

Ephemeris Propagator Spec and Test Procedures

[Explore](#)

[Known Bugs](#)

[Failing Tests](#)

[Other Findings](#)

[Requirements](#)

[Interface/Functional Spec](#)

[Overview](#)

[Description](#)

[Fields](#)

[GUI](#)

[Remarks](#)

[Interaction with the Propagate Command](#)

[Examples](#)

[Test Procedures](#)

[Assumptions](#)

[System Tests](#)

[Edge/Corner/Stress](#)

[Unique Validation](#)

[Unique Mode Tests](#)

[Unique GUI Tests](#)

Explore

Known Bugs

Bugs known at start of final spec and testing activities

- GMT-374: Change to SPK propagator doesn't "take"
- GMT-2395: SPK Propagator has rounding error
- GMT-2142: Meter Level Error Growth in SPK Propagation
- GMT-2141: SPK Propagation Results in Extra Line in Report Files
- Resolved Invalid: GMT-390: Backwards propagation with SPK looks for wrong epoch
- GMT-2684: Propagator Panel accepts values even when there is an error in the panel
- Closed This GMT-2312: Ephemeris propagator has large numeric error

- GMT-391: Crash when SPK propagator generates an exception

Failing Tests

I have not dug into these to make sure they are all true failures - SPH

----- Validation Bugs

These were all tc file errors. (lots of non-standard error messages though) : -SPH

EphemPropagator_Validation_MismatchedEpochFormat (script) [no warn or err]
 EphemPropagator_Validation_NoFiles1 (script) [no warn or err]
 EphemPropagator_Validation_NoFiles2 (script) [no warn or err]
 EphemPropagator_Validation_Type (script) [ptrn 1 failed]
 EphemPropagator_Validation_ForwardPropOffFile (script) [no warn or err]
 EphemPropagator_Validation_WrongId (script) [no warn or err]
 EphemPropagator_Validation_EpochBeforeStartOfFile (script) [no warn or err]
 EphemPropagator_Validation_EpochAfterEndOfFile (script) [no warn or err]
 EphemPropagator_Validation_CentralBody (script) [ptrn 1 failed]
 EphemPropagator_Validation_MismatchedEpochFormat2 (script) [no warn or err]
 EphemPropagator_Validation_BackwardPropOffFile (script) [no warn or err]
 EphemPropagator_Validation_WrongFiles (script) [no warn or err]

----- Numeric failures

EphemPropagator_EpochFormat_ChangeInMission (script) [pos err 55.437836,tol 0.001000]
 EphemPropagator_Origin_Multiple_Cassini[GMAT: 671 lines, Truth:673 lines]

Other Findings

- **Reported GMAT-3275:** Crash changing propagator type.
- **Reported: GMT-3292:** Changing origin of SPK propagator has unclear affect
- **Reported: GMT-3289** First step algorithm fails for backwards propagation using SPK propagator
- **Reported: GMT-3299:** Non-standard error message for invalid gregorian date. See spacecraft dialog, for valid message like this: The value of "asdf " for field "Epoch" is not an allowed value. The allowed values are: [DD Mon YYYY HH:MM:SS.sss]. Recommendation: Check in P2 Bug
- **Reported: GMT-3300:** Non-standard error message thrown setting step size to zero on SPK propagator. Recommendation: Check in P2 bug.
- **Reported: GMT-3301:** There is an option to set the epoch to "FromSpacecraft". This cannot be done through the GUI. If you load a script that has this setting, it is possible to

get an error message from the GUI. Recommendation. I lean towards fixing this because it is possible to get an error message in the GUI even though the configuration is valid.

- **Reported Enhancement GMT-3302:** Can't stop near epoch based stopping conditions in some cases when the epoch is less than one time step from the end of the file. I see why root location based stopping conditions may never be able to support this, but it seems like we could propagate to precisely the end of the file if the stopping condition is epoch based. Am I missing something that precludes this? For now, I documented this because the ephemeris interpolation is likely poor near the end of the file and the best solution is to provide more data.

