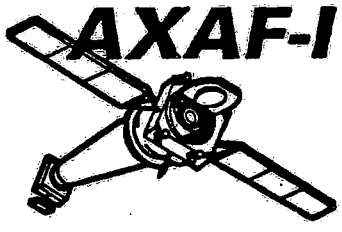
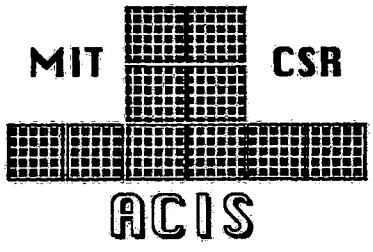
CSR

36-01215 01
Sept. 28, 1994





Advanced X-ray Astrophysics Facility

AXAF - I CCD Imaging Spectrometer

Software Test Plan

Submitted to:

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, AL 35812

Submitted by:

Center for Space Research
Massachusetts Institute of Technology
Cambridge, MA 02139

AXAF-I CCD Imaging Spectrometer
(ACIS)

Software Test Plan

Document No. 36-01215-01

Contract # NASA-37716

September 28, 1994

Submitted to:

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, AL 35812

Submitted By:

Massachusetts Institute of Technology
Center for Space Research
77 Massachusetts Avenue
Cambridge, MA 02139

Approvals:

Peter Ford
Software Project Manager
Massachusetts Institute of Technology

Philip J. Gray
Project Manager
Massachusetts Institute of Technology

CSR

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CENTER FOR SPACE RESEARCH
CAMBRIDGE, MASSACHUSETTS 02139

REVISION LOG	TITLE: Software Test Plan	DOC. NO. 36-01215

Revision	Date (mm/dd/yy)	ECO No.	Page(s) Affected	Reason	Approval
01	9/28/94			PDR Release	

TABLE OF CONTENTS

ACRONYMS & ABBREVIATIONS

ACIS	AXAF CCD Imaging Spectrometer
AXAF	Advanced X-ray Astrophysics Facility
CCB	Configuration Control Board
CCD	Charge Coupled Device
CEI	Contract End Item
CMP	Configuration Management Plan
DPA	Digital Processing Assembly
DR	Document Requirements
ECO	Engineering Change Order
ICD	Interface Controlled Description
MIT	Massachusetts Institute of Technology
MSFC	Marshall Space Flight Center
PE	ACIS Project Engineer
PM	ACIS Project Management
ROM	Read Only Memory
SDP	ACIS Software Development Plan
SE	System Engineer
SIS	Science Instrument Software
SMP	Software Management Plan
SPM	Software Project Management
SPR	Software Problem Report
SQA	Software Quality Assurance
SQAP	ACIS Software Quality Assurance Plan
SRTM	Software Requirements Traceability Matrix
STP	ACIS Software Test Plan
TBD	To Be Determined

1. INTRODUCTION

1.1 Purpose

This plan prescribes the approach taken to verify that the ACIS instrument satisfies the science and interface requirements assigned to software in the Software Requirements Traceability Matrix (SRTM) attached. The approach also identifies how the differences between expected (published) and actual results are detected and reported. This plan identifies the items to be tested, defines the levels of testing performed, test schedules, resource requirements, reporting requirements, and evaluation criteria.

This document is written in response to the tailored Sections of 2.2.3.1 and DM-18 of the Marshall Space Flight Center (MSFC) Software Management and Development Requirements Manual (MM-8075.1).

1.2 Scope

The SIS is not delivered separately, but as part of the Digital Processing Assembly (DPA). As a result, this plan is in effect beginning with the software implementation development phase and ends at the beginning of the DPA integration with the Power & Thermal-Control Structure. This plan covers the testing activities for unit, integration, system (feature) and validation.

1.3 Applicability

The processes described in this plan are used during the implementation, test and verification of the Advanced X-ray Astrophysics Facility (AXAF) Charge Coupled Device (CCD) Imaging Spectrometer (ACIS) instrument. ACIS contains a Digital Processing Assembly (DPA) which contains Science Instrument Software (SIS). The DPA controls instrument operation, extracts valid X-ray events, and processes the data for downlink.

