Orbiter MMU

PRE-ALPHA API SAMPLE

Not Final - Subject to change (Though it probably won't!)

Generated by Doxygen 1.8.14

Contents

1 Class Index **3**

1.1 Class List 3

2 Class Documentation **5**

2.1 Airlock Struct Reference 5

2.1.1 Member Data Documentation 5

2.1.1.1 isOpen 5

2.1.1.2 position 5

2.1.1.3 radius 5

2.1.1.4 type 6

2.2 oMMUCrew Struct Reference 6

2.2.1 Member Data Documentation 6

2.2.1.1 age 6

2.2.1.2 evaMesh 6

2.2.1.3 name 6

2.2.1.4 pulse 7

2.2.1.5 role 7

2.2.1.6 weight 7

2.3 OMMUManagement Class Reference 7

2.3.1 Detailed Description 8

2.3.2 Member Function Documentation 8

2.3.2.1 AddAirlock() 8

2.3.2.2 AddCrewMember() 9

2.3.2.3 AddDefaultCrew() 9

2.3.2.4 BeginEVA() . . . 9
2.3.2.5 CreateAirlockFromPort() . . . 9
2.3.2.6 DestroyOMMUInstance() . . . 10
2.3.2.7 GetCrewByIndex() . . . 10
2.3.2.8 GetOMMUInstance() . . . 10
2.3.2.9 ProcessClbkGeneric() . . . 11
2.3.2.10 ProcessClbkPreStep() . . . 11
2.3.2.11 RecallState() . . . 11
2.3.2.12 RemoveCrewMemberByID() . . . 11
2.3.2.13 SaveState() . . . 12
2.3.2.14 SetCrewLimit() . . . 12
2.3.2.15 TryIngress() . . . 12

3 File Documentation **15**

3.1 oMMU API/oMMU_API.cpp File Reference . . . 15
3.1.1 Typedef Documentation . . . 15
3.1.1.1 ioMMUFactory . . . 15
3.2 oMMU API/oMMU_API.h File Reference . . . 15
3.3 oMMU API/oMMU_Data.h File Reference . . . 16
3.3.1 Enumeration Type Documentation . . . 16
3.3.1.1 AirlockType . . . 16
3.3.1.2 oMMUStatus . . . 16

Chapter 1

Class Index

1.1 Class List

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

Airlock 5
oMMUCrew 6
OMMUManagement
    OMMU Interface class 7

Chapter 2

Class Documentation

2.1 Airlock Struct Reference

`#include <oMMU_Data.h>`

Public Attributes

- bool isOpen
- AirlockType type
- VECTOR3 position
- double radius

2.1.1 Member Data Documentation

2.1.1.1 isOpen

`bool Airlock::isOpen`

2.1.1.2 position

`VECTOR3 Airlock::position`

2.1.1.3 radius

`double Airlock::radius`

2.1.1.4 type

```
AirlockType Airlock::type
```

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

- oMMU API/oMMU_Data.h

2.2 oMMUCrew Struct Reference

```
#include <oMMU_Data.h>
```

Public Attributes

- std::string role
- std::string name
- int age
- double weight
- int pulse
- std::string evaMesh

2.2.1 Member Data Documentation

2.2.1.1 age

```
int oMMUCrew::age
```

2.2.1.2 evaMesh

```
std::string oMMUCrew::evaMesh
```

2.2.1.3 name

```
std::string oMMUCrew::name
```

2.2.1.4 pulse

```
int oMMUCrew::pulse
```

2.2.1.5 role

```
std::string oMMUCrew::role
```

2.2.1.6 weight

```
double oMMUCrew::weight
```

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

- oMMU API/oMMU_Data.h

2.3 OMMUManagement Class Reference

OMMU Interface class.

```
#include <oMMU_API.h>
```

Public Member Functions

Simulation

Simulation callbacks of OMMU - these should be placed at the top of their respective functions to ensure that they are called correctly.