To Do

- Done: Fix tc file issues for failing validation tests

Requirements

These are working requirements. They are included here for review and convenience purposes. After review, requirements are maintained in the formal SRS located at SourceForge in /trunk/doc/SystemDocs/Requirements. Send final requirements to CCB lead (SPH).

FRR-27.1.0	The system shall contain an ephemeris propagator that supports the following ephemeris types:
FRR-27.1.1	1) SPK
FRR-27.1.2	2) Code 500
FRR-27.1.3	3) STK .e
FRR-27.2	The ephemeris propagator shall interpolate the ephemeris file that is specified on the spacecraft object.
FRR-27.3.0	The system shall allow the user to specify the following settings for an ephemeris propagator:
FRR-27.3.1	1) Step size
FRR-27.3.2	2) Origin

FRR-27.3.3.0	3) Ephemeris start epoch with the following options:
FRR-27.3.3.1	1. Epoch in any format and system defined in FRR-2
FRR-27.3.3.2	2. Inherit the epoch from the spacecraft

Interface/Functional Spec

Overview

A propagator that uses SPK, Code 500, or STK .e ephemeris files.

Description

An SPK propagator propagates a spacecraft by interpolating user-provided SPK SPICE kernels. You configure a Propagator to use an SPK kernel by setting the Type field to SPK. SPK kernels and the NAIFId are defined on the Spacecraft Resource. You control propagation, including stopping conditions, using the Propagate command.

A Code 500 propagator propagates a spacecraft by interpolating user-provided Code-500 ephemerides. You configure a Propagator to use an code 500 ephemeris by setting the Type field to Code500. The ephemeris file is defined on the Spacecraft Resource. You control propagation, including stopping conditions, using the Propagate command.

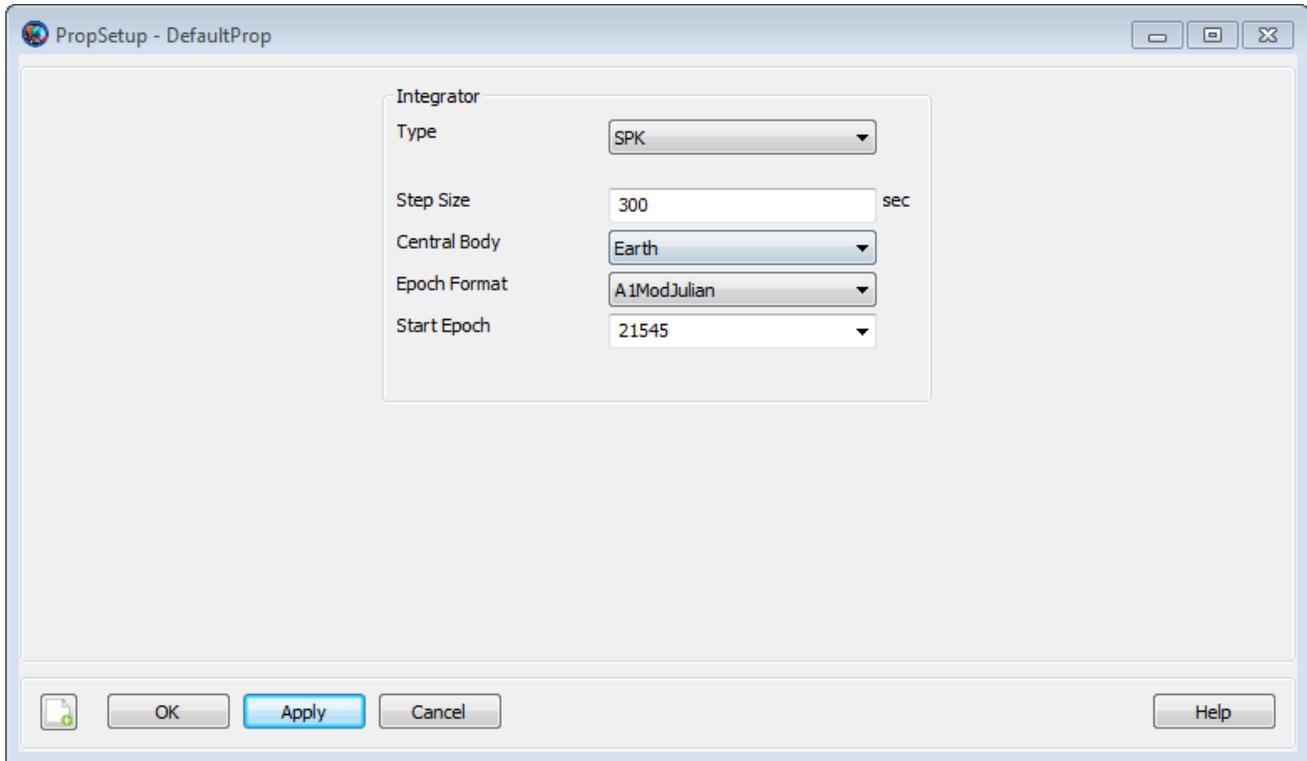
An STK .e propagator propagates a spacecraft by interpolating user-provided STK (.e) ephemerides. You configure a Propagator to use an STK ephemeris by setting the Type field to STK. The ephemeris file is defined on the Spacecraft Resource. You control propagation, including stopping conditions, using the Propagate command.

