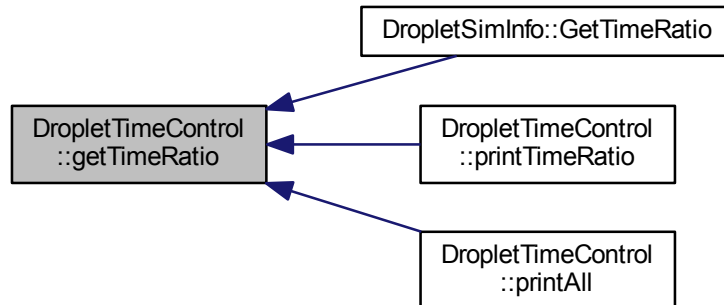
The last step's time ratio.

Definition at line 99 of file DropletTimeControl.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



7.13.2.3 double DropletTimeControl::getTotalDiff ()

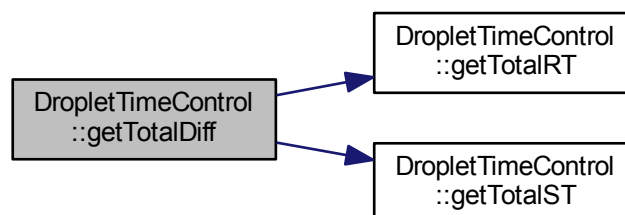
gets the difference between total real time and sim time.

Returns

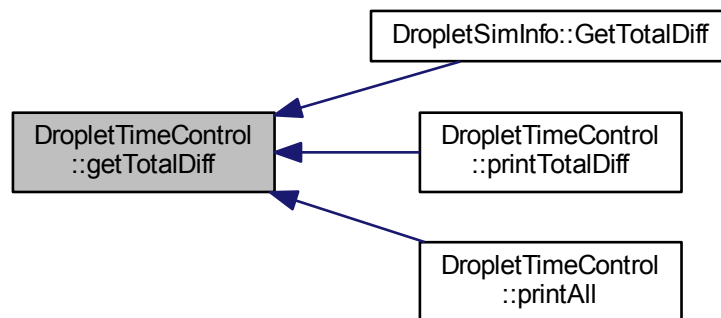
The difference between total real time and sim time.

Definition at line 94 of file `DropletTimeControl.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.13.2.4 double DropletTimeControl::getTotalRT ()

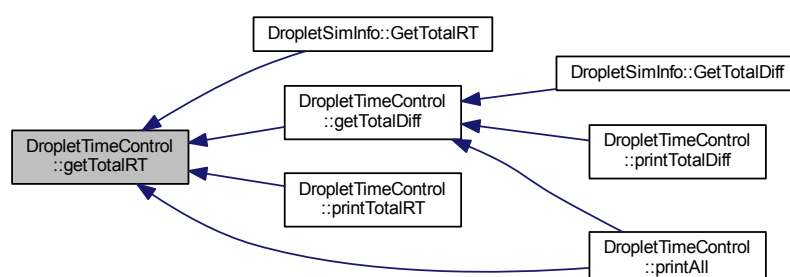
get total real time elapsed since the simulator started.

Returns

The total real time.

Definition at line 79 of file `DropletTimeControl.cpp`.

Here is the caller graph for this function:



7.13.2.5 double DropletTimeControl::getTotalST ()

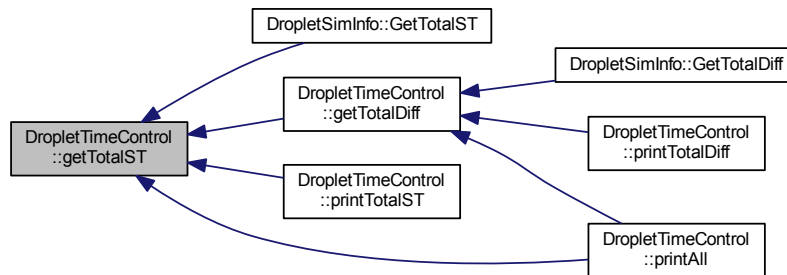
get total time simulated so far.

Returns

The total sim time.

Definition at line 84 of file DropletTimeControl.cpp.

Here is the caller graph for this function:

**7.13.2.6 void DropletTimeControl::initTimer (double sss)**

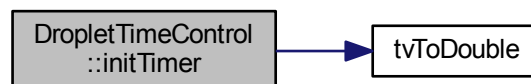
initializes the timer with a specified sim step size.

Parameters

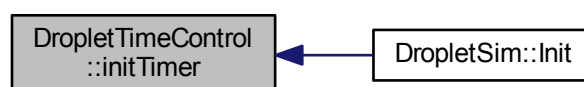
<code>sss</code>	sim step size.
------------------	----------------

Definition at line 45 of file DropletTimeControl.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

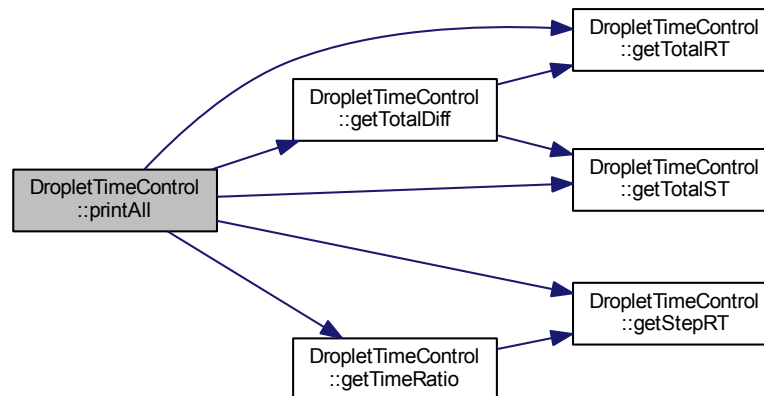


7.13.2.7 void DropletTimeControl::printAll ()

prints all calculated time values on a single line to the console.

Definition at line 131 of file DropletTimeControl.cpp.

Here is the call graph for this function:

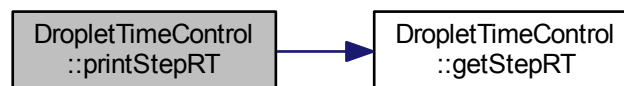


7.13.2.8 void DropletTimeControl::printStepRT ()

prints real time elapsed since the last step to the console.

Definition at line 115 of file DropletTimeControl.cpp.

Here is the call graph for this function:



7.13.2.9 void DropletTimeControl::printTimeRatio ()

prints the ratio between the sim time and real tim of the last step to the console.

Definition at line 126 of file DropletTimeControl.cpp.

Here is the call graph for this function:

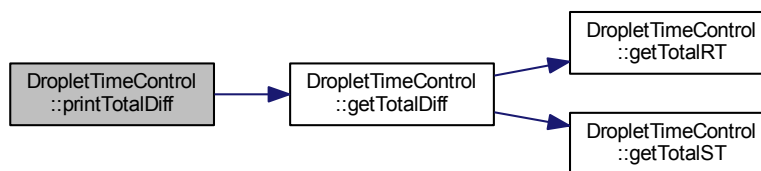


7.13.2.10 void DropletTimeControl::printTotalDiff ()

prints the difference between total real time and simulated time to the console.

Definition at line 120 of file `DropletTimeControl.cpp`.

Here is the call graph for this function:

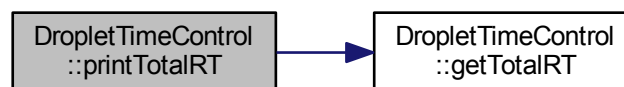


7.13.2.11 void DropletTimeControl::printTotalRT ()

prints total real time to the console.

Definition at line 105 of file `DropletTimeControl.cpp`.

Here is the call graph for this function:

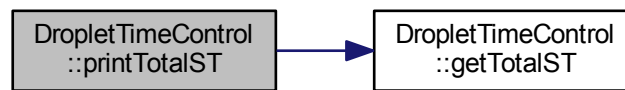


7.13.2.12 void DropletTimeControl::printTotalST ()

prints total simulator time to the console.

Definition at line 110 of file `DropletTimeControl.cpp`.

Here is the call graph for this function:



7.13.2.13 void DropletTimeControl::printVars ()

prints the stored timestamps: init time, current time, and time of the last step.

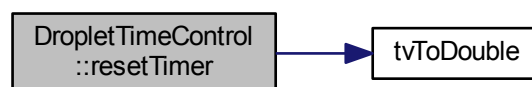
Definition at line 137 of file `DropletTimeControl.cpp`.

7.13.2.14 void DropletTimeControl::resetTimer ()

resets the timer.

Definition at line 71 of file `DropletTimeControl.cpp`.

Here is the call graph for this function:



7.13.2.15 void DropletTimeControl::updateTimer (double sss)

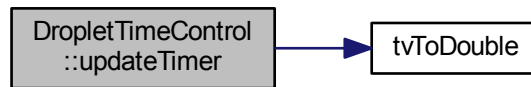
updates relevant variables and should be called each step. can be used to update the sim step size.

Parameters

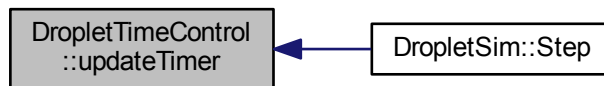
<code>sss</code>	sim step size.
------------------	----------------

Definition at line 54 of file `DropletTimeControl.cpp`.

Here is the call graph for this function:



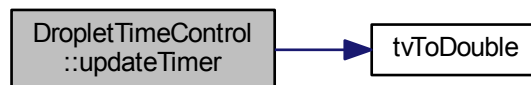
Here is the caller graph for this function:



7.13.2.16 void DropletTimeControl::updateTimer ()

Definition at line 63 of file `DropletTimeControl.cpp`.

Here is the call graph for this function:



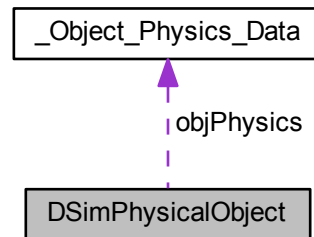
The documentation for this class was generated from the following files:

- [DropletTimeControl.h](#)
- [DropletTimeControl.cpp](#)

7.14 DSimPhysicalObject Class Reference

```
#include <DSimPhysicalObject.h>
```

Collaboration diagram for DSimPhysicalObject:



Public Member Functions

- [DSimPhysicalObject](#) ([ObjectPhysicsData](#) **objPhysics*)
- [DS_RESULT _InitPhysics](#) ([SimPhysicsData](#) **simPhysics*, std::pair< float, float > *startPosition*, float *startAngle*)
- [DS_RESULT _InitPhysics](#) ([SimPhysicsData](#) **simPhysics*, std::pair< float, float > *startPosition*, float *startHeight*, float *startAngle*)

Public Attributes

- [ObjectPhysicsData](#) * *objPhysics*

7.14.1 Detailed Description

Definition at line 19 of file `DSimPhysicalObject.h`.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 DSimPhysicalObject::DSimPhysicalObject ([ObjectPhysicsData](#) * *objPhysics*)

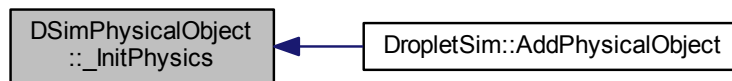
Definition at line 3 of file `DSimPhysicalObject.cpp`.

7.14.3 Member Function Documentation

7.14.3.1 DS_RESULT DSimPhysicalObject::_InitPhysics ([SimPhysicsData](#) * *simPhysics*, std::pair< float, float > *startPosition*, float *startAngle*)

Definition at line 8 of file `DSimPhysicalObject.cpp`.

Here is the caller graph for this function:



7.14.3.2 DS_RESULT DSimPhysicalObject::_InitPhysics (SimPhysicsData * *simPhysics*, std::pair< float, float > *startPosition*, float *startHeight*, float *startAngle*)

Definition at line 27 of file `DSimPhysicalObject.cpp`.

7.14.4 Member Data Documentation

7.14.4.1 ObjectPhysicsData* DSimPhysicalObject::objPhysics

Definition at line 24 of file `DSimPhysicalObject.h`.

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

- [DSimPhysicalObject.h](#)
- [DSimPhysicalObject.cpp](#)

7.15 IDroplet Class Reference

```
#include <IDroplet.h>
```

Public Member Functions

- [IDroplet](#) ([ObjectPhysicsData](#) *objPhysics)
- virtual [~IDroplet](#) ()
- [DS_RESULT _InitPhysics](#) ([SimPhysicsData](#) *simPhysics, std::pair< float, float > startPosition, float startAngle)
- virtual void [DropletInit](#) (void)
- virtual void [DropletMainLoop](#) (void)

Public Attributes

- struct {
 - [uint8_t](#) * buf
 - [uint8_t](#) data_len
 - [uint16_t](#) message_time
 - [uint8_t](#) printed_read_prompt
 - [uint8_t](#) read
 - [uint8_t](#) receivers_used
 - [droplet_id_type](#) sender_ID
 - [uint16_t](#) size

```
} global_rx_buffer
```

- [msg_order](#) [msg_return_order](#)

Protected Member Functions

Subsystem setup functions.

- void [reset_all_systems](#) (void)
- void [reset_rgb_led](#) (void)
- void [reset_ir_sensor](#) (uint8_t sensor_num)
- void [reset_rgb_sensor](#) (void)
- void [reset_motors](#) (void)
- void [reset_timers](#) (void)
- [droplet_id_type](#) [get_droplet_id](#) (void)
- [uint8_t](#) [rand_byte](#) (void)

Actuator subsystem functions.