2. REFERENCED DOCUMENTS

The following documents can be used as requirements for the design and manufacture of ACIS software and form a part of this document to the extent specified herein. The issue in effect at the time of contract award shall apply unless otherwise listed below.

2.1 NASA Documents

MM-8075.1	MSFC Software Management and Development Requirements Manual (Jan. 22, 1991)

2.2 ACIS Documents

ACIS-36-01208	ACIS Science Instrument Software Management Plan (SMP)
ACIS-36-01209	ACIS Science Instrument Software Development Plan (SDP)
ACIS-36-01101	ACIS CEI Specification
ACIS-36-01212	ACIS Software Quality Assurance Plan (SQAP)
ACIS-36-02401	ACIS Software Requirements Traceability Matrix (SRTM)
ACIS -36-01206	ACIS Configuration Management Plan (CMP)

3. OVERVIEW

3.1. Software Categorization

The software that is contained in the DPA can be divided into three categories. Each category uses different levels of testing and has different testing objectives. These categories are :

1). Software that is developed entirely at CSR and implements the requirements of the CEI specification.

2). The Nucleus Operating System which is purchased and is an integral part of the SIS.

3) Software that is developed at CSR that is not an integral part of the SIS but directly effects it's RAM image; i.e. RAM image checksum calculation and checking software.

Other software that is used in developing the SIS is either test software or software tools. These items will be tested as directed by the SPM.

3.2 Software Testing Objectives

3.2.1 SIS Application Software Objectives

This software implements the CEI requirements assigned to Software as identified in the SRTM. As a result this category of software receives the most testing emphasis The overall testing objective is to ensure that this software complies with or exceeds the requirements as described in the CEI specification. In addition, testing will ensure that the SIS will comply with or exceed other requirements as identified by the Science branch in the SRTM. TBD.

3.2.2 Purchased Operating System Software

This software provides the functions identified in the Software Development Plan (SDP). Testing will ensure that these functions operate as described in the "Executive Interface Class Specification" TBD before they are used in the SIS. Modification of operating system software will require that the functions, classes and use cases be thoroughly unit and integration tested.

3.2.3 Ancillary Developed Software

This software is generated in the support of the development of the SIS. Certain functions, such as, checksum generation and checking, Read Only Memory (ROM) image reformatting, etc. will require verification to the requirements statement in the applicable design documentation.

3.3 Definition of Testing Levels

3.3.1 Unit testing

All code will be unit tested to ensure that the individual unit (class) performs the required functions and outputs the proper results and data. Proper results are determined by using the design limits of the calling (client) function as specified in the design specification defining the called (server) function. Unit testing is typically white box testing and may require the use of software stubs and symbolic debuggers. This testing helps ensure proper operation of a module because tests are generated with knowledge

of the internal workings of the module. Test cases will be designed to incorporate the guidelines found the SQAP Attachment 2

3.3.2 Integration Testing

A purpose of integration testing is to verify that the client obtains what is desired from the server. This can only be accomplished if the test has some overlap. Integration testing for the SIS is divided into two categories. They are software to software and software to hardware interfacing.

3.3.2.2 Software to Software Interfacing

Integration is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. There are two levels of integration testing. One level is the process of testing software capability; e.g. being able to send a message via a Direct Memory Access (DMA) port, or the ability to acquire a row of Charged Coupled Device (CCD) data. During this level, each module is treated as a black box, while conflicts between functions or classes and between software and appropriate hardware are resolved. A second level of integration testing occurs when sufficient modules have been integrated to demonstrate a scenario e.g. type of science mode or the ability to queue and receive commands. During this phase, composite builds, or baselines, of the software are married to the engineering versions of the hardware. The objective is to take unit tested modules and build a program structure that has been dictated by design. Test cases must provide unexpected parameter values when design documentation does not explicitly specify calling requirements for client functions. Test cases will also be designed to incorporate the guidelines found on the SQAP Attachment - 2

3.3.2.2 Software to Hardware Interfacing

Software to hardware interfacing testing verifies that the software can interpret stimulus from the hardware. Conversely, this type of testing also verifies that the software correctly stimulates the hardware. Test case limits will be specified in appropriate hardware functional description specifications and ICDs. Test cases will also be designed to incorporate the following.