See Also: Spacecraft, Propagate.

Fields

See the [User Interface Spec](#) spreadsheet for reference information for fields. This section is usually empty other than the hyperlink.

GUI



To use an ephemeris propagator, on the Propagator dialog box, select SPK, Code 500, or STK in the Type drop-down menu. There are four fields you can configure for an ephemeris propagator: StepSize, CentralBody, EpochFormat, and StartEpoch. Note that changing the EpochFormat setting converts the input epoch to the selected Format. You can also type “FromSpacecraft” into the StartEpoch field and the propagator will use the epoch of the Spacecraft as the initial propagation epoch.

Open Issue: Behavior when changing back and forth between ephemeris and numeric type propagators is not clearly defined. It appears to me, that while the dialog box is up, Force Model settings are persistent when changing back and forth to and from SPK to integrators. Integrator settings and SPK Propagator settings are not persistent.. (e.g. can change to SPK, to Numeric) This behavior is probably acceptable and just need to make sure that it doesn't cause any unforeseen problems.

Remarks

To use an SPK propagator, you must specify the SPK kernels and NAIFId on the spacecraft Resource, configure a propagator to use SPK files as opposed to numerical methods, and configure the Propagate command to use the configured SPK propagator. The subsections and examples below discuss each of these items in detail.

To use a Code 500 or STK propagator, you must specify the ephemeris file on the spacecraft Resource, configure a propagator to use the corresponding propagation method, and configure the Propagate command to use the configured ephemeris propagator. The subsections and examples below discuss each of these items in detail.

