DRAFT
General Mission Analysis Tool (GMAT)
System Test Plan

Steven P. Hughes
Goddard Space Flight Center
Codes 583 and 595
Greenbelt, Maryland 20771

Darrel J. Conway
Thinking Systems, Inc.
6441 N Camino Libby
Tucson, Arizona 85718

March 16, 2010

Contents

List of Figures

List of Tables

Part I

Overview

Chapter 1

Introduction

1.1 Overview

The General Mission Analysis Tool (GMAT) is a spacecraft mission analysis tool tailored to support missions involving groups of spacecraft interacting throughout a modeled time period. The potential complexity of this problem makes GMAT an intricate software system. This complexity necessitates a rigorous testing environment to ensure that the system meets its objectives.

GMAT is designed using an object-oriented architecture[GDT] and coded using extensive object-oriented structures written in C++. The object based approach employed in GMAT's design and implementation makes the system robust and relatively easy to use for experienced analysts. The extent of the object model implemented to make GMAT a complete and robust system dictates a comprehensive testing philosophy, described in the GMAT Master Test Plan.[MTP] This document describes one component of the overall testing strategy, the system testing.

System testing is a black box form of testing, designed to exercise the GMAT system from the user's perspective. The system tests are designed to exercise all of the user accessible objects in GMAT.

1.2 Purpose of this Document

This document serves as the System Test Approach for the GMAT Project. Preparation for system testing consists of three major stages:

- The Test Approach sets the scope of system testing, the overall strategy to be adopted, the activities to be completed, the general resources required and the methods and processes to be used to test the release. It also details the activities, dependencies and effort required to conduct the System Test.
- Test Planning details the activities, dependencies and effort required to conduct the System Test.
- Test Cases documents the tests to be applied, the data to be processed, the automated testing coverage and the expected results.

This document covers the first two of these items, and established the framework used for the GMAT test case development. The test cases themselves exist as separate components, and are managed outside of and concurrently with this System Test Plan.

1.3 Overview of the GMAT Development and Testing Process

The GMAT development process identifies several review points for the system. GMAT development is conducted as a cooperative effort between an analysis team, typically composed of flight dynamics specialists, and a development team consisting of talented software developers. New requirements for the system are defined and written by the analysis team. Mathematical and design specifications are derived from these requirements and compiled into a format that can be used to code the new functionality. Requirements, Specifications, and Designs are reviewed by the development team prior to implementation. This review is typically conducted in an informal, iterative manner until the specifications are understood by all involved parties. The specifications and design documentation are then used to write the software.

During the development process, new features of a component under development may be detected that need further specification. When that happens, the new features are discussed and collected together. This may result in an immediate update to the design documents, or it may result in collection of the new feature implementation for inclusion in a final update performed when the component is ready for integration. In either case, the design documentation is updated to reflect the implemented functionality prior to formal acceptance of the related components.

During development, the software undergoes internal testing in the development team at both a unit and an integration level. Unit testing is intended to exercise all of the executable paths through the code, validating that the internal working of the code behaves correctly. Integration testing takes unit tested components and builds those components, either one at a time or collectively, into the system. From time to time, the development team will interact with the analysis team during integration testing to confirm that the observed behavior of the new code conforms to the expectations of the users. Unit testing and integration testing are performed in the course of the development of the software; neither will necessarily provide test results in a formal manner, though informal communications of the component and integrated test results are strongly encouraged.

When the GMAT development team completes integration of new functionality into the system, that new functionality is ready for system test. GMAT system testing follows a more formal test procedure than unit or integration testing. New components are exercised both from the GMAT scripting language and from the GMAT Graphical User Interface (GUI). The test cases exercised are documented using the procedures described later in this document. Test cases are managed using a traceability matrix that lists all of the elements of GMAT visible at the user level, and matches those elements to test cases that are executed in system testing. This master traceability matrix is used to generate a spreadsheet of test cases each time GMAT enters a system test cycle. All tests are tracked using this spreadsheet; formal system test is complete when every test case has been exercised and the results of the tests have been tabulated and accepted after review.

1.4 System Test Objectives

At a high level, System Test intends to prove that

- The functionality, delivered by the development team, is as specified by the Mathematical and Design Specifications[1].
- The software is stable and of high quality.
- The software models spacecraft missions faithfully.
- The software interfaces correctly with other systems, specifically MATLAB.
- The software user interfaces are stable, complete, and understandable by novice and experienced users.

[1] System test does not provide a formal mechanism for mapping the system requirements to the implemented functionality; that is the responsibility of Acceptance testing. The system test validates that the implemented functionality is correct.

These objectives are addressed through the development of a suite of test cases exercised on builds of the GMAT system. Each major release of GMAT is tested using this suite, and the results of the tests are collected and reviewd by all interested parties prior to release. This document describes the procedures followed for system testing.

1.5 Formal System Testing

While system tests can be performed as soon as new features are available, there is not a requirement that they must be performed at that time. However, system tests shall be performed prior to each major release of GMAT to the aerospace community. Part of the GMAT release process includes a review of the system test matrices and results to ensure that the system has maintained its integrity for the release. The review performed at each major release:

- Checks the System Test matrices to ensure full system coverage for User Elements, Parameters, Commands, and GUI Widgets.
- Ensures that the system tests have been run for all test cases.
- Ensures that the data produced from GMAT is consistent with known "truth" data.
- Ensures that system tests that failed have documented the cause or causes of the failure
- Ensures that any failures that must be addressed for the release have (1) been addressed and (2) that the resulting correction has been validated to meet the expected results.
- Ensures that all scripting elements of GMAT have been exercised, and function correctly.
- Ensures that all GUI elements of GMAT have been exercised, and function correctly.
- Ensures that the system is stable. Stability in this context means that GMAT
  - Does not crash
  - Produces identical results on rerun
  - Produces comparable results on all supported platforms
  - Allocates and releases memory consistently, without long term memory artifacts (aka "memory leaks")
  - Produces identical results when configured from the GUI, from a script file, and when saved to file and reloaded, both into the running instance and into a new image.
- Ensures that GMAT performs efficiently, both when executing mission sequences, and when saving and loading missions.

System test review is performed by members of the analysis and development teams. Detailed testing of the system numerics and scripting is performed by the domain experts on the analysis team. GUI testing is performed by the development team.

While the formal test responsibilities are as described in the previous paragraph, both teams are encouraged to exercise the features being tested by the other team to help identify any additional issues that exist. For example, the analysis team is encouraged to create all test cases using the GMAT GUI, and to report any difficulties encountered when following this approach. Similarly, the development team is encouraged to test the GUI in such a way as to produce functional models, to run those models, and to report any resulting anomalous behavior. This cross checking of functionality ensures that the system has been exercised as much as possible, given the resources available for development of GMAT.

1.6 Items Not Addressed in System Tests

The system tests described in this document are used to validate the stability and accessibility of GMAT components to users attempting to use the system to solve flight dynamics problems. These tests do not address several key system elements. Those elements are covered by other components of the GMAT test suite.

Specifically, the tests defined in this document do not address these items:

- Internal data representations and data flow in the GMAT code. These elements are tested in the GMAT unit and integration test processes.
- Numerical fidelity of the models. The detailed numerical testing of the components are part of the GMAT acceptance tests.
- Data range validation. The data range tests are performed as part of the integration tests.
- Requirements Validation. The mapping of GMAT capabilities to the system requirements is made and validated in the GMAT acceptance tests.

1.7 Document Layout

The remainder of this document describes the procedures followed to prepare for, conduct, and document the GMAT system tests. Chapter 2 describes the procedured followed when preparing for the system tests. Chapters 3 and 4 document the procedures followed when running the test cases. Chapter 5 describes the data collection and review procedures followed for the system. The Appendices at the end of the document provide additional information that may be useful during system test.

Part II

System Test Procedures

Chapter 2

System Test Preparation

The GMAT system tests are designed to perform a "black box" examination of GMAT as an assembled system. The system tests exercise all of the elements of the system from both the scripting and graphical user interface perspectives. Traceability matrices are maintained to ensure that the entire system is exercised upon completion of the system tests. This chapter describes these matrices, and provides instructions about how to maintain and extend them.

2.1 Test Process

Figure 2.1: The System Test Summary Page

System testing is performed in three stages: test preparation, system testing (consisting of Script based Testing and GUI Testing), and test result reporting. The test preparation phase, described in this chapter, is used to update the system test cases with tests covering new capabilities of GMAT, and to add or update existing test cases based on lessons learned from previous testing. Procedures followed when executing the script based are described in Chapter 3. GUI testing procedures are given in Chapter 4. Both of those chapters include descriptions of the data collection for individual tests. Chapter 5 describes the process of accumulating the test results so that the status of the system can be evaluated.

2.2 Test Preparation

GMAT system testing is managed from a set of OpenOffice^{OOo} spreadsheets. The test case structure and mapping between system functionality and corresponding tests is tracked using the "SystemTestMatrix.ods" spreadsheet[1]. This spreadsheet contains pages identifying detailed GMAT functionality and defined system test cases, and maps each element of functionality to one or more test cases.

The spreadsheet includes a summary page, shown in Figure 2.1, which computes coverage for the elements tabulated on the detail pages. If the tables in the spreadsheet are up to date, then the summary page is an indicator of the readiness of the system tests. Hence the first task that testers perform when preparing for system testing is to update the test matrices. Once the test matrices have been updated, the test cases are updated to cover any new functionality in the system. Test preparation is finished when a complete set of test cases has been developed, covering all of the elements in the updated test matrix tables.

Figure 2.2: An Object Test Matrix

To summarize, when a new piece of functionality is added to GMAT that users can access, the test team, working with the developers and users, updates the test matrices by performing three steps:

[1] All of the GMAT test tracking components are configuration controlled. Interested parties can obtain the current versions of these testing artifacts by contacting one of the GMAT team leads.

1. Identify and add all new elements of the system to the test matrices.
2. Identify test cases that cover the new elements. This may involve modifying existing test cases or creating new test cases, depending on the functionality of the new element.
3. Create or update the test cases as needed to implement the planned coverage identified in item 2.

When these steps have been performed, the coverage matrices are up to date, and the test team is ready to run the system test by executing all of the test cases in the matrices. The following paragraphs describe the procedure for executing these steps.

2.3 Updating the Element Lists in the Test Matrices

Figure 2.2 shows an example of the matrices used to identify GMAT's implemented functionality. Separate tables exist for the user accessible Components (Spacecraft, Solvers, Propagators, and so forth), Parameters that GMAT can calculate, Commands used when defining the mission sequence, Graphical User Interface elements (GuiElements), and miscellaneous other configurable elements. These tables capture a static view of every item that a user can interact with when running GMAT.

Each table lists the configurable elements in column A, and constructs, when appropriate, configurations and subconfigurations of those objects in columns B (labeled "Cases") and C ("Subcases"). Column D, "Notes", is used to indicate other considerations. Elements that are not yet scheduled for testing can be entered in the tables; when that happens, the entry in the "Notes" column should be set to the keyword "DEFERRED".

The first step in updating the test matrices is to ensure that the lists of accessible elements are complete, capturing any new elements and configurations added to the system since the last time the matrix was updated. Testers have two options for performing these updates: they can either edit the tables by hand, and check that all related formatting and equations are updated correctly, or they can use the macros built into the spreadsheet to add the new elements. The preferred approach is to use the macros, because that approach ensures that the calculations performed by the tables are correct.

Figure 2.3: The New Element Dialog

The summary page, shown in Figure 2.2, for the spreadsheet contains four buttons used to add elements to the test matrices: "Add Resource", "Add Parameter", "Add Command", and "Add GUI Element". When a user presses one of these buttons, a dialog box opens that is used to set some basic information for the new element that is being tested. Figure 2.3 shows an example of this dialog.

When this dialog is opened, users can change the type of new element being configured using the Element Type combo box. This option is provided in case the user selected the wrong button from the summary page. The user enters the name of the new element in the ElementName field.

Many of the elements that are tested can be exercised more than one way; for example, the Impulsive Burn element can be set to run using Velocity-Normal-Binormal (VNB) delta-V vectors or a coordinate system based

delta-V vector. Each of these modes should be tested independently, so a separate line should exist for each on the spreadsheet. The user reserves multiple lines on the spreadsheet by entering the number of lines required in the "Spreadsheet Lines Needed" field.

After setting the data correctly on the new element dialog, the user presses the 'OK" button. When this action is taken, the test matrix corresponding to the type of the new element is updated. New rows are inserted into the spreadsheet for the new element, and the formulas for the new rows are set. Finally, the fields that are used to calculate the test preparation statistics are updated. If more than one row was inserted, the spreadsheet page is set to the page containing the new element, with the active cell selected to the "Cases" field for the new element, so that the user can enter the test cases required for the new element. Each test case and subcase should be entered at this time so that the element descriptions in the test matrix reflect the capabilities that need to be tested.

At this point, all of the functionality in GMAT should be represented by rows in the test matrices. The next step is to plan test cases that cover elements of the system that are not already handled in the test suite.

Figure 2.4: A Test Case List

2.4 Updating the Test Case Lists

There are two categories of test cases used in system testing GMAT, designed to exercise the system using scripting and the graphical user interface. When new components are added to GMAT, the test coverage matrix is updated to exercise those new elements using the procedure described above. This update produces holes in the system test suite, requiring either an update of the current test cases or the development of new test cases, depending on the nature of the new components.

The test case lists are broken into two groups: tests based on script files designed to exercise all components used in modeling a mission, and user interface exercises designed to test the functionality and completeness of

the graphical user interface. The test tracking spreadsheet has separate pages for the GUI and script based test cases. Figure 2.4 shows the page for the script cases; the GUI test case page is similar.

When a test case is added to the test case list using the spreadsheet macros described below, that test case name is automatically picked up on the coverage tables. Once this update has been made and the new test cases have been added to the system test suite, users of the test matrix spreadsheet edit the matrices to indicate the covered functionality. In summary, the procedure for incorporating a new test case is to perform these three steps:

1. *Test case planning:* Identify and name the new test cases, and update the spreadsheet to list these cases.
2. *Test case writing:* Write the new test cases, and update any older test cases that need updating.
3. *Test Matrix Mapping:* Working from the new test cases, fill in the coverage tables for each new or changed test case to reflect the features actually covered.

Figure 2.5: The New Test Case Dialog

The procedure for adding a test case to the test case list is similar to the procedure for adding a new element to the test matrices. Test cases are added to the system test matrices using the "Add Script Testcase" and "Add GUI Testcase" buttons on the summary page of the spreadsheet. Pressing either of these buttons opens the New Test Case dialog, shown in Figure 2.5.

When a new test case has been identified, a user will open the system test spreadsheet and press the button for the desired test case type, opening this dialog. The user then enters the name of the new test case. The user enters a summary description of the test case as well to help track the goal of the test case. Finally, the user selects the desired frequency for execution of the test case; cases that can be automated and run frequently, or that test critical features of the system, should be set to run more frequently than those that are labor intensive or that test rarely used GMAT features.

The user accepts the new test case by selecting the "OK" button on the spreadsheet. When this action is taken, several things happen in the tables in the spreadsheet. First new test case is added to the appropriate page of the spreadsheet, along with its descriptions and execution frequency. The status of the test case is set to "Not started", indicating that the test case itself is not yet in the system test suite of test cases. The new test case is added to the column labels of the test matrices on the subsequent pages, and the formulae in in the spreadsheet are updated to track the new tests.

This step completes the test case planning phase of the preparation process. The next step is to write the test cases themselves.

2.5 Constructing the Test Cases

The steps described so far ensure that there is a plan in place to test every element of GMAT for a black box perspective. At this point, the test cases requires for the system test have been identified. Next the test team needs to write the test cases, given the new functionality of the system. The goal for each test case is to test an integrated set of system elements when executing a specified set of goals.

For the script based tests, this usually involves assembling a set of elements together and performing some computations in a mission sequence. The results of the execution of the script are compared to known good data in order to validate that the execution behaved as expected. Additionally, the script based testing checks to see that scripting errors are handled gracefully, producing error messages that are clear for typical GMAT users.

GUI based scripts have similar goals. The goals of the GUI test cases are to ensure that the GMAT user interface lets users configure all of the elements of the system, that this configuration is reflected in the internal components of the system, and that the user interface handles anomalous conditions robustly.

The following paragraphs describe the approach taken to ensure that these goals are met.

2.5.1 Updating Script Based Test Cases

Script based test cases consist of a script file and validated output files generated from the script. All script based tests should be created from the GMAT GUI, so that any related user interface issues can be identified during the process. Once a scripted test has been constructed, it should be saved with the same file name as entered in the test case table.

Each script based test should generate output in the form of a text file, using GMAT's reporting capabilities. Unless explicitly stated otherwise, the output file name should be the same as the script file name with the file extension ".report". The header comments on the script based tests should indicate the following information:

- The first line of the script should be "%% Id". This ensures that the CVS version information is stored with the script. This CVS information is the tracking identifier for each system test case.
- The primary elements being tested.
- Any ancillary items that should also be examined in the execution of the test.
- Any dependencies that need to be met to run the test successfully. For example, the FminconOptimizer requires a GMAT build that includes the MATLAB interfaces, a valid licensed MATLAB executable on the test machine, and a valid licensed copy of MATLAB's Optimization Toolbox.
- The name of the output files generated, is their name differs from the standard output file name.
- Whether the output data is expected to match data from previous runs.
- Any special steps that should be taken, either prior to the run or after it completes.

A sample script test case is provided here:

```
1 %% $Id: BasicProp.m,v 1.5 2006/10/11 16:37:00 dconway Exp $
2 %% GMAT System Test Script File
3 %
4 % This test case is designed to test the following elements:
5 %
6 % 1.  Spacecraft state specification in Earth MJ2000 Cartesian, Keplerian, and
```

```
%     Modified Keplerian Coordinates.
% 2.  Force models appropriate to LEO, HEO and GEO orbits.
% 3.  Basic orbit Propagation.
%
% The only output file is BasicPropHEOReport.txt, which contains various output
% parameters for the HEO spacecraft.  The data in this report should be the same
% from run to run.
%
% There are no external dependencies.
%
% This file has been edited to reduce size, so that it can be used as an example
% in the System Test Plan.

Create Spacecraft LEO;
GMAT LEO.DateFormat = TAIModJulian;
GMAT LEO.Epoch = 21545;
GMAT LEO.CoordinateSystem = EarthMJ2000Eq;
GMAT LEO.StateType = Cartesian;
GMAT LEO.X = 7100;
GMAT LEO.Y = 0;
GMAT LEO.Z = 1300;
GMAT LEO.VX = 0;
GMAT LEO.VY = 7.35;
GMAT LEO.VZ = 1;

Create Spacecraft HEO;
GMAT HEO.DateFormat = TAIGregorian;
GMAT HEO.Epoch = 12 Sep 2006 21:28:00.000;
GMAT HEO.CoordinateSystem = EarthMJ2000Eq;
GMAT HEO.StateType = Keplerian;
GMAT HEO.SMA = 43200;
GMAT HEO.ECC = 0.8;
GMAT HEO.INC = 78;
GMAT HEO.RAAN = 15;
GMAT HEO.AOP = 35;
GMAT HEO.TA = 120;

Create Spacecraft GEO;
GMAT GEO.DateFormat = UTCGregorian;
GMAT GEO.Epoch = 25 Dec 2010 00:00:00.000;
GMAT GEO.CoordinateSystem = EarthMJ2000Eq;
GMAT GEO.StateType = ModifiedKeplerian;
GMAT GEO.RadPer = 42164.5;
GMAT GEO.RadApo = 42165.5;
GMAT GEO.INC = 0.5;
GMAT GEO.RAAN = 90;
GMAT GEO.AOP = 90;
GMAT GEO.TA = 90;

Create ForceModel LeoProp_ForceModel;
GMAT LeoProp_ForceModel.CentralBody = Earth;
GMAT LeoProp_ForceModel.PrimaryBodies = {Earth};
GMAT LeoProp_ForceModel.Drag = Exponential;
GMAT LeoProp_ForceModel.Gravity.Earth.Degree = 20;
GMAT LeoProp_ForceModel.Gravity.Earth.Order = 20;
GMAT LeoProp_ForceModel.Gravity.Earth.PotentialFile = c:/GmatDataFiles/gravity/earth/JGM2.grv;
GMAT LeoProp_ForceModel.Drag.AtmosphereBody = Earth;
```

```
64
65  Create Propagator LeoProp;
66  GMAT LeoProp.FM = LeoProp_ForceModel;
67  GMAT LeoProp.Type = RungeKutta89;
68
69  Create ForceModel HeoProp_ForceModel;
70  GMAT HeoProp_ForceModel.CentralBody = Earth;
71  GMAT HeoProp_ForceModel.PrimaryBodies = {Earth};
72  GMAT HeoProp_ForceModel.Drag = MSISE90;
73  GMAT HeoProp_ForceModel.SRP = On;
74  GMAT HeoProp_ForceModel.Gravity.Earth.Degree = 4;
75  GMAT HeoProp_ForceModel.Gravity.Earth.Order = 4;
76  GMAT HeoProp_ForceModel.Gravity.Earth.PotentialFile = c:/GmatDataFiles/gravity/earth/JGM3.grv;
77  GMAT HeoProp_ForceModel.Drag.InputSource = Constant;
78
79  Create Propagator HeoProp;
80  GMAT HeoProp.FM = HeoProp_ForceModel;
81  GMAT HeoProp.Type = RungeKutta89;
82
83  Create ForceModel GeoProp_ForceModel;
84  GMAT GeoProp_ForceModel.CentralBody = Earth;
85  GMAT GeoProp_ForceModel.PrimaryBodies = {Earth};
86  GMAT GeoProp_ForceModel.PointMasses = {Sun, Luna, Jupiter, Venus};
87  GMAT GeoProp_ForceModel.SRP = On;
88  GMAT GeoProp_ForceModel.Gravity.Earth.Degree = 4;
89  GMAT GeoProp_ForceModel.Gravity.Earth.Order = 4;
90
91  Create Propagator GeoProp;
92  GMAT GeoProp.FM = GeoProp_ForceModel;
93  GMAT GeoProp.Type = PrinceDormand78;
94  Create ReportFile HeoReport;
95  GMAT HeoReport.Filename = BasicPropHEOReport.txt;
96  GMAT HeoReport.Precision = 16;
97  GMAT HeoReport.Add = {LEO.A1Gregorian, LEO.A1ModJulian, LEO.ElapsedSecs, ...
98     LEO.ElapsedDays, LEO.Earth.SMA, LEO.Earth.ECC, LEO.EarthMJ2000Eq.INC, ...
99     LEO.EarthMJ2000Eq.RAAN, LEO.EarthMJ2000Eq.AOP, LEO.Earth.TA};
100
101 %%---------------------------------------------------------------
102 %%----------  Mission Sequence
103 %%---------------------------------------------------------------
104 Propagate LeoProp(LEO, {LEO.ElapsedSecs = 8640.0});
105 Propagate HeoProp(HEO, {HEO.ElapsedSecs = 432000.0});
106 Propagate GeoProp(GEO, {GEO.ElapsedDays = 30.0});
```

If a script test case fails any of the system test criteria specified in Chapter 3, the tester creates a test report summarizing the nature of the failure. A sample completed report is shown here:

```
1  $Id: MatlabApsidesCheck.txt,v 1.3 2006/11/23 00:27:43 dconway Exp $
2
3
4  Tester: ___D. Conway_________________      Date: _11/21/06_____________
5
6
7  Platform:   _X_ Windows, Version: XP, Service Pack 2____
8
9              ___ Macintosh, OS X Version: _____________
```

```
10
11              ___ Linux, Distribution: ____________________
12
13
14  Description:
15
16  This test validates the MATLAB interface, including passing of arrays into
17  MATLAB and receipt of data back from MATLAB.
18
19
20  Script Test Results:
21
22  Loads Correctly:  [XX] Pass   [  ] Fail  Bug# ______
23
24  Runs Correctly:   [XX] Pass   [  ] Fail  Bug# ______
25                    [  ] Unable to evaluate
26
27  3D Visualization: [  ] Pass   [  ] Fail  Bug# ______
28                    [XX] Not Applicable
29                    [  ] Unable to evaluate
30
31  Plots:            [  ] Pass   [  ] Fail  Bug# ______
32                    [XX] Not Applicable
33                    [  ] Unable to evaluate
34
35  Output:           [XX] Pass   [  ] Fail  Bug# ______
36                    [  ] Not Applicable
37                    [  ] Unable to evaluate
38
39  Truth Data:       [  ] Pass   [XX] Fail  Bug# _511__
40                    [  ] Not Applicable
41                    [  ] Unable to evaluate
42
43  Rerun:            [XX] Pass   [  ] Fail  Bug# ______
44                    [  ] Not Applicable
45                    [  ] Unable to evaluate
46
47  Save and Load:    [  ] Pass   [XX] Fail  Bug# _512__
48                    [  ] Unable to evaluate
49
50  Summary:
51
52     Number of passed test elements      __4___
53
54     Total number of test elements       __6___
55
56     Test case status                          [ ] Pass  [X] Fail
57
58
59  Bugs Reported:
60
61     511, 512
62
63  Notes:
64
65  1. Truth data file shows a defect in data handling when receiving data from
66  MATLAB.  The MATLAB return only has 6 digits of precision.  A bug needs to be
```

```
entered into Bugzilla for this defect.

2. Save fails when there are multiple conditions on an If command.
```

2.5.2 Updating the GUI Test Cases

GUI based test cases consist of a text file describing the test. The GUI test cases may include additional files, depending on the nature of the test. For example, the script reading GUI test includes a script that needs to be read. The purpose of the GUI tests is to validate that the build is stable, and that the user interface panels provide complete coverage of the elements of the system visible to the user.

