Certification Test Plan

SSRF Conformance for OpenSSRF Software v3.1.0

Introduction

This is a certification test plan for the OpenSSRF software implementation of the SSRF v.3.1.0 data exchange specification. This test plan is written to conform with, and to produce results compatible with, the *Standard Spectrum Resource Format (SSRF) Certification Test Requirements* document.

Test Procedures

The following test procedures will be implemented and executed automatically in software.

Procedure A: Validate Document Format

- Validate numeric format patterns [automated test]
- Validate string format patterns [automated test]

Procedure B: Validate Document Semantics

- Validate required Data Items [automated test]
- Validate enumerated types [automated test]
- Validate names [automated test]
- Validate Data Items and Attributes [automated test]

During the course of testing all Data Items will be configured according to the following strategy:

- Decimal and integer fields will be populated with random numbers
- String fields will be populated with randomly selected text excerpts using the UTF-8 character set. Where applicable and possible non-English text and non-Latin characters will be used.
- Enumerated fields will be populated with randomly selected enumerated values

Certification Test Plan

An automated test suite will be developed to execute all required certification tests against all primary SSRF v.3.1.0 Datasets identified in the Appendix of the *Standard Spectrum Resource Format (SSRF) Certification Test Requirements* document.

Certification testing will be implemented according to the following general procedure, which is executed for each dataset:

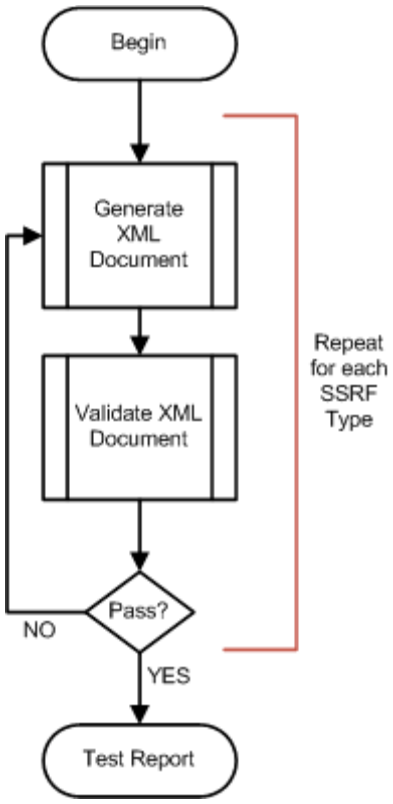
1. Generate a candidate SSRF XML document for the Dataset under test
2. Validate the XML document from Step 1 against the SSRF schema (XSD) document
3. Visually inspect the SSRF XML document (if required)
4. Generate a test report

Automated unit tests will be developed to execute the document format and semantics validation tests..

A single, consolidated report will be generated to summarize the individual test results. Individual test results will be included in the report for reference.

Test 1: Minimum Fill Positive Test

A minimum-fill test will be executed for each primary SSRF Dataset according to the following procedure:



1. Generate Candidate XML Documents

   A program will be developed to use the OpenSSRF software library to generate twenty-five SSRF XML documents. Only those Data Items identified as required in the Specification will be populated in each respective document.

2. Validate XML Documents

   A unit test program will be developed to automatically validate each XML document against the SSRF XSD schema. This will produce a validation report document for each XML document.

3. Visually Inspect XML Documents

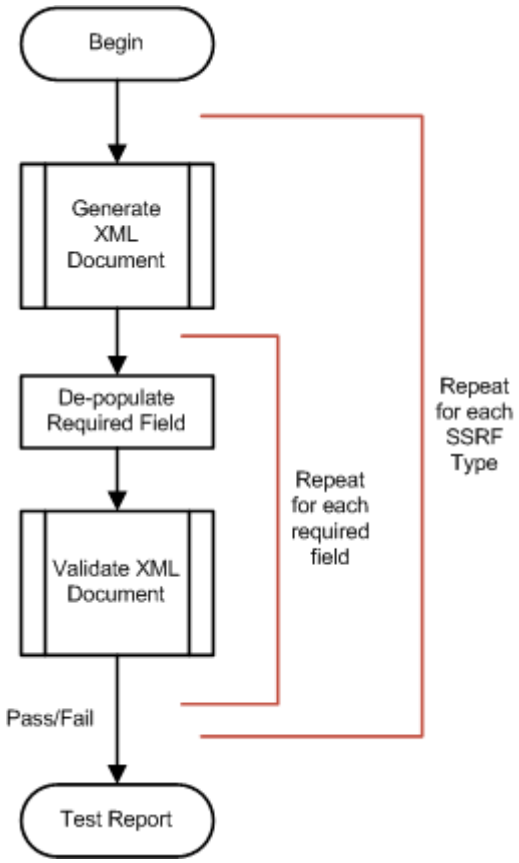
   Visual inspection is permissible, but not required for minimum-fill XML documents.

4. Generate Test Report

   A single test report document will be produced containing a table showing each primary SSRF Dataset and its respective pass-fail status of the minimum-fill positive test.

