HQ Space Operations Command (SpOC)
S3/6Z
Astrodynamics Standards Engineering Group

Astrodynamics Standards

Release Notes

Version 8.1

May 2021

Contents

Figures

1. Background

Version 8.1 (V8.1) is a minor release of the U.S. Space Force, Space Operations Command, Astrodynamics Standards software library. The Astro Standards are delivered as a collection of Dynamic Link Libraries (DLLs) for Windows and Shared Objects (SOs) for Linux. The libraries can be run on either 32- or 64-bit platforms, and version releases include wrappers and drivers to support a variety of customer/user environments including C#; C/C++; FORTRAN; Java/JNA; Python-2,3; MATLAB; VB.net; and most recently Julia, with this latest V8.1 Release. Within this document, the term Library is used to refer to either a Windows DLL or a Linux SO. The Library algorithms are designed to be compatible with systems and astrodynamics algorithms implemented into space operations and used by Warfighters and Analysts, including those of the Combined Space Operations Center (CSpOC) Battle Space Awareness (BSA) mission. The Astro Standards are also used to Verify and Validate (V&V) equivalent algorithms of these operational space-domain systems such as those that run at the 18th Space Control Squadron (18th SpCS) at Vandenberg AFB, and other operational locations critical to the National defense.

2. What's New

There are many new Application Programming Interfaces (APIs) that take advantage of the architecture change that was delivered in V8.0 and depicted in **Figure 2**. These new interfaces allow users to make a single API call instead of the chaining of APIs that was required in the previous version. There are also modified or new APIs in the TLE and Obs libraries that allow users to get, or set, the new fields of the csv format. There are fixes in the drivers for most languages. There is a new-supported language, Julia. There are also minor fixes to the documentation.

The most noticeable change is the decrease in runtime of many libraries. Some libraries have up to tenfold improvement in performance. There are also APIs that allow a bulk call to the SGP4/XP propagator to reduce calls to the API. This has a small benefit for most programming languages but allows around a two- to six-times improvement for Java/JNA.

For the SGP4/SGP4-XP Propagator:

1. The fastest way to obtain SGP4/SGP4-XP is by creating an account on https://www.space-track.org, and downloading it directly from there. No approval is required, but permissions will need to be granted by the administrators of space-track.org.
2. SGP4 is one unique Astro Standards library in the suite of Astro Standards libraries available in that it is U.S. Space Force, Space Operations Command-approved to "share with the world."

Other Applications within the Astro Standards Library (including SGP4/SGP4-XP):

1. For the balance of the Astro Standard Applications, use https://halfway.peterson.af.mil/SARP. The requestor must have a U.S. Government-issued CAC card and be logged into *NIPRnet*. This website cannot be accessed from the Internet.
2. Once logged-in to https://halfway.peterson.af.mil/SARP obtain additional details by referring to the document, "***Instructions for Requesting Astrodynamics Standards Software.pdf,"*** available upon logging into the SARP website.

Figure 1. Astodynamic Standards Distribution

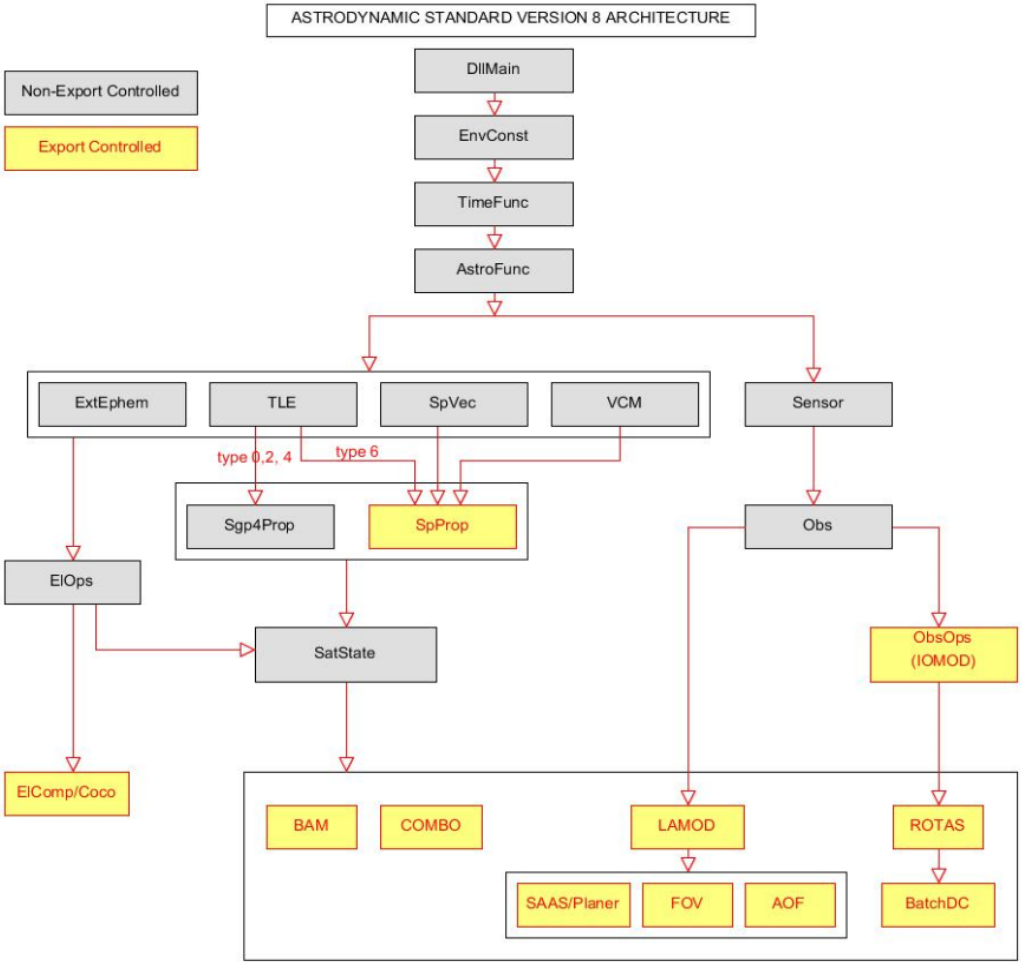

Figure 2. Astrodynamic Standards Version 8 Architecture

3. Tally of Bug Fixes / Improvements for Releases 8.1 and 8.0

Item	Current Release 8.1	Previous Release 8.0
Bug Fixes	*33*	*77*
New Features /Improvements	*79*	*113*
Target / Final Release Date	*Wed May 26, 2021*	*Mon November 9, 2020*

4. Astro Standards Bug Fixes and Enhancements, By Library

The following subsections are listed alphabetically by each of the 24 Astro Standards libraries for quick reference. The precise library name is shown before the "-" in the section heading and corresponds directly to the library names used in **Figure 2** of the new architecture for Version 8.0, and applicable to V8.1 as well.

