

Vrep-ROS Library

Generated by Doxygen 1.8.6

Sat Jan 17 2015 12:50:36

Contents

1	README	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	MyPublisher< T > Class Template Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	7
4.1.2.1	MyPublisher	7
4.1.3	Member Function Documentation	8
4.1.3.1	GetTopicName	8
4.1.3.2	operator<<	8
4.1.3.3	Push	8
4.2	MyROS Class Reference	8
4.2.1	Detailed Description	9
4.2.2	Constructor & Destructor Documentation	9
4.2.2.1	MyROS	9
4.2.3	Member Function Documentation	9
4.2.3.1	GetNodeHandlePointer	9
4.2.3.2	Run	9
4.3	MySubscriber< T > Class Template Reference	9
4.3.1	Detailed Description	10
4.3.2	Constructor & Destructor Documentation	10
4.3.2.1	MySubscriber	10
4.3.3	Member Function Documentation	10
4.3.3.1	callback	10
4.3.3.2	Get	10
4.3.3.3	Get	11

4.3.3.4	IsNew	11
4.4	MyVrep Class Reference	11
4.4.1	Constructor & Destructor Documentation	12
4.4.1.1	MyVrep	12
4.4.2	Member Function Documentation	12
4.4.2.1	EnablePublisher	12
4.4.2.2	EnableSubscriber	12
4.4.2.3	GetObjectHandle	13
4.4.2.4	GetObjectPose	13
4.4.2.5	GetTime	13
4.4.2.6	IsRun	14
4.4.2.7	LoadModel	15
4.4.2.8	RemoveModel	15
4.4.2.9	RemoveModel	15
4.4.2.10	SetJointTargetPosition	15
4.4.2.11	SetJointTargetPosition	15
4.4.2.12	SetJointTargetVelocity	16
4.4.2.13	SetJointTargetVelocity	16
5	File Documentation	17
5.1	myros.h File Reference	17
5.1.1	Detailed Description	17
5.2	myvrep.h File Reference	18
5.2.1	Detailed Description	18
	Index	19

Chapter 1

README

ROS と Vrep のライブラリです.

使い方の詳細は [Doxygen](#) を参照してください.

Chapter 2

Class Index

2.1 Class List

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

MyPublisher< T >	
Publisher オブジェクト	7
MyROS	
ROS の用意をする (いろいろする前に, まずこいつを呼んでください)	8
MySubscriber< T >	
Subscriber オブジェクト	9
MyVrep	11

Chapter 3

File Index

3.1 File List

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

myros.h	ROS の本体及びPublisherとSubscriberのライブラリ	17
myvrep.h	Vrep と ROS ノードをつなぐためのライブラリ	18

Chapter 4

Class Documentation

4.1 MyPublisher< T > Class Template Reference

Publisher オブジェクト

```
#include <myros.h>
```

Public Member Functions

- [MyPublisher](#) (ros::NodeHandle *node, std::string topicName, int queue_size=1)
Publisher を準備する
- template<class U >
void [Push](#) (U value)
値を publish する
- std::string [GetTopicName](#) ()
コンストラクタ時に設定した topic の名前を取得する (Subscribe にでもどうぞ)
- template<class U >
void [operator<<](#) (U value)
.Push する

4.1.1 Detailed Description

```
template<class T>class MyPublisher< T >
```

Publisher オブジェクト

Template Parameters

<i>T</i>	publish する型の名前
----------	----------------

4.1.2 Constructor & Destructor Documentation

4.1.2.1 `template<class T > MyPublisher< T >::MyPublisher (ros::NodeHandle * node, std::string topicName, int queue_size = 1) [inline]`

Publisher を準備する

Parameters

<i>node</i>	準備する node の名前
<i>topicName</i>	topic の名前
<i>queue_size</i>	queue のサイズ

4.1.3 Member Function Documentation

4.1.3.1 `template<class T> std::string MyPublisher<T>::GetTopicName ()` `[inline]`

コンストラクタ時に設定した topic の名前を取得する (Subscribe にでもどうぞ)

Returns

topic の名前

4.1.3.2 `template<class T> template<class U> void MyPublisher<T>::operator<< (U value)` `[inline]`

.Push する

Template Parameters

<i>U</i>	push する型
----------	----------

Parameters

<i>value</i>	push する値
--------------	----------

4.1.3.3 `template<class T> template<class U> void MyPublisher<T>::Push (U value)` `[inline]`

値を publish する

Template Parameters

<i>U</i>	publish する型 (T.data に変換できなければコンパイルエラー)
----------	--

Parameters

<i>value</i>	publish する値
--------------	-------------

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

- [myros.h](#)

