FRR-38 Calculation Parameters

- Explore
    - Known Bugs
    - Failing Tests
    - Other Findings
- Requirements
- Interface/Functional Spec
    - Overview
    - Description
    - GUI
        - General Usage
        - Special Considerations
    - Parameters
    - User Interface Design Trades for New Parameter Types
    - Parameter Select Dialog Behavior for Different Contexts and Parameters
        - Summary of Supported Object Types in PSDB
        - Parameter Attribute Definitions
        - Parameter Select Dialog Behavior by Context
    - Parameter Specifications By Resource
        - Spacecraft
        - FuelTank
        - Thruster
        - ImpulsiveBurn
        - Array, String, Variable
    - Examples
- Math Spec
    - Calculation Objects
        - Simple Parameters
            - MHA
- Test Procedures
        - Edge/Corner/Stress
        - Unique Validation
        - Unique Mode Tests
        - Unique GUI Tests
    - Space Point Parameter Test Procedures.
    - Forcem Model Parameter Test Procedures.

Explore

Known Bugs

JIRA ID	Summary	Recommendation
GMT-585	Multiple redundant properties/fields for epoch	P3
GMT-1098	Tab key navigation fails	Appears fixed
GMT-1346	Add ability for user-defined parameters	Improvement
GMT-1439	Parameter design for multiple dependency	Improvement
GMT-1483	Ability to Calculate and Output Sun Vector to any point	Improvement
GMT-1899	Beta Angle is Off by about .003 degrees	P1
GMT-1933	Earth Fixed RA changed from 0-360 to -180-180 measurement	P1 (document)

GMT-2011	Larger angular momentum errors in 09/15 build	P1, assign to Joel
GMT-2318	STM Parameters are wrong when using Coordinate System other than EarthMJ2000Eq	P1, assign to Joel
GMT-2475	Provide a parameter to check convergence status of a VF13 optimizer	Improvement
GMT-2673	User Reported: Parameter Select Dialog Box Silently Changes User's Selected Object	Needs verification
GMT-2755	ParameterCreateDialog does not close properly after adding string parameter on Mac	P2

Failing Tests

Failing tests listed as of 10/15/2012.

JIRA ID	Test	Summary	Rec.
TBD	`CbParams_Hyperbolic_2Body`	Incorrect `MA`, `OrbitPeriod`	Investigate
TBD	`CSParams_Uranus1_2Body_UranusFixed`	Numeric issues	Investigate
TBD	`CSParams_Saturn1_2Body_SaturnFixed`	Numeric issues	Investigate
TBD	`CSParams_Pluto1_2Body_PlutoFixed`	Test issues, plus numerics	Investigate
TBD	`CSParams_Neptune1_2Body_NeptuneMJ2000Ec`	Test issues, plus many incorrect parameters	Investigate
TBD	`CSParams_Neptune1_2Body_NeptuneFixed`	Numeric issues	Investigate
TBD	`CSParams_Hyperbolic_2Body`	Numeric issues	Investigate
TBD	`CSParams_GEO_2Body`	Numeric issues	Investigate

Other Findings

JIRA ID	Summary	Rec.
GMT-3164	Inconsistent naming: RadApo/RadPer and VelApoapsis/RadPeriapsis	P3
GMT-3165	ParameterSelectDialog is badly named	P2
GMT-3166	ParameterSelectDialog string consistency issues	P3
GMT-3167	ParameterSelectDialog doesn't remove item from available list once chosen	P2
GMT-3173	Clarification needed on HA output range	P1
GMT-3205	ParameterSelectDialog shouldn't use list box for single-parameter selection	P3

Requirements

> To reviewers: I'm considering removing "or set" and the "(read only)/(read/write)" bits from the requirements. All of the "write" parameters should already be described on the requirement for the parent resource.

ID	Requirements

FRR-38.1.0	The system shall allow the user to request or set the following object properties with respect to any celestial body in FRR-16:
FRR-38.1.1.0	1) Spacecraft and orbit parameters
FRR-38.1.1.1	1. Altitude (read only)
FRR-38.1.1.2	2. Beta angle (read only)
FRR-38.1.1.3	3. C3 energy (read only)
FRR-38.1.1.4	4. Eccentric anomaly (read/write)
FRR-38.1.1.5	5. Eccentricity (read/write)
FRR-38.1.1.6	6. Orbit energy (read only)
FRR-38.1.1.7	7. Magnitude of angular momentum (read only)
FRR-38.1.1.8	8. Planetodetic latitude (read only)
FRR-38.1.1.8	9. Longitude (read only)
FRR-38.1.1.10	10. Local sidereal time (read only)
FRR-38.1.1.11	11. Mean anomaly (read/write)
FRR-38.1.1.12	12. Mean hour angle (read only)
FRR-38.1.1.13	13. Mean motion (read only)
FRR-38.1.1.14	14. Orbit period (read only)
FRR-38.1.1.15	15. Radius of apoapsis (read/write)
FRR-38.1.1.16	16. Radius of periapsis (read/write)
FRR-38.1.1.17	17. Magnitude of position vector (read/write)
FRR-38.1.1.18	18. Semilatus rectum (read only)
FRR-38.1.1.19	19. Semi-major axis (read/write)
FRR-38.1.1.20	20. True anomaly (read/write)
FRR-38.1.1.21	21. Velocity at apoapsis (read only)
FRR-38.1.1.22	22. Velocity at periapsis (read only)
FRR-38.1.1.23	23. Hyperbolic anomaly (read/write)
FRR-38.1.1.24	24. Apoapsis crossing indicator (read only)
FRR-38.1.1.25	25. Periapsis crossing indicator (read only)
FRR-38.2.0	The system shall allow the user to request or set the following object properties with respect to any coordinate system in FRR-35:
FRR-38.2.1.0	1) Spacecraft and orbit parameters
FRR-38.2.1.1	1. Argument of periapsis (read/write)
FRR-38.2.1.2	2. Azimuth (read/write)
FRR-38.2.1.3	3. B dot T (read only)
FRR-38.2.1.4	4. B dot R (read only)
FRR-38.2.1.5	5. B vector angle (read only)
FRR-38.2.1.6	6. B vector magnitude (read only)
FRR-38.2.1.7	7. Declination (read/write)
FRR-38.2.1.8	8. Declination of velocity (read/write)

FRR-38.2.1.9	9.	Flight path angle (read/write)
(delete)	~~10.~~	~~Hyperbolic anomaly (read/write)~~
FRR-38.2.1.11	11.	Components of angular momentum (read only)
FRR-38.2.1.12	12.	Inclination (read/write)
FRR-38.2.1.13	13.	Orbit state transition matrix (read only)
FRR-38.2.1.14	14.	3x3 partitions of Orbit STM (read only)
FRR-38.2.1.15	15.	Right ascension (read/write)
FRR-38.2.1.16	16.	Right ascension of the ascending node (read/write)
FRR-38.2.1.17	17.	Right ascension of velocity (read/write)
FRR-38.2.1.18	18.	Magnitude of velocity (read/write)
FRR-38.2.1.19	19.	Components of Cartesian velocity (read/write)
FRR-38.2.1.20	20.	Components of Cartesian position (read/write)
FRR-38.2.1.21	21.	Right ascension of outgoing hyperbolic asymptote (read only)
FRR-38.2.1.22	22.	Declination of outgoing hyperbolic asymptote (read only)
FRR-38.2.1.23	23.	Equinoctial elements (read/write)
FRR-38.2.1.24	24.	Mean longitude (read/write)
FRR-38.3.0	The system shall allow the user to request or set the following object properties:	
FRR-38.3.1.0	1) Spacecraft and orbit parameters	
FRR-38.3.1.3	3.	Components of angular velocity (read/write)
FRR-38.3.1.4	4.	Drag coefficient (read/write)
FRR-38.3.1.5	5.	Reflectivity coefficient (read/write)
FRR-38.3.1.6	6.	Components of attitude direction cosine matrix (read/write)
FRR-38.3.1.7	7.	Drag area (read/write)
FRR-38.3.1.8	8.	Dry mass (read/write)
FRR-38.3.1.9	9.	Elapsed days (read only)
FRR-38.3.1.10	10.	Elapsed seconds (read only)
(delete)	~~11.~~	~~Euler angles for any sequence (read/write)~~
(delete)	~~12.~~	~~Euler angle rates for any sequence (read/write)~~
FRR-38.3.1.13	13.	Components of attitude quaternion (read only)
FRR-38.3.1.14	14.	SRP area (read/write)
FRR-38.3.1.15	15.	Epoch in TAI modified Julian (read/write)
FRR-38.3.1.16	16.	Epoch in TAI Gregorian (read/write)
FRR-38.3.1.19	19.	Epoch in TDB modified Julian (read/write)
FRR-38.3.1.20	20.	Epoch in TDB Gregorian (read/write)
FRR-38.3.1.21	21.	Epoch in TT modified Julian (read/write)
FRR-38.3.1.22	22.	Epoch in TT Gregorian (read/write)
FRR-38.3.1.23	23.	Epoch in UTC modified Julian (read/write)

