

SpiceModel

5.1

Generated by Doxygen 1.8.14

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	14
6.3.2.3	MAX_NAME_LENGTH	14
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() [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()	26
8.1.3.19	initialize_model()	27

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

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

9 File Documentation	47
9.1 spice_ephem.cc File Reference	47
9.1.1 Detailed Description	47
9.2 spice_ephem.hh File Reference	48
9.2.1 Detailed Description	48
9.3 spice_ephem_orient.cc File Reference	48
9.3.1 Detailed Description	49
9.4 spice_ephem_orient.hh File Reference	49
9.4.1 Detailed Description	49
9.5 spice_ephem_point.cc File Reference	49
9.5.1 Detailed Description	49
9.6 spice_ephem_point.hh File Reference	50
9.6.1 Detailed Description	50
Index	51

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	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:

jeod::SpiceEphemeris	
The S_define-level class that provides planetary ephemerides	17
jeod::SpiceEphemOrientation	
A SpiceEphemOrientation minimally extends EphemerisOrientation to include the JEOD and SPICE names and an update method for the target ephemeris reference frame	36
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 . . .	40

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	47
spice_ephem.hh	Define class for the SPICE ephemeris model	48
spice_ephem_orient.cc	Define the methods for the SPICE-specific ephemeris orientation class	48
spice_ephem_orient.hh	Define class SpiceEphemOrient, which extends EphemerisOrientation for use with the SPICE ephemeris model	49
spice_ephem_point.cc	Define the methods for the SPICE-specific ephemeris point class	49
spice_ephem_point.hh	Define class SpiceEphemPoint, which extends EphemerisPoint for use with the SPICE ephemeris model	50

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 55 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 54 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 53 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 52 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

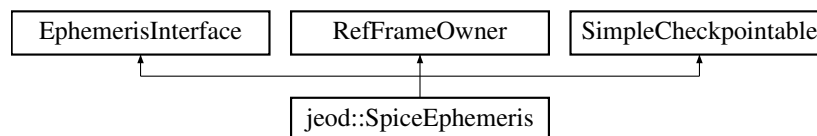
Data Structure Documentation

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](#) ()
SpiceEphemeris default constructor.
- [~SpiceEphemeris](#) () override
SpiceEphemeris destructor.
- [SpiceEphemeris](#) (const [SpiceEphemeris](#) &)=delete
- [SpiceEphemeris](#) & [operator=](#) (const [SpiceEphemeris](#) &)=delete
- void [initialize_model](#) (const TimeManager &time_manager, EphemeridesManager &ephem_manager)
Initialize the [SpiceEphemeris](#) model.
- void [activate](#) () override
Nominally, activate the object.
- void [deactivate](#) () override
Deactivate the [SpiceEphemeris](#) object.
- double [timestamp](#) () const override
Return time of last update.
- std::string [get_name](#) () 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](#) () override
Update ephemerides for subscribed items.
- void [simple_restore](#) () 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](#) ()
Load SPICE kernel files containing ephemeris data.
- void [process_spk](#) ()
Process spk objects and store array of loaded spk IDs.
- void [process_orientations](#) ()
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](#) ()
- *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.*

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](#) {"SPICE"}
- *Identifier for this model, set by the constructor.*
- double [update_time](#) {-99e99}
- *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 91 of file `spice_ephem.hh`.

8.1.2 Constructor & Destructor Documentation

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

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

[SpiceEphemeris](#) default constructor.

Definition at line 64 of file `spice_ephem.cc`.

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

8.1.2.2 `~SpiceEphemeris()`

```
jeod::SpiceEphemeris::~~SpiceEphemeris ( ) [override]
```

[SpiceEphemeris](#) destructor.

Definition at line 76 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 & ) [delete]
```

8.1.3 Member Function Documentation

8.1.3.1 activate()

```
void jeod::SpiceEphemeris::activate ( ) [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 106 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 601 of file `spice_ephem.cc`.

References `barycenter_frames`, `create_new_ephem_point()`, `ephem_mngr_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 756 of file `spice_ephem.cc`.

References `ephem_mngr_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 128 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 121 of file spice_ephem.hh.

References [planet_names](#).

8.1.3.6 create_barycenters()

```
void jeod::SpiceEphemeris::create_barycenters ( ) [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 628 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 468 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,
    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 432 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 ( ) [override]
```

Deactivate the [SpiceEphemeris](#) object.

