General Mission Analysis Tool (GMAT)

Training Manual

General Mission Analysis Tool (GMAT): Training Manual

Korean Aerospace Research Institute

Table of Contents

Preface

The GMAT Training Manual contains material for new and experienced users and is organized into the following sections:

- Introduction
- Creating Your First Mission
- How-tos
- Tutorials

Introduction

The Introduction section contains two major parts: *Introduction to GMAT* and *Getting Started*.

The *Introduction to GMAT* section contains a brief project and software overview and discusses project status, licensing, and contributors.

The *Getting Started* section describes how to install and start GMAT, presents an overview of the user interfaces, and provides information on configuring your system.

> **Note**
>
> We consider the the section called "User Interfaces Overview" essential reading. If you read nothing else, at least read this section as it will explain the basic philosophy and rules of GMAT's user interfaces.

Creating Your First Mission

The Creating Your First Mission section walks you step-by-step through a sample mission, including creating a spacecraft, a propagator, and an `OrbitView` graphical display, and propagating the spacecraft to orbit perigee.

How-tos

The How-tos section contains many short articles that each describe a single area of functionality. The purpose is of the how-to documentation is to show you how to use a specific feature in an analysis context, and these articles often start from the default mission that is loaded when you start GMAT. A how-to article is designed to take about five minutes to teach you how to perform a specific task.

Tutorials

The Tutorials section describes how to use GMAT for end-to-end analysis. Tutorials are designed to teach you how to use GMAT in the context of performing a real-world analysis, and are intended to take between 30 minutes and one day to complete. Each tutorial as a difficulty level, an approximate duration, and potentially a number of prerequisites, all of which are listed in its introduction.

Part I. Introduction

Table of Contents

Introduction to GMAT

GMAT is an open source trajectory design and optimization system developed by NASA and private industry. It is developed in an open source process to maximize technology transfer and permit anyone to develop and validate new algorithms and to enable those new algorithms to quickly transition into the high fidelity core.

GMAT is designed to model and optimize spacecraft trajectories in flight regimes ranging from low Earth orbit to lunar, interplanetary, and other deep space missions. The system supports constrained and unconstrained trajectory optimization and built-in features make defining cost and constraint functions trivial. GMAT also contains initial value solvers (propagators) and boundary value solvers and efficiently propagates spacecraft either singly or as coupled sets. GMAT's propagators naturally synchronize the epochs of multiple vehicles and shorten run times by avoiding fixed step integration and interpolation.

Users can interact with GMAT using either a graphical user interface (GUI) or a custom scripting language modeled after the syntax used in The MathWorks' MATLAB® system. All of the system elements can be expressed through either interface, and users can convert between the two in either direction.

Analysts model space missions in GMAT by first creating and configuring resources such as spacecraft, propagators, optimizers, and data files. These resources are then used in a Mission Sequence to model the trajectory of the spacecraft and simulate mission events. The mission sequence supports commands such as nonlinear constraints, minimization, propagatation, GMAT or MATLAB functions, inline equations, and script events.

GMAT can display trajectories in a realistic three-dimensional view, plot parameters against one another, and save parameters to files for later processing. The trajectory and plot capabilities are fully interactive, plotting data as a mission is run and allowing users to zoom into regions of interest. Trajectories and data can be viewed in any coordinate system defined in GMAT, and GMAT allows users to rotate the view and set the focus to any object in the display. The trajectory view can be animated so users can watch the evolution of the trajectory over time.

Licensing

GMAT is licensed under the NASA Open Source Agreement v1.3. The license text is contained in the file `License.txt` in root directory of the GMAT distribution, and is listed in Appendix A.

Platforms

GMAT is implemented to run on Windows, Linux and Macintosh platforms, using the wxWidgets cross platform UI toolkit, and can be built using either Microsoft Visual Studio or the GNU Compiler Collection (GCC). GMAT is written in ANSI standard C++ (approximately 380,000 non-comment source lines of code) using an object-oriented methodology, with a rich class structure designed to make new features simple to incorporate. On Windows and Linux, GMAT does not call any operating-system-unique functions or methods. Calls to the operating system are standard calls for reading and writing data files and for writing data to the screen. On the Mac, GMAT makes a call to the operating system to open X11, which is required to run MATLAB on the Mac.

User Interfaces

GMAT has several user interfaces. The interactive graphical user interface is introduced in more detail in later sections. The script interface is textual and also allows the user to configure and execute all aspects of GMAT. There also a secondary MATLAB interface that allows for running the system via calls from MATLAB to GMAT and allows GMAT to call MATLAB functions from within the GMAT command sequence. A low-level C API is also currently under development.

Status

While GMAT has undergone extensive testing and is mature software, at the present time we consider the software to be in beta form on Windows and alpha on Linux and Mac. GMAT is not yet sufficiently verified to be used as a primary operational analysis system. It has been used to optimize maneuvers for flight projects such as NASA's LCROSS and ARTEMIS missions, and the Lunar Reconnaissance Orbiter, and for optimization and analysis for the OSIRIS and MMS missions. However, for flight planning, we independently verify solutions generated in GMAT in the primary operational system.

The GMAT team is currently working on several activities including maintenance, bug fixes, and testing, along with selected new functionality.

Contributors

The Navigation and Mission Design Branch at NASA's Goddard Space Flight Center performs project management activities and is involved in most phases of the development process including requirements, algorithms, design, and testing. The Ground Software Systems Branch performs design, implementation, and integration testing. The Flight Software Branch contributes to design and implementation. GMAT contributors include volunteers and those paid for services they provide. We welcome new contributors to the project, either as users providing feedback about the features of the system, or as developers interested in contributing to the implementation of the system. Current and past contributors include:

- Thinking Systems, Inc. (system architecture and all aspects of development;
- Air Force Research Lab (all aspects of development)
- a.i. solutions (testing);
- Boeing (algorithms and testing);
- The Schafer Corporation (all aspects of development);
- Honeywell Technology Solutions (testing);
- Computer Sciences Corporation (requirements);

The NASA Jet Propulsion Laboratory (JPL) has provided funding for integration of the SPICE toolkit into GMAT. Additionally , the European Space Agency's (ESA) Advanced Concepts team has developed optimizer plug-ins for the Non-Linear Programming (NLP) solvers SNOPT (Sparse Nonlinear OPTimizer) and IPOPT (Interior Point OPTimizer).

Getting Started

Installation

Installers and files for all platforms are located on the GMAT SourceForge page at `https://sourceforge.net/projects/gmat`. GMAT releases are listed in chronological order with the most recent release at the top of the list. As of this writing the latest version is R2011a, released April 29, 2011.

Installing on Windows

The GMAT windows distribution contains an installer that will install and configure GMAT for you automatically. By default GMAT will be installed into your Application Data folder, and a shortcut will be placed in the Programs menu.

Installing on Mac

GMAT for Mac is released as a compressed archive. The archive can be uncompressed and installed by either double-clicking on it, or by running the following command in a Terminal window:

```
tar -zxf gmat-snowleopard-x86-R2011a-Alpha.tar.gz
```

Either method extracts the GMAT system into a folder named `GMAT-R2011a`; this is the GMAT root directory. You may move this folder in its entirety to the Applications folder or your desired installation directory.

Installing on Linux

GMAT for Linux is released as a compressed archive. The archive can be uncompressed by running the following command:

```
tar -zxf gmat-linux-x64-R2011a-Alpha.tar.gz
```

This command extracts the GMAT system into a folder named `GMAT-R2011a`; this is the GMAT root directory. Inside of that folder you will find two application launchers, `RunGmat.sh` and `RunGmatLauncher.sh`. `RunGmatLauncher.sh` assumes that GMAT is installed in the user's home directory; update that location if you installed GMAT to a different location. The Linux application is launched by running one of these two shell scripts. Each script file sets GMAT's load library path, then launches the application. If you would like to create a desktop shortcut to launch GMAT, set your launcher to run `RunGmatLauncher.sh`. You might want to set the icon for the launcher as well; GMAT's icon images are located in the `GMAT-R2011a/data/graphics/icons` directory.

Starting and Quitting GMAT

Starting a GMAT Session

On Microsoft Windows platforms there are several ways to start a GMAT session. If you used the GMAT installer, you can click the `GMAT R2011a` item in the Programs menu. If you installed

GMAT from a zip file, or by compiling the system, locate the bin folder in the GMAT root directory and double-click `GMAT.exe`.

On Mac, use the finder to open the `bin` folder located in your GMAT root directory and open the `GMAT` application. Alternatively, open a Terminal window, change to your installation directory, then type the command "**`open GMAT.app`**". Once GMAT is open, you can set it up to remain in the dock by clicking its dock icon, clicking Options, then Keep in Dock. This allows you to open GMAT in the future simply by clicking its dock icon.

Quitting a GMAT Session

To end a GMAT session on Windows or Linux, in the menu bar, click File, then click Exit. On Mac, in the menu bar, click **GMAT**, then click Quit GMAT, or type **Command**+**Q**.

Running the GMAT Demos

The GMAT distribution includes more than 30 sample missions. These samples show how to apply GMAT to problems ranging from the Hohmann transfer to libration point station-keeping to trajectory optimization. To locate and run a sample mission:

1. Open GMAT.
2. On the toolbar click Open.
3. Navigate to the `samples` folder located in the GMAT root directory.
4. Double-click a script file of your choice.
5. Click Run.

To run optimization missions, you need MATLAB®, and the MATLAB® Optimization Toolbox and/or the VF13ad plugin based on software in the Harwell Subroutine Library. These are proprietary libraries and are not distributed with GMAT. MATLAB® is not yet fully supported in the Mac and Linux GMAT releases, and therefore you cannot run optimization missions that use MATLAB's fmincon optimizer in the current Mac and Linux builds.

User Interfaces Overview

GMAT contains several user interfaces to design and execute your mission. The two primary interfaces are the graphical user interface (GUI) and the script interface. Each of these interfaces are interchangeable and support most functionality available in GMAT. When you work in the script interface, you are working in GMAT's custom script language. To avoid issues such as circular dependencies, there are some basic rules you must follow when writing scripts, or when working in the GUI. Below, we discuss these interfaces and then discuss the basic rules and best practices for working in each interface.

GUI Overview

When you start a GMAT session, the GMAT desktop is displayed and the default mission is loaded. The GMAT desktop has a native look and feel on each platform and most desktop components are supported on all platforms. The components of the desktop are discussed in detail in the Windows GUI section below and the differences for Mac and Linux platforms are discussed in separate sections.

Windows GUI

When you open GMAT on Windows and click Run in the Toolbar, GMAT executes the default mission as shown in the figure below. The tools listed below the figure are available in the GMAT desktop.

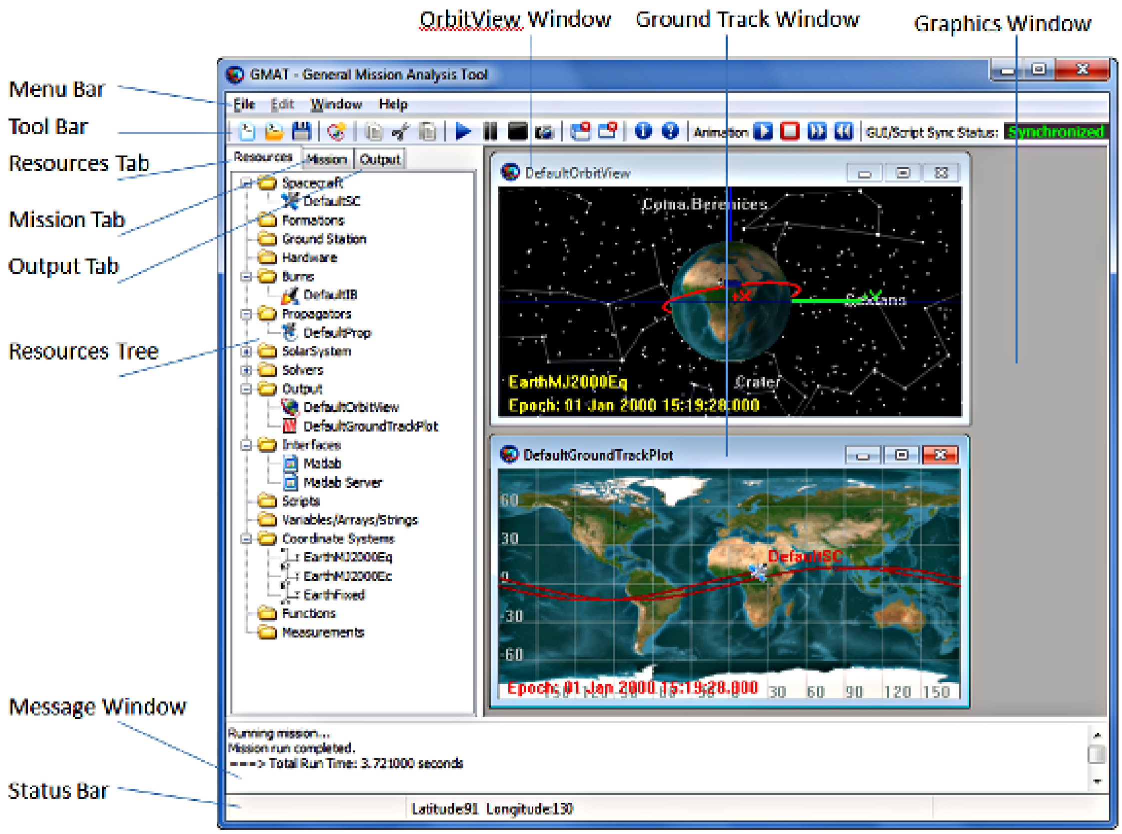


Figure 1. GMAT Desktop (Windows)

Menu Bar

The menu bar contains File, Edit, Window and Help functionality.

On Windows, the **File** menu contains standard **Open**, **Save**, **Save As**, and **Exit** functionality as well as **Open Recent** and **New Mission**. The **Edit** menu contains functionality for script editing and is only active when the script editor is active. The **Window** menu contains tools for organizing graphics windows and the script editor within the GMAT desktop. Examples include the ability to **Tile** windows, **Cascade** windows and **Close** windows. The Help menu contains links to **Online Help**, **Tutorials**, **Forums**, and the **Report An Issue** option links to GMAT's defect reporting system, the **Welcome Page**, and a **Provide Feedback** link.

On Mac, menus are nearly the same, with a few differences: the **File** menu does not contain an **Exit** option - instead, the **Quit GMAT** menu option is on the GMAT menu, as discussed before; tiling and cascading windows are not supported, so those options do not appear under the **Window** menu;

currently, email is not supported, so **Provide Feedback** is nonfunctional under the **Help** menu.

Toolbar
: The toolbar provides easy access to frequently used controls such as file controls, **Run**, **Pause**, and **Stop** for mission execution, and controls for graphics animation. On Windows and Linux, the toolbar is located at the top of the GMAT window; on Mac, it is located on the left of the GMAT frame. Because the toolbar is vertical, some toolbar options are abbreviated.