Configuring Spacecraft SPK Kernels

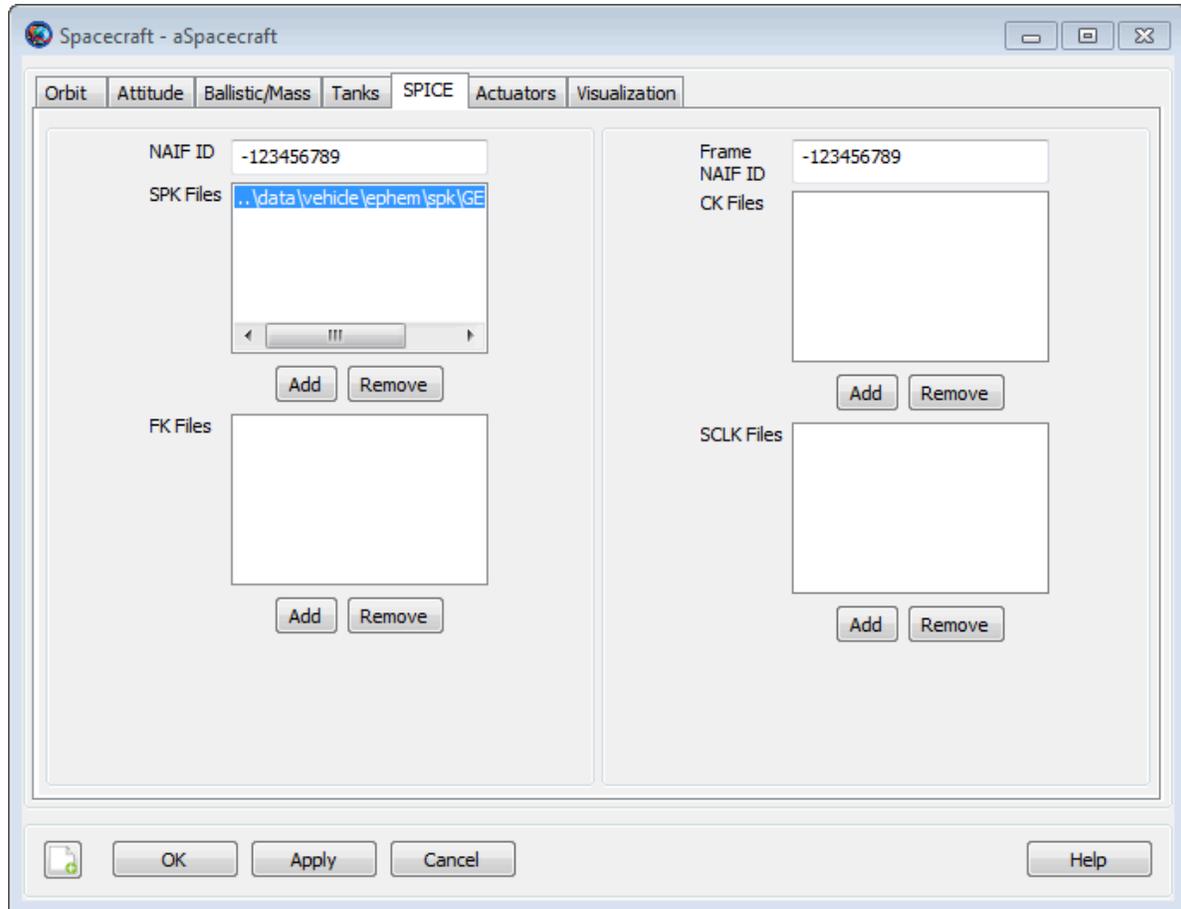
To use an SPK propagator, you must add the SPK kernels to the spacecraft and define the spacecraft NAIFId. SPK Kernels for selected spacecraft are [available here](#). Two sample vehicle spk kernels, (GEOSat.bsp and MoonTransfer.bsp) are distributed with GMAT for example purposes. An example of how to add spacecraft kernels via the script interface is shown below.

```
Create Spacecraft aSpacecraft;  
GMAT aSpacecraft.NAIFId = -123456789;  
GMAT aSpacecraft.OrbitSpiceKernelName =  
{ '..\data\vehicle\ephem\spk\GEOSat.bsp' };
```

To add spacecraft SPK kernels via the GUI:

1. On the spacecraft dialog box, click the SPICE tab.
2. Under the SPK Files list, click Add.
3. Browse to locate and select the desired SPK file
4. Repeat to add all necessary SPK kernels
5. In the NAIF ID field, enter the spacecraft integer NAIF id number.

Note: For a given mission, each spacecraft in the script should have a unique NAIF ID if the spacecraft are propagated with an SPK propagator.



You can add more than one kernel to a spacecraft as shown via scripting below, where the files GEOSat1.bsp and GEOSat2.bsp are dummy file names used for example purposes only, and are not distributed with GMAT. In the script, you can use relative path or absolute path to define the location of an SKP file. Relative paths are defined with respect to the GMAT bin directory of your local installation.

```
Create Spacecraft aSpacecraft;
GMAT aSpacecraft.OrbitSpiceKernelName =
{'..\data\vehicle\ephem\spk\GEOSat1.bsp',
'C:\MyDataFiles\GEOSat2.bsp'};
```

Configuring Other Ephemeris Sources on a Spacecraft

The GMAT GUI does not currently have an entry for the non-SPICE based ephemeris sources. Scripting for the ephemeris file is performed using the EphemerisName field:

```
Create Spacecraft STKSat;
GMAT STKSat.EphemerisName = 'soho.e';
```

Configuring an Ephemeris Propagator

You can define the StartEpoch of propagation of ephemeris propagators on either the Propagator Resource or inherit the StartEpoch from the spacecraft. Below is a script snippet that shows how to inherit the StartEpoch from the spacecraft.

