

# SpiceModel

5.0

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>11</b>
6.1	Models . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.2	Environment . . . . .	12
6.2.1	Detailed Description . . . . .	12
6.3	Spice . . . . .	13
6.3.1	Detailed Description . . . . .	13
6.3.2	Variable Documentation . . . . .	13
6.3.2.1	MAX_IDS . . . . .	13
6.3.2.2	MAX_MSG_LENGTH . . . . .	14
6.3.2.3	MAX_NAME_LENGTH . . . . .	14
6.3.2.4	MAX_PATH_LENGTH . . . . .	14

<b>7</b>	<b>Namespace Documentation</b>	<b>15</b>
7.1	jeod Namespace Reference . . . . .	15
7.1.1	Detailed Description . . . . .	15
<b>8</b>	<b>Data Structure Documentation</b>	<b>17</b>
8.1	jeod::SpiceEphemeris Class Reference . . . . .	17
8.1.1	Detailed Description . . . . .	20
8.1.2	Constructor & Destructor Documentation . . . . .	20
8.1.2.1	SpiceEphemeris() [1/2] . . . . .	20
8.1.2.2	~SpiceEphemeris() . . . . .	20
8.1.2.3	SpiceEphemeris() [2/2] . . . . .	20
8.1.3	Member Function Documentation . . . . .	20
8.1.3.1	activate() . . . . .	21
8.1.3.2	add_barycenter() . . . . .	21
8.1.3.3	add_descendants_r() . . . . .	21
8.1.3.4	add_orientation() . . . . .	22
8.1.3.5	add_planet_name() . . . . .	22
8.1.3.6	create_barycenters() . . . . .	22
8.1.3.7	create_new_ephem_orientation() . . . . .	22
8.1.3.8	create_new_ephem_point() . . . . .	23
8.1.3.9	deactivate() . . . . .	23
8.1.3.10	determine_root_node() . . . . .	24
8.1.3.11	ephem_activate() . . . . .	24
8.1.3.12	ephem_build_tree() . . . . .	24
8.1.3.13	ephem_initialize() . . . . .	24
8.1.3.14	ephem_update() . . . . .	25
8.1.3.15	find_parent_id() . . . . .	25
8.1.3.16	find_spice_id() . . . . .	26
8.1.3.17	get_name() . . . . .	26
8.1.3.18	initialize_items() . . . . .	27
8.1.3.19	initialize_model() . . . . .	27

8.1.3.20	<a href="#">initialize_time()</a>	27
8.1.3.21	<a href="#">introduce_item()</a>	28
8.1.3.22	<a href="#">jeod_2_spice_pfix()</a>	28
8.1.3.23	<a href="#">load_spice_files()</a>	28
8.1.3.24	<a href="#">mute_spice_errors()</a>	29
8.1.3.25	<a href="#">name_barycenter_frames()</a>	29
8.1.3.26	<a href="#">operator=()</a>	29
8.1.3.27	<a href="#">populate_item()</a>	29
8.1.3.28	<a href="#">process_orientations()</a>	30
8.1.3.29	<a href="#">process_spk()</a>	30
8.1.3.30	<a href="#">simple_restore()</a>	30
8.1.3.31	<a href="#">spice_2_jeod()</a>	30
8.1.3.32	<a href="#">timestamp()</a>	31
8.1.3.33	<a href="#">update_rot()</a>	31
8.1.3.34	<a href="#">update_trans()</a>	31
8.1.4	<a href="#">Friends And Related Function Documentation</a>	32
8.1.4.1	<a href="#">init_attrjeod__SpiceEphemeris</a>	32
8.1.4.2	<a href="#">InputProcessor</a>	32
8.1.5	<a href="#">Field Documentation</a>	32
8.1.5.1	<a href="#">barycenter_frames</a>	32
8.1.5.2	<a href="#">dyn_seconds</a>	32
8.1.5.3	<a href="#">ephem_mgr_local</a>	33
8.1.5.4	<a href="#">force_update</a>	33
8.1.5.5	<a href="#">ident</a>	33
8.1.5.6	<a href="#">inactive</a>	33
8.1.5.7	<a href="#">loaded_spk</a>	34
8.1.5.8	<a href="#">metakernel_filename</a>	34
8.1.5.9	<a href="#">orientation_names</a>	34
8.1.5.10	<a href="#">planet_names</a>	34
8.1.5.11	<a href="#">planetary_orientations</a>	35

8.1.5.12	<a href="#">root_item</a>	35
8.1.5.13	<a href="#">tdb_seconds</a>	35
8.1.5.14	<a href="#">update_time</a>	35
8.2	<a href="#">jeod::SpiceEphemOrientation Class Reference</a>	36
8.2.1	<a href="#">Detailed Description</a>	37
8.2.2	<a href="#">Constructor &amp; Destructor Documentation</a>	37
8.2.2.1	<a href="#">SpiceEphemOrientation() [1/2]</a>	37
8.2.2.2	<a href="#">~SpiceEphemOrientation()</a>	37
8.2.2.3	<a href="#">SpiceEphemOrientation() [2/2]</a>	37
8.2.3	<a href="#">Member Function Documentation</a>	37
8.2.3.1	<a href="#">get_spice_transformation()</a>	37
8.2.3.2	<a href="#">operator=()</a>	38
8.2.3.3	<a href="#">set_spice_frame_name()</a>	38
8.2.3.4	<a href="#">update()</a>	38
8.2.3.5	<a href="#">validate()</a>	39
8.2.4	<a href="#">Friends And Related Function Documentation</a>	39
8.2.4.1	<a href="#">init_attrjeod__SpiceEphemOrientation</a>	39
8.2.4.2	<a href="#">InputProcessor</a>	39
8.2.5	<a href="#">Field Documentation</a>	40
8.2.5.1	<a href="#">spice_frame_name</a>	40
8.3	<a href="#">jeod::SpiceEphemPoint Class Reference</a>	40
8.3.1	<a href="#">Detailed Description</a>	41
8.3.2	<a href="#">Member Enumeration Documentation</a>	42
8.3.2.1	<a href="#">Status</a>	42
8.3.3	<a href="#">Constructor &amp; Destructor Documentation</a>	43
8.3.3.1	<a href="#">SpiceEphemPoint() [1/2]</a>	43
8.3.3.2	<a href="#">~SpiceEphemPoint()</a>	43
8.3.3.3	<a href="#">SpiceEphemPoint() [2/2]</a>	43
8.3.4	<a href="#">Member Function Documentation</a>	43
8.3.4.1	<a href="#">get_parent_id()</a>	44
8.3.4.2	<a href="#">get_spice_id()</a>	44
8.3.4.3	<a href="#">get_status()</a>	44
8.3.4.4	<a href="#">operator=()</a>	45
8.3.4.5	<a href="#">set_parent_id()</a>	45
8.3.4.6	<a href="#">set_spice_id()</a>	46
8.3.4.7	<a href="#">set_status()</a>	46
8.3.5	<a href="#">Friends And Related Function Documentation</a>	46
8.3.5.1	<a href="#">init_attrjeod__SpiceEphemPoint</a>	47
8.3.5.2	<a href="#">InputProcessor</a>	47
8.3.6	<a href="#">Field Documentation</a>	47
8.3.6.1	<a href="#">parent_id</a>	47
8.3.6.2	<a href="#">spice_id</a>	47
8.3.6.3	<a href="#">status</a>	47

<b>9 File Documentation</b>	<b>49</b>
9.1 spice_ephem.cc File Reference . . . . .	49
9.1.1 Detailed Description . . . . .	49
9.2 spice_ephem.hh File Reference . . . . .	50
9.2.1 Detailed Description . . . . .	50
9.3 spice_ephem_orient.cc File Reference . . . . .	50
9.3.1 Detailed Description . . . . .	51
9.4 spice_ephem_orient.hh File Reference . . . . .	51
9.4.1 Detailed Description . . . . .	51
9.5 spice_ephem_point.cc File Reference . . . . .	51
9.5.1 Detailed Description . . . . .	51
9.6 spice_ephem_point.hh File Reference . . . . .	52
9.6.1 Detailed Description . . . . .	52
<b>Index</b>	<b>53</b>





# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	11
Environment . . . . .	12
Spice . . . . .	13



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">jeod</a>	Namespace jeod . . . . .	<a href="#">15</a>
----------------------	--------------------------	--------------------



