Mission Operations and Mission Services

NASA Code S Space Sciences Solar and Heliospheric Observatory (SOHO) Pass Operations Logging and Anomaly Reporting Interface System (POLARIS) Test Plan and Procedures

Effective: April xx, 2007

Expiration Date: TBD

Honeywell Technology Solutions Inc.
Goddard Corporate Park
Lanham, Maryland 20706-2291

Signature Page

Honeywell Technology Solution, Inc.
Lanham, Maryland 20706-2291

Preface

This document details the test plan and procedures for acceptance testing the new Pass Operations Logging and Anomaly Reporting Interface System (POLARIS) for the Solar and Heliospheric Observatory (SOHO). This work falls under the responsibility of the Mission Operations and Mission Services (MOMS) contract for the National Aeronautics and Space Administration (NASA) at Goddard Space Flight Center (GSFC) as part of Task Order #6.

Comments or questions concerning this document shall be addressed to:

TBD

Change Information Page

List of Effective Pages		
Page Number	**Version**	**Nature of Change**
Cover	Original	
Signature Page	Original	
Preface	Original	
Change Information Page	Original	
TBD	Original	

Document History				
Document Number	**Version - Change**	**Effective Date**	**Expiration Date**	**CCR Number**
TBD	Original	April X, 2007	TBD	NA

Contents

Preface ..... **iii**

1 Document Overview **1**

2 Test Approach **2**

2.1 General 2
2.2 System Testing 2
2.3 Acceptance Testing 2
2.3.1 General 2
2.3.2 Phase 1: POLARIS Component Testing 2
2.3.3 Phase 2: Testing with the SOHO Simulator 2
2.3.4 Phase 3: Primary Operations Testing 3

3 Acceptance Test Schedule **4**

3.1 General 4
3.2 Test Assumptions 5
3.3 Test Constraints 5
3.4 Schedule Impacts 5

4 Test Resources **6**

4.1 General 6
4.2 Test Environment 6
4.3 Test Team 6
4.4 Test Documentation and Other Applicable Documents 7

5 Test Plan and Procedures **8**

5.1 Test Case 1.0: PTI Tests 8
5.2 Test Case 2.0: Pass Generator Tests 12
5.3 Test Case 3.0: Test of ANS Alert System 15
5.4 Test Case 4.0: Test of Automation Scripts 20
5.5 Test Case 5.0: Test of Operational Scenarios 21

Appendix A: Planned Test Order **24**

Appendix B: Test Verification Matrix **25**

Appendix C: Requirements Not Mapped to Test Cases.....28

Appendix D: Automated TSTOL Procedures Flight-test Dates 29

Appendix E: Abbreviations and Acronyms.....32

List of Tables

Table 1. Acceptance Test Schedule 4
Table 2. Test Team.....6

SOHO POLARIS Test Plan & Procedures

1 Document Overview

This document details the Solar Heliospheric Observatory (SOHO) test plan and procedures for acceptance testing the new Pass Operations Logging and Anomaly Reporting Interface System (POLARIS) for readiness to support SOHO automated operations at Goddard Space Flight Center (GSFC). The plan itself is shown in Appendix B as a matrix that maps the planned tests to the new requirements being verified. High-level test procedures were documented when tests could be covered by existing operational procedures or work instructions. Detailed procedures were documented only when operational procedures were new or changed significantly. This plan also documents the test approach, schedule, and resources for this acceptance test effort.

2 Test Approach

2.1 General

The testing for SOHO POLARIS will be conducted in two parts. The first part will be system testing conducted by the software team. This testing will be conducted in a development environment in the IMOC. The second part will be acceptance testing conducted by the Flight Operations Team (FOT) in the operational environment.

2.2 System Testing

The software team is responsible for system testing conducted in the development environment. This testing will be similar to that performed by the FOT during phase 1 of acceptance testing, and is intended to ensure the automation software is ready to enter acceptance testing.

2.3 Acceptance Testing

2.3.1 General

The acceptance testing will be conducted in the operational environment by the FOT. The acceptance testing has been divided into three test phases. These test phases have been defined to minimize risk to nominal operations and provide the greatest confidence for success. Successful testing during each phase is generally required before proceeding to the next test phase, although there may be some overlap between phases for different components of the automated system. The test verification matrix in Appendix B will be used to identify which tests must be conducted successfully during each test phase. Problems identified during acceptance testing will be documented as Discrepancy Reports (DR)s using the Comprehensive Discrepancy System (CDS). For additional information on the planned test order during each test phase, refer to Appendix A.