To inherit the StartEpoch from the spacecraft using the GUI

1. Open the SPK propagator dialog box,
2. In the Start Epoch field., type “FromSpacecraft” or select “FromSpacecraft” from the drop-down menu.

To explicitly define the start epoch on the propagator Resource use the following syntax.

```
Create Propagator spkProp;  
spkProp.EpochFormat = 'UTCGregorian';  
spkProp.StartEpoch = '22 Jul 2014 11:29:10.811';  
  
Create Propagator spkProp2;  
spkProp2.EpochFormat = 'TAIModJulian';  
spkProp2.StartEpoch = '23466.5';
```

To configure the step size and central body, use the StepSize and Central Body fields, shown here for an SPK propagator.

```
Create Propagator spkProp;  
spkProp.Type = SPK;  
spkProp.StepSize = 300;  
spkProp.CentralBody = Earth;
```

For DJC: What does Central Body do? It appears to have no affect on graphics, propagation, or parameter output. Run the examples below if needed. The SPK files were committed to the application folder on SourceForge 14 Nov. 2012. -SPH

Interaction with the Propagate Command

Ephemeris propagators work with the Propagate command in the same way numerical propagators work with the Propagate command with the following exceptions:

- If a Propagate command uses an SPK propagator, then you can only propagate one spacecraft using that propagator. You can however, mix SPK propagators and numeric propagators in a single propagate command.
- The SPK Propagator will not propagate the STM or compute the orbit Jacobian (A matrix).

In the example below, we assume a Spacecraft named aSpacecraft and a Propagator named spkProp have been configured a-priori. An example command to propagate aSpacecraft to Earth Periapsis using spkProp is shown below.

```
Propagate spkProp(aSpacecraft) {aSpacecraft.Earth.Periapsis};
```

Below is a script snippet that demonstrates how to propagate backwards using an SPK propagator.

```
Propagate BackProp spkProp(aSpacecraft) {aSpacecraft.ElapsedDays =  
-1.5};
```

Behavior Near Ephemeris Boundaries

In general, ephemeris interpolation is less accurate near the boundaries of ephemeris files and we recommend providing ephemeris for significant periods beyond the initial and final epochs of your application for this and other reasons. When propagating near the beginning or end of ephemeris files, the use of the double precision arithmetic may affect results. For example, if an ephemeris file has an initial epoch TDBModJulian = 21545.00037249916, and you specify the StartEpoch in UTC Gregorian, round off error in time conversions and/or truncation of time using the Gregorian format (only accurate to millisecond) may cause the requested epoch to fall slightly outside of the range provided on the ephemeris file. The best solution is to provide extra ephemeris data to avoid time issues at the boundaries and the more subtle issue of poor interpolation.

Caution: To locate requested stopping conditions, GMAT needs to bracket the root of the stopping condition function. Then, GMAT uses standard root finding techniques to locate the stopping condition to the requested accuracy. If the requested stopping condition lies at or near the beginning or end of the ephemeris data, then bracketing the stopping condition may not be possible without stepping off the ephemeris file which will throw an error and stop execution. In this case, you must provide more ephemeris data to locate the desired stopping condition.

