

*Moving Jython Forward*

Proposal Submitted to Python Software Foundation

Fall 2004

Brian Zimmer  
bzimmer@ziclix.com

**Abstract**

Jython is a popular implementation of the Python programming language targeted for the Java Virtual Machine. It has a very active user community evident from the number of books published, technical articles submitted and constant references to it in articles and blogs. However, it continues to suffer a stagnant growth that can be attributed to a limited and often quiet development team. While other languages such as Groovy are making headlines with their increased release cycles and features Jython has not kept pace. The current release, 2.2b, is unacceptable given the growth of the C implementation. This proposal is interested developing many of the missing features from the C implementation of Python and documenting the current architecture to enable growing the current development team and moving Jython forward.

## Proposal

This proposal will put an end to the constant Jython-Users question, "Is Jython development dead?" It will address many areas of Jython development but the schedule may require refactoring to better suit the needs and requirements of the Jython user community.

### **New-Style Classes**

**February 2005 (40 hours)**

One of the primary missing features in the current version of Jython is the implementation of new style classes. Samuele Pedroni has done a significant amount of work to implement this crucial feature but it is incomplete. This proposal aims to complete the development through collaboration with Samuele and provide this critical new addition to Jython.

### **Close bugs & apply patches**

**March 2005 (20 hours)**

This proposal will close out critical bugs reported on the sourceforge tracker as well as apply community provided patches.

### **Jython 2.[3-4] Alpha**

**April 2005 (10 hours)**

The completion of the new-style class integration and the fixing of critical bugs will culminate in the release of the first alpha for the improved code base. This will also include the creation of repeatable build, deployment and installation scripts to facilitate future releases.

### **Improved Java Integration**

**May 2005 (80 hours)**

Perhaps Jython's strongest feature is its tight integration with Java and the JVM. While the Python language garners a tremendous following numerous Jython users come from Java, turning to Jython to ease some of the complexities of Java and increase developer productivity. This proposal will address the following issues:

- Support for boolean type
- JythonC integration into standard interpreter compiler
- PEP 302 (New Import Hooks)
- Logging (PEP 282, there exists a patch proposal for integration with Log4J)
- Flexible DataType marshalling from the Jython runtime to Java runtime
  - sets module ⇔ java.util.Set
  - datetime module ⇔ java.util.[Date|Calendar]
  - User-defined types

A common issue going forward will be the continued need to enable the Jython source to run on numerous versions of the JDK. Jython cannot run on the 1.5 version of the JVM because of changes to both bytecode and the core language (assert). This limitation will be fixed.

**Jython 2.[3-4] Beta****June 2005 (10 hours)**

Fix any outstanding issues found in the Alpha and incorporate the tighter Java integration features. The resulting work will be released as a Beta candidate.

**Jython 2.[3-4] Final****July 2005 (10 hours)**

Release a Jython Final.

**Missing Modules****August 2005 (40 hours)**

The C implementation of Python has introduced a number of useful modules and their absence complicates running the same Python source on the C and JVM implementations. It is proposed the following modules be written:

- sets
- select
- datetime

Community prioritization will dictate the remainder, if any, of the modules to be implemented. The submission of completed modules will be greatly encouraged.

**Missing Built-ins****October 2005 (80 hours)**

A number of built-ins are missing as well methods core classes have grown since 2.1. Some of the missing features have been documented in the *whatsnew* lists available with each new release.

- <http://www.python.org/doc/2.2.1/whatsnew/whatsnew22.html>
- <http://python.org/2.3/highlights.html>
- <http://python.org/2.4/highlights.html>

The omissions capable of being implemented in 100% Java will be completed and the unit tests provided for the C implementation confirmed to pass.

**Jython 2.[3-4]+ Beta****October 2005 (10 hours)****Jython 2.[3-4]+ Final****November 2005 (10 hours)****Documentation****Ongoing**

While the Python programming language is well documented the internal workings of the Jython code base remains relatively poorly documented. This proposal includes creating a developer document that includes an overview of the Jython runtime and how it works as well providing a style guide and coding standard. The current code does not have a consistent style, which has created issues for a distributed development team working with different IDEs and platforms. These inconsistencies will be addressed.

### Recruitment of New Developers

In addition, and perhaps most importantly, is facilitating the growth of the development team. This can be accomplished by actively pursuing through the mailing lists and bug reports individuals who have already expressed interest in developing Jython. The successful addition to the Jython code base of a significant patch or new feature will enable the submitter to be added to the development team and have CVS commit privileges if so desired. This should ensure the growth of the Jython development team and perhaps revitalize its activity.

### Budget

The funding will be compensation for time spent as well as the resources to purchase an additional machine to allow for more productive development. The breakdown of the hours required and rate proposed for each task is available in the following table:

<b>Milestone</b>	<b>Delivery Date</b>	<b>Hours</b>
New-Style Classes	Feb-05	40
Close bugs & apply patches	Mar-05	20
<b>Jython 2.[3-4] Alpha</b>	Apr-05	10
Improved Java Integration	May-05	80
Jython 2.[3-4] Beta	Jun-05	10
<b>Jython 2.[3-4] Final</b>	Jul-05	10
Missing Modules	Aug-05	40
Missing Built-ins	Oct-05	80
Jython 2.[3-4]+ Beta	Oct-05	10
<b>Jython 2.[3-4]+ Final</b>	Nov-05	10

The payments for this grant will be made on the successful completion of each delivery of three releases: April 2005, July 2005 and November 2005.

### Summary

Jython has long held the torch for being the most complete and feature-rich language other than Java to run on the JVM. It would be a loss for so much work to have gone into the Jython project only to let it atrophy.

### Qualifications

I am currently a Jython developer with commit status at sourceforge as well as the author of the popular DB API implementation, zxJDBC. I have been the technical reviewer on two Jython texts, *Jython Essentials* and *Jython for Java Programmer*, as well as have answered numerous questions both privately and publicly on developing with Jython. I have been using the Python programming language since 1996 on various platforms, such as NeXTSTEP, OS X, Linux and Windows. I am also a fulltime Java developer and have a long history with releasing quality Open Source software that others and I have used in development and production environments. I believe I have excellent qualifications to continue the development of Jython and further promote the use of the Python language.