

kitty, lightweight, Production-class Java application server performance diagnostic & administration utility

Abstract

A lightweight, production focused, Java-based application server performance diagnostic and management utility

Proposal

- Provide a lightweight utility for managing Tomcat and Geronimo application servers with powerful performance diagnostics and troubleshooting abilities primarily for supporting Tomcat in production/high volume use.
- (future) Provide support for all Java application servers

Background

The answer is simple, there is not a lightweight, command line administration utility that can be utilized across open source application servers. There are many utilities which have been created such as jmxsh, but they do not solve the problem of having a lightweight administration / debugging client for troubleshooting these open source application servers such as Apache Tomcat and Geronimo.

Rationale

There needs to be a lightweight, administration client that targets production use based on the experience of those administering Tomcat (and other open source Java application servers) in high-volume, large scale production environments. Such an administration tool will help further these open source application servers in production, large-installation grade implementations and better support such “industrial-grade” use.

Initial Goals

kitty is an existing open source project, with two contributors. We would bring in more folks with experience in managing high-volume production Web sites to contribute to the

architecture of the kitty project. Currently we have two committers both with high-volume, production Web experience. We'd also leverage feedback from the community in this context and integrate that into the utility to provide a truly powerful management and performance diagnostic utility for Tomcat/Geronimo and other Java application servers.

We will add common diagnostic hooks into the application as a first step, for example, show available memory, threading problems, JDBC, and Web application diagnostic hooks.

The application will run in script mode (future) for automation purposes, or interactive mode, so it can be used for ad-hoc troubleshooting.

Supported Platforms

- Apache Tomcat 6.0+

Future Support

- Apache Geronimo
- All other Java application servers

Known Risks

Currently the application is coded in Jython. Jython makes a suitable fit for many command-line administration tools. We plan on creating a pure, Groovy-based port of kitty in the next few weeks, primarily for ease of compilation to Java classes.

We understand that developing this in Jython makes it faster to develop the utility, but increases it's complexity for compilation. We are in the process of converting the project to Groovy to address this issue, within two weeks from the date of this proposal.

Initial Source

<http://github.com/msacks/kitty>

External Dependencies

Jython 2.5.1

Documentation

- README (Documentation) <http://github.com/msacks/kitty/blob/master/README>
- How Kitty was Born <http://www.tomcatexpert.com/blog/2010/05/17/creating-custom-tools-monitoring-tomcat>

Initial Committers

Matthew Sacks (matthew@glasscodeinc.com)
Peary Chiu (pearychiu@gmail.com)
Jim Jagielski (jim@jimgjag.com)
Stuart Williams (pid@pidster.com)

Required Resources

- * Subversion
- * Jira
- * Wiki
- * Website Space
- * Hudson

Mailing Lists

kitty-dev
kitty-commits
kitty-user

Subversion Repository

<https://svn.apache.org/repos/asf/incubator/kitty>

Issue Tracking

Jira; project known as 'kitty'

Affiliations

Matthew Sacks (self)
Peary Chiu (self, also employed by Edmunds Inc)
Jim Jagielski (ASF/ VMware)

Stuart Williams (VMware/ASF)

Champion

Jim Jagielski

Sponsors: Nominated Mentors

Jim Jagielski
Stuart Williams
Mark Thomas

Sponsor

Apache Incubator