

# VelocityTools2 GenericTools

## Overview

[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. These tools have no Java EE dependencies and are often safe to use as "singletons". Some of them are not thread-safe to allow both a no-arg constructor and configurability, but the state-changing methods are declared protected with the exception of `configure(Map)` which is by default rendered useless after being used just once. If you require strict thread-safety, be very cautious in using any configurable tools.

## Default Configuration

The default configuration provided for [GenericTools](#) is [here](#).

## Dependencies

TODO: just link to an anchor in a comprehensive dependencies overview page?

- Compile Time Requirements
  - commons-beanutils
  - commons-collections
  - commons-digester
  - commons-lang
  - commons-logging
  - velocity
- Runtime Requirements
  - commons-beanutils - core infrastructure, [MathTool](#), [SortTool](#)
  - commons-digester - xml configuration
  - commons-lang - [EscapeTool](#)
  - velocity - [ClassTool](#), [IteratorTool](#), [RenderTool](#)

## Tools

- [Alternator](#) & [AlternatorTool](#) - For easily alternating over values in a list or array.
- [ClassTool](#) - For simplifying reflective lookup of information about Classes and their fields, methods and constructors.
- [ConversionTool](#) - For converting String values to richer object types.
- [DateTool](#) & [ComparisonDateTool](#) - For manipulating, formatting, and comparing dates.
- [DisplayTool](#) - For controlling display of references (e.g. truncating values, "pretty printing" lists, and displaying alternates when a reference is null).
- [EscapeTool](#) - For common escaping needs in Velocity templates (e.g. escaping html, xml, javascript etc.)
- [FieldTool](#) - For (easy) access to static fields in a class, such as string constants.
- [IteratorTool](#) - Wraps lists to let the designer specify a condition to terminate the loop, and reuse the same list in different loops.
- [ListTool](#) - For working with arrays and lists, treats both transparently the same.
- [MathTool](#) - For performing math functions.
- [NumberTool](#) - For formatting and converting numbers.
- [RenderTool](#) - To evaluate and render arbitrary strings of VTL, including recursive rendering.
- [ResourceTool](#) - For simplified access to [ResourceBundles](#) for internationalization or other dynamic content needs.
- [SortTool](#) - Used to sort collections (or arrays, iterators, etc) on any arbitrary set of properties exposed by the objects contained within the collection.
- [ValueParser](#) - Used to retrieve and parse (aka convert) String values pulled from a map.