FRR-38.3.1.24	24.	Epoch in UTC Gregorian (read/write)
FRR-38.3.1.25	25.	Total mass (read only)
FRR-38.3.1.26	26.	Modified Rodrigues parameters (read/write)
FRR-38.3.1.27	27.	Euler angles (read/write)
FRR-38.3.1.28	28.	Euler angle rates (read/write)
FRR-38.3.1.29	29.	Epoch in A.1 modified Julian (read/write)
FRR-38.3.1.30	30.	Epoch in A.1 Gregorian (read/write)
FRR-38.3.1.31	31.	Attitude quaternion (read/write)
FRR-38.3.2.0	2)	Impulsive burn parameters
FRR-38.3.2.1	1.	Thrust vector element 1 (read/write)
FRR-38.3.2.2	2.	Thrust vector element 2 (read/write)
FRR-38.3.2.3	3.	Thrust vector element 3 (read/write)
(delete)	~~4.~~	~~Thrust vector velocity component (read/write)~~
(delete)	~~5.~~	~~Thrust vector normal component (read/write)~~
(delete)	~~6.~~	~~Thrust vector binormal component (read/write)~~
FRR-38.3.3.0	3)	Tank parameters
FRR-38.3.3.1	1.	Pressure (read/write)
FRR-38.3.3.2	2.	Volume (read/write)
FRR-38.3.3.3	3.	Fuel density (read/write)
FRR-38.3.3.4	4.	Fuel mass (read/write)
FRR-38.3.3.5	5.	Temperature (read/write)
FRR-38.3.3.6	6.	Reference temperature (read/write)
FRR-38.3.4.0	4)	Thruster parameters
FRR-38.3.4.1	1.	Duty cycle (read/write)
FRR-38.3.4.2	2.	Thrust scale factor (read/write)
FRR-38.3.4.3	3.	Gravitational acceleration for thrust equation (read/write)
FRR-38.3.4.4	4.	Thrust coefficients (read/write)
FRR-38.3.4.5	5.	Isp coefficients (read/write)
FRR-38.3.4.6	6.	Components of thrust direction (read/write)

Interface/Functional Spec

Overview

Data property available for use by commands

Description

Parameters are named resource properties that can be used to obtain data for use by Mission Sequence commands or by output resources. Som e parameters, such as the **Altitude** parameter of **Spacecraft**, are calculated values that can only be used to retrieve data. They cannot be set directly. Others, such as the **Element1** parameter of **ImpulsiveBurn**, share the same name as a resource field and can be used both to set data and retrieve it. Parameters are distinguished from resource fields by their extra functionality: fields are static resource properties that are usually set in initialization (or in the GUI Resources tree), while parameters can be calculated on the fly and used in plots, reports, and mathematical expressions.

Parameters are classified as one of four types: central-body-dependent parameters, coordinate-system-dependent parameters, attached-hardware parameters, and standalone parameters. Standalone parameters are the simplest type, in that they have no dependencies. The **ElapsedSecs** parameter of **Spacecraft** is an example of this; it is simply referenced as *Spacecraft*.ElapsedSecs.

Central-body-dependent parameters, as the name suggests, have a value that is dependent on the chosen celestial body. The **Altitude** paramete r of **Spacecraft** is an example of this. To reference this parameter, you must specify a central body, such as *Spacecraft*.Mars.Altitude. Any built-in central body or user-defined **Asteroid**, **Comet**, **Moon**, or **Planet** is valid as a dependency. If the dependency is omitted, **Earth** is assumed.

Likewise, coordinate-system-dependent parameters have a value that is dependent on the chosen coordinate system. The **DEC** parameter of **Spa cecraft** is an example of this. To reference this parameter, you must specify the name of a **CoordinateSystem** resource, such as *Spacecraft*. EarthFixed.DEC. Any default or user-defined **CoordinateSystem** resource is valid as a dependency. If the dependency is omitted, **EarthMJ20 00Eq** is assumed.

Attached-hardware parameters have no dependencies, but are themselves dependent on being attached to a **Spacecraft**. **FuelTank** and **Thrust er** parameters are examples of this. The **FuelMass** parameter of **FuelTank** cannot be referenced without first attaching the **FuelTank** to a **Space craft**. Then, the parameter can be referenced as: *Spacecraft.FuelTank*.FuelMass.

The individual parameters are resource-specific, and are documented along with the their parent resources. The GUI, however, has a parameter selection interface that is common to all parameters. This interface is documented in GUI, below.

See Also: Script Language, FuelTank, ImpulsiveBurn, Spacecraft, Thruster

GUI

Parameters can be used as input in several places throughout GMAT, such as the **ReportFile** and **XYPlot** resources and the **If/Else**, **Propagate**, and **Report** commands. In the GUI, all of these use a common interface called the **ParameterSelectDialog** that allows for interactive parameter selection. A basic **ParameterSelectDialog** window looks like the following:

The **ParameterSelectDialog** window is used to build a parameter, along with any dependencies, for use in a command or resource. Some resources and commands have different requirements for the types of parameters that can be used, so the **ParameterSelectDialog** can take slightly different forms, depending on where it's used. This section will describe the generic interface, then mention any resource- or command-specific exceptions.

General Usage

The first step in choosing a parameter is to select the object (or resource) type from the **Object Type** list in the upper left. Five types can appear in this list: **Spacecraft**, **ImpulsiveBurn**, **Variable**, **Array**, and **String**.

Once you've selected a type, The **Object List** box is populated with all existing resources of that type. Use this list to choose the specific resource you'd like to reference.

If the **Spacecraft** type is selected, the **Attached Hardware List** appears below the **Object List**. This list displays any hardware (such as **FuelTan k** resources) attached to the selected **Spacecraft**. If the **Array** type is selected, **Row** and **Col** boxes appear. Use these to specify a row and column to select an individual array element, or check **Select Entire Object** to choose the entire array.

Once a resource is selected, the **Object Properties** list is populated with all available parameters provided by that resource. Some resources, such as instances of **Variable** or **Array**, are themselves parameters, so this list remains empty.

Parameters with different dependency types are commingled in the **Object Properties** list. When you select one, the appropriate dependency (if any) appears below the list. For example, after selecting the **Spacecraft AOP** parameter, a **Coordinate System** list appears. After selecting the **S pacecraft Apoapsis** parameter, a **Central Body** list appears. And after selecting the **Spacecraft Cd** parameter, no dependency list appears. To select a range of parameters from the **Object Properties** list, hold down the Shift key while selecting the second endpoint of the range. To select multiple individual parameters, hold down the Ctrl key while making each selection.

To select a parameter, select the appropriate **Object Type**, the specific resource from the **Object List** or **Attached Hardware List**, the desired parameter from the **Object Properties** list, and the required dependency, and add it to the **Selected Value(s)** list on the right. There are six buttons available to control this list:

- **UP**: Move the selected item in the **Selected Value(s)** list up one position (if allowed).
- **DN**: Move the selected item in the **Selected Value(s)** list down one position (if allowed).
- **->**: Add the selected item in the **Object Properties** list to the **Selected Value(s)** list.

- **<-**: Remove the selected item in the **Selected Value(s)** list.
- **=>**: Add all items to the **Selected Value(s)** list.
- **<=**: Remove all items from the **Selected Value(s)** list.

When finished, the **Selected Value(s)** list contains the final selected parameters. Click **OK** to accept the selection.

