

SpiceModel

5.1

Generated by Doxygen 1.8.5

Mon Jul 31 2023 11:41:37

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Module Documentation	11
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	13
6.3.2.3	MAX_NAME_LENGTH	13
6.3.2.4	MAX_PATH_LENGTH	14
7	Namespace Documentation	15
7.1	jeod Namespace Reference	15
7.1.1	Detailed Description	15
8	Data Structure Documentation	17
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	20
8.1.2.2	~SpiceEphemeris	20
8.1.2.3	SpiceEphemeris	20
8.1.3	Member Function Documentation	20
8.1.3.1	activate	20
8.1.3.2	add_barycenter	20
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	22
8.1.3.9	deactivate	23
8.1.3.10	determine_root_node	23
8.1.3.11	ephem_activate	23
8.1.3.12	ephem_build_tree	23
8.1.3.13	ephem_initialize	23
8.1.3.14	ephem_update	24
8.1.3.15	find_parent_id	24
8.1.3.16	find_spice_id	24
8.1.3.17	get_name	24
8.1.3.18	initialize_items	25
8.1.3.19	initialize_model	25
8.1.3.20	initialize_time	25
8.1.3.21	introduce_item	25
8.1.3.22	jeod_2_spice_pfix	26
8.1.3.23	load_spice_files	26
8.1.3.24	mute_spice_errors	26
8.1.3.25	name_barycenter_frames	26
8.1.3.26	operator=	26
8.1.3.27	populate_item	26
8.1.3.28	process_orientations	27
8.1.3.29	process_spk	27
8.1.3.30	simple_restore	27
8.1.3.31	spice_2_jeod	27
8.1.3.32	timestamp	27
8.1.3.33	update_rot	28
8.1.3.34	update_trans	28

8.1.4	Friends And Related Function Documentation	28
8.1.4.1	init_attrjeod__SpiceEphemeris	28
8.1.4.2	InputProcessor	28
8.1.5	Field Documentation	28
8.1.5.1	barycenter_frames	28
8.1.5.2	dyn_seconds	28
8.1.5.3	ephem_mgr_local	28
8.1.5.4	force_update	29
8.1.5.5	ident	29
8.1.5.6	inactive	29
8.1.5.7	loaded_spk	29
8.1.5.8	metakernel_filename	29
8.1.5.9	orientation_names	29
8.1.5.10	planet_names	30
8.1.5.11	planetary_orientations	30
8.1.5.12	root_item	30
8.1.5.13	tdb_seconds	30
8.1.5.14	update_time	30
8.2	jeod::SpiceEphemOrientation Class Reference	31
8.2.1	Detailed Description	31
8.2.2	Constructor & Destructor Documentation	32
8.2.2.1	SpiceEphemOrientation	32
8.2.2.2	~SpiceEphemOrientation	32
8.2.2.3	SpiceEphemOrientation	32
8.2.3	Member Function Documentation	32
8.2.3.1	get_spice_transformation	32
8.2.3.2	operator=	32
8.2.3.3	set_spice_frame_name	32
8.2.3.4	update	32
8.2.3.5	validate	33
8.2.4	Friends And Related Function Documentation	33
8.2.4.1	init_attrjeod__SpiceEphemOrientation	33
8.2.4.2	InputProcessor	33
8.2.5	Field Documentation	33
8.2.5.1	spice_frame_name	33
8.3	jeod::SpiceEphemPoint Class Reference	33
8.3.1	Detailed Description	35
8.3.2	Member Enumeration Documentation	35
8.3.2.1	Status	35
8.3.3	Constructor & Destructor Documentation	35

8.3.3.1	SpiceEphemPoint	35
8.3.3.2	~SpiceEphemPoint	35
8.3.3.3	SpiceEphemPoint	35
8.3.4	Member Function Documentation	35
8.3.4.1	get_parent_id	35
8.3.4.2	get_spice_id	36
8.3.4.3	get_status	36
8.3.4.4	operator=	36
8.3.4.5	set_parent_id	36
8.3.4.6	set_spice_id	36
8.3.4.7	set_status	36
8.3.5	Friends And Related Function Documentation	37
8.3.5.1	init_attrjeod__SpiceEphemPoint	37
8.3.5.2	InputProcessor	37
8.3.6	Field Documentation	37
8.3.6.1	parent_id	37
8.3.6.2	spice_id	37
8.3.6.3	status	37
9	File Documentation	39
9.1	spice_ephem.cc File Reference	39
9.1.1	Detailed Description	39
9.2	spice_ephem.hh File Reference	40
9.2.1	Detailed Description	40
9.3	spice_ephem_orient.cc File Reference	40
9.3.1	Detailed Description	41
9.4	spice_ephem_orient.hh File Reference	41
9.4.1	Detailed Description	41
9.5	spice_ephem_point.cc File Reference	41
9.5.1	Detailed Description	41
9.6	spice_ephem_point.hh File Reference	42
9.6.1	Detailed Description	42
Index		43