Examples

Propagate a GEO spacecraft using an SPK Propagator. Define the start epoch from the spacecraft. Note: the SPK kernel GEOSat.bsp is distributed with GMAT.

```
Create Spacecraft aSpacecraft;
aSpacecraft.Epoch.UTCGregorian = '02 Jun 2004 12:00:00.000'
aSpacecraft.NAIFId = -123456789;
aSpacecraft.OrbitSpiceKernelName =
{'..\data\vehicle\ephem\spk\GEOSat.bsp'};

Create Propagator spkProp;
spkProp.Type = SPK;
spkProp.StepSize = 300;
spkProp.CentralBody = Earth;
spkProp.StartEpoch = FromSpacecraft;

Create OrbitView EarthView;
EarthView.Add = {aSpacecraft, Earth, Luna};
EarthView.ViewPointVector = [ 30000 -20000 10000 ];
EarthView.ViewScaleFactor = 2.5;

BeginMissionSequence;
Propagate spkProp(aSpacecraft) {aSpacecraft.TA = 90};
Propagate spkProp(aSpacecraft) {aSpacecraft.ElapsedDays = 2.4};
```

Simulate lunar transfer using an SPK Propagator. Define start epoch on the propagator. Note: the SPK kernel MoonTransfer.bsp is distributed with GMAT.

```
Create Spacecraft aSpacecraft;
aSpacecraft.NAIFId = -123456789;
aSpacecraft.OrbitSpiceKernelName =
{'..\data\vehicle\ephem\spk\MoonTransfer.bsp'};

Create Propagator spkProp;
spkProp.Type = SPK;
spkProp.StepSize = 300;
spkProp.CentralBody = Earth;
spkProp.EpochFormat = 'UTCGregorian';
spkProp.StartEpoch = '22 Jul 2014 11:29:10.811';

Create OrbitView EarthView;
EarthView.Add = {aSpacecraft, Earth, Luna};
```

```
EarthView.ViewPointVector = [ 30000 -20000 10000 ];
EarthView.ViewScaleFactor = 30;

BeginMissionSequence;
Propagate spkProp(aSpacecraft) {aSpacecraft.ElapsedDays = 12};
```

Test Procedures

Assumptions

Nominal System Testing of SPK propagator is broken down into several groups. The basic tests ensure correct functioning of the propagator, and other GMAT components required to propagate using an SPK propagator.

- Spacecraft SKP kernel
- Spacecraft NAIFId
- Step size
- Start Epoch
- Central body
- Propagate command interaction (Stopping condition, state data, etc.)

Additional tests include the interaction of SPK propagator with other system components including the following:

- Parameter computations
- Command Summary
- Graphical output (OrbitView, GroundTrackPlot, XYPlot): **Note: These will be tested under a separate test procedure and are not included here.**
- Textual output (EphemerisFile, ReportFile): **Note testing of EphemerisFile output will be tested under EphemerisFile and are not included here.**

Note: Extensive testing of Propagate command and SPK propagator testing are performed under the Propagate command feature and we only test basics of the interaction between SPK Propagator and Propagate command in these procedures.

System Tests

Basic SPK Propagator tests ensure the SPK propagator and components required to perform propagation using an SPK file function correctly. This includes configuring Spacecraft, SPKPropagator, and Propagate command.

Priority	Status	Summary
P1	Done:SPH	<p>Test setting StartEpoch on the Propagator in all supported formats.</p> <p>See Files:</p> <p>EphemPropagator_StartEpoch_A1Gregorian EphemPropagator_StartEpoch_A1ModJulian EphemPropagator_StartEpoch_TAIGregorian EphemPropagator_StartEpoch_TAIModJulian EphemPropagator_StartEpoch_TTGregorian EphemPropagator_StartEpoch_TTModJulian EphemPropagator_StartEpoch_UTCGregorian EphemPropagator_StartEpoch_UTCModJulian EphemPropagator_StartEpoch_TDBGregorian EphemPropagator_StartEpoch_TDBModJulian</p>
P1	Done:SPH	<p>Test setting Start epoch to inherit from Spacecraft using a few different epoch formats.</p> <p>See File: EphemPropagator_StartEpoch_FromSpacecraft</p>
P1	Done:SPH	<p>Test various step sizes and verify propagation occurs at that requested step size.</p> <p>See Files:</p> <p>EphemPropagator_Origin_Earth_HST_5MinStepSize EphemPropagator_Origin_Earth_HST_TestExtraStep EphemPropagator_StepSize_1day EphemPropagator_StepSize_1hour EphemPropagator_StepSize_2minutes</p>
P1	Done: SPH	<p>Test various central bodies and various JPL spacecraft kernels.</p> <p>See Files:</p> <p>EphemPropagator_Origin_Earth_Rosetta EphemPropagator_Origin_Luna_Clementine EphemPropagator_Origin_Mars_MarsExpress EphemPropagator_Origin_Multiple_Cassini EphemPropagator_Origin_Sun_Dawn</p>

