

ohCaptain

Generated by Doxygen 1.8.11

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	oCpt Namespace Reference	9
6	Class Documentation	11
6.1	oCpt::CoveragePathTask Class Reference	11
6.1.1	Detailed Description	11
6.1.2	Constructor & Destructor Documentation	12
6.1.2.1	CoveragePathTask()	12
6.1.2.2	~CoveragePathTask()	12
6.2	oCpt::DredgeTask Class Reference	12
6.2.1	Detailed Description	12
6.2.2	Constructor & Destructor Documentation	13
6.2.2.1	DredgeTask()	13
6.2.2.2	~DredgeTask()	13

6.3	oCpt::FollowTask Class Reference	13
6.3.1	Detailed Description	13
6.3.2	Constructor & Destructor Documentation	14
6.3.2.1	FollowTask()	14
6.3.2.2	~FollowTask()	14
6.4	oCpt::iTask Class Reference	14
6.4.1	Detailed Description	15
6.4.2	Member Typedef Documentation	15
6.4.2.1	ptr	15
6.4.3	Member Enumeration Documentation	15
6.4.3.1	TypeOf	15
6.4.4	Constructor & Destructor Documentation	15
6.4.4.1	iTask()	15
6.4.4.2	~iTask()	15
6.4.5	Member Function Documentation	16
6.4.5.1	start()=0	16
6.4.5.2	status()=0	16
6.4.5.3	stop()=0	16
6.5	oCpt::LogTask Class Reference	16
6.5.1	Detailed Description	17
6.5.2	Constructor & Destructor Documentation	17
6.5.2.1	LogTask()	17
6.5.2.2	~LogTask()	17
6.6	oCpt::PathTask Class Reference	17
6.6.1	Detailed Description	18
6.6.2	Constructor & Destructor Documentation	18
6.6.2.1	PathTask()	18
6.6.2.2	~PathTask()	18
6.7	oCpt::RouteTask Class Reference	18
6.7.1	Detailed Description	19

6.7.2	Constructor & Destructor Documentation	19
6.7.2.1	RouteTask()	19
6.7.2.2	~RouteTask()	19
6.8	oCpt::iTask::Status Class Reference	19
6.8.1	Member Typedef Documentation	20
6.8.1.1	ptr	20
6.8.2	Constructor & Destructor Documentation	20
6.8.2.1	Status()	20
6.8.2.2	~Status()	20
6.8.3	Member Function Documentation	20
6.8.3.1	progress()	20
6.8.3.2	running()	20
6.8.3.3	successful()	20
6.9	oCpt::Task Class Reference	21
6.9.1	Detailed Description	21
6.9.2	Constructor & Destructor Documentation	21
6.9.2.1	Task()	21
6.9.2.2	~Task()	22
6.9.3	Member Function Documentation	22
6.9.3.1	start()	22
6.9.3.2	status()	22
6.9.3.3	stop()	22
6.9.4	Member Data Documentation	22
6.9.4.1	_status	22
6.9.4.2	_typeof	22
6.10	oCpt::WorkTask Class Reference	23
6.10.1	Detailed Description	23
6.10.2	Constructor & Destructor Documentation	23
6.10.2.1	WorkTask()	23
6.10.2.2	~WorkTask()	23
7	File Documentation	25
7.1	include/Task.h File Reference	25
7.2	src/Task.cpp File Reference	25

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

oCpt	9
--------------------------------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

oCpt::iTask	14
oCpt::Task	21
oCpt::RouteTask	18
oCpt::CoveragePathTask	11
oCpt::FollowTask	13
oCpt::PathTask	17
oCpt::WorkTask	23
oCpt::DredgeTask	12
oCpt::LogTask	16
oCpt::iTask::Status	19

Chapter 3

Class Index

3.1 Class List

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

oCpt::CoveragePathTask	
An object representing a coverage path task	11
oCpt::DredgeTask	
An Object representing a dredging task	12
oCpt::FollowTask	
An object representing a follow the target task	13
oCpt::iTask	
Task interface, all tasks need to adhere to this structure	14
oCpt::LogTask	
An Object representing a data logging task	16
oCpt::PathTask	
An object representing a normal A to B type of path planning	17
oCpt::RouteTask	18
oCpt::iTask::Status	19
oCpt::Task	21
oCpt::WorkTask	23