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

EphemerisInterface	
jeod::SpiceEphemeris . . . . .	17
EphemerisOrientation	
jeod::SpiceEphemOrientation . . . . .	36
EphemerisPoint	
jeod::SpiceEphemPoint . . . . .	40
RefFrameOwner	
jeod::SpiceEphemeris . . . . .	17
SimpleCheckpointable	
jeod::SpiceEphemeris . . . . .	17



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::SpiceEphemeris</a>	
The S_define-level class that provides planetary ephemerides . . . . .	17
<a href="#">jeod::SpiceEphemOrientation</a>	
A <a href="#">SpiceEphemOrientation</a> minimally extends EphemerisOrientation to include the JEOD and SPICE names and an update method for the target ephemeris reference frame . . . . .	36
<a href="#">jeod::SpiceEphemPoint</a>	
A <a href="#">SpiceEphemPoint</a> minimally extends EphemerisPoint, primarily in order to contain the ID of the SPICE object that will be used to update the state of the target ephemeris reference frame . . .	40





## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">spice_ephem.cc</a>	Define the methods for the SPICE ephemeris model class . . . . .	49
<a href="#">spice_ephem.hh</a>	Define class for the SPICE ephemeris model . . . . .	50
<a href="#">spice_ephem_orient.cc</a>	Define the methods for the SPICE-specific ephemeris orientation class . . . . .	50
<a href="#">spice_ephem_orient.hh</a>	Define class SpiceEphemOrient, which extends EphemerisOrientation for use with the SPICE ephemeris model . . . . .	51
<a href="#">spice_ephem_point.cc</a>	Define the methods for the SPICE-specific ephemeris point class . . . . .	51
<a href="#">spice_ephem_point.hh</a>	Define class SpiceEphemPoint, which extends EphemerisPoint for use with the SPICE ephemeris model . . . . .	52



## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Environment](#)

#### 6.1.1 Detailed Description

## 6.2 Environment

### Modules

- [Spice](#)

### 6.2.1 Detailed Description

## 6.3 Spice

### Files

- file [spice\\_ephem.hh](#)  
*Define class for the SPICE ephemeris model.*
- file [spice\\_ephem\\_orient.hh](#)  
*Define class SpiceEphemOrient, which extends EphemerisOrientation for use with the SPICE ephemeris model.*
- file [spice\\_ephem\\_point.hh](#)  
*Define class SpiceEphemPoint, which extends EphemerisPoint for use with the SPICE ephemeris model.*
- file [spice\\_ephem.cc](#)  
*Define the methods for the SPICE ephemeris model class.*
- file [spice\\_ephem\\_orient.cc](#)  
*Define the methods for the SPICE-specific ephemeris orientation class.*
- file [spice\\_ephem\\_point.cc](#)  
*Define the methods for the SPICE-specific ephemeris point class.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Variables

- static const int [MAX\\_PATH\\_LENGTH](#) = 129
- static const int [MAX\\_NAME\\_LENGTH](#) = 33
- static const int [MAX\\_MSG\\_LENGTH](#) = 1841
- static const int [MAX\\_IDS](#) = 1000

#### 6.3.1 Detailed Description

#### 6.3.2 Variable Documentation

##### 6.3.2.1 MAX\_IDS

```
const int MAX_IDS = 1000 [static]
```

Definition at line 57 of file [spice\\_ephem.cc](#).

Referenced by [jeod::SpiceEphemeris::process\\_spk\(\)](#).

### 6.3.2.2 MAX\_MSG\_LENGTH

```
const int MAX_MSG_LENGTH = 1841 [static]
```

Definition at line 56 of file `spice_ephem.cc`.

Referenced by `jeod::SpiceEphemeris::load_spice_files()`, and `jeod::SpiceEphemeris::update_trans()`.

### 6.3.2.3 MAX\_NAME\_LENGTH

```
const int MAX_NAME_LENGTH = 33 [static]
```

Definition at line 55 of file `spice_ephem.cc`.

Referenced by `jeod::SpiceEphemeris::add_barycenter()`, `jeod::SpiceEphemeris::mute_spice_errors()`, `jeod::SpiceEphemeris::name_barycenter_frames()`, and `jeod::SpiceEphemeris::process_spk()`.

### 6.3.2.4 MAX\_PATH\_LENGTH

```
const int MAX_PATH_LENGTH = 129 [static]
```

Definition at line 54 of file `spice_ephem.cc`.

Referenced by `jeod::SpiceEphemeris::process_spk()`.

## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

- class [SpiceEphemeris](#)  
*The S\_define-level class that provides planetary ephemerides.*
- class [SpiceEphemOrientation](#)  
*A [SpiceEphemOrientation](#) minimally extends EphemerisOrientation to include the JEOD and SPICE names and an update method for the target ephemeris reference frame.*
- class [SpiceEphemPoint](#)  
*A [SpiceEphemPoint](#) minimally extends EphemerisPoint, primarily in order to contain the ID of the SPICE object that will be used to update the state of the target ephemeris reference frame.*

#### 7.1.1 Detailed Description

Namespace jeod.





## Chapter 8

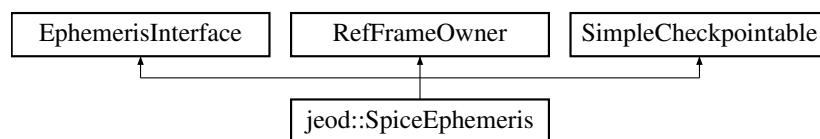
# Data Structure Documentation

### 8.1 jeod::SpiceEphemeris Class Reference

The S\_define-level class that provides planetary ephemerides.

```
#include <spice_ephem.hh>
```

Inheritance diagram for jeod::SpiceEphemeris:



#### Public Member Functions

- [SpiceEphemeris](#) (void)  
*SpiceEphemeris default constructor.*
- [~SpiceEphemeris](#) (void)  
*SpiceEphemeris destructor.*
- void [initialize\\_model](#) (const TimeManager &time\_manager, EphemeridesManager &ephem\_manager)  
*Initialize the [SpiceEphemeris](#) model.*
- virtual void [activate](#) (void)  
*Nominally, activate the object.*
- virtual void [deactivate](#) (void)  
*Deactivate the [SpiceEphemeris](#) object.*
- virtual double [timestamp](#) (void) const  
*Return time of last update.*
- virtual const char \* [get\\_name](#) (void) const  
*Return model name.*
- virtual void [ephem\\_initialize](#) (EphemeridesManager &ephem\_manager)  
*Complete the initialization process.*
- virtual void [ephem\\_activate](#) (EphemeridesManager &ephem\_manager)  
*Mark appropriate items in the model as active.*

- virtual void [ephem\\_build\\_tree](#) (EphemeridesManager &ephem\_manager)  
*Construct the ephemeris model portions of the reference frame tree.*
- virtual void [ephem\\_update](#) (void)  
*Update ephemerides for subscribed items.*
- virtual void [simple\\_restore](#) (void)  
*Set the SPICE model for a restart.*
- void [add\\_planet\\_name](#) (std::string planet\_name)
- void [add\\_orientation](#) (std::string object\_name)
- [SpiceEphemPoint](#) \* [find\\_spice\\_id](#) (int id\_to\_find)  
*Find a SPICE ID in the loaded list of SPICE objects.*

## Data Fields

- std::string [metakernel\\_filename](#)  
*The name of a text file containing the list of SPICE files to be loaded.*

## Protected Attributes

- bool [inactive](#)  
*If set to true, makes the model inactive.*

## Private Member Functions

- void [initialize\\_time](#) (const TimeManager &time\_manager)  
*Initialize [SpiceEphemeris](#) timing.*
- void [load\\_spice\\_files](#) (void)  
*Load SPICE kernel files containing ephemeris data.*
- void [process\\_spk](#) (void)  
*Process spk objects and store array of loaded spk IDs.*
- void [process\\_orientations](#) (void)  
*Load and locate all planetary orientation frames.*
- void [introduce\\_item](#) (EphemerisItem &item)  
*Introduce an [EphemerisItem](#) to the [EphemeridesManager](#).*
- void [populate\\_item](#) (EphemerisItem &item, std::string name)  
*Populate basic attributes of a new [SpiceEphemPoint](#).*
- [SpiceEphemPoint](#) \* [create\\_new\\_ephem\\_point](#) (std::string object\_name, std::string spice\_name)  
*Create a new [SpiceEphemPoint](#).*
- [SpiceEphemOrientation](#) \* [create\\_new\\_ephem\\_orientation](#) (std::string jeod\_name)  
*Create a new [SpiceEphemOrientation](#).*
- void [initialize\\_items](#) ()  
*Initialize the [SpiceEphemeris](#) item data.*
- std::string [spice\\_2\\_jeod](#) (std::string spice\_name)  
*Convert SPICE names to JEOD nomenclature.*
- std::string [jeod\\_2\\_spice\\_pfix](#) (std::string jeod\_name)  
*Convert JEOD body name to SPICE pfix frame name.*
- void [name\\_barycenter\\_frames](#) ()  
*Name all the planetary barycenters with their JEOD identifiers.*
- void [add\\_barycenter](#) (int spice\_id)  
*Add a barycenter corresponding to the given SPICE ID.*

