STANDARDIZED ASTRODYNAMIC ALGORITHMS (SAA)
SGP4 RELEASE PACKAGE README FILE
As of 31 October 2015

OVERVIEW:

The Standardized Astrodynamic Algorithms (SAA) SGP4 release package contains the following items:

- Executable code for the SAA in DLL (Dynamic Link Library) format.
- Example driver programs that access the DLL files and provides a means to run the SAA DLL.
- Source code for the driver programs in several programming languages (if available).
- Documentation for the SAAs in the format of Compiled HTML Markup (CHM) files for each of the DLLs
- Programmers Guide that provides example code to implement the drivers in several programming languages
- A test environment to verify the SAA run correctly and produces valid results.
- A VERDICT comparison program is included in this deliver that compares expected results with results of running the algorithm

FILE ORGANIZATION

The SAA SGP4 package is organized as follows:

```
Software_Version_xxxxx (where xxxxx can be WIN32, WIN64, Linux32 or Linux64)
.  Documents        (Program Documentation)
.  DriverExamples   (Driver Program examples using various programming languages)
.  .  <subdirs>     (Specifics for each programming language)
.  Lib              (SAA DLLs and required runtime libraries)
.  Verify           (Standalone Test Environment)
.  .  Baseline      (AFSPC certified results)
.  .  Diff          (Delivered empty – differences are recorded here after running script)
.  .  Execution     (Test cases, program environment, batch file)
.  .  TestResults   (Delivered empty - Results are recorded here after running script)
.  .  VERDICT       (Comparison program & example run script)
.  .  .  Reports    (Delivered empty - results are recorded here after running the VERDICT script)
```

After installing the files on your local machines' drive, a few steps are needed to verify and validate that the installation is correct.

TEST VERIFICATION INFORMATION

Step 1: A fully functioning test environment is provided with this release package. A single script can be run that will execute a series of test cases, compare the output produced against the set of certified SAA baseline results, and write the differences to a separate directory for easier comparisons.

To run the test script, perform the following:

If the test script was run previously and you wish to save any files in the Verify\Diff directory, rename the files using an extension other than .txt - the script deletes any existing .txt files when run.

Run the Test Cases script located in the Verify\Execution directory.

```
Run_Test_Cases.bat
```

Note: On some networks, running the executables or batch files is not possible when the files are loaded on a network drive. Copying the files on to your local drive is recommended.

The script will run a series of test cases using three input files (satf.082, sgp4_val.inp, rel14.inp), write the results to the Verify\TestResults directory, compare the results to the base-lined output files located in the Verify\Baseline directory, and write any differences to the Verify\Diff directory. The Diff directory should be examined after the Test Case script finishes. There should NOT be any numerical differences found in the results.

Step 2: After running the above Run_Test_Cases.bat script, the VERDICT program located in the Verify\VERDICT folder can be run.

This program examines two files and generates a report of differences found in numerical data. Unlike most other comparison programs, the comparison data can be in different locations in each file and be written in different numerical notation. An initialization file, written by the user, tells the program what values to read and how to interpret the data. If differences in the target data are found, the program will provide information on the magnitude of the difference found. Thus, the user can easily distinguish between a round-off value and a more serious difference in a file with a large amount of data.

VERDICT will use the data produced by the test cases. An example batch file, Run_VERDICT.bat, can be run to demonstrate a typical use of the VERDICT program.

If the VERDICT script was run previously and you wish to save any files in the Verify\VERDICT\Reports directory, rename the files using an extension other than .txt - the script deletes any existing .txt files when run.

Run the VERDICT script located in the Verify\VERDICT directory.

```
Run_VERDICT.bat
```

A file by file comparison of output data contained in the TestResults directory with those found in the Baseline directory is performed. Based on instructions contained in a user-defined .ini file, the program will search two files for differences in the target data and will generate a report on the changes found.

The VERDICT program output will specify whether any differences in the target data were found in the text files found in the Verify\VERDICT\Reports directory. If so, the program will display the values and provide information on the magnitude of the differences.

Suggestion #1: To see an example of the results if no differences are found, modify one of the files to be compared by changing a numeric value (see .ini file for target data).

Suggestion #2: If you modify the .ini file for your own use, test first with known differences in the targeted data, particular at both ends of the value. (Counting columns to specify the data to extract can be tedious). For example, if target field has a value of 0.999999, run against a copy of the file with the value modified to be 1.999991. Using this strategy, you can determine if the full field is being read for the comparison.