- void [rotate_duration](#) ([turn_direction](#) direction, [uint16_t](#) duration)
- void [rotate_steps](#) ([turn_direction](#) direction, [uint16_t](#) num_steps)
- void [move_duration](#) ([uint8_t](#) direction, [uint16_t](#) duration)
- void [move_steps](#) ([uint8_t](#) direction, [uint16_t](#) num_steps)
- [uint8_t](#) [is_moving](#) (void)
- [turn_direction](#) [is_rotating](#) (void)
- [uint32_t](#) [cancel_move](#) (void)
- [uint32_t](#) [cancel_rotate](#) (void)
- void [set_rgb](#) ([uint8_t](#) r, [uint8_t](#) g, [uint8_t](#) b)
- void [set_red_led](#) ([uint8_t](#) saturation)
- void [set_green_led](#) ([uint8_t](#) saturation)
- void [set_blue_led](#) ([uint8_t](#) saturation)
- [int8_t](#) [leg1_status](#) ()
- [int8_t](#) [leg2_status](#) ()
- [int8_t](#) [leg3_status](#) ()
- void [get_rgb](#) ([uint8_t](#) *r, [uint8_t](#) *g, [uint8_t](#) *b)
- [uint8_t](#) [ir_send](#) ([uint8_t](#) channel, char *send_buf, [uint8_t](#) length)
- [uint8_t](#) [check_for_new_messages](#) (void)
- [uint8_t](#) [set_timer](#) ([uint16_t](#) time, [uint8_t](#) index)
- [uint8_t](#) [check_timer](#) ([uint8_t](#) index)

Friends

Simulator backend functions

NOTE : These friend functions are meant for use by the simulator backend ONLY. DO NOT USE THESE IN ANY CODE WRITTEN FOR A CLASS DERIVED FROM IDroplet!!!

- void [AccessPhysicsData](#) ([IDroplet](#) *pDroplet, [ObjectPhysicsData](#) **objPhysics)
- void [AccessActuatorData](#) ([IDroplet](#) *pDroplet, [DropletActuatorData](#) **actData)
- void [AccessSensorData](#) ([IDroplet](#) *pDroplet, [DropletSensorData](#) **senseData)
- void [AccessCommData](#) ([IDroplet](#) *pDroplet, [DropletCommData](#) **commData)
- void [AccessCompData](#) ([IDroplet](#) *pDroplet, [DropletCompData](#) **compData)
- void [AccessTimeData](#) ([IDroplet](#) *pDroplet, [DropletTimeData](#) **timeData)

7.15.1 Detailed Description

[IDroplet](#) models the behavior and state of an individual droplet.

Definition at line 30 of file [IDroplet.h](#).

7.15.2 Constructor & Destructor Documentation

7.15.2.1 IDroplet::IDroplet (ObjectPhysicsData * *objPhysics*)

Constructor.

Parameters

<i>in, out</i>	<i>objPhysics</i>	If non-null, the object physics.
----------------	-------------------	----------------------------------

Definition at line 4 of file IDroplet.cpp.

7.15.2.2 IDroplet::~IDroplet () [virtual]

Destructor.

Definition at line 15 of file IDroplet.cpp.

7.15.3 Member Function Documentation

7.15.3.1 DS_RESULT IDroplet::_InitPhysics (SimPhysicsData * *simPhysics*, std::pair< float, float > *startPosition*, float *startAngle*)

Initialises the initialise physics.

Parameters

<i>in, out</i>	<i>simPhysics</i>	If non-null, the simulation physics.
	<i>startPosition</i>	The start position.
	<i>startAngle</i>	The start angle.

Returns

.

Definition at line 30 of file IDroplet.cpp.

Here is the caller graph for this function:



7.15.3.2 uint32_t IDroplet::cancel_move (void) [protected]

Cancel move.

Returns

number of steps taken
duration

Definition at line 197 of file IDroplet.cpp.

7.15.3.3 uint32_t IDroplet::cancel_rotate (void) [protected]

Cancel rotate.

Returns

number of steps taken
duration

Definition at line 214 of file IDroplet.cpp.

7.15.3.4 uint8_t IDroplet::check_for_new_messages (void) [protected]

Definition at line 320 of file IDroplet.cpp.

7.15.3.5 uint8_t IDroplet::check_timer (uint8_t index) [protected]

Definition at line 388 of file IDroplet.cpp.

7.15.3.6 void IDroplet::DropletInit (void) [virtual]

Droplet Simulator Function to be overridden.

Definition at line 395 of file IDroplet.cpp.

Here is the caller graph for this function:

**7.15.3.7 void IDroplet::DropletMainLoop (void) [virtual]**

Droplet Simulator Function to be overridden.

Definition at line 396 of file IDroplet.cpp.

Here is the caller graph for this function:



7.15.3.8 `droplet_id_type IDroplet::get_droplet_id (void)` [protected]

Gets droplet identifier.

Returns

The droplet identifier.

Definition at line 163 of file IDroplet.cpp.

7.15.3.9 `void IDroplet::get_rgb (uint8_t * r, uint8_t * g, uint8_t * b)` [protected]

Returns rgb values.

/return 0-255 for each rgb value.

Definition at line 288 of file IDroplet.cpp.

7.15.3.10 `uint8_t IDroplet::ir_send (uint8_t channel, char * send_buf, uint8_t length)` [protected]

Definition at line 295 of file IDroplet.cpp.

7.15.3.11 `uint8_t IDroplet::is_moving (void)` [protected]

Returns droplet movement status.

Returns

- 0 if droplet is not moving.
- 1 through 6 depending on movement direction.

Definition at line 231 of file IDroplet.cpp.

7.15.3.12 `turn_direction IDroplet::is_rotating (void)` [protected]

Returns droplet rotation status.

Returns

- 0 if droplet is not rotating.
- 1 if droplet is rotating CW.
- 1 if droplet is rotating CCW.

Definition at line 241 of file IDroplet.cpp.

7.15.3.13 `int8_t IDroplet::leg1_status ()` `[protected]`

Returns droplet leg 1 status.

/return 1 if leg is on power. /return -1 if leg is on ground. /return 0 if leg is floating

Definition at line 250 of file IDroplet.cpp.

7.15.3.14 `int8_t IDroplet::leg2_status ()` `[protected]`

Returns droplet leg 2 status.

/return 1 if leg is on power. /return -1 if leg is on ground. /return 0 if leg is floating

Definition at line 255 of file IDroplet.cpp.

7.15.3.15 `int8_t IDroplet::leg3_status ()` `[protected]`

Returns droplet leg 3 status.

/return 1 if leg is on power. /return -1 if leg is on ground. /return 0 if leg is floating

Definition at line 260 of file IDroplet.cpp.

7.15.3.16 `void IDroplet::move_duration (uint8_t direction, uint16_t duration)` `[protected]`

Move duration.

Parameters

<i>direction</i>	The direction (1 through 6, or use macros).
<i>duration</i>	The duration in ms.

Definition at line 174 of file IDroplet.cpp.

7.15.3.17 `void IDroplet::move_steps (uint8_t direction, uint16_t num_steps)` `[protected]`

Move steps.

Parameters

<i>dir</i>	The direction (1 through 6, or use macros).
<i>num_steps</i>	The number of steps.

Definition at line 180 of file IDroplet.cpp.

7.15.3.18 `uint8_t IDroplet::rand_byte (void)` `[protected]`

Gets a random number between 0 and 255 (inclusive).

Returns

The random number.

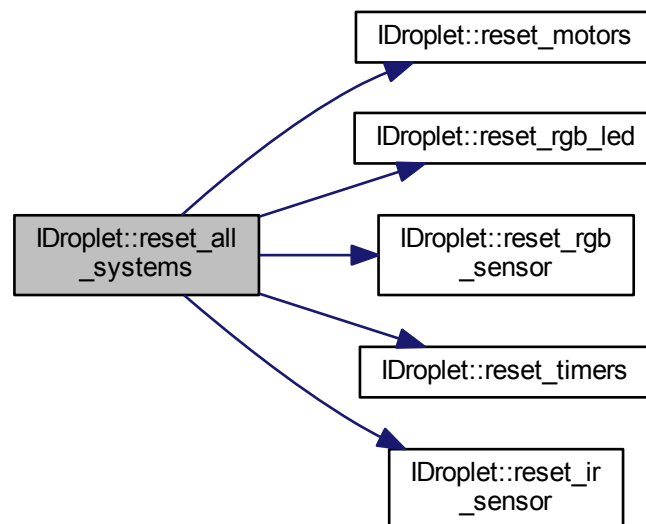
Definition at line 168 of file IDroplet.cpp.

7.15.3.19 void IDroplet::reset_all_systems (void) [protected]

Resets all systems.

Definition at line 101 of file IDroplet.cpp.

Here is the call graph for this function:



7.15.3.20 void IDroplet::reset_ir_sensor (uint8_t sensor_num) [protected]

Resets the IR LED.

Definition at line 123 of file IDroplet.cpp.

Here is the caller graph for this function:

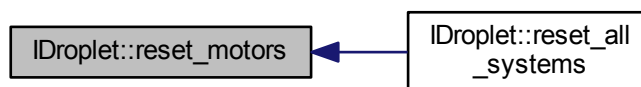


7.15.3.21 void IDroplet::reset_motors (void) [protected]

Resets the motors.

Definition at line 143 of file IDroplet.cpp.

Here is the caller graph for this function:

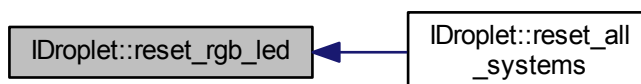


7.15.3.22 `void IDroplet::reset_rgb_led (void)` `[protected]`

Resets the RGB LED.

Definition at line 115 of file `IDroplet.cpp`.

Here is the caller graph for this function:

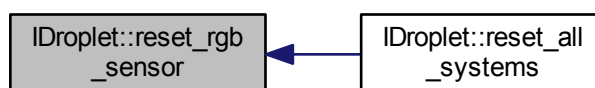


7.15.3.23 `void IDroplet::reset_rgb_sensor (void)` `[protected]`

Resets the RGB sensor.

Definition at line 136 of file `IDroplet.cpp`.

Here is the caller graph for this function:

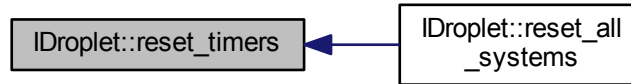


7.15.3.24 `void IDroplet::reset_timers (void)` `[protected]`

Resets the timers.

Definition at line 154 of file IDroplet.cpp.

Here is the caller graph for this function:



7.15.3.25 void IDroplet::rotate_duration (turn_direction *direction*, uint16_t *duration*) [protected]

Rotate duration.

Parameters

<i>direction</i>	The direction (1 = CW, -1 = CCW, or use macros).
<i>duration</i>	The duration in ms.

Definition at line 185 of file IDroplet.cpp.

7.15.3.26 void IDroplet::rotate_steps (turn_direction *direction*, uint16_t *num_steps*) [protected]

Rotate steps.

Parameters

<i>direction</i>	The direction (1 = CW, -1 = CCW, or use macros).
<i>num_steps</i>	The number of steps.

Definition at line 191 of file IDroplet.cpp.

7.15.3.27 void IDroplet::set_blue_led (uint8_t *saturation*) [protected]

Sets blue saturation value.

Parameters

<i>saturation</i>	The uint8_t to process.
-------------------	-------------------------

Definition at line 282 of file IDroplet.cpp.

7.15.3.28 void IDroplet::set_green_led (uint8_t *saturation*) [protected]

Sets green saturation value.

Parameters

<i>saturation</i>	The uint8_t to process.
-------------------	-------------------------

Definition at line 277 of file IDroplet.cpp.

7.15.3.29 void IDroplet::set_red_led (uint8_t *saturation*) [protected]

Sets red saturation value.

Parameters

<i>saturation</i>	The uint8_t to process.
-------------------	-------------------------

Definition at line 272 of file IDroplet.cpp.

7.15.3.30 void IDroplet::set_rgb (uint8_t *r*, uint8_t *g*, uint8_t *b*) [protected]

Sets a RGB value.

Parameters

<i>r</i>	The uint8_t to process.
<i>g</i>	The uint8_t to process.
<i>b</i>	The uint8_t to process.

Definition at line 265 of file IDroplet.cpp.

7.15.3.31 uint8_t IDroplet::set_timer (uint16_t *time*, uint8_t *index*) [protected]

Definition at line 378 of file IDroplet.cpp.

7.15.4 Friends And Related Function Documentation

7.15.4.1 void AccessActuatorData (IDroplet * *pDroplet*, DropletActuatorData ** *actData*) [friend]

Access actuator data.

NOTE : These friend functions are meant for use by the simulator backend ONLY. DO NOT USE THESE IN ANY CODE WRITTEN FOR A CLASS DERIVED FROM IDroplet!!!

Parameters

<i>in, out</i>	<i>pDroplet</i>	If non-null, the droplet.
<i>in, out</i>	<i>actData</i>	If non-null, information describing the act.

Definition at line 74 of file IDroplet.h.

7.15.4.2 void AccessCommData (IDroplet * *pDroplet*, DropletCommData ** *commData*) [friend]

Access communications data.

NOTE : These friend functions are meant for use by the simulator backend ONLY. DO NOT USE THESE IN ANY CODE WRITTEN FOR A CLASS DERIVED FROM IDroplet!!!

Parameters

<i>in, out</i>	<i>pDroplet</i>	If non-null, the droplet.
<i>in, out</i>	<i>commData</i>	If non-null, information describing the communications.

Definition at line 104 of file IDroplet.h.

7.15.4.3 `void AccessCompData (IDroplet * pDroplet, DropletCompData ** compData)` [friend]

Access component data.

NOTE : These friend functions are meant for use by the simulator backend ONLY. DO NOT USE THESE IN ANY CODE WRITTEN FOR A CLASS DERIVED FROM IDroplet!!!

Parameters

in, out	<i>pDroplet</i>	If non-null, the droplet.
in, out	<i>compData</i>	If non-null, information describing the component.

Definition at line 119 of file IDroplet.h.

7.15.4.4 `void AccessPhysicsData (IDroplet * pDroplet, ObjectPhysicsData ** objPhysics)` [friend]

Access physics data.

NOTE : These friend functions are meant for use by the simulator backend ONLY. DO NOT USE THESE IN ANY CODE WRITTEN FOR A CLASS DERIVED FROM IDroplet!!!