The ordering of the **Selected Value(s)** list is significant in certain circumstances (such as in the **Add** field of **ReportFile**), but not in others. See the documentation for each resource or command for details.

Special Considerations

Some resources and commands (such as the **Propagate** command **Parameter** argument) only accept a single parameter as input; in this context the **ParameterSelectDialog** only allows one parameter in the **Selected Value(s)** list and does not allow use of the **UP**, **DN**, and **=>** buttons.

In some instances (such as in the **Vary** command), only parameters that are also fields (and so can be set in the **Mission Sequence**) can be used. In this case only the allowed parameters will be shown in the **Object Properties** list.

In the **Propagate** command **Parameter** argument, only parameters of **Spacecraft** can be used. In this case only **Spacecraft** will be shown in the **Object Type** list.

Parameters

> To Reviewer: These tables will be merged into the parent reference pages and labeled as "Field" or "Parameter". For existing fields, the description already in the spec will be used instead of the description below. Each section below will be linked to the parent reference page.

> To Technical Writer: The "pencil" icon in the following tables is from http://www.famfamfam.com/lab/icons/silk/. They're free to use, but we need to credit them.

User Interface Design Trades for New Parameter Types

This is a temporary design trade study for user interfaces to provide force model and space point parameters. This information will be migrated and formalized once designs are selected.

For each parameter there are ate lease four pieces of information that must be specified.

Force Model Parameters	Space Point Parameters
1. Epoch 2. State 3. Spacecraft properties 4. Force Model 5. Parameter (density, AX,..) 6. Depedency??	1. Epoch 2. Dependency (coord. sys). 3. Body name 4. Parameter (X, VY,...)

Here are some design trades: So far, we have three high level approaches, which are identified in the left hand column. We need to identify Pros and Cons from an interface and implementation perspective to help us make a decision.

Design/Interface Description	Script Examples	Pros

Object Methods.
 	
This approach is entirely new and uses methods on built-in objects to expose data and functionality.
	`density = myForceModel.GetDensity(Sat)`
	
	`density = mySat.GetDensity(ForceModel)`
	
	`myCoordinateSystem.GetRotationMatrix()`
		• Powerful, used nearly all modern programmin languages
		• Easily extendible, unlike parameter approach based on dependency

Built-in math/GMAT functions.
 	
This approach provides more math function interfaces to get to lower level data
	`rho = GetForceModelData(myForce,mySat,'density');`
	
	`density = GetDensity(myForceModel,mySpaceraft);`
	`GetPosition(myLibPoint,'EarthMJ200Eq',epoch)`
	`GetVelocity(myLibPoint,EarthFixed,Sat.A1ModJulian)`
	
	`myImpulsiveBurn.EarthFixed.Element1`
		• Extends existing interface
		• MATLAB-lik

New Parameters.
 	
This approach extends the existing parameter approach
	`mySat.myForceModel.density`
	
	`myLibrationPoint.UTCModJulian = 21345`
	`x = myLibrationPoint.EarthMJ2000Eq.X`
	
	`myLibPoint.mySat.EarthMJ2000Eq.X`
	
	`myLibrationPoint.EarthMJ2000Eq.Z(21451)`
	`myLibrationPoint.EarthMJ2000Eq.Z(A1ModJulian = 21423)`
	
	`myImplulsiveBurn.EarthMJ2000Eq.Element1`
		• Extends existing interface tha users already know

Create Vector myVector

myVector.Type = Position

MyVector.Center = Earth

MyVector.Point = myLibrationPoint

Parameter Select Dialog Behavior for Different Contexts and Parameters

This section describes the different places the parameter select dialog box is used, and the parameter types that should appear in each context.

Summary of Supported Object Types in PSDB

The table below describes at a high level the objects that have parameters, whether those objects support get, set or both, and provides some simple examples.

Object Type	Get/Set	Description	Examples
Spacecraft	Get and Set	Spacecraft hardware, orbit, and attitude related information.	MySat.Earth.SMA
SpacePoint	Get and Set	Ephemeris for space point parameters (except spacecraft). • Planet • Moon • Comet • Asteroid • Librationpoint • BaryCenter • SSB Epoch parameters are set, Position Parameters are Get.	Sun.EarthMJ2000Eq.X
ImpulsiveBurn	Get and Set	ImpulsiveBurn related parameters	MyDeltaV.EarthMJ2000Eq.Element1
Variable	Get and Set	Variable data type	MyVar
Array	Get and Set	Array data type	MyArray, MyArray(2,2)
String	Get and Set	String data type	MyString

Parameter Attribute Definitions

Attribute Name	Definitions
Settable	The quantity can be set by the user. An example that is settable is MySat.X = 5; An example that is NOT settable is MySat.OrbitPeriod = 3600.
Plottable	The quantity is a scalar, numeric value. Examples that are: MySat.X or MyArray(1,1). An example that is NOT Plottable is MySat.Epoch.UTCGregorian.
Reportable	The quantity can be written to a report. For example, an entire Spacecraft can be selected in some contexts, but is not a reportable type.

Parameter Select Dialog Behavior by Context

The parameter select dialog behavior is primarily governed by three questions:

1. Which Resource types are supported by the Resource or Command context in which the PSDB is being used? Not all contexts support all Resource types. For example, the Vary command LHS does not support the String Resource.
2. Is the context getting data or setting data? Many parameters are not settable and when the PSDB is used to select a quantity to set, the available options are fewer than when simply getting a quantity.
3. What parameter attributes are supported by by the Resource or Command context in which the PSDB is being used? For example, the Achieve command does not support string parameters. The FiniteBurnCommand requires selecting an entire spacecraft.

In addition, some contexts do now allow dependencies, and some only allow selecting an entire Resource.

The table below describes the required behavior for all contexts where the PSDB is used.

Resource/Command	Reference	Get/Set	Special Limitations

ReportFile	Parameter list	Get	• Resource Type Restrictions
			  • None
			• Attribute Restrictions
			  • Does not support reporting an entire spacecraft

XYPlot	Selected X	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows selection of a **single parameter**
			  • Only supports type plottable

XYPlot	Selected Y	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows selection of parameters that are of type plottable

Propagate	Stopping condition LHS	Get	• Resource Type Restrictions
			  • Only allows Spacecraft parameters
			• Attribute Restrictions
			  • Only allows selection of parameters that are of type plottable

Propagate	Stopping condition RHS	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows selection of parameters that are of type plottable

Vary	Variable	Set	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Must be settable
			  • Must be plottable
			  • Does not support dependencies (unless default dependecy).

Achieve	Goal	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows plottable quantities

Achieve	Value	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows plottable quantities

Achieve	Tolerance	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows plottable quantities

Report	Parameter List	Get	• Resource Type Restrictions
			  • None
			• Attribute Restrictions
			  • Does not support reporting an entire spacecraft

Call MATLAB Function	Input list	Get	• Resource Type Restrictions
			  • None
			• Attribute Restrictions
			  • Can't send a whole object

Call MATLAB Function	Output list	Set	• Resource Type Restrictions
			  • None
			• Attribute Restrictions
			  • Must be settable
			  • Does not support dependencies (unless default dependecy). I.e. Cannot vary mySat.EarthMJ2000Eq.X

If	LHS	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows Plottable quantities

If	RHS	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows Plottable quantities

For	Index	Set	• Resource Type Restrictions
			  • Must be Resource type Variable
			• Attribute Restrictions
			  • Must be Plottable (satisfied implicitly from Resourse Restriction)
			  • Must be Settable (satisfied implicitly from Resourse Restriction)
			  • Does not support dependencies (unless default dependecy). (satisfied implicitly from Resourse Restriction)

For	Start	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows Plottable quantities

For	Increment	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows Plottable quantities

For	End	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows Plottable quantities

While	LHS	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows Plottable quantities

While	RHS	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows Plottable quantities

NonlinearConstraint	Constraint	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows plottable quantities

NonlinearConstraint	Constraint Value	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows plottable quantities

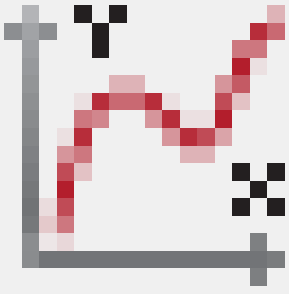
Minimize	Variable to be minimized	Get	• Resource Type Restrictions
			  • Does not allow String Resource type
			• Attribute Restrictions
			  • Only allows plottable quantities

BeginFiniteBurn	Spacecraft	Get	• Resource Type Restrictions
			  • Only available Resource type is Spacecraft.
			• Attribute Restrictions
			  • Can only select an entire spacecraft.