4.1 AOF - Area Overflight

AOF computes when overhead satellites have potential visibility to a geographic location or area on the surface of the Earth. Visibility is defined as a nominal Field of View (FOV), defined by a user-specified half-angle around the satellite's sub-point intersecting the defined points or areas on the surface.

NO CHANGES

4.2 AstroFunc - Astrodynamics Functions

This library includes various Astrodynamics functions for orbital element conversions; coordinate transformations and reference-frame transformations.

CHANGES:

a) Fix inefficiency in calculation of right ascension and declination.

NEW APIS:

1) ***ECRtoEFG***: Rotates ECR position and velocity to EFG coordinate system
2) ***EFGtoECR***: Rotates ECR position and velocity to EFG coordinate system
3) ***XYZtoLLH***: Converts ECI of Date Position to Latitude, Longitude, and Height above ellipse.
4) ***LLHtoXYP***: Converts Latitude, Longitude, and Height above ellipse to ECI of Date Position.
5) ***EFGtoECI***: Rotates EFG position and velocity to ECI TEME of Date coordinate system
6) ***ECItoEFG***: Rotates ECI TEME of Date position and velocity to EFG coordinate system

7) ***AzEltoRADecTime***: Converts Azimuth and Elevation to Right Ascension and Declination without user needing to determine
8) ***RaDecToAzElTime***: Converts Azimuth and Elevation to Right Ascension and Declination without user needing to determine
9) ***ECItoEFG***: Rotates ECI TEME of Date position and velocity to EFG coordinate system
10) ***EFGtoECI***: Rotates EFG position and velocity to ECI TEME of Date coordinate system

4.3 BAM – Breakup Analysis Module

The Breakup Analysis Module library uses the last good element set of the parent object along with user-selected best-quality element sets of a few of the breakup pieces. The post-breakup piece element sets are propagated backward in time, and the pre-breakup parent element set is propagated forward in time, to identify the "pinch-point" where the piece and parent element sets converge positionally in time. This convergence point corresponds to the time the breakup of the parent most likely occurred.

NO CHANGES

4.4 BatchDC - Batch Differential Correction (DC) Orbit Determination

BatchDC Performs a least-squares batch differential correction of orbital elements using empirical tracking data (sensor observations). It updates either SGP4 Keplerian elements (18th SPCS TLE) or SP state vectors (from a 18th SPCS VCM) using the appropriate propagator theory.

CHANGES:

a) Add new epoch placement options: beginning of day of epoch, beginning of day of last obs, and beginning of day of epoch

NEW APIS:

1) ***SpToCsv:*** Performs batch-least-square differential corrections to the specified satellite (VCM/SPVEC) and returns the corrected elements SGP4/SGP4-XP in the form of a csv-formatted element set.

4.5 Combo - Computation of Miss Between Orbits

Combo computes close approaches between satellite orbits based on user-specified screening volume, exclusion volume, and warning and alert thresholds expressed in either standoff radius or asset-centered UVW (radial, in-track, and cross-track) miss-distance criteria. Precomputed SGP4, SGP4-XP, or SP-based ephemerides are used in the evaluation. ***Note:*** SGP4 GP-based Combo results are appropriate for general understanding of the frequency of approaches between objects and coarse assessment of expected miss distances between objects, but *not* appropriate for collision-avoidance decisions. Only high-accuracy SP ephemerides that include

propagated covariance resulting in computed Probability of Collision P_c should be used for collision-avoidance decisions.

NO CHANGES

4.6 DllMain - Main Library

NO CHANGES

4.7 ElComp - Element Comparison, including Computation of Co-Orbital

Combines capabilities of Computation of Co-Orbital (previously the Coco library) into the Element Comparison library (ElComp), with the ElOps Library dependency.

ElComp includes Computation of Coplanar Orbits (COCO). The COCO algorithm calculates the difference in orbital plane between a pair of orbits defined by element sets. COCO includes evaluation of relative nodal rates between the two objects and reports when (how many days hence) the two object's ascending nodes will align.

ElComp performs Element Comparison. Element comparison is used to determine the degree to which two orbits are Same, Close, Similar, or None. Determination of Same, Close, Similar, or None is by user-specified comparison thresholds for the primary orbit versus the secondary for which differences are evaluated in inclination, right ascension of ascending node, perigee height, eccentricity, orbital period, and argument of perigee.

NO CHANGES

4.8 ElOps - Element Operations

Includes various orbital element operations that do not require a propagator.

NO CHANGES

4.9 EnvConst - Environmental Constants Utility

This library is used for loading and manipulating various Earth constants and FK data.

NO CHANGES

4.10 ExtEphem - External Ephemeris

NO CHANGES

4.11 FOV - Field of View

FOV determines times in which orbiting satellites fly through a ground-based observer's conical field of view. The field of view can be defined by a constant azimuth and elevation boresight, a constant right ascension and declination boresight, or as a line-of-site to an orbiting satellite. The input orbit descriptions may be either a SGP4 TLE, a SP VCM, or an externally generated ephemeris file.

NO CHANGES

4.12 Lamod - Look Angle Module

The Lamod Library computes sensor (ground-based or space-based) viewing opportunities ("look angles") for Earth-centered satellite orbits. The input orbit description may be either a SGP4 TLE, a SP Vector, a SP VCM, or an externally generated ephemeris file.

CHANGES:

a) Added ability to generate csv observations in LamodGenObs and LamodGenObsAtTime

b) Fixed performance issue introduced in V8.0.

4.13 Obs – Observations

This library is used for loading and manipulating observations including observations in B3 format, external TTY ASCII format, or comma-separated value csv format.

CHANGES:

a) Add new fields in xa_obs and xf_obs and modified APIs that call to retrieve or set csv data not in the B3

4.14 ObsOps - Observation Operations

Includes the Initial Orbit Module (IOMOD) capability to compute an initial set of orbital elements (18th SPCS TLE) from as few as three sensor observations, or four observations to include solution for the drag model parameter. The ObsOps library also allows users to manipulate observations and derive useful information such as Latitude and Longitude from Right Ascension, Declination and Height. Notably, the ObsOps library now has the capability to compute satellite range through a triangulation technique using two simultaneous angles-only tracks.

CHANGES:

a) Initial Orbit Determination is more thread safe.

4.15 Rotas - Report Association, Observation/Element-Set Association

Associates sensor observations against satellite element sets using the same algorithms used by the 18th SPCS. It computes observation residuals, compares the residuals against delta-height,

delta-in-track, and delta-beta residuals and assigns the ASTAT value (association status) to be either 1, 2, 3, or 4. ASTAT 1 corresponds to FULL, meaning observations fall within the un-multiplied height, in-track, and cross-track residuals. ASTAT 2 corresponds to CLOSE, meaning the observations residuals fall inside the cross-track limit and within the multiplied radial and in-track limits. ASTAT 3 corresponds to PLANE meaning the observation residuals meet the cross-track limit but fall outside the multiplied radial or multiplied in-track limits. ASTAT 4 corresponds to NONE meaning the observation residuals fall entirely outside the multiplied association box. The result of ASTAT 4 NONE is the observations cannot be tagged and are reported as Uncorrelated Track Observations (UCTs), meaning those observations must be correlated and tagged to another object.

