SED SPECView

Project Test Plan

Project ID:	1.3
Revision:	1.0
Project Lead:	Janet Evans
QA&T Lead:	Brian Thomas
Document Status:	Ready for Review
Software Version:	1.0 - Year 1 Release
Reviewed by:	*{LEAVE BLANK (to be filled in later)}*
Review Date:	*{LEAVE BLANK (to be filled in later)}*

1 Goal

This document describes the details of how the Spectral Energy Distribution (SED) SPECView project will be tested through its life cycle for requirements to be implemented in Year 1 of the project.

2 References

- SPECView derived requirements
- SED PDD
- Main SED testing page. Contains links to other SED project test plans.
- **SAMP interface document link here**
- **SED allowed serializations document and formatting (e.g. XLS, CSV, Tomcat ASCII, etc);**

3 Project Description

The SED SPECView project is the windowing/GUI component of the IRIS artifact of the SED project. It is an application which is oriented towards scientific end-users. It serves as the central software for user interaction with SED data. Broadly speaking it combines reading, discovery, writing, plotting, and analysis functionality of SEDs in one software application.

The application is implemented in Java and run using JDK Oracle/Sun? JDK 1.6u25. It utilizes the SED Library to enable SED read/write functionality, and implements a SAMP interface to pass SED data to Sherpa for analysis and fitting functionality, and interfaces with SED NED service to enable SED data discovery. Iris/SPECView supports the user plotting and GUI interfaces which facilitate end-user interaction with SED data.

4 Test Environment, Tools and Execution Considerations

This is a component of the Iris application, and shall be tested on Fedora Core 14 & OS X 10.6 using the Oracle/Sun? 1.6u25 JRE.

The scope of testing in this test plan for this application shall encompass static analysis, code coverage, unit testing of its core (SPECView) software infrastucture and component integration testing of SPECView/SAMP, SPECView/SEDLib interaction. User acceptance testing of the application and integration testing of the entire SED software suite (including interaction with SED Service) will be addressed in a separate Test plan (Iris test plan).

All unit and component integration tests should be automated using VAO supported software. For unit tests, this means JUnit unit test framework and build systems Ant 1.8 & Ivy. Static analysis using FindBugs? and PMD software is to be utilized. Code coverage is to be achieved using Cobertura.

5 Additional Information

To faciliate reporting of automated tests, test cases are to be implemented in files which are clearly labelled as that test case, ex. "TestCase3.java" and aggregate all of the unit tests for that test case within them. Alternatively, in the test cases listed below, the individual files which execute the (automated) tests for each test case may be indicated.

6 Test Cases

6.1 Static Analysis Tests

TestCase0 : Static Analysis of Code

Procedure: Automated execution of Findbugs and PMD software during build.

The build process shall incorporate Static Code Analysis software in order to catch defects. PMD and FindBugs? software shall be incorporated, and no defects of the worst level, Fatal/5, shall be allowed.

6.2 Unit tests

TestCase1 : VAO Platform Build Compliance

Requirement Identifier: **???**
Procedure: Automated unit test using Jenkins CI/Build environment

The platform shall be test built, packaged and all unit tests run on each of the VAO supported/configured platforms (**Windows 7 and Fedora Core 14? as specified by requirement ???**).

TestCase2 : Tool can load compliant formatted data

Requirement Identifier: SED.bui.1
Procedure: Automated test using JUnit

The user (tool) can read local spectroscopic and photometric tabular data in VO-compliant (VOtable or FITS described by the IVOA Spectral Data Model) representing IVOA serializations of photometry or spectral data. As part of this test, the test data should also be verified by automated test to ensure they conform with specified formatting.

TestCase3 : Tool can write compliant formatted data

Requirement Identifier: SED.bui.1.1, SED.bui.1.2
Procedure: Automated test using JUnit

The user (tool) can write a table containing photometric measurements (columns with spectral coordinates and flux measurement) and specroscopic data stored in compliant formats to a compliant format with the appropriate header metadata (as designated by document **???**). To ensure compliance of these data, they

must be run through automated format verification software. Round trip (write, read, write and compare) testing should occur for all compliant formats.

6.3 Component Integration tests

TestCase4 : SED Library integration

Requirement Identifier: Prj. 9
Procedure: Incorporate approriate SED Library release, verify all other tests and no stacktrace related to SED Library

Integration with the SED Library shall be tested. The appropriate SED Library shall be incorporated and utilized in all other testing. Verification will be by manual inspection of build configuration.

7 Schedule

Activity	Task	Dependencies	Responsible party	Start Date	End Date	Comments
Static Testing	implementation	VAO CI/Build system, VAO SVN commit of code	QA&T Lead and SED Service PD Team	Feb 2011	March 2011	Implemented by Beta Test 1
Unit Tests	implementation	Static Testing implemented	SED Service PD team	Feb 2011	March 2011	Implemented by Beta Test 1
Component integration	execution	Passed prior testing	QA&T team	April 2011	Jul 2011	Verify during beta1,2 test and at delivery