EndFiniteBurn	Spacecraft	Get	• Resource Type Restrictions
			  • Only available Resource type is Spacecraft.
			• Attribute Restrictions
			  • Can only select an entire spacecraft.

Parameter Specifications By Resource

Spacecraft

Parameter			**Units**	**Description**

Acceleration	N	Y	km/s^2	The total acceleration with respect to the inertial system. Data Type: Real Number Dependency: ForceModel.
AccelerationX	N	Y	km/s^2	The x-component of acceleration with respect to the inertial system. Data Type: Real Number Dependency: ForceModel.
AccelerationY	N	Y	km/s^2	The y-component of acceleration with respect to the inertial system. Data Type: Real Number Dependency: ForceModel.
AccelerationX	N	Y	km/s^2	The z-component of acceleration with respect to the inertial system. Data Type: Real Number Dependency: ForceModel.
AtmosDensity	N	Y	kg/km^3	The atmospheric density at the current spacecraft epoch and location. Data Type: Real Number Dependency: ForceModel.
ElapsedSecs	N	Y	s	Spacecraft Data Type: Real Number Dependency: None
ElapsedDays	N	Y	d	Spacecraft Data Type: Real Number Dependency: None
A1ModJulian	Y	Y	d	Spacecraft orbit epoch in the A.1 system and the Modified Julian format. Data Type: Real Number Dependency: None
A1Gregorian	Y	N	N/A	Spacecraft orbit epoch in the A.1 system and the Gregorian format. Data Type: String Dependency: None
TAIModJulian	Y	Y	d	Spacecraft orbit epoch in the TAI system and the Modified Julian format. Data Type: Real Number Dependency: None
TAIGregorian	Y	N	N/A	The spacecraft orbit epoch in the TAI system and the Gregorian format. Data Type: String Dependency: None
TTModJulian	Y	Y	d	The spacecraft orbit epoch in the TT system and the Modified Julian format. Data Type: Real Number Dependency: None
TTGregorian	Y	N	N/A	Spacecraft orbit epoch in the TT system and the Gregorian format. Data Type: String Dependency: None

TDBModJulian	Y	Y	d	Spacecraft orbit epoch in the TDB system and the Modified Julian format. Data Type: Real Number Dependency: None
TDBGregorian	Y	N	N/A	Spacecraft orbit epoch in the TDB system and the Gregorian format. Data Type: String Dependency: None
UTCModJulian	Y	Y	d	Spacecraft orbit epoch in the UTC system and the Modified Julian format. Data Type: Real Number Dependency: None
UTCGregorian	Y	N	N/A	Spacecraft orbit epoch in the UTC system and the Gregorian format. Data Type: String Dependency: None
CurrA1MJD	Y	Y	d	Deprecated. Spacecraft orbit epoch in the A.1 system and the Modified Julian format. Data Type: Real Number Dependency: None
X	Y	Y	km	Cartesian x-component of the spacecraft position. Data Type: Real Number Dependency: Coordinate System
Y	Y	Y	km	Cartesian y-component of the spacecraft position. Data Type: Real Number Dependency: Coordinate System
Z	Y	Y	km	Cartesian z-component of the spacecraft position. Data Type: Real Number Dependency: Coordinate System
VX	Y	Y	km/s	Cartesian x-component of the spacecraft velocity. Data Type: Real Number Dependency: Coordinate System
VY	Y	Y	km/s	Cartesian y-component of the spacecraft velocity. Data Type: Real Number Dependency: Coordinate System
VZ	Y	Y	km/s	Cartesian z-component of the spacecraft velocity. Data Type: Real Number Dependency: Coordinate System
SMA	Y	Y	km	Orbit semi-major axis. Data Type: Real Number Dependency: Central Body
ECC	Y	Y	N/A	Orbit eccentricity. Data Type: Real Number Dependency: Central Body
INC	Y	Y	°	Orbit inclination. Data Type: Real Number Dependency: Coordinate System Output Range: 0° INC 180°
RAAN	Y	Y	°	Orbit right ascension of the ascending node. Data Type: Real Number Dependency: Coordinate System Output Range: 0° RAAN < 360°

AOP	Y	Y	°	Orbit argument of periapsis. Data Type: Real Number Dependency: Coordinate System Output Range: 0° ≤ AOP < 360°
TA	Y	Y	°	True anomaly. Data Type: Real Number Dependency: Central Body Output Range: 0° ≤ TA < 360°
MA	N	Y	°	Mean anomaly. Data Type: Real Number Dependency: Central Body Output Range: 0° ≤ MA < 360° (elliptic orbits)
EA	N	Y	°	Eccentric anomaly. Data Type: Real Number Dependency: Central Body Output Range: 0° ≤ EA < 360°
HA	N	Y	°	Hyperbolic anomaly. Data Type: Real Number Dependency: Central Body
MM	N	Y	rad/s	Mean motion. Data Type: Real Number Dependency: Central Body
VelApoapsis	N	Y	km/s	Scalar velocity at apoapsis. Data Type: Real Number Dependency: Central Body
VelPeriapsis	N	Y	km/s	Scalar velocity at periapsis. Data Type: Real Number Dependency: Central Body
Apoapsis	N	Y	N/A	A parameter that equals zero when the spacecraft is at orbit apoapsis. This parameter can only be used as a stopping condition in the Propagate command. Data Type: Real Number Dependency: Central Body
Periapsis	N	Y	N/A	A parameter that equals zero when the spacecraft is at orbit periapsis. This parameter can only be used as a stopping condition in the Propagate command. Data Type: Real Number Dependency: Central Body
OrbitPeriod	N	Y	s	Osculating orbit period. Data Type: Real Number Dependency: Central Body
RadApo	Y	Y	km	Radius of apoapsis. Data Type: Real Number Dependency: Central Body
RadPer	Y	Y	km	Radius of periapsis. Data Type: Real Number Dependency: Central Body
C3Energy	N	Y	MJ/kg (km^2/s^2)	C_3 (characteristic) energy. Data Type: Real Number Dependency: Central Body
Energy	N	Y	MJ/kg (km^2/s^2)	Specific orbital energy. Data Type: Real Number Dependency: Central Body

RMAG	Y	Y	km	Magnitude of the orbital position vector. Data Type: Real Number Dependency: Central Body
RA	Y	Y	°	Right ascension of the orbital position. Data Type: Real Number Dependency: Coordinate System Output Range: -180° RA 180°
DEC	Y	Y	°	Declination of the orbital position. Data Type: Real Number Dependency: Coordinate System Output Range: -90° DEC 90°
VMAG	Y	Y	km/s	Magnitude of the orbital velocity vector. Data Type: Real Number Dependency: Coordinate System
RAV	Y	Y	°	Right ascension of orbital velocity. Data Type: Real Number Dependency: Coordinate System Output Range: -180° RAV 180°
DECV	Y	Y	°	Declination of orbital velocity. Data Type: Real Number Dependency: Coordinate System Output Range: -90° DECV 90°
AZI	Y	Y	°	Orbital velocity azimuth. Data Type: Real Number Dependency: Coordinate System Output Range: -180° AZI 180°
FPA	Y	Y	°	Orbital flight path angle. Data Type: Real Number Dependency: Coordinate System Output Range: 0° FPA 180°
EquinoctialH	Y	Y	N/A	Equinoctial H element. Data Type: Real Number Dependency: Coordinate System
EquinoctialK	Y	Y	N/A	Equinoctial K element. Data Type: Real Number Dependency: Coordinate System
EquinoctialP	Y	Y	N/A	Equinoctial P element. Data Type: Real Number Dependency: Coordinate System
EquinoctialQ	Y	Y	N/A	Equinoctial Q element. Data Type: Real Number Dependency: Coordinate System
MLONG	Y	Y	°	Mean longitude. Data Type: Real Number Dependency: Coordinate System Output Range: 0° MLONG < 360°
SemilatusRectum	N	Y	km	Semilatus rectum of the osculating orbit. Data Type: Real Number Dependency: Central Body
HMAG	N	Y	km^2/s	Magnitude of the angular momentum vector. Data Type: Real Number Dependency: Central Body