NO CHANGES

4.16 SAAS - Space Attack Assessment Software

For a specified launch location, profile, and maximum kill altitude, computes the kill ring and identifies all object's orbits that will penetrate the kill ring (entry or exit or both and the corresponding satellite numbers and times). Identified satellites only include those that orbit less than or equal to the maximum altitude of the launch capability for a specified launch forecast period of interest.

NO CHANGES

4.17 SatState - Satellite State

The SatState library is used for handling a mix of different types of predictions and for extracting derived information such as semi-major axis, minimum perigee and maximum apogee, or computation of Earth-Centered Inertial (ECI) positions from a ground sensor location (either Latitude Longitude and Height, or Earth-Centered Rotating position).

CHANGES:

a) Added new “One Step” APIs

NEW APIS:

1) ***SatStateEphCom_OS***: Same as SatStateEphemCom but without involving satkeys

4.18 Sensor – Sensor Processing

The Sensor library is used for loading sensors and defining sensor limits that define the coverage type, Field of Regard and Field of View.

CHANGES:

a) Allow ASW sensor file to load without sigma/bias data.

4.19 Sgp4Prop - SGP4 Propagator, extended to include SGP4-XP Capabilities

The Sgp4Prop library includes analytic propagation methods based on general perturbations (GP) theory. Sgp4Prop is used for generating ephemerides for satellites in Earth-centered orbits. It is the appropriate means for propagating orbital state using input TLEs for sensor-tasking purposes. SGP4 perturbations account for non-sphericity of Earth, Sun and Moon perturbative accelerations, and atmospheric drag according to Jacchia 1970 static density tables. The new "XP" version of SGP4 is appropriate for applications that require SP-level accuracy. The Extended Perturbations version of SGP4 is referred to as "SGP4-XP".

The propagation accuracy now available by SGP4-XP is vastly improved by inclusion of extended perturbations that can be tapped by using the new TLE ephemeris type 4, which replaces the legacy $\ddot{n}/6$ term in columns 45-52 of Line 1 with the object's solar radiation pressure model parameter, *AGOM* [m^2/kg]. The leading decimal point is assumed for the AGOM value in columns 45-52. As previously summarized in Figure 3, extended perturbations includes better lunar perturbation modeling, new and more resonance modeling, solar radiation pressure modelling (in addition to drag modelling), and the Geopotential model has been extended to include the J5 zonal term and legacy WGS-72 terms are replaced with EGM-96 terms to be consistent with SP propagation by the operational ASW. In addition, the static atmosphere model is replaced with the Jacchia-70 model that uses a generic solar flux and geomagnetic index that predicts future flux based on flux periodics dating back more than 40 years. Therefore, it is not necessary to maintain $F_{10.7}$ and A_p values for running the SGP4-XP propagator.

CHANGES:

a) SGP4-XP now used EGP-96 instead of global Earth Model.

NEW APIS:

2) ***Sgp4ReepochCsv***: Propagates a csv element set to a new epoch.
3) ***Sgp4GenEphems***: Propagates a satkey to multiple times and returns and array of corresponding positions and velocities
4) ***Sgp4GenEphems_OS:*** Propagates a TLE Array to multiple times without loading into memory and returns and array of corresponding positions and velocities.
5) ***Sgp4PropDs50UtcPosVel:*** Only returns Position and Velocity. Slightly speeds up return time
6) ***Sgp4PropAllSats:*** Propagates all requested satellites to specified time

4.20 SpProp - SP Propagator

The SpProp library uses a Special Perturbations (SP) theory to generate ephemerides from ECI state vectors produced by the BatchDC orbit determination library. SP theory includes perturbations accelerations due to Sun and Moon and other third-bodies (planets); non-sphericity of Earth including up to 70x70 zonals and tesserials in the EGM-96 geopotential model; dynamic calibration of the atmosphere (DCA); Earth-tides and ocean tides; and other techniques that improve orbit determination and prediction accuracy.

CHANGES:

a) Fix bug that happens when user uses flux data that stored in a separate file. This flux data wasn't cleared out correctly and therefore produced inconsistent results between runs

b) Fix bug when memory priority mode is used

c) Fix bug when covariance is all zeros

d) Allow SP to use SP elset's own Earth model instead of the global

4.21 SpVec - SP Vector

The SpVec library is used for loading SP Vectors.

NO CHANGES

4.22 TimeFunc – Time Functions

The TimeFunc library manages various time types and conversions among various time types such as Time Atomic International (TAI), Universal Time Corrected, TAI minus UTC offset (leap second), UT1 Rate, and determination of Greenwich hour angle, also known as Theta Greenwich.

NO CHANGES

4.23 TLE - Two-Line Element Set Processing

The TLE library is used for loading operational Two-Line Element sets (TLEs) and comma-separated value (csv) formatted orbital elements.

CHANGES:

a) Allow csv format in TleLinesToArray

b) Allow csv format in TleParseGP

NEW APIS:

1) ***TleGpArrayToCsv***: Converts the GP Array into a csv formatted element set.
2) ***SetTleKeyMode*:** Allows duplicate TLEs to be loaded in memory.
3) ***GetTleKeyMode:*** AllowsReturns the "mode"

4.24 VCM - Vector Covariance Message Processing

The VCM library is used for parsing and loading information from Vector Covariance Messages (VCMs).

CHANGES:

a) Added ability to read VCM-Lite from file. For users with access, these are in the SP_Vec folder on space-track.org.

5. Future Capabilities and Changes

- Astro Standards currently uses an AVL tree to store and search for satellites represented by "keys" in memory. In the next release, this construct will be replaced by memory pointers. There will be an estimated 8 - 20% improvement in performance as well as better thread-safety.
- Add Position, Partials, and Time Version 3 (PPT3) Navy propagator to Sgp4Prop in a future Release. This will allow Astro Standards to be compatible with the Navy theory. This will also allow creation of PPT3 elements. These will be distinguished by *element set type "3"*.
- Ability to use Right Ascension and Declination Rates in ROTAS and BatchDC.
- Release of unit tests along with the new Release.
- ***Other possible Future additions***:

▪ Breakup Analysis Tool	▪ Full CI / CD deploy to Jenkins	▪ Time, P, V => Acceleration
▪ Clean-up GEN Wrappers	▪ Parallel BatchDC	▪ TDOA/FDOA
▪ SGP4-XP version Decay	▪ Individual P-Card Controls	▪ Web-Assembly