2.3.2 Phase 1: POLARIS Component Testing

This test phase will independently test the various components of POLARIS. These include the TPOCC Interface (PTI), the Pass Generator Tool, the automated *Attention!* Notification System (ANS), and several tools/scripts required for POLARIS operability. Some tests in this phase will be performed concurrently as required to confirm interoperability between components.

2.3.3 Phase 2: Testing with the SOHO Simulator

This test phase will locally verify routine operational scenarios and procedures using the SOHO simulator. This is intended to minimize risk to nominal operations during the primary operations testing in the final test phase. Before transitioning to the primary operations test phase, the following conditions must be met: 1) Procedures for falling back to the current system in less than TBD minutes must be fully documented. 2) The TSM must perform a full system backup. 3) The SOHO Configuration Control Board (CCB) must give approval. These conditions will assure that fallback procedures are in place before any configuration changes are made to the operational string used for testing.

2.3.4 Phase 3: Primary Operations Testing

This final test phase of end-to-end testing will include bringing up the operational string with the automated software, configuring for various types of passes with the spacecraft, and performing commanding from the TPOCC to the spacecraft, and processing/responding to telemetry in the IMOC from the spacecraft. These tests will be performed with the FOT closely monitoring the automated system and a second operational string in hot backup. If any problem occurs with the automated software, the FOT will stop the automation and switch the backup string to primary to resume nominal manned operations as usual. Testing will be considered finished once the FOT is satisfied with all test results and the Mission Director and SOHO CCB have approved the software for operational use.

3 Acceptance Test Schedule

3.1 General

The acceptance test schedule shown in Table 1 is based on the assumptions detailed in the next section, with the understanding that unforeseen events and imposed test constraints can affect the schedule. With this in mind, the test team proposes a TBD-week acceptance test period with extended testing done during the final test phase. The test schedule also includes the planned release schedule for delivering DR fixes.

Table 1. Acceptance Test Schedule

Test Phase	Test Activity	Test Period	Test Week(s)
Phase 1 – POLARIS Component Testing			
	Proceed to next phase	NA	NA
Phase 2 – Testing with the SOHO Simulator			
	Proceed to next phase	NA	NA
Phase 3 – Primary Operations Testing	Operational Acceptance Testing		
	SOHO CCB Approval to Transition	NA	NA

3.2 Test Assumptions

TBD

3.3 Test Constraints

Test constraints limit test activities and can be imposed by the customer, the environment or the test process itself. The following is a list of known test constraints that could impact the schedule.

a. Test cases identified for completion during each test phase must be passed before proceeding to the next test phase.

b. Availability of operational resources for testing will determine how closely the test environment mirrors the operational environment.

c. Internal testing will be limited by the capabilities of the SOHO simulator and how closely the interface to the simulator emulates a ground station.

d. COBS TCM in Macros patch and Reaction Wheel Speed Limits Update patch uplinked to spacecraft and performing as expected.

e. DSN station involvement TBD

3.4 Schedule Impacts

Since specific schedule impacts cannot be foreseen, a general listing of the types of impacts that may occur are given below.

a. Availability of the MOC backup string and other operational resources for testing.

b. FOT operational activities that take priority over testing activities.

c. Holidays and vacations that coincide with the test schedule.

d. Staffing issues that limit both time and resources for testing.

e. Additional training required for new operational procedures.

f. Spacecraft emergencies or anomalies that require FOT intervention or analysis.

g. Additional software releases needed to fix DRs found during later test phases.

h. Critical problems that limit testing and require unplanned patch releases.

i. Delivered, software fixes that do not fully resolve documented problems.

j. System and/or hardware problems beyond the scope of the application being tested.

k. Problems that require significant analysis to document or test.

l. Availability of DSN Resources for testing – Testing with the DSN must be coordinated so as to not affect nominal operational services.

m. Availability/functionality of planned patches to the SOHO COBS.

4 Test Resources

4.1 General

This section details the resources available to support this acceptance effort. Additional resources may be identified and used as needs arise.

4.2 Test Environment

The IMOC will be used for all acceptance testing.

4.3 Test Team

The test team is a cross-functional team that includes support from multiple organizations. The acceptance testing will be performed by the FOT with direction and assistance from the test manager, however a successful test effort requires the support of all team members.

