HQ Space Operations Command (SpOC)
DCG-T/S9I
Astrodynamics Standards Engineering Group

Astrodynamics Standards

Known Issues

Version 9.4

May 2024

1. General

This document is to provide information on *Known Issues*. *Known Issues* is defined by Astro Standards as problems that may occur that we are aware of but have no control to resolve at the time of this release.

2. Known Issues

Issue	Comment
There is an issue when running Astro Standards **SpProp** library on **Windows Server 2019**. Any code generated using the SpProp package (e.g… SpProp, VCM, or SpVec) may incur precision issues. For example, if you propagate for more than 3 days, the location values will start to deviate by millimeters, and velocity values will deviate by micrometers/second.	This issue was discovered in version 8.2 and is known to exist in version 8.1. To our understanding, this issue is due to the native library dependencies on Windows Server 2019. This problem does not exist on Windows 10 or Linux.
Java Native Interface (JNI) and **Java Native Access (JNA)** are ways to call Astro Standards function calls. JNI has been known to be faster than JNA, but for Astro Standards, there is one case which it has been tested to be slower. JNI will be slower than JNA for any function call that uses 2 dimensional arrays.	It may not be noticeable for small data sets and/or small number of iterations. Astro Standards provides methods/functions ending in “2Das1D”, which takes a 1 dimensional array instead, which results in faster performance.
Rotas Chi Squared algorithm does not work when light time correction flag is set to “1”. This applies to angles only Observations.	Work around is to use a light time correction flag value of “0” or “2”.
Python 3.9 and 3.11 not working with Astro Standards on Windows	CDLL() function does not use the PATH environment variable to find the Astro Standards libraries. As a workaround, you need to use os.add_dll_directory() function call at the start of the LoadDllMainDll() function in DllMainWrapper.py file.