- void [create\\_barycenters](#) ()  
*Check what if any barycenters need creating and do so.*
- void [determine\\_root\\_node](#) (void)  
*Determine which item should be the root of the ref frame tree.*
- void [add\\_descendants\\_r](#) ([SpiceEphemPoint](#) \*parent)  
*Add all offspring from a given node of the ref frame tree.*
- int [find\\_parent\\_id](#) (int obj\_id)  
*Find the ID of the parent of a given SPICE object.*
- void [update\\_trans](#) ()  
*Update ephemerides of inertial frames supplied by spk files.*
- void [update\\_rot](#) ()  
*Update planetary orientations.*
- void [mute\\_spice\\_errors](#) ()  
*Mute SPICE errors so that they can be handled by the MessageHandler.*
- [SpiceEphemeris](#) (const [SpiceEphemeris](#) &)  
*Not implemented.*
- [SpiceEphemeris](#) & [operator=](#) (const [SpiceEphemeris](#) &)  
*Not implemented.*

### Private Attributes

- bool [force\\_update](#)  
*Is an update needed even if the time hasn't changed?*
- [JeodObjectVector](#)< std::string >::type [planet\\_names](#)  
*The names of all planets to load into the simulation.*
- [JeodObjectVector](#)< std::string >::type [orientation\\_names](#)  
*The names of all objects for which orientation is required.*
- [JeodPointerVector](#)< [SpiceEphemPoint](#) >::type [loaded\\_spk](#)  
*All the spk objects tracked by this ephemeris.*
- [JeodPointerVector](#)< [SpiceEphemOrientation](#) >::type [planetary\\_orientations](#)  
*Objects defining planet-fixed reference frames.*
- std::string [ident](#)  
*Identifier for this model, set by the constructor.*
- double [update\\_time](#)  
*Time of last update, dynamic time seconds.*
- [EphemerisRefFrame](#) [barycenter\\_frames](#) [10]  
*Array of barycenter frames corresponding to SSBary and the planets.*
- [SpiceEphemPoint](#) \* [root\\_item](#)  
*The root point in the reference frame tree.*
- const double \* [tdb\\_seconds](#)  
*The source of ephemeris time information.*
- const double \* [dyn\\_seconds](#)  
*The source of dynamic time information.*
- [EphemeridesManager](#) \* [ephem\\_mgr\\_local](#)  
*Local ephemeris manager pointer to eliminate cascading passes in methods where possible (note some in inherited methods must remain).*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_SpiceEphemeris](#) ()

### 8.1.1 Detailed Description

The `S_define`-level class that provides planetary ephemerides.

The [SpiceEphemeris](#) class constructs the ephemeris reference frame tree and updates the states of the planets based on data from a SPICE file or files.

Definition at line 92 of file `spice_ephem.hh`.

### 8.1.2 Constructor & Destructor Documentation

#### 8.1.2.1 `SpiceEphemeris()` [1/2]

```
jeod::SpiceEphemeris::SpiceEphemeris (  
    void )
```

[SpiceEphemeris](#) default constructor.

Definition at line 66 of file `spice_ephem.cc`.

References `loaded_spk`, `orientation_names`, `planet_names`, and `planetary_orientations`.

#### 8.1.2.2 `~SpiceEphemeris()`

```
jeod::SpiceEphemeris::~~SpiceEphemeris (  
    void )
```

[SpiceEphemeris](#) destructor.

Definition at line 90 of file `spice_ephem.cc`.

References `loaded_spk`, `orientation_names`, `planet_names`, and `planetary_orientations`.

#### 8.1.2.3 `SpiceEphemeris()` [2/2]

```
jeod::SpiceEphemeris::SpiceEphemeris (  
    const SpiceEphemeris & ) [private]
```

Not implemented.

### 8.1.3 Member Function Documentation

## 8.1.3.1 activate()

```
void jeod::SpiceEphemeris::activate (
    void ) [virtual]
```

Nominally, activate the object.

In the case of a [SpiceEphemeris](#) object, an inactive object cannot be activated once the simulation starts. Note that a [SpiceEphemeris](#) is active by default.

Definition at line 123 of file `spice_ephem.cc`.

References inactive.

## 8.1.3.2 add\_barycenter()

```
void jeod::SpiceEphemeris::add_barycenter (
    int id ) [private]
```

Add a barycenter corresponding to the given SPICE ID.

## Parameters

in	<i>id</i>	id of barycenter to add
----	-----------	-------------------------

Definition at line 672 of file `spice_ephem.cc`.

References `barycenter_frames`, `create_new_ephem_point()`, `ephem_mgr_local`, `loaded_spk`, and `MAX_NAME_LENGTH`.

Referenced by `create_barycenters()`.

## 8.1.3.3 add\_descendants\_r()

```
void jeod::SpiceEphemeris::add_descendants_r (
    SpiceEphemPoint * parent ) [private]
```

Add all offspring from a given node of the ref frame tree.

## Parameters

in, out	<i>parent</i>	Frame to attach to
---------	---------------	--------------------

Definition at line 848 of file `spice_ephem.cc`.

References `ephem_mgr_local`, `get_name()`, `jeod::SpiceEphemPoint::get_spice_id()`, and `loaded_spk`.

Referenced by `ephem_build_tree()`.

#### 8.1.3.4 `add_orientation()`

```
void jeod::SpiceEphemeris::add_orientation (
    std::string object_name ) [inline]
```

Definition at line 136 of file `spice_ephem.hh`.

References `orientation_names`.

#### 8.1.3.5 `add_planet_name()`

```
void jeod::SpiceEphemeris::add_planet_name (
    std::string planet_name ) [inline]
```

Definition at line 131 of file `spice_ephem.hh`.

References `planet_names`.

#### 8.1.3.6 `create_barycenters()`

```
void jeod::SpiceEphemeris::create_barycenters (
    void ) [private]
```

Check what if any barycenters need creating and do so.

Any new barycenters are added to the `all_loaded_items` list and registered with the dynamics manager.

Definition at line 705 of file `spice_ephem.cc`.

References `add_barycenter()`, and `loaded_spk`.

Referenced by `initialize_items()`.

#### 8.1.3.7 `create_new_ephem_orientation()`

```
SpiceEphemOrientation * jeod::SpiceEphemeris::create_new_ephem_orientation (
    std::string jeod_name ) [private]
```

Create a new [SpiceEphemOrientation](#).

#### Returns

Pointer to object

## Parameters

in	<i>jeod_name</i>	Name of new object
----	------------------	--------------------

Definition at line 518 of file `spice_ephem.cc`.

References `jeod_2_spice_pfix()`, `populate_item()`, `jeod::SpiceEphemOrientation::set_spice_frame_name()`, `tdb_↵seconds`, and `jeod::SpiceEphemOrientation::validate()`.

Referenced by `process_orientations()`.

## 8.1.3.8 create\_new\_ephem\_point()

```
SpiceEphemPoint * jeod::SpiceEphemeris::create_new_ephem_point (
    std::string object_name,
    std::string spice_name ) [private]
```

Create a new [SpiceEphemPoint](#).

## Returns

Pointer to object

## Parameters

in	<i>object_name</i>	Name of new object
in	<i>spice_name</i>	SPICE lookup name

Definition at line 479 of file `spice_ephem.cc`.

References `find_parent_id()`, `populate_item()`, `jeod::SpiceEphemPoint::set_parent_id()`, and `jeod::SpiceEphem↵Point::set_spice_id()`.

Referenced by `add_barycenter()`, and `process_spk()`.

## 8.1.3.9 deactivate()

```
void jeod::SpiceEphemeris::deactivate (
    void ) [virtual]
```

Deactivate the [SpiceEphemeris](#) object.

Definition at line 140 of file `spice_ephem.cc`.

References `inactive`.

**8.1.3.10 determine\_root\_node()**

```
void jeod::SpiceEphemeris::determine_root_node (
    void ) [private]
```

Determine which item should be the root of the ref frame tree.