Table 2. Test Team (TBD)

Mission Director	Bob Dutilly
Task Lead	
Software Manager	
Software Lead	
Systems Engineer	
Test Manager	
SOHO Ground System Engineer	
SOHO Ground System Engineer	
SOHO FOT	
TSM	

4.4 Test Documentation and Other Applicable Documents

This document was developed using the documents below. Other documents were either referenced during test planning activities or may be referenced during test execution.

a. Pass Operations Logging and Anomaly Reporting Interface System (POLARIS) for the Solar and Heliospheric Observatory (SOHO) Requirements Document, Version 1, July 12, 2006

b. Pass Operations Logging and Anomaly Reporting Interface System (POLARIS) for the Solar and Heliospheric Observatory (SOHO) Detailed Design Specification, Version 2, April 23, 2007

c. *Attention!* Notification System Software User's Guide, Online version

5 Test Plan and Procedures

All planned test cases have been organized into a test verification matrix to best show all planned tests and clearly demonstrate test coverage. The complete verification matrix that will be used for reporting test status is shown in Appendix B. Detailed test procedures have been documented only for those test cases that include new or significantly modified operational procedures. Refer to the SOHO Flight Operations Plan, for test cases that follow standard operating procedures. The need for additional test cases may be identified as testing progresses. Details of such additional testing will be captured in test logs and a test database.

5.1 Test Case 1.0: PTI Tests

Description: This test scenario will verify that the POLARIS TPOCC Interface and the PTI GUI startup, run, and interface properly with TPOCC. These tests will also serve to validate the various PTI scripts and confirm no adverse impact to nominal TPOCC string operations.

Success Criteria: The PTI and GUI must run and execute all scripts as expected. Routine TPOCC string activities for both commanding and telemetry continue to function nominally.

Test Procedures:

Test Case 1.1: PTI Startup		
Step/Action	Expected Results	Comments
1. Start the TPOCC software by running the "start_tpocc.pl" script	Script completes.	
2. Start PTI by running the "PTI.pl" script in a terminal window, or double-clicking the icon in the actions window.	Script completes.	
3. Open a new terminal window and run the command " “ps -ef | grep –v grep | grep PTI”	PTI process will be displayed.	
4. Verify user "polaris" ran the configured startup procedures.	Event window will display events indicating procedures run by user "polaris".	
5. Verify the file /home/soho/ops/POLARIS/logs exists.	File contains startup messages for the current time.	
6. Start a procedure in the TSTOL window.	TSTOL procedure and directives execute nominally.	

Test Case 1.2: PTI Interface to TPOCC		
Step/Action	Expected Results	Comments
1. Start the TPOCC software by running the "start_tpocc.pl" script	Script completes and activity is logged in the file /home/soho/ops/POLARIS/ log/PTI.log.	
2. Start PTI by running the "PTI.pl" script in a terminal window, or double-clicking the icon in the actions window.	Script completes and activity is logged in the file /home/soho/ops/POLARIS/ log/PTI.log.	
3. Start the TSTOL procedure "o_sohoup" by entering the following command in a terminal window: "PTI_client.pl 's o_sohoup' ".	Event log indicates TSTOL procedure "o_sohoup" started.	No authentication should be required, because the connection is to localhost, not a remote system.
4. Send a command load by entering the following command in a terminal window: "PTI_client.pl '/load *loadname*' ".	Event log indicates load entered in buffer and uplinked.	No authentication should be required, because the connection is to localhost, not a remote system
5. In the fot crontab, define the following command to run 5 minutes from the current time "PTI_client.pl 'sv 1+1' ".	Command executes at the desired time.	
6. Start a TSTOL procedure with a syntax error by entering the following command in a terminal window: "PTI_client.pl 's *procname'* ".	PTI runs the desired script to identify a syntax error has occurred.	

Test Case 1.2: PTI Interface to TPOCC		
Step/Action	Expected Results	Comments
7. Pause PTI by entering the terminal window command: "PTI_client.pl PAUSEPTI" and then enter the directive: "PTI_client.pl 'sv 1+1' ".	PTI pauses and does NOT issue the directive to TPOCC.	Event log indicates "PAUSEPTI" was recognized as PTI internal command, and also logs error msg "PTI server returned non-success response!"
8. Resume PTI by entering the terminal window command: "PTI_client.pl RESUMEPTI" and then enter the directive: "PTI_client.pl 'sv 1+1' ".	Directive is issued to TPOCC and executed.	Event log indicates "RESUMEPTI" was recognized as PTI internal command.
9. Exit PTI by entering the terminal window command: "PTI_client.pl EXITPTI"	PTI exits and PTI process is no longer displayed in process list for the workstation.	Event log indicates "EXITPTI" was recognized as PTI internal command.