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/ Task.h	25
src/ Task.cpp	25

Chapter 5

Namespace Documentation

5.1 oCpt Namespace Reference

Classes

- class [CoveragePathTask](#)
An object representing a coverage path task.
- class [DredgeTask](#)
An Object representing a dredging task.
- class [FollowTask](#)
An object representing a follow the target task.
- class [iTask](#)
[Task](#) interface, all tasks need to adhere to this structure.
- class [LogTask](#)
An Object representing a data logging task.
- class [PathTask](#)
An object representing a normal A to B type of path planning.
- class [RouteTask](#)
- class [Task](#)
- class [WorkTask](#)

Chapter 6

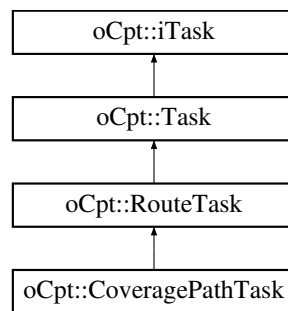
Class Documentation

6.1 oCpt::CoveragePathTask Class Reference

An object representing a coverage path task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::CoveragePathTask:



Public Member Functions

- [CoveragePathTask \(\)](#)
- virtual [~CoveragePathTask \(\)](#)

Additional Inherited Members

6.1.1 Detailed Description

An object representing a coverage path task.

All these types of tasks need a robot to cover a complete region in order to perform their tasks. According to {cao_region_1988} such a mobile robot should use the following criteria, for a region filling operation:

1. The mobile robot must move through an entire area, i.e., the overall travel must cover a whole region.
2. The mobile robot must fill the region without overlapping paths.
3. Continuous and sequential operations without any repetition of paths is required of the robot.
4. The robot must avoid all obstacles in a region.
5. Simple motion trajectories (e.g., straight lines or circles) should be used for simplicity in control.
6. An "optimal" path is desired under the available conditions. It is not always possible to satisfy all these criteria for a complex environment. Sometimes a priority consideration is required.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 oCpt::CoveragePathTask::CoveragePathTask ()

Constructor of the interface

Returns

6.1.2.2 oCpt::CoveragePathTask::~CoveragePathTask () [virtual]

The destructor

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

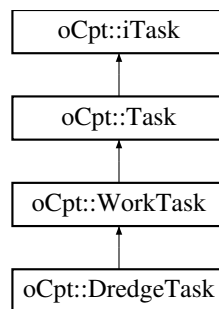
- [include/Task.h](#)
- [src/Task.cpp](#)

6.2 oCpt::DredgeTask Class Reference

An Object representing a dredging task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::DredgeTask:



Public Member Functions

- [DredgeTask \(\)](#)
- virtual [~DredgeTask \(\)](#)

Additional Inherited Members

6.2.1 Detailed Description

An Object representing a dredging task.

All these types tasks make use of an actuator and sensors to perform dredging tasks

6.2.2 Constructor & Destructor Documentation

6.2.2.1 oCpt::DredgeTask::DredgeTask ()

Constructor of the interface

Returns

6.2.2.2 oCpt::DredgeTask::~~DredgeTask () [virtual]

The destructor

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

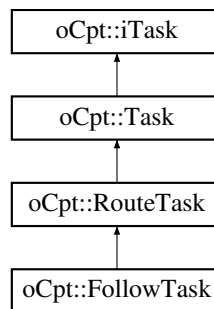
- [include/Task.h](#)
- [src/Task.cpp](#)

6.3 oCpt::FollowTask Class Reference

An object representing a follow the target task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::FollowTask:



Public Member Functions

- [FollowTask \(\)](#)
- virtual [~FollowTask \(\)](#)

Additional Inherited Members

6.3.1 Detailed Description

An object representing a follow the target task.

