RoadMap

Introduction

This page contains current thoughts of the developer community on the development and release plan for Velocity. Specific issues are listed in JIRA and should be tagged with the versions listed below.

Note that Velocity (as a mature open source project) successfully serves the needs of a wide variety of users across the Java community. In part, its strength comes from a simple syntax and a straightforward design that make it easy to use in most situations. In addition, the project strives to ensure backwards compatibility across all 1.x versions. Consequently, the Velocity developers tend to be conservative and new features tend to be incremental rather than revolutionary.

Having said all that, we welcome feedback and new ideas! Please send your thoughts to the velocity-dev mailing list or post enhancement requests on JIRA to take part in the discussion.

Release 1.6

Objective

Address well-defined enhancement requests and all outstanding bug reports. This update is intended to be a drop-in replacment for Velocity 1.5 and as such strict backwards compatibility is required. Issues accompanied by actual code will receive special consideration.

Specific issues are listed in JIRA

Issues of particular consideration include

- Substantial revision to user documentation with new DocBook format VELOCITY-463
- Allow macros to be defined in files loaded with #parse VELOCITY-281 (WGH)
- New directive #evaluate VELOCITY-509 (WGH)
- Move Texen and Anakia to become separately released subprojects VELOCITY-522 (WGH)
- Auto-magically support most List methods on array objects, so template authors can ignore the difference VELOCITY-533.
- As much as possible, support Iterable in #foreach (VELOCITY-379) and varargs in method calls (VELOCITY-534).

(other suggestions for critical issues to include? leave initials if you are enthused enough to drive the issue)

Release Plan

no specific timetable as of yet

Release 2.0

(under discussion)

Objective

Refactoring of the design and template language to promote consistency and ease of use. Specific focus:

- Cleaning up syntax and API inconsistencies
- Adding new features and syntax that complement existing ones

The basic operation and focus will remain the same. This release will not be a complete code rewrite. Instead it continues the incremental improvements but loosens backwards-compatibility requirements. Most old Velocity templates would still work fine with Velocity 2.0 and others will require a small amount of change.

Possible Enhancements

- Make parser pluggable, allowing for variations in technology and syntax
- Support for tag library ("VSTL"?) to use in JSP
- Improve whitespace processing
- Integrate commons-logging
- Template validator / syntax checker
- Add new directives (e.g. #break for #foreach loops?)
- support JSP Tag libraries
- Support nested custom directives (similar to nested JSP tags)
- Standard List methods that operate on List and array objects (allowing them to be used interchangeability)
- Upgrade runtime requirements to JDK 1.4 and clean up ORO and other pre JDK 1.4 dependencies
- add Groovy or Jython support