6. Past Releases of the Astrodynamics Standards

- v8.0 - Full release of on Windows (32/64) and Linux (32/64): 09 Nov 2020
- V7.9 - Full release of on Windows (32/64) and Linux (32/64): 15 May 2019
- V7.8.1 – Full non-beta release of on Windows (32/64) and Linux (32/64): 24 July 2018
- V7.8 - Full non-beta release of on Windows (32/64) and Linux (32/64): 15 June 2017
- V7.7 - Public Release of SGP4 on Windows (32/64) and Linux (32/64): 15 March 2016
- V7.beta6 - SGP4, SP, LAMOD, COMBO, ROTAS, IOMOD, BATCHDC, Linux (32/64): 28 Oct 2014
- V7.beta6 - SGP4, SP, LAMOD, COMBO, ROTAS, IOMOD, BATCHDC, Windows (32/64): 27 Oct '14
- V7.beta5 - SGP4, SP, LAMOD, COMBO on Windows(32/64): 25 August 2014
- V7.beta4 - SGP4, SP, LAMOD, COMBO on Linux(32/64): 25 September 2012
- V7.beta4 - SGP4, SP, LAMOD, COMBO on Windows(32/64): 17 September 2012
- V7.beta - COMBO on Windows: 28 October 2011
- V7.beta1 - LAMOD on Windows: 28 Jun 2011
- V7.beta - LAMOD on Windows: 02 Jun 2011
- V7.beta3 - SGP4, SP on Windows: 21 March 2011
- V7.beta2 - SGP4, SP on Linux: 06 October 2010
- V7.beta1 - SGP4, SP on Windows: 27 July 2010
- V7.beta - SGP4, SP on Windows: 08 December 2008

I. Index of V8.1 Astro Standards Bug Fixes and Improvements, by Jira Number

Commented [SZMGUAA1]: Do we want to include the Jiras of things that are unrelated to the users? I.e. WinWin and things related to the build process.

Remediations or Improvements for the following sets of JIRAs *have been implemented* for Astro Standards V8.1.

I-1. Version 8.1 List of Bug Fixes by JIRA Number:

1) *ASTROSTDS-150*: *TLE TleDataToArray / TleLinesToArray*. In TLE, the tests of TleDataToArray and TleLinesToArray now return 0 in element 30 of the numerical array (Element set number). It used to return 984 for the case below..The string output from these calls is also blank and used to return U98021G:

```
1 25291U 98021G   12262.34124505 -.00000362  00000-0 -13643-3 0  9843
2 25291 086.3927 228.2309 0002335 077.3451 282.8054 14.34209529756626
```

2) *ASTROSTDS-151*: *TleGPArrayToLines*. In TLE TleGPArrayToLines appears to be missing data on the first line. It looks like it's not including data from the xs_tle parameter.
   Before: `1 25291U 98021G   12262.34124505 -.00000362  00000 0  10000-1 2 0345`
   Now: `1 25291U          12262.34124505 -.00000362  00000 0  10000-1 2 0000`

3) *ASTROSTDS-152*: *TleUpdateSatFrArray*. In TLE, the test of TleUpdateSatFrArray uses the string "Cxxxxx UabcdDqqzYUSA Cv" to update a TLE, and this fails to update the security classification from U to C.

4) *ASTROSTDS-257*: *Need to redesign the way how the propagator is called. Current design doesn't work very well in multi-thread environment*. Astro Standards applications like (Lamod, Combo, et. al.) that need to call a propagator (SGP4/SP/ExternalEphem) needs to traverse the appropriate binary tree. This can be inefficient when there are many threads trying to access the propagator object at the same time. To avoid this problem, each thread should have its own prop object.

5) *ASTROSTDS-395*: *Investigate the slowness of LAMOD when using orbiting sensor*. LAMOD runs much slower when orbiting sensor elsets are SP elsets. Investigated issue and found that the slowness was due to huge SP data was copied to the caller. Replaced returning actual SP data with a pointer to it fixed the issue.

6) *ASTROSTDS-397*: *Fix GenDllWrappers to handle multiline method comments*. In GenDllWrappers repository, FuncPrototype.cs file, around line 241:
   varDescription = readLine.Substring(readLine.IndexOf("!") + 1).Trim();
   This line detects the function comments, but only once. Need to allow multiple line comments, so the javadoc generated can be more descriptive.

7) *ASTROSTDS-398*: *Allow ASW sensor file to be load without sigma/bias data.* The MITRE Corporation reported that some manually generated sensor files (using ASW format which allows more than 3-digit sensor number) failed to load. Investigation showed that these generated files don't follow the ASW official format which requires sigma/bias and the "Source of Reference" as the end of record indicator Need to modify SensorCardReading.f90 to relax this rule so that sensor location data can be loaded without other sensor data like sigma/bias/limits.

8) *ASTROSTDS-430*: *Fix bash scripts and directory permissions in Linux distributions*. Upon unzipping files on Linux systems, the bash scripts have an embedded control character. Also, the directories are missing the execute permission. The astestpreparation.sh script was written to mitigate this, but this script itself has the control character problem and throws permission denied errors due to the missing execute permission on the directories.

9) *ASTROSTDS-438*: *SatStateGetSatDataAll has the wrong API documentation*. Need to fix GenDllWrappers.exe. It is generating the same API documentation for "perigeeHt" and "perigee". Same for "apogeeHt" and "apogee". Once fixed, verify javadocs for JnaSatState class is generating the unique API documentation for each of these properties for the SatStateGetSatDataAll method.

10) *ASTROSTDS-439*: *Checking for obsType twice for same value in SensorDll Service.c*. See SensorDll_Service.c file. GetObsTypeStr(char obsType) method. It's checking for "obsTyp == '6'", twice.

11) *ASTROSTDS-440*: *SensorWrappers.cs SensorGetLines method has wrong argument type*. SensorWrappers.cs class has SensorGetLines method with the wrong 2nd argument type. The type should be "byte[]", but is generated as "out char"

12) *ASTROSTDS-443*: *Checking for senLim1L.elLim1 != 0 twice in SensorDll_Services.h*. In SensorDll_Services.h, PrintSensorData() Line 306:

```
 else if (senLim1L.viewType == '2')
{
    If (senLim1L.azLim1 != 0 && senLim1L.azLim2 != 0 && senLim1L.elLim1 != 0 &&
senLim1L.elLim1 != 0)   line 308  (*** should be corrected to senLimi1L.elLim2 ***)
        {
```

13) *ASTROSTDS-445*: *Allow SGP4-XP to use EGM-96 Earth model instead of the global on*. SGP4-XP will use EGM-96 for all related calculations regardless whatever Earth model is selected in the global settings.

14) *ASTROSTDS-447*: *Fix SP propagation bug when "memory priority" mode is used*. SP crashes with this error message when "memory priority" is selected:
forrtl: severe (151): allocatable array is already allocated
Stack trace terminated abnormally.

15) *ASTROSTDS-448*: *Fix SP propagation bug when all zeros covariance matrix was used*. Lockheed Martin reported that his MATLAB program crashed when trying to propagate a VCM with all zeros covariance matrix. The issue was that the SpHasCovMtx didn't check if the input "cov" was allocated. "cov" field was used to be a static 9x9 array but it was changed to allocatable in v8.0. Not checking whether "cov" is allocated caused the crash.

