JEOD Overview

JEOD API
Model Documents
Model Reference Manuals

The JEOD API

- At **html/jeod/index.html**
- Front page:

JEOD 3.2.1

Main Page | Related Pages | Modules | Namespaces | Data Structures | Files | Directories

JEOD

These pages present the JEOD 3.2 API in HTML format.
Click on

- "Main Page" to display this page.
- "Related Page" to display related pages.
- "Modules" to see a hierarchical presentation of the JEOD models and tools.
- "Data Structures" to see a clickable list of all the JEOD classes.
- "Files" to see a clickable list of the JEOD source and header files.

Additional Documentation

Key features that distinguish the current release, build instructions, and a history of JEOD are described in the **JEOD README file**.

The main documentation for JEOD is in the form of pdf files. These pdf files collectively form JSC-61777, "JSC Engineering Orbital Dynamics". This documentation tree is headed by the JEOD toplevel document.

Each model is accompanied with a document named <model_name>.pdf, where where <model_name> is the name of the model. This document describes the requirements, design, usage, and testing of the model. See the **Model Documents** page for a list of the model documentation files.

JEOD provides several other documents in addition to the toplevel document and the model documents. These include descriptions of the JEOD-wide verification and validation simulations and tutorials. See the **Toplevel Documents** page for a list of these additional documentation files.

Dependency Diagrams

Ensuring that the initialization, scheduled, and derivative jobs are called in the proper order can be a challenging task. The various simulations packaged with JEOD can serve as an exemplar for achieving this task. The diagrams and dot files listed below depict the dependencies in the dynamics comparison simulation (**click to see the S_define**):

- Initialization jobs: dependencies diagram and **dot file**.
- Scheduled jobs: dependencies diagram and **dot file**.
- Derivative jobs: dependencies diagram and **dot file**.

Caveat on Displayed Source Code

These doxygen pages include facsimiles of the JEOD source code. A word of warning: These are facsimiles. Some differences do exist between the true source code and the source presented in this API documentation. In particular,

- The line numbers differ between the displayed source and the true source.
- The combination of the preparation of files for doxygen and the processing by doxygen strips some comments from the displayed source.

Generated by doxygen 1.7.5.1

Related Pages

- README
- Top level documents
- Model documents

Documentation

(Uniform Presentation)