The GUI test cases forms are relatively simple. They provide, in outline form, guidelines for testing the GUI elements. Detailed instructions for the GUI tests are provided in Chapter 4.

A sample GUI test case is provided here:

```
$Id: ImpulsiveBurnPanel.txt,v 1.4 2006/10/13 19:22:24 dconway Exp $

Description: This test validates the functionality of the Impulsive Burn
configuration panel.

Procedure:

1. Open GMAT.  Create an ImpulsiveBurn resource.

   [  ] Pass   [  ] Fail  Bug# ______

2. Open the panel for the new ImpulsiveBurn.

   [  ] Pass   [  ] Fail  Bug# ______

3. Evaluate the aesthetic qualities of the panel.

   [  ] Pass   [  ] Fail  Bug# ______

4. Evaluate the panel functionality by exercising these elements:

     Axes ComboBox             [  ] Pass   [  ] Fail  Bug# ______

     Vector Format ComboBox    [  ] Pass   [  ] Fail  Bug# ______

     Vector Element 1 Text     [  ] Pass   [  ] Fail  Bug# ______

     Vector Element 2 Text     [  ] Pass   [  ] Fail  Bug# ______

     Vector Element 3 Text     [  ] Pass   [  ] Fail  Bug# ______

     Origin ComboBox           [  ] Pass   [  ] Fail  Bug# ______

5. Evaluate panel save/cancel/restore functionality.

     Cancel          [  ] Pass   [  ] Fail  Bug# ______

     Apply           [  ] Pass   [  ] Fail  Bug# ______
```

```
      Save            [  ] Pass   [  ] Fail  Bug# ______

      Restore         [  ] Pass   [  ] Fail  Bug# ______

      Window Icons    [  ] Pass   [  ] Fail  Bug# ______

6. Evaluate rename functionality.

   [  ] Pass   [  ] Fail  Bug# ______

7. Validate that the configured object is correct on run.

   [  ] Pass   [  ] Fail  Bug# ______

8. Perform additional experiments with the panel controls.

Summary:

Test case status:

   [  ] Pass   [  ] Fail

Bugs Reported:

Notes:




Tester:  ____________________________

Date:    ____________________________
```

Failed GUI tests provide information about the nature of the failure durectly on the test case form; there is no supplementary report for GUI test failures.

2.6 Ensuring Complete System Coverage

Once the test cases have been written, all that remains for test proparation is the confirmation that the test cases cover all of the new features of GMAT. This is accomplished by updating the test matrices based on the new and revised test cases. Each test case that has been added or changed since the last update is collected and used to update the matrices. For each page in the spreadsheet containing an element to test case table, the test team needs to update the matrix for these test cases. The test cases are listed across the top of the matrices. Each test case identifies the tested elements by placing an "X" marker in the row corresponding to that element. Updated test cases should be examined to ensure that elements previously tested are still tested; if an elemnet is no longer tested for a specific test case, the X for that element should be removed from the matrix.

The spreadsheet contains formulas that use these markers to determine if a given element has a corresponding test case. The far right side of the test matrices tables accumulates this data; every element that has at least one associated test case receives a coverage value of 1; uncovered elements receive a coverage value of 0. The far right side of the table also includes a column labeled "Row Count." The row count column simply counts the number of elements on the page.

The summary page examines each table in the spreadsheet and provides information about the coverage com-

Figure 2.6: A Test Tracking Spreadsheet

pleteness of the system tests. Once the coverage statistics report that the elements of the system are covered 100%, the system tests are ready to be run. The test team then generates a new spreadsheet for each type of system test by pressing the "Create Script Test Tracker" and "Create GUI Test tracker" buttons on the summary page. These buttons generate single page spreadsheets used to track progress through the system test. An example is shown in Figure 2.6.

This spreadsheet is used to track and report system test progress. As each system test is performed, the entry in the tracking spreadsheet is updated by the test team. Examination of this spreadsheet provides a status check on the system test.

The next two chapters provide instructions about the steps performed when running the system tests.

Chapter 3

Executing Script Driven Tests

The tests described in this chapter are designed to exercise all accessible objects in the core GMAT engine, in as many combinations as is feasible. This object coverage is performed by running GMAT scripts designed to interact with the accessible objects from the Graphical User Interface. Each script produces output. The system testers examine this output, and, when possible, compare it with the configuration managed output produced from previous runs of the scripts. The procedure followed when running scripted tests is documented in the sections of this chapter.

Figure 3.1: The Script Test Tracking Spreadsheet

3.1 Script Test Case Management

The System test cases are managed from a spreadsheet generated at the conclusion of the system test preparation process. Figure 3.1 shows an example of this test tracking spreadsheet for the script based tests[1], as it looks partway through a test cycle.

The test procedure for script based tests is relatively straightforward. Testers follow these steps when executing the system tests:

1. Obtain the latest versions of the scripts and known good results from the system test repository.
2. Identify the tests each tester needs to run.
3. Open GMAT[2].
4. Run each test following the procedure in 3.2.
5. As each test is run, record the summary results in a local copy of the test tracking spreadsheet.
6. When anomalies are found in testing, record them local test case report files.
7. At the end of each day or when testing is finished, whichever occurs first, gather the test case reports generated from the tests and place them in the folder used to gather the test results.
8. Close GMAT at the end of the test period.
9. At the end of each day or when testing is finished, whichever occurs first, save the local test tracking spreadsheet with the name ¡spreadsheetName¿_¡tester's initials¿ in the folder used to gather the test results.
10. Upon completion of all assigned test cases, report that status to the system test lead.

3.2 Running the Scripted System Tests

By their very nature, the GUI based tests described in Chapter 4 follow a relatively unstructured execution sequence that mandates more structured test case documents to ensure complete system testing. In contrast, the script based tests follow a linear execution sequence once the scripts have been written and debugged. The rest of this chapter describes the procedure followed for the scripted tests.

3.2.1 Procedure

Each scripted test case has an associated, configuration managed script. Most script test cases also have output data files used to compare the obtained script outputs with validated GMAT output files. A tester follows this procedure to perform the associated system test:

1. Open a blank test case report file[3].
2. Open the script in GMAT.

[1] The test tracking spreadsheets, unlike the traceability matrix spreadsheet, can be saved in either OpenOffice or Excel format.

[2] GMAT should only be opened one time for any given testing period. All tests run during that test period – typically a morning or afternoon – should be run in the same instance of GMAT. This helps ensure that the system is stable over long periods of time. If the system is shut down, either by the user or through a system crash, that event should be noted.

[3] The test case report file is only needed for script based tests is an anomaly is found during testing. In practice, the test case report only needs to be opened when an anomaly is found.

3. Compare the resources displayed in GMAT with the resources defined in the script. Enter any anomalies in the test case report.

4. Compare the mission sequence in the script with the mission sequence displayed in GMAT. Enter any anomalies in the test case report.

5. Run the script.

6. Examine each plot and 3D view that opens. Enter any anomalies on the in the test case report.

7. Compare the output results from the run with the known good data. Enter any anomalies in the test case report.

8. Press the run button.

9. Examine each plot and 3D view that opens. Enter any anomalies on the in the test case report.

10. Compare the output results from the run with the known good data. Enter any anomalies in the test case report.

11. Open the script in the editor window, and press the "Build and Run" button.

12. Examine each plot and 3D view that opens. Enter any anomalies on the in the test case report.

13. Compare the output results from the run with the known good data. Enter any anomalies in the test case report.

14. Save the script to a new file with the name Saved_¡Test case name¿.

15. Load the saved script into GMAT.

16. Repeat steps 3 through 11

17. If any anomalies have been found, fill in the header and summary data on the test case report, and save it with the file name "¡test case¿_YYYYMMDD.report", where YYYYMMDD indicate the year, month and day the test was run.

3.2.2 A Note on Run Frequency

The script based tests can be run much more frequently than is feasible for the GUI tests. Scripts that are identified as being run more frequently than at the system test frequency follow a somewhat abbreviated procedure from that defined at the system test level. The purpose of the more frequent testing is to help catch errors in the system prior to format system testing. Teh abbreviated test procedure performed for each weekly or monthly test is presented here:

1. Open the script in GMAT.

2. Run the script.

3. Examine each plot and 3D view that opens. Report any anomalies.

4. Compare the output results from the run with the known good data. Report any anomalies.

5. If any anomalies have been found, enter a new anomaly in the bug tracking system.

These tests follow the full system test procedure when run as part of the system test suite.

3.2.3 Reporting Results

At the start of the system test process, a central location was established for collection of the test results. The final step performed by the system testers is to copy their test case worksheets and local test tracking worksheet to this central location. This action is performed each day the system tests are run so that the progress of the system test execution can be evaluated. Upon completion of all system testing by a specific tester, a final update is made and the system test lead is notified that that tester has completed the assigned tests. Chapter 5 describes the consolidation of the collected test results into a system test report.

Chapter 4

Executing Tests for the Graphical User Interface

The tests described in this chapter are designed to exercise all of the controls and other elements visible from the GMAT graphical user interface (GUI). The GMAT GUI is designed to present a consistent, easy to use interface into the underlying engine so that users of the system can view, configure, and interact with the elements of the system during all phases of mission modeling. System testers work with these elements, using them both to perform the expected tasks and to attempt to perform undesired actions. The former set of actions exercises the engine to ensure that the system can be configured correctly. The latter tests are run to ensure that users cannot configure GMAT incorrectly.

4.1 GUI Test Case Management

The GUI test cases are managed using a test tracking spreadsheet generated at the end of test preparation, described in Chapter 2. Figure 4.1 shows an example of this spreadsheet partway through a testing cycle.

The test procedure for GUI based tests requires extensive exercising of the components in the GUI. Testers follow these steps when executing the system tests:

1. Obtain the latest versions of the GUI test cases and a local copy of the test case tracking spreadsheet[1].
2. Identify the tests that the tester needs to run.
3. Open GMAT[2].
4. Run each test following the procedure in Section 4.2.
5. As each test is run, record the results of the test on the test case worksheet retrieved in step 1.
6. When anomalies are found in testing, record them on the test case worksheet and enter them in the bug tracking database.
7. Close GMAT at the end of the test period.

[1] The test tracking spreadsheet is generated from the Systen Test Matrix spreadsheet using an OpenOffice macro, as described in Section 2.6.

[2] GMAT should only be opened one time for any given testing period. All tests run during that test period – typically a morning or afternoon – should be run in the same instance of GMAT. This helps ensure that the system is stable over long periods of time. If the system is shut down, either by the user or through a system crash, that event should be noted.

Figure 4.1: The GUI Test Tracking Spreadsheet

8. At the end of each day or when testing is finished, whichever occurs first, gather the completed test case worksheets and place them in the folder used to gather the test results.

9. At the end of each day or when testing is finished, whichever occurs first, save the local gui test tracking spreadsheet with the name ¡spreadsheetName¿_¡tester's initials¿ in the folder used to gather the test results.

10. Upon completion of all assigned test cases, report that status to the system test lead.

The procedure for running a single test case is described next.

4.2 Running the GUI System Tests

By their very nature, the script based tests described in Chapter 3 follow a linear execution sequence once the scripts have been written and debugged. In contrast, interactions performed using the GMAT GUI are less structured – users can use the controls on the GUI in a seemingly random fashion – so the test cases for the GUI include allowances for interacting with the GUI elements by the tester in a more free form manner than the script based tests allow.

4.2.1 Sample GUI Test Case

A sample GUI test case is shown here:

```
$Id $


Tester: _______________________________      Date: _________________________


Description:

This test validates the functionality of the OpenGL panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Create and open the appropriate object panel.

   Create OpenGL Resource                    [ ] Pass  [ ] Fail  Bug# ______

   Open OpenGL Resource                      [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.

   Show Plot Check Box                       [ ] Pass  [ ] Fail  Bug# ______

   Collect Data Text Field                   [ ] Pass  [ ] Fail  Bug# ______

   Update Plot Text Field                    [ ] Pass  [ ] Fail  Bug# ______

   Number of Points to Redraw Text Field     [ ] Pass  [ ] Fail  Bug# ______

   Draw Wireframe Check Box                  [ ] Pass  [ ] Fail  Bug# ______

   Draw Targeting Check Box                  [ ] Pass  [ ] Fail  Bug# ______

   Draw Ecliptic Plane Check Box             [ ] Pass  [ ] Fail  Bug# ______

   Draw XY Plane Check Box                   [ ] Pass  [ ] Fail  Bug# ______

   Draw Axes Check Box                       [ ] Pass  [ ] Fail  Bug# ______

   Draw Grid Check Box                       [ ] Pass  [ ] Fail  Bug# ______

   Draw Earth/Sun Lines Check Box            [ ] Pass  [ ] Fail  Bug# ______

   Spacecraft List                           [ ] Pass  [ ] Fail  Bug# ______

```

```
   Selected Spacecraft List                  [ ] Pass  [ ] Fail  Bug# ______

   Celestial Object List                     [ ] Pass  [ ] Fail  Bug# ______

   Selected Celestial Object List            [ ] Pass  [ ] Fail  Bug# ______

   --> (Add) Selection Button                [ ] Pass  [ ] Fail  Bug# ______

   <-- (Remove) Selection Button             [ ] Pass  [ ] Fail  Bug# ______

   < = (Remove All) Selection Button         [ ] Pass  [ ] Fail  Bug# ______

   Show Object Check Box                     [ ] Pass  [ ] Fail  Bug# ______

   Orbit Color Select Box                    [ ] Pass  [ ] Fail  Bug# ______

   Target Color Select Box                   [ ] Pass  [ ] Fail  Bug# ______

   Use Initial View Definition Check Box     [ ] Pass  [ ] Fail  Bug# ______

   Use Perspective Mode Check Box            [ ] Pass  [ ] Fail  Bug# ______

   Use Fixed FOV Angle Check Box             [ ] Pass  [ ] Fail  Bug# ______

   Field of View Text Field                  [ ] Pass  [ ] Fail  Bug# ______

   Coordinate System Combo Box               [ ] Pass  [ ] Fail  Bug# ______

   View Point Reference Combo Box (see 4a)   [ ] Pass  [ ] Fail  Bug# ______

   View Point Vector Combo Box (see 4b)      [ ] Pass  [ ] Fail  Bug# ______

   View Scale Factor Text Field              [ ] Pass  [ ] Fail  Bug# ______

   View Direction Combo Box (see 4c)         [ ] Pass  [ ] Fail  Bug# ______

   Coordinate System Combo Box               [ ] Pass  [ ] Fail  Bug# ______

   Axis Combo Box                            [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

   a. Select 'Vector' for View Point Reference

      Vector 1 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 2 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 3 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

   b. Select 'Vector' for View Point Vector

      Vector 1 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 2 Text Field                    [ ] Pass  [ ] Fail  Bug# ______
```

```
110       Vector 3 Text Field                    [ ] Pass  [ ] Fail  Bug# ______
111
112    c. Select 'Vector' for View Direction
113
114       Vector 1 Text Field                    [ ] Pass  [ ] Fail  Bug# ______
115
116       Vector 2 Text Field                    [ ] Pass  [ ] Fail  Bug# ______
117
118       Vector 3 Text Field                    [ ] Pass  [ ] Fail  Bug# ______
119
120    Use Perspective Mode Check Box            [ ] Pass  [ ] Fail  Bug# ______
121    --- select checkbox to check following
122
123    Use Fixed FOV Angle Check Box             [ ] Pass  [ ] Fail  Bug# ______
124    --- select checkbox to check following
125
126    Field of View Text Field                  [ ] Pass  [ ] Fail  Bug# ______
127
128
129 5. Evaluate data.
130
131    Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______
132
133    Show Script                               [ ] Pass  [ ] Fail  Bug# ______
134
135
136 6. Evaluate panel control.
137
138    Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______
139
140    Cancel                                    [ ] Pass  [ ] Fail  Bug# ______
141
142    Apply                                     [ ] Pass  [ ] Fail  Bug# ______
143
144    OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______
145
146    Help [DEFERRED]
147
148    Restore                                   [ ] Pass  [ ] Fail  Bug# ______
149
150    Minimize                                  [ ] Pass  [ ] Fail  Bug# ______
151
152    Maximize                                  [ ] Pass  [ ] Fail  Bug# ______
153
154    Close                                     [ ] Pass  [ ] Fail  Bug# ______
155
156
157 7. Evaluate rename functionality.
158
159    Rename (on resource tree)                 [ ] Pass  [ ] Fail  Bug# ______
160
161
162 Summary:
163
164    Number of passed test elements      ______
165
166    Total number of test elements       ______
```

```
167
168     Test case status                          [ ] Pass  [ ] Fail
169
170
171  Bugs Reported:
172
173
174
175  Notes:
176
177
```

Figure 4.2: The OpenGLPlot Setup Panel

The test case worksheet shown here is the test case for the OpenGL plot setup panel. The panel, shown in Figure 4.2, is a fairly complex GUI panel, containing text fields, combo boxes, check boxes, text lists, and action buttons which open color selection dialogs. Each element is included in the test plan worksheet, along with the standard control processes that need to be exercised. Each test criterion is evaluated using this worksheet, and given a pass or fail evaluation.

4.2.2 Procedure

Each GUI test case has a worksheet like the one shown above. A tester follows this procedure to perform the associated system test:

1. Open the test case worksheet.
2. Follow the procedure outlined in the test case.
   - Section 4.3 provides detailed instructions about the process that should be followed when testing each type of GUI element.
   - Each item identified in the worksheet is marked as either passing or failing the test. If the item fails, an associated bug is entered or identified in the bug tracking system and listed on the worksheet.

- After completing the tests on the worksheet, the tester experiments with the component for an additional period (typically ten to fifteen minutes), checking to be sure that the component is stable and behaves correctly when bad data is entered, and when random actions are taken using that component.
- Once every item on the worksheet has been evaluated and the final period of usability testing has been performed, the number of pass and fail evaluations are counted and recorded in the summary section of the test case worksheet. Any bugs identified on the worksheet are listed in this section, and any additional notes that need to be recorded are also listed here[3].

3. Summarize the results of the tests.
   - Once every item on the worksheet has been evaluated, an overall pass or fail evaluation is made and recorded in the summary section. Any bugs identified on the worksheet are listed in this section, and any additional notes that need to be recorded are also listed here.
   - Add the tester's name and the data the test was run to the worksheet.
   - Save the completed test case worksheet.
4. Update the local test tracking worksheet to indicate that the test was run and the results of the run.
5. Save the test tracking worksheet.

4.2.3 Reporting Results

At the start of the system test process, a central location was established for collection of the test results. The final step performed by the system testers is to copy their test case worksheets and local test tracking worksheet to this central location. This action is performed each day the system tests are run so that the progress of the system test execution can be evaluated. Upon completion of all system testing by a specific tester, a final update is made and the system test lead is notified that that tester has completed the assigned tests. Chapter 5 describes the consolidation of the collected test results into a system test report.

4.3 Procedural Rules

The steps described in the preceding sections lay out the procedures followed when testing the GUI elements of GMAT. In this section, the criteria that must be evaluated are defined for these tests.

4.3.1 Test Procedures for All Elements

Aesthetics

Description: This set of tests verifies platform-specific look and feel of a panel, as extended by the GMAT GUI Philosophy document.[Dove] Each criterion must be met to pass the aesthetics tests.

- All of the data input fields and bounding boxes can be seen at the panel size displayed when the panel is first opened, for all tabs on the panel.
- The blank space surrounding the data area is not distracting, and does not dominate the appearance of the interface. As a guideline, for platforms that allow control of the surrounding white space, that region should not consume more than 20% of the total space dedicated to the panel when it is opened.
- The data area does not appear too crowded; the surrounding blank space is appropriately sized.
- The window cannot be resized so that the data cannot be seen.

[3] These data are collected using an automation tool to build a status report for the system tests.

General Panel Functionality

Description: This is the list of tests associated with basic panel functionality: open, close, rename, minimize, ok, cancel, help, show script, command summary. Additionally, the behavior of open panels needs to be consistent with deletion actions taken on the resource and mission trees – if an object in the tree is deleted, any open panel associated with that object should close. All of these functions must pass.

- New objects of the type being tested can be created from the appropriate tree on the Resource or Mission panels.
- Double clicking in a new object opens the panel for that object.
- Double clicking in a object that has an open panel brings the panel for that object to the front of the displayed panels.
- New objects can be renamed.
- Default objects, when they exist, can be renamed.
- Default objects, when they exist, can be deleted.
  - The object can be renamed.
  - References to the renamed object are updated in related elements of the system.
- Renaming works after making changes to the data on the object panel.
  - The object can be renamed while the panel is open.
  - A change can be made on the panel, and then the object can be renamed before the change is applied.
  - A change can be made on the panel, the change can be applied, and then the object can be renamed.
  - For each of the above cases, references to the object's name are updated throughout the system when the object's name is changed.
- Changes made on the panel and applied using the OK button appear on the panel when it is reopened.
- Changes made on the panel and applied using the Apply button are visible in the script when viewed using the Show Script dialog.
- When you open the panel, make a minor change in the panel, and click button to close the panel (on Windows, this is the small "x" button in the upper right hand corner; on the Mac, it is the red button on the left side of the frame controls, and on Linux, varies based on the configuration of the Linux window manager), you are prompted to save data before closing. Check that:
  - The prompt does appear.
  - Selecting "Yes" updates the underlying data.
  - Selecting "No" discards the changes.
- Cancelling closes the opened panel without changing the underlying data.
  - The object does not change when you open the panel and press the Cancel button without making any changes.
  - The object does not change when you open the panel, make a minor change in the data, and press the Cancel button.
  - The object does not change when you open the panel and click the close button in the panel's frame to close the panel, but the panel does close without prompting.
- The panel is minimized when the minimize button on the panel frame is pressed.
- The panel reopens to previous size when maximize icon on the minimized panel is pressed
- The tab key navigates the open panel in agreement with style and GUI design philosophy. Navigation is orderly and sensible using the tab key.

Panel Data Element Completeness and Correctness

Description: This set of tests verifies that all data elements that should appear on the panel are present on the panel. It also tests that all elements that should appear in "Show Script" dialog appear there, and that items that should not appear in show script do not appear there.

- Verify that only data elements that occur in the Range Test Plan appear in show script and that the user does not see any other object fields.
- Verify that defaults agree with the values in the Range Test Plan.
- Press the "Show Script" button, and verify that all elements on the GUI panel also appear on the show script dialog. Verify that these elements match the description in the Range Test Plan.
- Verify that all data elements that appear in Show Script also appear on the GUI. (This step validates that all scriptable settings also appear in the GUI.)

4.3.2 Procedures for Specific Control Types

The following table provides additional guidelines that should be followed when testing each specific type of control.

Table 4.1: Tests for Data Objects on All Panels

Element Type	Tests
Check Boxes	• Set all check boxes to off (unchecked), hit show script, and verify that the functionality is indeed turned off for each radio button and check box. • Set all check buttons to on (checked), hit apply, and show script and verify that the functionality is indeed turned on for each radio button and check box.
Radio Buttons	• For each radio button on panel, select the button, and ensure that it activates and all others are deactivated. Hit Apply, and then check show script to ensure that the configuration was properly saved.
Combo Boxes	• For each combo box on the panel, ensure that all options that appear in Range Test Plan appear in the pull down menu. • For each Combo box on the panel, select each allowable option, hit apply and show script and check to see that the option was correctly saved. • Check to ensure that the combo box is not editable.

Table 4.1: (Tests for Data Objects on All Panels...continued)

Element Type	Tests
Text Fields	• For each text field enter "DNE" and ensure that if GMAT should reject this string that the string is rejected. (Currently, this is not an acceptable value for any GMAT field unless the user has created an appropriate object type and named it DNE, and is using it correctly in the GUI.) • Perform all range tests as described in Range Test Plan. • For all numeric fields, enter an allowed numeric value, hit apply and show script and check that the value was saved. • If user-defined objects can appear in the combo box, create one object for all allowable object types for the particular combo box, and ensure that it appears in the combo box. Also, hit apply and ensure that each case appears in show script.
Action Buttons	• For each button ensure that clicking on the button brings up the appropriate panel. • For the panel opened up, perform all tests defined in Section 4.3.1 and Table 4.1
Selection Lists	• First Item • Second Item
Tabbed Panels	• First Item • Second Item

4.3.3 Usability Testing

The tests described in the preceding paragraphs are meant to exercise all of the elements of the graphical user interface. One important aspect of the interface not covered by those tests is the usability of the system: the GUI may perform error free as designed, and still be difficult to use in practice. Usability testing is performed to capture information about this aspect of the GUI.

Chapter 5

Reporting and Reviewing Test Results

This chapter describes the process followed for tracking the state of the system test process and for reporting the results of the testing.