Definition at line 120 of file `spice_ephem.cc`.

References `inactive`.

8.1.3.10 determine_root_node()

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

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

Definition at line 673 of file spice_ephem.cc.

References find_spice_id(), loaded_spk, and root_item.

Referenced by ephemer_build_tree(), and initialize_items().

8.1.3.11 ephemer_activate()

```
void jeod::SpiceEphemeris::ephemer_activate (
    EphemeridesManager & ephemer_manager ) [override]
```

Mark appropriate items in the model as active.

Parameters

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

Definition at line 744 of file spice_ephem.cc.

8.1.3.12 ephemer_build_tree()

```
void jeod::SpiceEphemeris::ephemer_build_tree (
    EphemeridesManager & ephemer_manager ) [override]
```

Construct the ephemeris model portions of the reference frame tree.

Parameters

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

Definition at line 788 of file spice_ephem.cc.

References add_descendants_r(), determine_root_node(), inactive, loaded_spk, and root_item.

8.1.3.13 ephemer_initialize()

```
void jeod::SpiceEphemeris::ephemer_initialize (
    EphemeridesManager & ephemer_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

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

Definition at line 732 of file spice_ephem.cc.

8.1.3.14 ephem_update()

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

Update ephemerides for subscribed items.

Definition at line 835 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 881 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 857 of file spice_ephem.cc.

References loaded_spk.

Referenced by determine_root_node().

8.1.3.17 get_name()

```
std::string jeod::SpiceEphemeris::get_name ( ) const [override]
```

Return model name.

Returns

Name

Definition at line 138 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 ( ) [private]
```

Initialize the [SpiceEphemeris](#) item data.

Definition at line 482 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 ephem manager

Definition at line 151 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 200 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 518 of file spice_ephem.cc.

References ephemer_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 557 of file spice_ephem.cc.

Referenced by create_new_ephem_orientation().

8.1.3.23 load_spice_files()

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

Load SPICE kernel files containing ephemeris data.

Definition at line 226 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 ( ) [private]
```

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

Definition at line 978 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 ( ) [private]
```

Name all the planetary barycenters with their JEOD identifiers.

Definition at line 575 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 & ) [delete]
```

8.1.3.27 `populate_item()`

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

Populate basic attributes of a new [SpiceEphemPoint](#).

Parameters

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

Definition at line 419 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 ( ) [private]
```

Load and locate all planetary orientation frames.

Definition at line 401 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 ( ) [private]
```

Process spk objects and store array of loaded spk IDs.

Definition at line 252 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 ( ) [override]
```

Set the SPICE model for a restart.

Definition at line 191 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 542 of file `spice_ephem.cc`.

Referenced by `name_barycenter_frames()`.

8.1.3.32 `timestamp()`

```
double jeod::SpiceEphemeris::timestamp ( ) const [override]
```

Return time of last update.

Returns

Timestamp
Units: day

Definition at line 129 of file `spice_ephem.cc`.

References `update_time`.

8.1.3.33 `update_rot()`

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

Update planetary orientations.

Definition at line 967 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 ( ) [private]
```

Update ephemerides of inertial frames supplied by spk files.

Definition at line 906 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 95 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 197 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 212 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 218 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 162 of file spice_ephem.hh.

Referenced by ephemer_update().

8.1.5.5 ident

```
std::string jeod::SpiceEphemeris::ident {"SPICE"} [private]
```

Identifier for this model, set by the constructor.

trick_units(–)

Definition at line 187 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 154 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 177 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 143 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 172 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 167 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 182 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 202 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 207 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 {-99e99} [private]
```

Time of last update, dynamic time seconds.

trick_units(s)

Definition at line 192 of file spice_ephem.hh.

Referenced by ephem_update(), timestamp(), and update_trans().

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

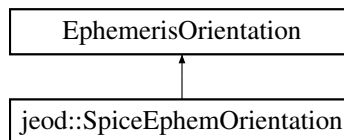
- [spice_ephem.hh](#)
- [spice_ephem.cc](#)

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=default
- [SpiceEphemOrientation](#) (const [SpiceEphemOrientation](#) &)=delete
- [SpiceEphemOrientation](#) & operator= (const [SpiceEphemOrientation](#) &)=delete
- 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 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 83 of file `spice_ephem_orient.hh`.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 SpiceEphemOrientation() [1/2]

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

[SpiceEphemOrientation](#) default constructor.