4.2 MyROS Class Reference

ROS の用意をする (いろいろする前に, まずこいつを呼んでください)

```
#include <myros.h>
```

Public Member Functions

- [MyROS](#) (int argc, char **argv, const std::string &nodeName, const int loopRate=10)
ROS の node を制作する
- bool [Run](#) ()

`While(Run())`で使用する. `Subscriber` の `callback` を呼んだり, `sleep()` を呼んだりする

- `ros::NodeHandle * GetNodeHandlePointer ()`
`nodeHandle` のポインタを取得する

4.2.1 Detailed Description

ROS の用意をする (いろいろする前に, まずこいつを呼んでください)

4.2.2 Constructor & Destructor Documentation

4.2.2.1 `MyROS::MyROS (int argc, char ** argv, const std::string & nodeName, const int loopRate = 10)` `[inline]`

ROS の node を制作する

Parameters

<code>argc</code>	コマンドライン引数の数
<code>argv</code>	コマンドライン引き数
<code>nodeName</code>	node の名前
<code>loopRate</code>	sleep させる時間

4.2.3 Member Function Documentation

4.2.3.1 `ros::NodeHandle* MyROS::GetNodeHandlePointer ()` `[inline]`

`nodeHandle` のポインタを取得する

Returns

`nodeHandle` のポインタ

4.2.3.2 `bool MyROS::Run ()` `[inline]`

`While(Run())`で使用する. `Subscriber` の `callback` を呼んだり, `sleep()` を呼んだりする

Returns

ROS を 続行するかどうか

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

- [myros.h](#)

4.3 MySubscriber< T > Class Template Reference

Subscriber オブジェクト

```
#include <myros.h>
```

Public Member Functions

- void [callback](#) (const typename T::ConstPtr &msg)

callback 関数. 外向けのものではない

- **MySubscriber** (ros::NodeHandle *node, const std::string &topicName, const int queue_size=1)
subscriber を制作する .
- template<class U >
U **Get** ()
取得している 値を返す
- T **Get** ()
取得している 値を返す
- bool **IsNew** ()
前回の *Get()* から 更新があったかどうか
- template<class U >
void **operator**>> (U &dst)

4.3.1 Detailed Description

template<class T>class MySubscriber< T >

Subscriber オブジェクト

Template Parameters

<i>T</i>	Subscribe する肩
----------	---------------

4.3.2 Constructor & Destructor Documentation

4.3.2.1 template<class T> **MySubscriber**< T >::MySubscriber (ros::NodeHandle * *node*, const std::string & *topicName*, const int *queue_size* = 1) [inline]

subscriber を制作する .

Parameters

<i>node</i>	subscribe する node
<i>topicName</i>	topic の名前
<i>queue_size</i>	queue のサイズ

4.3.3 Member Function Documentation

4.3.3.1 template<class T> void **MySubscriber**< T >::callback (const typename T::ConstPtr & *msg*) [inline]

callback 関数. 外向けのものではない

Parameters

<i>msg</i>	callback されてくる message
------------	------------------------

4.3.3.2 template<class T> template<class U > U **MySubscriber**< T >::Get () [inline]

取得している 値を返す

Template Parameters

U	変換してほしい型の名前 (不可能な場合, コンパイルエラー)
---	--------------------------------

Returns

取得している値

4.3.3.3 `template<class T> T MySubscriber<T>::Get () [inline]`

取得している値を返す

Returns

取得している値 (内部 `data` ではなく, まるまる全てを返すので注意)

4.3.3.4 `template<class T> bool MySubscriber<T>::IsNew () [inline]`

前回の `Get()` から更新があったかどうか

Return values

<i>true</i>	更新があった
<i>false</i>	更新がなかった

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

- [myros.h](#)

4.4 MyVrep Class Reference

Public Member Functions

- [MyVrep](#) (ros::NodeHandle *node)
コンストラクタ, 名前空間と *infocallback* のスタンバイ
- float [GetTime](#) ()
取得した時間を返す
- bool [IsRun](#) ()
Vrep がいま動いているかどうかを返す
- int [GetObjectHandle](#) (std::string objectName)
Object の *Handle* を取得する
- void [LoadModel](#) (std::string filename)
Model を読み込む
- void [RemoveModel](#) (int handle)
Model を除去する
- void [RemoveModel](#) (std::string modelName)
Model を除去する
- void [Start](#) ()
Vrep を *Run* させる
- void [Stop](#) ()
Vrep を止める
- bool [EnablePublisher](#) (std::string topicName, int queueSize, int streamCmd, int auxInt1, int auxInt2, std::string auxString)
Vrep 上のオブジェクトに対し, *Publisher* を要求する