GMAT allows you to simultaneously edit the raw script file representation of your mission and the GUI representation of your mission. It is possible to make inconsistent changes in these mission representations. The **GUI/Script Sync Status** indicator located in the toolbar shows you the state of the two mission representations. See the the section called "GUI/Script Interactions and Synchronization" section for further discussion.

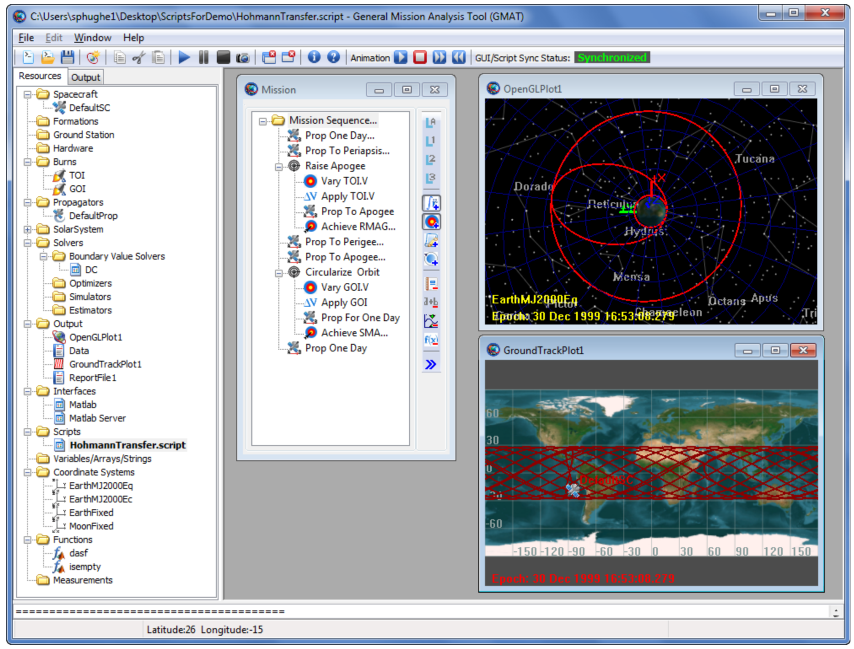
Resources Tab
: The **Resources** tab brings the Resources Tree to the foreground of the desktop.

Resources Tree
: The Resources Tree is a tree structure that displays all configured GMAT resources and organizes them into logical groups. All objects created in a GMAT script using a **Create** command are found in the **Resources Tree** in the GMAT desktop.

Mission Tab
: The **Mission** tab brings the Mission Tree to the foreground of the desktop.

Mission Tree
: The Mission Tree is a tree structure that displays GMAT commands that control the time-ordered sequence of events in a mission. The Mission Tree contains all script lines that occur after the `BeginMissionSequence` command in a GMAT script. You can undock the Mission Tree as shown in the figure below by right-clicking on the **Mission** tab and dragging it into the graphics window. You can also follow these steps:

1. Click on the **Mission** tab to bring the Mission Tree to the foreground.
2. Right-click on the **Mission Sequence** folder in the Mission Tree and select **Undock Mission Tree** in the menu.



Output Tab — The **Output** tab brings the Output Tree to the foreground of the desktop.

Output Tree — The **Output Tree** is a tree structure that contains GMAT output such as report files, event reports, and ephemeris files.

Message Window — When you run a mission in GMAT, information including warnings, errors, and progress are written to the message window. For example, If there is a syntax error in a script file, a detailed error message is written to the message window detailing the error.

Script Interface Overview

The GMAT script editor is a textual interface that supports most functionality in GMAT. In Figure 2 below, the script editor is shown maximized in the GMAT desktop and the items relevant to script editing are labeled.

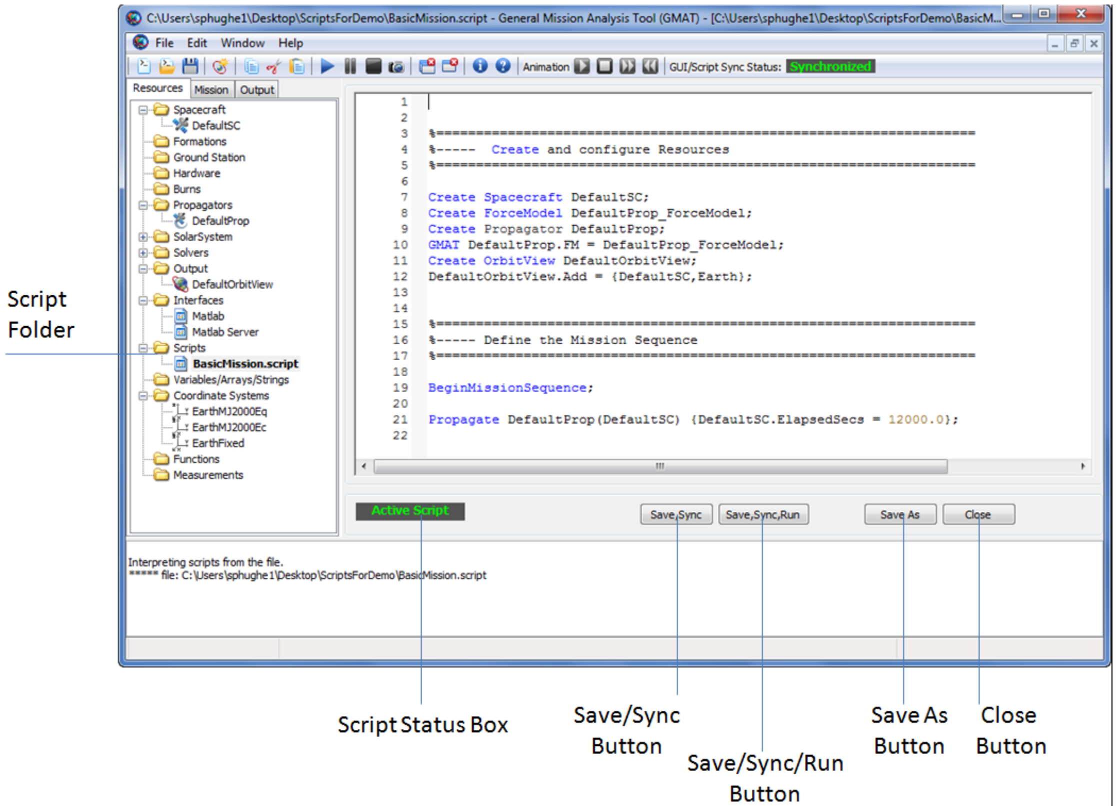


Figure 2. GMAT Script Editor

Script Folder	The GMAT desktop allows you to have multiple script files open simultaneously. Open script files are displayed in the **Scripts** Folder in the Resources Tree. Double click on a script in the **Scripts** folder to open it the script editor. The GMAT desktop displays each script that is open for view in a separate script editor. GMAT uses bold face font to identify which, if any, of the scripts in the **Scripts** folder are loaded into the GUI. Only one script can be loaded into the GUI at a time.
Script Status Box	The **Script Status** box indicates whether or not the script being edited is loaded in the GUI. The box says "Active Script" for the script currently loaded into the GUI and "Inactive Script" for all others.
Save,Sync Button	The **Save,Sync** button saves any script file changes to disk, makes the script active, and synchronizes the GUI with the script.
Save,Sync,Run Button	The **Save,Sync,Run** button saves any script file changes to disk, makes the script active, synchronizes the GUI with the script, and executes the script.
Save As Button	When you click **Save As**, GMAT displays the **Choose A File** dialog box and allows you to save the script using a new file name. After saving, GMAT loads the script into the GUI, making the new file the active script.
Close	The **Close** button closes the script editor.

GUI/Script Interface Interactions and Rules

The GMAT desktop supports a script interface and a GUI interface and these interfaces are designed to be consistent with each other. You can think of the script and GUI as different "views" of the same data: the resources and the Mission Command Sequence. GMAT allows you to switch between views (script and GUI) and have the same view open in an editable state simultaneously. Below we describe the behavior, interactions, and rules of the script and GUI interfaces so you can avoid confusion and potential loss of data.

GUI/Script Interactions and Synchronization

GMAT allows you to simultaneously edit both the script file representation and the GUI representation of your mission. It is possible to make inconsistent changes in these representations. The **GUI/ Script Sync Status** window located in the toolbar shows you the state of the two representations, as described in the following table. On Mac, the status is indicated in abbreviated form in the left-hand toolbar. **Synchronized** (green) indicates that the script and GUI contain the same information. **GUI Modified** (yellow) indicates that there are changes in the GUI that have not been saved to the script. **Script Modified** (yellow) indicates that there are changes in the script that have not been loaded into the GUI. **Unsynchronized** (red) indicates that there are changes in both the script and the GUI.

> **Caution**
>
> GMAT will NOT attempt to merge or resolve simultaneous changes in the Script and GUI and you must choose which representation to save if you have made changes in both interfaces.

The Save button in the toolbar saves the GUI representation over the script. The Save/Sync button on the script editor saves the script representation and loads it into the GUI.

How the GUI Maps to a Script

Clicking the **Save** button in the toolbar saves the GUI representation to the script file; this is the same file you edit when working in the script editor. GUI items that appear in the Resources Tree appear before the `BeginMissionSequence` command in a script file and are written in a predefined order. GUI items that appear in the Mission Tree appear after the `BeginMissionSequence` command in a script file in the same order as they appear in the GUI.

> **Caution**
>
> If you have a script file that has custom formatting such as spacing and data organization, you should work exclusively in the script. If you load your script into the GUI, then click **Save** in the toolbar, you will lose the formatting of your script. (You will NOT, however, lose the data.)

How the Script Maps to the GUI

Clicking the **Save/Sync** button on the script editor saves the script representation and loads it into the GUI. When you work in a GMAT script, you work in the raw file that GMAT reads and writes.

Each script file must contain a command called `BeginMissionSequence`. Script lines that appear before the `BeginMissionSequence` command create and configure models and this data will appear in the Resources Tree in the GUI. Script lines that appear after the `BeginMissionSequence` command define your mission sequence and appear in the Mission Tree in the GUI. Here is a brief script example to illustrate:

```
Create Spacecraft Sat
Sat.X = 3000
BeginMissionSequence
Sat.X = 1000
```

The line `Sat.X = 3000` sets the x-component of the Cartesian state to 3000; this value will appear on the **Orbit** tab of the **Spacecraft** dialog box. However, because the line `Sat.X = 1000` appears after the `BeginMissionSequence` command, the line `Sat.X = 1000` will appear as an assignment command in the Mission Tree in the GUI.

Basic Script Syntax Rules

- Each script file must contain one and only one `BeginMissionSequence` command.
- GMAT commands are not allowed before the `BeginMissionSequence` command.
- You cannot use inline math statements (equations) before the `BeginMissionSequence` command in a script file. (GMAT considers in-line math statements to be an assignment command. You cannot use equations in the Resources Tree, so you can also not use equations before the `BeginMissionSequence` command.)
- In the GUI, you can only use in-line math statements in an `Assignment` command. So, you cannot type **`3000 + 4000`** or **`Sat.Y - 8`** in the text box for setting a spacecraft's dry mass.
- GMAT's script language is case-sensitive.

Data and Configuration

Below we discuss the files and data distributed with GMAT and that are required for GMAT execution. GMAT uses many types of data files, including planetary ephemeris files, Earth orientation data, leap second files, and gravity coefficient files. This section describes how those files are organized and describe the controls provided so that you can customize the data files GMAT uses at run time.

File Structure

The default directory structure for GMAT is broken into eight main subdirectories, as shown in Figure 3. These directories organize the files and data used to run GMAT, including binary libraries, data files, texture maps, and 3D models. The only two files in the GMAT root directory are `license.txt`, which contains the text of the NASA Open Source Agreement, and `README.txt`, which contains user information for the current GMAT release. A summary of the contents of each subdirectory is described in further detail in the sections below.

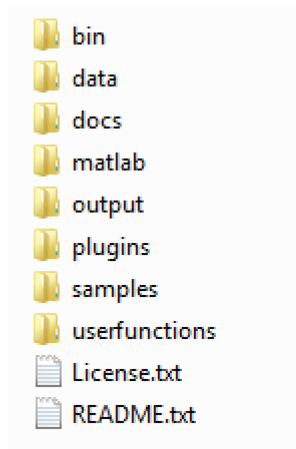


Figure 3. GMAT Root Directory Structure

`bin`

The `bin` directory contains all binary files required for the core functionality of GMAT. These libraries include the executable file (`GMAT.exe` on Windows, `GMAT.app` on Mac, and `GMAT` on Linux) and platform-specific support libraries. The `bin` directory also contains two text files: `gmat_startup_file.txt` and `gmat.ini`. The startup file is discussed in detail in a separate section below. The `gmat.ini` file is used to configure some GUI panels, set paths to external web links, and define GUI tooltip messages.

`data`

The `data` directory contains all required data files to run GMAT and is organized according to data type, as shown in Figure 4 and described below.

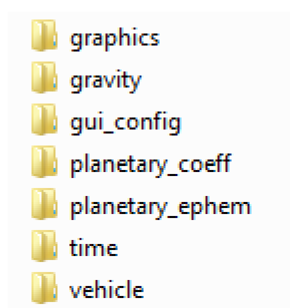


Figure 4. GMAT Data Directory Structure

The `graphics` subdirectory contains data files for GMAT's visualization utilities, as well as application icons and images. The `splash` directory contains the GMAT splash screen that is displayed briefly while GMAT is initializing. The `stars` directory contains a star catalogue used for displaying stars in 3D graphics. The texture folder contains texture maps used for the `OrbitView` 3D graphics resource. The `icons` directory contains graphics files for icons and images loaded at run time, such as the GMAT logo and toolbar icons.

The `gravity` subdirectory contains gravity coefficient files for each body with a default non-spherical gravity model. Within each subdirectory, the coefficient files are named according to the model they represent, and use the extension `.cof`.

The `gui_config` subdirectory contains files for configuring some of the GUI dialog boxes for GMAT resources and commands. These files allow you to easily create a GUI panel for a user-provided plugin, and are also used by some of the built-in GUI panels.

The `planetary_coeff` subdirectory contains the Earth orientation parameters (EOP) provided by the International Earth Rotation Service (IERS) and nutation coefficients for different nutation theories.

The `planetary_ephem` subdirectory contains planetary ephemeris data in both DE and SPK formats. The `de` directory contains the binary digital ephemeris DE405 files for the 8 planets, the Moon, and Pluto developed and distributed by JPL. The `spk` directory contains the DE421 SPICE kernel and kernels for selected comets, asteroids and moons. All ephemeris files distributed with GMAT are in the little-endian format.

The `time` subdirectory contains the JPL leap second kernel `naif0009.tls` and the GMAT leap second file `tai-utc.dat`.