Parameters

in, out	<i>pDroplet</i>	If non-null, the droplet.
in, out	<i>objPhysics</i>	If non-null, the object physics.

Definition at line 59 of file IDroplet.h.

7.15.4.5 `void AccessSensorData (IDroplet * pDroplet, DropletSensorData ** senseData)` [friend]

Access sensor data.

NOTE : These friend functions are meant for use by the simulator backend ONLY. DO NOT USE THESE IN ANY CODE WRITTEN FOR A CLASS DERIVED FROM IDroplet!!!

Parameters

in, out	<i>pDroplet</i>	If non-null, the droplet.
in, out	<i>senseData</i>	If non-null, information describing the sense.

Definition at line 89 of file IDroplet.h.

7.15.4.6 `void AccessTimeData (IDroplet * pDroplet, DropletTimeData ** timeData)` [friend]

Definition at line 124 of file IDroplet.h.

7.15.5 Member Data Documentation

7.15.5.1 `uint8_t* IDroplet::buf`

Definition at line 372 of file IDroplet.h.

7.15.5.2 `uint8_t IDroplet::data_len`

Definition at line 377 of file IDroplet.h.

7.15.5.3 `struct { ... } IDroplet::global_rx_buffer`

7.15.5.4 `uint16_t IDroplet::message_time`

Definition at line 380 of file IDroplet.h.

7.15.5.5 `msg_order IDroplet::msg_return_order`

Definition at line 364 of file IDroplet.h.

7.15.5.6 `uint8_t IDroplet::printed_read_prompt`

Definition at line 383 of file IDroplet.h.

7.15.5.7 `uint8_t IDroplet::read`

Definition at line 382 of file IDroplet.h.

7.15.5.8 `uint8_t IDroplet::receivers_used`

Definition at line 375 of file IDroplet.h.

7.15.5.9 `droplet_id_type IDroplet::sender_ID`

Definition at line 378 of file IDroplet.h.

7.15.5.10 `uint16_t IDroplet::size`

Definition at line 373 of file IDroplet.h.

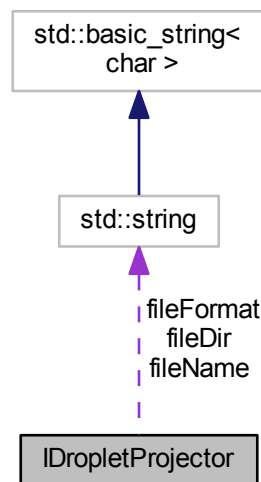
The documentation for this class was generated from the following files:

- [IDroplet.h](#)
- [IDroplet.cpp](#)

7.16 IDropletProjector Class Reference

```
#include <IDropletProjector.h>
```

Collaboration diagram for IDropletProjector:



Public Member Functions

- `IDropletProjector` (int `projLength`, int `projWidth`)
- `~IDropletProjector` ()
- virtual `DS_RESULT` `GetPixel` (float *xyLoc, `uint8_t` *rgbaVal)
- virtual `DS_RESULT` `GetPixels` (std::vector< float * > *xyLocs, std::vector< `uint8_t` * > *rgbaVals)
- virtual `DS_RESULT` `GetPixels` (float *xyTopLeft, float *xyBottomRight, `uint8_t` **rgbaVals)
- virtual `DS_RESULT` `LoadFile` (std::string `fileName`)
- virtual `DS_RESULT` `SetDirectory` (std::string `dirLocation`)

Protected Attributes

- bool `dataSet`
- std::string `fileDir`
- std::string `fileFormat`
- std::string `fileName`
- float `floorLength`
- float `floorWidth`
- BMP * `imgData`
- int `imgLength`
- int `imgWidth`
- int `projLength`
- float `projPixelLength`
- float `projPixelWidth`
- int `projWidth`

7.16.1 Detailed Description

`IDropletProjector` controls the projector inside the arena.

Definition at line 27 of file `IDropletProjector.h`.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 IDropletProjector::IDropletProjector (int *projLength*, int *projWidth*)

Constructor.

Parameters

<i>projLength</i>	Length of the project.
<i>projWidth</i>	Width of the project.

Definition at line 3 of file IDropletProjector.cpp.

7.16.2.2 IDropletProjector::~~IDropletProjector ()

Destructor.

Definition at line 19 of file IDropletProjector.cpp.

7.16.3 Member Function Documentation

7.16.3.1 DS_RESULT IDropletProjector::GetPixel (float * *xyLoc*, uint8_t * *rgbaVal*) [virtual]

Gets the RGBA value of a pixel at an xy location.

Parameters

in, out	<i>xyLoc</i>	If non-null, the xy location.
in, out	<i>rgbaVal</i>	If non-null, the RGBA value.

Returns

The pixel.

Definition at line 67 of file IDropletProjector.cpp.

7.16.3.2 DS_RESULT IDropletProjector::GetPixels (std::vector< float * > * *xyLocs*, std::vector< uint8_t * > * *rgbaVals*) [virtual]

Gets the RGBA values of pixels at a specified set of xy locations.

Parameters

in, out	<i>xyLocs</i>	If non-null, the xy locs.
in, out	<i>rgbaVals</i>	If non-null, the RGBA vals.

Returns

The pixels.

Definition at line 97 of file IDropletProjector.cpp.

7.16.3.3 DS_RESULT IDropletProjector::GetPixels (float * *xyTopLeft*, float * *xyBottomRight*, uint8_t ** *rgbaVals*) [virtual]

Gets the RGBA values of pixels in a specified area.

Parameters

<i>in, out</i>	<i>xyTopLeft</i>	If non-null, the xy top left.
<i>in, out</i>	<i>xyBottomRight</i>	If non-null, the xy bottom right.
<i>in, out</i>	<i>rgbaVals</i>	If non-null, the RGBA vals.

Returns

The pixels.

Definition at line 139 of file IDropletProjector.cpp.

7.16.3.4 DS_RESULT IDropletProjector::LoadFile (std::string *fileName*) [virtual]

Loads a bitmap file into the projector.

Parameters

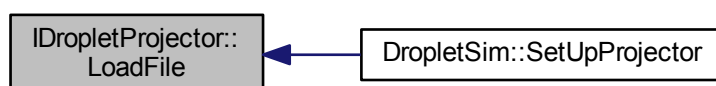
<i>fileName</i>	Filename of the file.
-----------------	-----------------------

Returns

The file.

Definition at line 32 of file IDropletProjector.cpp.

Here is the caller graph for this function:



7.16.3.5 DS_RESULT IDropletProjector::SetDirectory (std::string *dirLocation*) [virtual]

Sets the working directory for bitmap files.

Parameters

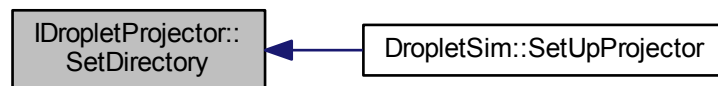
<i>dirLocation</i>	The dir location.
--------------------	-------------------

Returns

.

Definition at line 24 of file IDropletProjector.cpp.

Here is the caller graph for this function:



7.16.4 Member Data Documentation

7.16.4.1 `bool IDropletProjector::dataSet` [protected]

True if data is set.

Definition at line 42 of file IDropletProjector.h.

7.16.4.2 `std::string IDropletProjector::fileDir` [protected]

Definition at line 36 of file IDropletProjector.h.

7.16.4.3 `std::string IDropletProjector::fileFormat` [protected]

Definition at line 36 of file IDropletProjector.h.

7.16.4.4 `std::string IDropletProjector::fileName` [protected]

Definition at line 36 of file IDropletProjector.h.

7.16.4.5 `float IDropletProjector::floorLength` [protected]

Definition at line 33 of file IDropletProjector.h.

7.16.4.6 `float IDropletProjector::floorWidth` [protected]

Definition at line 33 of file IDropletProjector.h.

7.16.4.7 `BMP* IDropletProjector::imgData` [protected]

Information describing the image.

Definition at line 48 of file IDropletProjector.h.

7.16.4.8 int IDropletProjector::imgLength [protected]

Definition at line 31 of file IDropletProjector.h.

7.16.4.9 int IDropletProjector::imgWidth [protected]

Definition at line 31 of file IDropletProjector.h.

7.16.4.10 int IDropletProjector::projLength [protected]

Definition at line 32 of file IDropletProjector.h.

7.16.4.11 float IDropletProjector::projPixelLength [protected]

Definition at line 34 of file IDropletProjector.h.

7.16.4.12 float IDropletProjector::projPixelWidth [protected]

Definition at line 34 of file IDropletProjector.h.

7.16.4.13 int IDropletProjector::projWidth [protected]

Definition at line 32 of file IDropletProjector.h.

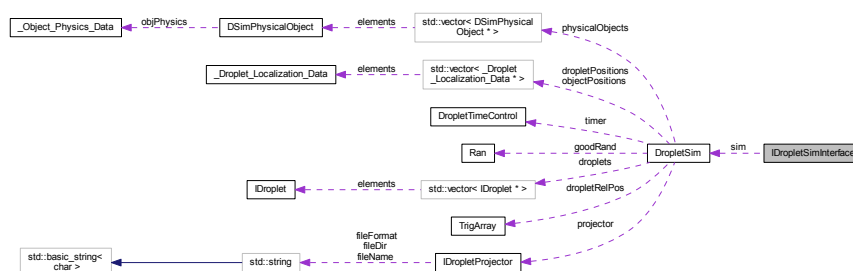
The documentation for this class was generated from the following files:

- [IDropletProjector.h](#)
- [IDropletProjector.cpp](#)

7.17 IDropletSimInterface Class Reference

```
#include <IDropletSimInterface.h>
```

Collaboration diagram for IDropletSimInterface:



Public Member Functions

- [IDropletSimInterface](#) (void)
- virtual [~IDropletSimInterface](#) (void)
- [DS_RESULT CreateDroplet](#) ()

- [DS_RESULT InitializeSim](#) (char *initFilePath)
- [DS_RESULT SetDropletCollisionShape](#) ([BasicObjectShapes](#) shape)

Protected Attributes

- [DropletSim](#) * sim

7.17.1 Detailed Description

Definition at line 18 of file IDropletSimInterface.h.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 IDropletSimInterface::IDropletSimInterface (void)

7.17.2.2 virtual IDropletSimInterface::~IDropletSimInterface (void) [virtual]

7.17.3 Member Function Documentation

7.17.3.1 DS_RESULT IDropletSimInterface::CreateDroplet ()

7.17.3.2 DS_RESULT IDropletSimInterface::InitializeSim (char * *initFilePath*)

7.17.3.3 DS_RESULT IDropletSimInterface::SetDropletCollisionShape ([BasicObjectShapes](#) *shape*)

7.17.4 Member Data Documentation

7.17.4.1 DropletSim* IDropletSimInterface::sim [protected]

Definition at line 24 of file IDropletSimInterface.h.

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

- [IDropletSimInterface.h](#)

7.18 Ran Struct Reference

```
#include <DropletUtil.h>
```

Public Member Functions

- [Ran](#) (uint64_t j)
- double [doub](#) ()
- uint64_t [int64](#) ()

Public Attributes

- uint64_t u
- uint64_t v
- uint64_t w

7.18.1 Detailed Description

Definition at line 22 of file DropletUtil.h.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 `Ran::Ran(uint64_t j)` `[inline]`

Definition at line 27 of file DropletUtil.h.

Here is the call graph for this function:



7.18.3 Member Function Documentation

7.18.3.1 `double Ran::doub()` `[inline]`

Definition at line 43 of file DropletUtil.h.

Here is the call graph for this function:



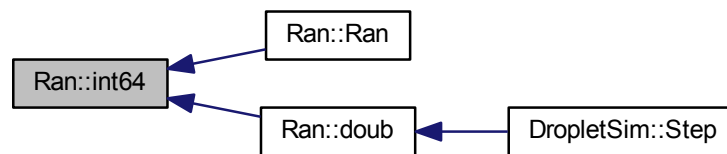
Here is the caller graph for this function:



7.18.3.2 `uint64_t Ran::int64()` `[inline]`

Definition at line 34 of file DropletUtil.h.

Here is the caller graph for this function:



7.18.4 Member Data Documentation

7.18.4.1 uint64_t Ran::u

Definition at line 24 of file DropletUtil.h.

7.18.4.2 uint64_t Ran::v

Definition at line 24 of file DropletUtil.h.

7.18.4.3 uint64_t Ran::w

Definition at line 24 of file DropletUtil.h.

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

- [DropletUtil.h](#)

7.19 SimSetupData Class Reference

```
#include <DropletSim.h>
```

Public Member Functions

- [SimSetupData](#) (int numRowsTiles, int numColTiles, float tileLength, float dropletRadius, float fps, bool auto-BuildBoundaryWalls)
- [SimSetupData](#) (const [SimSetupData](#) &setupData)

Friends

- class [DropletSim](#)

7.19.1 Detailed Description

Class containing simulation setup data.

Definition at line 293 of file DropletSim.h.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 SimSetupData::SimSetupData (int numRowsTiles, int numColTiles, float tileLength, float dropletRadius, float fps, bool autoBuildBoundaryWalls)

Constructor.

Parameters

<i>numRowTiles</i>	Number of row tiles.
<i>numColTiles</i>	Number of col tiles.
<i>tileLength</i>	Length of the tile.
<i>dropletRadius</i>	The droplet radius.
<i>fps</i>	The FPS.
<i>autoBuildBoundaryWalls</i>	true to automatically build boundary walls.

Definition at line 977 of file DropletSim.cpp.

7.19.2.2 SimSetupData::SimSetupData (const SimSetupData & setupData)

Copy Constructor.

Parameters

<i>setupData</i>	Information describing the setup.
------------------	-----------------------------------

Definition at line 998 of file DropletSim.cpp.

7.19.3 Friends And Related Function Documentation

7.19.3.1 friend class DropletSim [friend]

Definition at line 302 of file DropletSim.h.