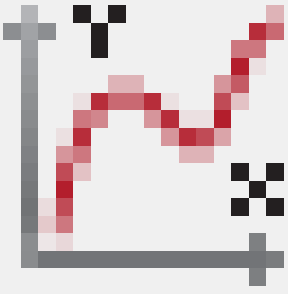
HX	N	Y	km^2/s	X component of the angular momentum vector. Data Type: Real Number Dependency: Coordinate System
HY	N	Y	km^2/s	Y component of the angular momentum vector. Data Type: Real Number Dependency: Coordinate System
HZ	N	Y	km^2/s	Z component of the angular momentum vector. Data Type: Real Number Dependency: Coordinate System
DLA	N	Y	°	Declination of the outgoing hyperbolic asymptote. Data Type: Real Number Dependency: Coordinate System Output Range: -90° DLA 90°
RLA	N	Y	°	Right ascension of the outgoing hyperbolic asymptote. Data Type: Real Number Dependency: Coordinate System Output Range: -180° RLA 180°
Altitude	N	Y	km	Distance to the plane tangent to the surface of the specified celestial body at the sub-satellite point. GMAT assumes the body is an ellipsoid. Data Type: Real Number Dependency: Central Body
MHA	N	Y	°	Angle between celestial body's body-fixed and inertial axes. For Earth, this is the Greenwich Hour Angle. Data Type: Real Number Dependency: Central Body Output Range: 0° MHA < 360°
Longitude	N	Y	°	Planetodetic longitude. Data Type: Real Number Dependency: Central Body Output Range: -180° Longitude 180°
Latitude	N	Y	°	Planetodetic latitude. Data Type: Real Number Dependency: Central Body Output Range: -90° Latitude 90°
LST	N	Y	°	Local sidereal time of the spacecraft from the celestial body's inertial x-axis. Data Type: Real Number Dependency: Central Body Output Range: 0° LST < 360°
BetaAngle	N	Y	°	Beta angle (or phase angle) between the orbit normal vector and the vector from the celestial body to the sun. Data Type: Real Number Dependency: Central Body Output Range: -90° BetaAngle 90°
BdotT	N	Y	km	B-plane **B**·**T** magnitude. See the BdotR parameter for notes on this calculation. Data Type: Real Number Dependency: Coordinate System

BdotR	N	Y	km	B-plane **B**·**R** magnitude. GMAT computes the B-plane coordinates in the coordinate system specified in the dependency. In many implementations, the B-plane coordinates are computed in a pseudo-rotating coordinate system where the ×r term is not applied when transforming velocity vectors. GMAT does apply the ×r term in the velocity transformation. When computing B-plane coordinates in inertial systems, this term is identically zero. For rotating systems such as the Sun-Earth body-body rotating system, the effect of including ×r is small but noticeable when comparing results between systems. When the rotation of the selected coordinate system is "fast", the values may differ significantly. Data Type: Real Number Dependency: Coordinate System
BVectorMag	N	Y	km	B-plane **B** vector magnitude. See the BdotR parameter for notes on this calculation. Data Type: Real Number Dependency: Coordinate System
BVectorAngle	N	Y	°	B-plane angle between the **B** vector and the **T** unit vector. See the BdotR parameter for notes on this calculation. Data Type: Real Number Dependency: Coordinate System Output Range: -180° BVectorAngle 180°
DCM11	Y	Y	(None)	Element (1,1) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM12	Y	Y	(None)	Element (1,2) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM13	Y	Y	(None)	Element (1,3) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM21	Y	Y	(None)	Element (2,1) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM22	Y	Y	(None)	Element (2,2) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM23	Y	Y	(None)	Element (2,3) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM31	Y	Y	(None)	Element (3,1) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM32	Y	Y	(None)	Element (3,2) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
DCM33	Y	Y	(None)	Element (3,3) of the attitude direction cosine matrix. Data Type: Real Number Dependency: (None)
EulerAngle1	Y	Y	°	Attitude Euler angle 1. Data Type: Real Number Dependency: (None) Output Range: 0° EulerAngle1 < 360°

EulerAngle2	Y	Y	°	Attitude Euler angle 2. Data Type: Real Number Dependency: (None) Output Range: 0° EulerAngle2 < 360°
EulerAngle3	Y	Y	°	Attitude Euler angle 3. Data Type: Real Number Dependency: (None) Output Range: 0° EulerAngle3 < 360°
MRP1	Y	Y	(None)	Attitude modified Rodrigues parameter 1. Data Type: Real Number Dependency: (None)
MRP2	Y	Y	(None)	Attitude modified Rodrigues parameter 2. Data Type: Real Number Dependency: (None)
MRP3	Y	Y	(None)	Attitude modified Rodrigues parameter 3. Data Type: Real Number Dependency: (None)
Q1	N	Y	(None)	Attitude quaternion element 1 (*a*). Data Type: Real Number Dependency: (None)
Q2	N	Y	(None)	Attitude quaternion element 2 (*b*). Data Type: Real Number Dependency: (None)
Q3	N	Y	(None)	Attitude quaternion element 3 (*c*). Data Type: Real Number Dependency: (None)
Q4	N	Y	(None)	Attitude quaternion element 4 (*d*). Data Type: Real Number Dependency: (None)
Quaternion	Y	N	(None)	Attitude quaternion. Data Type: Array (1×4) Dependency: (None)
AngularVelocityX	Y	Y	°/s	X component of the attitude angular velocity vector. Data Type: Real Number Dependency: (None)
AngularVelocityY	Y	Y	°/s	Y component of the attitude angular velocity vector. Data Type: Real Number Dependency: (None)
AngularVelocityZ	Y	Y	°/s	Z component of the attitude angular velocity vector. Data Type: Real Number Dependency: (None)
EulerAngleRate1	Y	Y	°/s	Rate of attitude Euler angle 1. Data Type: Real Number Dependency: (None)
EulerAngleRate2	Y	Y	°/s	Rate of attitude Euler angle 2. Data Type: Real Number Dependency: (None)
EulerAngleRate3	Y	Y	°/s	Rate of attitude Euler angle 3. Data Type: Real Number Dependency: (None)

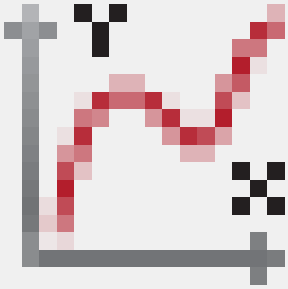
DryMass	Y	Y	kg	Dry mass (without propellant). Data Type: Real Number Dependency: (None)
Cd	Y	Y	(None)	Coefficient of drag. Data Type: Real Number Dependency: (None)
Cr	Y	Y	(None)	Coefficient of reflectivity. Data Type: Real Number Dependency: (None)
DragArea	Y	Y	m^2	Area used to compute acceleration due to atmospheric drag. Data Type: Real Number Dependency: (None)
SRPArea	Y	Y	m^2	Area used to compute acceleration due to solar radiation pressure. Data Type: Real Number Dependency: (None)
TotalMass	N	Y	kg	Total mass, including fuel mass from attached Fuel Tank resources. Data Type: Real Number Dependency: (None)
OrbitSTM	N	N	(None)	State transition matrix. Data Type: Array (6×6) Dependency: Coordinate System
OrbitSTMA	N	N	(None)	Upper-left quadrant of the state transition matrix. Data Type: Array (3×3) Dependency: Coordinate System
OrbitSTMB	N	N	(None)	Upper-right quadrant of the state transition matrix. Data Type: Array (3×3) Dependency: Coordinate System
OrbitSTMC	N	N	(None)	Lower-left quadrant of the state transition matrix. Data Type: Array (3×3) Dependency: Coordinate System
OrbitSTMD	N	N	(None)	Lower-right quadrant of the state transition matrix. Data Type: Array (3×3) Dependency: Coordinate System