5.1 System Test Status

The status of the system tests is tracked using the Script and GUI test tracking spreadsheets described in Chapters 3 and 4. System testers update their copies of these spreadsheet daily during system testing. Once a week or upon request, the system test lead consolidates these spreadsheets, collecting the test results in master system test spreadsheets that can be reviewed by interested parties.

5.2 The System Test Report

At the conclusion of system test cycle, the reports generated during system test are consolidated into a single document. This document is prepared using the following outline:

I. Overview
   A. Executive Summary
   B. Test Results
   C. Recommendations

II. Script Test Case Results
   A. Test Result Statistics
   B. Summary of Failed Tests (if any)
   C. Test Results
      i. ParametersinCommands Test Case Report
      ii. CbParams_GMAT_GEO_2Body Test Case Report
      ...

III. GUI Test Case Results
   A. Test Result Statistics

B. Summary of Failed Tests (if any)

C. Test Results

   i. Mainframe Test Case Worksheet

   ii. Resource Tree Test Case Worksheet

   ...

5.3 System Test Review

The final step in the system test process is to perform a review of the test results. In preparation for this review, each team member and reviewer reviews the System Test Report, highlighting any issues that raise concerns. These parties then meet and discuss the findings of the system testing. The outcome of this review is a list of action items, assigned to specific individuals or teams, and a recommendation about the status of the system for release.

A typical release recommendation will fall into one of three categories: (1) GMAT is ready for release, (2) GMAT is ready for release, contingent on specific items being addressed and approved prior to that release, or (3) GMAT is not ready for release, and needs to meet specific items and be reviewed again before release will be approved.

Following this review, a summary documenting the findings of the review is written and provided to all team members and interested parties. Once GMAT has been released as an open source project, a public version of this summary is made available with the other project artifacts.

Part III

Tests

Chapter 6

Stress Tests

Name	TC-4 Rapid Rerun Via Run Button
Requirements	FR-XXX
Summary	Test opening a script and running it numerous times back-to-back using the run button on the toolbar.
PreConditions	BS-5, BS-6, BS-7, BS-8, BS-9
Steps	1. Load BS-5. 2. Run the mission using the run button on the toolbar. 3. The run button is greyed out during execution of the script. Hover the mouse button over the run button and when it becomes active again quickly press it to rerun the mission. 4. Perform the previous step 5 times.
Alternative Paths	Perform all steps above for the other base states listed in the Base States row of this table.
Expected Results	The system should rerun the missions over and over again without crashing.)

Table 6.1: TC-4 Rapid Rerun Via Play Button

Name	TC-5 Rapid Rerun Via Build and Run
Requirements	FR-XXX
Summary	Test opening a script and running it numerous times back-to-back using the run button on the toolbar.
PreConditions	BS-5, BS-6, BS-7, BS-8, BS-9
Steps	1. Load BS-5. 2. Run the mission by right-clicking on the script name in the mission tree and selecting build and run. 3. The run button is greyed out during execution of the script. Hover the mouse button over the script in the script folder in the resource tree. When the run button is active again quickly right-click on the script folder and select build and run. 4. Perform the previous step 5 times.
Alternative Paths	Perform all steps above for the other base states listed in the Base States row of this table.
Expected Results	The system should rerun the missions over and over again without crashing.)

Table 6.2: TC-5 Rapid Rerun Via Build and Run

Name	TC-6 Running a Collection of Scripts Several Times
Requirements	FR-XXX
Summary	This cases tests loading and running a set of tests in different orders many times in the same session.
PreConditions	BS-5, BS-6, BS-7, BS-8, BS-9
Steps	1. Load BS-5 via the folder icon in the toolbar. 2. Run BS-5 by right-clicking on the script name in the resource tree and selecting build and run. 3. Load BS-6 via the folder icon in the toolbar. 4. Run BS-6 by right-clicking on the script name in the resource tree and selecting build and run. 5. Load BS-7 via the folder icon in the toolbar. 6. Run BS-7 by right-clicking on the script name in the resource tree and selecting build and run. 7. Load BS-8 via the folder icon in the toolbar. 8. Run BS-8 by right-clicking on the script name in the resource tree and selecting build and run. 9. Load BS-9 via the folder icon in the toolbar. 10. Run BS-9 by right-clicking on the script name in the resource tree and selecting build and run. 11. Run BS-5 by right-clicking on the script name in the resource tree and selecting build and run. 12. Run BS-6 by right-clicking on the script name in the resource tree and selecting build and run. 13. Run BS-7 by right-clicking on the script name in the resource tree and selecting build and run. 14. Run BS-8 by right-clicking on the script name in the resource tree and selecting build and run. 15. Run BS-9 by right-clicking on the script name in the resource tree and selecting build and run. 16. Run BS-8 by right-clicking on the script name in the resource tree and selecting build and run. 17. Run BS-8 by right-clicking on the script name in the resource tree and selecting build and run. 18. Run BS-6 by right-clicking on the script name in the resource tree and selecting build and run. 19. Run BS-5 by right-clicking on the script name in the resource tree and selecting build and run. 20. Run BS-8 by right-clicking on the script name in the resource tree and selecting build and run. 21. Run BS-6 by right-clicking on the script name in the resource tree and selecting build and run. 22. Run BS-7 by right-clicking on the script name in the resource tree and selecting build and run. 23. Run BS-9 by right-clicking on the script name in the resource tree and selecting build and run.
Alternative Paths	Perform all steps above for the other base states listed in the Base States row of this table.
Expected Results	The system should rerun the missions over and over again without crashing.)

Table 6.3: TC-6 Running a Collection of Scripts Several Times

Chapter 7

Resources

7.1 Spacecraft

7.1.1 Nominal Behavior of Spacecraft Orbit Panel

Name	STC-3 Conversion to Keplerian-type Elements Disallowed when Celestial Body Not at Origin
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to a Keplerian state when the coordinate system does not have a celestial body (i.e. mu value) at the origin.
PreConditions	BS-2
Steps	1. Load BS-2. 2. Open the dialog box for DefaultSC. 3. Change the StateType to Keplerian. 4. Click on the down arrow on the Coordinate System drop-down menu and inspect the available Coordinate Systems. 5. Change the StateType to Modified Keplerian. 6. Click on the down arrow on the Coordinate System drop-down menu and inspect the available Coordinate Systems. 7. Change the StateType to Equinoctial. 8. Click on the down arrow on the Coordinate System drop-down menu and inspect the available Coordinate Systems.
Expected Results	The only coordinate systems available in the inspection steps above should be EarthMJ2000Eq, EarthMJ2000Ec, and Earth Fixed. Coordinate Systems CS_ESL2 and CS_SSBary are NOT available because these orbit state representations are only valid for coordinate systems with a central body at the origin.

Table 7.1: STC-3 Conversion to Keplerian-type Elements Disallowed when Celestial Body Not at Origin.

Name	STC-4 Conversion to Disallowed Coordinate System from Keplerian-type Elements
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to a new coordinate system that does not have a celestial body at the center.
PreConditions	BS-2
Data	1. Load BS-2. 2. Open the dialog box for DefaultSC. 3. Change the Coordinate System to CS_ESL2. 4. Click on the down arrow on the State Type drop-down menu and inspect the available State Types. 5. Change the Coordinate System to CS_SSBary. 6. Click on the down arrow on the State Type drop-down menu and inspect the available State Types.
Expected Results	The only State Types available are Cartesian, SphericalRADEC, and SphericalAZFPA. State Types Keplerian, Modified Keplerian, and Equinoctial are NOT available because these orbit state representations are only valid for coordinate systems with a central body at the origin.

Table 7.2: STC-4 Conversion to Disallowed Coordinate System from Keplerian-type Elements

Name	STC-5 GUI Epoch and State Independence for Time Dependent Coordinate System
Requirements	FR-1.3
Summary	This test is to verify that changing the epoch of the spacecraft, does not effect the orbit state, even when the coordinate system is, for example, a libration point coordinate system that has a time varying origin and axis system.
PreConditions	BS-2
Steps	1. Load BS-2 2. Open the dialog box for DefaultSC 3. Change the CoordinateSystem to CS_ESL2 4. Change the Epoch Format to UTCGregorian 5. Change the Epoch value to 01 Jan 2010 12:00:00.000 6. Hit Ok to close the dialog box 7. Reopen the dialog box for DefaultSC.
Expected Results	The data in the GUI should agree with the data below to at least 12 significant figures. • DefaultSC.X = 273083.6097699367 ; • DefaultSC.Y = -1332500.504835084; • DefaultSC.Z = -576402.9744365886 ; • DefaultSC.VX = 0.2990482122160891; • DefaultSC.VY = 7.400368588891073; • DefaultSC.VZ = 1.021835464804587 ;

Table 7.3: STC-5 GUI Epoch and State Independence for Time Dependent Coordinate System

Name	STC-6 Orbit State Conversion for Singular Conic Section
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations with a cartesian state that results in a singular conic section.
PreConditions	BS-2 and TD-4
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Enter the Cartesian state data from TD-5. 4. Hit Apply. 5. Change the State Type to Keplerian and verify the following error message is thrown: "GMAT does not support parabolic orbits in conversion from Cartesian to Keplerian state". 6. Change the State Type to Modified Keplerian and verify the following error message is thrown: "GMAT does not support parabolic orbits in conversion from Cartesian to Keplerian state". 7. Change the state to SphericalRADEC and verify the numeric data with TD-5. 8. Change the state to SphericalAZEL and verify the numeric data with TD-5. 9. Change the State Type to Equinoctial and verify the following error message is thrown: "GMAT does not support parabolic orbits in conversion from Cartesian to Equinoctial state".
Expected Results	The only State Types available are Cartesian, SphericalRADEC, and SphericalAZFPA. State Types Keplerian, Modified Keplerian, and Equinoctial are NOT available because they are undefined.

Table 7.4: STC-6 Orbit State Conversion for Singular Conic Section

Name	STC-7 Orbit State Conversion for Circular, Equatorial Orbit
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1 and TD-5
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Enter the Cartesian state data from TD-5. 4. Hit Apply. 5. Change the state to Keplerian and verify the numeric data with TD-5. 6. Change the state to Modified Keplerian and verify the numeric data with TD-5. 7. Change the state to SphericalRADEC and verify the numeric data with TD-5. 8. Change the state to SphericalAZEL and verify the numeric data with TD-5. 9. Change the state to Equinoctial and verify the numeric data with TD-5.
Expected Results	The truth data is contained in TD-5.

Table 7.5: STC-7 Orbit State Conversion for Circular, Equatorial Orbit

Name	STC-8 Orbit State Conversion for for Circular, Inclined Orbit
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1 and TD-6
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Enter the Cartesian state data from TD-6. 4. Hit Apply. 5. Change the state to Keplerian and verify the numeric data with TD-6. 6. Change the state to Modified Keplerian and verify the numeric data with TD-6. 7. Change the state to SphericalRADEC and verify the numeric data with TD-6. 8. Change the state to SphericalAZEL and verify the numeric data with TD-6. 9. Change the state to Equinoctial and verify the numeric data with TD-6.
Expected Results	The truth data is contained in TD-6.

Table 7.6: STC-8 Orbit State Conversion for Circular, Inclined Orbit

Name	STC-9 Orbit State Conversion for Orbit with Zero Velocity
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1

Data	1. Load BS-1.
	2. Open the dialog box for DefaultSC.
	3. Enter the following Cartesian State data:
	  1. X = 7000
	  2. Y = 7000
	  3. Z = 7000
	  4. VX = 0;
	  5. VY = 0;
	  6. VZ = 0;
	4. Hit Apply.
	5. Change the state to Keplerian and verify that the following error message is returned: The orbit is a singular conic section and the Keplerian elements are undefined.
	6. Change the state to Modified Keplerian and verify that the following error message is returned: The orbit is a singular conic section and the Modified Keplerian elements are undefined.
	7. Change the state to SphericalRADEC and verify that the following error message is returned: The orbit is a singular conic section and the SphericalRADEC elements are undefined.
	8. Change the state to SphericalAZEL and verify that the following error message is returned: The orbit is a singular conic section and the SphericalAZEL elements are undefined.
	9. Change the state to Equinoctial and verify that the following error message is returned: The orbit is a singular conic section and the Equinoctial elements are undefined.

Expected Results	The truth data is described above.

Table 7.7: STC-9 Orbit State Conversion for Orbit with Zero Velocity

Name	STC-10 Orbit State Conversion for Orbit with Zero Position
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1

Data	1. Load BS-1.
	2. Open the dialog box for DefaultSC.
	3. Enter the following Cartesian State data:
	  1. X = 0.0
	  2. Y = 0.0
	  3. Z = 0.0
	  4. VX = 7.0;
	  5. VY = 7.0;
	  6. VZ = 7.0;
	4. Hit Apply.
	5. Change the state to Keplerian and verify that the following error message is returned: The orbit is a singular conic section and the Keplerian elements are undefined.
	6. Change the state to Modified Keplerian and verify that the following error message is returned: The orbit is a singular conic section and the Modified Keplerian elements are undefined.
	7. Change the state to SphericalRADEC and verify that the following error message is returned: The orbit is a singular conic section and the SphericalRADEC elements are undefined.
	8. Change the state to SphericalAZEL and verify that the following error message is returned: The orbit is a singular conic section and the SphericalAZEL elements are undefined.
	9. Change the state to Equinoctial and verify that the following error message is returned: The orbit is a singular conic section and the Equinoctial elements are undefined.

Expected Results	The truth data is described above.

Table 7.8: STC-10 Orbit State Conversion for Orbit with Zero Position

Name	STC-11 Orbit State Conversion for Orbit with Zero State
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1

Data	1. Load BS-1.
	2. Open the dialog box for DefaultSC.
	3. Enter the following Cartesian State data:
	  1. X = 0.0
	  2. Y = 0.0
	  3. Z = 0.0
	  4. VX = 0.0;
	  5. VY = 0.0;
	  6. VZ = 0.0;
	4. Hit Apply.
	5. Change the state to Keplerian and verify that the following error message is returned: The orbit is a singular conic section and the Keplerian elements are undefined.
	6. Change the state to Modified Keplerian and verify that the following error message is returned: The orbit is a singular conic section and the Modified Keplerian elements are undefined.
	7. Change the state to SphericalRADEC and verify that the following error message is returned: The orbit is a singular conic section and the SphericalRADEC elements are undefined.
	8. Change the state to SphericalAZEL and verify that the following error message is returned: The orbit is a singular conic section and the SphericalAZEL elements are undefined.
	9. Change the state to Equinoctial and verify that the following error message is returned: The orbit is a singular conic section and the Equinoctial elements are undefined.

Expected Results	The truth data is described above.

Table 7.9: STC-11 Orbit State Conversion for Orbit with Zero State

Name	STC-12 Performing Modulo on Keplerian Angular Elements
Requirements	FR-1.3
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Keplerian 4. Change INC to 370.0 degrees. 5. Change RAAN to 380.0 degrees. 6. Change AOP to 390.0 degrees. 7. Change TA t0 400.0 degrees. 8. Change the State Type to Cartesian. 9. Change the State Type to Keplerian.
Expected Results	INC = 10.0 degrees, RAAN = 20 degrees, AOP = 30.0 degrees, and TA = 40.0 degrees. (All values match to 14 sig. figs.)

Table 7.10: STC-12 Performing Modulo on Keplerian Angular Elements

Name	STC-13 Performing Modulo on Modified Keplerian Angular Elements
Requirements	FR-1.3
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Modified Keplerian 4. Change INC to 370.0 degrees. 5. Change RAAN to 380.0 degrees. 6. Change AOP to 390.0 degrees. 7. Change TA t0 400.0 degrees. 8. Change the State Type to Cartesian. 9. Change the State Type to Keplerian.
Expected Results	INC = 10.0 degrees, RAAN = 20 degrees, AOP = 30.0 degrees, and TA = 40.0 degrees. (All values match to 14 sig. figs.)

Table 7.11: STC-13 Performing Modulo on Modified Keplerian Angular Elements

Name	STC-14 Performing Modulo on SphericalRADEC Angular Elements
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Spherical 4. Change RA to 370.0 degrees. 5. Change DEC to 380.0 degrees. 6. Change RAV to 390.0 degrees. 7. Change DECV t0 400.0 degrees. 8. Change the State Type to Cartesian. 9. Change the State Type to Keplerian.
Expected Results	RA = 10.0 degrees, DEC = 20 degrees, RAV = 30.0 degrees, and DECV = 40.0 degrees. (All values match to 14 sig. figs.)

Table 7.12: STC-14 Performing Modulo on SphericalRADEC Angular Elements

7.1.2 Input Validation of Spacecraft Orbit Panel

7.2 Spacecraft Attitude Tab

Name	STC-15 Performing Modulo on SphericalAZFPA Angular Elements
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Spherical 4. Change RA to 370.0 degrees. 5. Change DEC to 380.0 degrees. 6. Change AZI to 390.0 degrees. 7. Change FPA t0 400.0 degrees. 8. Change the State Type to Cartesian. 9. Change the State Type to Keplerian.
Expected Results	RA = 10.0 degrees, DEC = 20 degrees, AZI = 30.0 degrees, and FPA = 40.0 degrees. (All values match to 14 sig. figs.)

Table 7.13: STC-15 Performing Modulo on SphericalAZFPA Angular Elements

Name	STC-16 Performing Modulo on Equinoctial Angular Elements
Requirements	FR-1.3
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Spherical 4. Change RA to 370.0 degrees. 5. Change DEC to 380.0 degrees. 6. Change AZI to 390.0 degrees. 7. Change FPA t0 400.0 degrees. 8. Change the State Type to Cartesian. 9. Change the State Type to Keplerian.
Expected Results	RA = 10.0 degrees, DEC = 20 degrees, AZI = 30.0 degrees, and FPA = 40.0 degrees. (All values match to 14 sig. figs.)

Table 7.14: STC-16 Performing Modulo on Equinoctial Angular Elements

Name	STC-27 Performing Modulo on Keplerian Elements for Circular, Equatorial orbit
Requirements	FR-1.1
Summary	This case tests GUI behavior when attempting to convert to element representations when the cartesian state results in a circular, equatorial orbit.
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Keplerian 4. Change ECC to 0.0. 5. Change INC to 0.0 degrees. 6. Change RAAN to 380.0 degrees. 7. Change AOP to 390.0 degrees. 8. Change TA to 430.0 degrees. 9. Hit Apply.
Expected Results	RAAN = 0.0 degrees, AOP = 0.0 degrees, and TA = 120.0 degrees. (All values match to 14 sig. figs.)

Table 7.15: STC-27 Performing Modulo on Keplerian Elements for Circular, Equatorial orbit

Name	STC-28 Performing Modulo on Keplerian Elements for Circular, Inclined Orbit
Requirements	FR-1.1
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Keplerian 4. Change ECC to 0.0. 5. Change INC to 45 degrees. 6. Change RAAN to 380.0 degrees. 7. Change AOP to 390.0 degrees. 8. Change TA to 430.0 degrees. 9. Hit Apply.
Expected Results	RAAN = 20.0 degrees, AOP = 0.0 degrees, and TA = 70.0 degrees. (All values match to 14 sig. figs.)

Table 7.16: STC-28 Performing Modulo on Keplerian Elements for Circular, Inclined orbit

Name	STC-18 Orbit State Conversion when Orbit is Near Parabolic
Requirements	FR-1.1
Summary	SMA is undefined for parabolic orbits. This test check behavior as orbit approaches parabolic from ECC ¡ 1 side for Keplerian state type.
PreConditions	BS-1
Steps	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Change the State Type to Keplerian 4. Change ECC to 0.99999999 5. Hit Apply.
Expected Results	The following error message should be displayed: The value of "0.99999999" for field "ECC" is not an allowed value. The allowed values are: [0.0 *leq* Real Number $\leq$ 0.9999999, or Real Number $\geq$ 1.0000001].)

Table 7.17: STC-18 Orbit State Conversion when Orbit is Near Parabolic

Name	STC-19 Orbit State Conversion for Nearly Singular Cartesian State
Requirements	FR-1.3
PreConditions	BS-1

Steps	1. Load BS-1.
	2. Open the dialog box for DefaultSC.
	3. Set the Cartesian state to the following values
	  1. X = 6999.998216286026
	  2. Y = 0
	  3. Z = -5.002359263770285
	  4. VX = 10.63431352889248
	  5. VY = 0
	  6. VZ = -0.003772975815698364
	4. Hit Apply.
	5. Change state type to Keplerian and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the Cartesian state to the Keplerian elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	6. Change state type to Modified Keplerian and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the Cartesian state to the Modified Keplerian elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	7. Change state type to Equinoctial and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the Cartesian state to the Equinoctial elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	8. Change the following states.
	  1. X = 1e-10
	  2. Y = 1e-10
	  3. Z = 1e-10
	9. Hit Apply
	10. Repeat steps 5, 6, and 7.
	11. Change state type to SphericalRADec and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialRADEC state to the Cartesian State so conversion was aborted. The Right Ascension and Declination of position are undefined."
	12. Change state type to SphericalAZFPA and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialAZFPA state to the Cartesian State so conversion was aborted. The Right Ascension and Declination of position are undefined."
	13. Change the following states:
	  1. X = 6999.998216286026
	  2. Y = 0
	  3. Z = -5.002359263770285
	  4. VX = 1e-10
	  5. VY = 1e-10
	  6. VZ = 1e-10
	14. Hit Apply
	15. Repeat steps 5, 6, and 7.
	16. Change state type to SphericalRADec and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the Cartesian state to the SphericalRADEC so conversion was aborted. The Right Ascension and Declination of velocity are undefined."
	17. Change state type to SphericalAZFPA and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialAZFPA state to the Cartesian State so conversion was aborted. The Right Ascension and Declination of velocity are undefined."

Expected Results	Test results are described above.

Table 7.18: STC-19 Orbit State Conversion for Nearly Singular Cartesian State

Name	STC-20 Orbit State Conversion for Near Singular SphericalAZEl State
Requirements	FR-1.3
PreConditions	BS-1

Steps	1. Load BS-1.
	2. Open the dialog box for DefaultSC.
	3. Change the state type to SphercialAZFPA
	4. Set the SphercialAZFPA state to the following values
	  1. RMAG = 7000.00000
	  2. RA = 0
	  3. DEC = -0.04094487109516581
	  4. VMAG = 10.63431419820442
	  5. AZI = 0
	  6. FPA = 0.02061675296478873
	5. Hit Apply.
	6. Change state type to Keplerian and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialAZFPA state to the Keplerian elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	7. Change state type to Modified Keplerian and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialAZFPA state to the Modified Keplerian elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	8. Change state type to Equinoctial and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialAZFPA state to the Equinoctial elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	9. Change RMAG to 0.00001 and click apply.
	10. Repeat steps 5, 6, and 7.
	11. Change RMAG to 70000.
	12. Change VMAG to 1e-14.
	13. Repeat steps 5, 6, and 7.
	14. Change state type to SphericalRADEC and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialAZFPA state to the SphericalRADEC state so conversion was aborted. The Right Ascension and Declination of velocity are undefined."

Expected Results	Test results are described above.

Table 7.19: STC-20 Orbit State Conversion for Near Singular SphericalAZEl state

Name	STC-20 Orbit GUI conversion for near singular SphericalRADEC state
Requirements	FR-1.3
PreConditions	BS-1

Steps	1. Load BS-1.
	2. Open the dialog box for DefaultSC.
	3. Change the state type to SphercialRADEC.
	4. Set the SphercialRADec state to the following values:
	  1. RMAG = 7000.00000
	  2. RA = 0
	  3. DEC = -0.04094487109516581
	  4. VMAG = 10.63431419820442
	  5. RAV = 0
	  6. DECV = -0.0203281181043372
	5. Hit Apply.
	6. Change state type to Keplerian and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialRADEC state to the Keplerian elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	7. Change state type to Modified Keplerian and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialRADEC state to the Modified Keplerian elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	8. Change state type to Equinoctial and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialRADEC state to the Equinoctial elements so conversion was aborted. The radius of periapsis must be greater than 1 meter."
	9. Change RMAG to 0.00001 and click apply.
	10. Repeat steps 5, 6, and 7.
	11. Change RMAG to 70000.
	12. Change VMAG to 1e-14.
	13. Repeat steps 5, 6, and 7.
	14. Change state type to SphericalRADEC and ensure that the following error is thrown: "Warning: A nearly singular conic section was encountered while converting from the SphercialRADEC state to the SphericalAZEL state so conversion was aborted. The Azimuth and Flight Path Angle are undefined."

Expected Results	Test results are described above.

Table 7.20: STC-21 Orbit State Conversion for Nearly Singular SphericalRADEC State

<table>
<tr><th>Name</th><th>STC-23 Epoch Conversion in the Spacecraft Orbit Dialog Box</th></tr>
<tr><td>Requirements</td><td>FRR-2.3</td></tr>
<tr><td>Summary</td><td>This test case represents $n(n-1)$ tests where n is the number of epoch formats supported as input types in GMAT. Each test case is designated a unique number. For example, STC-23.32 test GUI conversion from A1Gregorian to TAIModJulian. The procedures described below must be performed for each test case in the table below.</td></tr>
<tr><td>PreConditions</td><td>To run this test you need to load BS-1 and have data defined in TD-2 available.</td></tr>
<tr><td>Steps</td><td>1. Select subtest number. (STC-23.32, for example)
2. Create a new spacecraft.
3. Change the Epoch Format to the format defined in the first column of the row containing the test case ID. (A1Gregorian, for STC-23.32)
4. Enter the epoch in the Define Format from TD-2.
5. Change the Epoch Format to the format defined in the first row of the column containing the test case Id. (TAIModJulian for STC-23.32)
6. Verify that the new epoch exactly matches the value for that format given in TD-2.