The `vehicle` subdirectory contains ephemeris data and 3D models for selected spacecraft. The `ephem` directory contains SPK ephemeris files, including orbit, attitude, frame, and time kernels. The `models` directory contains 3D model files in 3DS or POV format for use by GMAT's `OrbitView` visualization resource.

docs

The `docs` directory contains end-user documentation, including PDF versions of the Mathematical Specification, Architectural Specification, and Estimation Specification. The GMAT User's Guide is available in the `help` subdirectory in PDF and HTML formats, and as a Windows HTML Help file.

matlab

The `matlab` directory contains M-files required for GMAT's MATLAB interfaces, including the interface to the fmincon optimizer and interfaces for driving GMAT from MATLAB. All files in the `matlab` directory and its subdirectories must be included in your MATLAB path for the MATLAB interfaces to function properly.

output

The `output` directory is the default location for file output such as ephemeris files and report files. If no path information is provided for reports or ephemeris files created during a GMAT session, then those files will be written to the output folder.

plugins

The `plugins` directory contains optional plugins that are not required for use of GMAT. The `proprietary` subdirectory is used for for third-party libraries that cannot be distributed freely and is an empty folder in the open source distribution.

samples

The `samples` directory contains over 30 sample missions, ranging from a Hohmann transfer to libration point station-keeping to Mars B-plane targeting. These files are intended to demonstrate GMAT's capabilities and to provide you with a potential starting point for building common mission types for your application and flight regime. Samples with specific requirements are located in subdirectories such as `NeedMatlab` and `NeedVF13ad`.

userfunctions

The `userfunctions` directory contains GMAT and MATLAB functions that are included in the GMAT distribution. You can also store your own custom GMAT and MATLAB functions in these folders.

Configuring GMAT Data Files

You can configure the data files GMAT loads at run time by editing the `gmat_startup_file.txt` file located in the `bin` directory. The startup file contains path information for data files such as ephemeris, earth orientation parameters and graphics files. By editing the startup file, you can customize which files are loaded and used during a GMAT session. Below we describe the customization features available in the startup file. The order of lines in the startup file does not matter.

Leap Second and EOP files

GMAT reads several files that are used for high fidelity modelling of time and coordinate systems: the leap second files and the Earth orientation parameters (EOP) provided by the IERS. The EOP file is updated daily by the IERS. To update your local file with the latest data, simply replace the file `eopc04_08.62-now` in the `data/planetary_coeff` directory. Updated versions of this file are available from the IERS [http://data.iers.org/products/213/14444/orig/eopc04_08.62-now].

There are two leap second files provided with GMAT in the `data/time` directory. The `naif0009.tls` file is used by the JPL SPICE libraries when computing ephemerides. When a new leap second is added, you can replace this file with the new file from NAIF [ftp://naif.jpl.nasa.gov/pub/naif/generic_kernels/lsk/]. GMAT reads the `tai-utc.dat` file for all time computations requiring leap seconds that are not performed by the SPICE utilities. You can modify this file if a new leap second is added by simply duplicating the last row and updating it with the correct information for the new leap second. For example, if a new leapsecond were added on 01 Jul 2013, then you would add the following line to the bottom of `tai-utc.dat` file:

```
2013 JUL 1 =JD 2456474.5 TAI-UTC= 35.0 S + (MJD - 41317.) X 0.0
```

Loading Custom Plugins

Custom plugins are loaded by adding a line to the startup file (`bin/gmat_startup_file.txt`) specifying the name and location of the plugin file. In order for a plugin to work with GMAT, the plugin library must be placed in the folder referenced in the startup file. You specify the path to a plugin file using the "PLUGIN" keyword and specify the file by providing its name without the file extension (.dll on Windows). For example, to load a Windows plugin named `libVF13Optimizer.dll` located in the `plugins/proprietary` directory, you would add this line to your startup file:

```
PLUGIN = ../plugins/proprietary/libVF13Optimizer
```

User-defined Function Paths

If you create custom GMAT or MATLAB functions, you can provide the path to those files and GMAT will locate them at run time. The default startup file is configured so you can place GMAT function files (with a `.gmf` extension) in the `userfunctions/gmat` directory and place MATLAB functions (with a `.m` extension) in the `userfunctions/matlab` directory. GMAT automatically searches those locations at run time. You can change the location of the search path to your GMAT or MATLAB functions by changing these lines in your startup file to reflect the location of your files with respect to the GMAT `bin` folder:

```
GMAT_FUNCTION_PATH = ../userfunctions/gmat
MATLAB_FUNCTION_PATH = ../userfunctions/matlab
```

If you wish to organize your custom functions in multiple folders, you can add multiple search paths to the startup file. For example,

```
GMAT_FUNCTION_PATH = ../MyFunctions/utils
GMAT_FUNCTION_PATH = ../MyFunctions/StateConversion
GMAT_FUNCTION_PATH = ../MyFunctions/TimeConversion
```

Configuring the MATLAB Interface

GMAT features a MATLAB interface that allows you to run MATLAB functions from within GMAT.

This interface is packaged as an optional GMAT plugin. To use it, make sure the following line is present in your `gmat_startup_file.txt` and has no comment symbol (`#`) in front of it.

```
PLUGIN = ../plugins/libMatlabInterface
```

The MATLAB interface must be able to find your MATLAB installation. The procedure for setting this information varies by platform.

Windows

On Windows, MATLAB must be properly configured in two places: the system `Path` variable and the Windows registry. Both locations must be configured for the same MATLAB version.

1. The following three directories must exist in your system's `Path` variable, where *`<MATLAB>`* is the path to the MATLAB root directory:

   ```
   <MATLAB>/bin/win32
   <MATLAB>/bin
   ```

   If you have multiple versions of MATLAB installed, GMAT will use the one that appears first in the system path.

   > **Caution**
   >
   > The above folders are added to your system path during MATLAB installation. However, for some versions of MATLAB (e.g. 2010a), MATLAB and Windows are dis-

tributed with libraries that have the same name. This may cause the Windows libraries to load instead of the MATLAB libraries. As a result, you may need to put the folders above at the beginning of your system path.

2. When you install MATLAB, it automatically registers itself as a COM server in the Windows registry. If you have mulple versions of MATLAB installed, it may be necessary to re-register a certain version manually. This can be done by running the following command. This may require administrator privileges.

```
matlab.exe -regserver
```

3. Add GMAT's MATLAB files to your MATLAB path. This can be done by placing the following line in a file named `startup.m` in your user MATLAB directory, where *`<GMAT>`* is the path to your GMAT root directory.

```
addpath(genpath('<GMAT>/matlab'));
```

Mac OS X

On Mac OS X, to use MATLAB with GMAT, you must set the `MATLABFORGMAT` environment variable in your `environment.plist` file, located in the `.MacOSX` directory in your home folder. This environment variable should point to the location of your MATLAB installation (application bundle). GMAT will not interface with MATLAB without this environment variable being set.

The current Mac application includes the ability to make calls to MATLAB functions from within GMAT, but does not support calls MATLAB to GMAT (including the `fmincon` optimizer).

Note that when GMAT opens MATLAB, it will open X11 first (as is required for MATLAB execution). GMAT currently does not automatically close X11 after quitting MATLAB, so you will need to quit X11 manually.

To add the environment variable:

1. If the `environment.plist` file already exists in your `.MacOSX` directory, edit the file using the Property List Editor to add the `MATLABFORGMAT` variable and set it to point to the location of your MATLAB application (e.g. `/Applications/MATLAB_R2010a/MATLAB_R2010a.app`).
2. If you do not have an `environment.plist` in your `.MacOSX` directory, using a terminal window:
    1. Create the `.MacOSX` directory as a subdirectory in your home folder (if it does not exist).
    2. Open the Property List Editor, create the `MATLABFORGMAT` variable as described above.
    3. Save the property list as `environment.plist` in your `.MacOSX` directory.

You must logout and log back in for this to take effect.

Other Resources

If you have further questions or need help for GMAT, or want to provide feedback, here are some additional resources:

- Official Homepage: `http://gmat.gsfc.nasa.gov`

- User Forum: `http://gmat.ed-pages.com/forum`
- Wiki: `http://gmat.ed-pages.com/wiki`
- Mailing Lists and Project Resources: `http://sourceforge.net/projects/gmat`
- Blog: `http://gmat.sourceforge.net/blog`
- Documentation: `http://gmat.sourceforge.net/docs`
- Bug Tracker: `http://pows003.gsfc.nasa.gov/bugzilla`
- Official Contact: `<gmat@gsfc.nasa.gov>`

Part II. Creating Your First Mission

Table of Contents

Simulating an Orbit

Objective and Overview

> **Note**
>
> The most fundamental capability of GMAT is to propagate spacecraft, which said another way is to simulate the orbital motion. The ability to propagate spacecraft is used in nearly every practical aspect of space mission analysis from simple orbital predictions--when will the International Space Station be over my house?--to complex analyses that determine the thirster firing sequence required to send a spacecraft to the Moon or Mars.

This tutorial will teach you how to use GMAT to propagate a spacecraft. You will learn how to configure a **Spacecraft** and a **Propagator**, and then set up a **Propagate** command to propagate the spacecraft to orbit perigee, which is the point of minimum distance between a spacecraft and Earth. The basic steps in this tutorial are:

1. Configure the**Spacecraft** and define its epoch and orbital elements.
2. Configure the **Propagator**.
3. Modify the default **OrbitView** to visualize the spacecraft trajectory.
4. Modify the **Propagate** command to propagate the spacecraft to the perigee.
5. Run the mission and analyze the results.

Configure the Spacecraft

In this section, you will rename a **Spacecraft** and set the **Spacecraft**'s initial epoch and classical orbital elements. You'll need GMAT open, with the Default Mission loaded. To load the Default Mission click **New Mission** in the **Toolbar** or start a new GMAT session.

Rename the Spacecraft

1. In the **Resources Tree**, right-click **DefaultSC**, and select **Rename**.
2. In the **Rename** dialog box, type **Sat** .
3. Click **OK**.

Set the Spacecraft Epoch

1. In the **Resources Tree**, double-click on **Sat**. Click the **Orbit** tab if it is not already selected.
2. In the **Epoch Format** box, select **UTCGregorian**. You'll see the value in the **Epoch** field change to the UTC Gregorian epoch format.
3. In in the **Epoch** field, type **22 Jul 2014 11:29:10.811**.
4. Click **Apply** to save these changes.

Set the Keplerian Orbital Elements

1. In the **StateType** box select **Keplerian** . In the **Elements** list, you will see the GUI reconfigure to display the Keplerian state representation.

2. In the **SMA** box, type **83474.318**.
3. Set the remaining orbital elements as shown in the table below.

Table 1. Sat Orbit State Settings

Field	Value
ECC	**0.89652**
INC	**12.4606**
RAAN	**292.8362**
AOP	**218.9805**
TA	**180**

4. Click **OK**.
5. In the **Toolbar**, Click the **Save** button. If this is the first time you have saved the mission, you'll be prompted to provide a name and location for the file.

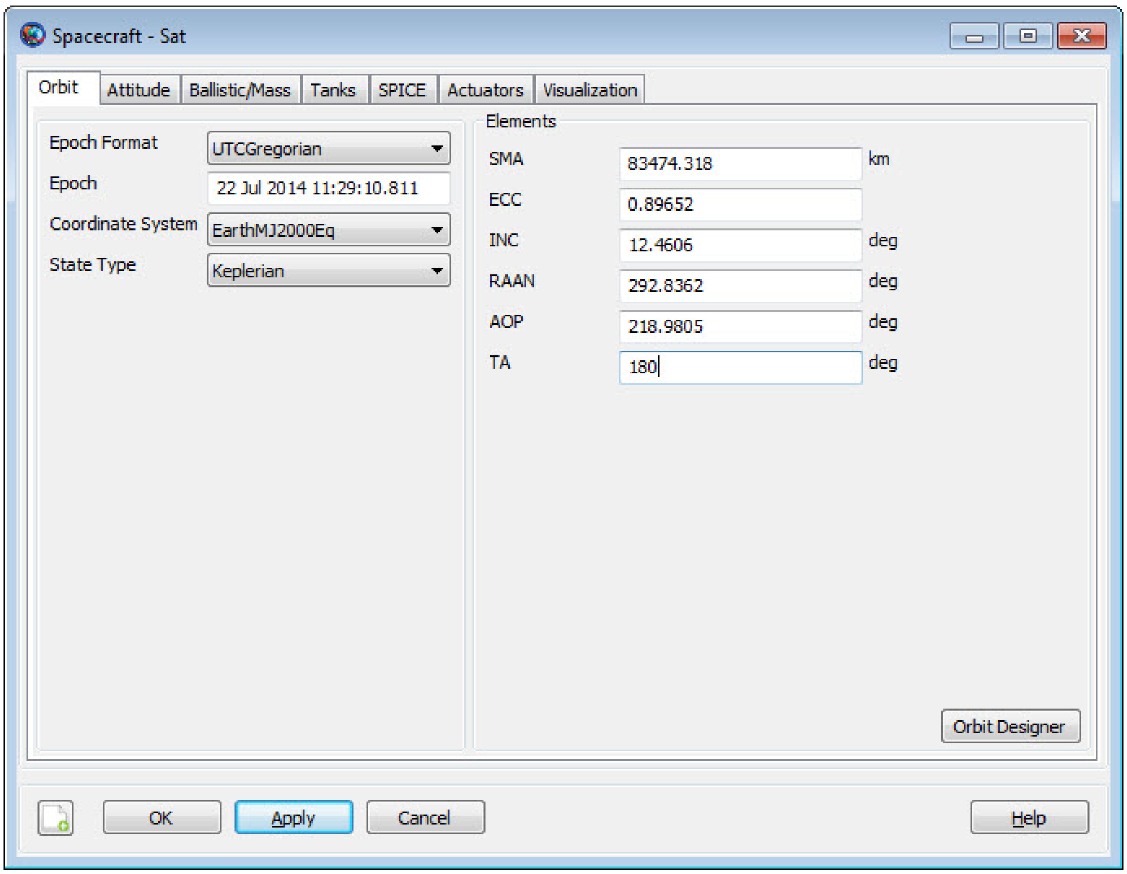


Figure 5. Spacecraft State Setup

Configure the Propagator

In this section you'll rename the **Propagator**, and configure the force model.

Rename the Propagator

Here you'll rename the propagator with a more descriptive name:

1. In the **Resources Tree**, double-click **DefaultProp**, and select **Rename**.
2. In the **Rename** dialog box, type **LowEarthProp**.
3. Click **OK**.

Configure the Force Model

Now configure the force model. For this tutorial you will use an Earth 10x10 spherical harmonic model, Jacchia-Roberts atmospheric model, solar radiation pressure, and point mass perturbations from the Sun and Moon.

1. In the **Resources Tree**, double-click **LowEarthProp**,
2. In the **Gravity** list, type **10** in the **Degree** box.
3. In the **Order** box, Type **10**.
4. In **Atmosphere Model** box, select **JacchiaRoberts**.
5. Click the **Select** button next to the **Point Masses** text box. This opens the **CelestialBodySelect** dialog box.
6. In the **Available Bodies** list, click **Sun**, then click **->** to add **Sun** to the **Selected Bodies** list.
7. Add the moon (named Luna in GMAT) using the same procedure above.
8. Click **Use Solar Radiation Pressure** to toggle it on.
9. Click **OK**.