- bool [EnableSubscriber](#) (std::string topicName, int queueSize, int streamCmd, int auxInt1, int auxInt2, std::string auxString)
Subscriber を要求する
- bool [SetJointTargetVelocity](#) (int handle, float value)
Joint の速度を設定する
- bool [SetJointTargetVelocity](#) (std::string objectName, float value)
Joint の速度を設定する
- bool [SetJointTargetPosition](#) (int handle, float value)
Joint の位置を設定する (要:Control loop enabled)
- bool [SetJointTargetPosition](#) (std::string objectName, float value)
Joint の位置を設定する (要:Contorol loop enabled)
- geometry_msgs::Pose [GetObjectPose](#) (int handle, int relativeToObjectHandle=-1)
Objct の位置と角度を取得する

4.4.1 Constructor & Destructor Documentation

4.4.1.1 MyVrep::MyVrep (ros::NodeHandle * node) [inline]

コンストラクタ, 名前空間と infocallback のスタンバイ

Parameters

<i>node</i>	nodeHandle のポインタ
-------------	------------------

4.4.2 Member Function Documentation

4.4.2.1 bool MyVrep::EnablePublisher (std::string topicName, int queueSize, int streamCmd, int auxInt1, int auxInt2, std::string auxString) [inline]

Vrep 上のオブジェクトに対し, Publisherを 要求する

Parameters

<i>topicName</i>	Publish するときの topic
<i>queueSize</i>	queue サイズ
<i>streamCmd</i>	streamCmd
<i>auxInt1</i>	auxInt1(streamCmd 参照)
<i>auxInt2</i>	auxInt2(streamCmd 参照)
<i>auxString</i>	auxString(streamCmd 参照)

Return values

<i>true</i>	成功
<i>false</i>	失敗

4.4.2.2 bool MyVrep::EnableSubscriber (std::string topicName, int queueSize, int streamCmd, int auxInt1, int auxInt2, std::string auxString) [inline]

Subscriber を 要求する

Parameters

<i>topicName</i>	要求する topic の名前
<i>queueSize</i>	queue のサイズ
<i>streamCmd</i>	streamCmd
<i>auxInt1</i> (stream- Cmd 参 照)	
<i>auxInt2</i> (stream- Cmd 参 照)	
<i>aux- String</i> (stream- Cmd 参 照)	

Return values

<i>true</i>	成功
<i>false</i>	失敗

4.4.2.3 int MyVrep::GetObjectHandle (std::string *objectName*) [inline]

Object のHandle を取得する

Parameters

<i>objectName</i>	取得するオブジェクトの名前
-------------------	---------------

Returns

取得したオブジェクト

Return values

<i>-1</i>	オブジェクトのハンドル取得に失敗
-----------	------------------

4.4.2.4 geometry_msgs::Pose MyVrep::GetObjectPose (int *handle*, int *relativeToObjectHandle* = -1) [inline]

Object の位置と角度を取得する

Parameters

<i>handle</i>	取得するオブジェクトの handle
<i>relativeToObject- Handle</i>	座標を決めるときの基準点 (指定がない場合は絶対座標)

Returns

Object の位置と角度

4.4.2.5 float MyVrep::GetTime () [inline]

取得した時間を返す

Returns

Vrep 上での時間

4.4.2.6 `bool MyVrep::IsRun () [inline]`

Vrep がいま動いているかどうかを返す

Return values

<i>true</i>	Vrep がいま動いている
<i>false</i>	Vrep がいま止まっている

4.4.2.7 void MyVrep::LoadModel (std::string filename) [inline]

Model を読み込む

Parameters

<i>filename</i>	読み込むファイル (.ttm)
-----------------	-----------------

4.4.2.8 void MyVrep::RemoveModel (int handle) [inline]

Model を除去する

Parameters

<i>handle</i>	除去する Model のハンドル
---------------	------------------

4.4.2.9 void MyVrep::RemoveModel (std::string modelName) [inline]

Model を除去する

Parameters

<i>modelName</i>	除去する Model の名前
------------------	----------------

4.4.2.10 bool MyVrep::SetJointTargetPosition (int handle, float value) [inline]

Joint の位置を設定する (要:Control loop enabled)

Parameters

<i>handle</i>	joint のハンドル
<i>value</i>	joint の位置

Return values

<i>true</i>	成功
<i>false</i>	失敗

4.4.2.11 bool MyVrep::SetJointTargetPosition (std::string objectName, float value) [inline]

Joint の位置を設定する (要:Contorol loop enabled)