- virtual oMMUStatus ProcessClbkPreStep ()=0
  Simulate OMMU crew functions - this should be called in clbkPreStep.
- virtual bool ProcessClbkGeneric (int msg, void ∗data)=0
  Process generic callback messages - this should be placed in clbkGeneric.
- virtual bool RecallState (char ∗line)=0
- virtual void SaveState (FILEHANDLE scn)=0

Configuration

Configuration functions.

- virtual void SetCrewLimit (int crewLimit)=0
  Sets the maximum amount of crew that the vessel is able to contain.
- virtual int AddAirlock (const Airlock &airlock)=0
  Creates an arbitary airlock of the given parameters.
- virtual int CreateAirlockFromPort (int portID, bool openByDefault)=0
  Creates an airlock based on an existing docking port.

Crew Management

Crew management functions.

- virtual int TryIngress (VESSEL ∗hMMU, double ∗ret)=0
  Called when an MMU attempts to enter the vessels' airlock. TODO: Move to private API?
- virtual oMMUCrew ∗ GetCrewByIndex (int index)=0
  Returns the crew member at the given index, or nullptr if no crew member is present.
- virtual oMMUStatus BeginEVA (int crewIndex, int airlockIndex=0)=0
  EVA or Transfer the crew member at the given index.
- virtual oMMUStatus AddCrewMember (const oMMUCrew &crewToAdd)=0
  Adds the crew member specified by the given struct to the vessel if possible.
- virtual oMMUStatus RemoveCrewMemberByID (int index)=0
  Removes the crew member at the given index.
- virtual int AddDefaultCrew ()=0

Static Public Member Functions

Core

Core functions of OMMU - used for initialization / destruction of the OMMUManagement interface

- static OMMUManagement ∗ GetOMMUInstance (VESSEL ∗hVessel)
  Instantiates an instance of the OMMUManagement class and then returns a pointer to this instance.
- static void DestroyOMMUInstance (OMMUManagement ∗hMMU)
  Destroys an instance of the OMMUManagement class - this should be called in your vessel destructor.

2.3.1 Detailed Description

OMMU Interface class.

2.3.2 Member Function Documentation

2.3.2.1 AddAirlock()

```
virtual int OMMUManagement::AddAirlock (
            const Airlock & airlock )  [pure virtual]
```

Creates an arbitary airlock of the given parameters.

Parameters

airlock	Reference to the airlock data structure that should be tied to this airlock. See sample code for example.

Returns

ID of the airlock

2.3.2.2 AddCrewMember()

```
virtual oMMUStatus OMMUManagement::AddCrewMember (
            const oMMUCrew & crewToAdd )  [pure virtual]
```

Adds the crew member specified by the given struct to the vessel if possible.

Parameters

crewToAdd	

Returns

OMMUStats::CrewAdded on success, context specific failure message otherwise

2.3.2.3 AddDefaultCrew()

```
virtual int OMMUManagement::AddDefaultCrew ( )  [pure virtual]
```

2.3.2.4 BeginEVA()

```
virtual oMMUStatus OMMUManagement::BeginEVA (
            int crewIndex,
            int airlockIndex = 0 )  [pure virtual]
```

EVA or Transfer the crew member at the given index.

Parameters

crewIndex	Index of the crew member to be EVA'd
airlockIndex	Index of the airlock to EVA from,

Returns

OMMUStatus message - EVASucess on sucess, context specific failure message otherwise

2.3.2.5 CreateAirlockFromPort()

```
virtual int OMMUManagement::CreateAirlockFromPort (
            int portID,
            bool openByDefault )  [pure virtual]
```

Creates an airlock based on an existing docking port.

Parameters

portID	numeric ID of the docking port the airlock should be created against.
openByDefault	Sets the airlock's default state.

Returns

ID of the airlock

2.3.2.6 DestroyOMMUInstance()

```
void OMMUManagement::DestroyOMMUInstance (
            OMMUManagement * hMMU )  [static]
```

Destroys an instance of the OMMUManagement class - this should be called in your vessel destructor.

Parameters

hMMU	Pointer to the OMMUMangement object that should be finalized and destroyed

2.3.2.7 GetCrewByIndex()

```
virtual oMMUCrew* OMMUManagement::GetCrewByIndex (
            int index )  [pure virtual]
```

Returns the crew member at the given index, or nullptr if no crew member is present.