Test Case 1.3: PTI GUI		
Step/Action	Expected Results	Comments
1. Verify no other GUIs are running.	TBD	TBD
2. Start PTI GUI by running PTIGUI.pl or double-clicking on the PTI GUI icon in the actions window.	PTI GUI should appear.	
3. Click on the "FC", "CC", and "MC" buttons, one at a time.	After clicking each privilege button, the color should change from red to green.	
4. Pause PTI by clicking on the "Running" button.	The button color should change to yellow, and the text should change to "Paused".	Event log indicates "PAUSEPTI" was recognized as PTI internal command.
5. From the command line, enter the directive: "PTI_client.pl 'sv 1+1' ".	PTI does NOT issue the directive to TPOCC	Event log indicates error msg "PTI server returned non-success response!"
6. Resume PTI by clicking on the "Paused" button.	The button color should change to green, and the text should change to " Running".	Event log indicates "RESUMEPTI" was recognized as PTI internal command.
7. Exit PTI by clicking on the "Exit PTI" button and click on "Exit" button to confirm.	Confirmation dialog should appear and the PTI log (not the log in the GUI) should indicate that PTI exited. The GUI should then close.	
8. Repeat step 2 to start multiple PTI GUIs.	Multiple PTI GUIs should appear.	
9. Repeat steps 3-7 in second GUI.	Both GUIs should function identically (i.e., when PTI is paused, both should indicate "Paused", etc.)	

5.2 Test Case 2.0: Pass Generator Tests

Description: This test scenario will verify the ability of the pass generator tool to build pass procedures for all types of passes expected to be run in automated mode.

Success Criteria: Each pass procedure generated by the pass generator tool must contain all required activities for each type of pass. Additionally, the pass procedures must be free of syntax errors and run all called procedures without problems.

Test Procedures:

Test Case 2.1: Build Pass Procedures for all types of night non-ranging passes:		
Step/Action	Expected Results	Comments
1. Long pass -From short gap -Into short gap	Final pass procedure contains all required activities as verified by OE.	Long pass (full dump and MDI-M/H transitions) Short gap (less than 3 hours)
2. Short pass -From short gap -Into short gap	Final pass procedure contains all required activities as verified by OE.	Short pass (insufficient for full dump) Short gap (less than 3 hours)
3. Long pass -From short gap -Into long gap	Final pass procedure contains all required activities as verified by OE.	Long pass (full dump and MDI-M/H transitions) Short gap (less than 3 hours) Long gap (more than 5 hours)
4. Short pass -From short gap -Into long gap	Final pass procedure contains all required activities as verified by OE.	Short pass (insufficient for full dump) Short gap (less than 3 hours) Long gap (more than 5 hours)
5. Long pass -From long gap -Into short gap	Final pass procedure contains all required activities as verified by OE.	Long pass (full dump and MDI-M/H transitions) Short gap (less than 3 hours) Long gap (more than 5 hours)
6. Short pass -From long gap -Into short gap	Final pass procedure contains all required activities as verified by OE.	Short pass (insufficient for full dump) Short gap (less than 3 hours) Long gap (more than 5 hours)

Test Case 2.1: Build Pass Procedures for all types of night non-ranging passes:		
Step/Action	Expected Results	Comments
7. Long pass -From long gap -Into long gap	Final pass procedure contains all required activities as verified by OE.	Long pass (full dump and MDI-M/H transitions) Long gap (more than 5 hours)
8. Short pass -From long gap -Into long gap	Final pass procedure contains all required activities as verified by OE.	Short pass (insufficient for full dump) Long gap (more than 5 hours)

Test Case 2.2: Build Pass Procedures for all types of night ranging passes		
Step/Action	Expected Results	Comments
1. Repeat all steps from Test Case 2.1, but include ranging activities on all passes.	Final pass procedure contains all required activities as verified by OE.	

Test Case 2.3: Add SVM activities to daytime pass procedures		
Step/Action	Expected Results	Comments
1. Build pass procedure for daytime non-ranging pass.	Basic pass procedure is built.	
2. Add SVM activities to pass procedure.	Final pass procedure contains all required activities as verified by OE.	
3. Build pass procedure for daytime ranging pass.	Basic pass procedure is built, containing ranging activities.	
4. Repeat step 2 for pass procedure built in step 3.	Final pass procedure contains all required activities as verified by OE.	
5. Attempt to add SVM procedures not included in automated procedure list.	TBD	Currently no software method is in place to prevent this. Responsibility lies with OE, so this test step may need to be deleted.

Test Case 2.4: Build Pass Procedures for all types of station handovers:		
Step/Action	Expected Results	Comments
1. Non-ranging to non-ranging handover.	Final pass procedure contains all required activities as verified by OE.	
2. Non -ranging to ranging handover.	Final pass procedure contains all required activities as verified by OE.	
3. Ranging to non-ranging handover.	Final pass procedure contains all required activities as verified by OE.	
4. Ranging to ranging handover.	Final pass procedure contains all required activities as verified by OE.	