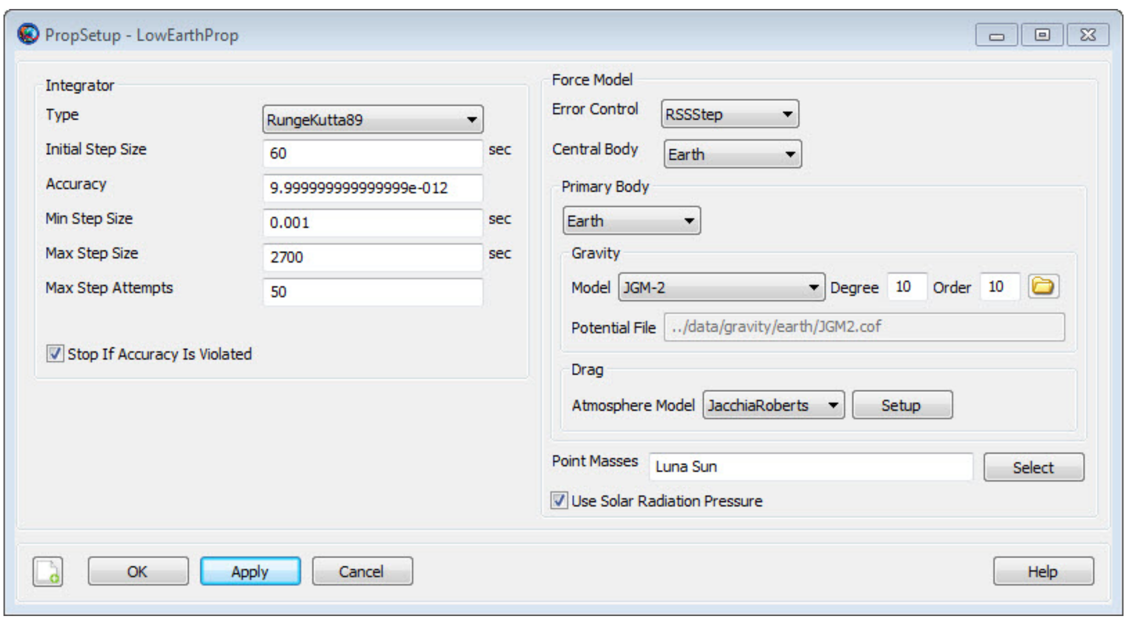


Figure 6. Force Model Configuration

Configuring the Orbit View Plot

Below you will configure an **OrbitView** plot so you can visualize **Sat** and its trajectory. The orbit of **Sat** is highly eccentric. To view the entire orbit in the plot, we need to adjust the settings of **DefaultOrbitView**.

1. In the **Resources Tree**, double-click on **DefaultOrbitView**.
2. In the three boxes next to **ViewPointVector**, type the values **-60000**, **30000**, and **20000** respectively.
3. In the **Drawing Options** list, un-check **DrawXY Plane**.
4. Click **OK**.

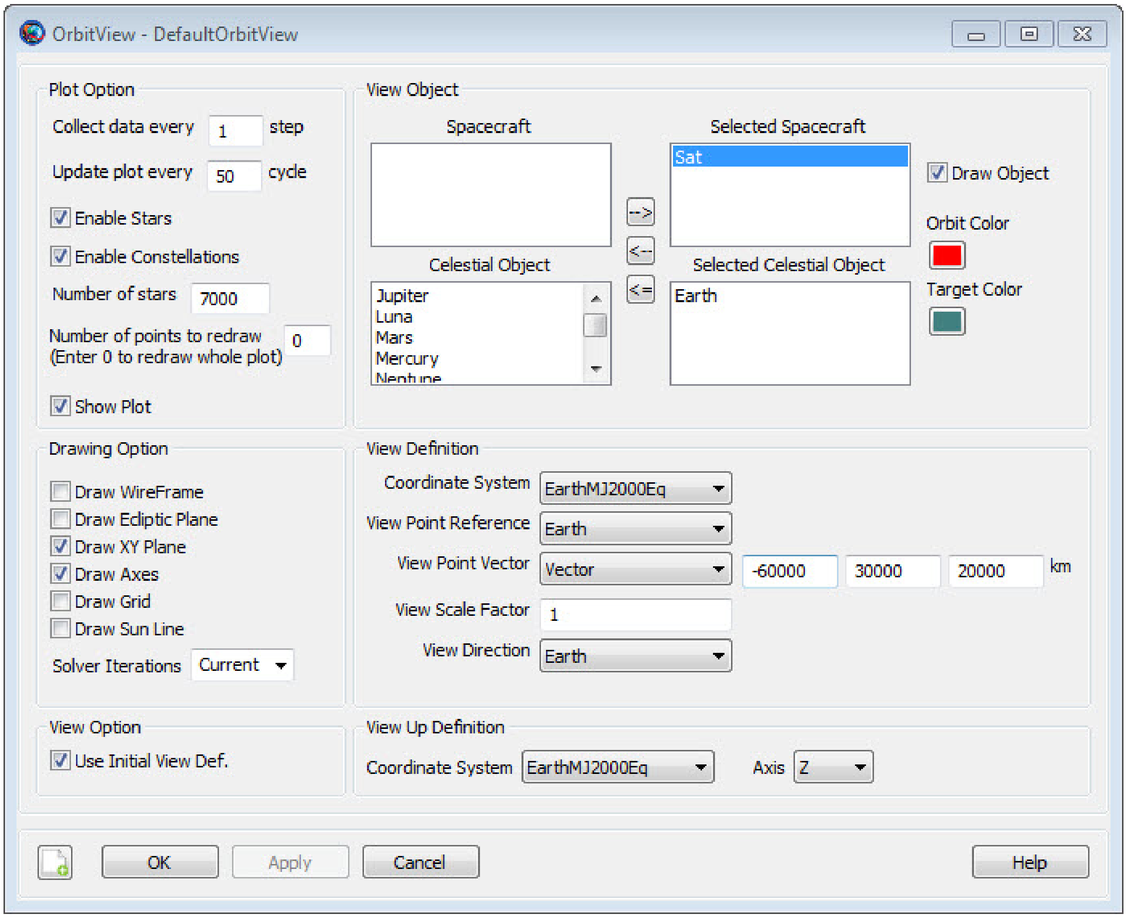


Figure 7. DefaultOrbitView Configuration

Configure the Propagate Command

This is the last step in the tutorial before running the mission. Below you will configure a Propagate command to propagate (model the motion of) Sat to orbit perigee.

1. Click the **Mission** tab to bring the **Mission Tree** to the foreground.
2. Double-click on **Propagate1**.
3. In the **Stopping Conditions** list, right click the (**...**) button next to Sat.ElapsedSecs to bring up the **Parameter Select Dialog**
4. In the **Object** list, select **Sat** if it is not already selected. This directs GMAT to associate the stopping condition with the spacecraft **Sat**.
5. In the **Object Properties** list, double-click **Periapsis** to add it to the **Selected Values** list as shown in Figure 8.

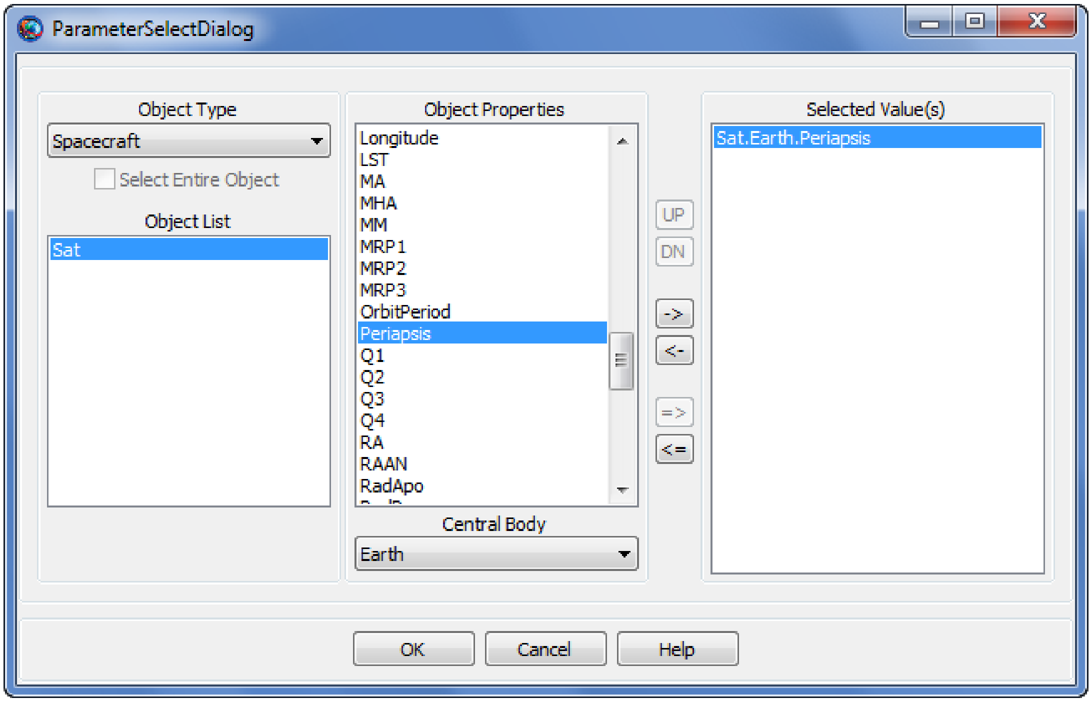


Figure 8. Propagate Command Parameter Select Dialog Configuration

6. Click **OK**.

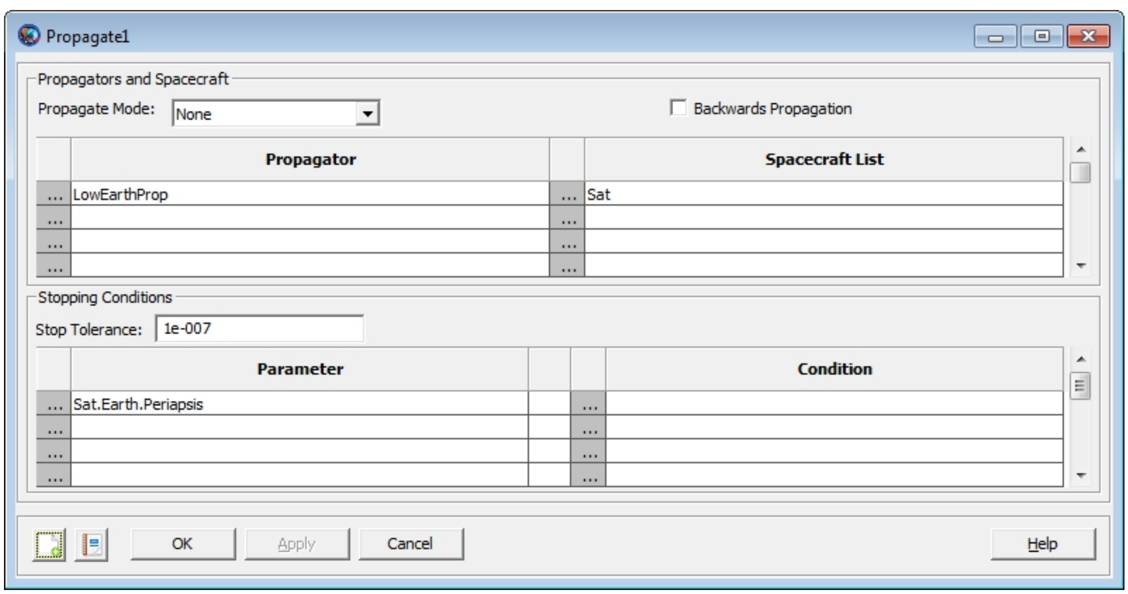


Figure 9. Propagate Command Configuration

Run and Analyze the Results

Congratulations, you have now configured your first GMAT mission and are ready to run the mission and analyze the results.

1. Click the **Save** button in the **Toolbar** to save your mission.
2. Click the **Run** button in the **Toolbar**.

You will see GMAT propagate the orbit and stop at orbit periapsis. Figure 10 below illustrates what you should see after correctly completing this tutorial. Here are a few things you can try to explore the results of this tutorial:

1. Manipulate the **Orbit View** plot using your mouse to orient the trajectory so that you can to verify that at the final location the spacecraft is at perigee.
2. Click the **Mission Tab** to bring the **Mission Tree** to the foreground.
3. Left-click on **Propagate1** and select **Command Summary** to see data on the final state of **Sat**.
4. What values for longitude and latitude do you see for **Sat**.
5. Are the values for the final longitude and latitude of **Sat** consistent with the **Ground Track** plot **Sat**?
6. Close the **Command Summary** dialog box.
7. Click in the **DefaultOrbitView** graphics window to bring the window to the foreground.
8. Click **Start Animation** in the **Toolbar** to animate the mission and watch the orbit propagate from apogee to perigee.

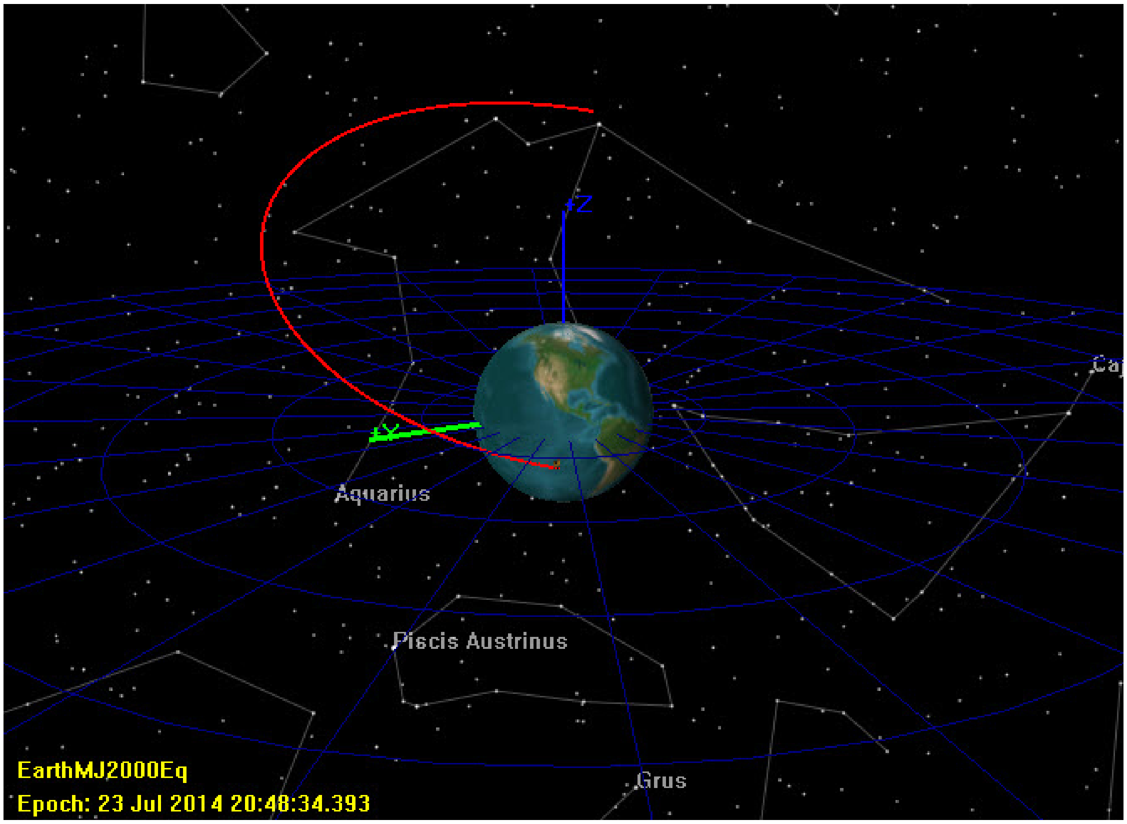


Figure 10. Orbit View Plot after Mission Run

Part III. How-tos

Table of Contents

Configuring a Spacecraft

Setting the Initial Epoch

You can configure the initial epoch of a Spacecraft using several time systems (TAI, TDB, UTC, etc) and in several formats (Gregorian Date, Modified Julian Date). In this How To you'll learn how to set a spacecraft's epoch in UTC Gregorian format. Starting from the default mission:

1. In the **Resources** tree, double-click on **DefaultSC** to open it.
2. Make sure the **Orbit** tab is selected
3. In the **EpochFormat** list, select **UTCGregorian**.
4. Type **`04 July 2014 09:30:15.235`** in the **Epoch** text box.