Definition at line 44 of file `spice_ephem_orient.cc`.

8.2.2.2 ~SpiceEphemOrientation()

```
jeod::SpiceEphemOrientation::~~SpiceEphemOrientation ( ) [override], [default]
```

8.2.2.3 SpiceEphemOrientation() [2/2]

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

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 115 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 & ) [delete]
```

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 105 of file `spice_ephem_orient.hh`.

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 54 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 99 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 85 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 115 of file `spice_ephem_orient.hh`.

Referenced by `get_spice_transformation()`.

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

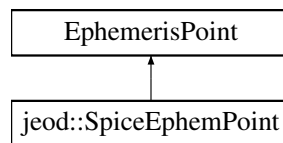
- [spice_ephem_orient.hh](#)
- [spice_ephem_orient.cc](#)

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=default
- [SpiceEphemPoint](#) (const [SpiceEphemPoint](#) &)=delete
- [SpiceEphemPoint](#) & operator= (const [SpiceEphemPoint](#) &)=delete
- virtual void [set_status](#) ([SpiceEphemPoint::Status](#) new_status)
Set the active status.
- virtual [SpiceEphemPoint::Status](#) [get_status](#) () const
Return current status.
- virtual void [set_spice_id](#) (int new_id)
Set ID of associated SPICE kernel object.
- virtual int [get_spice_id](#) () 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](#) () const
Return ID of associated parent SPICE kernel object.

Protected Attributes

- [Status](#) status {[Active](#)}
The status for the ephemeris reference frame associated with this item.
- int [spice_id](#) {32767}
The SPICE kernel object to be used to maintain the target frame's state.
- int [parent_id](#) {32767}
The SPICE ID of the parent to this object.

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 82 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 90 of file `spice_ephem_point.hh`.

8.3.3 Constructor & Destructor Documentation

8.3.3.1 SpiceEphemPoint() [1/2]

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

[SpiceEphemPoint](#) default constructor.

Definition at line 40 of file `spice_ephem_point.cc`.

8.3.3.2 `~SpiceEphemPoint()`

```
jeod::SpiceEphemPoint::~~SpiceEphemPoint ( ) [override], [default]
```

8.3.3.3 `SpiceEphemPoint()` [2/2]

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

8.3.4 Member Function Documentation

8.3.4.1 `get_parent_id()`

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

Return ID of associated parent SPICE kernel object.

Returns

Name

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

References `parent_id`.

8.3.4.2 `get_spice_id()`

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

Return ID of associated SPICE kernel object.

Returns

Name

Definition at line 76 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 ( ) const [virtual]
```

Return current status.

Returns

Current object status

Definition at line 58 of file spice_ephem_point.cc.

References status.

8.3.4.4 operator=()

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

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_id</i>	New SPICE ID
----	---------------	--------------

Definition at line 85 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 67 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 49 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 84 of file spice_ephem_point.hh.

8.3.6 Field Documentation

8.3.6.1 parent_id

```
int jeod::SpiceEphemPoint::parent_id {32767} [protected]
```

The SPICE ID of the parent to this object.

trick_units(–)

Definition at line 134 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 {32767} [protected]
```

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

trick_units(–)

Definition at line 129 of file spice_ephem_point.hh.

Referenced by get_spice_id(), and set_spice_id().

8.3.6.3 status