Definition at line 752 of file spice\_ephem.cc.

References find\_spice\_id(), loaded\_spk, and root\_item.

Referenced by ephem\_build\_tree(), and initialize\_items().

**8.1.3.11 ephem\_activate()**

```
void jeod::SpiceEphemeris::ephem_activate (
    EphemeridesManager & ephem_manager ) [virtual]
```

Mark appropriate items in the model as active.

**Parameters**

<i>in, out</i>	<i>ephem_manager</i>	Ephemerides manager
----------------	----------------------	---------------------

Definition at line 830 of file spice\_ephem.cc.

**8.1.3.12 ephem\_build\_tree()**

```
void jeod::SpiceEphemeris::ephem_build_tree (
    EphemeridesManager & ephem_manager ) [virtual]
```

Construct the ephemeris model portions of the reference frame tree.

**Parameters**

<i>in, out</i>	<i>ephem_manager</i>	Ephemerides manager
----------------	----------------------	---------------------

Definition at line 888 of file spice\_ephem.cc.

References add\_descendants\_r(), determine\_root\_node(), inactive, loaded\_spk, and root\_item.

**8.1.3.13 ephem\_initialize()**

```
void jeod::SpiceEphemeris::ephem_initialize (
    EphemeridesManager & ephem_manager ) [virtual]
```



Complete the initialization process.

This method should be called after all other ephemeris models have completed their basic initialization and after all planets have registered themselves with the ephemeris manager.

#### Parameters

in, out	<i>ephem_manager</i>	Ephemerides manager
---------	----------------------	---------------------

Definition at line 812 of file spice\_ephem.cc.

#### 8.1.3.14 ephem\_update()

```
void jeod::SpiceEphemeris::ephem_update (
    void ) [virtual]
```

Update ephemerides for subscribed items.

Definition at line 936 of file spice\_ephem.cc.

References dyn\_seconds, force\_update, inactive, update\_rot(), update\_time, and update\_trans().

#### 8.1.3.15 find\_parent\_id()

```
int jeod::SpiceEphemeris::find_parent_id (
    int obj_id ) [private]
```

Find the ID of the parent of a given SPICE object.

Here "parent" means both:

1. the frame to which the object is connected in the JEOD reference frame tree (the "parent" frame).
2. the object that will be used as the "observer" in the calls to SPICE to obtain the state of the given object.

#### Returns

of parent  
Units: ID

#### Parameters

in	<i>obj_id</i>	child ID of which parent is to be found
----	---------------	---

Definition at line 993 of file spice\_ephem.cc.

Referenced by `create_new_ephem_point()`.

#### 8.1.3.16 `find_spice_id()`

```
SpiceEphemPoint * jeod::SpiceEphemeris::find_spice_id (
    int id_to_find )
```

Find a SPICE ID in the loaded list of SPICE objects.

##### Returns

Pointer to object

##### Parameters

in	<i>id_to_find</i>	SPICE lookup name
----	-------------------	-------------------

Definition at line 965 of file `spice_ephem.cc`.

References `loaded_spk`.

Referenced by `determine_root_node()`.

#### 8.1.3.17 `get_name()`

```
const char * jeod::SpiceEphemeris::get_name (
    void ) const [virtual]
```

Return model name.

##### Returns

Name

Definition at line 167 of file `spice_ephem.cc`.

References `ident`.

Referenced by `add_descendants_r()`, and `update_trans()`.

### 8.1.3.18 initialize\_items()

```
void jeod::SpiceEphemeris::initialize_items (
    void ) [private]
```

Initialize the [SpiceEphemeris](#) item data.

Definition at line 537 of file `spice_ephem.cc`.

References `create_barycenters()`, `determine_root_node()`, `introduce_item()`, `loaded_spk`, `planetary_orientations`, and `root_item`.

Referenced by `initialize_model()`.

### 8.1.3.19 initialize\_model()

```
void jeod::SpiceEphemeris::initialize_model (
    const TimeManager & time_manager,
    EphemeridesManager & ephem_manager )
```

Initialize the [SpiceEphemeris](#) model.

This method is called before the planets have been registered with the reference frame manager, so we don't know whether the ephemeris items should be enabled or disabled.

#### Parameters

in	<i>time_manager</i>	Time manager
in	<i>ephem_manager</i>	Incoming ephemeris manager

Definition at line 184 of file `spice_ephem.cc`.

References `ephem_mgr_local`, `inactive`, `initialize_items()`, `initialize_time()`, `load_spice_files()`, `mute_spice_errors()`, `name_barycenter_frames()`, `process_orientations()`, and `process_spk()`.

### 8.1.3.20 initialize\_time()

```
void jeod::SpiceEphemeris::initialize_time (
    const TimeManager & time_manager ) [private]
```

Initialize [SpiceEphemeris](#) timing.

#### Parameters

in	<i>time_manager</i>	Time manager
----	---------------------	--------------

Definition at line 242 of file `spice_ephem.cc`.

References `dyn_seconds`, and `tdb_seconds`.

Referenced by `initialize_model()`.

#### 8.1.3.21 `introduce_item()`

```
void jeod::SpiceEphemeris::introduce_item (
    EphemerisItem & item ) [private]
```

Introduce an `EphemerisItem` to the `EphemeridesManager`.

##### Parameters

in	<i>item</i>	Item to introduce
----	-------------	-------------------

Definition at line 575 of file `spice_ephem.cc`.

References `ephem_mgr_local`.

Referenced by `initialize_items()`.

#### 8.1.3.22 `jeod_2_spice_pfix()`

```
std::string jeod::SpiceEphemeris::jeod_2_spice_pfix (
    std::string jeod_name ) [private]
```

Convert JEOD body name to SPICE pfix frame name.

##### Returns

name of pfix frame that SPICE associates with the given body. Handles high precision frames for Earth and Moon unless IAU is specified by user

##### Parameters

in	<i>jeod_name</i>	Name of the JEOD body.
----	------------------	------------------------

Definition at line 621 of file `spice_ephem.cc`.

Referenced by `create_new_ephem_orientation()`.

#### 8.1.3.23 `load_spice_files()`

```
void jeod::SpiceEphemeris::load_spice_files (
    void ) [private]
```

Load SPICE kernel files containing ephemeris data.

Definition at line 272 of file `spice_ephem.cc`.

References `MAX_MSG_LENGTH`, and `metakernel_filename`.

Referenced by `initialize_model()`.

#### 8.1.3.24 `mute_spice_errors()`

```
void jeod::SpiceEphemeris::mute_spice_errors (
    void ) [private]
```

Mute SPICE errors so that they can be handled by the MessageHandler.

Definition at line 1098 of file `spice_ephem.cc`.

References `MAX_NAME_LENGTH`.

Referenced by `initialize_model()`.

#### 8.1.3.25 `name_barycenter_frames()`

```
void jeod::SpiceEphemeris::name_barycenter_frames (
    void ) [private]
```

Name all the planetary barycenters with their JEOD identifiers.

Definition at line 641 of file `spice_ephem.cc`.

References `barycenter_frames`, `MAX_NAME_LENGTH`, and `spice_2_jeod()`.

Referenced by `initialize_model()`.

#### 8.1.3.26 `operator=()`

```
SpiceEphemeris& jeod::SpiceEphemeris::operator= (
    const SpiceEphemeris & ) [private]
```

Not implemented.

#### 8.1.3.27 `populate_item()`

```
void jeod::SpiceEphemeris::populate_item (
    EphemerisItem & item,
    std::string object_name ) [private]
```

Populate basic attributes of a new `SpiceEphemPoint`.

**Parameters**

<i>in, out</i>	<i>item</i>	Pointer to item to populate
<i>in</i>	<i>object_name</i>	Name of the item

Definition at line 460 of file `spice_ephem.cc`.

Referenced by `create_new_ephem_orientation()`, and `create_new_ephem_point()`.

**8.1.3.28 process\_orientations()**

```
void jeod::SpiceEphemeris::process_orientations (
    void ) [private]
```

Load and locate all planetary orientation frames.

Definition at line 438 of file `spice_ephem.cc`.

References `create_new_ephem_orientation()`, `orientation_names`, and `planetary_orientations`.

Referenced by `initialize_model()`.

**8.1.3.29 process\_spk()**

```
void jeod::SpiceEphemeris::process_spk (
    void ) [private]
```

Process spk objects and store array of loaded spk IDs.

Definition at line 304 of file `spice_ephem.cc`.