<table>
<tr><th></th><th>UTCGregorian</th><th>UTCModJulian</th><th>TAIGregorian</th><th>TAIModJulian</th><th>A1Gregorian</th><th>A1ModJulian</th><th>TTGregorian</th><th>TTModJulian</th></tr>
<tr><td>UTCGregorian</td><td>N/A</td><td>23.1</td><td>23.2</td><td>23.3</td><td>23.4</td><td>23.5</td><td>23.6</td><td>23.7</td></tr>
<tr><td>UTCModJulian</td><td>23.8</td><td>N/A</td><td>23.9</td><td>23.10</td><td>23.11</td><td>23.12</td><td>23.13</td><td>23.14</td></tr>
<tr><td>TAIGregorian</td><td>23.15</td><td>23.16</td><td>N/A</td><td>23.17</td><td>23.18</td><td>23.19</td><td>23.20</td><td>23.21</td></tr>
<tr><td>TAIModJulian</td><td>23.22</td><td>23.23</td><td>23.24</td><td>N/A</td><td>23.25</td><td>23.26</td><td>23.27</td><td>23.28</td></tr>
<tr><td>A1Gregorian</td><td>23.29</td><td>23.30</td><td>23.31</td><td>23.32</td><td>N/A</td><td>23.33</td><td>23.34</td><td>23.35</td></tr>
<tr><td>A1ModJulian</td><td>23.36</td><td>23.37</td><td>23.38</td><td>23.39</td><td>23.40</td><td>N/A</td><td>23.41</td><td>23.42</td></tr>
<tr><td>TTGregorian</td><td>23.43</td><td>23.44</td><td>23.44</td><td>23.45</td><td>23.46</td><td>23.47</td><td>N/A</td><td>23.48</td></tr>
<tr><td>TTModJulian</td><td>23.49</td><td>23.50</td><td>23.51</td><td>23.52</td><td>23.53</td><td>23.54</td><td>23.55</td><td>N/A</td></tr>
</table>
</td></tr>
<tr><td>Expected Results</td><td>The expected numeric results are described above and in TD-2.</td></tr>
</table>

Table 7.21: STC-23 Epoch Conversion in the Spacecraft Orbit Dialog Box

<table>
<tr><th>Name</th><th>STC-24 State Conversion in the Spacecraft Orbit Dialog Box</th></tr>
<tr><td>Requirements</td><td>FRR-2.3</td></tr>
<tr><td>Summary</td><td>This test case represents $n(n-1)$ tests where n is the number of state representations supported as input types in GMAT. Each test case is designated a unique number. For example, STC-24.17 tests GUI conversion from SphericalRADEC to Keplerian elements. The procedures described below must be performed for each test case in the table below.</td></tr>
<tr><td>PreConditions</td><td>To run this test you need to load BS-1 and have data defined in TD-1 available.</td></tr>
<tr><td>Steps</td><td>
1. Select subtest number. (STC-24.17, for example)
2. Create a new spacecraft.
3. Change the Epoch Format to the format defined in the first column of the row containing the test case ID. (SphericalRADEC, for STC-24.17)
4. Enter the epoch in the Define Format from TD-1.
5. Change the Epoch Format to the format defined in the first row of the column containing the test case Id. (Keplerian for STC-24.17)
6. Verify that the new state matches the value for that format given in TD-1 to 14 significant figures.
<table>
<tr><th></th><th>Cartesian</th><th>Keplerian</th><th>TAIGregorian</th><th>SphericalRADEC</th><th>SphericalAZFPA</th><th>Equinoctial</th></tr>
<tr><td>Cartesian</td><td>N/A</td><td>24.1</td><td>24.2</td><td>24.3</td><td>24.4</td><td>24.5</td></tr>
<tr><td>Keplerian</td><td>24.6</td><td>N/A</td><td>24.7</td><td>24.8</td><td>24.9</td><td>24.10</td></tr>
<tr><td>TAIGregorian</td><td>25.11</td><td>24.12</td><td>N/A</td><td>24.13</td><td>24.14</td><td>24.15</td></tr>
<tr><td>SphericalRADEC</td><td>24.16</td><td>24.17</td><td>24.18</td><td>N/A</td><td>24.19</td><td>24.20</td></tr>
<tr><td>SphericalAZFPA</td><td>24.21</td><td>24.22</td><td>24.23</td><td>24.24</td><td>N/A</td><td>24.25</td></tr>
<tr><td>Equinoctial</td><td>24.26</td><td>24.27</td><td>24.28</td><td>24.29</td><td>24.30</td><td>N/A</td></tr>
</table>
</td></tr>
<tr><td>Expected Results</td><td>The expected numeric results are described above and in TD-1.</td></tr>
</table>

Table 7.22: STC-24 State Conversion in the Spacecraft Orbit Dialog Box

<table>
<tr><td>Name</td><td>STC-25 Coordinate System Conversion in the Spacecraft Orbit Dialog Box</td></tr>
<tr><td>Requirements</td><td>FRR-2.3</td></tr>
<tr><td>Summary</td><td>This test case represents $n(n-1)$ tests where n is the number of epoch formats supported as input types in GMAT. Each test case is designated a unique number. For example, STC-25.32 test GUI conversion from A1Gregorian to TAIModJulian. The procedures described below must be performed for each test case in the table below.</td></tr>
<tr><td>PreConditions</td><td>To run this test you need to load BS-1 and have data defined in TD-2 available.</td></tr>
<tr><td>Steps</td><td>
1. Select subtest number. (STC-25.32, for example)
2. Create a new spacecraft.
3. Change the Epoch Format to the format defined in the first column of the row containing the test case ID. (A1Gregorian, for STC-25.32)
4. Enter the epoch in the Define Format from TD-2.
5. Change the Epoch Format to the format defined in the first row of the column containing the test case Id. (TAIModJulian for STC-25.32)
6. Verify that the new epoch exactly matches the value for that format given in TD-2.
<table>
<tr><th></th><th>EarthMJ2000Eq</th><th>EarthMJ2000Ec</th><th>EarthFixed</th><th>LunaFixed</th><th>EarthMoonRot</th><th>SunMJ2000Ec</th><th>CS_ESL2</th><th>CS_SSBary</th><th>PhobosFixed</th></tr>
<tr><td>EarthMJ2000Eq</td><td>N/A</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>56</td></tr>
<tr><td>EarthMJ2000Ec</td><td>8</td><td>N/A</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>57</td></tr>
<tr><td>EarthFixed</td><td>15</td><td>16</td><td>N/A</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>58</td></tr>
<tr><td>LunaFixed</td><td>22</td><td>23</td><td>24</td><td>N/A</td><td>25</td><td>26</td><td>27</td><td>28</td><td>59</td></tr>
<tr><td>EarthMoonRot</td><td>29</td><td>30</td><td>31</td><td>32</td><td>N/A</td><td>33</td><td>34</td><td>35</td><td>60</td></tr>
<tr><td>SunMJ2000Ec</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td><td>N/A</td><td>41</td><td>42</td><td>61</td></tr>
<tr><td>CS_ESL2</td><td>43</td><td>44</td><td>44</td><td>45</td><td>46</td><td>47</td><td>N/A</td><td>48</td><td>62</td></tr>
<tr><td>CS_SSBary</td><td>49</td><td>50</td><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>N/A</td><td>63</td></tr>
<tr><td>PhobosFixed</td><td>49</td><td>50</td><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>N/A</td><td>64</td></tr>
</table>
</td></tr>
<tr><td>Expected Results</td><td>The expected numeric results are described above and in TD-2.</td></tr>
</table>

Table 7.23: STC-25 Coordinate System Conversion in the Spacecraft Orbit Dialog Box

Name	STC-26 Attitude Conversion in the Spacecraft Attitude Dialog Box
Requirements	FRR-3.3
Summary	This test case represents $n(n-1)$ tests where n is the number of attitude representations supported as input types in GMAT. Each test case is designated a unique number. For example, STC-26.32 tests conversion from a 231 to a 232 Euler angle sequence. The procedures described below must be performed for each test case in the table below.
PreConditions	To run this test you need to load BS-1 and have data defined in TD-8 available.
Steps	1. Select subtest number. (STC-26.32, for example) 2. Create a new spacecraft. 3. Open the dialog box for the new spacecraft. 4. Click on the Attitude tab. 5. Change the AttitudeStateType to the format defined in the first column of the row containing the test case ID. (Euler Angles, for STC-26.32) 6. If the AttitudeStateType is EulerAngles, change the EulerAngleSequence to the sequence defined in the first column of the row containing the test case ID. (231, for STC-26.32) 7. Enter the attitude state for the test ID using the data from TD-8. 8. Hit Apply. 9. Change the AttitudeStateType to the format defined in the first row of the column containing the test case Id. (Euler Angles, for STC-26.32) 10. If the AttitudeStateType is EulerAngles, Change the EulerAngleSequence to the format defined in the first row of the column containing the test case Id. (232, for STC-26.32) 11. Verify that the new epoch exactly matches the value for that format given in TD-8. 12. Compare the new Euler Angles to those in TD-8 for the new attitude representation. The values should agree to at least 13 significant figures.
Expected Results	The expected numeric results are described above and in TD-8.

	q	123	231	312	132	321	213	121	232	313	131	323	212
q	X	1	2	3	4	5	6	7	8	9	10	11	12
123	13	X	14	15	16	17	18	19	20	21	22	23	24
231	25	26	X	27	28	29	30	31	32	33	34	35	36
312	37	38	39	X	40	41	42	43	44	45	46	47	48
132	49	50	51	52	X	53	54	55	56	57	58	59	60
321	61	62	63	64	65	X	66	67	68	69	70	71	72
213	73	74	75	76	77	78	X	79	80	81	82	83	84
121	85	86	87	88	89	90	91	X	92	93	94	95	96
232	97	98	99	100	101	102	103	104	X	105	106	107	108
313	109	110	111	112	113	114	115	116	X	117	118	119	120
131	121	122	123	124	125	126	127	128	129	X	130	131	132
323	133	134	135	136	137	138	139	140	141	142	143	X	144
212	145	146	147	148	149	150	151	152	153	154	155	156	X

Table 7.24: STC-26 Attitude Conversion in the Spacecraft Attitude Dialog Box

Name	Attitude GUI Behavior When Entering Zero Quaternion
Requirements	FR-3.1
PreConditions	BS-1
Data	1. Load BS-1. 2. Open the dialog box for DefaultSC. 3. Click on the attitude tab. 4. Set all values of the quaternion to zero. 5. Click Ok.
Expected Results	The following warning is displayed: The magnitude of a quaternion must be greater than 1e-10.

Table 7.25: STC-17 Attitude GUI Behavior When Entering Zero Quaternion

7.3 Differential Corrector

Name	TC-1 Differential Corrector Dialog Box Range Tests - Disallowed Values
Requirements	FR-19
Summary	This case verifies the Differential Corrector dialog box rejects disallowed data.
PreConditions	BS-1
Data	1. Load BS-1. 2. In the solvers folder in the mission tree, right-click on the Boundary Value Solvers folder and add a Differential Corrector. 3. Open the dialog box for the new Differential Corrector. 4. In the Max Iterations field, enter -2, and hit Apply. Ensure the following error message is provided: The value of "-2" for field "Maximum Iterations" is not an allowed value. The allowed values are: [Integer Number > 0]. 5. In the Max Iterations field, enter DNE, and hit Apply. Ensure the following error message is provided: The value of "DNE" for field "Maximum Iterations" is not an allowed value. The allowed values are: [Integer Number > 0]. 6. In the Max Iterations field, enter 23.6, and hit Apply. Ensure the following error message is provided: The value of "23.6" for field "Maximum Iterations" is not an allowed value. The allowed values are: [Integer Number > 0].
Expected Results	Test results are described above.

Table 7.26: TC-1 Differential Corrector Dialog Box Range Tests - Disallowed Values

Name	TC-2 Differential Corrector Dialog Box Range Tests- Allowed Values
Requirements	FR-19
Summary	This case verifies the Differential Corrector accepts allowed data.
PreConditions	BS-1
Data	1. Load BS-1. 2. In the solvers folder in the mission tree, right-click on the Boundary Value Solvers folder and add a Differential Corrector. 3. Open the dialog box for the new Differential Corrector. 4. Set the Max Iterations to 56. 5. In the ReportFile field type `.\output\DCReport.txt` 6. Uncheck the ShowProgress box. 7. Set the DerivativeMethod drop-down menu to CentralDifference. 8. Set the ReportStyle drop-down menu to Verbose. 9. Click the Apply button. 10. Click the Show Script button.

Expected Results	`Create DifferentialCorrector DC1;`
	`GMAT DC1.ShowProgress = false;`
	`GMAT DC1.ReportStyle = 'Verbose';`
	`GMAT DC1.ReportFile = '.\output\DCData.txt';`
	`GMAT DC1.MaximumIterations = 56;`
	`GMAT DC1.DerivativeMethod = CentralDifference;`

Table 7.27: TC-2 Differential Corrector Dialog Box Range Tests- Allowed Values

7.4 Mission Tree

Name	TC-3 Adding and Deleting All Commands
Requirements	FR-XXX
Summary	This case tests adding and deleting all supported GMAT commands via the Mission Tree GUI. This test does not check for adding and deleting commands inside of branch commands.
PreConditions	BS-5

Data	1. Load BS-1.
	2. Click on the mission tree tab.
	3. Right-click on the Mission Tree folder, select Append, then BeginFiniteBurn
	4. Follow the steps in the line above for the following commands:
	  1. Maneuver
	  2. CallFunction
	  3. Optimize
	  4. Equation
	  5. Report
	  6. Save
	  7. Toggle
	  8. If
	  9. If/Else
	  10. BeginFiniteBurn
	  11. Propagate
	  12. For
	  13. While
	  14. PenUp
	  15. PenDown
	  16. RunEstimator
	  17. EndFiniteBurn
	  18. RunSimulator
	  19. ScriptEvent
	  20. Stop
	  21. Target
	5. Right-click on the Maneuver command and the top of the tree and select delete.
	6. Perform the same steps above and delete all commands in the order they were created.

Expected Results	All objects are created and deleted and no error messages or warnings should be thrown.)

Table 7.28: TC-3 TEST NAME TEXT

Chapter 8

Test Data

Name	TD-1 Equivalent State Representations in EarthMJ2000Eq
Description	This table contains equivalent states in all GMAT state representations that have a central body at the origin.
Source	This is data comes from GMAT and allows testing consistency between state conversions. We have numerical tests that verify that GMAT is correctly performing conversions correctly via the script when compared to truth data from STK. The data below is to ensure that the GUI state conversions agree with conversions in the script. (The test data assumes $\mu = 398600.4415$)
Data	1. Cartesian State 1. X = -2011.554639349956 2. Y = 7587.193672855249 3. Z = 1362.382029017782 4. VX = -7.694247416401868 5. VY = -0.9065479140190984 6. VZ = 0.4284953758282981 2. Keplerian State 1. SMA = 10000 2. ECC = 0.25 3. INC = 10.0 4. RAAN = 25.0 5. AOP = 35.0 6. TA = 45.0 7. MA = 27.24378911263291; 8. EA = 35.57751520702131; 3. Modified Keplerian 1. RadPer = 7500 2. RadApo = 12500 3. INC = 10.0 4. RAAN = 25.0 5. AOP = 35.0 6. TA = 45.0 4. Spherical RADec 1. RMAG = 7966.67714229061 2. RA = 104.8489182889519 3. DEC = 9.846551939834079 4. VMAG = 7.759309293508375 5. AZI = 88.24621654190652 6. FPA = 81.45684518510772 5. Spherical RADec 1. RMAG = 7966.67714229061 2. RA = 104.8489182889519 3. DEC = 9.846551939834079 4. VMAG = 7.759309293508375 5. AZI = -173.2803040524276 6. FPA = 3.165677683357204 6. Equinoctial 1. SMA = 10000 2. h = 0.2165063509461095 3. k = 0.125 4. p = 0.03697430690134294 5. q = 0.07929165703097431 6. MLONG = 87.2437891126329

Table 8.1: TD-1 Equivalent State Representations

Name	TD-2 Equivalent Epoch Representations
Description	This table contains equivalent epoch representations in all formats and systems supported as input types.
Source	Need to verify source.
Data	1. 04 Jul 2004 12:34:56.789 UTC 2. 23191.0242683912 UTC 3. 04 Jul 2004 12:35:28.789 TAI 4. 23191.02463876157 TAI 5. 04 Jul 2004 12:35:28.823 A1 6. 23191.02463915951 A1 7. 04 Jul 2004 12:36:00.973 TT 8. 23191.0250112616 TT

Table 8.2: TD-2 Equivalent Epoch Representations

Name	TD-3 Equivalent States in Different Coordinate Systems
Description	This table contains equivalent states in various coordinate systems. The epoch is 04 Feb 2001 11:59:28.000 UTC.
Source	This is data comes from GMAT and allows testing consistency between state conversions. We have numerical tests that verify that GMAT is correctly performing conversions correctly via the script when compared to truth data from STK. The data below is to ensure that the GUI state conversions agree with conversions in the script.

Data	1. EarthMJ2000Eq
	  1. 5071.298226925739
	  2. 7611.115643763225
	  3. 3591.57811088299
	  4. -5.443963856628132
	  5. 2.768170139549618
	  6. 1.993434604659487
	2. EarthMJ2000Eq
	  1. 5071.298226925739
	  2. 8411.709801736053
	  3. 267.6805571735547
	  4. -5.443963856628132
	  5. 3.332689195355913
	  6. 0.7278256465019841
	3. EarthFixed System —-
	  1. -1863.0575794316
	  2. 8953.983671172193
	  3. 3591.903847242492
	  4. -5.139752276668541
	  5. -1.799622038729856
	  6. 1.992983974458626
	4. LunaFixed System —-
	  1. 356291.5164882602
	  2. -33974.36961780395
	  3. 16667.54936624816
	  4. -3.267914664134139
	  5. -5.374404304814976
	  6. 0.7247736501205496
	5. EarthMoonRot System
	  1. 8968.872502970022
	  2. -3907.723350348471
	  3. 914.1141166948707
	  4. 2.581713440043834
	  5. 5.842794343296872
	  6. 0.3253861743578399
	6. SunMJ2000Ec System
	  1. 125912803.1220472
	  2. 163450023.4650465
	  3. 66337826.42493016
	  4. -33.57498440092737
	  5. 0.0008503562866066794
	  6. 1.493151380860197
	7. CS_ESL2 System
	  1. -560121.5615799141
	  2. 284125.8138175178
	  3. 13140.72166463266
	  4. 5.466911578720965
	  5. 0.7694483179080936
	  6. 0.6355953786263187
	8. CS_SSBary System
	  1. 147428584.0468952
	  2. -9578.340467568873
	  3. 266.8937442606475
	  4. 6.68279484246637
	  5. 1.370672075811688
	  6. 0.726953785084683

Name	TD-4 Equivalent State Representations for a Singular Conic Section
Description	This table contains equivalent states in all GMAT state representations that have a central body at the origin.
Source	STK. The data below is to ensure that the GUI state conversions agree with conversions in the script. (The test data assumes $\mu = 398600.4415$)
Data	1. Cartesian State 1. X = 7000 2. Y = 7000 3. Z = 7000 4. VX = -4.04145188432738 5. VY = -4.04145188432738 6. VZ = -4.04145188432738 2. Keplerian State: Undefined (e = 1); 3. Modified Keplerian: Undefined (e = 1); 4. Spherical RADec 1. RMAG = 12124.355652982142 2. RA = 45 3. DEC = 35.2643896827546470 4. VMAG = 6.999999999999998 5. AZI = 0 6. FPA = 180 5. Spherical RADec 1. RMAG = 12124.355652982142 2. RA = 45 3. DEC = 35.2643896827546470 4. VMAG = 6.999999999999998 5. DECV = 225 6. RAV = 35.264389682754661 6. Equinoctial: Undefined (e = 1);

Table 8.4: TD-4 Equivalent State Representations for a Singular Conic Section

Name	TD-5 Equivalent State Representations for a Circular, Equatorial Orbit
Description	This table contains equivalent states in all GMAT state representations that have a central body at the origin.
Source	Hand calculations based on Math spec for all except Equinoctial which is from STK. (The test data assumes $\mu = 398600.4415$)
Data	1. Cartesian State 1. X = 4949.747468305833 2. Y = 4949.747468305833 3. Z = 0.0 4. VX = -5.335865450622125 5. VY = 5.335865450622125 6. VZ = 0 2. Keplerian State 1. SMA = 7000 2. ECC = 0.0 3. INC = 0.0 4. RAAN = 0.0 5. AOP = 0.0 6. TA = 45.0 3. Modified Keplerian 1. RadPer = 7000 2. RadApo = 7000 3. INC = 0.0 4. RAAN = 0.0 5. AOP = 0.0 6. TA = 45.0 4. Spherical RADec 1. RMAG = 7000 2. RA = 45.0 3. DEC = 0.0 4. VMAG = 7.5460532872678359 5. RAV = 135.00000000000000 6. DECV= 0.0 5. Spherical RADec 1. RMAG = 7000 2. RA = 45.0 3. DEC = 0.0 4. VMAG = 7.5460532872678359 5. AZI = 90 6. FPA = 90 6. Equinoctial 1. SMA = 7000 2. h = 0.0 3. k = 0.0 4. p = 0.0 5. q = 0.0 6. MLONG = 45.0

Table 8.5: TD-5 Equivalent State Representations for a Circular, Equatorial Orbit

Name	TD-6 Equivalent State Representations for a Circular, Inclined (retrograde) Orbit
Description	This table contains equivalent states in all GMAT state representations that have a central body at the origin.
Source	Hand calculations based on Math spec for all except Equinoctial which is from STK. (The test data assumes $\mu = 398600.4415$)
Data	1. Cartesian State 1. X = -5975.5752861126311 2. Y = 480.14719831222595 3. Z = -3416.4248371584213 4. VX = 3.8002690670377621 5. VY = 0.9160734111800478 6. VZ = -6.5182010133917370 2. Keplerian State 1. SMA = 6900 2. ECC = 0.0 3. INC = 98 4. RAAN = 0.0 5. AOP = 0.0 6. TA = 210.0 3. Modified Keplerian 1. RadPer = 6900 2. RadApo = 6900 3. INC = 98 4. RAAN = 0.0 5. AOP = 0.0 6. TA = 210.0 4. Spherical RADec 1. RMAG = 6900 2. RA = 175.4060606593105 3. DEC = -29.67858910292156 4. VMAG = 7.6005381340755180 5. RAV = 13.55286811093926 6. DECV= -59.04786932043024 5. Spherical RADec 1. RMAG = 6900 2. RA = 175.4060606593105 3. DEC = -29.67858910292156 4. VMAG = 7.6005381340755180 5. AZI = 189.2177489242794 6. FPA = 90 6. Equinoctial 1. SMA = 6900 2. h = 0.0 3. k = 0.0 4. p = 0.0 5. q = 1.1503684072210094 6. MLONG = 210.0

Table 8.6: TD-6 Equivalent State Representations for a Circular, Inclined (retrograde) Orbit

Name	Equivalent Attitude Representations
Description	This table contains equivalent states in various coordinate systems. The epoch is 04 Feb 2001 11:59:28.000 UTC.
Source	This is data comes from GMAT and allows testing consistency between state conversions. We have numerical tests that verify that GMAT is correctly performing conversions correctly via the script when compared to truth data from STK. The data below is to ensure that the GUI state conversions agree with conversions in the script.
Data	• Quaternion 1. 0.05431254465935684 2. 0.1536190745285137 3. 0.6870053865727263 4. 0.7081489435519108 • Euler Angles

Sequence	θ_1 (deg.)	θ_2 (deg.)	θ_3 (deg.)
123	-8.063660107792256	16.98949666527142	89.46981218935655
231	86.45739059790678	81.76589766113709	-69.50363702414091
312	86.99826598215141	16.73807760458099	8.584547140308942
132	80.12240308416057	73.0024763951768	88.26525116243468
321	89.48768989493576	8.21825906808993	16.91694843278077
213	17.14924429081384	-7.709559339371771	87.09931277680732
121	81.78141688239583	89.49295108504926	-73.00981782084581
232	16.7597858904212	87.12555469792072	7.719330705131926
313	114.6605285775403	18.75489666719195	-26.39703015347826
131	-8.218583117604172	89.49295108504926	16.99018217915419
323	24.66052857754036	18.75489666719195	63.60296984652174
212	-73.2402141095788	87.12555469792072	97.71933070513192

• Direction Cosine Matrix

0.008849557522123823	0.9896911631236142	-0.1429443491946786
-0.9563173917401783	0.05014749262536867	0.2879970056070911
0.2921963921524783	0.134151521118533	0.9469026548672567

Table 8.7: TD-8 Equivalent Attitude Representations

Chapter 9

Base States

Name	BS-1 The Default Mission
Summary	This base state configures GMAT to the default mission.
Description	See BS-1.script

Table 9.1: BS-1 The Default Mission

Name	BS-5 Create All Objects
Summary	This base state creates instantiations of all object types supported in GMAT. Several object types have more than one instantiation. For example, to have a meaningful formation, you need at least 2 spacecraft.
Description	See BS-5-CreateAllObjects.script

Table 9.2: BS-5 Create All Objects

Name	BS-6 Target Hohmann Transfer
Summary	This base state is the script that configures GMAT to target a Hohmann transfer from LEO to GEO.
Description	To load this base state load Ex_TargetHohmannTransfer.script from the test repository described in Ch.2.