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:

jeod	Namespace jeod	15
----------------------	--------------------------	--------------------

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	31
EphemerisPoint	
jeod::SpiceEphemPoint	33
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:

jeod::SpiceEphemeris	The S_define-level class that provides planetary ephemerides	17
jeod::SpiceEphemOrientation	A SpiceEphemOrientation minimally extends EphemerisOrientation to include the JEOD and S-PICE names and an update method for the target ephemeris reference frame	31
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 . . .	33

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

spice_ephem.cc	Define the methods for the SPICE ephemeris model class	39
spice_ephem.hh	Define class for the SPICE ephemeris model	40
spice_ephem_orient.cc	Define the methods for the SPICE-specific ephemeris orientation class	40
spice_ephem_orient.hh	Define class SpiceEphemOrient, which extends EphemerisOrientation for use with the SPICE ephemeris model	41
spice_ephem_point.cc	Define the methods for the SPICE-specific ephemeris point class	41
spice_ephem_point.hh	Define class SpiceEphemPoint, which extends EphemerisPoint for use with the SPICE ephemeris model	42

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 `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 `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 `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 `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

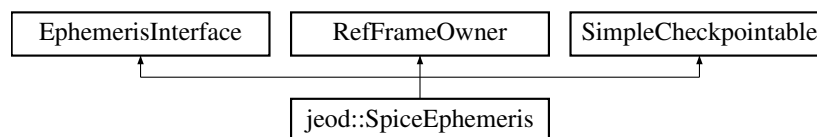
Data Structure Documentation

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) override
SpiceEphemeris destructor.
- void `initialize_model` (const TimeManager &time_manager, EphemeridesManager &ephem_manager)
Initialize the `SpiceEphemeris` model.
- void `activate` (void) override
Nominally, activate the object.
- void `deactivate` (void) override
Deactivate the `SpiceEphemeris` object.
- double `timestamp` (void) const override
Return time of last update.
- const char * `get_name` (void) const override
Return model name.
- void `ephem_initialize` (EphemeridesManager &ephem_manager) override
Complete the initialization process.
- void `ephem_activate` (EphemeridesManager &ephem_manager) override
Mark appropriate items in the model as active.
- void `ephem_build_tree` (EphemeridesManager &ephem_manager) override
Construct the ephemeris model portions of the reference frame tree.
- void `ephem_update` (void) override
Update ephemerides for subscribed items.

- void `simple_restore` (void) override
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, const std::string &name)
Populate basic attributes of a new `SpiceEphemPoint`.
- `SpiceEphemPoint` * `create_new_ephem_point` (std::string object_name, const 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 ephem 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 `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 `jeod::SpiceEphemeris::~~SpiceEphemeris (void) [override]`

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

Not implemented.

8.1.3 Member Function Documentation

8.1.3.1 `void jeod::SpiceEphemeris::activate (void) [override]`

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

Add a barycenter corresponding to the given SPICE ID.

Parameters

<code>in</code>	<code>id</code>	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 void jeod::SpiceEphemeris::add_descendants_r (SpiceEphemPoint * *parent*) [private]

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

Parameters

<i>in, out</i>	<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 `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 `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 `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 `SpiceEphemOrientation * jeod::SpiceEphemeris::create_new_ephem_orientation (std::string jeod_name) [private]`

Create a new [SpiceEphemOrientation](#).

Returns

Pointer to object

Parameters

<i>in</i>	<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 `SpiceEphemPoint * jeod::SpiceEphemeris::create_new_ephem_point (std::string object_name, const 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::SpiceEphemPoint::set_spice_id()`.

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

8.1.3.9 void jeod::SpiceEphemeris::deactivate (void) [override]

Deactivate the [SpiceEphemeris](#) object.

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

References `inactive`.

8.1.3.10 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 void jeod::SpiceEphemeris::ephem_activate (EphemeridesManager & ephem_manager) [override]

Mark appropriate items in the model as active.

Parameters

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

Definition at line 830 of file `spice_ephem.cc`.

8.1.3.12 void jeod::SpiceEphemeris::ephem_build_tree (EphemeridesManager & ephem_manager) [override]

Construct the ephemeris model portions of the reference frame tree.