References `create_new_ephem_point()`, `loaded_spk`, `MAX_IDS`, `MAX_NAME_LENGTH`, `MAX_PATH_LENGTH`, and `planet_names`.

Referenced by `initialize_model()`.

**8.1.3.30 simple\_restore()**

```
void jeod::SpiceEphemeris::simple_restore (
    void ) [virtual]
```

Set the SPICE model for a restart.

Definition at line 230 of file `spice_ephem.cc`.

**8.1.3.31 spice\_2\_jeod()**

```
std::string jeod::SpiceEphemeris::spice_2_jeod (
    std::string spice_name ) [private]
```

Convert SPICE names to JEOD nomenclature.

**Returns**

JEOD-friendly name of a SPICE name

## Parameters

in	<i>spice_name</i>	Name of a SPICE object.
----	-------------------	-------------------------

Definition at line 603 of file `spice_ephem.cc`.

Referenced by `name_barycenter_frames()`.

8.1.3.32 `timestamp()`

```
double jeod::SpiceEphemeris::timestamp (
    void ) const [virtual]
```

Return time of last update.

## Returns

Timestamp  
Units: day

Definition at line 154 of file `spice_ephem.cc`.

References `update_time`.

8.1.3.33 `update_rot()`

```
void jeod::SpiceEphemeris::update_rot (
    void ) [private]
```

Update planetary orientations.

Definition at line 1082 of file `spice_ephem.cc`.

References `dyn_seconds`, `planetary_orientations`, and `tdb_seconds`.

Referenced by `ephem_update()`.

8.1.3.34 `update_trans()`

```
void jeod::SpiceEphemeris::update_trans (
    void ) [private]
```

Update ephemerides of inertial frames supplied by spk files.

Definition at line 1020 of file `spice_ephem.cc`.

References `get_name()`, `loaded_spk`, `MAX_MSG_LENGTH`, `root_item`, `tdb_seconds`, and `update_time`.

Referenced by `ephem_update()`.

## 8.1.4 Friends And Related Function Documentation

### 8.1.4.1 init\_attrjeod\_\_SpiceEphemeris

```
void init_attrjeod__SpiceEphemeris ( ) [friend]
```

### 8.1.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 94 of file spice\_ephem.hh.

## 8.1.5 Field Documentation

### 8.1.5.1 barycenter\_frames

```
EphemerisRefFrame jeod::SpiceEphemeris::barycenter_frames[10] [private]
```

Array of barycenter frames corresponding to SSBary and the planets.

trick\_units(-)

Definition at line 207 of file spice\_ephem.hh.

Referenced by add\_barycenter(), and name\_barycenter\_frames().

### 8.1.5.2 dyn\_seconds

```
const double* jeod::SpiceEphemeris::dyn_seconds [private]
```

The source of dynamic time information.

trick\_units(-)

Definition at line 222 of file spice\_ephem.hh.

Referenced by ephem\_update(), initialize\_time(), and update\_rot().



#### 8.1.5.3 ephemer\_mgr\_local

```
EphemeridesManager* jeod::SpiceEphemeris::ephemer_mgr_local [private]
```

Local ephemer manager pointer to eliminate cascading passes in methods where possible (note some in inherited methods must remain).

trick\_units(–)

Definition at line 228 of file spice\_ephem.hh.

Referenced by add\_barycenter(), add\_descendants\_r(), initialize\_model(), and introduce\_item().

#### 8.1.5.4 force\_update

```
bool jeod::SpiceEphemeris::force_update [private]
```

Is an update needed even if the time hasn't changed?

trick\_units(–)

Definition at line 172 of file spice\_ephem.hh.

Referenced by ephemer\_update().

#### 8.1.5.5 ident

```
std::string jeod::SpiceEphemeris::ident [private]
```

Identifier for this model, set by the constructor.

trick\_units(–)

Definition at line 197 of file spice\_ephem.hh.

Referenced by get\_name().

#### 8.1.5.6 inactive

```
bool jeod::SpiceEphemeris::inactive [protected]
```

If set to true, makes the model inactive.

This is set to false (i.e., model is active) by the constructor. Setting this flag to true prior to initialization time will result in the model never doing anything. Setting this flag to true after the model has been active for some time turns it off from that point forward. Turning on the model after it has been inactive for some time is not supported. trick\_units(–)

Definition at line 162 of file spice\_ephem.hh.

Referenced by activate(), deactivate(), ephemer\_build\_tree(), ephemer\_update(), and initialize\_model().

#### 8.1.5.7 loaded\_spk

```
JeodPointerVector<SpiceEphemPoint>::type jeod::SpiceEphemeris::loaded_spk [private]
```

All the spk objects tracked by this ephemeris.

trick\_io(\*\*)

Definition at line 187 of file spice\_ephem.hh.

Referenced by `add_barycenter()`, `add_descendants_r()`, `create_barycenters()`, `determine_root_node()`, `ephem↔_build_tree()`, `find_spice_id()`, `initialize_items()`, `process_spk()`, `SpiceEphemeris()`, `update_trans()`, and `~Spice↔Ephemeris()`.

#### 8.1.5.8 metakernel\_filename

```
std::string jeod::SpiceEphemeris::metakernel_filename
```

The name of a text file containing the list of SPICE files to be loaded.

Must be in format needed by SPICE `furnsh_c()` function. See comments for method `load_spice_files()` in `spice_ephem.cc` for format details.`trick_units(-)`

Definition at line 149 of file spice\_ephem.hh.

Referenced by `load_spice_files()`.

#### 8.1.5.9 orientation\_names

```
JeodObjectVector<std::string>::type jeod::SpiceEphemeris::orientation_names [private]
```

The names of all objects for which orientation is required.

trick\_io(\*\*)

Definition at line 182 of file spice\_ephem.hh.

Referenced by `add_orientation()`, `process_orientations()`, `SpiceEphemeris()`, and `~SpiceEphemeris()`.

#### 8.1.5.10 planet\_names

```
JeodObjectVector<std::string>::type jeod::SpiceEphemeris::planet_names [private]
```

The names of all planets to load into the simulation.

trick\_io(\*\*)

Definition at line 177 of file spice\_ephem.hh.

Referenced by `add_planet_name()`, `process_spk()`, `SpiceEphemeris()`, and `~SpiceEphemeris()`.

#### 8.1.5.11 planetary\_orientations

```
JeodPointerVector<SpiceEphemOrientation>::type jeod::SpiceEphemeris::planetary_orientations  
[private]
```

Objects defining planet-fixed reference frames.

trick\_io(\*\*)

Definition at line 192 of file spice\_ephem.hh.

Referenced by initialize\_items(), process\_orientations(), SpiceEphemeris(), update\_rot(), and ~SpiceEphemeris().

#### 8.1.5.12 root\_item

```
SpiceEphemPoint* jeod::SpiceEphemeris::root_item [private]
```

The root point in the reference frame tree.

trick\_units(-)

Definition at line 212 of file spice\_ephem.hh.

Referenced by determine\_root\_node(), ephem\_build\_tree(), initialize\_items(), and update\_trans().

#### 8.1.5.13 tdb\_seconds

```
const double* jeod::SpiceEphemeris::tdb_seconds [private]
```

The source of ephemeris time information.

trick\_units(-)

Definition at line 217 of file spice\_ephem.hh.

Referenced by create\_new\_ephem\_orientation(), initialize\_time(), update\_rot(), and update\_trans().

#### 8.1.5.14 update\_time

```
double jeod::SpiceEphemeris::update_time [private]
```

Time of last update, dynamic time seconds.

trick\_units(s)

Definition at line 202 of file spice\_ephem.hh.

Referenced by ephem\_update(), timestamp(), and update\_trans().