Table 9.3: BS-6 Target Hohmann Transfer

Name	BS-7 Ace Station Keeping
Summary	This base state is the script that performs Earth-Sun L2 station keeping for ACE.
Description	To load this base state load Ex_ACEStationKeeping.script from the test repository described in Ch.2.

Table 9.4: BS-7 Ace Station Keeping

Name	BS-8 MMS Double Lunar Swingby
Summary	This base state is the script that performs a double lunar swingby for MMS.
Description	To load this base state load Ex_MMSDoubleLunarSwingby.script from the test repository described in Ch.2.

Table 9.5: BS-8 MMS Double Lunar Swingby

Name	BS-9 MMS Double Lunar Swingby
Summary	This base state is the script that performs uses a GMAT function to propagate an MMS formation and calculate the quality factor.
Description	To load this base state load Ex_GmatFunction.script from the test repository described in Ch.2.

Table 9.6: BS-9 MMS Double Lunar Swingby

Part IV

Checklists

Chapter 10

Resources

10.1 Differential Corrector

```
$Id: DifferentialCorrectorPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the Differential Corrector panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create DifferentialCorrector Resource     [ ] Pass  [ ] Fail  Bug# ______

   Open DifferentialCorrector Resource       [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Max Iterations Text Field                 [ ] Pass  [ ] Fail  Bug# ______
```

```
   Targeter Text File Text Field            [ ] Pass  [ ] Fail  Bug# ______

   Show Progress Check Box                  [ ] Pass  [ ] Fail  Bug# ______

   Use Central Differences Check Box        [ ] Pass  [ ] Fail  Bug# ______

   Report Style Combo Box                   [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail
```

Bugs Reported:

Notes:

10.2 Impulsive Burn

$Id: ImpulsiveBurnPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____ Date: _____ Build: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Impulsive Burn panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add ImpulsiveBurn Resource [ ] Pass [ ] Fail Bug# _____

   Open ImpulsiveBurn Resource [ ] Pass [ ] Fail Bug# _____

   Close ImpulsiveBurn Resource [ ] Pass [ ] Fail Bug# _____

   Delete ImpulsiveBurn Resource [ ] Pass [ ] Fail Bug# _____

   Clone ImpulsiveBurn Resource [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

   Coordinate System [ ] Pass [ ] Fail Bug# _____

   Origin Combo Box [ ] Pass [ ] Fail Bug# _____

```
   Axes Combo Box                          [ ] Pass  [ ] Fail  Bug# ______

   Thrust Vector Element 1 Text Field      [ ] Pass  [ ] Fail  Bug# ______

   Thrust Vector Element 2 Text Field      [ ] Pass  [ ] Fail  Bug# ______

   Thrust Vector Element 3 Text Field      [ ] Pass  [ ] Fail  Bug# ______

   Decrement Mass Button                   [ ] Pass  [ ] Fail  Bug# ______

   Tank Combo Box                          [ ] Pass  [ ] Fail  Bug# ______

   Isp Text Field                          [ ] Pass  [ ] Fail  Bug# ______

   Gravitational Accel Field               [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete           [ ] Pass  [ ] Fail  Bug# ______

   Show Script                             [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                      [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                 [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                [ ] Pass  [ ] Fail  Bug# ______

   Close                                   [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)               [ ] Pass  [ ] Fail  Bug# ______

Summary:
```

```
   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.3 Finite Burn

```
$Id: FiniteBurnPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: __________________

Description:

This test validates the functionality of the Finite Burn panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Finite Burn Resource                [ ] Pass  [ ] Fail  Bug# ______

   Open Finite Burn Resource               [ ] Pass  [ ] Fail  Bug# ______

   Close Finite Burn Resource              [ ] Pass  [ ] Fail  Bug# ______

   Delete Finite Burn Resource             [ ] Pass  [ ] Fail  Bug# ______

   Clone Finite Burn Resource              [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.
```

```
   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Use Thruster Combo Box                    [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)                 [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:
```

Notes:

10.4 Formation

$Id: FormationPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____ Date: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Formation panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create Spacecraft Resource [ ] Pass [ ] Fail Bug# _____

   Create Spacecraft Resource [ ] Pass [ ] Fail Bug# _____

   Create Formation Resource [ ] Pass [ ] Fail Bug# _____

   Open Formation Resource [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

   Space Objects List [ ] Pass [ ] Fail Bug# _____

   Space Objects in Formation List [ ] Pass [ ] Fail Bug# _____

   -> Selection Button [ ] Pass [ ] Fail Bug# _____

   <- Selection Button [ ] Pass [ ] Fail Bug# _____

```
   <= Selection Button                    [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete          [ ] Pass  [ ] Fail  Bug# ______

   Show Script                            [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                     [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                 [ ] Pass  [ ] Fail  Bug# ______

   Apply                                  [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                              [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                [ ] Pass  [ ] Fail  Bug# ______

   Minimize                               [ ] Pass  [ ] Fail  Bug# ______

   Maximize                               [ ] Pass  [ ] Fail  Bug# ______

   Close                                  [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)              [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                       [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.5 Propagator

```
$Id: PropagatorPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Propagator panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Propagator Resource                  [ ] Pass  [ ] Fail  Bug# ______

   Open Propagator Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Close LibrationPoint Resource            [ ] Pass  [ ] Fail  Bug# ______

   Clone LibrationPoint Resource            [ ] Pass  [ ] Fail  Bug# ______

   Delete LibrationPoint Resource           [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Integrator Type Combo Box                [ ] Pass  [ ] Fail  Bug# ______

   Initial Step Size Text Field             [ ] Pass  [ ] Fail  Bug# ______

   Accuracy Text Field                      [ ] Pass  [ ] Fail  Bug# ______

   Minimum Step Size Text Field             [ ] Pass  [ ] Fail  Bug# ______

   Maximum Step Size Text Field             [ ] Pass  [ ] Fail  Bug# ______
```

```
Maximum Step Attempts Text Field          [ ] Pass  [ ] Fail  Bug# ______

Minimum Integration Error Text Field      [ ] Pass  [ ] Fail  Bug# ______

Nominal Integration Error Text Field      [ ] Pass  [ ] Fail  Bug# ______

Error Control Combo Box                   [ ] Pass  [ ] Fail  Bug# ______

Central Body Combo Box                    [ ] Pass  [ ] Fail  Bug# ______

Primary Bodies Combo Box                  [ ] Pass  [ ] Fail  Bug# ______

Primary Bodies Action Button
   Available Bodies List                  [ ] Pass  [ ] Fail  Bug# ______
   Bodies Selected List                   [ ] Pass  [ ] Fail  Bug# ______
   -> (Add) Selection Button              [ ] Pass  [ ] Fail  Bug# ______
   <- (Remove) Selection Button           [ ] Pass  [ ] Fail  Bug# ______
   <= (Remove All) Selection Button       [ ] Pass  [ ] Fail  Bug# ______
   Orbit Color Selection Button           [ ] Pass  [ ] Fail  Bug# ______
   Cancel Button                          [ ] Pass  [ ] Fail  Bug# ______
   OK Button                              [ ] Pass  [ ] Fail  Bug# ______
   Help [DEFERRED]

Primary Bodies Text List                  [ ] Pass  [ ] Fail  Bug# ______

Gravity Field Type Combo Box              [ ] Pass  [ ] Fail  Bug# ______

Gravity Model File Text Field             [ ] Pass  [ ] Fail  Bug# ______

Gravity Field Degree Text Field           [ ] Pass  [ ] Fail  Bug# ______

Gravity Field Order Text Field            [ ] Pass  [ ] Fail  Bug# ______

Gravity Model File Select Action Button   [ ] Pass  [ ] Fail  Bug# ______

Atmosphere Model Type Combo Box           [ ] Pass  [ ] Fail  Bug# ______

Atmosphere Model Setup Action Button      [ ] Pass  [ ] Fail  Bug# ______
   User Input Radio Button                [ ] Pass  [ ] Fail  Bug# ______
   Solar Flux Text Field                  [ ] Pass  [ ] Fail  Bug# ______
   Average Solar Flux Text Field          [ ] Pass  [ ] Fail  Bug# ______
   Geomagnetic Index (Kp) Text Field      [ ] Pass  [ ] Fail  Bug# ______
   File Input Radio Button                [ ] Pass  [ ] Fail  Bug# ______
   File Name Text Field                   [ ] Pass  [ ] Fail  Bug# ______
   File Selection Button                  [ ] Pass  [ ] Fail  Bug# ______
   Cancel Button                          [ ] Pass  [ ] Fail  Bug# ______
   OK Button                              [ ] Pass  [ ] Fail  Bug# ______
   Help [DEFERRED]

Magnetic Field Type Combo Box             [ ] Pass  [ ] Fail  Bug# ______

Magnetic Field Degree Text Field          [ ] Pass  [ ] Fail  Bug# ______

Magnetic Field Order Text Field           [ ] Pass  [ ] Fail  Bug# ______

Magn. Field File Select Action Button     [ ] Pass  [ ] Fail  Bug# ______
```

```
   Point Masses Select Action Button
      Available Bodies List                [ ] Pass  [ ] Fail  Bug# ______
      Bodies Selected List                 [ ] Pass  [ ] Fail  Bug# ______
      -> (Add) Selection Button            [ ] Pass  [ ] Fail  Bug# ______
      <- (Remove) Selection Button         [ ] Pass  [ ] Fail  Bug# ______
      <= (Remove All) Selection Button     [ ] Pass  [ ] Fail  Bug# ______
      Orbit Color Selection Button         [ ] Pass  [ ] Fail  Bug# ______
      Cancel Button                        [ ] Pass  [ ] Fail  Bug# ______
      OK Button                            [ ] Pass  [ ] Fail  Bug# ______
      Help [DEFERRED]

   Point Masses Text List                  [ ] Pass  [ ] Fail  Bug# ______

   Solar Radiation Pressure Check Box      [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete           [ ] Pass  [ ] Fail  Bug# ______

   Show Script                             [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                      [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                 [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                [ ] Pass  [ ] Fail  Bug# ______

   Close                                   [ ] Pass  [ ] Fail  Bug# ______


7. Evaluate rename functionality.

   Rename (on resource tree)               [ ] Pass  [ ] Fail  Bug# ______


Summary:
```

```
   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.6 Propagator

```
$Id: PropagatorPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: __________________

Description:

This test validates the functionality of the Propagator panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Propagator Resource                  [ ] Pass  [ ] Fail  Bug# ______

   Open Propagator Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Close LibrationPoint Resource            [ ] Pass  [ ] Fail  Bug# ______

   Clone LibrationPoint Resource            [ ] Pass  [ ] Fail  Bug# ______

   Delete LibrationPoint Resource           [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______
```

3. Evaluate the individual panel elements.

Integrator Type Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Initial Step Size Text Field	[ ] Pass	[ ] Fail	Bug# _____
Accuracy Text Field	[ ] Pass	[ ] Fail	Bug# _____
Minimum Step Size Text Field	[ ] Pass	[ ] Fail	Bug# _____
Maximum Step Size Text Field	[ ] Pass	[ ] Fail	Bug# _____
Maximum Step Attempts Text Field	[ ] Pass	[ ] Fail	Bug# _____
Minimum Integration Error Text Field	[ ] Pass	[ ] Fail	Bug# _____
Nominal Integration Error Text Field	[ ] Pass	[ ] Fail	Bug# _____
Error Control Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Central Body Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Primary Bodies Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Primary Bodies Action Button			
Available Bodies List	[ ] Pass	[ ] Fail	Bug# _____
Bodies Selected List	[ ] Pass	[ ] Fail	Bug# _____
-> (Add) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
<- (Remove) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
<= (Remove All) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
Orbit Color Selection Button	[ ] Pass	[ ] Fail	Bug# _____
Cancel Button	[ ] Pass	[ ] Fail	Bug# _____
OK Button	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Primary Bodies Text List	[ ] Pass	[ ] Fail	Bug# _____
Gravity Field Type Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Gravity Model File Text Field	[ ] Pass	[ ] Fail	Bug# _____
Gravity Field Degree Text Field	[ ] Pass	[ ] Fail	Bug# _____
Gravity Field Order Text Field	[ ] Pass	[ ] Fail	Bug# _____
Gravity Model File Select Action Button	[ ] Pass	[ ] Fail	Bug# _____
Atmosphere Model Type Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Atmosphere Model Setup Action Button	[ ] Pass	[ ] Fail	Bug# _____
User Input Radio Button	[ ] Pass	[ ] Fail	Bug# _____
Solar Flux Text Field	[ ] Pass	[ ] Fail	Bug# _____
Average Solar Flux Text Field	[ ] Pass	[ ] Fail	Bug# _____
Geomagnetic Index (Kp) Text Field	[ ] Pass	[ ] Fail	Bug# _____
File Input Radio Button	[ ] Pass	[ ] Fail	Bug# _____

```
      File Name Text Field                  [ ] Pass  [ ] Fail  Bug# ______
      File Selection Button                 [ ] Pass  [ ] Fail  Bug# ______
      Cancel Button                         [ ] Pass  [ ] Fail  Bug# ______
      OK Button                             [ ] Pass  [ ] Fail  Bug# ______
      Help [DEFERRED]

   Magnetic Field Type Combo Box            [ ] Pass  [ ] Fail  Bug# ______

   Magnetic Field Degree Text Field         [ ] Pass  [ ] Fail  Bug# ______

   Magnetic Field Order Text Field          [ ] Pass  [ ] Fail  Bug# ______

   Magn. Field File Select Action Button    [ ] Pass  [ ] Fail  Bug# ______

   Point Masses Select Action Button
      Available Bodies List                 [ ] Pass  [ ] Fail  Bug# ______
      Bodies Selected List                  [ ] Pass  [ ] Fail  Bug# ______
      -> (Add) Selection Button             [ ] Pass  [ ] Fail  Bug# ______
      <- (Remove) Selection Button          [ ] Pass  [ ] Fail  Bug# ______
      <= (Remove All) Selection Button      [ ] Pass  [ ] Fail  Bug# ______
      Orbit Color Selection Button          [ ] Pass  [ ] Fail  Bug# ______
      Cancel Button                         [ ] Pass  [ ] Fail  Bug# ______
      OK Button                             [ ] Pass  [ ] Fail  Bug# ______
      Help [DEFERRED]

   Point Masses Text List                   [ ] Pass  [ ] Fail  Bug# ______

   Solar Radiation Pressure Check Box       [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______
```

```
   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)                 [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.7 Coordinate System

```
$Id: CoordinateSystemPanel.txt,v 1.4 2006/11/22 21:32:34 dconway Exp $

Tester: _______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _______________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Coordinate System panel (for
non-default coordinate systems).
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.
```

```
   Create Coordinate System Resource        [ ] Pass  [ ] Fail  Bug# ______

   Open Coordinate System Resource          [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.

   Origin Combo Box                         [ ] Pass  [ ] Fail  Bug# ______

   Axes Type Combo Box                      [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

   a. Select Object Referenced Axes Type:
      Primary Body Combo Box                [ ] Pass  [ ] Fail  Bug# ______
      Secondary Body Combo Box              [ ] Pass  [ ] Fail  Bug# ______
      X Combo Box                           [ ] Pass  [ ] Fail  Bug# ______
      Y Combo Box                           [ ] Pass  [ ] Fail  Bug# ______
      Z Combo Box                           [ ] Pass  [ ] Fail  Bug# ______

   b. Select TOEEq/MOEEq:
      Epoch Format Combo Box                [ ] Pass  [ ] Fail  Bug# ______
      Epoch Text Field                      [ ] Pass  [ ] Fail  Bug# ______

   c. Select BodyFixed/Equator/GSM/
      TODEc/TODEq/TOEEc/TOEEq:
      Update Interval Text Field            [ ] Pass  [ ] Fail  Bug# ______


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______
```

```
   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)                 [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: AddCoordSystemDialog.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________     Date: _________________________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _______________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Create Coordinate System Dialog.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Open Coord. System Create Dialog          [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.
```

```
   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.

   Coordinate System Name Text Field         [ ] Pass  [ ] Fail  Bug# ______

   Origin Combo Box                          [ ] Pass  [ ] Fail  Bug# ______

   Axes Type Combo Box                       [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

    a. Select Object Referenced Axes Type:
      Primary Body Combo Box                 [ ] Pass  [ ] Fail  Bug# ______
      Secondary Body Combo Box               [ ] Pass  [ ] Fail  Bug# ______
      X Combo Box                            [ ] Pass  [ ] Fail  Bug# ______
      Y Combo Box                            [ ] Pass  [ ] Fail  Bug# ______
      Z Combo Box                            [ ] Pass  [ ] Fail  Bug# ______

   b. Select TOEEq/MOEEq:
      Epoch Format Combo Box                 [ ] Pass  [ ] Fail  Bug# ______
      Epoch Text Field                       [ ] Pass  [ ] Fail  Bug# ______

   c. Select BodyFixed/Equator/GSM/
      TODEc/TODEq/TOEEc/TOEEq:
      Update Interval Text Field             [ ] Pass  [ ] Fail  Bug# ______


5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script [N/A]


6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______
```

```
7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.8 Array

```
$Id: ArrayPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _______________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Array panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create Array Resource                    [ ] Pass  [ ] Fail  Bug# ______

   Open Array Resource                      [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.
```

```
   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Name Text Field   (read-only)             [ ] Pass  [ ] Fail  Bug# ______

   Row Text Field   (read-only)                [ ] Pass  [ ] Fail  Bug# ______

   Column Text Field (read-only)             [ ] Pass  [ ] Fail  Bug# ______

   Row Combo Box                             [ ] Pass  [ ] Fail  Bug# ______

   Column Combo Box                          [ ] Pass  [ ] Fail  Bug# ______

   Value Text Field                          [ ] Pass  [ ] Fail  Bug# ______

   Update Button                             [ ] Pass  [ ] Fail  Bug# ______

   Values Grid (Text Fields)                 [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.
```

```
   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.9 Barycenter

```
$Id: BarycenterPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: ________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the Barycenter panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Barycenter Resource               [ ] Pass  [ ] Fail  Bug# ______

   Open Barycenter Resource              [ ] Pass  [ ] Fail  Bug# ______

   Close Barycenter Resource             [ ] Pass  [ ] Fail  Bug# ______

   Clone Barycenter Resource             [ ] Pass  [ ] Fail  Bug# ______
```

```
   Delete Barycenter Resource            [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                      [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Available Bodies List                 [ ] Pass  [ ] Fail  Bug# ______

   Bodies Selected List                  [ ] Pass  [ ] Fail  Bug# ______

   --> (Add) Action Button               [ ] Pass  [ ] Fail  Bug# ______

   <-- (Remove) Selection Button         [ ] Pass  [ ] Fail  Bug# ______

   < = (Remove All) Selection Button     [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete         [ ] Pass  [ ] Fail  Bug# ______

   Show Script                           [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                    [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                [ ] Pass  [ ] Fail  Bug# ______

   Apply                                 [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                             [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                               [ ] Pass  [ ] Fail  Bug# ______

   Minimize                              [ ] Pass  [ ] Fail  Bug# ______

   Maximize                              [ ] Pass  [ ] Fail  Bug# ______

   Close                                 [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)             [ ] Pass  [ ] Fail  Bug# ______
```

```
Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.10 Variable

```
$Id: CreateVariableDialog.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Create Variable dialog, which
appears via other panels (e.g. ReportFile, XYPlotPanel)
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create other panel Resource              [ ] Pass  [ ] Fail  Bug# ______

   Open other panel Resource                [ ] Pass  [ ] Fail  Bug# ______

   Open Create Action Button                [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.
```

```
   Panel Aesthetics                        [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Variable Name Text Field                [ ] Pass  [ ] Fail  Bug# ______

   Variable Expression Text Field          [ ] Pass  [ ] Fail  Bug# ______

   Paste Property Button                   [ ] Pass  [ ] Fail  Bug# ______

   Paste Variables Button                  [ ] Pass  [ ] Fail  Bug# ______

   Create Object Button                    [ ] Pass  [ ] Fail  Bug# ______

   Color Selection Button                  [ ] Pass  [ ] Fail  Bug# ______

   String Name Text Field                  [ ] Pass  [ ] Fail  Bug# ______

   String Value Text Field                 [ ] Pass  [ ] Fail  Bug# ______

   Array Name Text Field                   [ ] Pass  [ ] Fail  Bug# ______

   Array Row Text Field                    [ ] Pass  [ ] Fail  Bug# ______

   Array Column Text Field                 [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete           [ ] Pass  [ ] Fail  Bug# ______

   Show Script [N/A]

6. Evaluate panel control.

   Tab Key Navigation                      [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore [N/A]

   Minimize [N/A]

   Maximize [N/A]
```

```
   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename [N/A]

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.11 Solar System Objects

```
$Id: SolarSystemEarthPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                        [ ] Pass  [ ] Fail  Bug# ______

   Open Resource                            [ ] Pass  [ ] Fail  Bug# ______
```

```
   Close Resource                          [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                        [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                        [ ] Pass  [ ] Fail  Bug# ______

      Equatorial Radius Text Field         [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field         [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                     [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                        [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body          [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source      [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405)[ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                      [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                    [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
      Initial A1 Epoch                     [ ] Pass  [ ] Fail  Bug# ______
      SMA                                  [ ] Pass  [ ] Fail  Bug# ______
      ECC                                  [ ] Pass  [ ] Fail  Bug# ______
      INC                                  [ ] Pass  [ ] Fail  Bug# ______
      RAAN                                 [ ] Pass  [ ] Fail  Bug# ______
      AOP                                  [ ] Pass  [ ] Fail  Bug# ______
      TA                                   [ ] Pass  [ ] Fail  Bug# ______

   Orientation Tab
      Spin Axis RA Constant                [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                    [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant               [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                   [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                    [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                        [ ] Pass  [ ] Fail  Bug# ______
      Nutation Update Interval             [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                 [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete           [ ] Pass  [ ] Fail  Bug# ______
```

```
   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a        [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemJupiterPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________
```

```
        ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                          [ ] Pass  [ ] Fail  Bug# ______

   Open Jupiter Resource                      [ ] Pass  [ ] Fail  Bug# ______

   Close Jupiter Resource                     [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                           [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                           [ ] Pass  [ ] Fail  Bug# ______

      Equatorial Radius Text Field            [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field            [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                        [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                           [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body             [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source         [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405)   [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                         [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                       [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
      Initial A1 Epoch                        [ ] Pass  [ ] Fail  Bug# ______
      SMA                                     [ ] Pass  [ ] Fail  Bug# ______
      ECC                                     [ ] Pass  [ ] Fail  Bug# ______
      INC                                     [ ] Pass  [ ] Fail  Bug# ______
      RAAN                                    [ ] Pass  [ ] Fail  Bug# ______
      AOP                                     [ ] Pass  [ ] Fail  Bug# ______
      TA                                      [ ] Pass  [ ] Fail  Bug# ______
```

```
   Orientation Tab
      Spin Axis RA Constant                [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                    [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant               [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                   [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                    [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                        [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                 [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete           [ ] Pass  [ ] Fail  Bug# ______

   Show Script                             [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                      [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                 [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                [ ] Pass  [ ] Fail  Bug# ______

   Close                                   [ ] Pass  [ ] Fail  Bug# ______


7. Evaluate rename functionality.

   Rename (on resource tree)     n/a       [ ] Pass  [ ] Fail  Bug# ______


Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail


Bugs Reported:
```

Notes:

$Id: SolarSystemLunaPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____ Date: _____ Build: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource [ ] Pass [ ] Fail Bug# _____

   Open Resource [ ] Pass [ ] Fail Bug# _____

   Close Resource [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field [ ] Pass [ ] Fail Bug# _____

      Equatorial Radius Text Field [ ] Pass [ ] Fail Bug# _____

      Flattening Factor Text Field [ ] Pass [ ] Fail Bug# _____

      Texture Map File [ ] Pass [ ] Fail Bug# _____

      Browse Button [ ] Pass [ ] Fail Bug# _____

Orbit Tab

Ephemeris Data Central Body	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Date Ephemeris Source	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Data Ephemeris File (DE405)	[ ] Pass	[ ] Fail	Bug# _____
NAIF ID (Spice)	[ ] Pass	[ ] Fail	Bug# _____
SPK Files (Spice)	[ ] Pass	[ ] Fail	Bug# _____