All these types of tasks need to follow a (moving) target

6.3.2 Constructor & Destructor Documentation

6.3.2.1 oCpt::FollowTask::FollowTask ()

Constructor of the interface

Returns

6.3.2.2 oCpt::FollowTask::~~FollowTask () [virtual]

The destructor

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

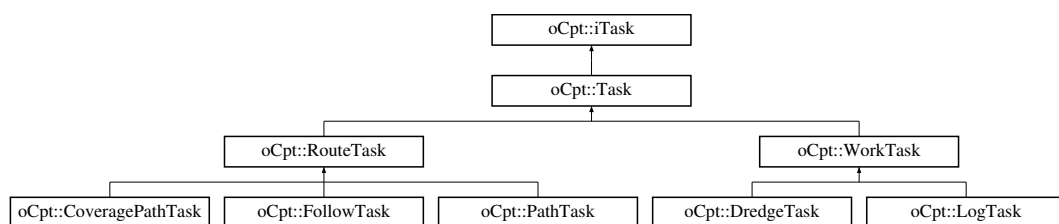
- include/[Task.h](#)
- src/[Task.cpp](#)

6.4 oCpt::iTask Class Reference

[Task](#) interface, all tasks need to adhere to this structure.

```
#include <Task.h>
```

Inheritance diagram for oCpt::iTask:



Classes

- class [Status](#)

Public Types

- enum [TypeOf](#) { [Route](#) = 1, [Work](#) = 2 }
- typedef boost::shared_ptr< [iTask](#) > [ptr](#)
Boost shared_ptr to a task.

Public Member Functions

- [iTask\(\)](#)
- virtual [~iTask\(\)](#)
- virtual void [start\(\)](#)=0
- virtual [iTask::Status::ptr status\(\)](#)=0
- virtual void [stop\(\)](#)=0

6.4.1 Detailed Description

[Task](#) interface, all tasks need to adhere to this structure.

This interface make sure that all task adheres to the same runtime rules and enable run-time polymorphism

6.4.2 Member Typedef Documentation

6.4.2.1 typedef boost::shared_ptr<iTask> oCpt::iTask::ptr

Boost shared_ptr to a task.

6.4.3 Member Enumeration Documentation

6.4.3.1 enum oCpt::iTask::TypeOf

Enumeration indicating which type of task the object is

Enumerator

Route

Work

6.4.4 Constructor & Destructor Documentation

6.4.4.1 oCpt::iTask::iTask()

Constructor of the interface

Returns

6.4.4.2 oCpt::iTask::~~iTask() [virtual]

Deconstructor of the interface

6.4.5 Member Function Documentation

6.4.5.1 `virtual void oCpt::iTask::start () [pure virtual]`

The start command for a task

Implemented in [oCpt::Task](#).

6.4.5.2 `virtual iTask::Status::ptr oCpt::iTask::status () [pure virtual]`

Retrieves the [Status](#) of a task

Returns

Boost shared_ptr of the task status

Implemented in [oCpt::Task](#).

6.4.5.3 `virtual void oCpt::iTask::stop () [pure virtual]`

The stop command for a task

Implemented in [oCpt::Task](#).

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

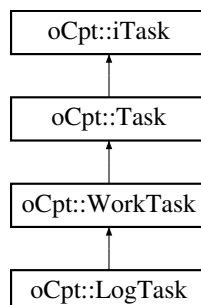
- [include/Task.h](#)
- [src/Task.cpp](#)

6.5 oCpt::LogTask Class Reference

An Object representing a data logging task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::LogTask:



Public Member Functions

- [LogTask](#) ()
- virtual [~LogTask](#) ()

Additional Inherited Members

6.5.1 Detailed Description

An Object representing a data logging task.