FuelTank

Parameter			**Units**	**Description**
FuelMass	Y	Y	kg	Mass of fuel in the tank. Data Type: Real Number Dependency: (None)

Volume	Y	Y	m^3	Volume of the tank. GMAT checks to ensure that the input volume of the tank is larger than the calculated volume of fuel loaded in the tank and throws an exception in the case that the calculated fuel volume is larger than the input tank volume. Data Type: Real Number Dependency: (None)
FuelDensity	Y	Y	kg/m^3	Density of the fuel. Data Type: Real Number Dependency: (None)
Pressure	Y	Y	kPa	Pressure in the tank. Data Type: Real Number Dependency: (None)
Temperature	Y	Y	°C	Temperature of the fuel and ullage in the tank. GMAT currently assumes ullage and fuel are always at the same temperature. Data Type: Real Number Dependency: (None)
RefTemperature	Y	Y	°C	The temperature of the tank when fuel was loaded. Data Type: Real Number Dependency: (None)

Thruster

Parameter			**Units**	**Description**
DutyCycle	Y	Y	(None)	Fraction of time that the thrusters are on during a maneuver. The thrust applied to the spacecraft is scaled by this amount. Note that this scale factor also affects mass flow rate. Data Type: Real Number Dependency: (None)
ThrustScaleFactor	Y	Y	(None)	Scale factor that is multiplied by the thrust vector, for a given thruster, before the thrust vector is added into the total acceleration. Note that the value of this scale factor does not affect the mass flow rate. Data Type: Real Number Dependency: (None)
GravitationalAccel	Y	Y	m/s^2	Value of the gravitational acceleration used for the FuelTank/Thruster calculations. Data Type: Real Number Dependency: (None)
C1	Y	Y	N	Thrust coefficient C_1. Data Type: Real Number Dependency: (None)
C2	Y	Y	N/kPa	Thrust coefficient C_2. Data Type: Real Number Dependency: (None)

C3	Y	Y	N	Thrust coefficient C_3. Data Type: Real Number Dependency: (None)
C4	Y	Y	N/kPa	Thrust coefficient C_4. Data Type: Real Number Dependency: (None)
C5	Y	Y	N/kPa^2	Thrust coefficient C_5. Data Type: Real Number Dependency: (None)
C6	Y	Y	N/kPa^{C_7}	Thrust coefficient C_6. Data Type: Real Number Dependency: (None)
C7	Y	Y	(None)	Thrust coefficient C_7. Data Type: Real Number Dependency: (None)
C8	Y	Y	N/kPa^{C_9}	Thrust coefficient C_8. Data Type: Real Number Dependency: (None)
C9	Y	Y	(None)	Thrust coefficient C_9. Data Type: Real Number Dependency: (None)
C10	Y	Y	$N/kPa^{C_{11}}$	Thrust coefficient C_{10}. Data Type: Real Number Dependency: (None)
C11	Y	Y	(None)	Thrust coefficient C_{11}. Data Type: Real Number Dependency: (None)
C12	Y	Y	N	Thrust coefficient C_{12}. Data Type: Real Number Dependency: (None)
C13	Y	Y	(None)	Thrust coefficient C_{13}. Data Type: Real Number Dependency: (None)
C14	Y	Y	1/kPa	Thrust coefficient C_{14}. Data Type: Real Number Dependency: (None)
C15	Y	Y	(None)	Thrust coefficient C_{15}. Data Type: Real Number Dependency: (None)
C16	Y	Y	1/kPa	Thrust coefficient C_{16}. Data Type: Real Number Dependency: (None)
K1	Y	Y	s	I_{sp} coefficient K_1. Data Type: Real Number Dependency: (None)
K2	Y	Y	s/kPa	I_{sp} coefficient K_2. Data Type: Real Number Dependency: (None)

K3	Y	Y	s	I_{sp} coefficient K_3. Data Type: Real Number Dependency: (None)
K4	Y	Y	s/kPa	I_{sp} coefficient K_4. Data Type: Real Number Dependency: (None)
K5	Y	Y	s/kPa^2	I_{sp} coefficient K_5. Data Type: Real Number Dependency: (None)
K6	Y	Y	s/kPa^{C_7}	I_{sp} coefficient K_6. Data Type: Real Number Dependency: (None)
K7	Y	Y	(None)	I_{sp} coefficient K_7. Data Type: Real Number Dependency: (None)
K8	Y	Y	s/kPa^{C_9}	I_{sp} coefficient K_8. Data Type: Real Number Dependency: (None)
K9	Y	Y	(None)	I_{sp} coefficient K_9. Data Type: Real Number Dependency: (None)
K10	Y	Y	$s/kPa^{C_{11}}$	I_{sp} coefficient K_{10}. Data Type: Real Number Dependency: (None)
K11	Y	Y	(None)	I_{sp} coefficient K_{11}. Data Type: Real Number Dependency: (None)
K12	Y	Y	s	I_{sp} coefficient K_{12}. Data Type: Real Number Dependency: (None)
K13	Y	Y	(None)	I_{sp} coefficient K_{13}. Data Type: Real Number Dependency: (None)
K14	Y	Y	1/kPa	I_{sp} coefficient K_{14}. Data Type: Real Number Dependency: (None)
K15	Y	Y	(None)	I_{sp} coefficient K_{15}. Data Type: Real Number Dependency: (None)
K16	Y	Y	1/kPa	I_{sp} coefficient K_{16}. Data Type: Real Number Dependency: (None)
ThrustDirection1	Y	Y	(None)	ThrustDirection1, divided by the RSS of the three direction components, forms the x component of the spacecraft thrust vector direction. Data Type: Real Number Dependency: (None)

ThrustDirection2	Y	Y	(None)	ThrustDirection2, divided by the RSS of the three direction components, forms the y component of the spacecraft thrust vector direction. Data Type: Real Number Dependency: (None)
ThrustDirection3	Y	Y	(None)	ThrustDirection3, divided by the RSS of the three direction components, forms the z component of the spacecraft thrust vector direction. Data Type: Real Number Dependency: (None)

ImpulsiveBurn

To compute ImpulsiveBurn parameters, GMAT requires that an ImpulsiveBurn has been executed using a Maneuver command like this

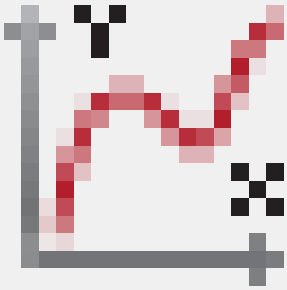
Manevuer myImpulsiveBurn(mySat)

In the case that an ImpulsiveBurn has not been applied, GMAT will output zeros for the maneuver components and issue a warning. We recommended that you evaluate maneuver parameters immediately after maneuvers are applied using the Maneuver command like this

Manevuer myImpulsiveBurn(mySat)

myVar = mySat.MyCoordinateSystem.Element1

This avoids issues that may occur if the manuever coordinate system is time varying, and the maneuver parameters are requested after further manipulation of the participants using other commands (such as Propagate). In that case, it is possible that the participants are no longer at the epoch of the maneuver, and unexpected results can occur due to epoch mismatches.

Parameter			**Units**	**Description**
Element1	Y	Y	(None)	X-component of the applied impulsive burn (delta-V). Data Type: Real Number Dependency: CoordinateSystem
Element2	Y	Y	(None)	Y-component of the applied impulsive burn (delta-V). Data Type: Real Number Dependency: CoordinateSystem
Element3	Y	Y	(None)	Z-component of the applied impulsive burn (delta-V). Data Type: Real Number Dependency: CoordinateSystem
V	Y	Y	(None)	Deprecated. X-component of the applied impulsive burn (delta-V). If the Axes of the ImpulsiveBurn is not VNB, this parameter returns the x-component of the burn, not the velocity component. Data Type: Real Number Dependency: (None)
N	Y	Y	(None)	Deprecated. Y-component of the applied impulsive burn (delta-V). If the Axes of the ImpulsiveBurn is not VNB, this parameter returns the y-component of the burn, not the normal component. Data Type: Real Number Dependency: (None)

B	Y	Y	(None)	Deprecated. Z-component of the applied impulsive burn (delta-V). If the Axes of the ImpulsiveBurn is not VNB, this parameter returns the z-component of the burn, not the bi-normal component. Data Type: Real Number Dependency: (None)

Array, String, Variable

Array, String, and Variable resources are themselves parameters, and can be used as any other parameter would. All of these are writable parameters, though only Variable resources and individual elements of Array resources can be plotted.