Initial TwoBody State

Initial A1 Epoch	[ ] Pass	[ ] Fail	Bug# _____
SMA	[ ] Pass	[ ] Fail	Bug# _____
ECC	[ ] Pass	[ ] Fail	Bug# _____
INC	[ ] Pass	[ ] Fail	Bug# _____
RAAN	[ ] Pass	[ ] Fail	Bug# _____
AOP	[ ] Pass	[ ] Fail	Bug# _____
TA	[ ] Pass	[ ] Fail	Bug# _____

Orientation Tab

Spin Axis RA Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis RA Rate	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Constant	[ ] Pass	[ ] Fail	Bug# _____
Rotation Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Date Source	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____
Maximize	[ ] Pass	[ ] Fail	Bug# _____
Close	[ ] Pass	[ ] Fail	Bug# _____

```
7. Evaluate rename functionality.

   Rename (on resource tree)     n/a         [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemMarsPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                         [ ] Pass  [ ] Fail  Bug# ______

   Open Mars Resource                        [ ] Pass  [ ] Fail  Bug# ______

   Close Mars Resource                       [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.
```

```
   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                          [ ] Pass  [ ] Fail  Bug# ______

      Equatorial Radius Text Field           [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field           [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                       [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                          [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body            [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source        [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405)  [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                        [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                      [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
      Initial A1 Epoch                       [ ] Pass  [ ] Fail  Bug# ______
      SMA                                    [ ] Pass  [ ] Fail  Bug# ______
      ECC                                    [ ] Pass  [ ] Fail  Bug# ______
      INC                                    [ ] Pass  [ ] Fail  Bug# ______
      RAAN                                   [ ] Pass  [ ] Fail  Bug# ______
      AOP                                    [ ] Pass  [ ] Fail  Bug# ______
      TA                                     [ ] Pass  [ ] Fail  Bug# ______


   Orientation Tab
      Spin Axis RA Constant                  [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                      [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant                 [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                     [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                      [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                          [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                   [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.
```

```
   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a         [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemMercuryPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:
```

```
This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                          [ ] Pass  [ ] Fail  Bug# ______

   Open Mercury Resource                      [ ] Pass  [ ] Fail  Bug# ______

   Close Mercury Resource                     [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                           [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                           [ ] Pass  [ ] Fail  Bug# ______

      Equatorial Radius Text Field            [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field            [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                        [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                           [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body             [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source         [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405)   [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                         [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                       [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
      Initial A1 Epoch                        [ ] Pass  [ ] Fail  Bug# ______
      SMA                                     [ ] Pass  [ ] Fail  Bug# ______
      ECC                                     [ ] Pass  [ ] Fail  Bug# ______
      INC                                     [ ] Pass  [ ] Fail  Bug# ______
      RAAN                                    [ ] Pass  [ ] Fail  Bug# ______
      AOP                                     [ ] Pass  [ ] Fail  Bug# ______
      TA                                      [ ] Pass  [ ] Fail  Bug# ______

  Orientation Tab
      Spin Axis RA Constant                   [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                       [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant                  [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                      [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                       [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                           [ ] Pass  [ ] Fail  Bug# ______
```

```
      Rotation Date Source                  [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a        [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

```
$Id: SolarSystemNeptunePanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                        [ ] Pass  [ ] Fail  Bug# ______

   Open Neptune Resource                    [ ] Pass  [ ] Fail  Bug# ______

   Close Neptune Resource                   [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                         [ ] Pass  [ ] Fail  Bug# ______

      Equatorial Radius Text Field          [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field          [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                      [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                         [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body           [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source       [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405) [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                       [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                     [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
```

```
      Initial A1 Epoch                    [ ] Pass  [ ] Fail  Bug# ______
      SMA                                 [ ] Pass  [ ] Fail  Bug# ______
      ECC                                 [ ] Pass  [ ] Fail  Bug# ______
      INC                                 [ ] Pass  [ ] Fail  Bug# ______
      RAAN                                [ ] Pass  [ ] Fail  Bug# ______
      AOP                                 [ ] Pass  [ ] Fail  Bug# ______
      TA                                  [ ] Pass  [ ] Fail  Bug# ______

   Orientation Tab
      Spin Axis RA Constant               [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                   [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant              [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                  [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                   [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                       [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete          [ ] Pass  [ ] Fail  Bug# ______

   Show Script                            [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                     [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                 [ ] Pass  [ ] Fail  Bug# ______

   Apply                                  [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                              [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                [ ] Pass  [ ] Fail  Bug# ______

   Minimize                               [ ] Pass  [ ] Fail  Bug# ______

   Maximize                               [ ] Pass  [ ] Fail  Bug# ______

   Close                                  [ ] Pass  [ ] Fail  Bug# ______


7. Evaluate rename functionality.

   Rename (on resource tree)     n/a      [ ] Pass  [ ] Fail  Bug# ______


Summary:
```

```
   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemEarthPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                         [ ] Pass  [ ] Fail  Bug# ______

   Open Resource                             [ ] Pass  [ ] Fail  Bug# ______

   Close Resource                            [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                          [ ] Pass  [ ] Fail  Bug# ______
```

```
      Equatorial Radius Text Field          [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field          [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                      [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                         [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body           [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source       [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405) [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                       [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                     [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
      Initial A1 Epoch                      [ ] Pass  [ ] Fail  Bug# ______
      SMA                                   [ ] Pass  [ ] Fail  Bug# ______
      ECC                                   [ ] Pass  [ ] Fail  Bug# ______
      INC                                   [ ] Pass  [ ] Fail  Bug# ______
      RAAN                                  [ ] Pass  [ ] Fail  Bug# ______
      AOP                                   [ ] Pass  [ ] Fail  Bug# ______
      TA                                    [ ] Pass  [ ] Fail  Bug# ______

   Orientation Tab
      Spin Axis RA Constant                 [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                     [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant                [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                    [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                     [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                         [ ] Pass  [ ] Fail  Bug# ______
      Nutation Update Interval              [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                  [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]
```

```
   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a        [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Solar System panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Open the appropriate object panel.

   Open SolarSystem Resource                [ ] Pass  [ ] Fail  Bug# ______
```

```
   Close SolarSystem Resource                [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Ephemeris Update Interval Text Field      [ ] Pass  [ ] Fail  Bug# ______

   Ephemeris Source List                     [ ] Pass  [ ] Fail  Bug# ______

   Ephemeris File Name Text Field            [ ] Pass  [ ] Fail  Bug# ______

   Browse (File Select) Action Button        [ ] Pass  [ ] Fail  Bug# ______

   Use TT for Ephemeris Check Box            [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.
```

```
   n/a


Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail


Bugs Reported:



Notes:




$Id: SolarSystemPlanetPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Planet Resource                     [ ] Pass  [ ] Fail  Bug# ______

   Open Planet Resource                    [ ] Pass  [ ] Fail  Bug# ______

   Close Planet Resource                   [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                        [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements. (* all from Earth values)
```

```
   Properties Tab

      mu Text Field                          [ ] Pass  [ ] Fail  Bug# ______

      Equatorial Radius Text Field           [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field           [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                       [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                          [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body            [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source        [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405)  [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                        [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                      [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
      Initial A1 Epoch                       [ ] Pass  [ ] Fail  Bug# ______
      SMA                                    [ ] Pass  [ ] Fail  Bug# ______
      ECC                                    [ ] Pass  [ ] Fail  Bug# ______
      INC                                    [ ] Pass  [ ] Fail  Bug# ______
      RAAN                                   [ ] Pass  [ ] Fail  Bug# ______
      AOP                                    [ ] Pass  [ ] Fail  Bug# ______
      TA                                     [ ] Pass  [ ] Fail  Bug# ______

   Orientation Tab
      Spin Axis RA Constant                  [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                      [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant                 [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                     [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                      [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                          [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                   [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______
```

```
   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a        [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemPlutoPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ____________________


Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:
```

1. Add and open the appropriate object panel.

Add Moon Resource	[ ] Pass	[ ] Fail	Bug# _____
Open Pluto Resource	[ ] Pass	[ ] Fail	Bug# _____
Close Pluto Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements. (* all from Earth values)

Properties Tab

mu Text Field	[ ] Pass	[ ] Fail	Bug# _____
Equatorial Radius Text Field	[ ] Pass	[ ] Fail	Bug# _____
Flattening Factor Text Field	[ ] Pass	[ ] Fail	Bug# _____
Texture Map File	[ ] Pass	[ ] Fail	Bug# _____
Browse Button	[ ] Pass	[ ] Fail	Bug# _____

Orbit Tab

Ephemeris Data Central Body	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Date Ephemeris Source	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Data Ephemeris File (DE405)	[ ] Pass	[ ] Fail	Bug# _____
NAIF ID (Spice)	[ ] Pass	[ ] Fail	Bug# _____
SPK Files (Spice)	[ ] Pass	[ ] Fail	Bug# _____
Initial TwoBody State			
Initial A1 Epoch	[ ] Pass	[ ] Fail	Bug# _____
SMA	[ ] Pass	[ ] Fail	Bug# _____
ECC	[ ] Pass	[ ] Fail	Bug# _____
INC	[ ] Pass	[ ] Fail	Bug# _____
RAAN	[ ] Pass	[ ] Fail	Bug# _____
AOP	[ ] Pass	[ ] Fail	Bug# _____
TA	[ ] Pass	[ ] Fail	Bug# _____

Orientation Tab

Spin Axis RA Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis RA Rate	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Constant	[ ] Pass	[ ] Fail	Bug# _____
Rotation Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Date Source	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

```
5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a        [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemSaternPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________
```

```
Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _______________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                          [ ] Pass  [ ] Fail  Bug# ______

   Open Satern Resource                       [ ] Pass  [ ] Fail  Bug# ______

   Close Satern Resource                      [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                           [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                           [ ] Pass  [ ] Fail  Bug# ______

      Equatorial Radius Text Field            [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field            [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                        [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                           [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body             [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source         [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405)   [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                         [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                       [ ] Pass  [ ] Fail  Bug# ______

      Initial TwoBody State
      Initial A1 Epoch                        [ ] Pass  [ ] Fail  Bug# ______
      SMA                                     [ ] Pass  [ ] Fail  Bug# ______
      ECC                                     [ ] Pass  [ ] Fail  Bug# ______
      INC                                     [ ] Pass  [ ] Fail  Bug# ______
```

```
      RAAN                                  [ ] Pass  [ ] Fail  Bug# ______
      AOP                                   [ ] Pass  [ ] Fail  Bug# ______
      TA                                    [ ] Pass  [ ] Fail  Bug# ______

   Orientation Tab
      Spin Axis RA Constant                 [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                     [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant                [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                    [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                     [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                         [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                  [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______


7. Evaluate rename functionality.

   Rename (on resource tree)     n/a        [ ] Pass  [ ] Fail  Bug# ______


Summary:

   Number of passed test elements      ______

   Total number of test elements       ______
```

```
   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:



Notes:



$Id: SolarSystemSunPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Planet Resource                       [ ] Pass  [ ] Fail  Bug# ______

   Add Comet Resource                        [ ] Pass  [ ] Fail  Bug# ______

   Add Asteriod Resource                     [ ] Pass  [ ] Fail  Bug# ______

   Open Sun Resource                         [ ] Pass  [ ] Fail  Bug# ______

   Close Sun Resource                        [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements. (* all from Earth values)

   Properties Tab

      mu Text Field                          [ ] Pass  [ ] Fail  Bug# ______
```

```
      Equatorial Radius Text Field          [ ] Pass  [ ] Fail  Bug# ______

      Flattening Factor Text Field          [ ] Pass  [ ] Fail  Bug# ______

      Texture Map File                      [ ] Pass  [ ] Fail  Bug# ______

      Browse Button                         [ ] Pass  [ ] Fail  Bug# ______

   Orbit Tab
      Ephemeris Data Central Body           [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Date Ephemeris Source       [ ] Pass  [ ] Fail  Bug# ______
      Ephemeris Data Ephemeris File (DE405) [ ] Pass  [ ] Fail  Bug# ______
      NAIF ID (Spice)                       [ ] Pass  [ ] Fail  Bug# ______
      SPK Files (Spice)                     [ ] Pass  [ ] Fail  Bug# ______

   Orientation Tab
      Spin Axis RA Constant                 [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis RA Rate                     [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Constant                [ ] Pass  [ ] Fail  Bug# ______
      Spin Axis DEC Rate                    [ ] Pass  [ ] Fail  Bug# ______
      Rotation Constant                     [ ] Pass  [ ] Fail  Bug# ______
      Rotation Rate                         [ ] Pass  [ ] Fail  Bug# ______
      Rotation Date Source                  [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______


6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______
```

```
7. Evaluate rename functionality.

   Rename (on resource tree)     n/a         [ ] Pass  [ ] Fail  Bug# ______


Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail


Bugs Reported:



Notes:



$Id: SolarSystemUranusPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Moon Resource                         [ ] Pass  [ ] Fail  Bug# ______

   Open Uranus Resource                      [ ] Pass  [ ] Fail  Bug# ______

   Close Uranus Resource                     [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______
```

3. Evaluate the individual panel elements. (* all from Earth values)

Properties Tab

mu Text Field	[ ] Pass	[ ] Fail	Bug# _____
Equatorial Radius Text Field	[ ] Pass	[ ] Fail	Bug# _____
Flattening Factor Text Field	[ ] Pass	[ ] Fail	Bug# _____
Texture Map File	[ ] Pass	[ ] Fail	Bug# _____
Browse Button	[ ] Pass	[ ] Fail	Bug# _____

Orbit Tab

Ephemeris Data Central Body	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Date Ephemeris Source	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Data Ephemeris File (DE405)	[ ] Pass	[ ] Fail	Bug# _____
NAIF ID (Spice)	[ ] Pass	[ ] Fail	Bug# _____
SPK Files (Spice)	[ ] Pass	[ ] Fail	Bug# _____
Initial TwoBody State			
Initial A1 Epoch	[ ] Pass	[ ] Fail	Bug# _____
SMA	[ ] Pass	[ ] Fail	Bug# _____
ECC	[ ] Pass	[ ] Fail	Bug# _____
INC	[ ] Pass	[ ] Fail	Bug# _____
RAAN	[ ] Pass	[ ] Fail	Bug# _____
AOP	[ ] Pass	[ ] Fail	Bug# _____
TA	[ ] Pass	[ ] Fail	Bug# _____

Orientation Tab

Spin Axis RA Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis RA Rate	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Constant	[ ] Pass	[ ] Fail	Bug# _____
Rotation Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Date Source	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____

```
   Apply                                  [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                              [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                [ ] Pass  [ ] Fail  Bug# ______

   Minimize                               [ ] Pass  [ ] Fail  Bug# ______

   Maximize                               [ ] Pass  [ ] Fail  Bug# ______

   Close                                  [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a      [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                       [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SolarSystemVenusPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Sun panel.
(* indicates sub-panel whose functionality is tested separately)
```

Procedure:

1. Add and open the appropriate object panel.

Add Moon Resource	[ ] Pass	[ ] Fail	Bug# _____
Open Venus Resource	[ ] Pass	[ ] Fail	Bug# _____
Close Venus Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements. (* all from Earth values)

Properties Tab

mu Text Field	[ ] Pass	[ ] Fail	Bug# _____
Equatorial Radius Text Field	[ ] Pass	[ ] Fail	Bug# _____
Flattening Factor Text Field	[ ] Pass	[ ] Fail	Bug# _____
Texture Map File	[ ] Pass	[ ] Fail	Bug# _____
Browse Button	[ ] Pass	[ ] Fail	Bug# _____

Orbit Tab

Ephemeris Data Central Body	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Date Ephemeris Source	[ ] Pass	[ ] Fail	Bug# _____
Ephemeris Data Ephemeris File (DE405)	[ ] Pass	[ ] Fail	Bug# _____
NAIF ID (Spice)	[ ] Pass	[ ] Fail	Bug# _____
SPK Files (Spice)	[ ] Pass	[ ] Fail	Bug# _____
Initial TwoBody State			
Initial A1 Epoch	[ ] Pass	[ ] Fail	Bug# _____
SMA	[ ] Pass	[ ] Fail	Bug# _____
ECC	[ ] Pass	[ ] Fail	Bug# _____
INC	[ ] Pass	[ ] Fail	Bug# _____
RAAN	[ ] Pass	[ ] Fail	Bug# _____
AOP	[ ] Pass	[ ] Fail	Bug# _____
TA	[ ] Pass	[ ] Fail	Bug# _____

Orientation Tab

Spin Axis RA Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis RA Rate	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Constant	[ ] Pass	[ ] Fail	Bug# _____
Spin Axis DEC Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Constant	[ ] Pass	[ ] Fail	Bug# _____
Rotation Rate	[ ] Pass	[ ] Fail	Bug# _____
Rotation Date Source	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

```
   N/A

5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)     n/a        [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: LunaPanel.txt,v 1.4 2006/11/22 21:32:34 dconway Exp $
```

Tester: _____ Date: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Luna panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

Create Luna Resource	[ ] Pass	[ ] Fail	Bug# _____
Open Luna Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

mu Text Field	[ ] Pass	[ ] Fail	Bug# _____
Equatorial Radius Text Field	[ ] Pass	[ ] Fail	Bug# _____
Flattening Factor Text Field	[ ] Pass	[ ] Fail	Bug# _____
Initial Epoch Text Field	[ ] Pass	[ ] Fail	Bug# _____
Semimajor Axis Text Field	[ ] Pass	[ ] Fail	Bug# _____
Eccentricity Text Field	[ ] Pass	[ ] Fail	Bug# _____
Inclination Text Field	[ ] Pass	[ ] Fail	Bug# _____
RA of Ascending Node Text Field	[ ] Pass	[ ] Fail	Bug# _____
Argument of Perigee Text Field	[ ] Pass	[ ] Fail	Bug# _____
True Anomaly Text Field	[ ] Pass	[ ] Fail	Bug# _____
Rotation Data Source Combo Box	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____
Maximize	[ ] Pass	[ ] Fail	Bug# _____
Close	[ ] Pass	[ ] Fail	Bug# _____

7. Evaluate rename functionality.

Rename (on resource tree)	[ ] Pass	[ ] Fail	Bug# _____

Summary:

Number of passed test elements	_____
Total number of test elements	_____
Test case status	[ ] Pass [ ] Fail

Bugs Reported:

Notes:

10.12 Finite Burn

```
$Id: FiniteBurnPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Finite Burn panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Finite Burn Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Open Finite Burn Resource                [ ] Pass  [ ] Fail  Bug# ______

   Close Finite Burn Resource               [ ] Pass  [ ] Fail  Bug# ______

   Delete Finite Burn Resource              [ ] Pass  [ ] Fail  Bug# ______

   Clone Finite Burn Resource               [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.

   Use Thruster Combo Box                   [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______
```

```
6. Evaluate panel control.

   Tab Key Navigation                    [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                [ ] Pass  [ ] Fail  Bug# ______

   Apply                                 [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                             [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                               [ ] Pass  [ ] Fail  Bug# ______

   Minimize                              [ ] Pass  [ ] Fail  Bug# ______

   Maximize                              [ ] Pass  [ ] Fail  Bug# ______

   Close                                 [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)             [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.13 Formation

```
$Id: FormationPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: ______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________
```

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Formation panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

Create Spacecraft Resource	[ ] Pass	[ ] Fail	Bug# _____
Create Spacecraft Resource	[ ] Pass	[ ] Fail	Bug# _____
Create Formation Resource	[ ] Pass	[ ] Fail	Bug# _____
Open Formation Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Space Objects List	[ ] Pass	[ ] Fail	Bug# _____
Space Objects in Formation List	[ ] Pass	[ ] Fail	Bug# _____
-> Selection Button	[ ] Pass	[ ] Fail	Bug# _____
<- Selection Button	[ ] Pass	[ ] Fail	Bug# _____
<= Selection Button	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

```
   Tab Key Navigation                    [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                [ ] Pass  [ ] Fail  Bug# ______

   Apply                                 [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                             [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                               [ ] Pass  [ ] Fail  Bug# ______

   Minimize                              [ ] Pass  [ ] Fail  Bug# ______

   Maximize                              [ ] Pass  [ ] Fail  Bug# ______

   Close                                 [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)             [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.14 Fuel Tank

```
$Id: FuelTankPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _______________

            ___ Linux, Distribution: ___________________
```

Description:

This test validates the functionality of the Fuel Tank panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

Add FuelTank Resource	[ ] Pass	[ ] Fail	Bug# _____
Open FuelTank Resource	[ ] Pass	[ ] Fail	Bug# _____
Close FuelTank Resource	[ ] Pass	[ ] Fail	Bug# _____
Clone FuelTank Resource	[ ] Pass	[ ] Fail	Bug# _____
Rename FuelTank Resource	[ ] Pass	[ ] Fail	Bug# _____
Delete FuelTank Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Fuel Mass Text Field	[ ] Pass	[ ] Fail	Bug# _____
Pressure Text Field	[ ] Pass	[ ] Fail	Bug# _____
Temperature Text Field	[ ] Pass	[ ] Fail	Bug# _____
Reference Temperature Text Field	[ ] Pass	[ ] Fail	Bug# _____
Volume Text Field	[ ] Pass	[ ] Fail	Bug# _____
Fuel Density Text Field	[ ] Pass	[ ] Fail	Bug# _____
Pressure Model Drop Down Field	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

```
6. Evaluate panel control.

   Tab Key Navigation                    [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                [ ] Pass  [ ] Fail  Bug# ______

   Apply                                 [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                             [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                               [ ] Pass  [ ] Fail  Bug# ______

   Minimize                              [ ] Pass  [ ] Fail  Bug# ______

   Maximize                              [ ] Pass  [ ] Fail  Bug# ______

   Close                                 [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)             [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.15 GMAT Function

```
$Id: GMATFunctionPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: ______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________
```

```
___ Macintosh, OS X Version: _____________

___ Linux, Distribution: ___________________
```

Description:

This test validates the functionality of the GMAT Function panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

Create GMATFunction Resource	[ ] Pass	[ ] Fail	Bug# ______
Open GMATFunction Resource	[ ] Pass	[ ] Fail	Bug# ______

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# ______

3. Evaluate the individual panel elements.

File Text Field	[ ] Pass	[ ] Fail	Bug# ______
Browse (File Select) Action Button	[ ] Pass	[ ] Fail	Bug# ______
Load Action Button	[ ] Pass	[ ] Fail	Bug# ______
Save Action Button	[ ] Pass	[ ] Fail	Bug# ______

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# ______
Show Script	[ ] Pass	[ ] Fail	Bug# ______

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# ______
Cancel	[ ] Pass	[ ] Fail	Bug# ______
Apply	[ ] Pass	[ ] Fail	Bug# ______

```
   OK (Save)                              [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                [ ] Pass  [ ] Fail  Bug# ______

   Minimize                               [ ] Pass  [ ] Fail  Bug# ______

   Maximize                               [ ] Pass  [ ] Fail  Bug# ______

   Close                                  [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)              [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                       [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.16 Ground Station

```
$Id: GroundStationPanel.txt,v 1.4 2009/10/08 08:00:00 jbez Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the GroundStation Attitude panel.
(* indicates sub-panel whose functionality is tested separately)
```

Procedure:

1. Add and open the appropriate object panel.

Add GroundStation Resource	[ ] Pass	[ ] Fail	Bug# _____
Open GroundStation Resource	[ ] Pass	[ ] Fail	Bug# _____
Close GroundStation Resource	[ ] Pass	[ ] Fail	Bug# _____
Delete GroundStation Resource	[ ] Pass	[ ] Fail	Bug# _____
Clone GroundStation Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Central Body Combo Box	[ ] Pass	[ ] Fail	Bug# _____
State Type Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Horizion Reference Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Location 1,2 3 Field	[ ] Pass	[ ] Fail	Bug# _____
Id Text Field	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____

```
   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______


7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______


Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail


Bugs Reported:


Notes:
```

10.17 Impulsive Burn

```
$Id: ImpulsiveBurnPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _______________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Impulsive Burn panel.
(* indicates sub-panel whose functionality is tested separately)
```

Procedure:

1. Add and open the appropriate object panel.

Add ImpulsiveBurn Resource	[ ] Pass	[ ] Fail	Bug# _____
Open ImpulsiveBurn Resource	[ ] Pass	[ ] Fail	Bug# _____
Close ImpulsiveBurn Resource	[ ] Pass	[ ] Fail	Bug# _____
Delete ImpulsiveBurn Resource	[ ] Pass	[ ] Fail	Bug# _____
Clone ImpulsiveBurn Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Coordinate System	[ ] Pass	[ ] Fail	Bug# _____
Origin Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Axes Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Thrust Vector Element 1 Text Field	[ ] Pass	[ ] Fail	Bug# _____
Thrust Vector Element 2 Text Field	[ ] Pass	[ ] Fail	Bug# _____
Thrust Vector Element 3 Text Field	[ ] Pass	[ ] Fail	Bug# _____
Decrement Mass Button	[ ] Pass	[ ] Fail	Bug# _____
Tank Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Isp Text Field	[ ] Pass	[ ] Fail	Bug# _____
Gravitational Accel Field	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

```
   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)                 [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.18 Libration Point

```
$Id: LibrationPointPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________
```

Description:

This test validates the functionality of the Libration Point panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

Add LibrationPoint Resource	[ ] Pass	[ ] Fail	Bug# _____
Open LibrationPoint Resource	[ ] Pass	[ ] Fail	Bug# _____
Close LibrationPoint Resource	[ ] Pass	[ ] Fail	Bug# _____
Clone LibrationPoint Resource	[ ] Pass	[ ] Fail	Bug# _____
Delete LibrationPoint Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Primary Body Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Secondary Body Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Libration Point Combo Box	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____

```
   OK (Save)                              [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                [ ] Pass  [ ] Fail  Bug# ______

   Minimize                               [ ] Pass  [ ] Fail  Bug# ______

   Maximize                               [ ] Pass  [ ] Fail  Bug# ______

   Close                                  [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)              [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                       [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.19 MATLAB function

```
$Id: MATLABFunctionPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the MATLAB Function panel.
(* indicates sub-panel whose functionality is tested separately)
```

Procedure:

1. Create and open the appropriate object panel.

   Create MATLABFunction Resource [ ] Pass [ ] Fail Bug# _____

   Open MATLABFunction Resource [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

   Path Text Field [ ] Pass [ ] Fail Bug# _____

   Browse (File Select) Action Button [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete [ ] Pass [ ] Fail Bug# _____

   Show Script [ ] Pass [ ] Fail Bug# _____

6. Evaluate panel control.

   Tab Key Navigation [ ] Pass [ ] Fail Bug# _____

   Cancel [ ] Pass [ ] Fail Bug# _____

   Apply [ ] Pass [ ] Fail Bug# _____

   OK (Save) [ ] Pass [ ] Fail Bug# _____

   Help [DEFERRED]

   Restore [ ] Pass [ ] Fail Bug# _____

   Minimize [ ] Pass [ ] Fail Bug# _____

   Maximize [ ] Pass [ ] Fail Bug# _____

   Close [ ] Pass [ ] Fail Bug# _____

7. Evaluate rename functionality.