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

- [DropletSim.h](#)
- [DropletSim.cpp](#)

7.20 TrigArray Class Reference

```
#include <DropletUtil.h>
```

Public Member Functions

- [TrigArray](#) (unsigned int numRowsTiles)
- [~TrigArray](#) (void)
- [DS_RESULT AddData](#) (unsigned int d1, unsigned int d2, float dist, float angle)
- float [GetAngle](#) (unsigned int d1, unsigned int d2)
- [DS_RESULT GetData](#) (unsigned int d1, unsigned int d2, float *dist, float *angle)
- float [GetDistance](#) (unsigned int d1, unsigned int d2)
- [DS_RESULT RemoveData](#) (unsigned int d1)

7.20.1 Detailed Description

Definition at line 46 of file DropletUtil.h.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 TrigArray::TrigArray (unsigned int *numDroplets*)

Definition at line 3 of file DropletUtil.cpp.

7.20.2.2 TrigArray::~TrigArray (void)

Definition at line 17 of file DropletUtil.cpp.

7.20.3 Member Function Documentation

7.20.3.1 DS_RESULT TrigArray::AddData (unsigned int *d1*, unsigned int *d2*, float *dist*, float *angle*)

Definition at line 23 of file DropletUtil.cpp.

7.20.3.2 float TrigArray::GetAngle (unsigned int *d1*, unsigned int *d2*)

Definition at line 68 of file DropletUtil.cpp.

7.20.3.3 DS_RESULT TrigArray::GetData (unsigned int *d1*, unsigned int *d2*, float * *dist*, float * *angle*)

Definition at line 45 of file DropletUtil.cpp.

7.20.3.4 float TrigArray::GetDistance (unsigned int *d1*, unsigned int *d2*)

Definition at line 58 of file DropletUtil.cpp.

7.20.3.5 DS_RESULT TrigArray::RemoveData (unsigned int *d1*)

Definition at line 36 of file DropletUtil.cpp.

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

- [DropletUtil.h](#)
- [DropletUtil.cpp](#)

Chapter 8

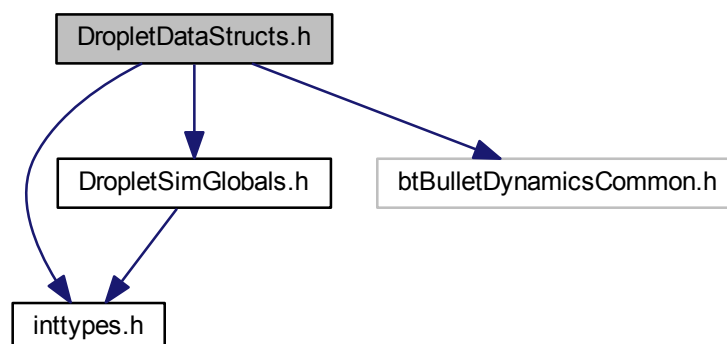
File Documentation

8.1 doxygen.dox File Reference

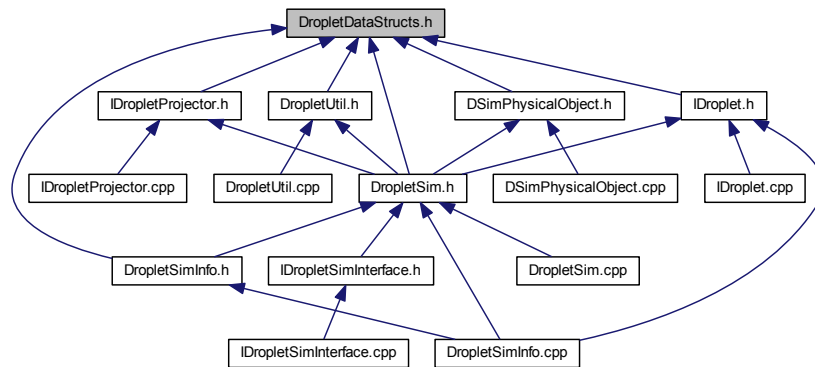
8.2 DropletDataStructs.h File Reference

This file contains private data structures to be used by the simulator only! Public data structures used for returning information to an external caller are defined in [DropletSimInfo.h](#).

```
#include <inttypes.h>
#include "DropletSimGlobals.h"
#include "btBulletDynamicsCommon.h"
Include dependency graph for DropletDataStructs.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [_Droplet_Actuator_Data](#)
- struct [_Droplet_Communication_Data](#)
- struct [_Droplet_Component_Data](#)
- struct [_Droplet_Localization_Data](#)
- struct [_Droplet_Sensor_Data](#)
- struct [_Droplet_Sim_Comm_Channel_Data](#)
- struct [_Droplet_Timing_Data](#)
- struct [_Object_Physics_Data](#)
- struct [_Simulator_Physics_Data](#)

Macros

- `#define _DROPLET_DATA_STRUCTS`

Typedefs

- typedef struct [_Droplet_Actuator_Data](#) DropletActuatorData
- typedef struct [_Droplet_Sim_Comm_Channel_Data](#) DropletCommChannelData
- typedef struct [_Droplet_Communication_Data](#) DropletCommData
- typedef struct [_Droplet_Component_Data](#) DropletCompData
- typedef struct [_Droplet_Sensor_Data](#) DropletSensorData
- typedef struct [_Droplet_Timing_Data](#) DropletTimeData
- typedef struct [_Droplet_Localization_Data](#) GPSInfo
- typedef struct [_Object_Physics_Data](#) ObjectPhysicsData
- typedef struct [_Simulator_Physics_Data](#) SimPhysicsData

8.2.1 Detailed Description

This file contains private data structures to be used by the simulator only! Public data structures used for returning information to an external caller are defined in [DropletSimInfo.h](#).

Definition in file [DropletDataStructs.h](#).

8.2.2 Macro Definition Documentation

8.2.2.1 `#define _DROPLET_DATA_STRUCTS`

Definition at line 12 of file DropletDataStructs.h.

8.2.3 Typedef Documentation

8.2.3.1 `typedef struct _Droplet_Actuator_Data DropletActuatorData`

Defines an alias representing information describing the droplet actuator.

8.2.3.2 `typedef struct _Droplet_Sim_Comm_Channel_Data DropletCommChannelData`

Defines an internally used struct to store message information per channel.

8.2.3.3 `typedef struct _Droplet_Communication_Data DropletCommData`

Defines an alias representing information describing the droplet communication.

8.2.3.4 `typedef struct _Droplet_Component_Data DropletCompData`

Defines an alias representing information describing the droplet component.

8.2.3.5 `typedef struct _Droplet_Sensor_Data DropletSensorData`

Defines an alias representing information describing the droplet sensor.

8.2.3.6 `typedef struct _Droplet_Timing_Data DropletTimeData`

8.2.3.7 `typedef struct _Droplet_Localization_Data GPSInfo`

Stores localization information for droplets.

8.2.3.8 `typedef struct _Object_Physics_Data ObjectPhysicsData`

Defines an alias representing information describing the object physics.

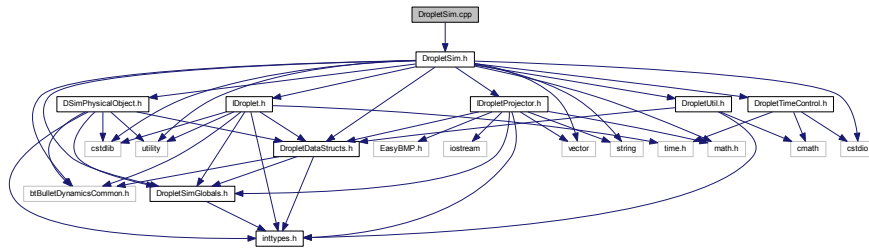
8.2.3.9 `typedef struct _Simulator_Physics_Data SimPhysicsData`

Defines an alias representing information describing the simulator physics.

8.3 DropletSim.cpp File Reference

```
#include "DropletSim.h"
```

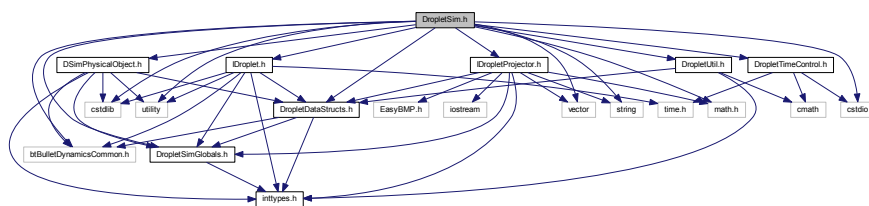
Include dependency graph for DropletSim.cpp:



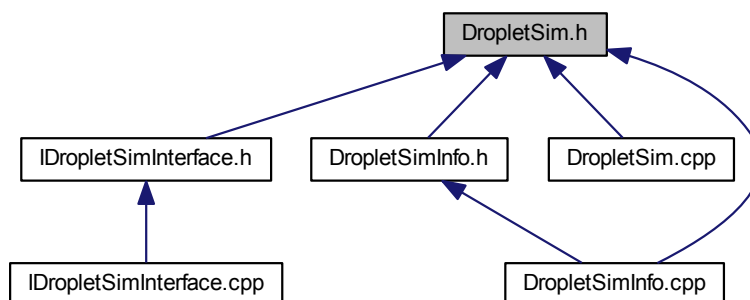
8.4 DropletSim.h File Reference

```
#include <btBulletDynamicsCommon.h>
#include "DropletDataStructs.h"
#include "DropletSimGlobals.h"
#include "DropletTimeControl.h"
#include "IDropletProjector.h"
#include "DropletUtil.h"
#include "IDroplet.h"
#include "DSimPhysicalObject.h"
#include <vector>
#include <cstdio>
#include <cstdlib>
#include <utility>
#include <string>
#include <math.h>
```

Include dependency graph for DropletSim.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DropletSim](#)
- class [SimSetupData](#)

Macros

- `#define _DROPLET_SIM`

8.4.1 Macro Definition Documentation

8.4.1.1 `#define _DROPLET_SIM`

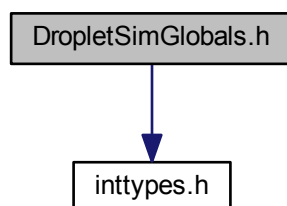
Definition at line 4 of file `DropletSim.h`.

8.5 DropletSimGlobals.h File Reference

This file contains global variables.

```
#include <inttypes.h>
```

Include dependency graph for `DropletSimGlobals.h`:



- `#define STEP_TIME 250`
- `#define TURN_CLOCKWISE 1`
- `#define TURN_COUNTERCLOCKWISE -1`
- `#define TURN_OFF 0`
- `#define WALK_STEP_TIME 1000.f / 60.f`

Typedefs

- `typedef uint16_t droplet_id_type`
droplet ID types.
- `typedef unsigned char DS_RESULT`
Simulator Error Codes.
- `typedef uint8_t move_direction`
droplet move directions
- `typedef uint8_t msg_order`
- `typedef short int turn_direction`
droplet rotation directions

8.5.1 Detailed Description

This file contains global variables.

Definition in file [DropletSimGlobals.h](#).

8.5.2 Macro Definition Documentation

8.5.2.1 `#define _DROPLET_SIM_GLOBALS`

Definition at line 10 of file [DropletSimGlobals.h](#).

8.5.2.2 `#define BROADCAST_THRESHOLD 25.f`

Definition at line 95 of file [DropletSimGlobals.h](#).

8.5.2.3 `#define DROPLET_ANGULAR_DAMPING .9f`

Definition at line 110 of file [DropletSimGlobals.h](#).

8.5.2.4 `#define DROPLET_ID_START 100`

Definition at line 84 of file [DropletSimGlobals.h](#).

8.5.2.5 `#define DROPLET_LINEAR_DAMPING .9f`

Definition at line 109 of file [DropletSimGlobals.h](#).

8.5.2.6 `#define DROPLET_NUM_TIMERS 5`

Definition at line 83 of file [DropletSimGlobals.h](#).

8.5.2.7 #define DROPLET_REL_POS_UPDATE_TIME .1f

Definition at line 94 of file DropletSimGlobals.h.

8.5.2.8 #define DS_ERROR 2

Definition at line 31 of file DropletSimGlobals.h.

8.5.2.9 #define DS_FATAL 3

Definition at line 32 of file DropletSimGlobals.h.

8.5.2.10 #define DS_SUCCESS 0

Definition at line 29 of file DropletSimGlobals.h.

8.5.2.11 #define DS_WARNING 1

Definition at line 30 of file DropletSimGlobals.h.

8.5.2.12 #define FLOOR_FRICTION .5f

Definition at line 106 of file DropletSimGlobals.h.

8.5.2.13 #define IMPULSE_SCALING 2.f

Definition at line 103 of file DropletSimGlobals.h.

8.5.2.14 #define IR_BUFFER_SIZE IR_MSG_HEADER + IR_MAX_DATA_SIZE

Definition at line 99 of file DropletSimGlobals.h.

8.5.2.15 #define IR_MAX_DATA_SIZE 72

Definition at line 98 of file DropletSimGlobals.h.

8.5.2.16 #define IR_MSG_HEADER sizeof(droplet_id_type) + sizeof(uint8_t)

Definition at line 97 of file DropletSimGlobals.h.

8.5.2.17 #define IR_RX_STATUS_BUSY 1

a macro that defines IR RX status busy

Definition at line 72 of file DropletSimGlobals.h.

8.5.2.18 #define MOTOR_POS_SCALING .95f

Definition at line 104 of file DropletSimGlobals.h.

8.5.2.19 #define MOVE_OFF 0

Definition at line 54 of file DropletSimGlobals.h.

8.5.2.20 #define MOVE_TIME 10000

Definition at line 116 of file DropletSimGlobals.h.

8.5.2.21 #define NEWEST_MSG_FIRST 1

Definition at line 91 of file DropletSimGlobals.h.

8.5.2.22 #define NORTH 1

Definition at line 55 of file DropletSimGlobals.h.

8.5.2.23 #define NORTH_EAST 2

Definition at line 56 of file DropletSimGlobals.h.

8.5.2.24 #define NORTH_WEST 6

Definition at line 60 of file DropletSimGlobals.h.

8.5.2.25 #define OBJECT_ANGULAR_DAMPING .9f

Definition at line 113 of file DropletSimGlobals.h.

8.5.2.26 #define OBJECT_LINEAR_DAMPING .9f

Definition at line 112 of file DropletSimGlobals.h.

8.5.2.27 #define OLDEST_MSG_FIRST 0

Definition at line 90 of file DropletSimGlobals.h.

8.5.2.28 #define PHYSICS_GRAVITY 9.8f

Definition at line 102 of file DropletSimGlobals.h.

8.5.2.29 #define SAFE_DELETE(x) { if(x) { delete x; x = NULL; } }

Definition at line 19 of file DropletSimGlobals.h.

8.5.2.30 #define SOUTH 4

Definition at line 58 of file DropletSimGlobals.h.

8.5.2.31 `#define SOUTH_EAST 3`

Definition at line 57 of file DropletSimGlobals.h.

8.5.2.32 `#define SOUTH_WEST 5`

Definition at line 59 of file DropletSimGlobals.h.

8.5.2.33 `#define STAGGERED_START false`

Definition at line 87 of file DropletSimGlobals.h.

8.5.2.34 `#define STEP_TIME 250`

Definition at line 117 of file DropletSimGlobals.h.

8.5.2.35 `#define TURN_CLOCKWISE 1`

Definition at line 43 of file DropletSimGlobals.h.

8.5.2.36 `#define TURN_COUNTERCLOCKWISE -1`

Definition at line 44 of file DropletSimGlobals.h.

8.5.2.37 `#define TURN_OFF 0`

Definition at line 42 of file DropletSimGlobals.h.

8.5.2.38 `#define WALK_STEP_TIME 1000.f / 60.f`

Definition at line 119 of file DropletSimGlobals.h.

8.5.3 Typedef Documentation

8.5.3.1 `uint16_t droplet_id_type`

droplet ID types.

Definition at line 85 of file DropletSimGlobals.h.

8.5.3.2 `unsigned char DS_RESULT`

Simulator Error Codes.

Definition at line 33 of file DropletSimGlobals.h.

8.5.3.3 `uint8_t move_direction`

droplet move directions

Definition at line 61 of file DropletSimGlobals.h.

8.5.3.4 typedef uint8_t msg_order

Definition at line 92 of file DropletSimGlobals.h.

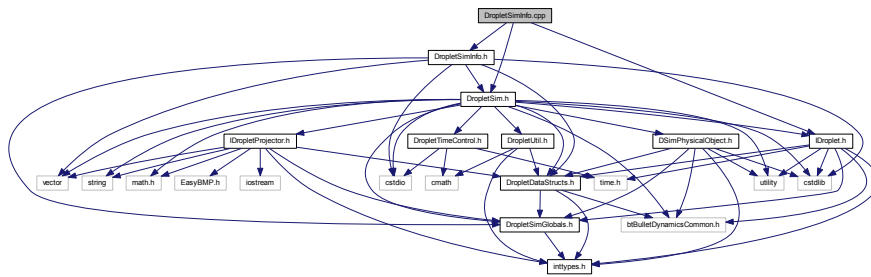
8.5.3.5 short int turn_direction

droplet rotation directions

Definition at line 45 of file DropletSimGlobals.h.

8.6 DropletSimInfo.cpp File Reference