The script for the epoch settings configured above is shown below.

```
Create Spacecraft DefaultSC;
GMAT DefaultSC.DateFormat = UTCGregorian;
GMAT DefaultSC.Epoch = '04 Jul 2014 09:30:15.235';
```

Configuring the Orbit

To learn how to to define the initial state for a spacecraft orbit, you'll configure GMAT to propagate the International Space Station (ISS). Starting from the default mission, first set the initial epoch:

1. In the **Resources** tree, right-click on **DefaultSC**, and click **Rename**.
2. Type **ISS** in the **Rename** Dialog Box and then click **OK**.
3. On the **Resources** tree, double-click on **ISS**.
4. Make sure the **Orbit** tab is selected.
5. Click on the **Epoch Format** drop-down menu and select **UTCGregorian**.
6. Type **`21 Oct 2011 14:01:29.130`** in the **Epoch** text box.

Now follow the steps below to set the orbital state for ISS:

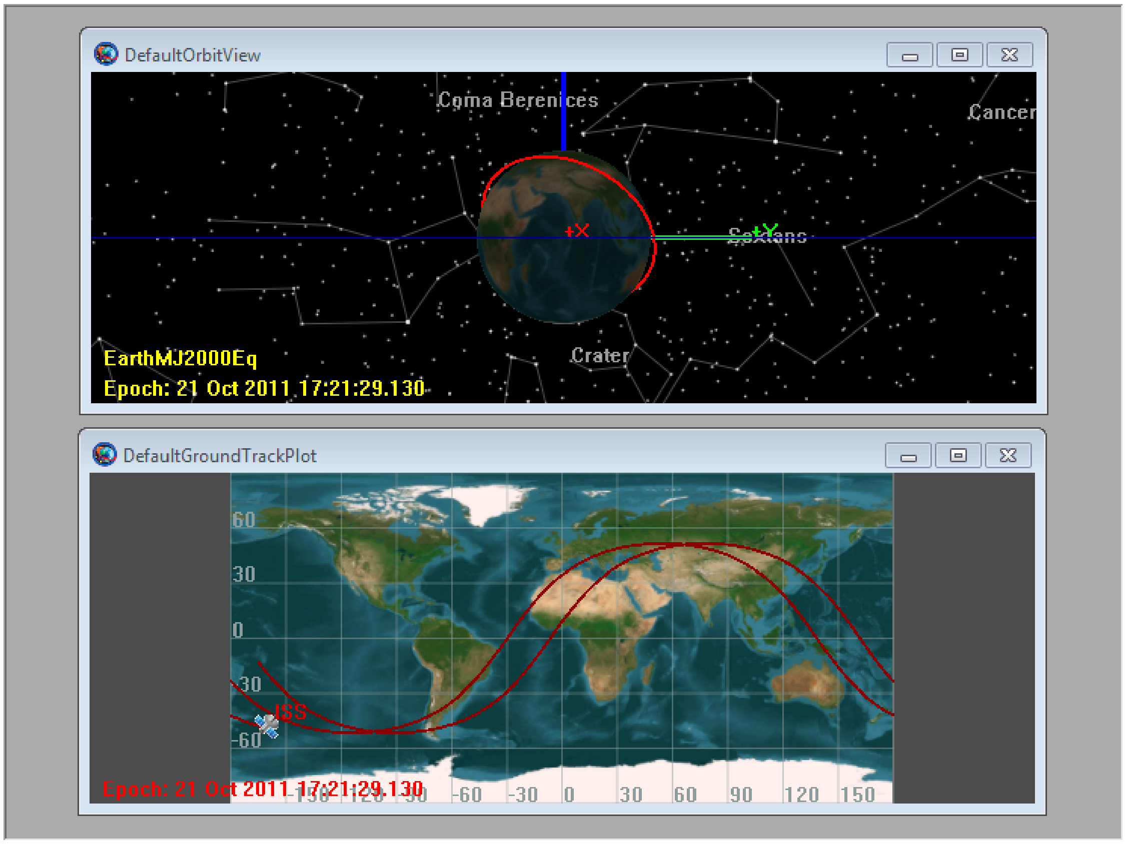
1. In the **Resources** tree, double-click on **DefaultSC** to open it.
2. Make sure the **Orbit** tab is selected.
3. Click the **State Type** drop-down menu and select **Keplerian**.
4. In the **Elements** parameter list on the **Spacecraft** dialog box, type **`6771.907`** in the **SMA** text box.
5. Type **`0.00103`** in the **ECC** text box.
6. Type **`51.597`** in the **INC** text box.
7. Type **`244.300`** in the **RAAN** text box.
8. Type **`353.735`** in the **AOP** text box.
9. Type **`199.683`** in the **TA** text box.
10. Click **OK**.

The script for the spacecraft state configured above is show below.

```
Create Spacecraft ISS
```

```
ISS.DateFormat=UTCGregorian
ISS.Epoch = 21 Oct 2011 14:01:29.130
ISS.SMA = 6771.907
ISS.ECC = 0.00103
ISS.INC = 51.597
ISS.RAAN = 244.300
ISS.AOP = 353.735
ISS.TA = 199.683
```

Click **Run** in the Toolbar and you will see plots like those shown below in the **Graphics Window**.



Configuring Physical Properties

To configure the physical properties of a spacecraft, start from the default mission and perform the following steps.

1. In the **Resources** tree, double-click on **DefaultSC** to open it.
2. Click on the **Ballistic/Mass** tab in the **Spacecraft** dialog box.
3. Type **`450`** in the **DryMass** text box.
4. Type **`2.0`** in the **Coefficient of Drag** text box.
5. Type **`1.7`** in the **Coefficient of Reflectivity** text box.
6. Type **`10.5`** in the **Drag Area** text box.
7. Type **`12.5`** in the **SRP Area** text box.

The script for the physical settings configured above is shown below.

```
Create Spacecraft DefaultSC
DefaultSC.DryMass = 450
DefaultSC.Cd = 2.0
DefaultSC.Cr = 1.7
DefaultSC.DragArea = 10.5
DefaultSC.SRPArea = 12.5
```

Propagating a Spacecraft

Configuring the Force Model

In the example below, you'll learn how to configure a high fidelity propagator for low Earth orbits. Starting from the default mission:

1. In the **Resources** tree, double-click on **DefaultProp** to open it.
2. In the **Gravity** section, type 21 in the **Degree** box.
3. Type 21 in the **Order** box.
4. Click on the **Atmosphere Model** drop-down list and select MSISE90 .
5. Click the **Select** button next to the **PointMasses** box.
6. Click on **Sun** and then click the right arrow **->** to add the **Sun** to your force model.
7. Add **Moon**, and **Jupiter** using the same steps as above.
8. Click **OK** on the **CelestialBodySelect** dialog box.
9. Click on the **UseSolarRadiationPressure** check box. The box should be checked now.
10. Click **OK** on the **PropSetup** dialog box.

The script for the force model configured above is shown below.

```
Create ForceModel DefaultProp_ForceModel;
DefaultProp_ForceModel.CentralBody = Earth;
DefaultProp_ForceModel.PrimaryBodies = {Earth};
DefaultProp_ForceModel.PointMasses = {Jupiter, Luna};
DefaultProp_ForceModel.Drag = MSISE90;
DefaultProp_ForceModel.SRP = On;
DefaultProp_ForceModel.ErrorControl = RSSStep;
DefaultProp_ForceModel.GravityField.Earth.Degree = 21;
DefaultProp_ForceModel.GravityField.Earth.Order = 21;
DefaultProp_ForceModel.GravityField.Earth.PotentialFile = 'JGM2.cof';
```

Propagating for a Duration

GMAT can propagate a spacecraft for a duration of time, such as 60 seconds, 30 days, or one orbit period. Starting from the default mission:

1. Click the **Mission** tab to show the **Mission** tree.
2. Double-click **Propagate1**. The default mission is configured to propagate the DefaultSC spacecraft for 12000 seconds.
3. In the **Parameter** column, to the left of **DefaultSC.ElapsedSecs**, click .... This will display a window allowing you to choose a new type of duration parameter.
4. In the **Object Properties** list, click **ElapsedDays**, then click -> to add it to the **Selected Value(s)** list.
5. Click OK.
6. In the **Condition** column, double-click the value **`0.0`** and enter **`30`** instead.
7. Click OK, then click Run.

GMAT will propagate the spacecraft for 30 days.

Propagating to an Orbit Condition

GMAT can propagate a spacecraft to a specific orbit condition, such as periapsis, an altitude value, or a latitude value. Starting from the default mission:

1. Click the **Mission** tab to show the **Mission** tree.
2. Double-click **Propagate1**. The default mission is configured to propagate the DefaultSC spacecraft for 12000 seconds.
3. In the **Parameter** column, to the left of **DefaultSC.ElapsedSecs**, click .... This will display a window allowing you to choose a new type of duration parameter.
4. In the **Object Properties** list, click **Periapsis**.
5. In the **Central Body** list, make sure **Earth** is selected. Then click -> to add it to the **Selected Value(s)** list.
6. Click OK to close the **ParameterSelectDialog** window.
7. Click OK, then click Run.

GMAT will propagate the spacecraft until it reaches periapsis.

Reporting Data

GMAT provides several ways to report mission data (such as altitude or delta-V values) to plain text files. GMAT can report data at each integration time step in the mission or at specific mission events, such as periapsis passage. The report functionality is controlled via the **ReportFile** resource and the **Report** and **Toggle** commands.

Reporting Data During a Propagation Span

You can report data at each integration step in the mission sequence by creating a **ReportFile** resource and adding data to it. Starting from the default mission:

1. On the **Resources** tree, right-click the **Output** folder, point to **Add**, and click **ReportFile**.
2. Double-click the **ReportFile1** resource.
3. In the **Parameter List** area, click **Edit**.
4. In the **Selected Value(s)** list, click **DefaultSC.EarthMJ2000.X** and click **<-** to remove it from the list.
5. In the **Object Properties** list, click **Altitude** and click **->** to add it to the **Selected Value(s)** list.
6. Add **DefaultSC.A1ModJulian** to the **Selected Value(s)** list if it doesn't already exist.
7. Click **OK**, then in the **ReportFile - ReportFile1** dialog box, click **OK** again.
8. Click **Run**. To view the generated report, on the **Output** tree, double-click **ReportFile1**.

The script for the report data configured above is shown below.

```
Create ReportFile ReportFile1;
GMAT ReportFile1.Add = {DefaultSC.A1ModJulian, DefaultSC.Earth.Altitude};
```

Reporting Data at a Specific Mission Event

You can report data to a **ReportFile** at a desired location in the mission sequence using the **Report** command. In this How To, you'll learn how to report spacecraft altitude at orbit apogee. Starting from the default mission:

1. In the **Resources** tree, right-click on the **Output** folder, point to **Add**, and click **ReportFile**.
2. In the **Output** folder, double-click on **ReportFile1** to open it.
3. In the **Parameter List** area, click the **Edit** button.
4. In the **Selected Values** list, click on **DefaultSC.EarthMJ2000Eq.X** and click the left arrow **<-** to remove it from the list.
5. Remove **DefaultSC.A1ModJulian** from the **Selected Value(s)** list using the step above.
6. Click **OK** to close the **ParameterSelectDialog** box and then click **OK** again to close the **ReportFile1** dialog box.

Now you will configure the **Propagate1** command to propagate to periapsis then issue a command to report to **ReportFile1**.

1. Click on the **Mission** tab to bring the mission tree to the foreground.
2. In the **Mission** tree, double-click on **Propagate1** to open it.
3. In the **Stopping Conditions** list, click on the ellipses **...** to the left of **DefaultSC.ElapsedSecs**.

4. In the **Object Properties** list, click on **Apoapsis** then click on the right arrow **->** to add it to the **Selected Value(s)** list.
5. Click **OK** to close the **ParameterSelectDialog** box and then click **OK** again to close the **Propagate1** dialog box.
6. Right-click on **Propagate1**, point to **Insert After**, and then select **Report**.
7. Double-click **Report1** and click on the **View** button.
8. Click on the remove all button **<=** to remove all items from the **Selected Value(s)** list.
9. Click on **TA** in the **Object Properties** list then click the right arrow **->** to add it to the **Selected Value(s)** list.
10. Add **Altitude** to the **Selected Value(s)** list using the same step as above.
11. Click **OK** to close the **ParameterSelectDialog** box and then click **OK** again to close the **Report1** dialog box.
12. In the Toolbar, click **Run**.
13. Click the **Output** tab.
14. In the **Reports** folder, click **ReportFile1** to you will see the requested data.

The script for the report data configured above is shown below.

```
Create ReportFile ReportFile1;
BeginMissionSequence;
Propagate DefaultProp(DefaultSC) {DefaultSC.Earth.Apoapsis};
Report ReportFile1 DefaultSC.Earth.TA DefaultSC.Earth.Altitude;
```

Creating a CCSDS Ephemeris File

The CCSDS Orbit Ephemeris Message (OEM) is a standardized text-based ephemeris format. In GMAT, you can easily create an OEM file with your desired interpolation order and data frequency. Starting from the default mission:

1. In the **Resources** tree, right-click the **Output** folder, point to Add, and click EphemerisFile. A new resource called **EphemerisFile1** appears in the tree.
2. Double-click **EphemerisFile1** to open it.
3. Make sure that in the **File Format** list, **CCSDS-OEM** is selected.
4. Click **Ok**.
5. Click **Run**. The OEM file will be written to a file named `EphemerisFile1.eph` in GMAT's output folder.

Creating an SPK Ephemeris File

An SPK ephemeris is a binary file format used by the SPICE Toolkit created by NAIF. GMAT can write spacecraft state information to this format using your desired interpolation order and data frequency. Starting from the default mission:

1. In the **Resources** tree, right-click the **Output** folder, point to Add, and click EphemerisFile. A new resource called **EphemerisFile1** appears in the tree.
2. Double-click **EphemerisFile1** to open it.
3. In the **File Format** list, click **SPK**.
4. In the **File Name** box, replace the default value with **`EphemerisFile1.bsp`**. An SPK ephemeris requires the `.bsp` extension.
5. Click **Ok**.