		EphemPropagator_Origin_Sun_MRO EphemPropagator_Origin_Sun_StarDust
P1	Done:SPH	Test forward and back slashes in path for SPK kernel See: EphemPropagator_StartEpoch_TDBModJulian
P1	Done:SPH	Test back propagation using an SPK kernel. See files: (first case uses for loop) EphemPropagator_BackPropagation EphemPropagator_BackPropagation_NoForLoop
P1	Done:SPH	Test propagation to selected stopping conditions using SPK propagator. (NOTE: Heavy duty tests for stopping condition are located in Propagate command test procedures. See files in FRC-10_Propagate named 'StopCond_GMAT_EarthSPK_ParamBattery' 'StopCond_GMAT_EarthSPK_StopGroupings' 'StopCond_GMAT_EarthSPK_BackProp'
P1	Done:SPH	Test back-to-back SPK propagation using same SPK propagator in multiple Propagate commands. Test in forward propagation. (tests hand off of epoch between propagate commands) See files: EphemPropagator_BackToBack_ForwardProp_EpochFromProp EphemPropagator_BackToBack_ForwardProp_EpochFromSat
P1	Done:SPH	Test back-to-back SPK propagation using same SPK propagator in multiple Propagate commands. Test in backward propagation. (tests hand off of epoch between propagate commands) See files: EphemPropagator_BackToBack_BackwardProp_EpochFromProp EphemPropagator_BackToBack_BackwardProp_EpochFromSat

P1	Done:SPH	<p>Test back-to-back SPK propagation using different SPK propagator in multiple Propagate commands. SPK Propagators have the same origin. (tests hand off of epoch and origin between propagate commands)</p> <p>See: EphemPropagator_PropStressTest_DifferentPropsSameOrigin</p>
P1	Done:SPH	<p>Test back-to-back SPK propagation using different SPK propagator in multiple Propagate commands. SPK Propagators do not have the same origin. (tests hand off of epoch and origin between propagate commands)</p> <p>See: EphemPropagator_PropStressTest_DifferentPropsSameOrigin</p>
P1	Done:SPH	<p>Test epoch passing of same SPK prop to two different spacecraft propagated back to back.</p> <p>See: EphemPropagator_SamePropDifferentSats</p>
P1	Done:SPH	<p>Test putting SPK Propagator in a for loop in step mode</p> <p>See Files: EphemPropagator_StepSize_1hour_ForLoop EphemPropagator_StepSize_2minutes_ForLoop EphemPropagator_StepSize_1day_ForLoop</p>
P1	Done:SPH	<p>Test putting SPK Propagator in for loop in stopping condition.</p> <p>See Files: EphemPropagator_PropStressTest_EpochFromProp EphemPropagator_PropStressTest_EpochFromSat</p>
P1	Done:SPH	<p>Test setting NAIFId on spacecraft</p> <p>See Files: EphemPropagator_Origin_Luna_Clementine EphemPropagator_Origin_Sun_Dawn EphemPropagator_Origin_Sun_MRO EphemPropagator_Origin_Ceres_Cassini for multi EphemPropagator_Origin_Earth_Rosetta</p>

P1	Done:SPH	<p>Test setting SPK Kernels on spacecraft. Test at least one case that requires more than one SPK kernel.</p> <p>For single kernels:</p> <p>EphemPropagator_Origin_Luna_Clementine EphemPropagator_Origin_Sun_Dawn EphemPropagator_Origin_Sun_MRO</p> <p>For multiple Kernels:</p> <p>EphemPropagator_Origin_Ceres_Cassini for multi EphemPropagator_Origin_Earth_Rosetta</p>
P1	Done:SPH	<p>Test starting propagation precisely at beginning of file propagating forward</p> <p>See:</p> <p>EphemPropagator_ForwardProp_StartPreciselyAtFileStart</p>
P1	Done:SPH	<p>Test starting propagation precisely at end of file propagating backwards</p> <p>See:</p> <p>EphemPropagator_BackwardProp_StartPreciselyAtFileEnd</p>

Tests for interactions with GMAT parameter subsystem.

