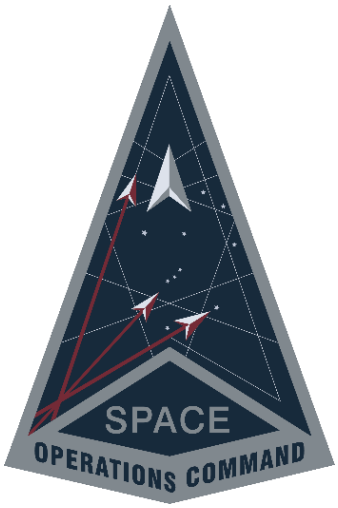
Astrodynamics Standards
Shared Library



Satellite State
(SatState)

Version 9.4

May 2024

Contents

1. Introduction

SatState provides an easy way to interface with the existing libraries - TLE, SpVec, VCM, Sgp4Prop, SpProp, and ExtEphem - to determine the satellites' state at any requested time.

SatState also provides library functions to return ephemeris comparison data between interested satellites. For the users, especially when they might be working with many different orbital element types/propagators/external ephemerides, SatState provides many library functions that help to simplify writing driver code.

If you are on Windows, the shared library files will end in ".dll". For example, "SatState.dll". If you are on Linux, the shared library will begin with "lib" and end in ".so", and will be all lowercase. For example, libsatstate.so.

2. Prerequisites

The following shared libraries MUST be loaded and initialized before using SatState:

- AstroFunc
- DllMain
- ElOps
- EnvConst
- ExtEphem
- Sgp4Prop
- SpProp (optional)
- SpVec
- TimeFunc
- TLE
- VCM

3. Getting Started

To get started, please read the README.txt file that came in the root directory of your distribution. In addition to an overall description contained in the distribution, it has a description of a "**wrapper**".

To get started with **SatState**, there is a "wrapper" specific to SatState, under the **SampleCode** directory. Under your language of choice, you will see a "**DriverExample/wrapper**" subdirectory. The files under this directory will have all the Application Programming Interfaces (APIs) available. For SatState specific APIs, you should see a source file labelled with "SatState" in the file name. This will be where you will find all the APIs for that specific library. The "DriverExample" directory will also contain several examples of applications that should run by simply running the runExample.bat or runExample.sh script. You can use these examples as a starting point for building your application.

If you do not see your programming language under "SampleCode", look in the HTML documentation for the APIs. Open a browser to the "Documentation/APIDocs/index.html" file. This document will show all the APIs regardless of programming language.

The Astrodynamics Standards libraries should work with any language capable of using Dynamic Link Library (on Windows) or Shared object (on Linux) files.

4. Understanding SATSTATE

SatState is designed to be called by other libraries (LAMOD, COMBO, ROTAS, ...) that request satellites' state at any requested time. Whenever a satellite state is requested, usually by passing the time and the satellite's satKey, the SatState will call the appropriate libraries to access the satellite's data and compute the satellite's state. If the satellite's input data is of orbital element types, SatState calls the appropriate propagator (SGP4 or SP) to propagate the satellite; if the satellite's input data is external ephemeris, the SatState calls the ExtEphem's interpolator to interpolate the external ephemeris data. In short, SatState provides a generic and simple way for returning satellite states to the requesting applications internally or externally (the end users).

Unlike other libraries, SatState does not have its own binary tree. However, it has access to other libaries' shared data so that whenever requested, it can compute satellite states.

5. Calling SATSTATE

The only API that the users have to call is SatStateInit. Depending on the users' needs, the rest of the Application Programming Interfaces (APIs) might be used in their applications.