Examples

Using parameters in the Mission Sequence:

```
Create Spacecraft aSat
Create Propagator aProp
Create ReportFile aReport

BeginMissionSequence

% propagate for 100 steps
For i=1:100
 Propagate aProp(aSat)
 % write four parameters (one standalone, three
coordinate-system-dependent) to a file
 Report aReport aSat.TAIGregorian aSat.EarthFixed.X aSat.EarthFixed.Y
aSat.EarthFixed.Z
EndFor
```

Using parameters as plot data:

```
Create Spacecraft aSat
Create Propagator aProp

Create XYPlot aPlot
aPlot.XVariable = aSat.TAIModJulian
aPlot.YVariables = {aSat.Earth.Altitude, aSat.Earth.ECC}

BeginMissionSequence

% propagate for 100 steps
For i=1:100
 Propagate aProp(aSat)
EndFor
```

Using parameters as stopping conditions:

```
Create Spacecraft aSat
aSat.SMA = 6678

Create ForceModel anFM
anFM.Drag.AtmosphereModel = MSISE90

Create Propagator aProp
aProp.FM = anFM

BeginMissionSequence

Propagate aProp(aSat) {aSat.Earth.Altitude = 100, aSat.ElapsedDays = 365}
```

Math Spec

Calculation Objects

Simple Parameters

MHA

Description: MHA is the mean hour angle of the x-axis of the selected central body's inertial equatorial frame, measured from the body's prime meridian.

Dependency:

Given:

Find:

Test Procedures

I generated truth data using STK 9 through the STK Object Model. These scripts are located in `extern\Resources\FRR-38_CalculationParameters\stk`. I matched data between the two tools as follows:

1. Central-body physical properties (mu, radii) are configured in the STK .cb files in the scenario directory, based on default GMAT data.
2. I used EOP data for GMAT and STK from 2012-10-30. I had to replace the default application-wide data for each tool.
3. Some mu values are hard-coded in the GMAT and STK scripts. These are all set to the GMAT default mu values.

The following table lists all available parameters and where each is tested (if at all). Calculation tests are tests that perform some calculation that affects the parameter. Read tests check that the parameter can be read in a script. Write tests check that the parameter can be assigned to in the Mission Sequence.

If a script name appears in a column, the parameter is tested as a part of the Calculation Parameters tests (specifically in the named script). If another requirement appears, then the parameter is being tested as a part of the test suite for that requirement. If (N/A) appears, that test type does not apply to the parameter. A ( ) indicates that the parameter is not currently being tested, and that it should be tested as a part of the Calculation Parameters tests unless another requirement is listed afterwards.

There is an assumption being made that the ability to plot plottable parameters is being tested as part of XYPlot QA.

Even if a parameter has a test name listed, it needs work until the following are true:

- central-body parameters are tested using:
    - central body of the Spacecraft CoordinateSystem

    - another planet (Venus or Earth)
    - Luna (or another planet)
    - user-defined body (UserMars or UserSaturn)
- coordinate-system parameters are tested using:
    - Spacecraft CoordinateSystem
    - EarthMJ2000Eq (or Venus)
    - EarthFixed
    - user-created body-fixed (central body of the Spacecraft CoordinateSystem or Saturn)

All tests that need work by any of the above criteria are highlighted red in the table. Finished tests are highlighted green.

Parameter	Calculation Tests	Read Tests	Write Tests
ElapsedSecs	SpacecraftElapsedTimeParams	SpacecraftElapsedTimeParams	(N/A)
ElapsedDays	SpacecraftElapsedTimeParams	SpacecraftElapsedTimeParams	(N/A)
A1ModJulian	FRR-2	FRR-2	FRR-2
A1Gregorian	FRR-2	FRR-2	FRR-2
TAIModJulian	FRR-2	FRR-2	FRR-2
TAIGregorian	FRR-2	FRR-2	FRR-2
TTModJulian	FRR-2	FRR-2	FRR-2
TTGregorian	FRR-2	FRR-2	FRR-2
TDBModJulian	FRR-2	FRR-2	FRR-2
TDBGregorian	FRR-2	FRR-2	FRR-2
UTCModJulian	FRR-2	FRR-2	FRR-2
UTCGregorian	FRR-2	FRR-2	FRR-2
CurrA1MJD	FRR-2	FRR-2	FRR-2
X	FRR-1	FRR-1	FRR-1
Y	FRR-1	FRR-1	FRR-1
Z	FRR-1	FRR-1	FRR-1
VX	FRR-1	FRR-1	FRR-1
VY	FRR-1	FRR-1	FRR-1
VZ	FRR-1	FRR-1	FRR-1
SMA	FRR-1	FRR-1	FRR-1
ECC	FRR-1	FRR-1	FRR-1
INC	FRR-1	FRR-1	FRR-1
RAAN	FRR-1	FRR-1	FRR-1
AOP	FRR-1	FRR-1	FRR-1
TA	FRR-1	FRR-1	FRR-1
MA	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
EA	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_EA	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_EA	(N/A)
HA	CbParams_*_2Body_* CbParams_GEO_* HyperbolicAnomaly_Valid	CbParams_*_2Body_* CbParams_GEO_* HyperbolicAnomaly_Valid	(N/A)
MM	CBParams_Earth_MM CbParams_*_2Body_* CbParams_GEO_*	CBParams_Earth_MM CbParams_*_2Body_* CbParams_GEO_*	(N/A)
VelApoapsis	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_VelApoapsis	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_VelApoapsis	(N/A)

VelPeriapsis	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_VelPeriapsis	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_VelPeriapsis	(N/A)
Apoapsis	FRC-10	FRC-10	(N/A)
Periapsis	FRC-10	FRC-10	(N/A)
OrbitPeriod	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
RadApo	FRR-1	FRR-1	FRR-1
RadPer	FRR-1	FRR-1	FRR-1
C3Energy	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
Energy	CBParams_Earth_OrbitEnergy CbParams_*_2Body_* CbParams_GEO_*	CBParams_Earth_OrbitEnergy CbParams_*_2Body_* CbParams_GEO_*	(N/A)
RMAG	FRR-1	FRR-1	FRR-1
RA	FRR-1	FRR-1	FRR-1
DEC	FRR-1	FRR-1	FRR-1
VMAG	FRR-1	FRR-1	FRR-1
RAV	FRR-1	FRR-1	FRR-1
DECV	FRR-1	FRR-1	FRR-1
AZI	FRR-1	FRR-1	FRR-1
FPA	FRR-1	FRR-1	FRR-1
EquinoctialH	FRR-1	FRR-1	FRR-1
EquinoctialK	FRR-1	FRR-1	FRR-1
EquinoctialP	FRR-1	FRR-1	FRR-1
EquinoctialQ	FRR-1	FRR-1	FRR-1
MLONG	FRR-1	FRR-1	FRR-1
SemilatusRectum	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_SemiLatusRectum	CbParams_*_2Body_* CbParams_GEO_* CBParams_Earth_SemiLatusRectum	(N/A)
HMAG	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
HX	CSParams_*_2Body CSParams_*_2Body_*	CSParams_*_2Body CSParams_*_2Body_*	(N/A)
HY	CSParams_*_2Body CSParams_*_2Body_*	CSParams_*_2Body CSParams_*_2Body_*	(N/A)
HZ	CSParams_*_2Body CSParams_*_2Body_*	CSParams_*_2Body CSParams_*_2Body_*	(N/A)
DLA	SelectedHyperbolicParams_*	SelectedHyperbolicParams_*	(N/A)
RLA	SelectedHyperbolicParams_*	SelectedHyperbolicParams_*	(N/A)
Altitude	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
MHA	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
Longitude	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
Latitude	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
LST	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)
BetaAngle	CbParams_*_2Body_* CbParams_GEO_*	CbParams_*_2Body_* CbParams_GEO_*	(N/A)