Priority	Status	Summary
P1	Done:SPH	<p>Test CentralBody dependent parameter computations when propagator is SPK. (compute parameters w/r/t to prop central body)</p> <p>See File: EphemPropagator_CBParams_EarthOrigin_EarthMoonSun MarsParams</p>
P1	Done:SPH	<p>Test CoordinateSystem dependent parameter computations when propagator is SPK. (compute parameters w/r/t to prop</p>

		central body) See File: EphemPropagator_CSPParams_EarthOrigin_EarthMoonSun MarsParams
P1	Done:SPH	Test CentralBody dependent parameter computations when propagator is SPK. (compute parameters with respect to a body that is NOT the prop central body) See File: EphemPropagator_CBParams_EarthOrigin_EarthMoonSun MarsParams
P1	Done:SPH	Test CoordinateSystem dependent parameter computations when propagator is SPK. (compute parameters with respect to a body that is NOT the prop central body) See File: EphemPropagator_CSPParams_EarthOrigin_EarthMoonSun MarsParams

Edge/Corner/Stress

Priority	Status	Summary
P1	Done:SPH	Test propagating off spk file forward See File: EphemPropagator_Validation_ForwardPropOffFile
P1	Done:SPH	Test propagating off spk file backward: See File: EphemPropagator_Validation_BackwardPropOffFile
P1	Done:SPH	Test initial epoch is before start of file See File: EphemPropagator_Validation_EpochAfterEndOfFile
P1	Done:SPH	Test initial epoch is after start of file.

		See File: EphemPropagator_Validation_EpochBeforeStartOfFile
P1	Done:SPH	Test propagation using step mode in for loop, step mode in While loop, to cyclic, non-cyclic, and elapsed time stopping condition, forwards and backwards propagation: See Files: EphemPropagator_PropStressTest_EpochFromProp EphemPropagator_PropStressTest_EpochFromSat

Unique Validation

Priority	Status	Summary
P2	Deferred	What happens when setting SPK type fields on numeric type propagator: Most are disallowed, but you can set the following and they are silently accepted: DefaultProp.InitialStepSize = 300; DefaultProp.FM = InternalODEModel;
P2	Deferred	What happens when setting numeric type fields on an SPK type propagator. It appears they are not allowed.
P1	Done:SPH	Test propagating multiple spacecraft using the same SPK propagator is not allowed. See: EphemPropagator_Validation_MultipleSatsinSPKPropagator
P1	Done:SPH	Test non-existent SPK kernel See: EphemPropagator_Validation_NoFiles1 EphemPropagator_Validation_NoFiles2
P1	Done:SPH	Test invalid SPK kernel: See: EphemPropagator_Validation_InvalidKernelFile

P1	Done:SPH	NAIF Id is not on file See: EphemPropagator_Validation_NoFiles1 EphemPropagator_Validation_NoFiles2 EphemPropagator_Validation_WrongFiles
P1	Done:SPH	Test mismatched epoch and epoch format See this file for ModJulianFormat with Gregorian Epoch EphemPropagator_Validation_MismatchedEpochFormat2 See this file for Gregorian Format with ModJulian epoch. EphemPropagator_Validation_MismatchedEpochFormat

Unique Mode Tests

Not required because configuration of Propagator is not supported in Command mode.

Priority	Status	Summary

Unique GUI Tests

These are tests that are unique to the GUI interface for this feature that are not covered by the standard GUI test template and procedures.

Priority	Status	Summary
		Test changing epoch format and ensure epoch conversion is correct.
		Test changing from spk to integrator and make sure nothing crazy happens.

STK Ephem Propagator Tests