Parameters

index	Index to retrieve a crew member from

Returns

A pointer to the crew member data structure, or nullptr if no crew member is present.

2.3.2.8 GetOMMUInstance()

```
OMMUManagement * OMMUManagement::GetOMMUInstance (
            VESSEL * hVessel )  [static]
```

Instantiates an instance of the OMMUManagement class and then returns a pointer to this instance.

Parameters

hVessel	Vessel implementing OMMU, usually 'this'

Returns

Returns an instance of OMMU, or null if an error has occured

2.3.2.9 ProcessClbkGeneric()

```
virtual bool OMMUManagement::ProcessClbkGeneric (
            int msg,
            void * data )  [pure virtual]
```

Process generic callback messages - this should be placed in clbkGeneric.

Parameters

msg	
data	

Returns

True on message consumption, false otherwise

2.3.2.10 ProcessClbkPreStep()

```
virtual oMMUStatus OMMUManagement::ProcessClbkPreStep ( )  [pure virtual]
```

Simulate OMMU crew functions - this should be called in clbkPreStep.

Returns

Not used at this time

2.3.2.11 RecallState()

```
virtual bool OMMUManagement::RecallState (
            char * line )  [pure virtual]
```

2.3.2.12 RemoveCrewMemberByID()

```
virtual oMMUStatus OMMUManagement::RemoveCrewMemberByID (
            int index )  [pure virtual]
```

Removes the crew member at the given index.

Parameters

index	Index of the crew member to remove

Returns

OMMUStatus::OK on success, context specific failure message otherwise.

2.3.2.13 SaveState()

```
virtual void OMMUManagement::SaveState (
            FILEHANDLE scn )  [pure virtual]
```

2.3.2.14 SetCrewLimit()

```
virtual void OMMUManagement::SetCrewLimit (
            int crewLimit )  [pure virtual]
```

Sets the maximum amount of crew that the vessel is able to contain.

Parameters

crewLimit	The new crew limit.

Returns

Returns CREW_LIMIT_CHANGE_SUCESS on sucess, -1 otherwise

2.3.2.15 TryIngress()

```
virtual int OMMUManagement::TryIngress (
            VESSEL * hMMU,
            double * ret )  [pure virtual]
```

Called when an MMU attempts to enter the vessels' airlock. TODO: Move to private API?

Parameters

hMMU	pointer to the MMU vessel attempting to enter.
ret	additional data return pointer.

Returns

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

- oMMU API/oMMU_API.h
- oMMU API/oMMU_API.cpp

Chapter 3

File Documentation

3.1 oMMU API/oMMU_API.cpp File Reference

```
#include "oMMU_API.h"
```

Typedefs

- typedef OMMUManagement *(__cdecl * ioMMUFactory) (VESSEL *hVessel)

3.1.1 Typedef Documentation

3.1.1.1 ioMMUFactory

```
typedef OMMUManagement*(__cdecl * ioMMUFactory) (VESSEL *hVessel)
```

3.2 oMMU API/oMMU_API.h File Reference

```
#include "OrbiterAPI.h"
#include "oMMU_Data.h"
#include "Orbitersdk.h"
```

Classes

- class OMMUManagement

  OMMU Interface class.

3.3 oMMU API/oMMU_Data.h File Reference

```
#include <string>
```

Classes

- struct Airlock
- struct oMMUCrew

Enumerations

- enum oMMUStatus {
  OK, EVASucess, EVAFailure, AirlockClosed,
  VesselFull, VesselAirlockClosed, VesselEmpty, CrewTransferSucess,
  CrewTransferFailure }
- enum AirlockType { dockingPort, sphere }

3.3.1 Enumeration Type Documentation

3.3.1.1 AirlockType

```
enum AirlockType
```

Enumerator

dockingPort	
sphere	

3.3.1.2 oMMUStatus

```
enum oMMUStatus
```

Enumerator

OK	
EVASucess	
EVAFailure	
AirlockClosed	
VesselFull	
VesselAirlockClosed	
VesselEmpty	
CrewTransferSucess	
CrewTransferFailure	