```
#include "DropletSimInfo.h"
#include "DropletSim.h"
#include "IDroplet.h"
Include dependency graph for DropletSimInfo.cpp:
```

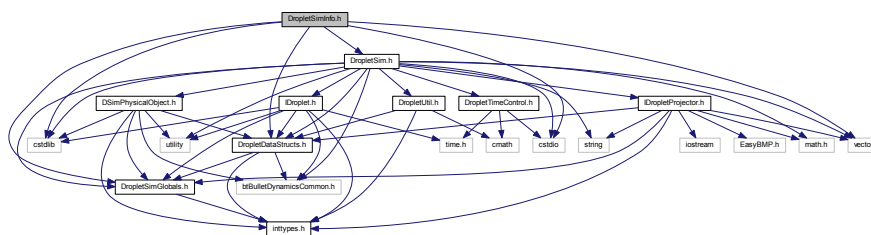


8.7 DropletSimInfo.h File Reference

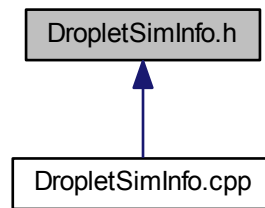
Declares the droplet simulation information class that gives access to info stored by the structs declared in DropletDataStructs and timing info calculated by [DropletTimeControl](#). Detailed info on the data structs can be found in: [cu-droplet/DropletSimulator/DropletSimLibrary/include/DropletDataStructs.h](#) Details on Droplet Simulator return codes (DS_RESULT) can be found in: [cu-droplet/DropletSimulator/DropletSimLibrary/include/DropletSimGlobals.h](#).

```
#include "DropletSimGlobals.h"
#include "DropletSim.h"
#include "DropletDataStructs.h"
#include <vector>
#include <cstdio>
#include <cstdlib>
```

Include dependency graph for DropletSimInfo.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Droplet_Motion_Direction_Data](#)
- class [DropletSimInfo](#)

Helper class used for retrieving information from the simulator. Retrieves any interesting info stored in DropletDataStructs.

Typedefs

- typedef struct [Droplet_Motion_Direction_Data](#) DirInfo

8.7.1 Detailed Description

Declares the droplet simulation information class that gives access to info stored by the structs declared in DropletDataStructs and timing info calculated by [DropletTimeControl](#). Detailed info on the data structs can be found in: [cu-droplet\DropletSimulator\DropletSimLibrary\include\DropletDataStructs.h](#) Details on Droplet Simulator return codes (DS_RESULT) can be found in: [cu-droplet\DropletSimulator\DropletSimLibrary\include\DropletSimGlobals.h](#).

Definition in file [DropletSimInfo.h](#).

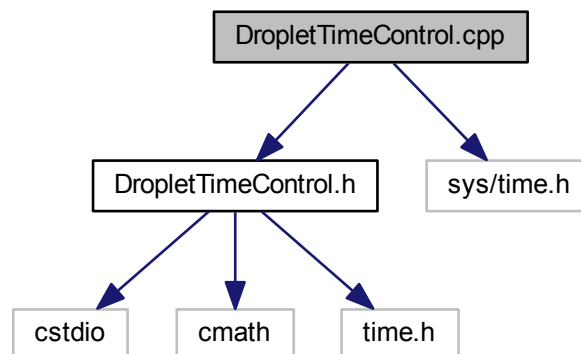
8.7.2 Typedef Documentation

8.7.2.1 typedef struct [Droplet_Motion_Direction_Data](#) DirInfo

8.8 DropletTimeControl.cpp File Reference

```
#include "DropletTimeControl.h"  
#include <sys/time.h>
```

Include dependency graph for DropletTimeControl.cpp:



Functions

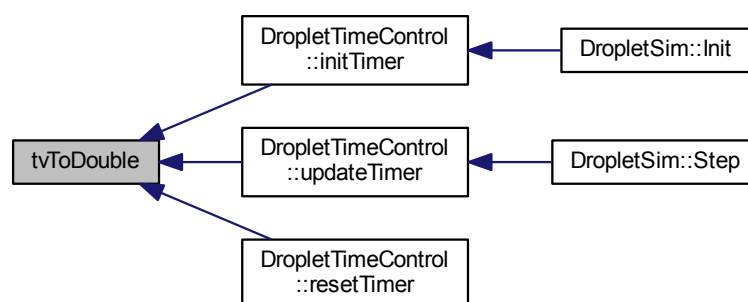
- double [tvToDouble](#) (struct timeval time)

8.8.1 Function Documentation

8.8.1.1 double tvToDouble (struct timeval *time*)

Definition at line 41 of file DropletTimeControl.cpp.

Here is the caller graph for this function:

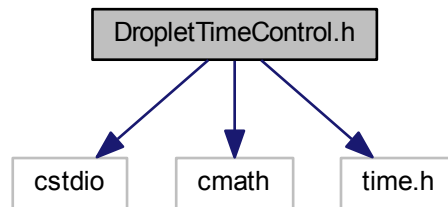


8.9 DropletTimeControl.h File Reference

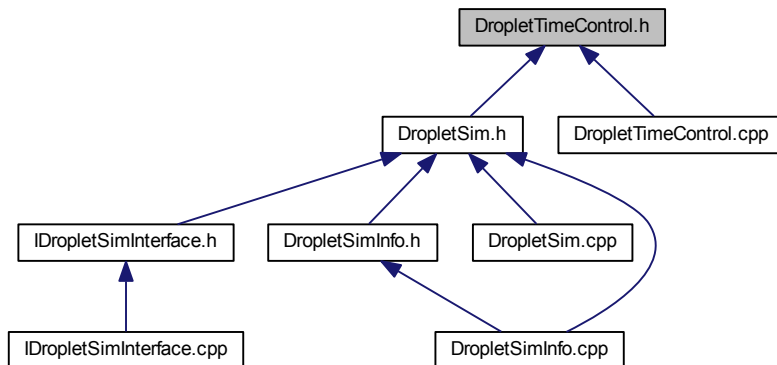
Declares the droplet time control class that gathers timing info. Times are returned as doubles in seconds. The UNIX compilation calculates time with microsecond accuracy and the windows compilation calculates time with millisecond accuracy but both are only displayed to millisecond accuracy.

```
#include <cstdio>
#include <cmath>
#include <time.h>
```

Include dependency graph for DropletTimeControl.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DropletTimeControl](#)

class used for calculating timing information for the Droplet Simulator. In order to access timing information from outside the simulator, use [DropletSimInfo](#).

Macros

- `#define _DROPLET_TIME_CONTROL`

A macro that defines droplet time control.

8.9.1 Detailed Description

Declares the droplet time control class that gathers timing info. Times are returned as doubles in seconds. The UNIX compilation calculates time with microsecond accuracy and the windows compilation calculates time with

millisecond accuracy but both are only displayed to millisecond accuracy.

Definition in file [DropletTimeControl.h](#).

8.9.2 Macro Definition Documentation

8.9.2.1 #define _DROPLET_TIME_CONTROL

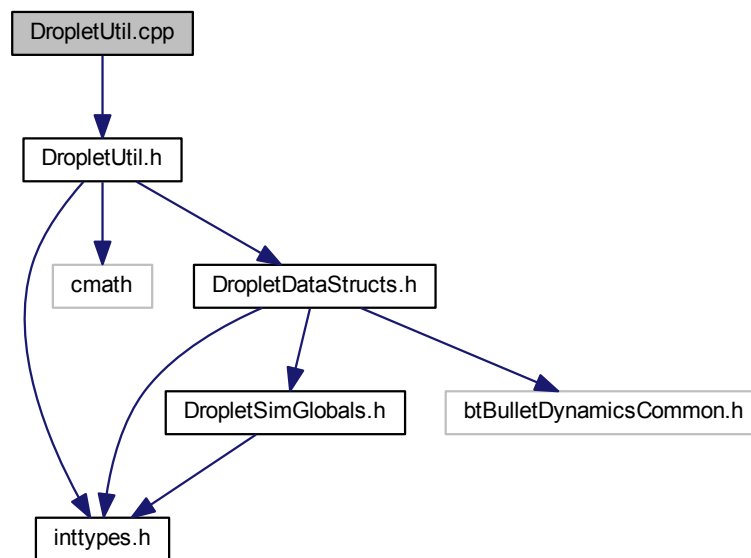
A macro that defines droplet time control.

Definition at line 34 of file [DropletTimeControl.h](#).

8.10 DropletUtil.cpp File Reference

```
#include "DropletUtil.h"
```

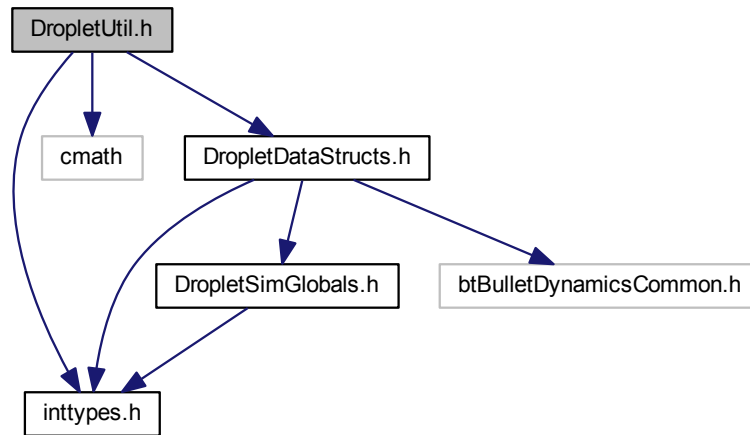
Include dependency graph for [DropletUtil.cpp](#):



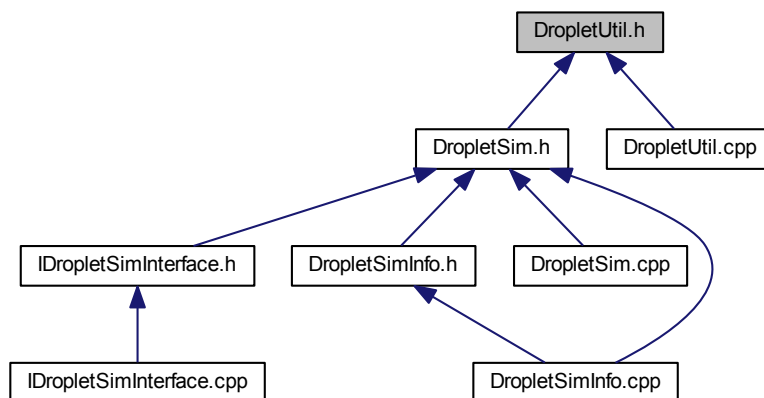
8.11 DropletUtil.h File Reference