16) *ASTROSTDS-458*: *Fix memory leak in SpToEGP*. We suspect there's a memory leak in the SpToEGP method in the BatchDCDllExports.f90 file. It could be in the call to ExtrapolationGP method.

17) *ASTROSTDS-477*: *VCM loader fails when reading in a time with seconds=60*. When a VCM is created from a system that handles microseconds (e.g.. 59.9995), the VCM will round up (e.g.. 60.000) to milliseconds. V7.0 was able to handle this, but V8.1 fails. So when loading a VCM, the time field should allow more than 59 seconds, more than 59 minutes, more than 24 hours, more than the number of days in a specified month. When it goes over these times, it should update the next time field (i.e.. 60 seconds will be 1 min, 0 seconds, 60 min will be 1 hr, 0 min, etc).

18) *ASTROSTDS-479*: *Getting error in Julia driver example on Windows*. Getting the following warning message, but since it exits, it is handled as an error. Either fix the warning or exit with 0 so the runExample.bat script exits normally without error. [14:37:45] [Step 1/1] cmd.exe : â"Œ Warning: Assignment to `iErrCode` in soft scope is ambiguous because a global variable by the same name [14:37:45] [Step 1/1] cmd.exe : â"Œ Warning: Assignment to `iErrCode` in soft scope is ambiguous because a global variable by the same name [14:37:45] [Step 1/1] exists: `iErrCode` will be treated as a new local. Disambiguate by using `local iErrCode` to suppress this warning or [14:37:45] [Step 1/1] `global iErrCode` to assign to the existing global variable.[14:37:45] [Step 1/1] At C:\TeamCity\buildAgent\work\f6d9555a71058333\buildscripts\RunDriverExamples.ps1:81 char:2[14:37:45] [Step 1/1] cmd.exe /c runExample.bat[14:37:45] [Step 1/1] + ~~~~~~~~~~~~~~~~~~~~~~~~~~[14:37:45] [Step 1/1] CategoryInfo : NotSpecified: (â"Œ Warning: As...lobal variable.:String) [], RemoteException[14:37:45] [Step 1/1] + FullyQualifiedErrorId : NativeCommandError[14:37:45] [Step 1/1] [14:37:45] [Step 1/1] â"" @ C:\TeamCity\buildAgent\work\f6d9555a71058333\DriverExamples\Sgp4Prop\jl_SGP4prop_simple.jl:17

19) *ASTROSTDS-481*: *BatchDC free format doesn't have an entry to specify DC method*. BatchDC free format doesn't have an entry to specify DC method. The only way the user can specify this option is through the use of DC's P-card.

20) *ASTROSTDS-483*: *BatchDC free format doesn't have an entry to specify DC method - Update BatchDC document*. BatchDC free format doesn't have an entry to specify DC method. The only way the user can specify this option is through the use of DC's P-card.

21) *ASTROSTDS-484*: *Javadoc is skipping the first java source file*. Not sure why, but javadoc doesn't generate the documentation for the first file specified in the sourcepath argument.

22) *ASTROSTDS-487*: *Fix validation routine for B3/type 4 obs when rate fields using overpunch J-Rs*. When the same TLE and SENSORSN.DAT files are loaded through the WinWin tool, used just the one sensor (Clear) to generate an obs file, checked the output type, and compared the B3Obs at around the same time. Both have J values, seem to be around the same values in other parameters, and have the same formats:

ATSAT:

```
U900011832008316084300157154 3098568 42478111 -023718 00577 J0848 00000  4 0  19000190001
U900011832008316085300156128 2902318 42966411 0120490 -2533 J0317 00000  4 0  19000190001
```

WinWin:

```
U900011832008316083875856620 3182017 42710181 -084996 01904 J0579 00000  4 0  00000000000
U900011832008316084475857202 3063294 42460321 0001901 00001 J0872 00000  4 0  00000000000
U900011832008316085075856623 2944627 42733221 0088724 -1898 J0568 00000  4 0  00000000000
```

However, when loading the observation file that WinWin created back into WinWin it rejects the B3Obs that have these J values. Also ran these WinWin obs through ROTAS with the C wrapper we are using, and the obs with J values were rejected again. We are using SAAL 7.9 for our sim, and I'm using 8.0 for WinWin. We don't believe that the B3Obs from ATSAT are wrong, but need to determine why both are rejected.

If load a different file with opticals they are all accepted when they have J-R for declination:

```
U900019732008118262100 5J80764 2301064                                  5 0  19000190001
U900019732008118263100 5K07195 2302246                                  5 0  19000190001
```

So it seems that this is an issue with radar observations and their rates.

23) *ASTROSTDS-500*: *BatchDCWrapper.BatchDCLoadCard breaks with column 7 in VCM is 3, 2, or 4*.

All of our engines are matching up neatly with the previous version EXCEPT BatchDC. With no changes to our code, the BatchDC test now returns "CheckConvergence: Error! Cannot recover DC." for every single test.

The test (which you may remember from 3 years ago) is to propagate a VCM for a day, set those as obs and then run a BatchDC on the VCM. We should get back a VCM very close to the original. I dug WAY down in the history and found the attached sample code you had sent us. Our current BatchDC code is extremely similar to this, but I noted one change that testing suggests is the crux of the problem:

```
BatchDCWrapper.BatchDCLoadCard("     1          1111111                0099       1    P");
```

If put that BatchDC card into our code, things work and a VCM is produced. It breaks when column 7 is changed to a '3' instead of a '1'. If I'm reading the documentation correctly, '1' indicates to produce a VCM epoched to the last obs. A '3' indicates to produce a VCM epoched to the epoch of the original VCM. Of note, '5' and '6' also produce VCMs as expected. '2' and '4' run forever when I keep columns 9-23 blank – I'm guessing because it's trying to propagate back to 1950 or whatever happens to be in memory.

It's a little sloppy, but using a '4' and setting column 9-23 to the VCM epoch does give the expected result.

24) *ASTROSTDS-505*: *Relax rule for TLE CSV format so the first character can be blank or ','*. Current rule requires the first character of a TLE CSV to be 'U', 'C', or 'S'. Need to relax this rule so the first

character can be blank or nothing (that means the first character can also be comma). This is to allow converting TLE to CSV without problem when TLE's classification is blank.

25) *ASTROSTDS-506*: *Fix bug in Sgp4ReepochTle()*. The Sgp4ReepochTle() doesn't use the correct earth constants for reepoching the elset therefore the computation is less accurate.

26) *ASTROSTDS-508*: *Generate API Documentation not triggering*. The "Generate API Documentation" TeamCity configuration is not kicking off automatically even though there's a VCS and finish build trigger. Changed branch for VCS trigger to "*" and finish dependency to "<default>" and added a snapshot dependency for Generate Wrappers.

