
oem

Aug 29, 2020

Contents

1	OrbitEphemerisMessage	3
2	Ephemeris Components	7
2.1	EphemerisSegment	7
2.2	HeaderSection	8
2.3	MetaDataSection	9
2.4	DataSection	9
2.5	CovarianceSection	10
3	OEM Data Types	11
3.1	State	11
3.2	Covariance	12
4	Indices and tables	13
	Index	15

The primary interface between the *oem* package and OEM files is the *OrbitEphemerisMessage* class. In addition, the package provides objects to represent each component of an OEM ephemeris.

OrbitEphemerisMessage

See also:

Module *Ephemeris Components*

Module *OEM Data Types*

class oem.OrbitEphemerisMessage (*header, segments*)

Bases: object

Python representation of an Orbit Ephemeris Message.

This class provides the primary interface between the OEM module and an OEM file.

header

Object containing the OEM header section.

Type *HeaderSection*

Examples

The *OrbitEphemerisMessage* class can load directly from a file:

```
>>> ephemeris = OrbitEphemerisMessage.open(file_path)
```

An OEM is made up of one or more data segments available through an iterator:

```
>>> for segment in ephemeris:
...     for state in segment:
...         # Iterate through states
...         pass
...     for covariance in segment.covariances:
...         # Iterate through covariances
...         pass
```

It is also possible to iterate through the states and covariances in all segments with the *.states* and *.covariances* properties.

To determine if a particular epoch is contained in the useable time range of any of the segments in an ephemeris, use *in*:

```
>>> epoch in ephemeris
True
```

To sample a state at an arbitrary epoch, simply call the ephemeris with an astropy Time object

```
>>> epoch = Time("2020-01-01T00:00:00", scale="utc")
>>> ephemeris(epoch)
State(2020-01-01T00:00:00.000)
```

Note that this type of sampling is only supported if the time system of the target ephemeris is supported by astropy Time objects.

The *save_as* method enables saving of copies of an OEM in both KVN and XML formats.

```
>>> oem.save_as("new.oem", file_format="xml")
```

To convert directly between KVN and XML formats, use the *convert* class method. For example, to convert a KVN OEM to XML:

```
>>> oem.convert("input.oem", "output.oem", "xml")
```

classmethod *convert* (*in_file_path*, *out_file_path*, *file_format*)

Convert an OEM to a particular file format.

This method will succeed and produce an output file even if the input file is already in the desired format. Comments are not preserved.

Parameters

- **in_file_path** (*str* or *Path*) – Path to original ephemeris.
- **out_file_path** (*str* or *Path*) – Desired path for converted ephemeris.
- **file_format** (*str*) – Desired output format. Options are ‘kvn’ and ‘xml’.

copy ()

Create an independent copy of this instance.

covariances

Return a list of covariances in all segments.

classmethod *open* (*file_path*)

Open an Orbit Ephemeris Message file.

This method supports both KVN and XML formats.

Parameters **file_path** (*str* or *Path*) – Path of file to read.

Returns New OEM instance.

Return type *OrbitEphemerisMessage*

resample (*step_size*, *in_place=False*)

Resample ephemeris data.

Replaces the existing ephemeris state data in this OEM with new states sampled at the desired sampling interval. The new sampling applies to all segments contained in this OEM.

Parameters

- **step_size** (*float*) – Sample step size in seconds.

- **in_place** (*bool, optional*) – Toggle in-place resampling. Default is False.

Returns

Resampled OEM. Output is an independent instance if `in_place` is True.

Return type *OrbitEphemerisMessage*

Examples

Open an ephemeris file, convert it to a 60-second sampling interval and save the result to a new file:

```
>>> oem = OrbitEphemerisMessage.open("input.oem")
>>> oem.resample(60, in_place=True)
>>> oem.save_as("output.oem")
```

To do the same thing without in-place operations:

```
>>> oem = OrbitEphemerisMessage.open("input.oem")
>>> new_oem = oem.resample(60)
>>> new_oem.save_as("output.oem")
```

save_as (*file_path, file_format='kvn'*)

Write OEM to file.