6. Click **Run**. The SPK file will be written to a file named `EphemerisFile1.bsp` in GMAT's output folder.

Visualizing Data

Manipulating the 3D Orbit View

GMAT's OrbitView resource offers a three-dimensional realistic view of your mission trajectory in any coordinate system or viewpoint you choose. The view itself can be manipulated using the mouse. Starting from the default mission:

1. Click Run. This will run the mission and will result in a **DefaultOrbitView** window on the GMAT desktop. The default view is centered at the Earth, in an Earth-centered inertial reference frame.
2. With the left mouse button, drag in the **DefaultOrbitView** window. This will rotate the view about the center of the active coordinate system (in this case, the center of the Earth).
3. With the right mouse button, drag left-to-right. This will zoom the view out from the center of the active coordinate system. Dragging right-to-left will zoom the view in.
4. With the wheel button (or middle button), drag up and down. This will rotate the view about an axis perpendicular to the screen.

Configuring the Ground Track Plot

GMAT's ground track plot can display one or more spacecraft on a two-dimensional map of a celestial body. You can choose which spacecraft are displayed, and which celestial body to use. Keeping the Earth as the central body, let's add a second spacecraft to the default plot. Starting with the default mission, first add a new spacecraft:

1. Right-click the **Spacecraft** folder and click Add Spacecraft to add **Spacecraft1**.
2. In the **Mission** tree, double-click **Propagate1**.
3. Under **Spacecraft List**, click ... to the left of **DefaultSC**.
4. In the **Available SpaceObject** list, click **Spacecraft1** and click -> to move it to the **SpaceObject Selected** list. Then click OK. This adds **Spacecraft1** to the **Spacecraft List** for **Propagate1**.
5. Click Apply, then click OK.

Then add the new spacecraft to the ground track plot:

1. In the **Resources** tree, in the **Output** folder, double-click **DefaultGroundTrackPlot**.
2. In the **Selected Objects** list, select **Spacecraft1**.
3. Click OK, then click Run.

After the run is complete, the **DefaultGroundTrackPlot** window will show the trajectory of the default spacecraft and **Spacecraft1** on a map of Earth.

Creating a 2D Plot

GMAT offers an XYPlot resource that allows you to visualize the relationship between multiple parameters (for example, orbit altitude and time). Starting from the default mission:

1. In the **Resources** tree, right-click the **Output** folder, point to Add, and click XYPlot.

2. Double-click the new **XYPlot1** resource. The default y-axis parameter is the Cartesian "X" position of the spacecraft.
3. Click Edit Y to change the y-axis parameter.
4. In the **Selected Value(s)** list, click **DefaultSC.EarthMJ2000Eq.X** and click <- to remove it from the list.
5. In the **Object Properties** list, click **Altitude**.
6. In the **Central Body** list, make sure **Earth** is selected, then click -> to add it to the **Selected Value(s)** list.
7. Click OK in the **ParameterSelectDialog** window, then click OK again in the **XYPlot - XYPlot1** window.
8. Click Run.

A new plot window will appear.

Part IV. Tutorials

Table of Contents

Simple Orbit Transfer

Objective and Overview

Final result: HohmannTransferDesign.script [scripts/HohmannTransferDesign.script]

> **Note**
>
> One of the most common problems in space mission design is to transfer from one circular orbit to another circular,coplanar orbit. Circular, coplanar transfers are used to raise low-Earth orbits that have degraded due to the effects of atmospheric drag. They are also used to transfer from a low-Earth orbit to a Geosynchronous orbit and to send spacecraft to Mars. There is a well known sequence of maneuvers, called the Hohmann transfer, that performs a circular, coplanar transfer using the least possible amount of fuel. A Hohmann transfer employs two maneuvers. The first maneuver raises the orbital apogee (or lowers orbital perigee) to the desired altitude and places the spacecraft in an elliptical transfer orbit. At the apogee (or perigee) of the elliptical transfer orbit, a second maneuver is applied to circularize the orbit

In this tutorial, you will use GMAT to perform a Hohmann transfer from a low-Earth parking orbit to a Geosynchronous mission orbit. This requires a **Target** sequence to determine the required maneuver magnitudes to achieve the desired final orbit conditions. In order to focus on the configuration of the **Target** sequence, you will make extensive use of the default configurations for spacecraft, propagators, and maneuvers. The **Target** sequence employs two velocity-direction maneuvers and two propagation sequences. The purpose of the first maneuver is to raise orbit apogee to 42165 km. The purpose of the second maneuver is to nearly circularize the orbit and yield a final eccentricity of 0.005. The basic steps of this tutorial are:

1. Create and configure a **Differential Corrector**.
2. Modify the **DefaultOrbitView** to visualize the trajectory.
3. Create two default **ImpulsiveBurns**.
4. Create a **Target** sequence to (1) raise apogee to GEO altitude and (2) circularize the orbit.
5. Run the mission and analyze the results.

Configure Maneuver, Differential Corrector, and Graphics

For this tutorial, you'll need GMAT open, with the Default Mission loaded. To load the Default Mission click **New Mission** in the **Toolbar** or start a new GMAT session. We will use the default configurations for a spacecraft (**DefaultSC**), a propagator (**DefaultProp**), and maneuvers. **DefaultSC** is configured to a near circular orbit and **DefaultProp** is configured to use Earth as the central body with a gravity model of degree and order 4. The default impulsive burn model uses the Velocity Normal Bi-normal (VNB) coordinate system. You may want to open the dialog boxes for these objects and inspect them more closely as we will leave the settings of those objects at their default values.

Create the Differential Corrector

You will need a **Differential Corrector** later in this tutorial so we'll create one now. You can leave the settings at their defaults.

1. In the **Resource Tree**, locate the **Solvers** folder and expand it if it is minimized.
2. Right-click the **Boundary Value Solvers** folder, point to **Add**, and then select **Differential Corrector**.

Modify the default Orbit View

We need to make minor modifications to DefaultOrbitView so that the entire final orbit will fit in the graphics window.

1. In the **Resource Tree**, double-click the DefaultOrbitView.
2. Set the values shown in the table below.

Table 2. DefaultOrbitView settings

Field	Value
Solver Iterations	**Current**
ViewUpDefintion	**X**
ViewPointVector boxes	**0**,**0**, and **120000** respectively.

3. Click **OK**.

Create the Maneuvers.

We need two **ImpulsiveBurn** objects for this tutorial. Below, you'll rename the default **ImpulsiveBurn** and create a new maneuver.

1. In the **Burns** folder in the **Resource Tree**, right-click on **DefaultIB**, select **Rename**.
2. In the **Rename** dialog box, type **TOI**, an abbreviation for Transfer Orbit Insertion.
3. Right-click on the **Burns** folder, point to **Add**, and select **ImpulsiveBurn**.
4. Rename the new **ImpulsiveBurn** object to **GOI,** an abbreviation for Geosynchronous Orbit Insertion.

Configure the Target Sequence

Now you will configure a **Target** sequence to solve for the maneuver values required to raise the orbit to Geosynchronous altitude and circularize the orbit. We'll begin by creating the **Target** sequence and then discuss the function of each command. Finally, we'll configure the commands for our problem. To allow us to focus on the Target sequence, we'll assume you have learned how to propagate an orbit to a desire condition by taking the Create Your First Mission Tutorial.

Configure the Initial Propagate Sequence

1. Click on the **Mission tab** to bring the **Mission Tree** to the foreground.
2. Configure **Propagate1** to propagate to Periapsis. The procedures are discussed in Creating Your First Mission. You can optionally leave **Propagate1** with default settings.

Create the Target Sequence

Now create the commands necessary to perform the **Target** sequence. Figure 11 illustrates the configuration of the **Mission Tree** after you have complete the steps in this section. We'll discuss the **Target** Sequence after it has been created.

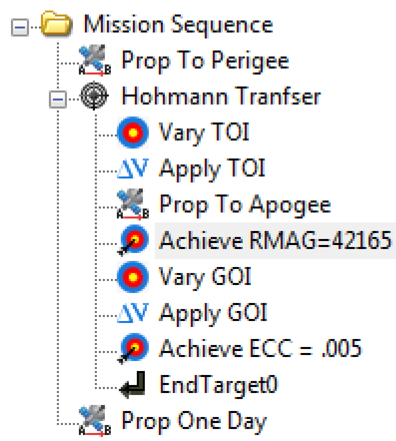


Figure 11. Target Sequence for the Hohmann Transfer

To create the **Target** Sequence:

1. In the **Mission Tree**, right-click on **Prop To Perigee**, point to **Insert After**, and select **Target**.
2. Right-click on **Target1** and select **Rename**.
3. In the **Rename** dialog box, type **Hohmann Transfer**, and click **OK**.
4. Right-click on **EndTarget0**, point to **Insert After**, and select **Propagate**.
5. Right-click on **Hohmann Transfer**, point to **Insert After**, and select **Vary**.
6. Rename **Vary1** to **Vary TOI**.
7. Complete the **Target** sequence by adding the commands inTable 3 after **Vary TOI** in the **Target** sequence.

Table 3. Additional Target Sequence Commands

Command	Name
Maneuver	**Apply TOI**
Propagate	**Prop to Apogee**
Achieve	**Achieve RMAG = 42165**
Vary	**Vary GOI**
Maneuver	**Apply GOI**
Achieve	**Achieve ECC = 0.005**

Note

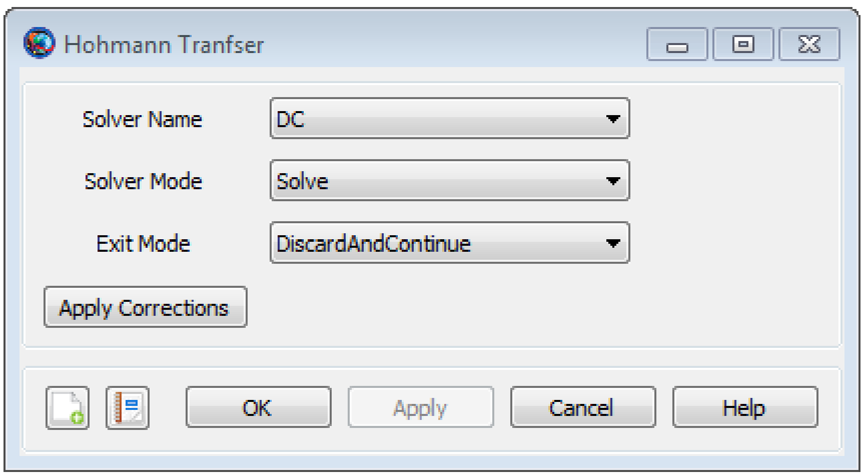
Let's discuss what the **Target** sequence does. We know that two maneuvers are required to perform the Hohmann transfer. We also know that for our current mission, the final orbit radius must be 42165 and the final orbital eccentricity must be 0.005. However, we do NOT know the size (delta v magnitudes) of the maneuvers that precisely achieve the desired orbital conditions. You use the **Target** sequence to solve for those precise maneuver values. But, you must tell GMAT what controls are available, (in this case two maneuver values) and what conditions must be satisfied (in this case orbital radius and eccentricity). You accomplish this using the **Vary** and **Achieve** commands. Using the **Vary** command, you tell GMAT what you would like it to solve for -- in this case the Delta V values for **TOI** and **GOI**. You use the **Achieve** command to tell GMAT what conditions the solution must satisfy-- in this case the final orbital conditions.

Configure the Target Sequence

Let's configure the target sequence.

Configure the Hohmann Transfer Command

1. Double-click **Hohmann Transfer** and change **ExitMode** to **SaveAndContinue**.
2. Change **ExitMode** to **SaveAndContinue.** This will save the solution of the targeting problem after you run it later in the tutorial.
3. Click **OK**.



Configure the Vary TOI Command

1. Double-click **Vary TOI**. Notice that the variable in the **Variable in the Setup** list is **TOI.Element1**. Element1 of **TOI** is the velocity component of TOI in the local VNB system. That's what we need, so we'll keep it.
2. In the **InitialValue** box, enter 1.0.
3. In the **MaxStep** box, enter 0.5.
4. Click **OK**.

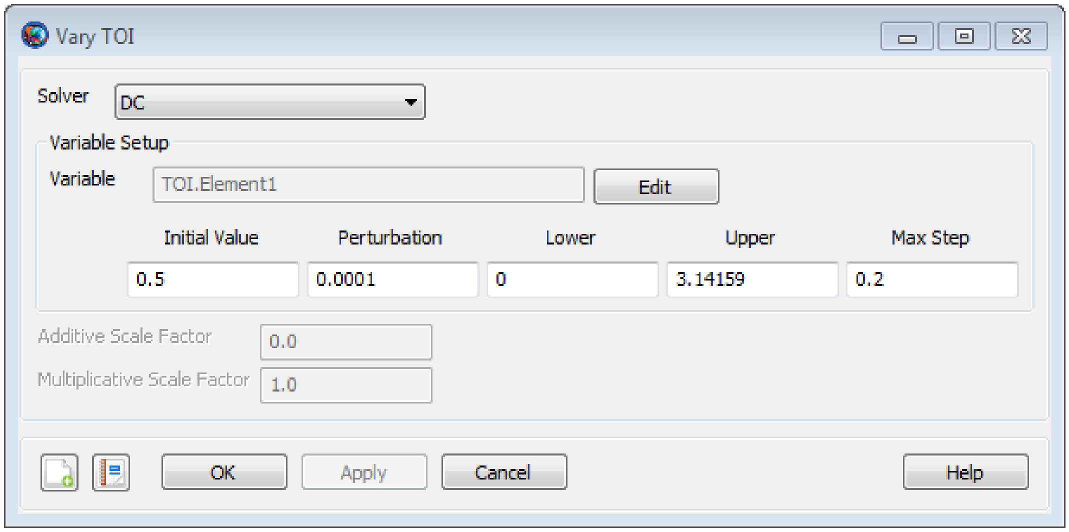
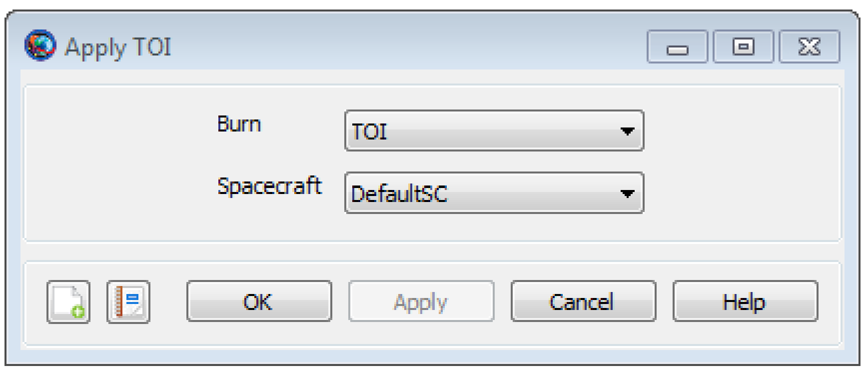


Figure 12. Vary TOI dialog box.