```
Status jeod::SpiceEphemPoint::status {Active} [protected]
```

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

trick_units(–)

Definition at line 124 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 <algorithm>
#include <cstdint>
#include <fstream>
#include <set>
#include "SpiceUsr.h"
#include "environment/ephemerides/ephem_interface/include/ephem_messages.↵
hh"
#include "environment/time/include/time_manager.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.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/matrix3x3.hh"
#include "utils/math/include/vector3.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 "environment/ephemerides/ephem_item/include/ephem_orient.hh"
#include "utils/sim_interface/include/jeod_class.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 "environment/ephemerides/ephem_item/include/ephem_point.hh"  
#include "utils/sim_interface/include/jeod_class.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, [41](#)
- ~SpiceEphemeris
 - jeod::SpiceEphemeris, [20](#)
- activate
 - jeod::SpiceEphemeris, [20](#)
- add_barycenter
 - jeod::SpiceEphemeris, [21](#)
- add_descendants_r
 - jeod::SpiceEphemeris, [21](#)
- add_orientation
 - jeod::SpiceEphemeris, [21](#)
- 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, [25](#)
- force_update
 - jeod::SpiceEphemeris, [33](#)
- get_name
 - jeod::SpiceEphemeris, [26](#)
- get_parent_id
 - jeod::SpiceEphemPoint, [42](#)
- get_spice_id
 - jeod::SpiceEphemPoint, [42](#)
- get_spice_transformation
 - jeod::SpiceEphemOrientation, [37](#)
- get_status
 - jeod::SpiceEphemPoint, [42](#)
- ident
 - jeod::SpiceEphemeris, [33](#)
- inactive
 - jeod::SpiceEphemeris, [33](#)
- init_attrjeod__SpiceEphemOrientation
 - jeod::SpiceEphemOrientation, [39](#)
- init_attrjeod__SpiceEphemPoint
 - jeod::SpiceEphemPoint, [44](#)
- init_attrjeod__SpiceEphemeris
 - jeod::SpiceEphemeris, [32](#)
- initialize_items
 - jeod::SpiceEphemeris, [26](#)
- initialize_model
 - jeod::SpiceEphemeris, [26](#)
- initialize_time
 - jeod::SpiceEphemeris, [27](#)
- InputProcessor
 - jeod::SpiceEphemOrientation, [39](#)
 - jeod::SpiceEphemPoint, [44](#)
 - jeod::SpiceEphemeris, [32](#)
- introduce_item
 - jeod::SpiceEphemeris, [27](#)
- 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, [39](#)
- SpiceEphemOrientation, [37](#)
- update, [38](#)
- validate, [38](#)
- jeod::SpiceEphemPoint, [40](#)

- ~SpiceEphemPoint, 41
- get_parent_id, 42
- get_spice_id, 42
- get_status, 42
- init_attrjeod__SpiceEphemPoint, 44
- InputProcessor, 44
- operator=, 43
- parent_id, 44
- set_parent_id, 43
- set_spice_id, 43
- set_status, 44
- spice_id, 45
- SpiceEphemPoint, 41, 42
- Status, 41
- status, 45
- jeod::SpiceEphemeris, 17
 - ~SpiceEphemeris, 20
 - activate, 20
 - add_barycenter, 21
 - add_descendants_r, 21
 - add_orientation, 21
 - 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_mgr_local, 32
 - ephem_update, 25
 - find_parent_id, 25
 - find_spice_id, 25
 - force_update, 33
 - get_name, 26
 - ident, 33
 - inactive, 33
 - init_attrjeod__SpiceEphemeris, 32
 - initialize_items, 26
 - initialize_model, 26
 - initialize_time, 27
 - InputProcessor, 32
 - introduce_item, 27
 - jeod_2_spice_pfx, 28
 - load_spice_files, 28
 - loaded_spk, 33
 - metakernel_filename, 34
 - mute_spice_errors, 28
 - 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
 - set_parent_id, 43
 - 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_pfx
 - 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, 28
 - name_barycenter_frames
 - jeod::SpiceEphemeris, 29
 - operator=
 - jeod::SpiceEphemOrientation, 38
 - jeod::SpiceEphemPoint, 43
 - jeod::SpiceEphemeris, 29
 - orientation_names
 - jeod::SpiceEphemeris, 34
 - parent_id
 - jeod::SpiceEphemPoint, 44
 - 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, 43

- set_spice_frame_name
 - jeod::SpiceEphemOrientation, 38
- set_spice_id
 - jeod::SpiceEphemPoint, 43
- set_status
 - jeod::SpiceEphemPoint, 44
- 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, 47
- spice_ephem.hh, 48
- spice_ephem_orient.cc, 48
- spice_ephem_orient.hh, 49
- spice_ephem_point.cc, 49
- spice_ephem_point.hh, 50
- spice_frame_name
 - jeod::SpiceEphemOrientation, 39
- spice_id
 - jeod::SpiceEphemPoint, 45
- SpiceEphemOrientation
 - jeod::SpiceEphemOrientation, 37
- SpiceEphemPoint
 - jeod::SpiceEphemPoint, 41, 42
- SpiceEphemeris
 - jeod::SpiceEphemeris, 20
- Status
 - jeod::SpiceEphemPoint, 41
- status
 - jeod::SpiceEphemPoint, 45
- 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, 38