```
   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                           [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.20 OpenGL Plot

```
$Id: OpenGLPanel.txt,v 1.4 2006/11/22 21:27:52 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the OpenGL panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create OpenGL Resource                   [ ] Pass  [ ] Fail  Bug# ______

   Open OpenGL Resource                     [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______
```

3. Evaluate the individual panel elements.

Show Plot Check Box	[ ] Pass	[ ] Fail	Bug# _____
Collect Data Text Field	[ ] Pass	[ ] Fail	Bug# _____
Update Plot Text Field	[ ] Pass	[ ] Fail	Bug# _____
Number of Points to Redraw Text Field	[ ] Pass	[ ] Fail	Bug# _____
Draw Wireframe Check Box	[ ] Pass	[ ] Fail	Bug# _____
Draw Targeting Check Box	[ ] Pass	[ ] Fail	Bug# _____
Draw Ecliptic Plane Check Box	[ ] Pass	[ ] Fail	Bug# _____
Draw XY Plane Check Box	[ ] Pass	[ ] Fail	Bug# _____
Draw Axes Check Box	[ ] Pass	[ ] Fail	Bug# _____
Draw Grid Check Box	[ ] Pass	[ ] Fail	Bug# _____
Draw Earth/Sun Lines Check Box	[ ] Pass	[ ] Fail	Bug# _____
Spacecraft List	[ ] Pass	[ ] Fail	Bug# _____
Selected Spacecraft List	[ ] Pass	[ ] Fail	Bug# _____
Celestial Object List	[ ] Pass	[ ] Fail	Bug# _____
Selected Celestial Object List	[ ] Pass	[ ] Fail	Bug# _____
--> (Add) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
<-- (Remove) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
< = (Remove All) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
Show Object Check Box	[ ] Pass	[ ] Fail	Bug# _____
Orbit Color Select Box	[ ] Pass	[ ] Fail	Bug# _____
Target Color Select Box	[ ] Pass	[ ] Fail	Bug# _____
Use Initial View Definition Check Box	[ ] Pass	[ ] Fail	Bug# _____
Use Perspective Mode Check Box	[ ] Pass	[ ] Fail	Bug# _____
Use Fixed FOV Angle Check Box	[ ] Pass	[ ] Fail	Bug# _____
Field of View Text Field	[ ] Pass	[ ] Fail	Bug# _____
Coordinate System Combo Box	[ ] Pass	[ ] Fail	Bug# _____
View Point Reference Combo Box (see 4a)	[ ] Pass	[ ] Fail	Bug# _____

```
   View Point Vector Combo Box (see 4b)      [ ] Pass  [ ] Fail  Bug# ______

   View Scale Factor Text Field              [ ] Pass  [ ] Fail  Bug# ______

   View Direction Combo Box (see 4c)         [ ] Pass  [ ] Fail  Bug# ______

   Coordinate System Combo Box               [ ] Pass  [ ] Fail  Bug# ______

   Axis Combo Box                            [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   a. Select 'Vector' for View Point Reference

      Vector 1 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 2 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 3 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

   b. Select 'Vector' for View Point Vector

      Vector 1 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 2 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 3 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

   c. Select 'Vector' for View Direction

      Vector 1 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 2 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

      Vector 3 Text Field                    [ ] Pass  [ ] Fail  Bug# ______

   Use Perspective Mode Check Box            [ ] Pass  [ ] Fail  Bug# ______
   --- select checkbox to check following

   Use Fixed FOV Angle Check Box             [ ] Pass  [ ] Fail  Bug# ______
   --- select checkbox to check following

   Field of View Text Field                  [ ] Pass  [ ] Fail  Bug# ______

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______
```

```
   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                 [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                [ ] Pass  [ ] Fail  Bug# ______

   Close                                   [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)               [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                        [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.21 Parameter Select (Create Only)

```
$Id: ParameterSelectCreateOnlyDialog.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________
```

```
Description:

This test validates the functionality of the Parameter Select dialog, which
appears via other panels (e.g. CallFunctionPanel) - this dialog only allows
the user to create and select new variables, not select already-existing ones.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Create and open the appropriate object panel.

   Create other panel Resource              [ ] Pass  [ ] Fail  Bug# ______

   Open other panel Resource                [ ] Pass  [ ] Fail  Bug# ______

   Open Create Action Button                [ ] Pass  [ ] Fail  Bug# ______
   (e.g. Output Variable Field)


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.

   Create Action Button*                    [ ] Pass  [ ] Fail  Bug# ______

   Selected List                            [ ] Pass  [ ] Fail  Bug# ______

   -> (Add) Selection Button                [ ] Pass  [ ] Fail  Bug# ______

   <- (Remove) Selection Button             [ ] Pass  [ ] Fail  Bug# ______

   <= (Remove All) Selection Button         [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script [N/A]

   Command Summary [N/A]


6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______
```

```
   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore [N/A]

   Minimize [N/A]

   Maximize [N/A]

   Close                                   [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename [N/A]

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.22 Parameter Select

```
$Id: ParameterSelectDialog.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________
```

```
Description:

This test validates the functionality of the Parameter Select dialog, which
appears via other panels (e.g. IfCmdPanel, WhileCmdPanel)
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Create and open the appropriate object panel.

   Create other panel Resource              [ ] Pass  [ ] Fail  Bug# ______

   Open other panel Resource                [ ] Pass  [ ] Fail  Bug# ______

   Open Create Action Button                [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.

   Create Action Button*                    [ ] Pass  [ ] Fail  Bug# ______

   Object Type Combo Box                    [ ] Pass  [ ] Fail  Bug# ______

   Object Combo Box                         [ ] Pass  [ ] Fail  Bug# ______

   Property Selection List                  [ ] Pass  [ ] Fail  Bug# ______

   Selected List                            [ ] Pass  [ ] Fail  Bug# ______

   -> (Add) Selection Button                [ ] Pass  [ ] Fail  Bug# ______

   <- (Remove) Selection Button             [ ] Pass  [ ] Fail  Bug# ______

   <= (Remove All) Selection Button         [ ] Pass  [ ] Fail  Bug# ______


4. Evaluate panel-specific functionality.

   N/A


5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script [N/A]

   Command Summary [N/A]


6. Evaluate panel control.
```

```
   Tab Key Navigation                          [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                      [ ] Pass  [ ] Fail  Bug# ______

   Apply                                       [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                   [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore [N/A]

   Minimize [N/A]

   Maximize [N/A]

   Close                                       [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename [N/A]

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.23 Propagator

```
$Id: PropagatorPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________
```

```
            ___ Linux, Distribution: ____________________


Description:

This test validates the functionality of the Propagator panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Propagator Resource                  [ ] Pass  [ ] Fail  Bug# ______

   Open Propagator Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Close LibrationPoint Resource            [ ] Pass  [ ] Fail  Bug# ______

   Clone LibrationPoint Resource            [ ] Pass  [ ] Fail  Bug# ______

   Delete LibrationPoint Resource           [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.

   Integrator Type Combo Box                [ ] Pass  [ ] Fail  Bug# ______

   Initial Step Size Text Field             [ ] Pass  [ ] Fail  Bug# ______

   Accuracy Text Field                      [ ] Pass  [ ] Fail  Bug# ______

   Minimum Step Size Text Field             [ ] Pass  [ ] Fail  Bug# ______

   Maximum Step Size Text Field             [ ] Pass  [ ] Fail  Bug# ______

   Maximum Step Attempts Text Field         [ ] Pass  [ ] Fail  Bug# ______

   Minimum Integration Error Text Field     [ ] Pass  [ ] Fail  Bug# ______

   Nominal Integration Error Text Field     [ ] Pass  [ ] Fail  Bug# ______

   Error Control Combo Box                  [ ] Pass  [ ] Fail  Bug# ______

   Central Body Combo Box                   [ ] Pass  [ ] Fail  Bug# ______

   Primary Bodies Combo Box                 [ ] Pass  [ ] Fail  Bug# ______

   Primary Bodies Action Button
      Available Bodies List                 [ ] Pass  [ ] Fail  Bug# ______
      Bodies Selected List                  [ ] Pass  [ ] Fail  Bug# ______
      -> (Add) Selection Button             [ ] Pass  [ ] Fail  Bug# ______
```

```
   <- (Remove) Selection Button          [ ] Pass  [ ] Fail  Bug# ______
   <= (Remove All) Selection Button      [ ] Pass  [ ] Fail  Bug# ______
   Orbit Color Selection Button          [ ] Pass  [ ] Fail  Bug# ______
   Cancel Button                         [ ] Pass  [ ] Fail  Bug# ______
   OK Button                             [ ] Pass  [ ] Fail  Bug# ______
   Help [DEFERRED]

Primary Bodies Text List                 [ ] Pass  [ ] Fail  Bug# ______

Gravity Field Type Combo Box             [ ] Pass  [ ] Fail  Bug# ______

Gravity Model File Text Field            [ ] Pass  [ ] Fail  Bug# ______

Gravity Field Degree Text Field          [ ] Pass  [ ] Fail  Bug# ______

Gravity Field Order Text Field           [ ] Pass  [ ] Fail  Bug# ______

Gravity Model File Select Action Button  [ ] Pass  [ ] Fail  Bug# ______

Atmosphere Model Type Combo Box          [ ] Pass  [ ] Fail  Bug# ______

Atmosphere Model Setup Action Button     [ ] Pass  [ ] Fail  Bug# ______
   User Input Radio Button               [ ] Pass  [ ] Fail  Bug# ______
   Solar Flux Text Field                 [ ] Pass  [ ] Fail  Bug# ______
   Average Solar Flux Text Field         [ ] Pass  [ ] Fail  Bug# ______
   Geomagnetic Index (Kp) Text Field     [ ] Pass  [ ] Fail  Bug# ______
   File Input Radio Button               [ ] Pass  [ ] Fail  Bug# ______
   File Name Text Field                  [ ] Pass  [ ] Fail  Bug# ______
   File Selection Button                 [ ] Pass  [ ] Fail  Bug# ______
   Cancel Button                         [ ] Pass  [ ] Fail  Bug# ______
   OK Button                             [ ] Pass  [ ] Fail  Bug# ______
   Help [DEFERRED]

Magnetic Field Type Combo Box            [ ] Pass  [ ] Fail  Bug# ______

Magnetic Field Degree Text Field         [ ] Pass  [ ] Fail  Bug# ______

Magnetic Field Order Text Field          [ ] Pass  [ ] Fail  Bug# ______

Magn. Field File Select Action Button    [ ] Pass  [ ] Fail  Bug# ______

Point Masses Select Action Button
   Available Bodies List                 [ ] Pass  [ ] Fail  Bug# ______
   Bodies Selected List                  [ ] Pass  [ ] Fail  Bug# ______
   -> (Add) Selection Button             [ ] Pass  [ ] Fail  Bug# ______
   <- (Remove) Selection Button          [ ] Pass  [ ] Fail  Bug# ______
   <= (Remove All) Selection Button      [ ] Pass  [ ] Fail  Bug# ______
   Orbit Color Selection Button          [ ] Pass  [ ] Fail  Bug# ______
   Cancel Button                         [ ] Pass  [ ] Fail  Bug# ______
   OK Button                             [ ] Pass  [ ] Fail  Bug# ______
   Help [DEFERRED]

Point Masses Text List                   [ ] Pass  [ ] Fail  Bug# ______

Solar Radiation Pressure Check Box       [ ] Pass  [ ] Fail  Bug# ______
```

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete [ ] Pass [ ] Fail Bug# ______

   Show Script [ ] Pass [ ] Fail Bug# ______

6. Evaluate panel control.

   Tab Key Navigation [ ] Pass [ ] Fail Bug# ______

   Cancel [ ] Pass [ ] Fail Bug# ______

   Apply [ ] Pass [ ] Fail Bug# ______

   OK (Save) [ ] Pass [ ] Fail Bug# ______

   Help [DEFERRED]

   Restore [ ] Pass [ ] Fail Bug# ______

   Minimize [ ] Pass [ ] Fail Bug# ______

   Maximize [ ] Pass [ ] Fail Bug# ______

   Close [ ] Pass [ ] Fail Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree) [ ] Pass [ ] Fail Bug# ______

Summary:

   Number of passed test elements ______

   Total number of test elements ______

   Test case status [ ] Pass [ ] Fail

Bugs Reported:

Notes:

10.24 Report File

```
$Id: ReportFile.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: _____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the Report File panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create ReportFile Resource               [ ] Pass  [ ] Fail  Bug# ______

   Open ReportFile Resource                 [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Create Action Button*                    [ ] Pass  [ ] Fail  Bug# ______

   Object Type Combo Box                    [ ] Pass  [ ] Fail  Bug# ______

   Object Combo Box                         [ ] Pass  [ ] Fail  Bug# ______

   Property List                            [ ] Pass  [ ] Fail  Bug# ______

   Selected List                            [ ] Pass  [ ] Fail  Bug# ______

   --> (Add) Selection Button               [ ] Pass  [ ] Fail  Bug# ______

   <-- (Remove) Selection Button            [ ] Pass  [ ] Fail  Bug# ______

   < = (Remove All) Selection Button        [ ] Pass  [ ] Fail  Bug# ______

   (Move Up) Selection Button               [ ] Pass  [ ] Fail  Bug# ______

   (Move Down) Selection Button             [ ] Pass  [ ] Fail  Bug# ______
```

```
   File Text Field                          [ ] Pass  [ ] Fail  Bug# ______

   Browse (File Select) Action Button       [ ] Pass  [ ] Fail  Bug# ______

   Write Report Check Box                   [ ] Pass  [ ] Fail  Bug# ______

   Write Headers Check Box                  [ ] Pass  [ ] Fail  Bug# ______

   Left Justify Check Box                   [ ] Pass  [ ] Fail  Bug# ______

   Zero Fill Check Box                      [ ] Pass  [ ] Fail  Bug# ______

   Solver Iterations Check Box              [ ] Pass  [ ] Fail  Bug# ______

   Column Width Text Field                  [ ] Pass  [ ] Fail  Bug# ______

   Precision Text Field                     [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______
```

```
Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail


Bugs Reported:


Notes:
```

10.25 Spacecraft

```
$Id: SpacecraftActuatorsPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _______________________________      Date: ____________________________


Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Spacecraft Actuators panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Spacecraft Resource                  [ ] Pass  [ ] Fail  Bug# ______

   Create Thruster Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Open Spacecraft Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Select Actuators Tab                     [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.
```

```
   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Available Thrusters List                  [ ] Pass  [ ] Fail  Bug# ______

   Selected Thrusters List                   [ ] Pass  [ ] Fail  Bug# ______

   -> (Add) Selection Button                 [ ] Pass  [ ] Fail  Bug# ______

   <- (Remove) Selection Button              [ ] Pass  [ ] Fail  Bug# ______

   => (Add all) Selection Button             [ ] Pass  [ ] Fail  Bug# ______

   <= (Remove all) Selection Button          [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

   Open                                      [ ] Pass  [ ] Fail  Bug# ______

   Delete                                    [ ] Pass  [ ] Fail  Bug# ______

   Clone                                     [ ] Pass  [ ] Fail  Bug# ______
```

```
7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______


Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail


Bugs Reported:


Notes:



$Id: SpacecraftAttitudePanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Spacecraft Attitude panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Add and open the appropriate object panel.

   Add Spacecraft Resource                  [ ] Pass  [ ] Fail  Bug# ______

   Open Spacecraft Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Close Spacecraft Resource                [ ] Pass  [ ] Fail  Bug# ______

   Delete spacecraft Resource               [ ] Pass  [ ] Fail  Bug# ______

   Clone Spacecraft Resource                [ ] Pass  [ ] Fail  Bug# ______

   Select Attitude Tab                      [ ] Pass  [ ] Fail  Bug# ______
```

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Attitude Model Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Coordinate System Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Euler Angle Sequence Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Attitude State Type Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Attitude Elements Text Field			
Euler Angles Text Fields	[ ] Pass	[ ] Fail	Bug# _____
Quaternion Text Fields	[ ] Pass	[ ] Fail	Bug# _____
DCM Text Fields	[ ] Pass	[ ] Fail	Bug# _____
Attitude Rate State Type Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Attitude Rate Elements Text Field			
Euler Angle Rates Text Fields	[ ] Pass	[ ] Fail	Bug# _____
Angular Velocity Text Fields	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____

```
   Maximize                                [ ] Pass  [ ] Fail  Bug# ______

   Close                                   [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)               [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                        [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SpacecraftBallisticPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ___________  Build: ___________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: __________________

Description:

This test validates the functionality of the Spacecraft Ballistic/Mass panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Spacecraft Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Open Spacecraft Resource                [ ] Pass  [ ] Fail  Bug# ______

   Close Spacecraft Resource               [ ] Pass  [ ] Fail  Bug# ______
```

Delete Spacecraft Resource [ ] Pass [ ] Fail Bug# _____

Clone Spacecraft Resource [ ] Pass [ ] Fail Bug# _____

Select Ballistic/Mass Tab [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

Dry Mass Text Field [ ] Pass [ ] Fail Bug# _____

Coefficient of Drag Text Field [ ] Pass [ ] Fail Bug# _____

Coefficient of Reflectivity Text Field [ ] Pass [ ] Fail Bug# _____

Drag Area Text Field [ ] Pass [ ] Fail Bug# _____

Solar Radiation Pressure Area Text Field [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete [ ] Pass [ ] Fail Bug# _____

Show Script [ ] Pass [ ] Fail Bug# _____

6. Evaluate panel control.

Tab Key Navigation [ ] Pass [ ] Fail Bug# _____

Cancel [ ] Pass [ ] Fail Bug# _____

Apply [ ] Pass [ ] Fail Bug# _____

OK (Save) [ ] Pass [ ] Fail Bug# _____

Help [DEFERRED]

Restore [ ] Pass [ ] Fail Bug# _____

Minimize [ ] Pass [ ] Fail Bug# _____

Maximize [ ] Pass [ ] Fail Bug# _____

Close [ ] Pass [ ] Fail Bug# _____

```
7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SpacecraftOrbitPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Spacecraft Orbit panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Spacecraft Resource                  [ ] Pass  [ ] Fail  Bug# ______

   Open Spacecraft Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Close Spacecraft Resource                [ ] Pass  [ ] Fail  Bug# ______

   Delete Spacecraft Resource               [ ] Pass  [ ] Fail  Bug# ______

   Clone Spacecraft Resource                [ ] Pass  [ ] Fail  Bug# ______
```

```
   Select Orbit Tab                          [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Epoch Format Combo Box                    [ ] Pass  [ ] Fail  Bug# ______

   Epoch Text Field                          [ ] Pass  [ ] Fail  Bug# ______

   Coordinate System Combo Box               [ ] Pass  [ ] Fail  Bug# ______

   State Type Combo Box                      [ ] Pass  [ ] Fail  Bug# ______
   Anomaly Type if Keplerian                 [ ] Pass  [ ] Fail  Bug# ______

   Elements Text Field                       [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.
```

```
   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:



$Id: SpacecraftTanksPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the Spacecraft Tanks panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add Spacecraft Resource                  [ ] Pass  [ ] Fail  Bug# ______

   Add Tank Resource                        [ ] Pass  [ ] Fail  Bug# ______

   Open Spacecraft Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Select Tanks Tab                         [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______
```

3. Evaluate the individual panel elements.

Available Tanks List	[ ] Pass	[ ] Fail	Bug# _____
Selected Tanks List	[ ] Pass	[ ] Fail	Bug# _____
-> (Add) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
<- (Remove) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
=> (Add all) Selection Button	[ ] Pass	[ ] Fail	Bug# _____
<= (Remove all) Selection Button	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____
Maximize	[ ] Pass	[ ] Fail	Bug# _____
Close	[ ] Pass	[ ] Fail	Bug# _____
Open	[ ] Pass	[ ] Fail	Bug# _____
Clone	[ ] Pass	[ ] Fail	Bug# _____

7. Evaluate rename functionality.

Rename (on resource tree)	[ ] Pass	[ ] Fail	Bug# _____

```
Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail


Bugs Reported:


Notes:
```

10.26 fmincon Optimizer

```
$Id: SQPOptimizerPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _______________________________      Date: ____________________________


Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the SQP Optimizer panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Create and open the appropriate object panel.