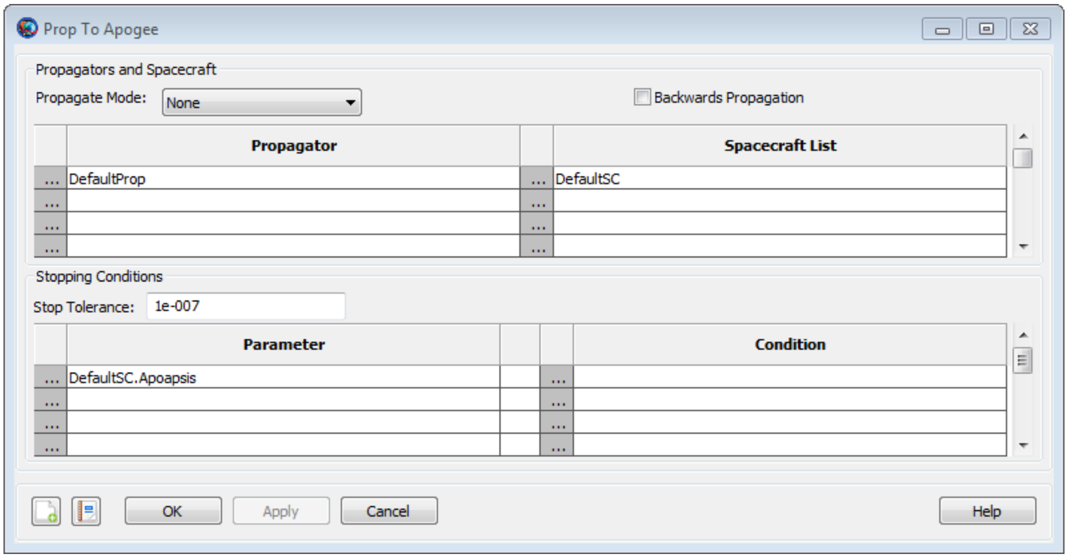
Configure the Apply TOI Command

1. Double-click **Apply TOI** . Notice that the command is already set to apply **TOI** to **DefaultSC**, so we don't need to change anything here.
2. Click OK.



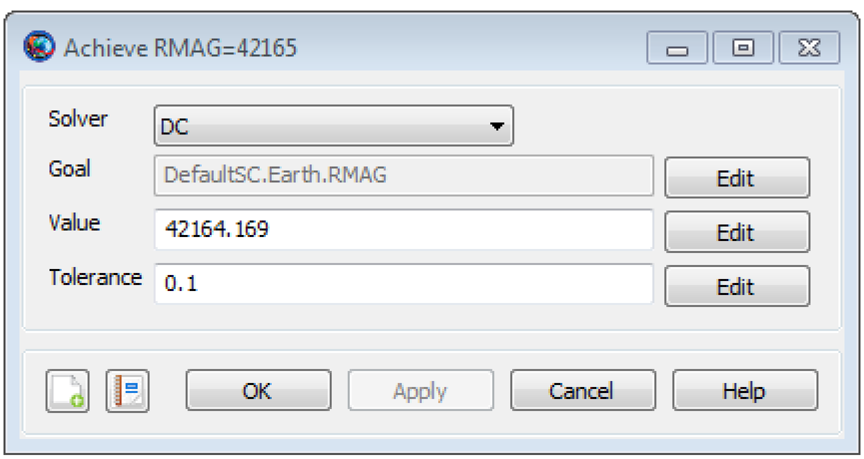
Configure the Prop to Apogee Command

1. Double-click **Prop to Apogee**.
2. In the **Parameter** list, click in the box that says **DefaultSC.ElapsedSecs**.
3. Type **DefaultSC.Apoapsis**.
4. Click **OK**.



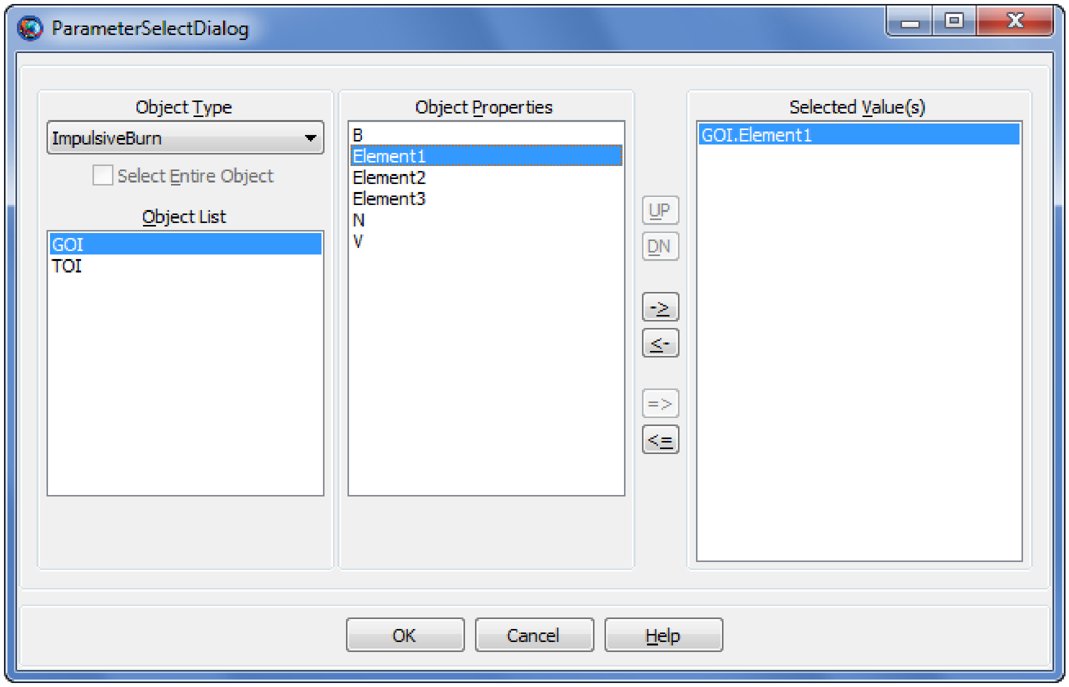
Configure the Achieve RMAG = 42165 Command

1. Double-click **Achieve RMAG = 42165**.
2. Notice that the goal is set to **DefaultSC.Earth.RMAG** - which is what we need, so we make no changes here.
3. In the Value box, enter **42164.169**, a more precise number for the radius of a Geostationary orbit.
4. **Click** OK.

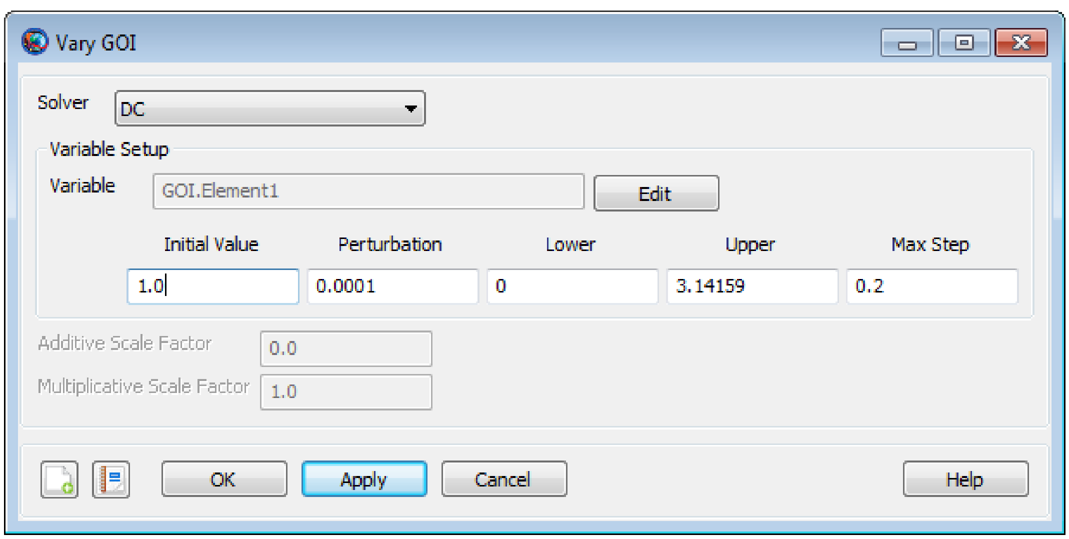


Configure the Vary GOI Command

1. Double-click **Vary GOI Command**.
2. Next to the **Variable** text box, click the **Edit** button.
3. Under **Object List**, click on **GOI**.
4. In the **Object Properties** list, double-click on **Element1**. (See the image below for results.)



5. Click **OK** to close the **Parameter Select** dialog box.
6. In the **Initial Value** box, type **1.0**.
7. In the **MaxStep** text box, type **0.2**.
8. **Click** OK.



Configure the Apply GOI Command

1. Double-click **Apply GOI**.
2. In the **Burn** box, select **GOI**
3. Click **OK**.

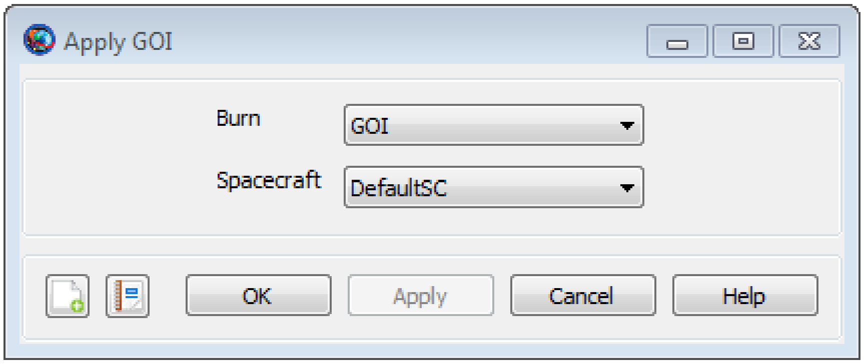


Figure: Maneuver2 Command

Configure the Achieve ECC = 0.005 Command

1. Double-click **Achieve ECC = 0.005**.
2. Next to the **Goal** box, click the **Edit** button.
3. In the **Object Properties** list, double-click **ECC** .
4. Click **OK** to close the **Parameter Select** dialog Box.
5. In the **Value** box, type **0.005**.
6. In the **Tolerance** box, type to **0.0001**.
7. Click OK.

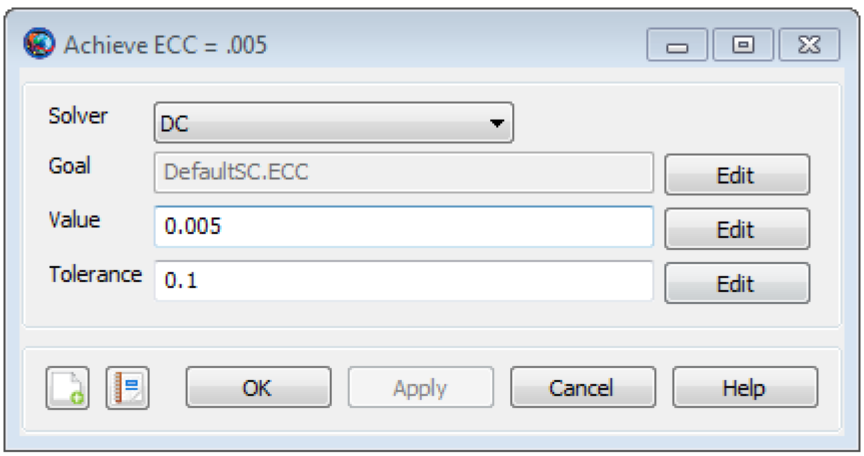
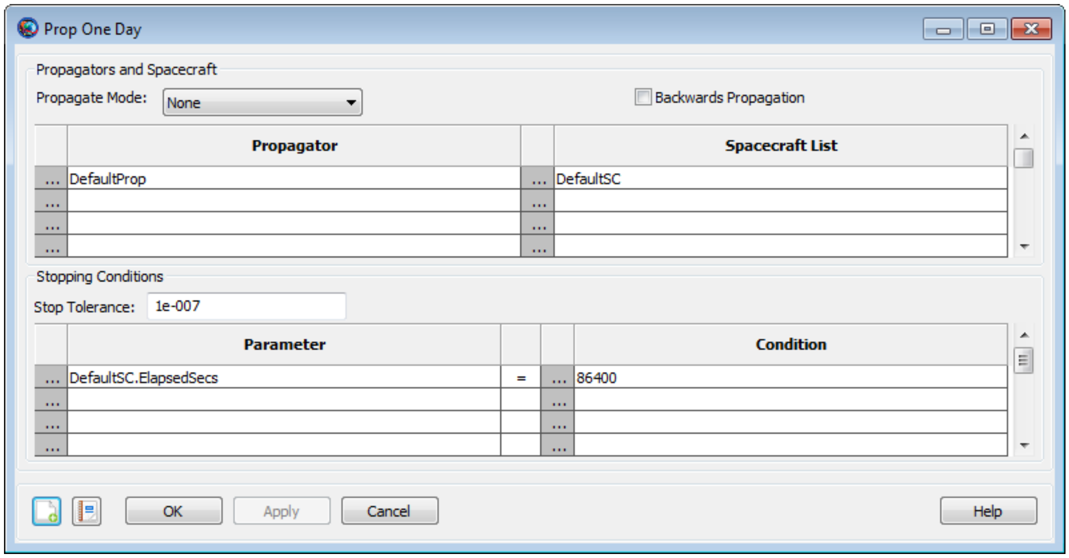


Figure: Achieve2 Command

Configure the Prop One Day Command

A Blank subsection.

1. Double-click **Prop One Day**.
2. In the **Condition** list, click in the box that says **12000**, and then type **86400** .
3. Click **OK**.



Running the Mission

Before running the mission, click **Save** in the **Toolbar** and save your file to the desired location. Now click on the **Run** in the **Toolbar**. As the mission runs, you will see GMAT solve the targeting problem and the iterations and perturbations are shown in light blue in the **DefaultOrbitView** window. After the mission is complete, the **OrbitView** should appear similar to the image shown below.