1. For digital values, the software must verify that it can input or output every bit. The software must be able to handle all bits on, all bits off and alternating checkerboard pattern i.e."010101" and"101010" to the resolution specified in the appropriate hardware specification.

2. If the hardware device has more capacity than the functional specification, e.g. use if a 16 bit input register but only 12 bits are used, and if the hardware does not tie the unused lines to some known state, then test cases must be generated to verify that the software handles these unused bits appropriately.

3. The preferred method of designing of each test case is to make the hardware place a value in the register or I/O port being tested and then read the device to obtain the value. When conditions can't be produced to cause the hardware to place the desired value in a register or I/O port, then a debugger could be used to place the value. Conversely, the software should output a value to the hardware and the results recorded. e.g. read the millivolt output from a D/A converter, record the fact that a particular device moved appropriately.

3.3.3 System (feature) Testing

System testing begins when sufficient hardware and software have been integrated to enable the operation and functions of the integrated ACIS Instrument. The purpose of system (functional) testing is to verify a particular operational envelope of the instrument. This is the responsibility of the Science Branch. with assistance from the Engineering Branch. System test cases will follow formal test procedures based on hardware, software, and science requirements as identified in the SRTM and specified in the CEI.

3.3.4 Verification/Validation Testing

Validation testing begins when sufficient hardware and software have been integrated that enables validation of requirements. The purpose of validation is to prove that the ACIS instrument performs at the boundary conditions specified in the Software Requirements Document and the CEI listed above in the Applicable Documents Section of this document. It is not possible to verify all combinations of limits and boundaries. The Verification Test Committee described in the following section determines which cases shall be developed and executed.

3.4 Testing Responsibilities

The Science Branch indicates or approves the test case requirements in the SRTM that are implemented in the Science Instrument Software (SIS). Test data specification and data reduction software for CCD Bias Level Determination, Overclock Level and Event Detection specification algorithms will be supplied by the Science Branch. (TBD) The Science Branch also actively participates in validating Science Requirements during the testing phase of software development.

Unit and integration testing is designed and performed by the software development group from specifications found in applicable design documents 'testing considerations' sections, this document, and the SQAP.

SQA will develop or ensure that all required Feature and Verification test procedures are developed. In addition, SQA will assure that the SIS in the DPA is validated and that internal as well as external deliverables are produced in accordance with the SQAP and this document. In addition, SQA will certify that the validation test results involving the SIS reflect the pass/fail criteria found in the STP. The SRTM is produced and maintained by SQA.

The PM will ensure that there will be sufficient physical and human engineering resources to support the testing effort.

Feature and validation testing development will be performed by the engineering, science and quality branches. Feature and complex verification test case specification will be determined by the Verification Test Committee (VTC). Refer to the next section for definition of VTC. Complex features are ones in which multiple requirements are involved but not explicitly stated. e.g. when more then 50% of randomly selected CCD pixels are bad, verify that at least 1000 candidate events can be detected.

3.4.1 Verification Test Committee

This committee is comprised of representatives from the Science, Engineering and Quality branches of ACIS development. These people will determine which test cases

should be developed to adequately verify that the ACIS instrument meets scientific requirements. SQA will document the test case specification in the SRTM and will develop or ensure development of appropriate test procedures from the specifications.

4. RESOURCE ALLOCATION AND SCHEDULES

The ACIS Software Test Function obtains its resources from the Engineering, Science and Quality Assurance branches of ACIS development at CSR. These branches and their reporting structures are described in the Software Quality Assurance Plan (SQAP) for ACIS.

Unit and integration tests will be developed and run by the software development group. SQA will ensure that these tests are run. According to the SRTM, TBD verification test cases need to be developed. The identification and specification of test requirements will be documented on the SRTM and will be accomplished by the VTC before the software CDR. Test procedure development will begin at software CDR and continue until the SIS software contains all the features specified in the SRTM. Validation testing begins with the Internal release of the SIS and continues until the final version of the software in loaded into ROMs.