Test Case 2.5: Build Pass Procedures for other types of passes:		
Step/Action	Expected Results	Comments
1. Pass with different uplink and downlink stations.	Final pass procedure contains all required activities as verified by OE.	
2. Downlink-only pass.	Final pass procedure contains all required activities as verified by OE.	
3. Pass with temporary limit changes.	Final pass procedure contains all required activities as verified by OE.	
4. Pass with automation starting mid-pass.	Final pass procedure contains all required activities as verified by OE.	
5. Pass requiring pass procedure to initiate a script.	TBD	

5.3 Test Case 3.0: Test of ANS Alert System

Description: This test scenario will verify the ANS interfaces to the TPOCC and the ability to detect various types of anomalies and limit violations. These test cases will include multiple external communication methods and notification escalation scenarios.

Success Criteria: All anomalous situations must be detected and notifications made to appropriate response personnel within specified timelines.

Test Procedures:

Test Case 3.1: ANS Violation Detection Tests		
Step/Action	Expected Results	Comments
1. Verify *Attention* services are running.	All Windows services required are running.	Services expected are *"Attention Notification System", "Attention Alarm Manager", "Attention Web Interface",* and one or more collector agents.
2. Start TPOCC software and start TSTOL procedure "o_sohoup".	TPOCC software starts. Procedure configures TPOCC for a pass and completes with no errors.	Procedure "o_sohoup" must include 1) call to EVTRPT to log events to a file in realtime, and 2) call script "log_monitor.pl".
3. Using the *"Attention Alarm Manager"* interface, view the Activity Log.	Event messages from TPOCC are visible in the Activity Log.	Only messages generated after "log monitor.pl" was started will be seen.
4. Trigger Red Low limit violation.	The action defined for a Red Low limit violation executes.	
5. Trigger Red High limit violation.	The action defined for a Red High limit violation executes.	
6. Trigger Yellow Low limit violation.	The action defined for a Yellow Low limit violation executes.	
7. Trigger Yellow High limit violation.	The action defined for a Yellow High limit violation executes.	
8. Trigger configuration monitor violation.	The action defined for a configuration monitor violation executes.	

Test Case 3.1: ANS Violation Detection Tests		
Step/Action	Expected Results	Comments
9. Trigger ACU software anomaly.	The action defined for an ACU software anomaly executes.	
10. Trigger COBS software anomaly.	The action defined for a COBS software anomaly executes.	
11. Trigger a hung TSTOL procedure.	The action defined for a hung TSTOL procedure executes.	

Test Case 3.2: ANS Individual Pager Notification and Response Tests		
Step/Action	Expected Results	Comments
1. Verify *Attention* services are running.	All Windows services required are running.	Services expected are *"Attention Notification System", "Attention Alarm Manager", "Attention Web Interface",* and one or more collector agents.
2. Start TPOCC software and start TSTOL procedure "o_sohoup".	TPOCC software starts. Procedure configures TPOCC for a pass and completes with no errors.	Procedure "o_sohoup" must include 1) call to EVTRPT to log events to a file in realtime, and 2) call script "log_monitor.pl".
3. Using the *"Attention Alarm Manager"* interface, view the Activity Log.	Event messages from TPOCC are visible in the Activity Log.	Only messages generated after "log monitor.pl" was started will be seen.
4. Trigger violation to initiate single pager notification.	*"Attention NS"* Action window shows expected action has been executed, and is waiting for a response. Target receives page with expected message.	
5. Login to the *"Attention NS"* web page and respond to the action specified in the pager message by setting the state of that action to 0.	The action is removed from the *"Attention NS"* Action window. The *NS* audit log shows that Event ID N was changed to state 0.	State 0 means the event is acknowledged; i.e., no further action required.

Test Case 3.2: ANS Individual Pager Notification and Response Tests		
Step/Action	Expected Results	Comments
6. Repeat step 4.	*"Attention NS"* Action window shows expected action has been executed, and is waiting for a response. Target receives page with expected message.	
7. Do NOT issue response to pager notification. Wait longer than the configured escalation time for that action.	The next person in the escalation chain should be notified.	
8. Repeat step 7.	After each configured timeout, the next person in the escalation chain should be notified.	

Test Case 3.3: ANS Group Pager Notification and Response Tests		
Step/Action	Expected Results	Comments
1. Verify *Attention* services are running.	All Windows services required are running.	Services expected are *"Attention Notification System", "Attention Alarm Manager", "Attention Web Interface",* and one or more collector agents.
2. Start TPOCC software and start TSTOL procedure "o_sohoup".	TPOCC software starts. Procedure configures TPOCC for a pass and completes with no errors.	Procedure "o_sohoup" must include 1) call to EVTRPT to log events to a file in realtime, and 2) call script "log_monitor.pl".
3. Trigger violation to initiate group pager notification.	*"Attention NS"* Action window shows expected action has been executed, and is waiting for a response. Entire target group receives page with expected message.	