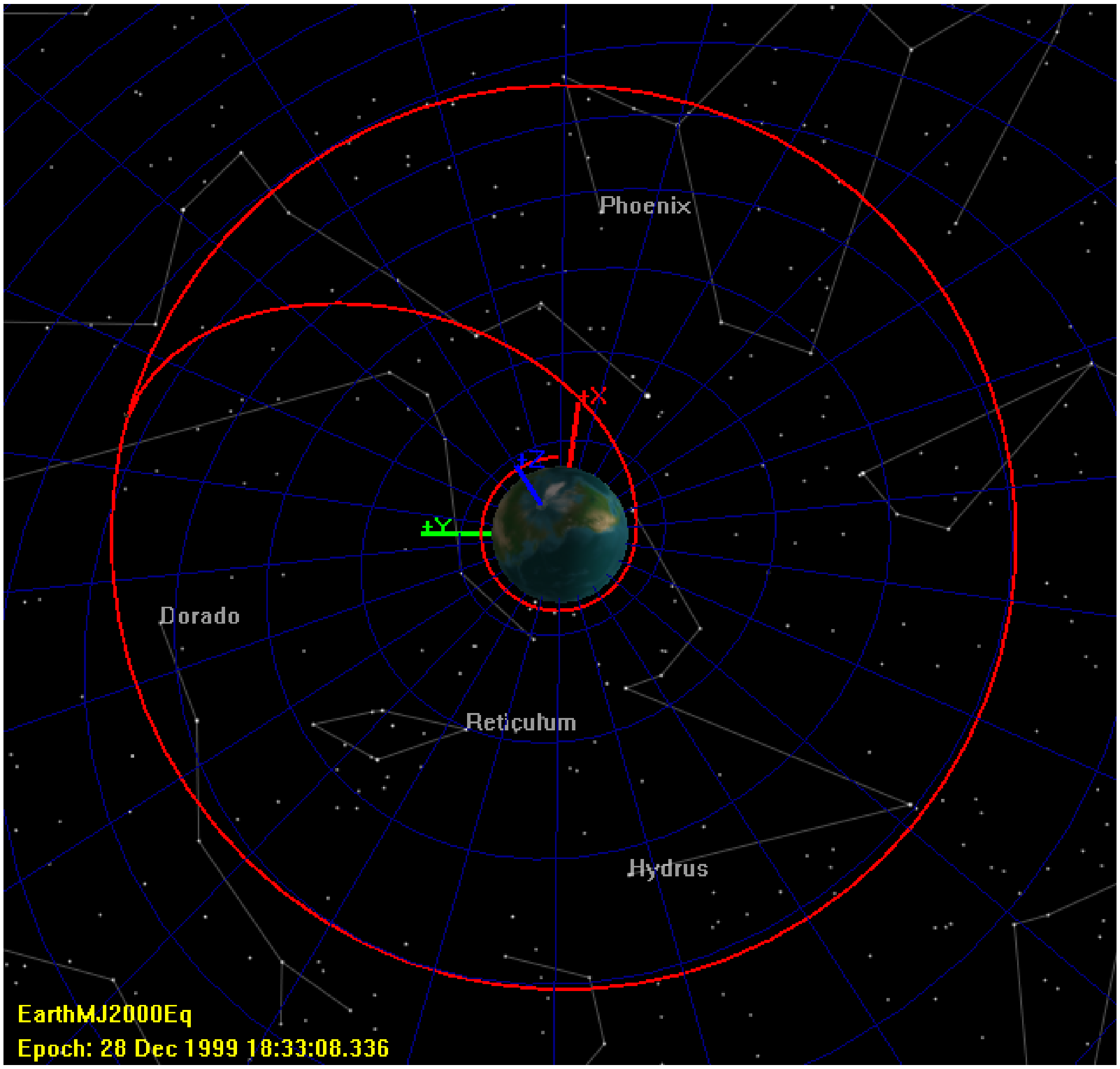


Figure: Output After Final Propagate Sequence

You can save the resulting solution so that if you make small changes to the problem and re-target, the initial guess for subsequent runs will use the solution from your work above.

1. Double-click on **Hohmann Transfer** in the **Mission Tree**.
2. Left-click on **Apply Corrections**.
3. Rerun the mission by clicking the **Run** button in the **Toolbar**. If you inspect the results in the message window, you will see that the **Target** sequence converges in one iteration because you gave provided the solution as the new initial guess.

Appendix A. NASA Open Source Agreement v1.3

```
NASA OPEN SOURCE SOFTWARE AGREEMENT
NASA OPEN SOURCE AGREEMENT VERSION 1.3

THIS OPEN SOURCE AGREEMENT ("AGREEMENT") DEFINES THE RIGHTS OF
 USE,
REPRODUCTION, DISTRIBUTION, MODIFICATION AND REDISTRIBUTION
 OF CERTAIN COMPUTER SOFTWARE ORIGINALLY RELEASED BY THE
 UNITED STATES GOVERNMENT AS REPRESENTED BY THE GOVERNMENT
 AGENCY LISTED BELOW ("GOVERNMENT AGENCY"). THE UNITED STATES
 GOVERNMENT, AS REPRESENTED BY GOVERNMENT AGENCY, IS AN INTENDED
 THIRD-PARTY BENEFICIARY OF ALL SUBSEQUENT DISTRIBUTIONS OR
 REDISTRIBUTIONS OF THE SUBJECT SOFTWARE. ANYONE WHO USES,
 REPRODUCES, DISTRIBUTES, MODIFIES OR REDISTRIBUTES THE SUBJECT
 SOFTWARE, AS DEFINED HEREIN, OR ANY PART THEREOF, IS, BY THAT ACTION,
 ACCEPTING IN FULL THE RESPONSIBILITIES AND OBLIGATIONS CONTAINED
 IN THIS AGREEMENT.
Government Agency: National Aeronautics and Space Administration
Government Agency Original Software Designation: GSC-16228-1
Government Agency Original Software Title: General Mission Analysis Tool (GMAT), version
 2011A
User Registration Requested. Please visit: http://opensource.gsfc.nasa.gov
Government Agency Point of Contact for Original Software: Dale Hithon, SRA Assistant, (301)
 286-2691 NOSA Version 1.3 Page 2 of 6
1. DEFINITIONS
A. "Contributor" means Government Agency, as the developer of the Original Software, and any
 entity that makes a Modification.
B. "Covered Patents" mean patent claims licensable by a Contributor that are necessarily infringed
 by the use or sale of its Modification alone or when combined with the Subject Software.
C. "Display" means the showing of a copy of the Subject Software, either directly or by means of
 an image, or any other device.
D. "Distribution" means conveyance or transfer of the Subject Software, regardless of means, to
 another.
E. "Larger Work" means computer software that combines Subject Software, or portions thereof,
 with software separate from the Subject Software that is not governed by the terms of this
 Agreement.
F. "Modification" means any alteration of, including addition to or deletion from, the substance
 or structure of either the Original Software or Subject Software, and includes derivative works, as
 that term is defined in the Copyright Statute, 17 USC 101. However, the act of including Subject
 Software as part of a Larger Work does not in and of itself constitute a Modification.
G. "Original Software" means the computer software first released under this Agreement
 by Government Agency with Government Agency designation GSC-16228-1 and entitled
 General Mission Analysis Tool (GMAT), including source code, object code and accompanying
 documentation, if any.
```

H. "Recipient" means anyone who acquires the Subject Software under this Agreement, including all Contributors.
I. "Redistribution" means Distribution of the Subject Software after a Modification has been made.
J. "Reproduction" means the making of a counterpart, image or copy of the Subject Software.
K. "Sale" means the exchange of the Subject Software for money or equivalent value.
L. "Subject Software" means the Original Software, Modifications, or any respective parts thereof.
M. "Use" means the application or employment of the Subject Software for any purpose. NOSA Version 1.3 Page 3 of 6
2. GRANT OF RIGHTS
A. Under Non-Patent Rights: Subject to the terms and conditions of this Agreement, each Contributor, with respect to its own contribution to the Subject Software, hereby grants to each Recipient a non-exclusive, world-wide, royalty-free license to engage in the following activities pertaining to the Subject Software:
1. Use
2. Distribution
3. Reproduction
4. Modification
5. Redistribution
6. Display
B. Under Patent Rights: Subject to the terms and conditions of this Agreement, each Contributor, with respect to its own contribution to the Subject Software, hereby grants to each Recipient under Covered Patents a non-exclusive, world-wide, royalty-free license to engage in the following activities pertaining to the Subject Software:
1. Use
2. Distribution
3. Reproduction
4. Sale
5. Offer for Sale
C. The rights granted under Paragraph B. also apply to the combination of a Contributor's Modification and the Subject Software if, at the time the Modification is added by the Contributor, the addition of such Modification causes the combination to be covered by the Covered Patents. It does not apply to any other combinations that include a Modification.
D. The rights granted in Paragraphs A. and B. allow the Recipient to sublicense those same rights. Such sublicense must be under the same terms and conditions of this Agreement. NOSA Version 1.3 Page 4 of 6
3. OBLIGATIONS OF RECIPIENT
A. Distribution or Redistribution of the Subject Software must be made under this Agreement except for additions covered under paragraph 3H.
1. Whenever a Recipient distributes or redistributes the Subject Software, a copy of this Agreement must be included with each copy of the Subject Software; and
2. If Recipient distributes or redistributes the Subject Software in any form other than source code, Recipient must also make the source code freely available, and must provide with each copy of the Subject Software information on how to obtain the source code in a reasonable manner on or through a medium customarily used for software exchange.
B. Each Recipient must ensure that the following copyright notice appears prominently in the Subject Software:
Copyright © 2002 - 2011 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. All Other Rights Reserved.

C. Each Contributor must characterize its alteration of the Subject Software as a Modification and must identify itself as the originator of its Modification in a manner that reasonably allows subsequent Recipients to identify the originator of the Modification. In fulfillment of these requirements, Contributor must include a file (e.g., a change log file) that describes the alterations made and the date of the alterations, identifies Contributor as originator of the alterations, and consents to characterization of the alterations as a Modification, for example, by including a statement that the Modification is derived, directly or indirectly, from Original Software provided by Government Agency. Once consent is granted, it may not thereafter be revoked.

D. A Contributor may add its own copyright notice to the Subject Software. Once a copyright notice has been added to the Subject Software, a Recipient may not remove it without the express permission of the Contributor who added the notice.

E. A Recipient may not make any representation in the Subject Software or in any promotional, advertising or other material that may be construed as an endorsement by Government Agency or by any prior Recipient of any product or service provided by Recipient, or that may seek to obtain commercial advantage by the fact of Government Agency's
or a prior Recipient's participation in this Agreement.

F. In an effort to track usage and maintain accurate records of the Subject Software, each Recipient, upon receipt of the Subject Software, is requested to register with Government Agency by visiting the following website: http://opensource.gsfc.nasa.gov. Recipient's name and personal information shall be used for statistical purposes only. Once a Recipient makes a Modification available, it is requested that the Recipient inform Government Agency at the web site provided above how to access the Modification. NOSA Version 1.3 Page 5 of 6

G. Each Contributor represents that its Modification is believed to be Contributor's original creation and does not violate any existing agreements, regulations, statutes or rules, and further that Contributor has sufficient rights to grant the rights conveyed by this Agreement.

H. A Recipient may choose to offer, and to charge a fee for, warranty, support, indemnity and/or liability obligations to one or more other Recipients of the Subject Software. A Recipient may do so, however, only on its own behalf and not on behalf of Government Agency or any other Recipient. Such a Recipient must make it absolutely clear that any such warranty, support, indemnity and/or liability obligation is offered by that Recipient alone. Further, such Recipient agrees to indemnify Government Agency and every other Recipient for any liability incurred by them as a result of warranty, support, indemnity and/or liability offered by such Recipient.

I. A Recipient may create a Larger Work by combining Subject Software with separate software not governed by the terms of this agreement and distribute the Larger Work as a single product. In such case, the Recipient must make sure Subject Software, or portions thereof, included in the Larger Work is subject to this Agreement.

J. Notwithstanding any provisions contained herein, Recipient is hereby put on notice that export of any goods or technical data from the United States may require some form of export license from the U.S. Government. Failure to obtain necessary export licenses may result in criminal liability under U.S. laws. Government Agency neither represents that a license shall not be required nor that, if required, it shall be issued. Nothing granted herein provides any such export license.

4. DISCLAIMER OF WARRANTIES AND LIABILITIES; WAIVER AND INDEMNIFICATION

A. No Warranty: THE SUBJECT SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THE SUBJECT SOFTWARE WILL CONFORM TO SPECIFICATIONS, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM

FROM INFRINGEMENT, ANY WARRANTY THAT THE SUBJECT SOFTWARE WILL BE ERROR FREE, OR ANY WARRANTY THAT DOCUMENTATION, IF PROVIDED, WILL CONFORM TO THE SUBJECT SOFTWARE. THIS AGREEMENT DOES NOT, IN ANY MANNER, CONSTITUTE AN ENDORSEMENT BY GOVERNMENT AGENCY OR ANY PRIOR RECIPIENT OF ANY RESULTS, RESULTING DESIGNS, HARDWARE, SOFTWARE PRODUCTS OR ANY OTHER APPLICATIONS RESULTING FROM USE OF THE SUBJECT SOFTWARE. FURTHER, GOVERNMENT AGENCY DISCLAIMS ALL WARRANTIES AND LIABILITIES REGARDING THIRD-PARTY SOFTWARE, IF PRESENT IN THE ORIGINAL SOFTWARE, AND DISTRIBUTES IT "AS IS."

B. Waiver and Indemnity: RECIPIENT AGREES TO WAIVE ANY AND ALL CLAIMS AGAINST THE UNITED STATES GOVERNMENT, ITS CONTRACTORS AND SUBCONTRACTORS, AS WELL AS ANY PRIOR RECIPIENT. IF RECIPIENT'S USE OF

NOSA Version 1.3 Page 6 of 6

THE SUBJECT SOFTWARE RESULTS IN ANY LIABILITIES, DEMANDS, DAMAGES, EXPENSES OR LOSSES ARISING FROM SUCH USE, INCLUDING ANY DAMAGES FROM PRODUCTS BASED ON, OR RESULTING FROM, RECIPIENT'S USE OF THE SUBJECT SOFTWARE, RECIPIENT SHALL INDEMNIFY AND HOLD HARMLESS THE UNITED STATES GOVERNMENT, ITS CONTRACTORS AND SUBCONTRACTORS, AS WELL AS ANY PRIOR RECIPIENT, TO THE EXTENT PERMITTED BY LAW. RECIPIENT'S SOLE REMEDY FOR ANY SUCH MATTER SHALL BE THE IMMEDIATE, UNILATERAL TERMINATION OF THIS AGREEMENT.

5. GENERAL TERMS

A. Termination: This Agreement and the rights granted hereunder will terminate automatically if a Recipient fails to comply with these terms and conditions, and fails to cure such noncompliance within thirty (30) days of becoming aware of such noncompliance. Upon termination, a Recipient agrees to immediately cease use and distribution of the Subject Software. All sublicenses to the Subject Software properly granted by the breaching Recipient shall survive any such termination of this Agreement.

B. Severability: If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement.

C. Applicable Law: This Agreement shall be subject to United States federal law only for all purposes, including, but not limited to, determining the validity of this Agreement, the meaning of its provisions and the rights, obligations and remedies of the parties.

D. Entire Understanding: This Agreement constitutes the entire understanding and agreement of the parties relating to release of the Subject Software and may not be superseded, modified or amended except by further written agreement duly executed by the parties.

E. Binding Authority: By accepting and using the Subject Software under this Agreement, a Recipient affirms its authority to bind the Recipient to all terms and conditions of this Agreement and that that Recipient hereby agrees to all terms and conditions herein.

F. Point of Contact: Any Recipient contact with Government Agency is to be directed to the designated representative as follows: Dale Hithon, SRA Assistant, (301) 286-2691