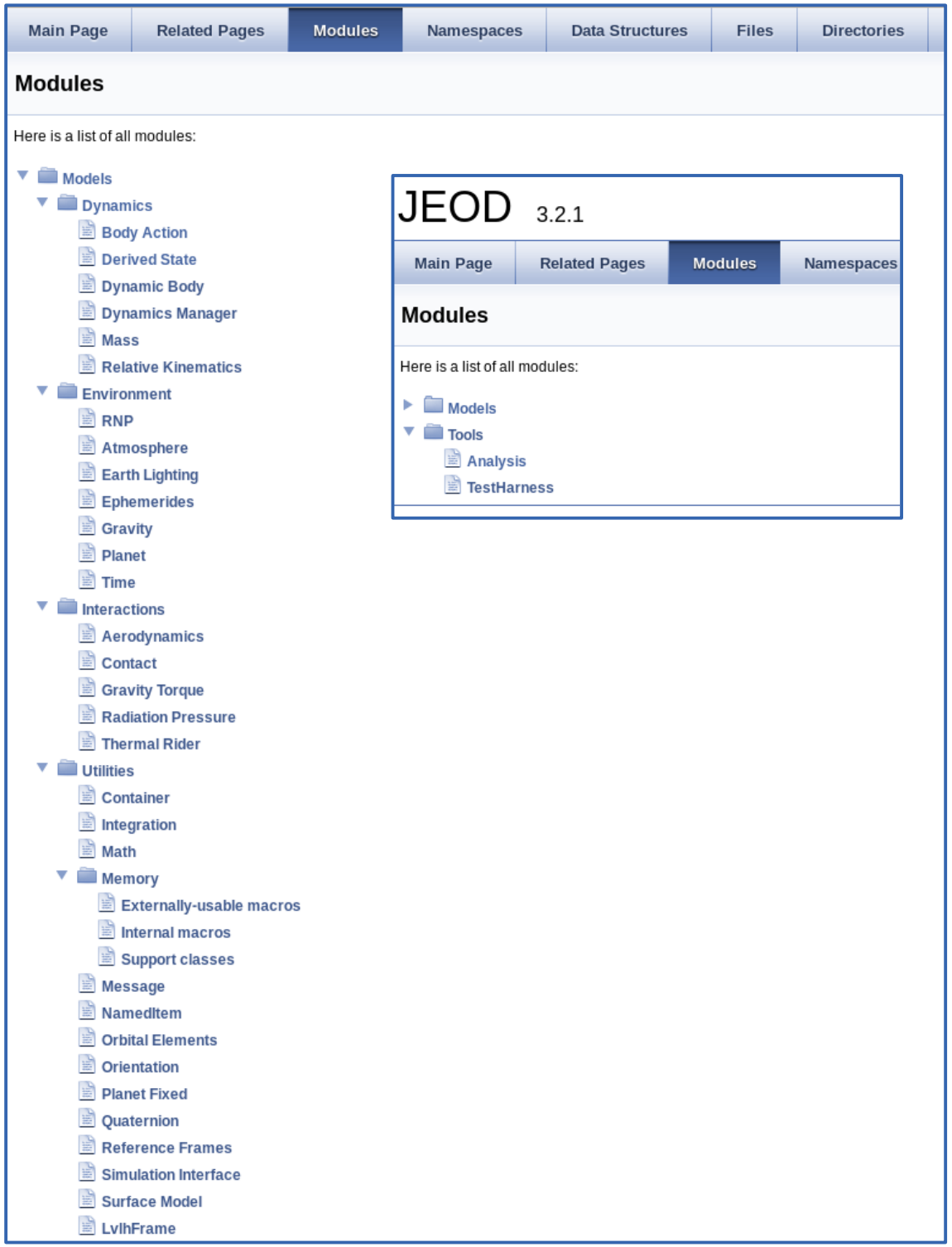
Main Document (<model>.pdf)

- Introduction
- Requirements
- Product Specification
  - Conceptual Design
  - Mathematical Formulations
  - Detailed Design
  - Inventory
- User Guide
  - Simulation Users (inputs, outputs)
  - Simulation Developers (instantiation, calling)
  - Model Developers (extending)
- Inspection, Tests, Metrics

Reference Manual

(refman.pdf)

- Data Structure Index
- File Index
- Data Structrure Documentation
  - Inheritance Diagram
- File Documentation



Modules

- List of all models and tools
- Link to each

Models

- Lists classes within each model
- Brief description of each
- Link to documentation

Main Page | Related Pages | Modules | Namespaces | Data Structures | Files | Directories

Body Action

Dynamics

The Body Action Model (model document) provides mechanisms for performing asynchronous actions on mass bodies and dynamic bodies. More...

Collaboration diagram for Body Action:

Data Structures

class	**BodyAction** **BodyAction** is the base class for the **BodyAction** model. More...
class	**BodyActionMessages** Specifies the message IDs used in the **BodyAction** model. More...
class	**DynBodyFrameSwitch** Switch a DynBody's integration frame to a specified frame when the body switches to that integration frame's sphere of influence. More...
class	**DynBodyInit** Base class for initialize the state of a **DynBody**. More...
class	**DynBodyInitLvlhRotState** Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame. More...
class	**DynBodyInitLvlhState** Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame. More...

Data Structures / Classes

JEOD 3.2.1

Main Page | Related Pages | Modules | Namespaces | Data Structures | Files | Directories

Data Structures | Class Hierarchy | Data Fields

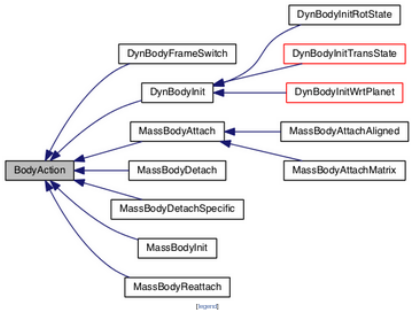
BodyAction Class Reference

Body Action

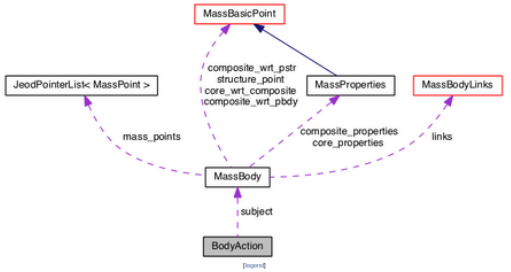
BodyAction is the base class for the BodyAction model. More...