BdotT	SelectedHyperbolicParams_*	SelectedHyperbolicParams_*	(N/A)
BdotR	SelectedHyperbolicParams_*	SelectedHyperbolicParams_*	(N/A)
BVectorMag	SelectedHyperbolicParams_*	SelectedHyperbolicParams_*	(N/A)
BVectorAngle	SelectedHyperbolicParams_*	SelectedHyperbolicParams_*	(N/A)
DCM11	FRR-3	FRR-3	✗ (FRR-3)
DCM12	FRR-3	FRR-3	✗ (FRR-3)
DCM13	FRR-3	FRR-3	✗ (FRR-3)
DCM21	FRR-3	FRR-3	✗ (FRR-3)
DCM22	FRR-3	FRR-3	✗ (FRR-3)
DCM23	FRR-3	FRR-3	✗ (FRR-3)
DCM31	FRR-3	FRR-3	✗ (FRR-3)
DCM32	FRR-3	FRR-3	✗ (FRR-3)
DCM33	FRR-3	FRR-3	✗ (FRR-3)
EulerAngle1	FRR-3	FRR-3	✗ (FRR-3)
EulerAngle2	FRR-3	FRR-3	✗ (FRR-3)
EulerAngle3	FRR-3	FRR-3	✗ (FRR-3)
MRP1	FRR-3	FRR-3	✗ (FRR-3)
MRP2	FRR-3	FRR-3	✗ (FRR-3)
MRP3	FRR-3	FRR-3	✗ (FRR-3)
Q1	FRR-3	FRR-3	(N/A)
Q2	FRR-3	FRR-3	(N/A)
Q3	FRR-3	FRR-3	(N/A)
Q4	FRR-3	FRR-3	(N/A)
Quaternion	✗ (FRR-3)	✗ (FRR-3)	✗ (FRR-3)
AngularVelocityX	FRR-3	FRR-3	✗ (FRR-3)
AngularVelocityY	FRR-3	FRR-3	✗ (FRR-3)
AngularVelocityZ	FRR-3	FRR-3	✗ (FRR-3)
EulerAngleRate1	FRR-3	FRR-3	✗ (FRR-3)
EulerAngleRate2	FRR-3	FRR-3	✗ (FRR-3)
EulerAngleRate3	FRR-3	FRR-3	✗ (FRR-3)
DryMass	(N/A)	SpacecraftPhysicalParams	SpacecraftPhysicalParams
Cd	(N/A)	SpacecraftPhysicalParams	SpacecraftPhysicalParams
Cr	(N/A)	SpacecraftPhysicalParams	SpacecraftPhysicalParams
DragArea	(N/A)	SpacecraftPhysicalParams	SpacecraftPhysicalParams
SRPArea	(N/A)	SpacecraftPhysicalParams	SpacecraftPhysicalParams
TotalMass	FRR-11 FRR-9	FRR-11 FRR-9	(N/A)
OrbitSTM	FRR-13	FRR-13	(N/A)
OrbitSTMA	Params_STMSubsets_*	Params_STMSubsets_*	(N/A)

OrbitSTMB	Params_STMSubsets_*	Params_STMSubsets_*	(N/A)
OrbitSTMC	Params_STMSubsets_*	Params_STMSubsets_*	(N/A)
OrbitSTMD	Params_STMSubsets_*	Params_STMSubsets_*	(N/A)
FuelTank.FuelMass	FRR-11 FRR-9	TankParams ReportSatHardware	ReportSatHardware
FuelTank.Volume	(FRR-8)	TankParams ReportSatHardware	ReportSatHardware
FuelTank.FuelDensity	(N/A)	TankParams ReportSatHardware	ReportSatHardware
FuelTank.Pressure	(FRR-8)	TankParams ReportSatHardware	ReportSatHardware
FuelTank.Temperature	(N/A)	TankParams ReportSatHardware	ReportSatHardware
FuelTank.RefTemperature	(N/A)	TankParams ReportSatHardware	ReportSatHardware
Thruster.DutyCycle	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.ThrustScaleFactor	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.GravitationalAccel	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C1	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C2	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C3	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C4	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C5	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C6	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C7	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C8	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C9	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C10	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C11	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C12	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C13	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C14	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C15	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.C16	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K1	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K2	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K3	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K4	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K5	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K6	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K7	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K8	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K9	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K10	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K11	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K12	(N/A)	ReportSatHardware	ReportSatHardware

Thruster.K13	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K14	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K15	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.K16	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.ThrustDirection1	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.ThrustDirection2	(N/A)	ReportSatHardware	ReportSatHardware
Thruster.ThrustDirection3	(N/A)	ReportSatHardware	ReportSatHardware
ImpulsiveBurn.Element1	(N/A)	ImpulsiveBurnParams_VNB_Elements	ImpulsiveBurnParams_VNB_Elements
ImpulsiveBurn.Element2	(N/A)	ImpulsiveBurnParams_VNB_Elements	ImpulsiveBurnParams_VNB_Elements
ImpulsiveBurn.Element3	(N/A)	ImpulsiveBurnParams_VNB_Elements	ImpulsiveBurnParams_VNB_Elements
ImpulsiveBurn.V	(N/A)	ImpulsiveBurnParams_VNB_VNBComponents	ImpulsiveBurnParams_VNB_VNBCompon
ImpulsiveBurn.N	(N/A)	ImpulsiveBurnParams_VNB_VNBComponents	ImpulsiveBurnParams_VNB_VNBCompon
ImpulsiveBurn.B	(N/A)	ImpulsiveBurnParams_VNB_VNBComponents	ImpulsiveBurnParams_VNB_VNBCompon

Edge/Corner/Stress

Priority	Status	Summary
		Test parameters after setting epoch manually, instead of through propagation.

Unique Validation

Priority	Status	Summary
	✅	Set non-settable params
	✅	Set param w/ dependency
	✅	Read w/ incorrect dependency type
	✅	Set w/ incorrect data type
	✅	Set/get w/ incorrect parent (attached hardware)
	✅	Set/get w/ incorrect resource type
	✅	Plot non-plottable params

Unique Mode Tests

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

Space Point Parameter Test Procedures.

This list documents test written for Spacepoint parameters for epoch and cartesian state parameters of Barycenter, LibrationPoint, etc.

- System Tests
  - Input Tests
    - Done: Try setting non-default epoch in all supported formats and test output on a non-spacecraft-type space point (i.e. Libration Point or Barycenter
  - Output Tests
    - Test all parameters on all space point types (Spacecraft is already done)
      - Done: Position parameters
      - Done: (Bug 3973) Time Parameters
    - Test Multiple Coordinate systems for for a mixture of space point types. (EarthFixed, Moon Ecliptic, etc. for LibrationPoint, Barycenter, CelestialBody)
    - Done: Test default dependency (EarthMJ2000Eq) on a few space point types
- Validation
  - Done: Test invalid dependency is caught
  - Done: Test invalid parameter that works for spacecraft is caught

Forcem Model Parameter Test Procedures.

- System Tests
  - Input Tests
    - Try all force types in the force model
      - Harmonic gravity
        - FMParams_GMAT_ISS_EarthSunLuna_EGM96_MSISE90_SRP
        - FMParams_GMAT_GEO_EarthSunLuna_EGM96_MSISE90_SRP_SolidAndPoleTide
      - Point masses
        - FMParams_GMAT_ISS_EarthSunLuna_EGM96_MSISE90_SRP
      - Drag
        - FMParams_GMAT_ISS_EarthSunLuna_EGM96_MSISE90_SRP
      - SRP
        - FMParams_GMAT_ISS_EarthSunLuna_EGM96_MSISE90_SRP
      - Relativity
        - FMParams_GMAT_ISS_Earth_Relativity
      - Thrust
        - Thruster_FullPoly_EarthSat_EarthProp_3EarthThrusters_3CS_2
    - Try different central bodies in the force model
      - FMParams_GMAT_ISS_EarthSunLuna_EGM96_MSISE90_SRP
      - FM_Params_GMAT_Mars1_AllPlanets_0_0_0_Short
    - Try one case with Spacecraft about one body and force model about another
  - Output Tests
    - Test all three acceleration parameters
      - Tested in all scripts
    - Test total acceleration parameter
      - FMParams_GMAT_Mars1_AllPlanets_0_0_0_Short_TotalAccel
- Validation
  - Test invalid dependency is caught