Parameters

<i>objectName</i>	joint の名前
<i>value</i>	joint の位置

Return values

<i>true</i>	成功
<i>false</i>	失敗

4.4.2.12 `bool MyVrep::SetJointTargetVelocity (int handle, float value)` `[inline]`

Joint の速度を設定する

Parameters

<i>handle</i>	Joint のハンドル
<i>value</i>	設定する速度

Return values

<i>true</i>	成功
<i>false</i>	失敗

4.4.2.13 `bool MyVrep::SetJointTargetVelocity (std::string objectName, float value)` `[inline]`

Joint の速度を設定する

Parameters

<i>objectName</i>	Joint の名前
<i>value</i>	設定する速度

Return values

<i>true</i>	成功
<i>false</i>	失敗

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

- [myvrep.h](#)

Chapter 5

File Documentation

5.1 myros.h File Reference

ROS の本体及びPublisher と Subscriber のライブラリ

```
#include <memory>
#include <ros/ros.h>
#include <string>
```

Classes

- class [MyROS](#)
ROS の用意をする (いろいろする前に, まずこいつを呼んでください)
- class [MySubscriber< T >](#)
Subscriber オブジェクト
- class [MyPublisher< T >](#)
Publisher オブジェクト

Functions

- `template<typename T, typename... Args>`
`std::unique_ptr< T > make_unique (Args &&...args)`

5.1.1 Detailed Description

ROS の本体及びPublisher と Subscriber のライブラリ

Author

kurenaif

Date

2014-12-17

5.2 myvrep.h File Reference

VrepとROSノードをつなぐためのライブラリ

```
#include "myros.h"
#include <vrep_common/VrepInfo.h>
#include <vrep_common/simRosGetObjectHandle.h>
#include <vrep_common/simRosLoadModel.h>
#include <vrep_common/simRosRemoveModel.h>
#include <vrep_common/simRosStartSimulation.h>
#include <vrep_common/simRosStopSimulation.h>
#include <vrep_common/simRosEnablePublisher.h>
#include <vrep_common/simRosEnableSubscriber.h>
#include <vrep_common/simRosSetJointTargetVelocity.h>
#include <vrep_common/simRosSetJointTargetPosition.h>
#include <geometry_msgs/PoseStamped.h>
#include <vrep_common/simRosGetObjectPose.h>
```

Classes

- class [MyVrep](#)

5.2.1 Detailed Description

VrepとROSノードをつなぐためのライブラリ

Author

kurenaif

Date

2014-12-17

Index

- callback
 - MySubscriber, [10](#)
- EnablePublisher
 - MyVrep, [12](#)
- EnableSubscriber
 - MyVrep, [12](#)
- Get
 - MySubscriber, [10](#), [11](#)
- GetNodeHandlePointer
 - MyROS, [9](#)
- GetObjectHandle
 - MyVrep, [13](#)
- GetObjectPose
 - MyVrep, [13](#)
- GetTime
 - MyVrep, [13](#)
- GetTopicName
 - MyPublisher, [8](#)
- IsNew
 - MySubscriber, [11](#)
- IsRun
 - MyVrep, [13](#)
- LoadModel
 - MyVrep, [15](#)
- MyPublisher
 - GetTopicName, [8](#)
 - MyPublisher, [7](#)
 - MyPublisher, [7](#)
 - operator<<, [8](#)
 - Push, [8](#)
- MyPublisher< T >, [7](#)
- MyROS, [8](#)
 - GetNodeHandlePointer, [9](#)
 - MyROS, [9](#)
 - MyROS, [9](#)
 - Run, [9](#)
- MySubscriber
 - callback, [10](#)
 - Get, [10](#), [11](#)
 - IsNew, [11](#)
 - MySubscriber, [10](#)
 - MySubscriber, [10](#)
- MySubscriber< T >, [9](#)
- MyVrep, [11](#)
 - EnablePublisher, [12](#)
 - EnableSubscriber, [12](#)
- GetObjectHandle, [13](#)
- GetObjectPose, [13](#)
- GetTime, [13](#)
- IsRun, [13](#)
- LoadModel, [15](#)
- MyVrep, [12](#)
- MyVrep, [12](#)
- RemoveModel, [15](#)
- SetJointTargetPosition, [15](#)
- SetJointTargetVelocity, [16](#)
- myros.h, [17](#)
- myvrep.h, [18](#)
- operator<<
 - MyPublisher, [8](#)
- Push
 - MyPublisher, [8](#)
- RemoveModel
 - MyVrep, [15](#)
- Run
 - MyROS, [9](#)
- SetJointTargetPosition
 - MyVrep, [15](#)
- SetJointTargetVelocity
 - MyVrep, [16](#)