Parameters

in, out	<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 void jeod::SpiceEphemeris::ephem_initialize (EphemeridesManager & ephem_manager) [override]

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

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

Definition at line 812 of file `spice_ephem.cc`.

8.1.3.14 `void jeod::SpiceEphemeris::ephem_update (void)` [override]

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 `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

<i>in</i>	<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 `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

<i>in</i>	<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 `const char * jeod::SpiceEphemeris::get_name (void) const` [override]

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 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 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 ephem 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 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 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 `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

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

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

Referenced by `create_new_ephem_orientation()`.

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

Not implemented.

8.1.3.27 `void jeod::SpiceEphemeris::populate_item (EphemerisItem & item, const 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 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 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 void jeod::SpiceEphemeris::simple_restore (void) [override]

Set the SPICE model for a restart.

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

8.1.3.31 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

<i>in</i>	<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 double jeod::SpiceEphemeris::timestamp (void) const [override]

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 `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 `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 `void init_attrjeod__SpiceEphemeris () [friend]`

8.1.4.2 `friend class InputProcessor [friend]`

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

8.1.5 Field Documentation

8.1.5.1 `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 `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 `EphemeridesManager* jeod::SpiceEphemeris::ephem_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 `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 `ephem_update()`.

8.1.5.5 `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 `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()`, `ephem_build_tree()`, `ephem_update()`, and `initialize_model()`.

8.1.5.7 `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 `~SpiceEphemeris()`.

8.1.5.8 `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 `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 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 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 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 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 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:

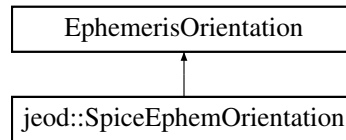
- [spice_ephem.hh](#)
- [spice_ephem.cc](#)

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.
- [~SpiceEphemOrientation](#) () override
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 jeod::SpiceEphemOrientation::SpiceEphemOrientation (void)

[SpiceEphemOrientation](#) default constructor.

Definition at line 45 of file `spice_ephem_orient.cc`.

8.2.2.2 jeod::SpiceEphemOrientation::~~SpiceEphemOrientation (void) [override]

[SpiceEphemOrientation](#) destructor.

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

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

Not implemented.

8.2.3 Member Function Documentation

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

Not implemented.

8.2.3.3 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 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 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 void init_attrjeod__SpiceEphemOrientation () [friend]

8.2.4.2 friend class InputProcessor [friend]

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

8.2.5 Field Documentation

8.2.5.1 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:

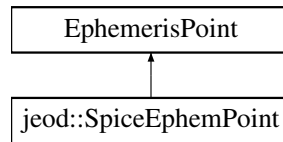
- [spice_ephem_orient.hh](#)
- [spice_ephem_orient.cc](#)

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.
- [~SpiceEphemPoint](#) () override
[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 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 jeod::SpiceEphemPoint::SpiceEphemPoint (void)

[SpiceEphemPoint](#) default constructor.

Definition at line 41 of file spice_ephem_point.cc.

8.3.3.2 jeod::SpiceEphemPoint::~~SpiceEphemPoint (void) [override]

[SpiceEphemPoint](#) destructor.

Definition at line 55 of file spice_ephem_point.cc.

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

Not implemented.

8.3.4 Member Function Documentation

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

Not implemented.

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

Set ID of associated parent SPICE kernel object.

Parameters

<i>in</i>	<i>new_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 `void jeod::SpiceEphemPoint::set_spice_id(int new_id) [virtual]`

Set ID of associated SPICE kernel object.

Parameters

<i>in</i>	<i>new_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 `void jeod::SpiceEphemPoint::set_status(SpiceEphemPoint::Status new_status) [virtual]`

Set the active status.

Parameters

<i>in</i>	<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 `void init_attrjeod_SpiceEphemPoint () [friend]`

8.3.5.2 `friend class InputProcessor [friend]`

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

8.3.6 Field Documentation

8.3.6.1 `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 `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 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.

Definition in file [spice_ephem.cc](#).

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.

Definition in file [spice_ephem.hh](#).

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.

Definition in file [spice_ephem_orient.cc](#).

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.

Definition in file [spice_ephem_orient.hh](#).

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.

Definition in file [spice_ephem_point.cc](#).

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.

Definition in file [spice_ephem_point.hh](#).