Parameters

- **file_path** (*str or Path*) – Desired path for output ephemeris.
- **file_format** (*str, optional*) – Type of file to output. Options are 'kvn' and 'xml'. Default is 'kvn'.

states

Return a list of states in all segments.

steps (*step_size*)

Sample Ephemeris at equal time intervals.

This method returns a generator producing states at equal time intervals spanning the useable duration of all segments in the parent OEM.

Parameters **step_size** (*float*) – Sample step size in seconds.

Yields *State* – Sample state.

Examples

Sample states at 60-second intervals:

```
>>> for state in oem.steps(60):
...     pass
```

Note that spacing between steps will only be constant within segments; when crossing from one segment to another the spacing will vary. To avoid this behavior with multi-segment OEMs, use the segment interface directly:

```
>>> for segment in oem:
...     for state in segment.steps(60):
...         pass
```

Ephemeris Components

See also:

Module *OrbitEphemerisMessage*

Module *OEM Data Types*

2.1 EphemerisSegment

class oem.components.**EphemerisSegment** (*metadata, state_data, covariance_data=None, version='2.0'*)

Bases: object

OEM ephemeris segment.

Container for a single OEM ephemeris segment.

copy ()

Create an independent copy of this instance.

covariances

Return list of Covariances in this segment.

has_accel

Evaluate if segment contains acceleration data.

has_covariance

Evaluate if segment contains covariance data.

resample (*step_size, in_place=False*)

Resample ephemeris data.

Replaces the existing ephemeris state data in this EphemerisSegment with a new list of states sampled at the desired sampling interval.

Parameters

- **step_size** (*float*) – Sample step size in seconds.

- **in_place** (*bool*, *optional*) – Toggle in-place resampling. Default is False.

Returns

Resampled EphemerisSegment. Output is an independent instance if `in_place` is True.

Return type *EphemerisSegment*

states

Return list of States in this segment.

steps (*step_size*)

Sample Segment at equal time intervals.

This method returns a generator producing states at equal time intervals spanning the useable duration of the parent EphemerisSegment.

Parameters **step_size** (*float*) – Sample step size in seconds.

Yields *State* – Sampled state.

Examples

Sample states in each segment of an OEM at 60-second intervals:

```
>>> for segment in oem:
...     for state in segment.steps(60):
...         pass
```

useable_start_time

Return epoch of start of useable state data range

useable_stop_time

Return epoch of end of useable state data range

2.2 HeaderSection

class oem.components.HeaderSection (*fields*)

Bases: oem.base.KeyValueSection

OEM header section.

Container for a single OEM header section.

Examples

This class behaves similar to a dict allowing membership checks, iteration over keys, and value set/get.

```
>>> "CCSDS_OEM_VERS" in header:
True
```

```
>>> keys = [key for key in header]
```

```
>>> metadata["ORIGINATOR"] = 'ORIG_NAME'
```

```
>>> metadata["ORIGINATOR"]
'ORIG_NAME'
```

copy()

Create an independent copy of this instance.

version

2.3 MetaDataSection

class oem.components.**MetaDataSection** (metadata, version='2.0')

Bases: oem.base.KeyValueSection

OEM metadata section.

Container for a single OEM metadata section.

Examples

This class behaves similar to a dict allowing membership checks, iteration over keys, and value set/get.

```
>>> "OBJECT_NAME" in metadata:
True
```

```
>>> keys = [key for key in metadata]
```

```
>>> metadata["CENTER_NAME"] = 'Mars'
```

```
>>> metadata["CENTER_NAME"]
'Mars'
```

copy()

Create an independent copy of this instance.

useable_start_time

Return epoch of start of useable state data range

useable_stop_time

Return epoch of end of useable state data range

2.4 DataSection

class oem.components.**DataSection** (states, version='2.0')

Bases: object

OEM data section.

Container for a single OEM ephemeris state data section.

copy()

Create an independent copy of this instance.

has_accel

Evaluate if section contains acceleration data.

states

Return a list of States in this section.

