VelocityTools2

Table of Contents

1. Overview 2. Configuration
    - XML
    - Properties
    - Java
    3. "Standalone" Use
    4. GenericTools
    5. VelocityView
    6. VelocityStruts
    7. Integration w/a Web Framework
    8. Creating Your Own Tools

Overview

A Velocity "tool" is just a POJO (plain old java object) that is "useful" in a template and is not meant to be rendered in the output. In other words, a "tool" (in Velocity-speak) is meant to be used but not seen themselves (e.g. for formatting dates or numbers, url building, etc).

The VelocityTools project is, first of all, a collection of such useful Java classes, as well as infrastructure to easily, automatically and transparently make these tools available to your Velocity templates. Other aims of the project include providing easy integration of Velocity into the view-layer of your web applications (via the VelocityViewTag and VelocityViewServlet) and integration with Struts 1.x applications.

In recognition of these varying aims, the VelocityTools project is divided into three parts: GenericTools, VelocityView and VelocityStruts. Each of these parts builds on the previous one and has its own JAR distribution.

GenericTools is the set of classes that provide basic infrastructure for using tools in standard Java SE Velocity projects, as well as a set of tools for use in generic Velocity templates. Perenial favorites here are the DateTool, NumberTool and RenderTool, though there are many others available as well.

VelocityView includes all of the GenericTools and adds infrastructure and specialized tools for using Velocity in the view layer of web applications (Java EE projects). This includes the VelocityViewServlet or VelocityLayoutServlet for processing Velocity template requests and the VelocityViewTag for embedding Velocity in JSP. Popular tools here are the LinkTool and the ParameterTool.

VelocityStruts includes both VelocityView and GenericTools and adds tools for use in Struts 1.x applications. These tools match the functions of the key Struts taglibs and provide access to Struts resources, messages, tiles, validation functions and more.

It is worth noting that these tools, being POJOs (though some require a little configuration) are generally useful and may be used within your java classes or even other template languages, though you may need to instantiate and configure them manually (not difficult) to do so. VelocityTools2 has been designed with this in mind. Ask on the user mailing list if you have questions about using VelocityTools without Velocity.

Why 2.0?

Those already familiar with VelocityTools 1.x may be curious about the goals and motivations behind developing VelocityTools2. In planning and developing the 2.0 release, there were three main goals:

- Transparent, on-demand tool instantiation (aka lazy loading for tools) - This allows you to have many tools available (even ones that are expensive to instantiate), but not waste time instantiating or initializing/configuring those you don't use for every request.
- More accessible, reusable infrastructure - This allows easier access to tools outside of your templates and provides better means to integrate VelocityTools support into other web frameworks or even other view technologies (e.g. VelocityViewTag).
- Simpler, more flexible toolbox configuration - Now you can configure via a properties file, plain java, or really whatever you want (though perhaps with a bit more work). No XML required anymore; though, if you do use it, it's a lot better than it was.

These things could not have been accomplished without rewriting most of the core infrastructure and configuration code. At the same time, we wanted to make it easy for people to upgrade from the 1.x series. This led to the decision to aim for complete backwards compatibility. Granted much has been deprecated and those who directly used or extended the 1.x infrastructure will have to update their code in just a few releases, but for the time being they should be able to use 2.0 in their applications with few problems. Those who merely used the old toolbox.xml format or the old tools directly will be immediately able to take advantage of the new infrastructure without even updating their configuration, though eventually even the old toolbox.xml format may be retired. Don't cry. You'll like the new one better once you get to know it! It can do much more with much less typing.

See our planning and progress tracking page for more specific details on the goals and changes that went into VelocityTools2.