27) *ASTROSTDS-511*: *Remove all "STOP" statements from Fortran code*. These "STOP" statements would be really bad since they stop the program instantly and this behavior can be unpredictable when being called from a driver program.

28) *ASTROSTDS-526*: *Fix bug in SatStateGetCovUVW() when covariance matrix doesn't exist or is all zeros*.

29) *ASTROSTDS-539*: *Fix bug in FovTargetElset()*. When Source and Victim are the same, FovTargetElset() should return an error code. However, the errCode wasn't set so whatever its uninitialized value was returned.

30) *ASTROSTDS-540*. *CompSunMoonPos(), CompSunPos(), CompMoonPos(), IsPointSunLit() are now using FK5 instead of FK4*. Spectr mode which is on when FK4 is picked. However it was mistakenly turned on when FK5 was actually picked.

31) *ASTROSTDS-543*. *Clarify that sensor location is in ECR coordinate frame in sensor document*. Here's what was stated in the document for sensor position: X-Geocentric (meters), Y-Geocentric (meters), Z-Geocentric (meters). Clearly the information was confusing. Changed the wordings to clarify the coordinate frame: ECR position X (meters), ECR position Y (meters), ECR position Z (meters).

32) *ASTROSTDS-544*. *Correct description for XF_SENLOC_POSX, XF_SENLOC_POSY, XF_SENLOC_POSZ named constants*. Here's what was stated in the document for sensor position: X-Geocentric (meters), Y-Geocentric (meters), Z-Geocentric (meters). Clearly the information was confusing. Changed the wordings to clarify the coordinate frame: ECR position X (meters), ECR position Y (meters), ECR position Z (meters).

33) *ASTROSTDS-546*. *Fix bug in BatchDC's C-driver code*. Remove unused parameters from PrintAcceptCrit().

I-2. Version 8.1 List of New Features / Improvements by JIRA Number:

1) *ASTROSTDS-362*. *Create Java driver example for Aof*. mimic C_Aof driver example.
2) *ASTROSTDS-384*: *New function to convert ECR to EFG using time in days since 1950 UTC*.
3) *ASTROSTDS-385*. *New function to convert EFG to ECR using time in days since 1950 UTC*.
4) *ASTROSTDS-386*. *New function to convert XYZ to LLH using time in days since 1950 UTC*.
5) *ASTROSTDS-387*. *New function to convert LLH to XYZ using time in days since 1950 UTC*.
6) *ASTROSTDS-388*. *New function to convert EFG to ECI using time in days since 1950 UTC*.
7) *ASTROSTDS-389*. *New function to convert ECI to EFG using time in days since 1950 UTC*.
8) *ASTROSTDS-390*. *New function to convert RA/Dec to Az/El using time in days since 1950 UTC*.
9) *ASTROSTDS-391*. *New function to convert Az/El to RA/Dec using time in days since 1950 UTC*.
10) *ASTROSTDS-392*. *Allow individual obs's sensor sigmas/biases data (along with other data) to also be returned via API calls*.
11) *ASTROSTDS-393*. *Remove Matlab/Fortran/VB/C# driver examples for SGP4 and SP*. Matlab/Fortran/VB/C# driver examples are being copied to all package distributions. It should be copied to all distributions except Sgp4Prop, Sgp4Prop_small, and SpProp. Sgp4Prop is distributed to everyone, so no need for everyone to see all driver examples. Same issue for Sp.
12) *ASTROSTDS-394*. *Migrate all TeamCity configurations to use Synology file share.* All the scripts in TeamCity need to have the DistDir pointing to the Synology file share. Should be a single change on TeamCity, but there might be other references. Must be a Windows and Linux parameter change.
13) *ASTROSTDS-400*. *Fix inefficiency in ComputeRADec*. Removed inefficiency caused by dividing a cos by the numerator and denominator.
14) *ASTROSTDS-401*. *Create javadocs without JNA class references*. Create javadocs without JNA class references because the javadocs are used by all users regardless of language they use, so having classes called "Jnaxxxx" will be confusing and are specific to Java. However, we don't want to remove the Jna from the Java wrappers that are already created because existing users are currently using those classes, and changing this will cause their code to break, not to mention our Java driver examples.
15) *ASTROSTDS-403*. *Move/remove Ghost doc comments*. Remove all the "start doc-example" Ghost doc blocks, and move the "start doc-remarks" to the method comment.
16) *ASTROSTDS-404*. *Move logging to inside conditional statements*. The Fortran code has write commands that will write to log files regardless if needed or not, which causes the process to be slower than necessary. Investigate all Fortran code. See SGP4 code as an example.
17) *ASTROSTDS-407*. *Add a new interface to the DLL propagator to build ephemerides*. Most of the time when the propagator is called, the user is interested in lots of points; so, for eample, if one needs 1,440 points, say every minute in a 24-hour period, he will need to make 1,440 calls to the DLL. Instead, provide a method like this: getEphemeris ([OEI or satkey],start,step,points,px[],py[],pz[],vx[],vy[],vz[]), where the user user only needs to call the method once and get back the entire 1,440 points.

18) *ASTROSTDS-409*. *Create new API to calculate delta w/o satkey*. Similar to SatStateEphCom, but allow user to pass in 2 satellite states instead of satkeys. Added SatStateEphCom_OC() to allow this capability.

19) *ASTROSTDS-410*. *Fix comments to "beta"*. Delta "beta", by definition, is the difference in orbital plane in degrees, which is delta angular momentum. However, "beta" in the code is actually an estimate of beta, when we don't have full state obs. This "beta" is calculated with asin(dot(priU, secW)), whereas actual beta is calculated with acos(dot(priW, secW)). Need to update the javadocs for all "beta" fields with this clarification, which is in ROTAS, BATCHDC, and SATSTATE, ...at least. Added "asin(dot(priU, secW))" to comments for XA_OBSRES_BETA, and "acos(dot(priW, secW))" for XA_OBSRES_ANGMOM in RotasTypes.f90 to clarify the difference between "beta" and angular momentum.

20) *ASTROSTDS-419*. *Add API with the option to turn off unneeded code*. Some users call the SGP4 propagator, but with some of the calculations not necessary, but is taking up time to return a response. Need to add the option to turn off those calculations. Not sure if this only applies to the SGP4 propagator.

21) *ASTROSTDS-422*. *Add csv format to LamodGenObs and LamodGenObsAtTime*. Added ability to generate csv obs in Lamod, BindC branch.

22) *ASTROSTDS-423*. *Add SpToCsv*.

23) *ASTROSTDS-424*. *Add ObsFieldsToCsv*.

24) *ASTROSTDS-425*. *Allow csv format in TleLinesToArray*.

25) *ASTROSTDS-426*. *Allow csv format in TleParseGP*.

26) *ASTROSTDS-427*. *add TleGpFieldsToCsv*.

27) *ASTROSTDS-428*. *add TleGpArrayToCsv*.