If there are no differences are found running VERDICT with no modifications as suggested above, then the text files will report “No Discrepancies Found” which indicates that you have successfully installed the package and you are now ready to build an application using the DLLs or integrate the DLLs into an existing application. Examples of using the DLLs in a number of environments can be found in the

Driver_Examples folder to include C, DotNet (C#), Fortran, Java, Matlab and Python. The Programmer's Guide (word document) and the SAA CHM files found in the Documents folder are excellent resources for assisting users in implementing the DLLs (See information about CHM files).

Caution: The name of the files in the Verify\TestResults, Verify\Baseline, Verify\Diff and Verify\VERDICT\Reports directories are the same for each test case. It is recommended that you don't modify the Run_Test_Cases.bat or Run_VERDICT.bat and change the names of the files as they have been configured specifically for expected results based AFSPC Verification and Validation (V&V) process. The name of the file indicates the data comparison that is being performed as described below.

- xxx_C_LatLonHeight.txt - This file contains time, latitude, longitude, height, and position: Latitude, Longitude, Height

- xxx_C_MeanElem.txt – This file contains time, and mean Keplerian elements (semi-major axis replaced with mean motion) : Eccentricity, Inclination, Node (right ascension of ascending node), Omega (Argument of perigee), MA (Mean Anomaly)

- xxx_C_NodalApPer.txt – This file contains time, nodal period, apogee, perigee: Nodal Period (Minutes or Revs/Day), Anomaly Period, Apogee and Perigee

- xxx_C_OscElem.txt – This file contains time, and osculating Keplerian elements (mean anomaly replaced with true anomaly): Altitude, Eccentricity, Inclination, Node (right ascension of ascending node), Omega (Argument of perigee), True Anomaly

- xxx_C_OscState.txt – This file contains time, position and velocity outputs: X, Y, Z, XDot, YDot, ZDot

V7 DRIVER NOTES (reference the examples in the C subdirectory)

MT suffix stands for Multi-Thread. These MT examples demonstrate that the DLLs can be used in multi-thread applications. The examples are not fully functional.

To access the mathematical algorithms encapsulated within the DLL/Share Object library structures, a separate driver program (executable) is required. This driver program can be written in several programming languages, thus allowing the end user the ability to tailor the products into their own unique environments.

USAGE NOTES:

The included C_driver program is called with the following required and optional parameters. The start, stop, and step used are passed are specificed in the input file.

Usage : C_Software inFile outFile [-Ilibpath] [-DlogFile]

inFile: File containing TLEs and 6P-Card (which controls start, stop times and step size)

outFile: Base name for five output files

-IlibPath: Optional, Specified path "libpath" to the SAA library.

-DlogFile: Optional logFile to enable writing debug data to the specified file

The -I parameter would be used to tell the program to search for the SAA DLLs in a location other than the current directory, Lib. If the -I parameter is used, do not place a space between the -I and the path name. The pathname must end with a \ character. Example: -I..\..\Lib\ The -I parameter will only find the SAA DLL files, not other required runtime libraries, so it is recommended to add the pathname to the PATH logical instead.

EXTERNAL RUNTIME LIBRARIES

Certain non-AFSPC-developed runtime libraries may need to be copied to the location where the driver program is run. Alternately, the (Windows) PATH logical can be set to search a directory location for the runtime libraries (as is done in the Run_Test_Cases batch file).

```
e.g. SET PATH=../../Lib;%PATH%
```

OTHER NOTES:

The Lib directory may contain 32 or 64-bit DLL files depending on the delivery package. A file list is included in the Lib directory to help identify the files (by date/size) in case of mix up. The C language driver writes a log file to the execution directory. This file needs to be writable.

COMPILED HTLM MARKUP FILES (CHM)

Since CHM files are considered active content, opening the files over the network will not have the active links. These files should be saved on a local drive/desktop.

CHM files are Microsoft™ Compressed HTML help files in a proprietary format. Under Linux/BSD or UNIX, you can open .chm files by using xchm

```
$ xchm file.chm
```

There are other free programs that can be used on Linux as well. See https://www.linux.com/news/software/applications/8209-chm-viewers-for-linux

ADDITIONAL RESOURCES:

Additional assistance on the use of the algorithms can be requested by emailing AFSPC at afspc.astrostandards@us.af.mil

A list of Frequently Asked Questions can be found at www.astrodynamicstandards.org

Documentation can be found in the Documents folder.