2.5 CovarianceSection

class oem.components.CovarianceSection(*covariances*, *version*='2.0')

Bases: object

OEM covariance section.

Container for a single OEM covariance section.

copy()

Create an independent copy of this instance.

covariances

Return a list of covariances in this section.

See also:

Module *OrbitEphemerisMessage*

Module *Ephemeris Components*

3.1 State

class oem.components.**State** (*epoch, position, velocity, acceleration=None, version='2.0'*)

Bases: object

Basic Cartesian state.

epoch

Epoch date and time.

Type DateTime

position

3-element array describing the position at epoch.

Type ndarray

velocity

3-element array describing the velocity at epoch.

Type ndarray

acceleration

3-element array describing the acceleration at epoch. If unavailable, this attribute is None.

Type ndarray

copy ()

Create an independent copy of this instance.

has_accel

vector

3.2 Covariance

class oem.components.**Covariance** (*epoch, frame, matrix, version='2.0'*)

Bases: object

Basic 6x6 covariance.

epoch

Epoch date and time.

Type DateTime

frame

Reference from of this covariance.

Type str

matrix

6x6 covariance matrix.

Type ndarray

copy ()

Create an independent copy of this instance.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

A

acceleration (*oem.components.State* attribute), 11

C

convert() (*oem.OrbitEphemerisMessage* class method), 4

copy() (*oem.components.Covariance* method), 12

copy() (*oem.components.CovarianceSection* method), 10

copy() (*oem.components.DataSection* method), 9

copy() (*oem.components.EphemerisSegment* method), 7

copy() (*oem.components.HeaderSection* method), 9

copy() (*oem.components.MetaDataSection* method), 9

copy() (*oem.components.State* method), 11

copy() (*oem.OrbitEphemerisMessage* method), 4

Covariance (class in *oem.components*), 12

covariances (*oem.components.CovarianceSection* attribute), 10

covariances (*oem.components.EphemerisSegment* attribute), 7

covariances (*oem.OrbitEphemerisMessage* attribute), 4

CovarianceSection (class in *oem.components*), 10

D

DataSection (class in *oem.components*), 9

E

EphemerisSegment (class in *oem.components*), 7

epoch (*oem.components.Covariance* attribute), 12

epoch (*oem.components.State* attribute), 11

F

frame (*oem.components.Covariance* attribute), 12

H

has_accel (*oem.components.DataSection* attribute), 9

has_accel (*oem.components.EphemerisSegment* attribute), 7

has_accel (*oem.components.State* attribute), 11

has_covariance (*oem.components.EphemerisSegment* attribute), 7

header (*oem.OrbitEphemerisMessage* attribute), 3

HeaderSection (class in *oem.components*), 8

M

matrix (*oem.components.Covariance* attribute), 12

MetaDataSection (class in *oem.components*), 9

O

open() (*oem.OrbitEphemerisMessage* class method), 4

OrbitEphemerisMessage (class in *oem*), 3

P

position (*oem.components.State* attribute), 11

R

resample() (*oem.components.EphemerisSegment* method), 7

resample() (*oem.OrbitEphemerisMessage* method), 4

S

save_as() (*oem.OrbitEphemerisMessage* method), 5

State (class in *oem.components*), 11

states (*oem.components.DataSection* attribute), 9

states (*oem.components.EphemerisSegment* attribute), 8

states (*oem.OrbitEphemerisMessage* attribute), 5

steps() (*oem.components.EphemerisSegment* method), 8

steps() (*oem.OrbitEphemerisMessage* method), 5

U

useable_start_time
(*oem.components.EphemerisSegment* attribute), 8

useable_start_time
 (*oem.components.MetadataSection attribute*),
 9

useable_stop_time
 (*oem.components.EphemerisSegment attribute*), 8

useable_stop_time
 (*oem.components.MetadataSection attribute*),
 9

V

vector (*oem.components.State attribute*), 11

velocity (*oem.components.State attribute*), 11

version (*oem.components.HeaderSection attribute*), 9