All these types of tasks make use of a sensor to record and log

6.5.2 Constructor & Destructor Documentation

6.5.2.1 oCpt::LogTask::LogTask ()

Constructor of the interface

Returns

6.5.2.2 oCpt::LogTask::~~LogTask () [virtual]

The deconstructor

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

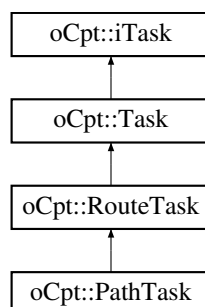
- include/[Task.h](#)
- src/[Task.cpp](#)

6.6 oCpt::PathTask Class Reference

An object representing a normal A to B type of path planning.

```
#include <Task.h>
```

Inheritance diagram for oCpt::PathTask:



Public Member Functions

- [PathTask](#) ()
- virtual [~PathTask](#) ()

Additional Inherited Members

6.6.1 Detailed Description

An object representing a normal A to B type of path planning.

All these types of tasks need to plan an optimum route between A and B, either in time, energy consumption or

6.6.2 Constructor & Destructor Documentation

6.6.2.1 `oCpt::PathTask::PathTask ()`

Constructor of the interface

Returns

6.6.2.2 `oCpt::PathTask::~~PathTask ()` [virtual]

The deconstructor

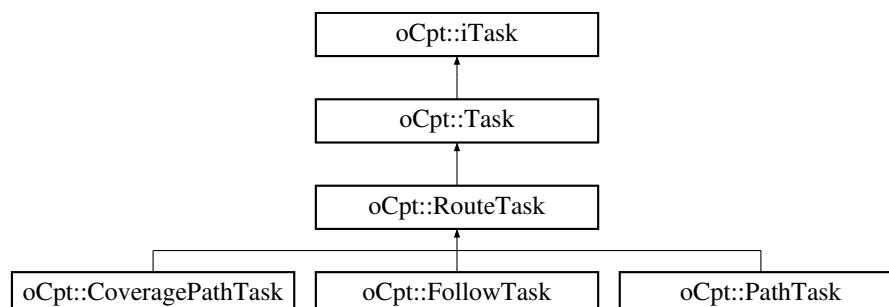
The documentation for this class was generated from the following files:

- [include/Task.h](#)
- [src/Task.cpp](#)

6.7 oCpt::RouteTask Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::RouteTask:



Public Member Functions

- [RouteTask](#) ()
- virtual [~RouteTask](#) ()

Additional Inherited Members

6.7.1 Detailed Description

An object representing route related tasks

6.7.2 Constructor & Destructor Documentation

6.7.2.1 oCpt::RouteTask::RouteTask ()

Constructor of the interface

Returns

6.7.2.2 oCpt::RouteTask::~~RouteTask () [virtual]

The destructor

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

- include/[Task.h](#)
- src/[Task.cpp](#)

6.8 oCpt::iTask::Status Class Reference

```
#include <Task.h>
```

Public Types

- typedef boost::shared_ptr< [iTask::Status](#) > ptr
Boost shared_ptr to the task status.

Public Member Functions

- [Status](#) ()
- virtual [~Status](#) ()
- double [progress](#) ()
- bool [running](#) ()
- bool [successful](#) ()

6.8.1 Member Typedef Documentation

6.8.1.1 `typedef boost::shared_ptr<iTask::Status> oCpt::iTask::Status::ptr`

Boost `shared_ptr` to the task status.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 `oCpt::iTask::Status::Status ()`

Constructor of the [iTask](#)

Returns

6.8.2.2 `oCpt::iTask::Status::~~Status () [virtual]`

Deconstructor

6.8.3 Member Function Documentation

6.8.3.1 `double oCpt::iTask::Status::progress ()`

Show the progress of the task

Returns

double between 0..1

6.8.3.2 `bool oCpt::iTask::Status::running ()`

Returns the running state of the task

Returns

bool where running is true

6.8.3.3 `bool oCpt::iTask::Status::successful ()`

Returns if the task was completed succesfully

Returns

bool where a succesfully completed task is true, task in progress or failed are false