```
#include <inttypes.h>
#include <cmath>
#include "DropletDataStructs.h"
```

Include dependency graph for DropletUtil.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Ran](#)
- class [TrigArray](#)

Macros

- `#define` [_DROPLET_UTIL](#)

8.11.1 Macro Definition Documentation

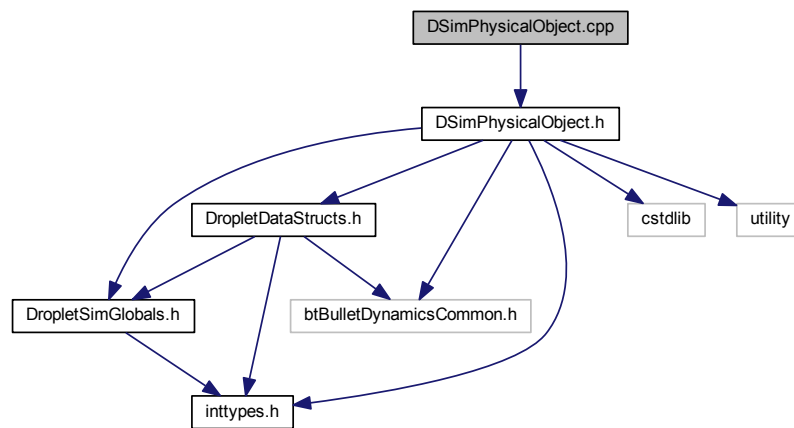
8.11.1.1 #define _DROPLET_UTIL

Definition at line 10 of file DropletUtil.h.

8.12 DSImPhysicalObject.cpp File Reference

```
#include "DSimPhysicalObject.h"
```

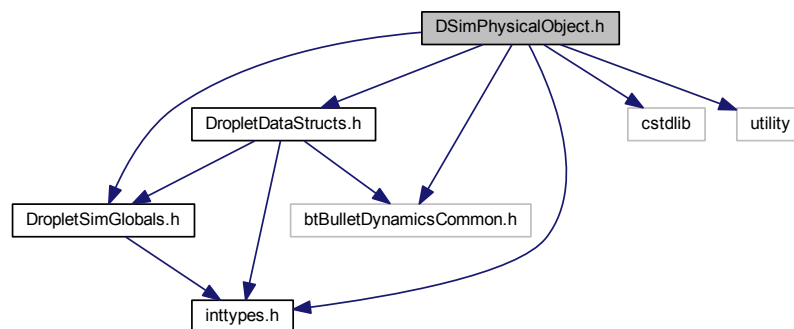
Include dependency graph for DSImPhysicalObject.cpp:



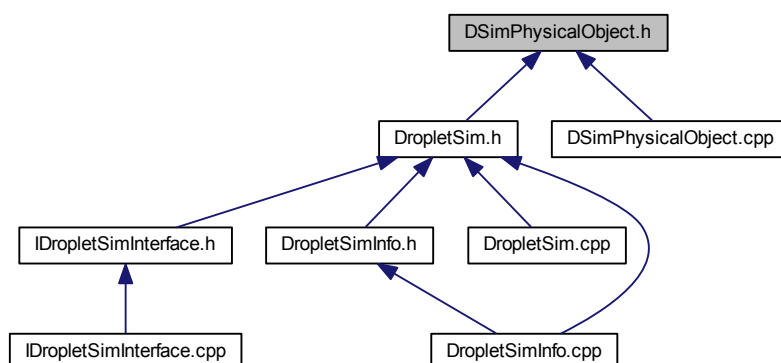
8.13 DSImPhysicalObject.h File Reference

```
#include <inttypes.h>
#include "btBulletDynamicsCommon.h"
#include "DropletSimGlobals.h"
#include "DropletDataStructs.h"
#include <cstdlib>
#include <utility>
```

Include dependency graph for DSImPhysicalObject.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DSIMPhysicalObject](#)

Macros

- [#define _DSIM_PHYSICAL_OBJECT](#)

8.13.1 Macro Definition Documentation

8.13.1.1 [#define _DSIM_PHYSICAL_OBJECT](#)

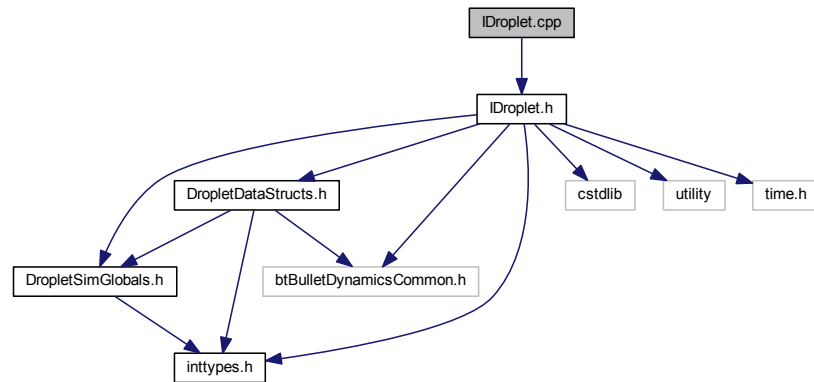
Definition at line 4 of file DSIMPhysicalObject.h.

8.14 features.dox File Reference

8.15 IDroplet.cpp File Reference

```
#include "IDroplet.h"
```

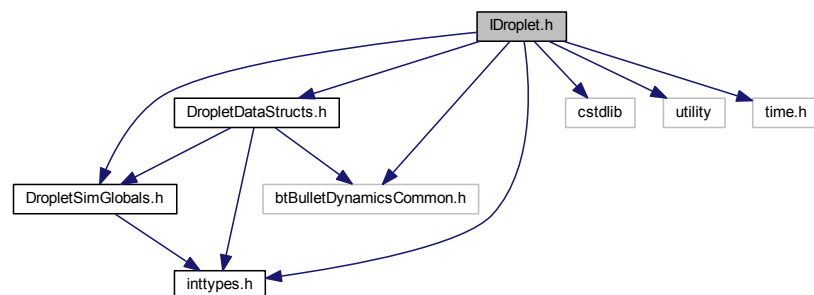
Include dependency graph for IDroplet.cpp:



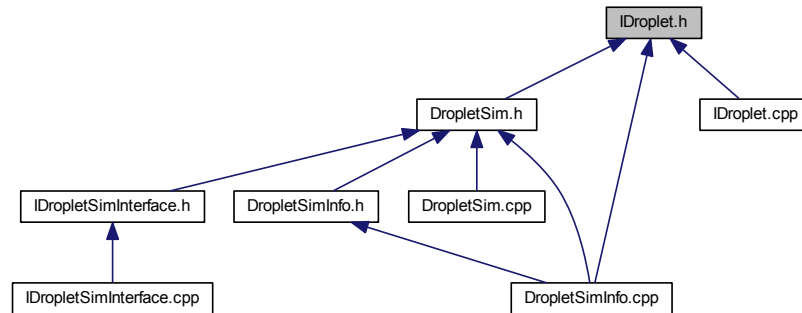
8.16 IDroplet.h File Reference

```
#include <inttypes.h>
#include "btBulletDynamicsCommon.h"
#include "DropletSimGlobals.h"
#include "DropletDataStructs.h"
#include <cstdlib>
#include <utility>
#include <time.h>
```

Include dependency graph for IDroplet.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IDroplet](#)

Macros

- `#define _I_DROPLET`

8.16.1 Macro Definition Documentation

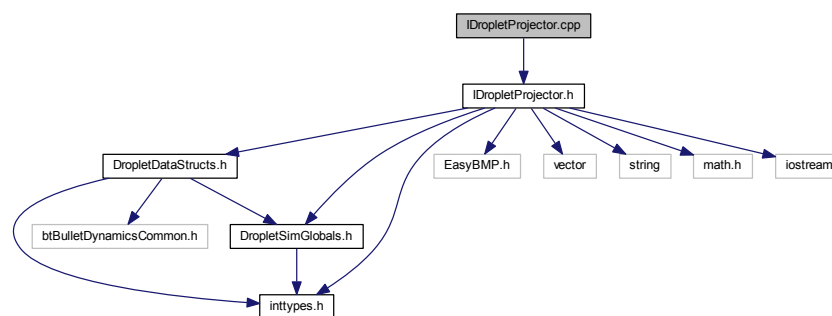
8.16.1.1 `#define _I_DROPLET`

Definition at line 4 of file `IDroplet.h`.

8.17 IDropletProjector.cpp File Reference

```
#include "IDropletProjector.h"
```

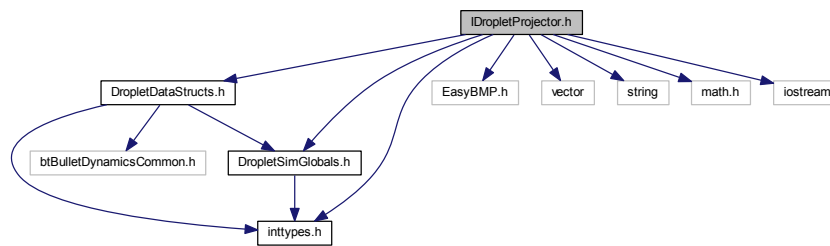
Include dependency graph for `IDropletProjector.cpp`:



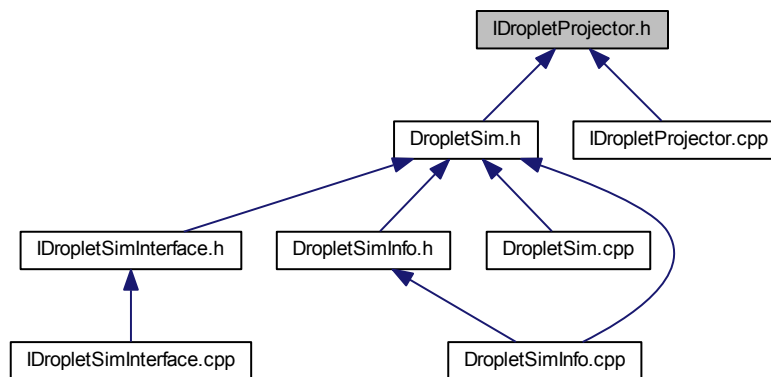
8.18 IDropletProjector.h File Reference

```
#include "DropletDataStructs.h"
#include "DropletSimGlobals.h"
#include <inttypes.h>
#include <EasyBMP.h>
#include <vector>
#include <string>
#include <math.h>
#include <iostream>
```

Include dependency graph for IDropletProjector.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IDropletProjector](#)

Macros

- `#define` [_DROPLET_PROJECTOR](#)

8.18.1 Macro Definition Documentation

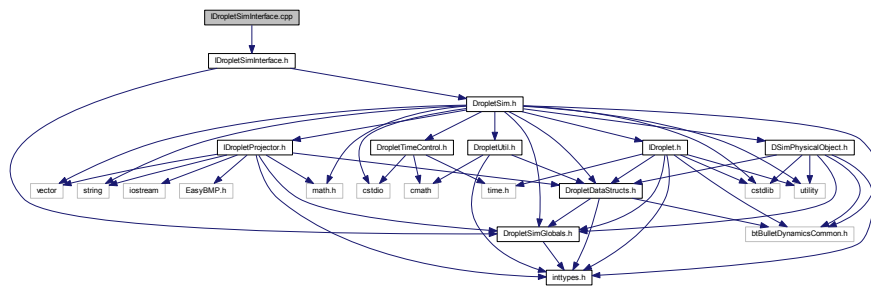
8.18.1.1 #define _DROPLET_PROJECTOR

Definition at line 4 of file IDropletProjector.h.

8.19 IDropletSimInterface.cpp File Reference

```
#include "IDropletSimInterface.h"
```

Include dependency graph for IDropletSimInterface.cpp:

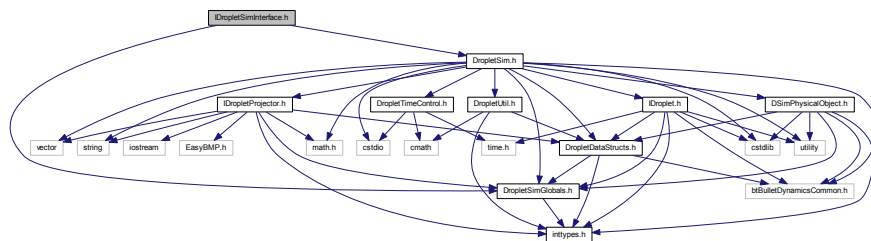


8.20 IDropletSimInterface.h File Reference

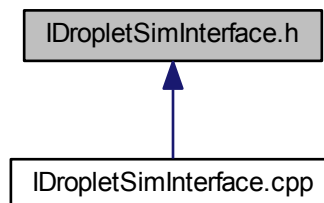
```
#include "DropletSim.h"
```

