

# WSRP Profile

## Table of contents

1 Introduction.....	2
1.1 Types of Test Assertions.....	2
1.2 Sample Test Assertion.....	3
1.3 Elements and Attributes of a Test Assertion.....	3

## 1. Introduction

The Profile is an XML document that contains a description of each Test Assertion in the test kit. The Profile is found in the `profile` directory and is called `WSRPProfileTestAssertions.xml`. If you open the file in a browser window, an XSL stylesheet is loaded which converts the XML to HTML for more convenient viewing.

The Profile was created initially from the conformance statements found in the WSRP Specification. The conformance statements are documented in a [spreadsheet](#) on the OASIS WSRP site. The conformance statements were boiled down into Test Assertions and documented in the Profile.

The Profile is read at runtime by the Analyzer to determine which Test Assertions should be executed. Test Assertions which are "enabled" are executed, the rest are skipped.

### 1.1. Types of Test Assertions

There are 3 types of Test Assertions. The primary focus the V1 Conformance Test Kit is to verify the message content of a running WSRP service. As a result, the vast majority of Test Assertions are of the Message type.

- Discovery -- Verifies UDDI entries
  - WSRP Profile has no UDDI assertions
- Description -- Verifies WSDL entries
  - WSRP Profile has some WSDL assertions
- Message -- Verifies message content in the log file
  - WSRP Profile has many message assertions
  - Primary focus of test kit.

If the configuration options for the Analyzer requires WSDL processing, then the Analyzer reads the WSDL and execute the Test Assertions from the Profile that are in the description artifact.

After the WSDL is evaluated, the Analyzer processes the message log file one message at a time. In general, for each message in the log file, the Analyzer executes each enabled Test Assertion in the Profile, although the `entryType` attribute results in some high level filtering. Test Assertions which have an `entryType` of `requestMessage` are executed only for requests, those having an `entryType` of `responseMessage` are executed only for responses, and those with an `entryType` of `anyMessage`, are executed for all messages.

A Test Asssertion evaluates each artifact passed to it, and generates a result. The result can

be Pass, Fail, or Not Applicable.

The Analyzer finds the proper Java class to execute for a Test Assertion based on a naming convention where the Java class has the same name as the assertion id in the Profile.

## 1.2. Sample Test Assertion

```
<testAssertion id="RP0070" entryType="responseMessage" type="required"
enabled="false">
  <context>getServiceDescriptionResponse</context>
  <assertionDescription>
    If the Consumer uses a Producer who has set requiresInitCookie
    to a value other than "none", it shall:
    1. Invoke initCookie for each portlet from such a
    Producer for each new end-user;
    2. Return the set cookie(s) only for this end-user
    (see AS006, AS023 and AS024).
  </assertionDescription>
  <failureMessage>
    When a Producer sets ServiceDescription.requiresInitCookie
    to a value other than "none", the Consumer
    has to invoke initCookie()
  </failureMessage>
  <failureDetailDescription>{SOAP message}{any XML parser error
messages}</failureDetailDescription>
  <additionalEntryTypeList>
    <messageInput>none</messageInput>
    <wsdlInput>none</wsdlInput>
    <uddiInput>none</uddiInput>
  </additionalEntryTypeList>
  <prereqList/>
  <referenceList>
    <reference profileID="WSRP1"/>
  </referenceList>
  <comments/>
</testAssertion>
```

## 1.3. Elements and Attributes of a Test Assertion

Many of the elements and attributes in the Test Assertion are not used, but the ones that are important are:

Name	Description
id attribute	This is the unique name of the Test Case. Naming conventions are discussed in more

	detail below.
Type attribute	This is determines to which type of message log td the Test Case applies. For example: responseMessage, requestMessage, binding.
type attribute	"required" or "recommended". Used to distinguish MUST from SHOULD in Conformance Statements.
enabled attribute	"true" or "false". If "true", execute the test case on the data. If "false", just skip it. This is used when the Test Case is added to the Profile but the code hasn't been written yet. The Test Case will show up on the report as not enabled.
assertionDescription element	This is the same as the Conformance Statement text.
failureMessage element	This text is output in the report when a Test Case fails.
prereqList element	This element is used to list other Test Cases (by id) that must be run before this one. If a prerequisite Test Case fails, then so does this one.