#include <body_action.hh>

Inheritance diagram for BodyAction:



Collaboration diagram for BodyAction:



Public Member Functions

	BodyAction () Construct a **BodyAction**.
virtual	**~BodyAction** () Destruct a **BodyAction**.
virtual void	**shutdown** () Release resources allocated by a **BodyAction** object.
const char *	**get_identifier** (void) const Accessor for action_identifier.
virtual void	**initialize** (**DynManager** &dyn_manager)
virtual bool	**is_ready** (void) In general, determine if the initializer is ready to be applied.
virtual void	**apply** (**DynManager** &dyn_manager)

Data Fields

MassBody *	**subject** trick_units(--) The body that is the subject of this action.
bool	**activate**

- Inheritance
- Collaboration
- Fields
- Clickable
- Legend

Data Structures

- Alphabetical list of structures / classes
- Each goes to page seen on previous slide

- Break - Exercises 2, 3

Files

- Alphabetical list of files
- Each has link to actual code content
- Link to detailed description, including dependency graph.
  - Note – code dependencies, not execution dependencies; difference in option-usage

JEOD 3.2.1

Main Page | Related Pages | Modules | Namespaces | Data Structures | Files | Directories

File List | Globals

models › interactions › aerodynamics › src

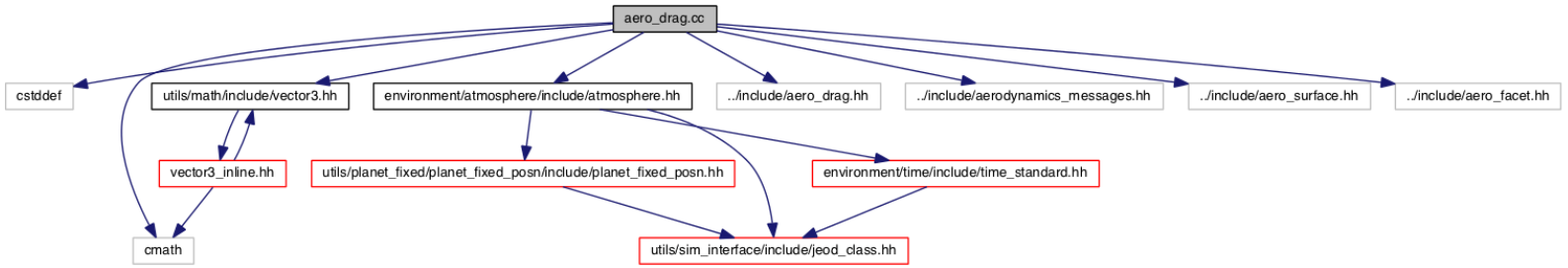
aero_drag.cc File Reference

Aerodynamics

Orbital aerodynamic force and torque computation, and related classes. More...

```
#include <cstddef>
#include <cmath>
#include "utils/math/include/vector3.hh"
#include "environment/atmosphere/include/atmosphere.hh"
#include "../include/aero_drag.hh"
#include "../include/aerodynamics_messages.hh"
#include "../include/aero_surface.hh"
#include "../include/aero_facet.hh"
```

Include dependency graph for aero_drag.cc:



Go to the source code of this file.

Detailed Description

Orbital aerodynamic force and torque computation, and related classes.

Assumptions and Limitations

- Orbital body modelled as a series of geometric plates, or as a simple ballistic coefficient or coefficient of drag, unless the user has overriden any of these behaviors
- All plate information referenced to the orbital vehicle's structural frame.

Directories

- Directory index of files
- Dependency graphs at all levels