

VelocityWhitespaceGobbleNone

Argument and Explanation of the "No Gobbling" Position

The basic argument here is that Velocity is *generic* template language and engine. The language is meant to be easy to learn, and the engine ought to be simple to use. Thus, their behavior ought to be very **simple and consistent**. No unnecessary complications should be added.

Currently, this is not the case. The rules for when/whether/what whitespace is gobbled by the template engine and what whitespace is included in the output are inconsistent and confusing. Most in the community already agree these should be changed, so I will not even try to describe them here for you.

The simplest rule for a template engine to follow would be to not gobble any whitespace. This means that any whitespace in the template that is outside of the template language's directives and references would be included in the output. Another name for this might be **WYTIWYG** ("what you type is what you get"). No surprises. If you put a new line, space, or tab in your template, that character would be output in the place where you put it. This would be a very simple and consistent rule that is an ideal fit for a generic templating technology.

Another argument for this behavior is the view that in the non-functional portions of a template, whitespace should not be special. In the vernacular of the Velocity community, non-VTL (Velocity Template Language) content in a template is called "schmoo". So, the argument here is quite simply that "*all schmoo should be schmoo*" with no discrimination between whitespace characters and other characters. The addition/continuing of whitespace gobbling necessarily violates this. Treating particular characters within the schmoo differently effectively makes them a part of VTL and thereby adds to the language, complicating it unnecessarily.

As we are developing a general purpose templating system, we should not make assumptions about what purpose the various schmoo characters might serve once the system has output them. Eliminating some of those characters during template processing on the assumption that they ~~being non-visible~~ are unnecessary, secondary, or otherwise trivial is stepping beyond our bounds. We can still legitimately provide non-default, configurable means for users of Velocity to handle their schmoo in more application specific ways, but we should not be making such decisions by default on their behalf.

Support for Other Gobbling Rules

On the subject of providing built in support for alternate whitespace/schmoo handling, it is worth noting that the idea of adding such flexibility already has general consensus in this debate. Those in favor of [VelocityWhitespaceGobbleStructuredTemplates](#) have also discussed this possibility amidst their arguments. Likewise, mention of the possible need for supporting backwards compatible whitespace handling also indicates a desire for configurable whitespace handling.

This being the case, it seems sensible to have the *baseline*, underlying behavior of the Velocity engine be to not gobble any whitespace. This would leave it intact for whatever filter/handler is configured to deal with it. If, on the other hand, the base behavior is to gobble particular whitespace, then it seems likely other whitespace handling rules will be difficult to overlay on top of that. Please note though, that even if the *baseline functionality* doesn't gobble, a different handler might still be provided as the *default configuration*. But of course, this is all speculation. Until an actual implementation is developed, we will not really be sure if/how this will/can be made to work. 😊