ID	Task or milestone	Sep 94	Oct 94	Nov 94	Dec 94	Jan 95	Feb 95	Mar 95	Apr 95	May 95	Jun 95	Jul 95	Aug 95	Sep 95	Oct 95	Nov 95	Dec 95	Jan 96	Feb 96	Mar 96
1	Identify and specify feature and validation test cases	==	==	==	==	==	==	=|												
2	Develop and debug test procedures for test cases							|=	==	==	==	==	==	=|						
3	Unit and integration testing						|==	==	==	==	==	==	==	=|						
4	Run feature and validation tests													|=	==	==	==	==	==	=|
5	Software CDR							*												
6	Internal SW Test Review													*						
7	SW Configuration Inspection																			*

Figure - 1 Test Development Schedule.

The milestones on this chart are copied from the ACIS Software Development Schedule dated 07/22/94.

5. TEST ACCEPTANCE CRITERIA

Test acceptance criteria for simple feature and verification tests are found in the SRTM. Acceptance criteria for complex feature and verification tests will be determined by the Validation Test Committee. TBD

Testing documentation must show that the tests have shown meet or exceed the test criteria. A test case format is shown in Attachment - 2. In all cases the result of the test must result in a simple pass/fail statement and verify the objective of the test. e.g. The pass/fail statement should be "Pass/Fail, Does the system properly handle all 64 different Format Tags ?".

5.1 Testing Methods

The testing methodology used can be any of the following:

a. Inspection: This method of test is used when it is unnecessary to generate data and execute the program, e.g. "the program will handle at least 10 cases". The test consists of inspecting

the code to see that it contains at least 11 branch statements, (10 plus the default).
b. Analysis: This method of test is used when it is not practical to generate the complete data set to test the program. In these cases, a sub-set of the data can be used and then extrapolated or interpolated to verify points of operation that cannot be generated.

c. Demonstration: This method of test is used to verify operational documentation. It also used to verify desecrate operations, e.g. to verify the Format Tag of the Serial Telemetry Packet Format, up to 64 test cases will be generated if the Format Tag values are not contiguous.

d. Measurement: This method of test occurs when data or stimuli are supplied, results measured, recorded, and checked against what was expected.

The SIS uses data compression algorithms. As a result, the analysis form of testing is used to verify the operation of these algorithms. Test data and the tools needed to read this data will be developed by members of the software development team and the science branch. The scope of this effort is TBD and will be known by end of 1994.

6. TEST MANAGEMENT EVALUATION AND REPORTING PROCEDURES

6.1 The ACIS STP Plan

This plan can be modified and controlled as specified in the ACIS Configuration Management Plan (CMP).

6.2 Test Data Results Analysis Procedures

Currently there are three types of algorithms being developed by Software Development with assistance from the Science Branch.. They are Bias Level Determination, Overclock Level Determination and Event Detection. The test data needed to validate these algorithms is TBD. Result evaluation software is also TBD. These efforts will be completed by May 95.

6.3 Requirements Traceability

Traceability is identified through the use of the SRTM which will tie individual Contract End Item (CEI) Specifications, applicable Interface Controlled Description (ICD)s and Software Requirements Document entries to test cases

SQA is included in the review process for all software document generation. During these reviews, checklists and Traceability spreadsheets are used to ensure that requirements are met by both the design and test functions.

6.4 Test Documentation

The testing function for the SIS generates of the following items:

- The SRTM (Attachment -1)
- Test Procedures (Appendix -1)
- Test Reports (Appendix - 2)

The SRTM and the Test Procedures are controlled as specified in the CMP. Unit and integration tests procedures use numeric releases. Verification and feature tests use numeric releases until they are debugged. At that time alphabetic releases are given to ensure that any changes to requirements, hardware or SIS are reflected in appropriate test cases.

6.5 Test Execution

Unit and integration tests are run in an interactive environment by software development. It is anticipated that most of these tests will be automated as they will be run many times due to regression testing. By use of an interactive environment, it is anticipated that as discrepancies are found, they are fixed before testing continues. By use of this method, the developer is assured of fixing all detected bugs with the minimum of documentation.

As feature and validation tests are run, intermediate results are recorded on the Test Report. These tests may be run by personnel from the Science, QA, or Engineering branches. As a result, when a test does not pass, it will be run again to try a verify the failure. In all cases failures will be recorded on the Test Report and on Software Problem Reports (SPRs). The SRTM is updated by use of the Test Reports. SPR status is maintained by SQA and the SPM. The purpose of running these tests is to verify requirements identified on the SRTM. There is an entry in the SRTM for each requirement that indicates if the requirement has been successfully validated.