28) *ASTROSTDS-429*. *New epoch placement flags*. Allow placement at beginning of day of last obs, beginning of day of epoch, and nodal crossing prior to epoch.

29) *ASTROSTDS-431*. *Upgrade to latest JNA jar file*. Current version of jna.jar is 3.4.0. Current version is at 5.70.

30) *ASTROSTDS-432*. *Add Sgp4CsvReepoch*. Add API to re-epoch to a csv formatted element set.

31) *ASTROSTDS-433*. *Add more options for XF_OBS and XA_OBS*. Create more options for xa_obs and xf_obs.

32) *ASTROSTDS-434*. *Add ability to read VCM-Lite from file*. VCM-Lite exists on space-track to provide a more compact VCM without covariance. This allows users to read those in directly.

33) *ASTROSTDS-435*. *Add SGP4 threading tests to TeamCity*. 1. Create Java classes that puts SGP4 propagations in multiple threads. 2. Create scripts that build and run the example. 3. Have the scripts run from TeamCity. Multiple threads are run and for each thread, a formatted string of positions are printed to a string buffer. Then a hash is created from the buffer. Then a comparison is run to ensure all hashes are the same, which would mean that each thread created the exact same positions regardless of thread. If a different hash is detected, the program exits with -1 and TeamCity will show the test as failed.

34) *ASTROSTDS-436*. *Remove hardcoded "V8.0" and "//astrofs..." in build scripts*. Previously, the hardcoded strings were default values, but this causes an error if we move to new versions and file shares when testing from command line. These values work for TeamCity because the configurations use the environment variables set. Removing these environment variables will force the users to ensure the version and file share are set correctly.

35) *ASTROSTDS-437*. *Add logic to allow dynamic/variable step sizes in propagators*. This is mainly for Highly Elliptical Orbits where the orbit is very fast at perigee and significantly slower deltas toward apogee. This was asked by ISSA and Space Fence.

36) *ASTROSTDS-444*. *Allow SP to use SP elset's own Earth model instead of using the global one*. In previous version, SP was using the global Earth constants regardless what Earth model was selected in SP elset. This isn't totally correct.

37) *ASTROSTDS-446*. *Allow SGP4 to always use WGS-72 regardless what was set in the global Earth model*. In v8.0 and prior, there was an enforcement that WGS-72 must be selected for SGP4 to work. This enforcement was removed in v8.1 so SGP4 was no longer tied to what was set in the global Earth model and it still uses WGS-72.

38) *ASTROSTDS-449*. *Propagate all input satellites to the time expressed in days since 1950 for SGP4*. Create a Sgp4PropAllSats method. Propagate all input satellites, represented by their satKeys, to the time expressed in days since 1950, UTC for SGP4.

39) *ASTROSTDS-450*. *Generate ephemerides for input satellite for time and step size for SGP4*. Create a Sgp4GenEphems method. Generate ephemerides for the input satellite, represented by its satkey, for a specified time span and step size for SGP4.

40) *ASTROSTDS-451*. *Generate ephemerides for input satellite for time and step size for MT for SGP4*. Create a Sgp4GenEphems_MT method. Generate ephemerides for the input satellite, represented by its satkey, for a specified time span and step size for SGP4 for Multithreaded applications.

41) *ASTROSTDS-452*. *Propagate satellite and store pos and vel in given parameters for SGP4*. Create a Sgp4PropDs50UtcPosVel method. Propagates a satellite, represented by the satKey, to the time expressed in days since 1950, UTC for SGP4. The resulting data will be stored in the pos and vel parameters provided.

42) *ASTROSTDS-461*. *Add Julia wrappers*. Add Julia Language wrappers. This includes modifying GenDllWrappers application to generate the wrappers, creating the driver examples for various packages, and adding the configuration to TeamCity to build and test the package. This is only for 64-bit implementations in Windows and Linux. Downloaded 32 and 64 bit for Windows and Linux from https://julialang.org/downloads/. Copied to SYNOLOGYFS server under S36Share/S36/ASTRO/Software/Julia.

43) *ASTROSTDS-468*. *Handling of 9-digit satkey and 8-digit sensor key*. Determine if 9-digit satkey and 8-digit sensor key work with larger numbers. Need to convert senNum and satNum to integer 8 so when the satKey is created, those numbers are preserved.

44) *ASTROSTDS-473*. *Add multi-thread tests in SGP4*. Create an application to test if SGP4Prop can be run in a multithreaded environment. Have the application automatically launched by TeamCity.

45) *ASTROSTDS-474*. *Add multi-thread tests in SP*. Create an application to test if SpProp can be run in a multithreaded environment. Have the application automatically launched by TeamCity.

46) *ASTROSTDS-476*. *Add note in documentation about license file being pre-requisite*. Add a note in the documentation about having the SGP4_Open_License.txt file located in the run directory and/or DLL/SO directory in order to use the libraries. Suggest putting this note in the Pre-Requisites section.

47) *ASTROSTDS-478*. *Allow users to specify the location of the DLLs at runtime when running unit tests*. When debugging, developers must set the location of the DLLs in utility.cs, setting the ASTROSTDDLLPATH constant. This constant contains the directory where the DLLs are located, but at compile time. The alternative is to set the directory to the current directory. You cannot specify any other directory at runtime. Implemented different ways to specify the path to Astro Standards folder at run time. One uses a new API to set the path; another one uses environment variable LD_LIBRARY_PATH to set that path.

48) *ASTROSTDS-480*. *Investigate speed performance of JNI vs JNA*. Suggest first convert SGP4Prop to JNI. This task must also include all the effort it will take to modify GenDllWrappers.exe to generate JNI code and the process it will take to create JNI vs JNA.

49) *ASTROSTDS-485*. *Create performance test between ISSA and AstroStds*. Create performance test to compare AstroStds code against ISSA. Added ISSAPerformance, ISSAPropagator, ASPerformance, ASPropagatorMSE, ASPropagatorPosVel, and ASPropagatorGenEphem classes in issa_test repo.

50) *ASTROSTDS-486*. *Remove HTML tags from wrappers for all languages except Java*. To mimic the formatting of GhostDocs' doc-remarks that were moved to the comments for the functions/methods, HTML tags were added, so the same formatting would appear in javadocs. However, these tags would also appear in the wrappers for all other languages and may seem inappropriate. need to have GenDllWrappers remove the following tags from the comment blocks for all languages except Java: <table></table>; <tr></tr>; <td></td>; ; ; ; ;
; ; .

51) *ASTROSTDS-492*. *Create performance tests between JNA and JNI*. Create performance tests between JNA and JNI. ASPerformance.java class checked into issa_test repository.

52) *ASTROSTDS-493*. *Consolidate SAAL Utilities libraries into SGP4Prop*. Modify TeamCity build scripts to add all the SAAL Utilities libraries to the SGP4Prop and SGP4Prop Small packages. Remove the SAAL Utilities package. Need to edit ps1 and sh scripts in fortrandlls repository. Just add all the packages from SAAL Utilities to SGP4Prop and SGP4Prop Small.