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

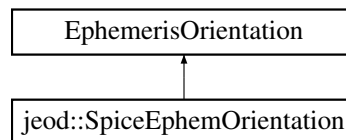
- [spice\\_ephem.hh](#)
- [spice\\_ephem.cc](#)

## 8.2 jeod::SpiceEphemOrientation Class Reference

A [SpiceEphemOrientation](#) minimally extends EphemerisOrientation to include the JEOD and SPICE names and an update method for the target ephemeris reference frame.

```
#include <spice_ephem_orient.hh>
```

Inheritance diagram for jeod::SpiceEphemOrientation:



### Public Member Functions

- [SpiceEphemOrientation](#) ()  
*SpiceEphemOrientation default constructor.*
- virtual [~SpiceEphemOrientation](#) ()  
*SpiceEphemOrientation destructor.*
- void [update](#) (double time\_tdb, double time\_dyn)  
*Update the rotational state of the target frame.*
- void [validate](#) (double time\_tdb)  
*Confirm that the target frame exists in the loaded SPICE kernels.*
- void [get\\_spice\\_transformation](#) (double time\_tdb, double trans6x6[6][6])  
*Populate the SPICE 6 x 6 matrix via sxform\_c().*
- void [set\\_spice\\_frame\\_name](#) (const std::string new\_name)  
*Setter for the name of the SPICE frame.*

### Private Member Functions

- [SpiceEphemOrientation](#) (const [SpiceEphemOrientation](#) &)  
*Not implemented.*
- [SpiceEphemOrientation](#) & [operator=](#) (const [SpiceEphemOrientation](#) &)  
*Not implemented.*

### Private Attributes

- std::string [spice\\_frame\\_name](#)  
*SPICE name of the target reference frame.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_SpiceEphemOrientation](#) ()

### 8.2.1 Detailed Description

A [SpiceEphemOrientation](#) minimally extends EphemerisOrientation to include the JEOD and SPICE names and an update method for the target ephemeris reference frame.

Definition at line 85 of file spice\_ephem\_orient.hh.

### 8.2.2 Constructor & Destructor Documentation

#### 8.2.2.1 SpiceEphemOrientation() [1/2]

```
jeod::SpiceEphemOrientation::SpiceEphemOrientation (
    void )
```

[SpiceEphemOrientation](#) default constructor.

Definition at line 45 of file spice\_ephem\_orient.cc.

#### 8.2.2.2 ~SpiceEphemOrientation()

```
jeod::SpiceEphemOrientation::~~SpiceEphemOrientation (
    void ) [virtual]
```

[SpiceEphemOrientation](#) destructor.

Definition at line 56 of file spice\_ephem\_orient.cc.

#### 8.2.2.3 SpiceEphemOrientation() [2/2]

```
jeod::SpiceEphemOrientation::SpiceEphemOrientation (
    const SpiceEphemOrientation & ) [private]
```

Not implemented.

### 8.2.3 Member Function Documentation

#### 8.2.3.1 get\_spice\_transformation()

```
void jeod::SpiceEphemOrientation::get_spice_transformation (
    double time_tdb,
    double trans6x6[6][6] )
```

Populate the SPICE 6 x 6 matrix via sxform\_c().

**Parameters**

in	<i>time_tdb</i>	Ephem time (TDB)
out	<i>trans6x6</i>	Spice matrix

Definition at line 145 of file `spice_ephem_orient.cc`.

References `spice_frame_name`.

Referenced by `update()`, and `validate()`.

**8.2.3.2 operator=()**

```
SpiceEphemOrientation& jeod::SpiceEphemOrientation::operator= (
    const SpiceEphemOrientation & ) [private]
```

Not implemented.

**8.2.3.3 set\_spice\_frame\_name()**

```
void jeod::SpiceEphemOrientation::set_spice_frame_name (
    const std::string new_name ) [inline]
```

Setter for the name of the SPICE frame.

**Parameters**

<i>new_name</i>	Name of the SPICE frame
-----------------	-------------------------

Definition at line 111 of file `spice_ephem_orient.hh`.

References `spice_frame_name`.

Referenced by `jeod::SpiceEphemeris::create_new_ephem_orientation()`.

**8.2.3.4 update()**

```
void jeod::SpiceEphemOrientation::update (
    double time_tdb,
    double time_dyn )
```

Update the rotational state of the target frame.

**Parameters**

in	<i>time_tdb</i>	Ephemeris time (TDB) Units: s
in	<i>time_dyn</i>	dyn time for timestamp Units: s

Definition at line 69 of file `spice_ephem_orient.cc`.

References `get_spice_transformation()`.

**8.2.3.5 validate()**

```
void jeod::SpiceEphemOrientation::validate (
    double time_tdb )
```

Confirm that the target frame exists in the loaded SPICE kernels.

**Parameters**

in	<i>time_tdb</i>	Ephemeris time (TDB)
----	-----------------	----------------------

Definition at line 123 of file `spice_ephem_orient.cc`.

References `get_spice_transformation()`.

Referenced by `jeod::SpiceEphemeris::create_new_ephem_orientation()`.

**8.2.4 Friends And Related Function Documentation****8.2.4.1 init\_attrjeod\_\_SpiceEphemOrientation**

```
void init_attrjeod__SpiceEphemOrientation ( ) [friend]
```

**8.2.4.2 InputProcessor**

```
friend class InputProcessor [friend]
```

Definition at line 86 of file `spice_ephem_orient.hh`.

## 8.2.5 Field Documentation

### 8.2.5.1 spice\_frame\_name

```
std::string jeod::SpiceEphemOrientation::spice_frame_name [private]
```

SPICE name of the target reference frame.

trick\_units(-)

Definition at line 122 of file spice\_ephem\_orient.hh.

Referenced by `get_spice_transformation()`, and `set_spice_frame_name()`.

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

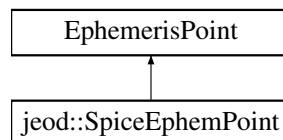
- [spice\\_ephem\\_orient.hh](#)
- [spice\\_ephem\\_orient.cc](#)

## 8.3 jeod::SpiceEphemPoint Class Reference

A [SpiceEphemPoint](#) minimally extends `EphemerisPoint`, primarily in order to contain the ID of the SPICE object that will be used to update the state of the target ephemeris reference frame.

```
#include <spice_ephem_point.hh>
```

Inheritance diagram for `jeod::SpiceEphemPoint`:



### Public Types

- enum [Status](#) { `IsRoot` = 0, `InTree` = 1, `Active` = 2 }

*Enumerates the status values of a [SpiceEphemPoint](#).*



## Public Member Functions

- [SpiceEphemPoint](#) ()  
*SpiceEphemPoint* default constructor.
- virtual [~SpiceEphemPoint](#) ()  
*SpiceEphemPoint* destructor.
- virtual void [set\\_status](#) ([SpiceEphemPoint::Status](#) new\_status)  
Set the active status.
- virtual [SpiceEphemPoint::Status](#) [get\\_status](#) (void) const  
Return current status.
- virtual void [set\\_spice\\_id](#) (int new\_id)  
Set ID of associated SPICE kernel object.
- virtual int [get\\_spice\\_id](#) (void) const  
Return ID of associated SPICE kernel object.
- virtual void [set\\_parent\\_id](#) (int new\_id)  
Set ID of associated parent SPICE kernel object.
- virtual int [get\\_parent\\_id](#) (void) const  
Return ID of associated parent SPICE kernel object.

## Protected Attributes

- [Status](#) status  
The status for the ephemeris reference frame associated with this item.
- int [spice\\_id](#)  
The SPICE kernel object to be used to maintain the target frame's state.
- int [parent\\_id](#)  
The SPICE ID of the parent to this object.

## Private Member Functions

- [SpiceEphemPoint](#) (const [SpiceEphemPoint](#) &)  
Not implemented.
- [SpiceEphemPoint](#) & [operator=](#) (const [SpiceEphemPoint](#) &)  
Not implemented.

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_SpiceEphemPoint](#) ()

### 8.3.1 Detailed Description

A [SpiceEphemPoint](#) minimally extends [EphemerisPoint](#), primarily in order to contain the ID of the SPICE object that will be used to update the state of the target ephemeris reference frame.

Definition at line 84 of file `spice_ephem_point.hh`.

## 8.3.2 Member Enumeration Documentation

### 8.3.2.1 Status

enum `jeod::SpiceEphemPoint::Status`

Enumerates the status values of a `SpiceEphemPoint`.

## Enumerator

IsRoot	The target reference frame is extant, active, and is the root of the reference frame tree. Hence its state is the trivial state.
InTree	The target reference frame is extant and active. However, present responsibility for updating the frame lies with some other ephemeris model.
Active	The target reference frame is extant, active, and is to be updated by this ephemeris model.

Definition at line 94 of file `spice_ephem_point.hh`.

### 8.3.3 Constructor & Destructor Documentation

#### 8.3.3.1 SpiceEphemPoint() [1/2]

```
jeod::SpiceEphemPoint::SpiceEphemPoint (
    void )
```

[SpiceEphemPoint](#) default constructor.

Definition at line 41 of file `spice_ephem_point.cc`.

#### 8.3.3.2 ~SpiceEphemPoint()

```
jeod::SpiceEphemPoint::~~SpiceEphemPoint (
    void ) [virtual]
```

[SpiceEphemPoint](#) destructor.

