

# Velocity Generator (2.1 legacy document)

## Table of contents

1 Comments.....4

## Table of contents

1 Velocity Generator.....	3
1.1 Sitemap Configuration.....	3

**Warning:**

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

## 1. Velocity Generator

Allows Velocity template to be used as a generator. Builds upon the Velocity servlet functionality - overrides the output method in order to pipe the results into SAX events.

- Name : velocity
- Class: org.apache.cocoon.generation.VelocityGenerator
- Cacheable: ????

`<map:generate type="velocity"/>`

Cocoon Generator that produces dynamic XML SAX events from a Velocity template file.

### 1.1. Sitemap Configuration

Attributes:

**usecache (optional; default: 'false')**

set to 'true' to enable template caching on the 'cocoon' resource loader

**checkInterval (optional; default: '0')**

This is the number of seconds between modification checks when caching is turned on. When this is an integer > 0, this represents the number of seconds between checks to see if the template was modified. If the template has been modified since last check, then it is reloaded and reparsed. Otherwise nothing is done. When <= 0, no modification checks will take place, and assuming that the property cache (above) is true, once a template is loaded and parsed the first time it is used, it will not be checked or reloaded after that until the application or servlet engine is restarted.

Child Elements:

**`<property name="propertyName" value="propertyValue"/>` (optional; 0..n)**

An additional property to pass along to the Velocity template engine during initialization

**`<resource-loader name="loaderName" class="javaClassName"/>` (optional; 0..n; children: property\*)**

The default configuration uses the 'cocoon' resource loader which resolves resources via the Cocoon SourceResolver. Additional resource loaders can be added with this configuration element. Configuration properties for the resource loader can be specified by adding a child property element of the resource-loader element. The prefix '`<name>.resource.loader.`' is automatically added to the property name.

**`<export-object key="objectMapKey" name="velocityContextName"/>` (optional; 0..n)**

Export the object specified by *key* from the Cocoon object map to the Velocity context of the template. The object can be accessed from the template as *name*. Be careful to use a valid VTL variable name.

Default Java objects exported to the Velocity context:

**request (org.apache.cocoon.environment.Request)**

The Cocoon current request

**template (java.lang.String)**

The path of the template file currently being evaluated

**response (org.apache.cocoon.environment.Response)**

The Cocoon response associated with the current request

**context (org.apache.cocoon.environment.Context)**

The Cocoon context associated with the current request

**parameters (org.apache.avalon.framework.parameters.Parameters)**

Any parameters passed to the generator in the pipeline

## 1. Comments

add your comments