53) *ASTROSTDS-496*. *Add driverexamples_data and driverexamples_julia to scripts*. Need to update createDevEnv.* and createNewVersion.sh scripts to include driverexamples_data and driverexamples_julia repositories.

54) *ASTROSTDS-499*. *Create unit test to check for epoch placement*. All of our engines are matching up neatly with the previous version EXCEPT BatchDC. With no changes to our code, the BatchDC test now returns "CheckCovergeence: Error! Cannot recover DC." for every single test. The test (from 3 years ago) is to propagate a VCM for a day, set those as obs and then run a BatchDC on the VCM. We should get back a VCM very close to the original. I dug WAY down in the history and found the attached sample code you had sent us. Our current BatchDC code is extremely similar to this, but I noted one change that testing suggests is the crux of the problem:

```
BatchDCWrapper.BatchDCLoadCard("     1          1111111          0099     1   P");
```

If we put that BatchDC card into our code, things work and a VCM is produced. It breaks when column 7 is changed to a '3' instead of a '1'. Upon reading the documentation correctly, '1' indicates to produce a VCM epoched to the last obs. A '3' indicates to produce a VCM epoched to the epoch of the original VCM. Of note, '5' and '6' also produce VCMs as expected. '2' and '4' run forever when I keep columns 9-23 blank – assuming because it's trying to propagate back to 1950 or whatever happens to be in memory. Using a '4' and setting column 9-23 to the VCM epoch does give the result I expect.

55) *ASTROSTDS-501*. *Remove Dinh's name from all driver examples*. Software Engineer's name is in the comment blocks. Totally remove Software Engineer's name from all driver examples code, make file, ...

56) *ASTROSTDS-502*. *Create TestInterface method in JNI*. See JnaDllMain class for an example of this method. This will be the template to start building JNI classes. In order for data to be passed between Java and JNI, there needs to be a way to translate char, int, long, double, and String data types. And 1 and 2 dimensional arrays of these data types. This Jira is to create examples on how to do this. Created jni-examples Git repo.

57) *ASTROSTDS-503*. *Add Sgp4GetCount() to SGP4*. Add this new function to better help with debugging/unit test.

58) *ASTROSTDS-504*. *Add SpGetCount() to SP*. Add this new function to better help with debugging/unit test.

59) *ASTROSTDS-512*. *Remove SGP4 dependency on GEO model setting*. Allow SGP4 to use its own WGS-72 earth constants regardless of input GEO model setting.

60) *ASTROSTDS-513*. *Remove SGP4-XP dependency on GEO model setting*. Allow SGP4-XP to use its own EGM-96 earth constants regardless of input GEO model setting.

61) *ASTROSTDS-517*. *Change calling convention in C#'s [DllImport]*. Change calling convention in C# wrappers [DllImport] to include: "CallingConvention=CallingConvention.Cdecl" to avoid 'pinvokestackimbalance' exception in C# driver in debug mode.

62) *ASTROSTDS-518*. *Use satellite elset type to determine what earth constants to be used in any conversions*. Use satellite elset type to determine what earth constants to be use in any conversions that need earth constants. For example, TLEs type 0, 2 will use WGS-72, TLEs type 4 will use EGM-96, and VCMs will use whatever set in "GEOPOTENTIAL:" field. In previous versions all conversions that need access to earth constants use whatever set in the global earth constants (EnvGeoIdx/WGS-72 by default).

63) *ASTROSTDS-519*. *Allow TleParseGP() to be used with a CSV tle*. In previous version, only two lines of a tle is allowed. This version allow to enter a CSV tle as well.

64) *ASTROSTDS-520*. *Allow TleLinesToArray() to be used with a CSV tle*.

65) *ASTROSTDS-521*. *Add new method TleGPFieldsToCsv() to Tle DLL/SO*.

66) *ASTROSTDS-522*. *Add new method TleGPArrayToCsv() to Tle DLL/SO*.

67) *ASTROSTDS-523*. *Add new method SetTleKeyMode() to Tle DLL/SO*. This new method allows duplicate Tles to be loaded in memory (but having different satKey). Add new method GetTleKeyMode() to Tle DLL/SO.

68) *ASTROSTDS-524*. *Add new method GetTleKeyMode() to Tle DLL/SO*. This new method allows duplicate Tles to be loaded in memory (but having different satKey). Add new method SetTleKeyMode() to Tle DLL/SO.

69) *ASTROSTDS-525*. *Add new method SatStateEphCom_OS() to SatState DLL/SO.* This new method intends to be used in one step (_OS) with direct input states of primary and secondary satellites. The method removes the steps involving satKeys (loading /retrieving/passing satKeys).

70) *ASTROSTDS-527*. *Add new method Sgp4PropDs50UtcPosVel() to Sgp4Prop DLL/SO.* This method only returns satellite position and velocity.

71) *ASTROSTDS-528*. *Add new method Sgp4ReepochCsv() to Sgp4Prop DLL/SO*. Similar to Sgp4ReepochTle() but for CSV tle.

72) *ASTROSTDS-529*. *Add new method Sgp4GenEphems() to Sgp4Prop DLL/SO*. Similar to Sgp4ReepochTle() but for CSV tle.

73) *ASTROSTDS-530*. *Add new method Sgp4GenEphems_OS() to Sgp4Prop DLL/SO*. Similar to Sgp4ReepochTle() but for CSV tle.

74) *ASTROSTDS-531*. *Add new method Sgp4PropAllSats() to Sgp4Prop DLL/SO*. Similar to Sgp4ReepochTle() but for CSV tle.

75) *ASTROSTDS-532*. *Add new method ExtEphemToEGP() to BatchDC DLL/SO*. Using SGP4/SGP4-XP to fit against an external ephemeris (treated ephemerides as P obs).

76) *ASTROSTDS-538*. *Update BatchDC document for new epoch placement flags.* Allow placement at beginning of day of last obs, beginning of day of epoch, and nodal crossing prior to epoch. Implement new epoch placement flags.

77) *ASTROSTDS-541*. *Add new method Iomod_OS() to ObsOps*. This new method Iomod_OS (one step) allows the user to compute initial orbit vector directly from the input sensor/observation data in one step. It removes the additional steps: loading sensor/observation data, retrieving sensor/observation keys, and then pass that data to a subsequent call. Another benefit of using this new method is that it's thread-safe and no data is left in memory when this method exits.

78) *ASTROSTDS-542*. *Sgp4 propagator now returns specific error code (errCode) for each case*. The old version only returns a generic code (errCode = 1) for all error cases. The new version will return different values for different error cases. This helps getting specific error code in multi-thread application in which GetLastErrMsg() doesn't work correctly.

79) *ASTROSTDS-545*. *Refactor LAMOD's C driver code*. Replace multiple calls to LAMOD's APIs with a call to LamodComputeLookView(). This improves LAMOD's performance and make code easier to understand.