Test Case 3.3: ANS Group Pager Notification and Response Tests		
Step/Action	Expected Results	Comments
4. Login to the *"Attention NS"* web page and respond to the action specified in the pager message by setting the state of that action to 0.	The action is removed from the *"Attention NS"* Action window. The *NS* audit log shows that Event ID N was changed to state 0.	State 0 means the event is acknowledged; i.e., no further action required.

Test Case 3.4: ANS Email Notification Tests		
Step/Action	Expected Results	Comments
1. Verify *Attention* services are running.	All Windows services required are running.	Services expected are *"Attention Notification System", "Attention Alarm Manager", "Attention Web Interface",* and one or more collector agents.
2. Start TPOCC software and start TSTOL procedure "o_sohoup".	TPOCC software starts. Procedure configures TPOCC for a pass and completes with no errors.	Procedure "o_sohoup" must include 1) call to EVTRPT to log events to a file in realtime, and 2) call script "log_monitor.pl".
3. Trigger violation to initiate email notification.	*"Attention NS"* Action window shows expected action has been executed, and is waiting for a response. Target receives email with expected message.	
4. Reply to notification email.	*"Attention"* server receives email. The action is removed from the *"Attention NS"* Action window. The *NS* audit log shows that Event ID N was changed to state 0.	State 0 means the event is acknowledged; i.e., no further action required.

Test Case 3.5: ANS Notify by schedule		
Step/Action	Expected Results	Comments
1. Using the "*Attention Calendar Editor",* set up a calendar for each employee's schedule.	A calendar is created for each employee, reflecting the times and dates that user is available for notification.	
2. Set up an action in *"Attention NS"* that notifies the employees entered in Step 1.		
3. Trigger the action set up in Step 2.	The *"Attention NS"* software should only notify the person(s) marked as available on the calendar.	
4. Repeat Step 3 at another time, when the availability has changed.	The *"Attention NS"* software should notify different person(s), according to the calendar.	

Test Case 3.6: ANS Notify by modem		
Step/Action	Expected Results	Comments
1. Configure an action to notify a pager via the modem. The action should wait indefinitely after paging.		
2. Manually trigger the action set up in Step 1 by using the *"Attention NS"* UI.	The *NS* UI shows the event ID of the action executed. The *"Attention"* server dials the modem. The configured user receives the numeric pager notification containing the event ID of the action.	
3. Use the *"Attention NS"* web interface (by logging directly in to *sohoattn3*) to acknowledge the action.	The action is removed from the *NS* Action window. The *NS* audit log shows that Event ID N was changed to state 0.	State 0 means the event is acknowledged; i.e., no further action required.

5.4 Test Case 4.0: Test of Automation Scripts

Description: This test scenario will verify all scripts necessary to the operation of the automation software function properly.

Success Criteria: TBD

Test Procedures:

Test Case 4.1: Script "log_monitor.pl"		
Step/Action	Expected Results	Comments
1. Start TPOCC software.	TBD	
2. Start event report to file "evtrpt.rpt"		
3. Run "log_monitor.pl" on workstation.		
4. Enter command "ls" on TSTOL command line.		
5. Using the *", "Attention Alarm Manager"* interface, view the Activity Log.	Event messages from TPOCC (the lines output from the "ls" command) are visible in the Activity Log.	

Test Case 4.2: Script "fot_frm_mon"		
Step/Action	Expected Results	Comments
1. Start "fot_frm_mon".	Script "fot_frm_mon" starts. *"Attention NS"* GUI shows heartbeats from the fot_frm_mon.	
2. Wait for (or somehow cause) degraded telemetry.	At a pre-defined data quality level, fot_frm_mon sends an event to *"Attention NS"*. The NS GUI shows the action for degraded telemetry is executing.	

5.5 Test Case 5.0: Test of Operational Scenarios

Description: This test scenario will verify that the POLARIS automation software can perform all required pass activities from start to finish for the different types of passes planned to be conducted under automated operations. Ground and spacecraft anomalies will be introduced to test contingency responses to these situations.

Success Criteria: All pass activities must execute as expected to meet mission requirements. In contingency cases, the automation software must either resolve the situation or provide timely notification that manual intervention is required.