Definition at line 55 of file `spice_ephem_point.cc`.

#### 8.3.3.3 SpiceEphemPoint() [2/2]

```
jeod::SpiceEphemPoint::SpiceEphemPoint (
    const SpiceEphemPoint & ) [private]
```

Not implemented.

### 8.3.4 Member Function Documentation

#### 8.3.4.1 `get_parent_id()`

```
int jeod::SpiceEphemPoint::get_parent_id (
    void ) const [virtual]
```

Return ID of associated parent SPICE kernel object.

##### Returns

Name

Definition at line 132 of file `spice_ephem_point.cc`.

References `parent_id`.

#### 8.3.4.2 `get_spice_id()`

```
int jeod::SpiceEphemPoint::get_spice_id (
    void ) const [virtual]
```

Return ID of associated SPICE kernel object.

##### Returns

Name

Definition at line 106 of file `spice_ephem_point.cc`.

References `spice_id`.

Referenced by `jeod::SpiceEphemeris::add_descendants_r()`.

#### 8.3.4.3 `get_status()`

```
SpiceEphemPoint::Status jeod::SpiceEphemPoint::get_status (
    void ) const [virtual]
```

Return current status.

##### Returns

Current object status

Definition at line 80 of file `spice_ephem_point.cc`.

References `status`.

#### 8.3.4.4 operator=()

```
SpiceEphemPoint& jeod::SpiceEphemPoint::operator= (
    const SpiceEphemPoint & ) [private]
```

Not implemented.

#### 8.3.4.5 set\_parent\_id()

```
void jeod::SpiceEphemPoint::set_parent_id (
    int new_id ) [virtual]
```

Set ID of associated parent SPICE kernel object.

**Parameters**

in	<i>new</i> ↔ <i>_id</i>	New SPICE ID
----	----------------------------	--------------

Definition at line 119 of file `spice_ephem_point.cc`.

References `parent_id`.

Referenced by `jeod::SpiceEphemeris::create_new_ephem_point()`.

**8.3.4.6 set\_spice\_id()**

```
void jeod::SpiceEphemPoint::set_spice_id (
    int new_id ) [virtual]
```

Set ID of associated SPICE kernel object.

**Parameters**

in	<i>new</i> ↔ <i>_id</i>	New SPICE ID
----	----------------------------	--------------

Definition at line 93 of file `spice_ephem_point.cc`.

References `spice_id`.

Referenced by `jeod::SpiceEphemeris::create_new_ephem_point()`.

**8.3.4.7 set\_status()**

```
void jeod::SpiceEphemPoint::set_status (
    SpiceEphemPoint::Status new_status ) [virtual]
```

Set the active status.

**Parameters**

in	<i>new_status</i>	New status value
----	-------------------	------------------

Definition at line 67 of file `spice_ephem_point.cc`.

References `status`.

**8.3.5 Friends And Related Function Documentation**

### 8.3.5.1 init\_attrjeod\_\_SpiceEphemPoint

```
void init_attrjeod__SpiceEphemPoint ( ) [friend]
```

### 8.3.5.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 85 of file `spice_ephem_point.hh`.

## 8.3.6 Field Documentation

### 8.3.6.1 parent\_id

```
int jeod::SpiceEphemPoint::parent_id [protected]
```

The SPICE ID of the parent to this object.

trick\_units(-)

Definition at line 142 of file `spice_ephem_point.hh`.

Referenced by `get_parent_id()`, and `set_parent_id()`.

### 8.3.6.2 spice\_id

```
int jeod::SpiceEphemPoint::spice_id [protected]
```

The SPICE kernel object to be used to maintain the target frame's state.

trick\_units(-)

Definition at line 137 of file `spice_ephem_point.hh`.

Referenced by `get_spice_id()`, and `set_spice_id()`.

### 8.3.6.3 status

```
Status jeod::SpiceEphemPoint::status [protected]
```

The status for the ephemeris reference frame associated with this item.

trick\_units(-)

Definition at line 132 of file `spice_ephem_point.hh`.

Referenced by `get_status()`, and `set_status()`.

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

- [spice\\_ephem\\_point.hh](#)
- [spice\\_ephem\\_point.cc](#)





## Chapter 9

# File Documentation

### 9.1 spice\_ephem.cc File Reference

Define the methods for the SPICE ephemeris model class.

```
#include <cstdarg>
#include <set>
#include <fstream>
#include <algorithm>
#include "SpiceUsr.h"
#include "environment/ephemerides/ephem_interface/include/ephem_messages.↵
hh"
#include "environment/time/include/time_manager.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/spice_ephem.hh"
```

#### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### Variables

- static const int [MAX\\_PATH\\_LENGTH](#) = 129
- static const int [MAX\\_NAME\\_LENGTH](#) = 33
- static const int [MAX\\_MSG\\_LENGTH](#) = 1841
- static const int [MAX\\_IDS](#) = 1000

#### 9.1.1 Detailed Description

Define the methods for the SPICE ephemeris model class.

## 9.2 spice\_ephem.hh File Reference

Define class for the SPICE ephemeris model.

```
#include <string>
#include "environment/ephemerides/ephem_interface/include/ephem_interface.↵
hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↵
hh"
#include "environment/ephemerides/ephem_manager/include/ephem_manager.hh"
#include "environment/time/include/class_declarations.hh"
#include "utils/container/include/object_vector.hh"
#include "utils/container/include/pointer_vector.hh"
#include "utils/container/include/simple_checkpointable.hh"
#include "utils/ref_frames/include/ref_frame_interface.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "spice_ephem_orient.hh"
#include "spice_ephem_point.hh"
```

### Data Structures

- class [jeod::SpiceEphemeris](#)  
*The S\_define-level class that provides planetary ephemerides.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.2.1 Detailed Description

Define class for the SPICE ephemeris model.

## 9.3 spice\_ephem\_orient.cc File Reference

Define the methods for the SPICE-specific ephemeris orientation class.

```
#include "SpiceUsr.h"
#include "environment/ephemerides/ephem_interface/include/ephem_messages.↵
hh"
#include "utils/math/include/vector3.hh"
#include "utils/math/include/matrix3x3.hh"
#include "../include/spice_ephem_orient.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.3.1 Detailed Description

Define the methods for the SPICE-specific ephemeris orientation class.

## 9.4 spice\_ephem\_orient.hh File Reference

Define class SpiceEphemOrient, which extends EphemerisOrientation for use with the SPICE ephemeris model.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/ephemerides/ephem_item/include/ephem_orient.hh"
```

### Data Structures

- class [jeod::SpiceEphemOrientation](#)

*A [SpiceEphemOrientation](#) minimally extends EphemerisOrientation to include the JEOD and SPICE names and an update method for the target ephemeris reference frame.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.4.1 Detailed Description

Define class SpiceEphemOrient, which extends EphemerisOrientation for use with the SPICE ephemeris model.

## 9.5 spice\_ephem\_point.cc File Reference

Define the methods for the SPICE-specific ephemeris point class.

```
#include <string>
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/spice_ephem_point.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.5.1 Detailed Description

Define the methods for the SPICE-specific ephemeris point class.

## 9.6 spice\_ephem\_point.hh File Reference

Define class SpiceEphemPoint, which extends EphemerisPoint for use with the SPICE ephemeris model.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/ephemerides/ephem_item/include/ephem_point.hh"
```

### Data Structures

- class [jeod::SpiceEphemPoint](#)

A [SpiceEphemPoint](#) minimally extends *EphemerisPoint*, primarily in order to contain the ID of the SPICE object that will be used to update the state of the target ephemeris reference frame.

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.6.1 Detailed Description

Define class SpiceEphemPoint, which extends EphemerisPoint for use with the SPICE ephemeris model.