Index

- ~SpiceEphemOrientation
 - jeod::SpiceEphemOrientation, [32](#)
- ~SpiceEphemPoint
 - jeod::SpiceEphemPoint, [35](#)
- ~SpiceEphemeris
 - jeod::SpiceEphemeris, [20](#)
- activate
 - jeod::SpiceEphemeris, [20](#)
- Active
 - jeod::SpiceEphemPoint, [35](#)
- add_barycenter
 - jeod::SpiceEphemeris, [20](#)
- add_descendants_r
 - jeod::SpiceEphemeris, [20](#)
- add_orientation
 - jeod::SpiceEphemeris, [22](#)
- add_planet_name
 - jeod::SpiceEphemeris, [22](#)
- barycenter_frames
 - jeod::SpiceEphemeris, [28](#)
- create_barycenters
 - jeod::SpiceEphemeris, [22](#)
- create_new_ephem_orientation
 - jeod::SpiceEphemeris, [22](#)
- create_new_ephem_point
 - jeod::SpiceEphemeris, [22](#)
- deactivate
 - jeod::SpiceEphemeris, [23](#)
- determine_root_node
 - jeod::SpiceEphemeris, [23](#)
- dyn_seconds
 - jeod::SpiceEphemeris, [28](#)
- Environment, [12](#)
- ephem_activate
 - jeod::SpiceEphemeris, [23](#)
- ephem_build_tree
 - jeod::SpiceEphemeris, [23](#)
- ephem_initialize
 - jeod::SpiceEphemeris, [23](#)
- ephem_mgr_local
 - jeod::SpiceEphemeris, [28](#)
- ephem_update
 - jeod::SpiceEphemeris, [24](#)
- find_parent_id
 - jeod::SpiceEphemeris, [24](#)
- find_spice_id
 - jeod::SpiceEphemeris, [24](#)
- force_update
 - jeod::SpiceEphemeris, [29](#)
- get_name
 - jeod::SpiceEphemeris, [24](#)
- get_parent_id
 - jeod::SpiceEphemPoint, [35](#)
- get_spice_id
 - jeod::SpiceEphemPoint, [35](#)
- get_spice_transformation
 - jeod::SpiceEphemOrientation, [32](#)
- get_status
 - jeod::SpiceEphemPoint, [36](#)
- ident
 - jeod::SpiceEphemeris, [29](#)
- InTree
 - jeod::SpiceEphemPoint, [35](#)
- inactive
 - jeod::SpiceEphemeris, [29](#)
- init_attrjeod__SpiceEphemOrientation
 - jeod::SpiceEphemOrientation, [33](#)
- init_attrjeod__SpiceEphemPoint
 - jeod::SpiceEphemPoint, [37](#)
- init_attrjeod__SpiceEphemeris
 - jeod::SpiceEphemeris, [28](#)
- initialize_items
 - jeod::SpiceEphemeris, [25](#)
- initialize_model
 - jeod::SpiceEphemeris, [25](#)
- initialize_time
 - jeod::SpiceEphemeris, [25](#)
- InputProcessor
 - jeod::SpiceEphemeris, [28](#)
 - jeod::SpiceEphemOrientation, [33](#)
 - jeod::SpiceEphemPoint, [37](#)
- introduce_item
 - jeod::SpiceEphemeris, [25](#)
- IsRoot
 - jeod::SpiceEphemPoint, [35](#)
- jeod, [15](#)
- jeod::SpiceEphemPoint
 - Active, [35](#)
 - InTree, [35](#)
 - IsRoot, [35](#)
- jeod::SpiceEphemOrientation, [31](#)
- ~SpiceEphemOrientation, [32](#)