Test 2: Minimum Fill Negative Test

A negative fill test will be implemented for EACH Data Item identified as required in the Specification. A negative test will be iteratively executed for each required Data Item within each primary SSRF Dataset according to the following procedure:



1. Generate Candidate XML

   A program will be developed to use the OpenSSRF software library to generate twenty-five SSRF XML documents. Only those Data Items identified as required in the Specification will be populated.

   The Data Item under test will evaluated twice: first it will be de-populated (set to null) and then it will be mis-configured (if possible) with an invalid string or number value.

2. Validate XML document

   Each XML document will be automatically validated against the SSRF XSD schema.

3. Visually inspect XML documents

   Visual inspection is permitted, but is not required for invalid XML documents.

4. Test Report

   A single test report document will be produced containing a table showing each primary SSRF Dataset, that Dataset's required Data Items and Attributes, and the respective pass-fail status of the min-fill negative test.

Test 3: Maximum Fill Positive Test

A maximum-fill test will be executed for each primary SSRF Dataset according to the following procedure:

1. Generate Candidate XML

   A program will be developed to use the OpenSSRF software library to generate twenty-five SSRF XML documents. All Data Items in each Dataset will be populated to their maximum allowable size.

2. Validate XML documents

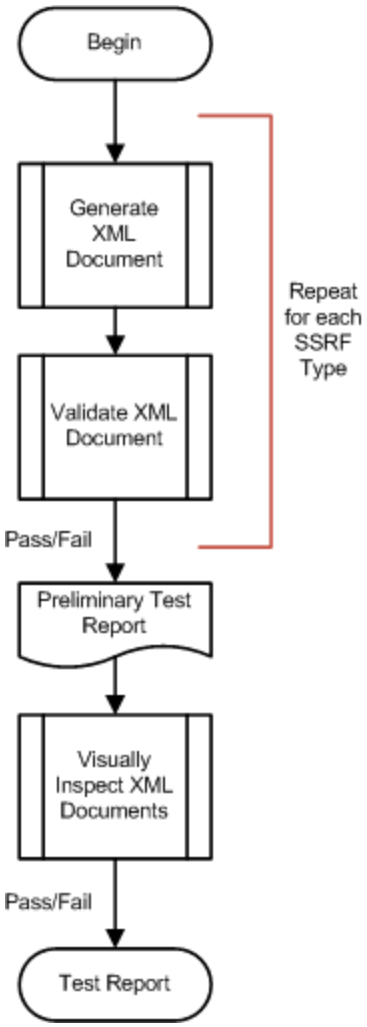
   A program will be developed to automatically validate each XML document against the SSRF XSD schema. This will produce a validation report document for each XML document.

3. Visually inspect XML documents

   Each of the twenty-five XML documents will be visually inspected.

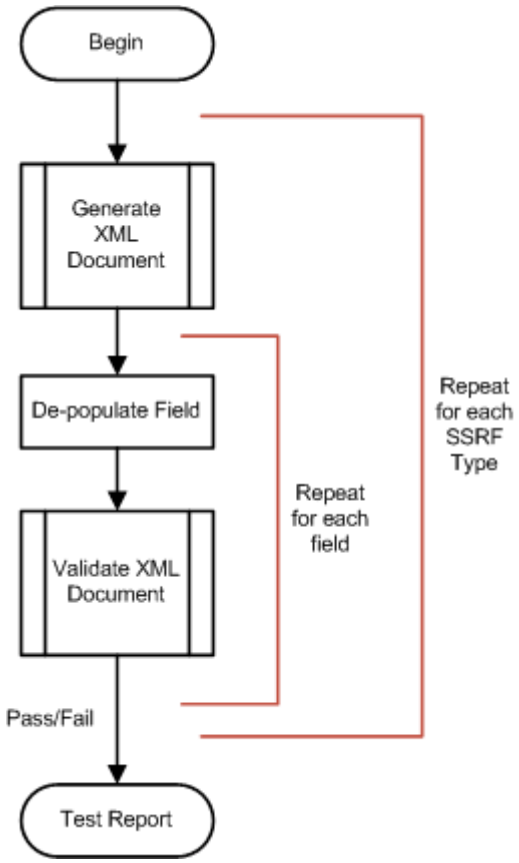
4. Test Report

   A single test report document will be produced containing a table showing each SSRF data type and its respective pass-fail status of the max-fill positive test.



Test 4: Maximum Fill Negative Test

A maximum Fill negative fill test will be implemented for ALL Data Items in each primary SSRF Dataset. A negative test will be iteratively executed for each Data Item within each primary SSRF Dataset according to the following procedure:



1. Generate Candidate XML

   A program will be developed to use the OpenSSRF software library to generate twenty-five SSRF XML documents. All data Data Items will be populated.

   The Data Item under test will be mis-configured (if possible) with a purposefully invalid string or number value.

2. Validate XML document

   Each XML document will be automatically validated against the SSRF XSD schema.

3. Visually inspect XML documents

   Visual inspection is permitted, but is not required for invalid XML documents.

4. Test Report

   A single test report document will be produced containing a table showing each primary SSRF Dataset, that Dataset's required Data Items and Attributes, and the respective pass-fail status of the min-fill negative test.

__END