# Index

- ~SpiceEphemOrientation
  - jeod::SpiceEphemOrientation, [37](#)
- ~SpiceEphemPoint
  - jeod::SpiceEphemPoint, [43](#)
- ~SpiceEphemeris
  - jeod::SpiceEphemeris, [20](#)
- activate
  - jeod::SpiceEphemeris, [20](#)
- add\_barycenter
  - jeod::SpiceEphemeris, [21](#)
- add\_descendants\_r
  - jeod::SpiceEphemeris, [21](#)
- add\_orientation
  - jeod::SpiceEphemeris, [22](#)
- add\_planet\_name
  - jeod::SpiceEphemeris, [22](#)
- barycenter\_frames
  - jeod::SpiceEphemeris, [32](#)
- create\_barycenters
  - jeod::SpiceEphemeris, [22](#)
- create\_new\_ephem\_orientation
  - jeod::SpiceEphemeris, [22](#)
- create\_new\_ephem\_point
  - jeod::SpiceEphemeris, [23](#)
- deactivate
  - jeod::SpiceEphemeris, [23](#)
- determine\_root\_node
  - jeod::SpiceEphemeris, [23](#)
- dyn\_seconds
  - jeod::SpiceEphemeris, [32](#)
- Environment, [12](#)
- ephem\_activate
  - jeod::SpiceEphemeris, [24](#)
- ephem\_build\_tree
  - jeod::SpiceEphemeris, [24](#)
- ephem\_initialize
  - jeod::SpiceEphemeris, [24](#)
- ephem\_mgr\_local
  - jeod::SpiceEphemeris, [32](#)
- ephem\_update
  - jeod::SpiceEphemeris, [25](#)
- find\_parent\_id
  - jeod::SpiceEphemeris, [25](#)
- find\_spice\_id
  - jeod::SpiceEphemeris, [26](#)
- force\_update
  - jeod::SpiceEphemeris, [33](#)
- get\_name
  - jeod::SpiceEphemeris, [26](#)
- get\_parent\_id
  - jeod::SpiceEphemPoint, [43](#)
- get\_spice\_id
  - jeod::SpiceEphemPoint, [44](#)
- get\_spice\_transformation
  - jeod::SpiceEphemOrientation, [37](#)
- get\_status
  - jeod::SpiceEphemPoint, [44](#)
- ident
  - jeod::SpiceEphemeris, [33](#)
- inactive
  - jeod::SpiceEphemeris, [33](#)
- init\_attrjeod\_\_SpiceEphemOrientation
  - jeod::SpiceEphemOrientation, [39](#)
- init\_attrjeod\_\_SpiceEphemPoint
  - jeod::SpiceEphemPoint, [46](#)
- init\_attrjeod\_\_SpiceEphemeris
  - jeod::SpiceEphemeris, [32](#)
- initialize\_items
  - jeod::SpiceEphemeris, [26](#)
- initialize\_model
  - jeod::SpiceEphemeris, [27](#)
- initialize\_time
  - jeod::SpiceEphemeris, [27](#)
- InputProcessor
  - jeod::SpiceEphemOrientation, [39](#)
  - jeod::SpiceEphemPoint, [47](#)
  - jeod::SpiceEphemeris, [32](#)
- introduce\_item
  - jeod::SpiceEphemeris, [28](#)
- jeod, [15](#)
- jeod::SpiceEphemOrientation, [36](#)
- ~SpiceEphemOrientation, [37](#)
- get\_spice\_transformation, [37](#)
- init\_attrjeod\_\_SpiceEphemOrientation, [39](#)
- InputProcessor, [39](#)
- operator=, [38](#)
- set\_spice\_frame\_name, [38](#)
- spice\_frame\_name, [40](#)
- SpiceEphemOrientation, [37](#)
- update, [38](#)
- validate, [39](#)
- jeod::SpiceEphemPoint, [40](#)

- ~SpiceEphemPoint, 43
- get\_parent\_id, 43
- get\_spice\_id, 44
- get\_status, 44
- init\_attrjeod\_\_SpiceEphemPoint, 46
- InputProcessor, 47
- operator=, 44
- parent\_id, 47
- set\_parent\_id, 45
- set\_spice\_id, 46
- set\_status, 46
- spice\_id, 47
- SpiceEphemPoint, 43
- Status, 42
- status, 47
- jeod::SpiceEphemeris, 17
  - ~SpiceEphemeris, 20
  - activate, 20
  - add\_barycenter, 21
  - add\_descendants\_r, 21
  - add\_orientation, 22
  - add\_planet\_name, 22
  - barycenter\_frames, 32
  - create\_barycenters, 22
  - create\_new\_ephem\_orientation, 22
  - create\_new\_ephem\_point, 23
  - deactivate, 23
  - determine\_root\_node, 23
  - dyn\_seconds, 32
  - ephem\_activate, 24
  - ephem\_build\_tree, 24
  - ephem\_initialize, 24
  - ephem\_mgrngr\_local, 32
  - ephem\_update, 25
  - find\_parent\_id, 25
  - find\_spice\_id, 26
  - force\_update, 33
  - get\_name, 26
  - ident, 33
  - inactive, 33
  - init\_attrjeod\_\_SpiceEphemeris, 32
  - initialize\_items, 26
  - initialize\_model, 27
  - initialize\_time, 27
  - InputProcessor, 32
  - introduce\_item, 28
  - jeod\_2\_spice\_pfix, 28
  - load\_spice\_files, 28
  - loaded\_spk, 33
  - metakernel\_filename, 34
  - mute\_spice\_errors, 29
  - name\_barycenter\_frames, 29
  - operator=, 29
  - orientation\_names, 34
  - planet\_names, 34
  - planetary\_orientations, 34
  - populate\_item, 29
  - process\_orientations, 30
- process\_spk, 30
- root\_item, 35
- simple\_restore, 30
- spice\_2\_jeod, 30
- SpiceEphemeris, 20
- tdb\_seconds, 35
- timestamp, 31
- update\_rot, 31
- update\_time, 35
- update\_trans, 31
- jeod\_2\_spice\_pfix
  - jeod::SpiceEphemeris, 28
- load\_spice\_files
  - jeod::SpiceEphemeris, 28
- loaded\_spk
  - jeod::SpiceEphemeris, 33
- MAX\_IDS
  - Spice, 13
- MAX\_MSG\_LENGTH
  - Spice, 13
- MAX\_NAME\_LENGTH
  - Spice, 14
- MAX\_PATH\_LENGTH
  - Spice, 14
- metakernel\_filename
  - jeod::SpiceEphemeris, 34
- Models, 11
- mute\_spice\_errors
  - jeod::SpiceEphemeris, 29
- name\_barycenter\_frames
  - jeod::SpiceEphemeris, 29
- operator=
  - jeod::SpiceEphemOrientation, 38
  - jeod::SpiceEphemPoint, 44
  - jeod::SpiceEphemeris, 29
- orientation\_names
  - jeod::SpiceEphemeris, 34
- parent\_id
  - jeod::SpiceEphemPoint, 47
- planet\_names
  - jeod::SpiceEphemeris, 34
- planetary\_orientations
  - jeod::SpiceEphemeris, 34
- populate\_item
  - jeod::SpiceEphemeris, 29
- process\_orientations
  - jeod::SpiceEphemeris, 30
- process\_spk
  - jeod::SpiceEphemeris, 30
- root\_item
  - jeod::SpiceEphemeris, 35
- set\_parent\_id
  - jeod::SpiceEphemPoint, 45

- set\_spice\_frame\_name
  - jeod::SpiceEphemOrientation, 38
- set\_spice\_id
  - jeod::SpiceEphemPoint, 46
- set\_status
  - jeod::SpiceEphemPoint, 46
- simple\_restore
  - jeod::SpiceEphemeris, 30
- Spice, 13
  - MAX\_IDS, 13
  - MAX\_MSG\_LENGTH, 13
  - MAX\_NAME\_LENGTH, 14
  - MAX\_PATH\_LENGTH, 14
- spice\_2\_jeod
  - jeod::SpiceEphemeris, 30
- spice\_ephem.cc, 49
- spice\_ephem.hh, 50
- spice\_ephem\_orient.cc, 50
- spice\_ephem\_orient.hh, 51
- spice\_ephem\_point.cc, 51
- spice\_ephem\_point.hh, 52
- spice\_frame\_name
  - jeod::SpiceEphemOrientation, 40
- spice\_id
  - jeod::SpiceEphemPoint, 47
- SpiceEphemOrientation
  - jeod::SpiceEphemOrientation, 37
- SpiceEphemPoint
  - jeod::SpiceEphemPoint, 43
- SpiceEphemeris
  - jeod::SpiceEphemeris, 20
- Status
  - jeod::SpiceEphemPoint, 42
- status
  - jeod::SpiceEphemPoint, 47
- tdb\_seconds
  - jeod::SpiceEphemeris, 35
- timestamp
  - jeod::SpiceEphemeris, 31
- update
  - jeod::SpiceEphemOrientation, 38
- update\_rot
  - jeod::SpiceEphemeris, 31
- update\_time
  - jeod::SpiceEphemeris, 35
- update\_trans
  - jeod::SpiceEphemeris, 31
- validate
  - jeod::SpiceEphemOrientation, 39