6.6 Regression Testing

All Feature and validation testing will be performed on brassboard hardware. As determined by the VTC, a selected group of these tests will be run on the flight hardware. All requirements involving performance pass/fail criteria will be run on flight hardware.

Appendix - 1 Test Procedure Example

These scripts should contain sufficient information to ensure that the author can obtain reproducible results from widely separated successive executions of the procedure. The test procedure can be in written form or electronically stored. *Italics* indicate the example entries for each of the following sections:

1. Identification: File Name: *Start-mode.tp* SRTM ID code: *SRS 3.2.1.2.4-a-1*

2 Scope : Describe whether this test is a part of a series, covers an entire requirement, or covers a specific portion of a requirement.

a. e.g. *This test is intended to verify that ACIS reacts properly to a start science mode command. under all anticipated conditions.*

3. Requirement being tested:

a. Briefly describe what the test is to accomplish and the testing method e.g. *This test will demonstrate that the SIS operates properly when requested to be started from any science mode state.*

4. Resources

a. Identify the hardware, interface software, test data file and special test tools that are required to run this test case. e.g. *a working DPA, DEA, RCTU, Command Generation Device, Telemetry Capture Device.*

5. Test Cases:

The following sub-sections identify each test case required to satisfy the Requirement being tested statement.

5.1 Test Case 01

a. Objective: Briefly describe what this test case is to accomplish, e.g. *This test case verifies that the science mode can be started from a just booted state. and an already running state.*

b. Time to test Estimate: *1 hour to setup; 10 mins to perform.*

c. Setup:
(1) Run TBD to obtain the version of SIS that is loaded.
(2) Record the versions of hardware, software and test equipment being used for this test on the test report.

d. Test Execution:
(1) Configure the ACIS instrument for a fairly long TBD science run
(2) Send the "Start <science mode> Run".
(3) RECORD: Was the ?? message sent ??
(4) Send the "Start <science mode> Run".
(5) RECORD: Was the warning message ??? sent ?
(6) Send the "Stop <science mode> Run".
(7) RECORD: Was the ??? message sent ??

e. PASS/FAIL Were all expected messages sent ?
(1) RECORD: yes or no

5.2 Test Case 02

a. Objective: *This test case verifies that the science mode can be started from a*

stopped state..

b. Time to test Estimate: *0 setup; 10 mins to perform*

c: Setup:
(1) This case must follow 5.1 above
(2) . . . .
.
.
.
3. PASS/FAIL ....
(1)RECORD:

Appendix - 2 Test Report Example

Record of running test *Science Mode.tp* SRTM ID code: *SRS 3.2.1.2.4-a-1*

Date of run: *mm/dd/yy* Place of test: .....

Resource Identification:

Target Hardware Name:_____ Version: _____

Test Equipment: Name:_____Version:_____

Test Equipment: Name:_____Version:_____

Telemetry Handler Name:_____Version:_____

Command Handler Name:_____Version:_____

SIS Software Version:_____

Data file name:_____ Version:_____

Configuration name:_____Version:_____

	Test Case Number	Execute Step ID	Recorded Value	Remarks	Pass?	Date

Appendix - 3 To Be Determined Summary

Item	Description	Section	Resoulution
1	Identificaion of requirements in SRTM by Science Branch	3.2.1	End of 94
2	Development of "Executive Interface Class Specification" by Software Development	3.2.2	End of 94
3	Bias, over-clocking, and data reduction algorythm verification data specification	3.4	End of 94
4	Identification of validation tests cases from SRTM by VTC	4.0	End of 94
5	Determiniation of Pass/Fail criteria for complex feature requirements by VTC	5.0	Internal SW Test Review
6	Algorythm test data and data reduction software specification	6.2	May 95
7	Mechanizm to report unique version of SIS software that is being executed.	Appendix - 1 5.1 c(1)	CDR
8	Determiniation of configuration to run for various test cases	Appendix - 1 5.1 d(1)	May 95