Team6 Jython Test Plan

Introduction:

This plan aims to develop and run a series of 25 unit test cases in total on the current, in-development, version of the Jython code base. In developing these tests, we will make use of the functionality of the Jython Bitbucket for ease of browsing the code as well as the JythonDeveloperGuide for clarifications.

The existing Jython code base consists of numerous classes, well beyond the scope of what 25 test cases would cover, so we opt instead to focus on smaller groupings and specific functionalities.

TestItems:

As stated in the introduction, the scope of our tests will be limited to a handful of specific libraries and the functionalities that they represent. At the time of writing, the aim of our test cases revolves around testing the accuracy and robustness of various String and text based functionalities. Beginning with the operations in the Jython String.py class, and subsequently moving on to the URL parsing functionalities of urlparse.py.

Risks:

Given the scope of the Jython project, and the amount of documentation that exists for it and its functions the number of risks posed to our testing should be minimal, although a few potential considerations are as follows:

1. Complex functions – in the case of a few of the libraries in the Jython project, the structure in very complex in terms of the ways in which the various classes will be interacting with each other, we will attempt to minimize this risk by limiting tests to libraries and functions whose use is well documented
2. Updates to software being tested – Since we are creating test cases based on an in-development version of Jython, it is possible, though unlikely, that the addition of some new functions may interfere with the existing functions that we will be testing.
3. Python – As Jython makes numerous calls to the Python system and classes, it is possible, though unlikely, that some python calls made in functions we test may be changed or obfuscated.

Strategy:

The individual test cases will be written uploaded to the Team6 GitHub repository, and automated by a Python script which will execute all of them in sequence. The outputs from these test cases will be checked against expected outputs, which will also be stored in the repository within the test case txt files.

A test will be considered as passed if it runs to completion and results in an output identical to that which is expected.

Environment:

These tests will be executed in a Virtual Ubuntu environment, using Java 1.7 and the associated JDK. As such, while these test should reflect the functionality of Jython under any supported environment that cannot necessarily be guaranteed.

Test Cases:

```
1
this will test the upper method, converting all lower case letters to upper case
jython/lib-python/2.7/string.py
upper()
caT
CAT

2
this will test the strip method, taking a string and removing any trailing or leading whitespace
jython/lib-python/2.7/string.py
strip()
this is a test for trim
this is a test for trim

3
this will test the lower method
jython/lib-python/2.7/string.py
lower()
CAT
cat

4
```

this will test the swapcase method changing lower case letters to upper case and upper case letters to lower case
jython/lib-python/2.7/string.py
swapcase()
teSTING fOr Cases
TEsting FoR cASES

5

this method will test the capwords method, which takes in a string and returns that string with all of the words capitalized
jython/lib-python/2.7/string.py
capwords()
this iS a tEst.
This Is A Test.