

# Simplified RFP for the Test Model

This is a “simplified” RFP document, in preparation for the real RFP. I prepared it in order to avoid the additional verbiage required in the official RFP and allow for a more pointed discussion.

The real RFP will be prepared and presented at the Ottawa meeting in June.

1. **Purpose:** To allow description of test cases specifications and test cases results in a standard format facilitating exchanges between static analysis and testing tools.
2. **Relationships** with other models:
  - a. Will use KDM for specification of fields and values. Do we also need ASTM?
  - b. Should link to the UML Testing Profile for possible derivation of test cases from an UML model
  - c. Other relationships?
3. **Evolution:** The test model should be built in such a way as to cover first the unit test and then expand to other types of tests.
4. **Initial model content** (see the previous point on Evolution):
  - a. Unit Test should describe the inputs and expected outputs of a program
  - b. Test Suite should allow the grouping of Unit Tests based on user-defined criteria
  - c. Unit Test Run should describe the results of running a Unit Test at a particular moment in time.
  - d. Suite Run should consolidate the results of Unit Test Runs at the Test Suite level.
5. **Additional object specific content**
  - a. **Unit Test**
    - i. Should have a “name” attribute and a “purpose” attribute and a “expected result” attribute.
    - ii. Should have an “author” attribute?
    - iii. Should have a coverage attribute, indicating the percentage of program statements exercised while running this test.
    - iv. Possible should point to the actual statements exercised in a KDM model?
    - v. Should have an “obsolete” Boolean attribute.
    - vi. Should have a timestamp used to compare with the program timestamp to determine if it is obsolete.
    - vii. Should point to a set of KDM fields. These are the fields describing the inputs or outputs of the program.
    - viii. Should be aware if a field is input, output or input output.
    - ix. If static analysis could reveal it, should be aware that some input or output fields are “dead” – not used in the program.
    - x. For each field, it should be aware of data type (“aware of X” means that the Test Model should facilitate quick and efficient access to “X”)

- xi. If the field is described by an enumeration, it should be aware of it.
- xii. For each input field, a value should be specified.
- xiii. Values do not have to conform to the restrictions defined in the program. For example, an integer field could have a string value. This is done for the purpose of testing invalid inputs.
- xiv. An “ANY VALID” value should be allowed, meaning that when the test is run, the tester is allowed to choose any valid value for this field.
- xv. An “ANY INVALID” value should be allowed, meaning that when the test is run, the tester is allowed to choose any invalid value for this field.
- xvi. An “IGNORE” value should be allowed for the dead fields (not used in the program)
- xvii. Output fields may have values to indicate expected results.

**b. Test suite**

- i. Should have “name”, “purpose” and “expected result” attributes.
- ii. Should have a “requirement id” attribute for possible link to a requirement system?
- iii. Could include any number of unit tests.

**c. Unit Test Run**

- i. Should have a timestamp attribute to indicate when the test was started.
- ii. Should have an “elapsed time” attribute to describe the duration of running the Unit Test.
- iii. Should have attributes to describe the circumstances of running a Unit Tests, including (but not limited) to the date and time and special environment conditions.
- iv. Should have a “Result” attribute, with values “Successful”, “Fail”, “Not run” and “Inconclusive”
- v. The Unit Test run may be related to one or zero unit tests. May be related to zero in case the Unit Test is removed, but the results are preserved.
- vi. Should have an “Observations” attribute, to capture any additional information related to this run.

**d. Test Suite Run**

- i. Should contain entries related to the unit tests runs related to a test suite.
- ii. Should have “time started” and “time ended” attributes.
- iii. Should have an “Observations” attribute to consolidate the observations from the included unit test runs.
- iv. Should have attributes to consolidate the “results” fields of the included unit test runs.