Test Procedures:

Test Case 5.1: Nominal non-ranging pass		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.2: Nominal ranging pass		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.3: Non-ranging to non-ranging handover		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.4: Ranging to non-ranging handover		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.5: Non-ranging to ranging handover		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.6: Ranging to ranging handover		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.7: Transition a pass from manned to automated operations		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.8: No telemetry at pass start		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.9: Telemetry dropout during pass		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.10: Unable to lock RCVR-1 for ranging pass		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.11: RCVR-1 loses lock during ranging pass		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Test Case 5.12: RCVR-2 loses lock during non-ranging pass		
Step/Action	Expected Results	Comments
1. TBD.		
2.		

Appendix A: Planned Test Order

The goal is to test all requirements during the first test phase, to minimize operational risks and increase expectations for success. If a requirement must be tested during a later test phase due to environment limitations, testing will proceed cautiously only after all other tests that can be conducted have passed.

Phase 1—POLARIS Component Testing

- Testing of the TPOCC Interface
- Testing of the Pass Generator tool
- Testing of the automated *Attention!* Notification System
- Testing of additional tools/scripts

Phase 2— Testing with the SOHO Simulator

- Local Test to Verify Routine Operational Scenarios and Procedures

Phase 3—Primary Operations Testing

- Testing the automated system end-to-end

Appendix B: Test Verification Matrix

Test Case	Test Phase			Requirements Tested
	1	2	3	
Test Case 1.0 – Verify that the POLARIS TPOCC Interface and the PTI GUI startup, run, and interface properly with TPOCC.				
1.1 Start up PTI and confirm operable.	X	X	X	2.1, 2.2, 2.4.1, 2.4.2, 2.4.3
1.2 Use PTI to run TSTOL procedure and send command load.	X	X	X	2.2, 2.4.1, 2.4.2, 2.4.3, 11.1, 11.2, 11.3
1.3 Start PTI GUI and confirm operable.	X	X	X	2.1, 2.2, 2.4.1, 2.4.2, 2.4.3
Test Case 2.0 – Verify the ability of the pass generator tool to build pass procedures for all types of passes expected to be run in automated mode.				
2.1 Build pass procedures for all types of night non-ranging passes.	X	X	X	4.1.1.3, 4.1.1.4, 4.1.1.5, 4.1.3.2, 4.1.2.5, 4.1.2.6, 4.3, 4.7, 4.9
2.2 Build pass procedures for all types of night ranging passes.	X	X	X	4.1.1.3, 4.1.1.4, 4.1.1.5, 4.1.1.6, 4.3, 4.7, 4.8, 4.9
2.3 Add SVM activities to daytime pass procedures.	X	X	X	3.1, 3.2, 4.1.1.2, 4.1.2.2, 4.1.2.3, 4.1.2.4, 4.1.3.1. 4.3, 4.4.3, 4.7, 4.9
2.4 Build pass procedures for all types of station handovers.	X	X	X	4.3, 4.7, 4.9
2.5 Build pass procedures for other types of passes.	X	X	X	4.2, 4.3, 4.4.1, 4.5, 4.7, 4.9
Test Case 3.0 – Verify the ANS interfaces to the TPOCC, detects various types of anomalies and limit violations, and provides required notifications.				
3.1 Confirm ANS detects red/yellow high/low limit violations and configuration monitor violations, software anomalies, and hung TSTOL procedures.	X	X	X	6.3, 6.3.2.4, 6.5, 7.6
3.2 Trigger individual pager notifications.	X		X	7.4, 9.1, 9.3, 9.7
3.2.1 Send reply to pager notification.	X		X	9.5

Test Case	Test Phase			Requirements Tested
	1	2	3	
3.2.2 Do not reply to pager notification and confirm notification escalation.	X		X	9.6, 9.7
3.3 Trigger group pager notifications.	X		X	7.4, 9.1, 9.3, 9.7
3.3.1 Send reply to pager notification.	X		X	9.5
3.3.2 Do not reply to pager notification and confirm notification escalation.	X		X	9.6, 9.7
3.4 Trigger email notifications.	X		X	7.4, 9.7, 9.8
3.5 Trigger notifications by schedule.	X		X	7.4, 9.1, 9.2, 9.3, 9.7
3.6 Trigger notifications by modem.	X		X	7.4, 9.1, 9.3, 9.7, 9.9
Test Case 4.0 – Verify automation scripts function properly.				
4.1 Run script "log_monitor.pl"	X		X	7.1
4.2 Run script "fot_frm_mon"	X	X	X	6.6
Test Case 5.0 – Verify the POLARIS automation software performs all required pass activities from start to finish for the different types of passes to be run under automated operations, including anomalous situations.				
5.1 Run an automated non-ranging pass.		X	X	5.1, 5.2, 5.3
5.2 Run an automated ranging pass.		X	X	5.1, 5.2, 5.3
5.3 Perform an automated non-ranging to non-ranging handover.		X	X	5.1, 5.2, 5.3
5.4 Perform an automated ranging to non-ranging handover.		X	X	5.1, 5.2, 5.3
5.5 Perform an automated non-ranging to ranging handover.		X	X	5.1, 5.2, 5.3
5.6 Perform an automated ranging to ranging handover.		X	X	5.1, 5.2, 5.3
5.7 Transition a pass from manned operations to automated operations.		X	X	5.4
5.8 Run an automated pass with no telemetry at AOS.		X	X	TBD
5.9 Run an automated pass with a telemetry dropout during the pass.		X	X	6.3.2.1, 6.4.1