```
#include "DropletSimGlobals.h"
```

Include dependency graph for IDropletSimInterface.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IDropletSimInterface](#)

Macros

- `#define` [_I_DROPLET_SIM_INTERFACE](#)

Enumerations

- enum [BasicObjectShapes](#) {
 [SPHERE](#), [CUBOID](#), [CYLINDER](#), [CONE](#),
 [PLANE](#) }

8.20.1 Macro Definition Documentation

8.20.1.1 `#define` [_I_DROPLET_SIM_INTERFACE](#)

Definition at line 4 of file IDropletSimInterface.h.

8.20.2 Enumeration Type Documentation

8.20.2.1 enum [BasicObjectShapes](#)

Enumerator

SPHERE

CUBOID

CYLINDER

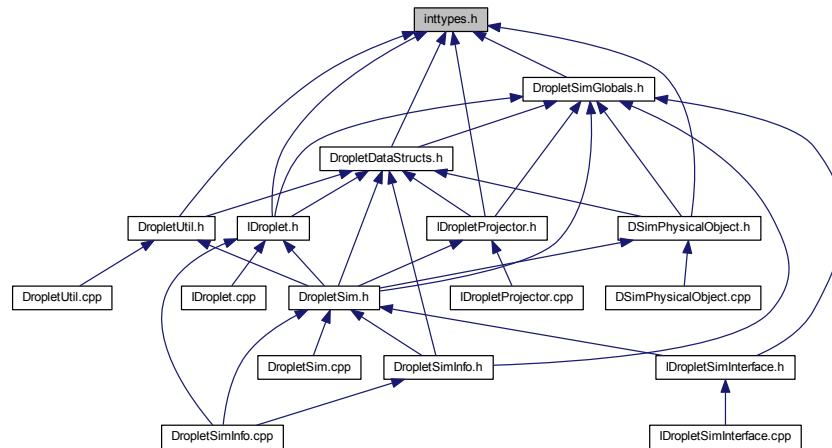
CONE

PLANE

Definition at line 9 of file IDropletSimInterface.h.

8.21 inttypes.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define __INTTYPES_H_`

Typedefs

- typedef short `int16_t`
- typedef int `int32_t`
- typedef long long `int64_t`
- typedef signed char `int8_t`
- typedef `int32_t` `intptr_t`
- typedef unsigned short `uint16_t`
- typedef unsigned int `uint32_t`
- typedef unsigned long long `uint64_t`
- typedef unsigned char `uint8_t`
- typedef `uint32_t` `uintptr_t`

8.21.1 Macro Definition Documentation

8.21.1.1 `#define __INTTYPES_H_`

Definition at line 23 of file `inttypes.h`.

8.21.2 Typedef Documentation

8.21.2.1 typedef short `int16_t`

Definition at line 31 of file `inttypes.h`.

8.21.2.2 `typedef int int32_t`

Definition at line 34 of file `inttypes.h`.

8.21.2.3 `typedef long long int64_t`

Definition at line 37 of file `inttypes.h`.

8.21.2.4 `typedef signed char int8_t`

Definition at line 28 of file `inttypes.h`.

8.21.2.5 `typedef int32_t intptr_t`

Definition at line 40 of file `inttypes.h`.

8.21.2.6 `typedef unsigned short uint16_t`

Definition at line 32 of file `inttypes.h`.

8.21.2.7 `typedef unsigned int uint32_t`

Definition at line 35 of file `inttypes.h`.

8.21.2.8 `typedef unsigned long long uint64_t`

Definition at line 38 of file `inttypes.h`.

8.21.2.9 `typedef unsigned char uint8_t`

Definition at line 29 of file `inttypes.h`.

8.21.2.10 `typedef uint32_t uintptr_t`

Definition at line 41 of file `inttypes.h`.

8.22 main.dox File Reference

Index

- ~DropletSim
 - DropletSim, [23](#)
- ~IDroplet
 - IDroplet, [48](#)
- ~IDropletProjector
 - IDropletProjector, [59](#)
- ~IDropletSimInterface
 - IDropletSimInterface, [63](#)
- ~TrigArray
 - TrigArray, [67](#)
- _DROPLET_SIM
 - DropletSim.h, [73](#)
- _DROPLET_UTIL
 - DropletUtil.h, [84](#)
- _Droplet_Actuator_Data, [13](#)
 - _oscillator, [13](#)
 - bOut, [13](#)
 - currMoveDir, [13](#)
 - currTurnDir, [13](#)
 - gOut, [14](#)
 - moveStepRemaining, [14](#)
 - moveTimeRemaining, [14](#)
 - rOut, [14](#)
 - rotateStepRemaining, [14](#)
 - rotateTimeRemaining, [14](#)
- _Droplet_Communication_Data, [14](#)
 - commChannels, [15](#)
 - sendActive, [15](#)
- _Droplet_Component_Data, [15](#)
 - capacitorPower, [15](#)
 - dropletID, [15](#)
 - leg1Power, [15](#)
 - leg2Power, [16](#)
 - leg3Power, [16](#)
- _Droplet_Localization_Data, [16](#)
 - lastRelPosUpdate, [16](#)
 - movedSinceLastUpdate, [16](#)
 - posX, [16](#)
 - posY, [16](#)
 - posZ, [17](#)
 - rotA, [17](#)
 - rotX, [17](#)
 - rotY, [17](#)
 - rotZ, [17](#)
- _Droplet_Sensor_Data, [17](#)
 - bln, [17](#)
 - gIn, [17](#)
 - rIn, [18](#)
- _Droplet_Sim_Comm_Channel_Data, [18](#)
- inBuf, [18](#)
- inMsgLength, [18](#)
- lastMsgInTimestamp, [18](#)
- lastMsgOutTimestamp, [18](#)
- outBuf, [18](#)
- outMsgLength, [19](#)
- _Droplet_Timing_Data, [19](#)
 - timer, [19](#)
 - trigger, [19](#)
- _I_DROPLET
 - IDroplet.h, [88](#)
- _InitPhysics
 - DSimPhysicalObject, [45](#), [46](#)
 - IDroplet, [48](#)
- _Object_Physics_Data, [19](#)
 - _worldID, [20](#)
 - colShapeIndex, [20](#)
 - friction, [20](#)
 - localInertia, [20](#)
 - mass, [20](#)
- _Simulator_Physics_Data, [20](#)
 - _colShapeIDCounter, [21](#)
 - _dynObjCollisionBM, [21](#)
 - _physicsWorldObjCounter, [21](#)
 - _staticObjCollisionBM, [21](#)
 - broadphase, [21](#)
 - collisionConfig, [21](#)
 - collisionDispatch, [21](#)
 - collisionShapes, [21](#)
 - constraintSolver, [21](#)
 - dynWorld, [21](#)
- __INTTYPES_H_
 - inttypes.h, [92](#)
- _colShapeIDCounter
 - _Simulator_Physics_Data, [21](#)
- _dynObjCollisionBM
 - _Simulator_Physics_Data, [21](#)
- _oscillator
 - _Droplet_Actuator_Data, [13](#)
- _physicsWorldObjCounter
 - _Simulator_Physics_Data, [21](#)
- _staticObjCollisionBM
 - _Simulator_Physics_Data, [21](#)
- _worldID
 - _Object_Physics_Data, [20](#)
- AccessActuatorData
 - IDroplet, [55](#)
- AccessCommData
 - IDroplet, [55](#)

- AccessCompData
 - IDroplet, [56](#)
- AccessPhysicsData
 - IDroplet, [56](#)
- AccessSensorData
 - IDroplet, [56](#)
- AccessTimeData
 - IDroplet, [56](#)
- AddCollisionShape
 - DropletSim, [24](#)
- AddData
 - TrigArray, [67](#)
- AddDroplet
 - DropletSim, [24](#)
- AddPhysicalObject
 - DropletSim, [24](#), [25](#)
- bln
 - _Droplet_Sensor_Data, [17](#)
- bOut
 - _Droplet_Actuator_Data, [13](#)
- BROADCAST_THRESHOLD
 - DropletSimGlobals.h, [75](#)
- BasicObjectShapes
 - IDropletSimInterface.h, [91](#)
- broadphase
 - _Simulator_Physics_Data, [21](#)
- buf
 - IDroplet, [56](#)
- CONE
 - IDropletSimInterface.h, [91](#)
- CUBOID
 - IDropletSimInterface.h, [91](#)
- CYLINDER
 - IDropletSimInterface.h, [91](#)
- cancel_move
 - IDroplet, [48](#)
- cancel_rotate
 - IDroplet, [49](#)
- capacitorPower
 - _Droplet_Component_Data, [15](#)
- check_for_new_messages
 - IDroplet, [49](#)
- check_timer
 - IDroplet, [49](#)
- Cleanup
 - DropletSim, [25](#)
- colShapeIndex
 - _Object_Physics_Data, [20](#)
- collisionConfig
 - _Simulator_Physics_Data, [21](#)
- collisionDispatch
 - _Simulator_Physics_Data, [21](#)
- collisionShapes
 - _Simulator_Physics_Data, [21](#)
- commChannels
 - _Droplet_Communication_Data, [15](#)
- constraintSolver
 - _Simulator_Physics_Data, [21](#)
- CreateDroplet
 - IDropletSimInterface, [63](#)
- CreateFloor
 - DropletSim, [25](#)
- currMoveDir
 - _Droplet_Actuator_Data, [13](#)
 - Droplet_Motion_Direction_Data, [22](#)
- currTurnDir
 - _Droplet_Actuator_Data, [13](#)
 - Droplet_Motion_Direction_Data, [22](#)
- DROPLET_ID_START
 - DropletSimGlobals.h, [75](#)
- DROPLET_NUM_TIMERS
 - DropletSimGlobals.h, [75](#)
- DS_ERROR
 - DropletSimGlobals.h, [76](#)
- DS_FATAL
 - DropletSimGlobals.h, [76](#)
- DS_RESULT
 - DropletSimGlobals.h, [78](#)
- DS_SUCCESS
 - DropletSimGlobals.h, [76](#)
- DS_WARNING
 - DropletSimGlobals.h, [76](#)
- DSimPhysicalObject, [44](#)
 - _InitPhysics, [45](#), [46](#)
 - DSimPhysicalObject, [45](#)
 - DSimPhysicalObject, [45](#)
 - objPhysics, [46](#)
- DSimPhysicalObject.cpp, [85](#)
- DSimPhysicalObject.h, [85](#)
- data_len
 - IDroplet, [56](#)
- dataSet
 - IDropletProjector, [61](#)
- DirInfo
 - DropletSimInfo.h, [80](#)
- doub
 - Ran, [64](#)
- doxygen.dox, [69](#)
- Droplet_Motion_Direction_Data, [22](#)
 - currMoveDir, [22](#)
 - currTurnDir, [22](#)
- droplet_id_type
 - DropletSimGlobals.h, [78](#)
- DropletActuatorData
 - DropletDataStructs.h, [71](#)
- DropletCommChannelData
 - DropletDataStructs.h, [71](#)
- DropletCommData
 - DropletDataStructs.h, [71](#)
- DropletCompData
 - DropletDataStructs.h, [71](#)
- DropletDataStructs.h, [69](#)
 - DropletActuatorData, [71](#)
 - DropletCommChannelData, [71](#)
 - DropletCommData, [71](#)

- DropletCompData, 71
- DropletSensorData, 71
- DropletTimeData, 71
- GPSInfo, 71
- ObjectPhysicsData, 71
- SimPhysicsData, 71
- dropletID
 - _Droplet_Component_Data, 15
- DropletInit
 - IDroplet, 49
- DropletMainLoop
 - IDroplet, 49
- dropletPositions
 - DropletSim, 28
- dropletRelPos
 - DropletSim, 28
- DropletSensorData
 - DropletDataStructs.h, 71
- DropletSim, 22
 - ~DropletSim, 23
 - AddCollisionShape, 24
 - AddDroplet, 24
 - AddPhysicalObject, 24, 25
 - Cleanup, 25
 - CreateFloor, 25
 - dropletPositions, 28
 - dropletRelPos, 28
 - DropletSim, 23
 - DropletSimInfo, 28
 - droplets, 28
 - DropletSim, 23
 - DropletSimInfo, 35
 - firstRun, 28
 - goodRand, 28
 - Init, 26
 - objectPositions, 29
 - physicalObjects, 29
 - projSet, 29
 - projector, 29
 - SetUpProjector, 26, 27
 - SimSetupData, 66
 - Step, 27
 - timer, 29
- DropletSim.cpp, 72
- DropletSim.h, 72
 - _DROPLET_SIM, 73
- DropletSimGlobals.h, 73
 - DROPLET_ID_START, 75
 - DS_ERROR, 76
 - DS_FATAL, 76
 - DS_RESULT, 78
 - DS_SUCCESS, 76
 - DS_WARNING, 76
 - droplet_id_type, 78
 - FLOOR_FRICTION, 76
 - IMPULSE_SCALING, 76
 - IR_BUFFER_SIZE, 76
 - IR_MAX_DATA_SIZE, 76
 - IR_MSG_HEADER, 76
 - MOTOR_POS_SCALING, 76
 - MOVE_OFF, 76
 - MOVE_TIME, 77
 - move_direction, 78
 - msg_order, 78
 - NEWEST_MSG_FIRST, 77
 - NORTH, 77
 - NORTH_EAST, 77
 - NORTH_WEST, 77
 - OLDEST_MSG_FIRST, 77
 - PHYSICS_GRAVITY, 77
 - SAFE_DELETE, 77
 - SOUTH, 77
 - SOUTH_EAST, 77
 - SOUTH_WEST, 78
 - STAGGERED_START, 78
 - STEP_TIME, 78
 - TURN_CLOCKWISE, 78
 - TURN_OFF, 78
 - turn_direction, 79
 - WALK_STEP_TIME, 78
- DropletSimInfo, 29
 - DropletSim, 35
 - DropletSim, 28
 - GetActuationData, 30
 - GetCommData, 30
 - GetCompData, 31
 - GetDropletColors, 31
 - GetDropletPositions, 31
 - GetMotionDirections, 32
 - GetObjectPositions, 32
 - GetPhysData, 32
 - GetRemainingMotionTimes, 32
 - GetSensorColors, 33
 - GetStepRT, 33
 - GetTimeRatio, 33
 - GetTotalDiff, 34
 - GetTotalIRT, 34
 - GetTotalST, 35
- DropletSimInfo.cpp, 79
- DropletSimInfo.h, 79
 - DirInfo, 80
- DropletTimeControl, 36
 - getStepRT, 37
 - getTimeRatio, 37
 - getTotalDiff, 38
 - getTotalIRT, 39
 - getTotalST, 39
 - initTimer, 40
 - printAll, 41
 - printStepRT, 41
 - printTimeRatio, 41
 - printTotalDiff, 42
 - printTotalIRT, 42
 - printTotalST, 42
 - printVars, 43
 - resetTimer, 43

- updateTimer, [43](#), [44](#)
- DropletTimeControl.cpp, [80](#)
 - tvToDouble, [81](#)
- DropletTimeControl.h, [81](#)
- DropletTimeData
 - DropletDataStructs.h, [71](#)