- get_spice_transformation, 32
- init_attrjeod__SpiceEphemOrientation, 33
- InputProcessor, 33
- operator=, 32
- set_spice_frame_name, 32
- spice_frame_name, 33
- SpiceEphemOrientation, 32
- update, 32
- validate, 33
- jeod::SpiceEphemPoint, 33
 - ~SpiceEphemPoint, 35
 - get_parent_id, 35
 - get_spice_id, 35
 - get_status, 36
 - init_attrjeod__SpiceEphemPoint, 37
 - InputProcessor, 37
 - operator=, 36
 - parent_id, 37
 - set_parent_id, 36
 - set_spice_id, 36
 - set_status, 36
 - spice_id, 37
 - SpiceEphemPoint, 35
 - Status, 35
 - status, 37
- jeod::SpiceEphemeris, 17
 - ~SpiceEphemeris, 20
 - activate, 20
 - add_barycenter, 20
 - add_descendants_r, 20
 - add_orientation, 22
 - add_planet_name, 22
 - barycenter_frames, 28
 - create_barycenters, 22
 - create_new_ephem_orientation, 22
 - create_new_ephem_point, 22
 - deactivate, 23
 - determine_root_node, 23
 - dyn_seconds, 28
 - ephem_activate, 23
 - ephem_build_tree, 23
 - ephem_initialize, 23
 - ephem_mgrngr_local, 28
 - ephem_update, 24
 - find_parent_id, 24
 - find_spice_id, 24
 - force_update, 29
 - get_name, 24
 - ident, 29
 - inactive, 29
 - init_attrjeod__SpiceEphemeris, 28
 - initialize_items, 25
 - initialize_model, 25
 - initialize_time, 25
 - InputProcessor, 28
 - introduce_item, 25
 - jeod_2_spice_pfix, 26
 - load_spice_files, 26
 - loaded_spk, 29
 - metakernel_filename, 29
 - mute_spice_errors, 26
 - name_barycenter_frames, 26
 - operator=, 26
 - orientation_names, 29
 - planet_names, 30
 - planetary_orientations, 30
 - populate_item, 26
 - process_orientations, 27
 - process_spk, 27
 - root_item, 30
 - simple_restore, 27
 - spice_2_jeod, 27
 - SpiceEphemeris, 20
 - tdb_seconds, 30
 - timestamp, 27
 - update_rot, 28
 - update_time, 30
 - update_trans, 28
- jeod_2_spice_pfix
 - jeod::SpiceEphemeris, 26
- load_spice_files
 - jeod::SpiceEphemeris, 26
- loaded_spk
 - jeod::SpiceEphemeris, 29
- MAX_IDS
 - Spice, 13
- MAX_MSG_LENGTH
 - Spice, 13
- MAX_NAME_LENGTH
 - Spice, 13
- MAX_PATH_LENGTH
 - Spice, 13
- metakernel_filename
 - jeod::SpiceEphemeris, 29
- Models, 11
- mute_spice_errors
 - jeod::SpiceEphemeris, 26
- name_barycenter_frames
 - jeod::SpiceEphemeris, 26
- operator=
 - jeod::SpiceEphemeris, 26
 - jeod::SpiceEphemOrientation, 32
 - jeod::SpiceEphemPoint, 36
- orientation_names
 - jeod::SpiceEphemeris, 29
- parent_id
 - jeod::SpiceEphemPoint, 37
- planet_names
 - jeod::SpiceEphemeris, 30
- planetary_orientations
 - jeod::SpiceEphemeris, 30
- populate_item

- jeod::SpiceEphemeris, [26](#)
- process_orientations
 - jeod::SpiceEphemeris, [27](#)
- process_spk
 - jeod::SpiceEphemeris, [27](#)
- root_item
 - jeod::SpiceEphemeris, [30](#)
- set_parent_id
 - jeod::SpiceEphemPoint, [36](#)
- set_spice_frame_name
 - jeod::SpiceEphemOrientation, [32](#)
- set_spice_id
 - jeod::SpiceEphemPoint, [36](#)
- set_status
 - jeod::SpiceEphemPoint, [36](#)
- simple_restore
 - jeod::SpiceEphemeris, [27](#)
- Spice, [13](#)
 - MAX_IDS, [13](#)
 - MAX_MSG_LENGTH, [13](#)
 - MAX_NAME_LENGTH, [13](#)
 - MAX_PATH_LENGTH, [13](#)
- spice_2_jeod
 - jeod::SpiceEphemeris, [27](#)
- spice_ephem.cc, [39](#)
- spice_ephem.hh, [40](#)
- spice_ephem_orient.cc, [40](#)
- spice_ephem_orient.hh, [41](#)
- spice_ephem_point.cc, [41](#)
- spice_ephem_point.hh, [42](#)
- spice_frame_name
 - jeod::SpiceEphemOrientation, [33](#)
- spice_id
 - jeod::SpiceEphemPoint, [37](#)
- SpiceEphemOrientation
 - jeod::SpiceEphemOrientation, [32](#)
- SpiceEphemPoint
 - jeod::SpiceEphemPoint, [35](#)
- SpiceEphemeris
 - jeod::SpiceEphemeris, [20](#)
- Status
 - jeod::SpiceEphemPoint, [35](#)
- status
 - jeod::SpiceEphemPoint, [37](#)
- tdb_seconds
 - jeod::SpiceEphemeris, [30](#)
- timestamp
 - jeod::SpiceEphemeris, [27](#)
- update
 - jeod::SpiceEphemOrientation, [32](#)
- update_rot
 - jeod::SpiceEphemeris, [28](#)
- update_time
 - jeod::SpiceEphemeris, [30](#)
- update_trans
 - jeod::SpiceEphemeris, [28](#)
- validate
 - jeod::SpiceEphemOrientation, [33](#)