Test Case	Test Phase			Requirements Tested
	1	2	3	
5.10 Run an automated pass with a telemetry dropout during the pass, followed by receipt of low rate telemetry.		X		6.3.2.3
5.11 Run an automated ranging pass with RCVR-1 unable to lock on the uplink carrier.		X	X	TBD
5.12 Run an automated ranging pass during which RCVR-1 loses lock on the uplink carrier.		X	X	6.3.2.2, 6.4.2
5.13 Run an automated non-ranging pass during which RCVR-2 loses lock on the uplink carrier.		X	X	6.3.2.2, 6.4.2

Appendix C: Requirements Not Mapped to Test Cases

Requirement	Reason Requirement Not Mapped to Test Case
2.3	TBD
4.1.1.1	Requirement no longer exists due to COBS Reaction Wheel Speed Limit Update patch
4.1.2.1	Activity is not to be automated. TBD
4.1.3.3	Activity is not to be automated. TBD
4.1.3.4	TBD
4.4.2	TBD
4.6	TBD
6.1	TBD
6.2	TBD
6.3.2.5	TBD
6.6	TBD
7.2	TBD
7.3	TBD
7.5	TBD
8.1	TBD
8.2	TBD
8.3	TBD
9.4	TBD
10.1	TBD
10.2	TBD
12.2	TBD
12.3	TBD

Appendix D: Automated TSTOL Procedures Flight-test Dates

Procedure Name	User "fot"	User "polaris"	Called by pass procedure
a_crs_drft_est			
a_hx_lim_upd			
a_nom_memdump			
a_r_str_set			
a_realtohex			
a_rolltm_set			
a_ssu_elig			
a_star_rel			
d_em_rec			
d_ir_ssr_dump			
d_ir_ssr_record			
d_ir_ssr_stp_rc			
d_ssr_dump			
d_ssr_ml_res			
d_ssr_pntr_rst			
d_ssr_record			
d_ssr_stp_dmp			
d_ssr_stp_rc			
d_tr_repoint			
i_exp_ck			
k_cdmu_mem_sp			
k_clr_anom			
k_clr_caflags			

Procedure Name	User "fot"	User "polaris"	Called by pass procedure
k_dutoidle			
k_fsm_clct			
k_gen_dump			
k_hrtomr			
k_idtodump			
k_idtomh			
k_idtomm			
k_idtorc			
k_ky_irtomr			
k_ky_mmtoidle			
k_lrtomr			
k_mhtoidle			
k_mhtomm			
k_mmtoidle			
k_mmtomh			
k_mrtoidle			
k_nom_dump			
k_obt_dist			
k_rctoidle			
k_rectodump			
k_rwmn_dlyadj			
k_set_intrec			
k_setvsubr			
k_ttag_ck			
m_fl_idtomh			
m_fl_idtomm			
m_fl_ky_mmtoid			
m_fl_mhtoid			

Procedure Name	User "fot"	User "polaris"	Called by pass procedure
m_fl_mhtomm			
m_fl_mmtoid			
m_fl_mmtomh			
o_clkcorr			
o_dlydload			
o_eit_bakeout			
o_esr_setlim			
o_irts_sel			
o_link_check			
o_lockmon			
o_mgram			
o_pages			
o_reports			
o_sohodown			
o_sohoup			

Appendix E: Abbreviations and Acronyms

Acronym	Definition
ANS	*Attention!* Notification System
CCB	Configuration Control Board
DR	Discrepancy Report
DSN	Deep Space Network
ESA	European Space Agency
FOT	Flight Operations Team
GSFC	Goddard Space Flight Center
IMOC	ISTP Mission Operations Center
ISTP	International Solar-Terrestrial Physics (program)
MOMS	Mission Operations and Mission Services
NASA	National Aeronautics and Space Administration
POCC	Payload Operations Control Center
SOHO	Solar and Heliospheric Observatory
TPOCC	Transportable POCC