- DropletUtil.cpp, [83](#)
- DropletUtil.h, [83](#)
 - _DROPLET_UTIL, [84](#)
- droplets
 - DropletSim, [28](#)
- dynWorld
 - _Simulator_Physics_Data, [21](#)
- FLOOR_FRICTION
 - DropletSimGlobals.h, [76](#)
- features.dox, [86](#)
- fileDir
 - IDropletProjector, [61](#)
- fileFormat
 - IDropletProjector, [61](#)
- fileName
 - IDropletProjector, [61](#)
- firstRun
 - DropletSim, [28](#)
- floorLength
 - IDropletProjector, [61](#)
- floorWidth
 - IDropletProjector, [61](#)
- friction
 - _Object_Physics_Data, [20](#)
- gIn
 - _Droplet_Sensor_Data, [17](#)
- gOut
 - _Droplet_Actuator_Data, [14](#)
- GPSInfo
 - DropletDataStructs.h, [71](#)
- get_droplet_id
 - IDroplet, [50](#)
- get_rgb
 - IDroplet, [50](#)
- GetActuationData
 - DropletSimInfo, [30](#)
- GetAngle
 - TrigArray, [67](#)
- GetCommData
 - DropletSimInfo, [30](#)
- GetCompData
 - DropletSimInfo, [31](#)
- GetData
 - TrigArray, [67](#)
- GetDistance
 - TrigArray, [67](#)
- GetDropletColors
 - DropletSimInfo, [31](#)
- GetDropletPositions
 - DropletSimInfo, [31](#)
- GetMotionDirections
 - DropletSimInfo, [32](#)
- GetObjectPositions
 - DropletSimInfo, [32](#)
- GetPhysData
 - DropletSimInfo, [32](#)
- GetPixel
 - IDropletProjector, [59](#)
- GetPixels
 - IDropletProjector, [59](#)
- GetRemainingMotionTimes
 - DropletSimInfo, [32](#)
- GetSensorColors
 - DropletSimInfo, [33](#)
- GetStepRT
 - DropletSimInfo, [33](#)
- getStepRT
 - DropletTimeControl, [37](#)
- GetTimeRatio
 - DropletSimInfo, [33](#)
- getTimeRatio
 - DropletTimeControl, [37](#)
- GetTotalDiff
 - DropletSimInfo, [34](#)
- getTotalDiff
 - DropletTimeControl, [38](#)
- GetTotalRT
 - DropletSimInfo, [34](#)
- getTotalRT
 - DropletTimeControl, [39](#)
- GetTotalST
 - DropletSimInfo, [35](#)
- getTotalST
 - DropletTimeControl, [39](#)
- global_rx_buffer
 - IDroplet, [57](#)
- goodRand
 - DropletSim, [28](#)
- IDropletSimInterface.h
 - CONE, [91](#)
 - CUBOID, [91](#)
 - CYLINDER, [91](#)
 - PLANE, [91](#)
 - SPHERE, [91](#)
- IDroplet, [46](#)
 - ~IDroplet, [48](#)
 - _InitPhysics, [48](#)
 - AccessActuatorData, [55](#)
 - AccessCommData, [55](#)
 - AccessCompData, [56](#)
 - AccessPhysicsData, [56](#)
 - AccessSensorData, [56](#)
 - AccessTimeData, [56](#)
 - buf, [56](#)
 - cancel_move, [48](#)
 - cancel_rotate, [49](#)
 - check_for_new_messages, [49](#)
 - check_timer, [49](#)
 - data_len, [56](#)

- DropletInit, 49
- DropletMainLoop, 49
- get_droplet_id, 50
- get_rgb, 50
- global_rx_buffer, 57
- IDroplet, 48
- IDroplet, 48
- ir_send, 50
- is_moving, 50
- is_rotating, 50
- leg1_status, 50
- leg2_status, 51
- leg3_status, 51
- message_time, 57
- move_duration, 51
- move_steps, 51
- msg_return_order, 57
- printed_read_prompt, 57
- rand_byte, 51
- read, 57
- receivers_used, 57
- reset_all_systems, 51
- reset_ir_sensor, 52
- reset_motors, 52
- reset_rgb_led, 53
- reset_rgb_sensor, 53
- reset_timers, 53
- rotate_duration, 54
- rotate_steps, 54
- sender_ID, 57
- set_blue_led, 54
- set_green_led, 54
- set_red_led, 55
- set_rgb, 55
- set_timer, 55
- size, 57
- IDroplet.cpp, 87
- IDroplet.h, 87
 - _I_DROPLET, 88
- IDropletProjector, 57
 - ~IDropletProjector, 59
 - dataSet, 61
 - fileDir, 61
 - fileFormat, 61
 - fileName, 61
 - floorLength, 61
 - floorWidth, 61
 - GetPixel, 59
 - GetPixels, 59
 - IDropletProjector, 59
 - IDropletProjector, 59
 - imgData, 61
 - imgLength, 61
 - imgWidth, 62
 - LoadFile, 60
 - projLength, 62
 - projPixelLength, 62
 - projPixelWidth, 62
 - projWidth, 62
 - SetDirectory, 60
- IDropletProjector.cpp, 88
- IDropletProjector.h, 89
- IDropletSimInterface, 62
 - ~IDropletSimInterface, 63
 - CreateDroplet, 63
 - IDropletSimInterface, 63
 - IDropletSimInterface, 63
 - InitializeSim, 63
 - SetDropletCollisionShape, 63
 - sim, 63
- IDropletSimInterface.cpp, 90
- IDropletSimInterface.h, 90
 - BasicObjectShapes, 91
- IMPULSE_SCALING
 - DropletSimGlobals.h, 76
- IR_BUFFER_SIZE
 - DropletSimGlobals.h, 76
- IR_MAX_DATA_SIZE
 - DropletSimGlobals.h, 76
- IR_MSG_HEADER
 - DropletSimGlobals.h, 76
- IR_RX_STATUS_BUSY
 - DropletSimGlobals.h, 76
- imgData
 - IDropletProjector, 61
- imgLength
 - IDropletProjector, 61
- imgWidth
 - IDropletProjector, 62
- inBuf
 - _Droplet_Sim_Comm_Channel_Data, 18
- inMsgLength
 - _Droplet_Sim_Comm_Channel_Data, 18
- Init
 - DropletSim, 26
- initTimer
 - DropletTimeControl, 40
- InitializeSim
 - IDropletSimInterface, 63
- int16_t
 - inttypes.h, 92
- int32_t
 - inttypes.h, 92
- int64
 - Ran, 64
- int64_t
 - inttypes.h, 93
- int8_t
 - inttypes.h, 93
- intptr_t
 - inttypes.h, 93
- inttypes.h, 92
 - __INTTYPES_H_, 92
 - int16_t, 92
 - int32_t, 92
 - int64_t, 93

- int8_t, [93](#)
- intptr_t, [93](#)
- uint16_t, [93](#)
- uint32_t, [93](#)
- uint64_t, [93](#)
- uint8_t, [93](#)
- uintptr_t, [93](#)
- ir_send
 - IDroplet, [50](#)
- is_moving
 - IDroplet, [50](#)
- is_rotating
 - IDroplet, [50](#)
- lastMsgInTimestamp
 - _Droplet_Sim_Comm_Channel_Data, [18](#)
- lastMsgOutTimestamp
 - _Droplet_Sim_Comm_Channel_Data, [18](#)
- lastRelPosUpdate
 - _Droplet_Localization_Data, [16](#)
- leg1_status
 - IDroplet, [50](#)
- leg1Power
 - _Droplet_Component_Data, [15](#)
- leg2_status
 - IDroplet, [51](#)
- leg2Power
 - _Droplet_Component_Data, [16](#)
- leg3_status
 - IDroplet, [51](#)
- leg3Power
 - _Droplet_Component_Data, [16](#)
- LoadFile
 - IDropletProjector, [60](#)
- localInertia
 - _Object_Physics_Data, [20](#)
- MOTOR_POS_SCALING
 - DropletSimGlobals.h, [76](#)
- MOVE_OFF
 - DropletSimGlobals.h, [76](#)
- MOVE_TIME
 - DropletSimGlobals.h, [77](#)
- main.dox, [93](#)
- mass
 - _Object_Physics_Data, [20](#)
- message_time
 - IDroplet, [57](#)
- move_direction
 - DropletSimGlobals.h, [78](#)
- move_duration
 - IDroplet, [51](#)
- move_steps
 - IDroplet, [51](#)
- moveStepRemaining
 - _Droplet_Actuator_Data, [14](#)
- moveTimeRemaining
 - _Droplet_Actuator_Data, [14](#)
- movedSinceLastUpdate
 - _Droplet_Localization_Data, [16](#)
- msg_order
 - DropletSimGlobals.h, [78](#)
- msg_return_order
 - IDroplet, [57](#)
- NEWEST_MSG_FIRST
 - DropletSimGlobals.h, [77](#)
- NORTH
 - DropletSimGlobals.h, [77](#)
- NORTH_EAST
 - DropletSimGlobals.h, [77](#)
- NORTH_WEST
 - DropletSimGlobals.h, [77](#)
- OLDEST_MSG_FIRST
 - DropletSimGlobals.h, [77](#)
- objPhysics
 - DSimPhysicalObject, [46](#)
- ObjectPhysicsData
 - DropletDataStructs.h, [71](#)
- objectPositions
 - DropletSim, [29](#)
- outBuf
 - _Droplet_Sim_Comm_Channel_Data, [18](#)
- outMsgLength
 - _Droplet_Sim_Comm_Channel_Data, [19](#)
- PLANE
 - IDropletSimInterface.h, [91](#)
- PHYSICS_GRAVITY
 - DropletSimGlobals.h, [77](#)
- physicalObjects
 - DropletSim, [29](#)
- posX
 - _Droplet_Localization_Data, [16](#)
- posY
 - _Droplet_Localization_Data, [16](#)
- posZ
 - _Droplet_Localization_Data, [17](#)
- printAll
 - DropletTimeControl, [41](#)
- printStepRT
 - DropletTimeControl, [41](#)
- printTimeRatio
 - DropletTimeControl, [41](#)
- printTotalDiff
 - DropletTimeControl, [42](#)
- printTotalRT
 - DropletTimeControl, [42](#)
- printTotalST
 - DropletTimeControl, [42](#)
- printVars
 - DropletTimeControl, [43](#)
- printed_read_prompt
 - IDroplet, [57](#)
- projLength
 - IDropletProjector, [62](#)
- projPixelLength

- IDropletProjector, 62
- projPixelWidth
 - IDropletProjector, 62
- projSet
 - DropletSim, 29
- projWidth
 - IDropletProjector, 62
- projector
 - DropletSim, 29
- rIn
 - _Droplet_Sensor_Data, 18
- rOut
 - _Droplet_Actuator_Data, 14
- Ran, 63
 - doub, 64
 - int64, 64
 - Ran, 64
 - u, 65
 - v, 65
 - w, 65
- rand_byte
 - IDroplet, 51
- read
 - IDroplet, 57
- receivers_used
 - IDroplet, 57
- RemoveData
 - TrigArray, 67
- reset_all_systems
 - IDroplet, 51
- reset_ir_sensor
 - IDroplet, 52
- reset_motors
 - IDroplet, 52
- reset_rgb_led
 - IDroplet, 53
- reset_rgb_sensor
 - IDroplet, 53
- reset_timers
 - IDroplet, 53
- resetTimer
 - DropletTimeControl, 43
- rotA
 - _Droplet_Localization_Data, 17
- rotX
 - _Droplet_Localization_Data, 17
- rotY
 - _Droplet_Localization_Data, 17
- rotZ
 - _Droplet_Localization_Data, 17
- rotate_duration
 - IDroplet, 54
- rotate_steps
 - IDroplet, 54
- rotateStepRemaining
 - _Droplet_Actuator_Data, 14
- rotateTimeRemaining
 - _Droplet_Actuator_Data, 14
- SPHERE
 - IDropletSimInterface.h, 91
- SAFE_DELETE
 - DropletSimGlobals.h, 77
- SOUTH
 - DropletSimGlobals.h, 77
- SOUTH_EAST
 - DropletSimGlobals.h, 77
- SOUTH_WEST
 - DropletSimGlobals.h, 78
- STAGGERED_START
 - DropletSimGlobals.h, 78
- STEP_TIME
 - DropletSimGlobals.h, 78
- sendActive
 - _Droplet_Communication_Data, 15
- sender_ID
 - IDroplet, 57
- set_blue_led
 - IDroplet, 54
- set_green_led
 - IDroplet, 54
- set_red_led
 - IDroplet, 55
- set_rgb
 - IDroplet, 55
- set_timer
 - IDroplet, 55
- SetDirectory
 - IDropletProjector, 60
- SetDropletCollisionShape
 - IDropletSimInterface, 63
- SetUpProjector
 - DropletSim, 26, 27
- sim
 - IDropletSimInterface, 63
- SimPhysicsData
 - DropletDataStructs.h, 71
- SimSetupData, 65
 - DropletSim, 66
 - SimSetupData, 66
 - SimSetupData, 66
- size
 - IDroplet, 57
- Step
 - DropletSim, 27
- TURN_CLOCKWISE
 - DropletSimGlobals.h, 78
- TURN_OFF
 - DropletSimGlobals.h, 78
- timer
 - _Droplet_Timing_Data, 19
 - DropletSim, 29
- TrigArray, 66
 - ~TrigArray, 67
 - AddData, 67
 - GetAngle, 67
 - GetData, 67

- GetDistance, [67](#)
- RemoveData, [67](#)
- TrigArray, [67](#)
- TrigArray, [67](#)
- trigger
 - _Droplet_Timing_Data, [19](#)
- turn_direction
 - DropletSimGlobals.h, [79](#)
- tvToDouble
 - DropletTimeControl.cpp, [81](#)
- u
 - Ran, [65](#)
- uint16_t
 - inttypes.h, [93](#)
- uint32_t
 - inttypes.h, [93](#)
- uint64_t
 - inttypes.h, [93](#)
- uint8_t
 - inttypes.h, [93](#)
- uintptr_t
 - inttypes.h, [93](#)
- updateTimer
 - DropletTimeControl, [43](#), [44](#)
- v
 - Ran, [65](#)
- w
 - Ran, [65](#)
- WALK_STEP_TIME
 - DropletSimGlobals.h, [78](#)