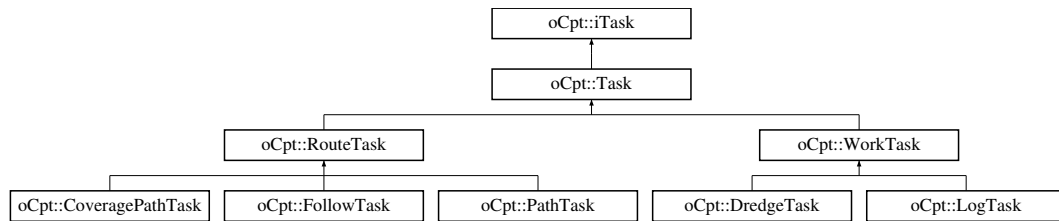
The documentation for this class was generated from the following files:

- [include/Task.h](#)
- [src/Task.cpp](#)

6.9 oCpt::Task Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::Task:



Public Member Functions

- [Task](#) ()
- virtual [~Task](#) ()
- virtual void [start](#) ()
- virtual [iTask::Status::ptr status](#) ()
- virtual void [stop](#) ()

Protected Attributes

- [iTask::Status::ptr _status](#)
a boost share_ptr to the status of a task
- [TypeOf_typeof](#)
Indicating the type of a task.

Additional Inherited Members

6.9.1 Detailed Description

The Base [Task](#) class

6.9.2 Constructor & Destructor Documentation

6.9.2.1 oCpt::Task::Task ()

The contructor

Returns

6.9.2.2 oCpt::Task::~~Task () [virtual]

The destructor

6.9.3 Member Function Documentation

6.9.3.1 void oCpt::Task::start () [virtual]

The start command for a task

Implements [oCpt::iTask](#).

6.9.3.2 iTask::Status::ptr oCpt::Task::status () [virtual]

Retrieves the Status of a task

Returns

Boost shared_ptr of the task status

Implements [oCpt::iTask](#).

6.9.3.3 void oCpt::Task::stop () [virtual]

The stop command for a task

Implements [oCpt::iTask](#).

6.9.4 Member Data Documentation

6.9.4.1 iTask::Status::ptr oCpt::Task::_status [protected]

a boost share_ptr to the status of a task

6.9.4.2 TypeOf oCpt::Task::_typeof [protected]

Indicating the type of a task.

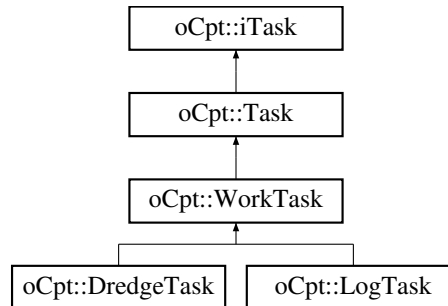
The documentation for this class was generated from the following files:

- include/[Task.h](#)
- src/[Task.cpp](#)

6.10 oCpt::WorkTask Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::WorkTask:



Public Member Functions

- [WorkTask\(\)](#)
- virtual [~WorkTask\(\)](#)

Additional Inherited Members

6.10.1 Detailed Description

An object representing work related tasks

6.10.2 Constructor & Destructor Documentation

6.10.2.1 oCpt::WorkTask::WorkTask()

Constructor of the interface

Returns

6.10.2.2 oCpt::WorkTask::~~WorkTask() [virtual]

The deconstructor

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

- include/[Task.h](#)
- src/[Task.cpp](#)

Chapter 7

File Documentation

7.1 include/Task.h File Reference

```
#include <boost/shared_ptr.hpp>
```

Classes

- class [oCpt::iTask](#)
Task interface, all tasks need to adhere to this structure.
- class [oCpt::iTask::Status](#)
- class [oCpt::Task](#)
- class [oCpt::RouteTask](#)
- class [oCpt::WorkTask](#)
- class [oCpt::CoveragePathTask](#)
An object representing a coverage path task.
- class [oCpt::FollowTask](#)
An object representing a follow the target task.
- class [oCpt::PathTask](#)
An object representing a normal A to B type of path planning.
- class [oCpt::LogTask](#)
An Object representing a data logging task.
- class [oCpt::DredgeTask](#)
An Object representing a dredging task.

Namespaces

- [oCpt](#)

7.2 src/Task.cpp File Reference

```
#include "../include/Task.h"
```

Namespaces

- [oCpt](#)