   Create SQPOptimizer Resource              [ ] Pass  [ ] Fail  Bug# ______

   Open SQPOptimizer Resource                [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______


3. Evaluate the individual panel elements.
```

Tolerance Fun Text Field [ ] Pass [ ] Fail Bug# _____

Tolerance Con Text Field [ ] Pass [ ] Fail Bug# _____

Tolerance X Text Field [ ] Pass [ ] Fail Bug# _____

Maximum Fun Evaluations Text Field [ ] Pass [ ] Fail Bug# _____

Maximum Iterations Text Field [ ] Pass [ ] Fail Bug# _____

Differential Minimum Change Text Field [ ] Pass [ ] Fail Bug# _____

Differential Maximum Change Text Field [ ] Pass [ ] Fail Bug# _____

Display Combo Box [ ] Pass [ ] Fail Bug# _____

Gradient Object Check Box [ ] Pass [ ] Fail Bug# _____

Gradient Constraint Check Box [ ] Pass [ ] Fail Bug# _____

Derivative Check Check Box [ ] Pass [ ] Fail Bug# _____

Diagnostics Check Box [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete [ ] Pass [ ] Fail Bug# _____

Show Script [ ] Pass [ ] Fail Bug# _____

6. Evaluate panel control.

Tab Key Navigation [ ] Pass [ ] Fail Bug# _____

Cancel [ ] Pass [ ] Fail Bug# _____

Apply [ ] Pass [ ] Fail Bug# _____

OK (Save) [ ] Pass [ ] Fail Bug# _____

Help [DEFERRED]

Restore [ ] Pass [ ] Fail Bug# _____

Minimize [ ] Pass [ ] Fail Bug# _____

Maximize [ ] Pass [ ] Fail Bug# _____

Close [ ] Pass [ ] Fail Bug# _____

```
7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.27 String

Panel.txt

10.28 Thruster

```
$Id: ThrusterPanel.txt,v 1.4 2006/11/22 21:32:34 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ______________________

            ___ Macintosh, OS X Version: _________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the Thruster panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.
```

Add Thruster Resource	[ ] Pass	[ ] Fail	Bug# _____
Open Thruster Resource	[ ] Pass	[ ] Fail	Bug# _____
Close Thruster Resource	[ ] Pass	[ ] Fail	Bug# _____
Clone Thruster Resource	[ ] Pass	[ ] Fail	Bug# _____
Rename Thruster Resource	[ ] Pass	[ ] Fail	Bug# _____
Delete Thruster Resource	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Coordinate System Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Origin Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Axes Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Element1 Field	[ ] Pass	[ ] Fail	Bug# _____
Element2 Field	[ ] Pass	[ ] Fail	Bug# _____
Element3 Field	[ ] Pass	[ ] Fail	Bug# _____
Duty Cycle	[ ] Pass	[ ] Fail	Bug# _____
Thruster Scale Factor Field	[ ] Pass	[ ] Fail	Bug# _____
Decrement Mass Button	[ ] Pass	[ ] Fail	Bug# _____
Tank Combo Box	[ ] Pass	[ ] Fail	Bug# _____
GravitationalAccel Field	[ ] Pass	[ ] Fail	Bug# _____
Edit Thruster Coefficient Action Button	[ ] Pass	[ ] Fail	Bug# _____
Edit Impulse Coefficient Action Button	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____

```
   Show Script                          [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                   [ ] Pass  [ ] Fail  Bug# ______

   Cancel                               [ ] Pass  [ ] Fail  Bug# ______

   Apply                                [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                            [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                              [ ] Pass  [ ] Fail  Bug# ______

   Minimize                             [ ] Pass  [ ] Fail  Bug# ______

   Maximize                             [ ] Pass  [ ] Fail  Bug# ______

   Close                                [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)            [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                           [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.29 Variable

```
$Id: VariablePanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: _________________________
```

```
Platform:   ___ Windows, Version: _____

            ___ Macintosh, OS X Version: _____

            ___ Linux, Distribution: _____

Description:

This test validates the functionality of the Variable panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create Variable Resource                 [ ] Pass  [ ] Fail  Bug# ______

   Open Variable Resource                   [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements. (* currently all read-only)

   Name Text Field                          [ ] Pass  [ ] Fail  Bug# ______

   Expression Text Field                    [ ] Pass  [ ] Fail  Bug# ______

   Color Selection Button                   [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______
```

```
   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)                [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

10.30 XY Plot

```
$Id: XYPlotPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: _________________________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: ________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the XYPlot panel.
(* indicates sub-panel whose functionality is tested separately)
```

Procedure:

1. Create and open the appropriate object panel.

   Create XYPlot Resource [ ] Pass [ ] Fail Bug# _____

   Open XYPlot Resource [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

   Create Action Button* [ ] Pass [ ] Fail Bug# _____

   Object Type Combo Box [ ] Pass [ ] Fail Bug# _____

   Object Combo Box [ ] Pass [ ] Fail Bug# _____

   Property Selection List [ ] Pass [ ] Fail Bug# _____

   Selected X List [ ] Pass [ ] Fail Bug# _____

   Selected Y List [ ] Pass [ ] Fail Bug# _____

   --> (Add) Selection Button (left) [ ] Pass [ ] Fail Bug# _____

   <-- (Remove) Selection Button (left) [ ] Pass [ ] Fail Bug# _____

   --> (Add) Selection Button (right) [ ] Pass [ ] Fail Bug# _____

   <-- (Remove) Selection Button (right) [ ] Pass [ ] Fail Bug# _____

   < = (Remove All) Selection Button (rt.) [ ] Pass [ ] Fail Bug# _____

   Show Plot Check Box [ ] Pass [ ] Fail Bug# _____

   Show Grid Check Box [ ] Pass [ ] Fail Bug# _____

   Draw Targeting Check Box [ ] Pass [ ] Fail Bug# _____

   Color Select Box [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete [ ] Pass [ ] Fail Bug# _____

```
   Show Script                          [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                   [ ] Pass  [ ] Fail  Bug# ______

   Cancel                               [ ] Pass  [ ] Fail  Bug# ______

   Apply                                [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                            [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                              [ ] Pass  [ ] Fail  Bug# ______

   Minimize                             [ ] Pass  [ ] Fail  Bug# ______

   Maximize                             [ ] Pass  [ ] Fail  Bug# ______

   Close                                [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on resource tree)            [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements       ______

   Total number of test elements        ______

   Test case status                     [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

Chapter 11

Commands

11.1 Achieve

11.2 Begin Finite Burn

```
$Id: BeginFiniteBurnCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the BeginFiniteBurn Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add BeginFiniteBurn Command              [ ] Pass  [ ] Fail  Bug# ______

   Open BeginFiniteBurn Command             [ ] Pass  [ ] Fail  Bug# ______

   Close BeginFiniteBurn Command            [ ] Pass  [ ] Fail  Bug# ______

   Clone BeginFiniteBurn Command            [ ] Pass  [ ] Fail  Bug# ______

   Rename BeginFiniteBurn Command           [ ] Pass  [ ] Fail  Bug# ______
```

```
   Delete BeginFiniteBurn Command          [ ] Pass  [ ] Fail  Bug# ______

   Insert After Command (17 options)       [ ] Pass  [ ] Fail  Bug# ______

   Insert Before Command (17 options)      [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                        [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Finite Burn Drop Down List              [ ] Pass  [ ] Fail  Bug# ______

   Spacecraft Box                          [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete           [ ] Pass  [ ] Fail  Bug# ______

   Show Script                             [ ] Pass  [ ] Fail  Bug# ______

   Command Summary (must run script)       [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                      [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                 [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                [ ] Pass  [ ] Fail  Bug# ______

   Close                                   [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.
```

```
   Rename (on mission tree)                 [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.3 Call Function

```
$Id: CallFunctionCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Call Function Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create Function Resource(s)              [ ] Pass  [ ] Fail  Bug# ______

   Create Call Function Command             [ ] Pass  [ ] Fail  Bug# ______

   Open Call Function Command               [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.
```

```
   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Output Parameter Select Dialog*           [ ] Pass  [ ] Fail  Bug# ______

   Function Combo Box                        [ ] Pass  [ ] Fail  Bug# ______

   Input Parameter Select Dialog*            [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

   Command Summary (must run script)         [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on mission tree)                  [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______
```

```
   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.4 End Finite Burn

```
$Id: EndFiniteBurnCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the EndFiniteBurn Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Add and open the appropriate object panel.

   Add EndFiniteBurn Command                 [ ] Pass  [ ] Fail  Bug# ______

   Open EndFiniteBurn Command                [ ] Pass  [ ] Fail  Bug# ______

   Close EndFiniteBurn Command               [ ] Pass  [ ] Fail  Bug# ______

   Clone EndFiniteBurn Command               [ ] Pass  [ ] Fail  Bug# ______

   Rename EndFiniteBurn Command              [ ] Pass  [ ] Fail  Bug# ______

   Delete EndFiniteBurn Command              [ ] Pass  [ ] Fail  Bug# ______

   Insert After Command (17 options)         [ ] Pass  [ ] Fail  Bug# ______

   Insert Before Command (17 options)        [ ] Pass  [ ] Fail  Bug# ______
```

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Finite Burn Drop Down List	[ ] Pass	[ ] Fail	Bug# _____
Spacecraft Box	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____
Command Summary (must run script)	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____
Maximize	[ ] Pass	[ ] Fail	Bug# _____
Close	[ ] Pass	[ ] Fail	Bug# _____

7. Evaluate rename functionality.

Rename (on mission tree)	[ ] Pass	[ ] Fail	Bug# _____

Summary:

Number of passed test elements	_____
Total number of test elements	_____

```
   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.5 GMAT Command

```
$Id: GMATCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ______________________

            ___ Macintosh, OS X Version: _________________

            ___ Linux, Distribution: ____________________

Description:

This test validates the functionality of the GMAT (assignment) Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create GMAT Command                       [ ] Pass  [ ] Fail  Bug# ______

   Open GMAT Command                         [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Assignment Text Box                       [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.
```

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____
Command Summary (must run script)	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____
Maximize	[ ] Pass	[ ] Fail	Bug# _____
Close	[ ] Pass	[ ] Fail	Bug# _____

7. Evaluate rename functionality.

Rename (on mission tree)	[ ] Pass	[ ] Fail	Bug# _____

Summary:

Number of passed test elements	_____
Total number of test elements	_____
Test case status	[ ] Pass [ ] Fail

Bugs Reported:

Notes:

11.6 If

$Id: IfCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____ Date: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the If Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create If Command [ ] Pass [ ] Fail Bug# _____

   Open If Command [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

   LHS Text Field [ ] Pass [ ] Fail Bug# _____

   LHS Parameter Select Dialog (rt-click)* [ ] Pass [ ] Fail Bug# _____

   If Condition Dialog (rt-click)* [ ] Pass [ ] Fail Bug# _____

   RHS Text Field [ ] Pass [ ] Fail Bug# _____

   RHS Parameter Select Dialog (rt-click)* [ ] Pass [ ] Fail Bug# _____

   Logic Selection Dialog (rt-click)* [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

```
   Data elements appear complete            [ ] Pass  [ ] Fail  Bug# ______

   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

   Command Summary (must run script)        [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on mission tree)                 [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.7 Maneuver

$Id: ManeuverCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____ Date: _____ Build: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Maneuver Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create Maneuver Command [ ] Pass [ ] Fail Bug# _____

   Open Maneuver Command [ ] Pass [ ] Fail Bug# _____

   Insert Before Command [ ] Pass [ ] Fail Bug# _____

   Insert After Command [ ] Pass [ ] Fail Bug# _____

   Delete Maneuver Command [ ] Pass [ ] Fail Bug# _____

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

   Burn Combo Box [ ] Pass [ ] Fail Bug# _____

   Spacecraft Combo Box [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete [ ] Pass [ ] Fail Bug# _____

```
   Show Script                              [ ] Pass  [ ] Fail  Bug# ______

   Command Summary (must run script)        [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                       [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                   [ ] Pass  [ ] Fail  Bug# ______

   Apply                                    [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                  [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                 [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                 [ ] Pass  [ ] Fail  Bug# ______

   Close                                    [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on mission tree)                 n/a

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                         [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.8 Minimize

```
$Id: MinimizeCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $
```

Tester: _____ Date: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Minimize Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

Create Minimize Command	[ ] Pass	[ ] Fail	Bug# _____
Open Minimize Command	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Solver Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Variable Text Field	[ ] Pass	[ ] Fail	Bug# _____
Choose Action Button (Par. Sel. Dialog)*	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____
Command Summary (must run script)	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____

```
   Cancel                                [ ] Pass  [ ] Fail  Bug# ______

   Apply                                 [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                             [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                               [ ] Pass  [ ] Fail  Bug# ______

   Minimize                              [ ] Pass  [ ] Fail  Bug# ______

   Maximize                              [ ] Pass  [ ] Fail  Bug# ______

   Close                                 [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on mission tree)              [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.9 Nonlinear Constraint

```
$Id: NonlinearConstraintCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _______________________________      Date: ____________________________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________
```

Description:

This test validates the functionality of the NonLinearConstraint Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

Create NonLinearConstraint Command	[ ] Pass	[ ] Fail	Bug# _____
Open NonLinearConstraint Command	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Optimizer Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Constraint Text Field	[ ] Pass	[ ] Fail	Bug# _____
Choose Action Button (Par. Sel. Dialog)*	[ ] Pass	[ ] Fail	Bug# _____
Operator Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Constraint Value Text Field	[ ] Pass	[ ] Fail	Bug# _____
Choose Action Button (Par. Sel. Dialog)*	[ ] Pass	[ ] Fail	Bug# _____
Tolerance Text Field	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____
Command Summary (must run script)	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____

```
   Apply                                  [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                              [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                [ ] Pass  [ ] Fail  Bug# ______

   Minimize                               [ ] Pass  [ ] Fail  Bug# ______

   Maximize                               [ ] Pass  [ ] Fail  Bug# ______

   Close                                  [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on mission tree)               [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                       [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.10 Propagate

```
$Id: PropagateCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: ______________________

            ___ Macintosh, OS X Version: _________________

            ___ Linux, Distribution: ____________________

Description:
```

```
This test validates the functionality of the Propagate Command panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Create and open the appropriate object panel.

   Create Propagate Command                 [ ] Pass  [ ] Fail  Bug# ______

   Open Propagate Command                   [ ] Pass  [ ] Fail  Bug# ______

   Insert Before Command                    [ ] Pass  [ ] Fail  Bug# ______

   Insert After Command                     [ ] Pass  [ ] Fail  Bug# ______

   Delete Propagate Command                 [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Propagate Mode Combo Box                 [ ] Pass  [ ] Fail  Bug# ______

   Propagate/SC/Backward Grid               [ ] Pass  [ ] Fail  Bug# ______

   Stopping Conditions Grid                 [ ] Pass  [ ] Fail  Bug# ______

   Stopping Condition Name Text Field       [ ] Pass  [ ] Fail  Bug# ______

   Update Action Button                     [ ] Pass  [ ] Fail  Bug# ______

   Delete Action Button                     [ ] Pass  [ ] Fail  Bug# ______

   Variable Text Field                      [ ] Pass  [ ] Fail  Bug# ______

   View Action Button (Par. Select Dialog)* [ ] Pass  [ ] Fail  Bug# ______

   Operator Combo Box                       [ ] Pass  [ ] Fail  Bug# ______

   Value Text Field                         [ ] Pass  [ ] Fail  Bug# ______

   View Action Button (Par. Select Dialog)* [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   a. Check SpaceObjects list:
      Create spacecraft SC1, SC2, and SC3
      Put SC1 and SC2 into Formation1
      Open up the Propagate command panel
      Open the Spacecraft List panel
       (i.e. SpaceObjectSelectDialog)
```

```
      Only Formation1 and SC3 should be in   [ ] Pass  [ ] Fail  Bug# ______
       the lists of SpaceObjects

5. Evaluate data.

   Data elements appear complete             [ ] Pass  [ ] Fail  Bug# ______

   Show Script                               [ ] Pass  [ ] Fail  Bug# ______

   Command Summary (must run script)         [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                        [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                    [ ] Pass  [ ] Fail  Bug# ______

   Apply                                     [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                                 [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                   [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                  [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                  [ ] Pass  [ ] Fail  Bug# ______

   Close                                     [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.

   Rename (on mission tree)                  [ ] Pass  [ ] Fail  Bug# ______

Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.11 Report

```
$Id: ReportCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Report Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create Report Command                    [ ] Pass  [ ] Fail  Bug# ______

   Open Report Command                      [ ] Pass  [ ] Fail  Bug# ______

   Insert Before Command                    [ ] Pass  [ ] Fail  Bug# ______

   Insert After Command                     [ ] Pass  [ ] Fail  Bug# ______

   Delete Report Command                    [ ] Pass  [ ] Fail  Bug# ______

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                         [ ] Pass  [ ] Fail  Bug# ______

3. Evaluate the individual panel elements.

   Report File Combo Box                    [ ] Pass  [ ] Fail  Bug# ______

   Paramenter List                          [ ] Pass  [ ] Fail  Bug# ______

   View Action Button                       [ ] Pass  [ ] Fail  Bug# ______

   Object Type Combo Box                    [ ] Pass  [ ] Fail  Bug# ______

   Select Entire Object Action Button       [ ] Pass  [ ] Fail  Bug# ______

   Object List Box                          [ ] Pass  [ ] Fail  Bug# ______

   Object Property List                     [ ] Pass  [ ] Fail  Bug# ______
```

```
   Selected Values List                    [ ] Pass  [ ] Fail  Bug# ______

   --> (Add) Selection Button              [ ] Pass  [ ] Fail  Bug# ______

   <-- (Remove) Selection Button           [ ] Pass  [ ] Fail  Bug# ______

   < = (Remove All) Selection Button       [ ] Pass  [ ] Fail  Bug# ______

   = > (Add All) Selection Button          [ ] Pass  [ ] Fail  Bug# ______

   Up Selection Button                     [ ] Pass  [ ] Fail  Bug# ______

   Down Selection Button                   [ ] Pass  [ ] Fail  Bug# ______

   OK Button                               [ ] Pass  [ ] Fail  Bug# ______

   Cancel Button                           [ ] Pass  [ ] Fail  Bug# ______

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete           [ ] Pass  [ ] Fail  Bug# ______

   Show Script                             [ ] Pass  [ ] Fail  Bug# ______

   Command Summary (must run script)       [ ] Pass  [ ] Fail  Bug# ______

6. Evaluate panel control.

   Tab Key Navigation                      [ ] Pass  [ ] Fail  Bug# ______

   Cancel                                  [ ] Pass  [ ] Fail  Bug# ______

   Apply                                   [ ] Pass  [ ] Fail  Bug# ______

   OK (Save)                               [ ] Pass  [ ] Fail  Bug# ______

   Help [DEFERRED]

   Restore                                 [ ] Pass  [ ] Fail  Bug# ______

   Minimize                                [ ] Pass  [ ] Fail  Bug# ______

   Maximize                                [ ] Pass  [ ] Fail  Bug# ______

   Close                                   [ ] Pass  [ ] Fail  Bug# ______

7. Evaluate rename functionality.
```

```
   Rename (on mission tree)                 n/a


Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.12 Save

```
$Id: SaveCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: _____________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: __________________


Description:

This test validates the functionality of the Save Command panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Create and open the appropriate object panel.

   Create Save Command                      [ ] Pass  [ ] Fail  Bug# ______

   Open Save Command                        [ ] Pass  [ ] Fail  Bug# ______

   Insert Before Command                    [ ] Pass  [ ] Fail  Bug# ______

   Insert After Command                     [ ] Pass  [ ] Fail  Bug# ______

   Delete Save Command                      [ ] Pass  [ ] Fail  Bug# ______
```

2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics [ ] Pass [ ] Fail Bug# _____

3. Evaluate the individual panel elements.

   Objects to Save Check Box(es) [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete [ ] Pass [ ] Fail Bug# _____

   Show Script [ ] Pass [ ] Fail Bug# _____

   Command Summary (must run script) [ ] Pass [ ] Fail Bug# _____

6. Evaluate panel control.

   Tab Key Navigation [ ] Pass [ ] Fail Bug# _____

   Cancel [ ] Pass [ ] Fail Bug# _____

   Apply [ ] Pass [ ] Fail Bug# _____

   OK (Save) [ ] Pass [ ] Fail Bug# _____

   Help [DEFERRED]

   Restore [ ] Pass [ ] Fail Bug# _____

   Minimize [ ] Pass [ ] Fail Bug# _____

   Maximize [ ] Pass [ ] Fail Bug# _____

   Close [ ] Pass [ ] Fail Bug# _____

7. Evaluate rename functionality.

   Rename (on mission tree) n/a

Summary:

   Number of passed test elements _____

   Total number of test elements _____

```
   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

11.13 Script

11.14 Target

```
$Id: TargetCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____________  Date: ____________  Build: ____________

Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________

Description:

This test validates the functionality of the Target Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

   Create Target Command                     [ ] Pass  [ ] Fail  Bug# ______

   Open Target Command                       [ ] Pass  [ ] Fail  Bug# ______

   Insert Before Command                     [ ] Pass  [ ] Fail  Bug# ______

   Insert After Command                      [ ] Pass  [ ] Fail  Bug# ______

   Delete Target Command                     [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                          [ ] Pass  [ ] Fail  Bug# ______
```

3. Evaluate the individual panel elements.

Solver Name Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Solver Mode Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Exit Mode Combo Box	[ ] Pass	[ ] Fail	Bug# _____
Apply Correction Action Button	[ ] Pass	[ ] Fail	Bug# _____

4. Evaluate panel-specific functionality.

N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____
Command Summary (must run script)	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____
Maximize	[ ] Pass	[ ] Fail	Bug# _____
Close	[ ] Pass	[ ] Fail	Bug# _____

7. Evaluate rename functionality.

Rename (on mission tree)	[ ] Pass	[ ] Fail	Bug# _____

Summary:

Number of passed test elements	_____

```
   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail


Bugs Reported:


Notes:
```

11.15 Toggle

```
$Id: ToggleCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $


Tester: _____________  Date: ____________  Build: ____________


Platform:   ___ Windows, Version: ________________________

            ___ Macintosh, OS X Version: _____________

            ___ Linux, Distribution: ___________________


Description:

This test validates the functionality of the Toggle Command panel.
(* indicates sub-panel whose functionality is tested separately)


Procedure:

1. Create and open the appropriate object panel.

   Create Toggle Command                   [ ] Pass  [ ] Fail  Bug# ______

   Open Toggle Command                     [ ] Pass  [ ] Fail  Bug# ______

   Insert Before Command                   [ ] Pass  [ ] Fail  Bug# ______

   Insert After Command                    [ ] Pass  [ ] Fail  Bug# ______

   Delete Toggle Command                   [ ] Pass  [ ] Fail  Bug# ______


2. Evaluate the aesthetic qualities of the panel.

   Panel Aesthetics                        [ ] Pass  [ ] Fail  Bug# ______
```

3. Evaluate the individual panel elements.

   Select Subscriber to Toggle Check Box(es) [ ] Pass [ ] Fail Bug# _____

   On/Off Radio Button [ ] Pass [ ] Fail Bug# _____

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

   Data elements appear complete [ ] Pass [ ] Fail Bug# _____

   Show Script [ ] Pass [ ] Fail Bug# _____

   Command Summary (must run script) [ ] Pass [ ] Fail Bug# _____

6. Evaluate panel control.

   Tab Key Navigation [ ] Pass [ ] Fail Bug# _____

   Cancel [ ] Pass [ ] Fail Bug# _____

   Apply [ ] Pass [ ] Fail Bug# _____

   OK (Save) [ ] Pass [ ] Fail Bug# _____

   Help [DEFERRED]

   Restore [ ] Pass [ ] Fail Bug# _____

   Minimize [ ] Pass [ ] Fail Bug# _____

   Maximize [ ] Pass [ ] Fail Bug# _____

   Close [ ] Pass [ ] Fail Bug# _____

7. Evaluate rename functionality.

   Rename (on mission tree) [ ] Pass [ ] Fail Bug# _____

Summary:

   Number of passed test elements _____

   Total number of test elements _____

   Test case status [ ] Pass [ ] Fail

Bugs Reported:

Notes:

11.16 Vary

$Id: VaryCmdPanel.txt,v 1.4 2006/11/22 21:32:33 dconway Exp $

Tester: _____ Date: _____ Build: _____

Platform: ___ Windows, Version: _____

___ Macintosh, OS X Version: _____

___ Linux, Distribution: _____

Description:

This test validates the functionality of the Vary Command panel.
(* indicates sub-panel whose functionality is tested separately)

Procedure:

1. Create and open the appropriate object panel.

Create Vary Command	[ ] Pass	[ ] Fail	Bug# _____
Open Vary Command	[ ] Pass	[ ] Fail	Bug# _____
Insert Before Command	[ ] Pass	[ ] Fail	Bug# _____
Insert After Command	[ ] Pass	[ ] Fail	Bug# _____
Delete Vary Command	[ ] Pass	[ ] Fail	Bug# _____

2. Evaluate the aesthetic qualities of the panel.

Panel Aesthetics	[ ] Pass	[ ] Fail	Bug# _____

3. Evaluate the individual panel elements.

Solver Combo Box	[ ] Pass	[ ] Fail	Bug# _____

Variable Text Field	[ ] Pass	[ ] Fail	Bug# _____	
View Action Button (Par. Select Dialog)*	[ ] Pass	[ ] Fail	Bug# _____	
Initial Value Text Field	[ ] Pass	[ ] Fail	Bug# _____	
Perturbation Text Field	[ ] Pass	[ ] Fail	Bug# _____	
Lower Text Field	[ ] Pass	[ ] Fail	Bug# _____	
Upper Text Field	[ ] Pass	[ ] Fail	Bug# _____	
Max Step Text Field	[ ] Pass	[ ] Fail	Bug# _____	
Additive Scale Factor Text Field	[ ] Pass	[ ] Fail	Bug# _____	
Multiplicative Scale Factor Text Field	[ ] Pass	[ ] Fail	Bug# _____	

4. Evaluate panel-specific functionality.

   N/A

5. Evaluate data.

Data elements appear complete	[ ] Pass	[ ] Fail	Bug# _____
Show Script	[ ] Pass	[ ] Fail	Bug# _____
Command Summary (must run script)	[ ] Pass	[ ] Fail	Bug# _____

6. Evaluate panel control.

Tab Key Navigation	[ ] Pass	[ ] Fail	Bug# _____
Cancel	[ ] Pass	[ ] Fail	Bug# _____
Apply	[ ] Pass	[ ] Fail	Bug# _____
OK (Save)	[ ] Pass	[ ] Fail	Bug# _____
Help [DEFERRED]			
Restore	[ ] Pass	[ ] Fail	Bug# _____
Minimize	[ ] Pass	[ ] Fail	Bug# _____
Maximize	[ ] Pass	[ ] Fail	Bug# _____
Close	[ ] Pass	[ ] Fail	Bug# _____

7. Evaluate rename functionality.

Rename (on mission tree)	n/a

```
Summary:

   Number of passed test elements      ______

   Total number of test elements       ______

   Test case status                          [ ] Pass  [ ] Fail

Bugs Reported:


Notes:
```

Bibliography

[Black] Rex Black, "Managing the Testing Process," Second Edition, Wiley Publishing, 2002.

[Craig] Rick D. Craig and Stefan P. Jaskiel, "Systematic Software Testing," Artech House, 2002.

[Dove] Edwin G. Dove, "GUI Philosophy for the General Mission Analysis Tool (GMAT)."

[MTP] GMAT Test Team, "General Mission Analysis Tool (GMAT) Master Test Plan."

[GDT] GMAT Development Team, "General Mission Analysis Tool (GMAT) Architectural Specification."

[hughes] Steven P. Hughes, "General Mission Analysis Tool (GMAT) Mathematical Specification."

[hughes2] Steven P. Hughes, "General Mission Analysis Tool (GMAT) User's Guide."

[matlab] The MathWorks, Inc, "MATLAB", available from http://www.mathworks.com.

[OOo] OpenOffice.org, "OpenOffice", available from http://www.openoffice.org/.

[opttools] The MathWorks, Inc, "Optimization Toolbox", available from http://www.mathworks.com.