

# VelocityTools2 StandaloneUse

## Just Use Them!

There's nothing particularly special about the tools in [VelocityTools2](#). No special interfaces, most don't *need* any configuration or API access, and those that do are now relatively easy to handle (compared to [VelocityTools](#) 1.x). So, if you need a tool, just create an instance, do any configuration you want or need and go. Nothing else to it.

## ToolManager

However, if you want to externalize your configuration or have your tools created and configured for you on demand and you are not working in a servlet environment, then we have created a simple [ToolManager](#) that you can use. It's relatively simple. Just create a [ToolManager](#) (can be created with or without default tools available), configure it (if you want to), ask it to create a context for you, and use the context as you would any other. Of course, there's more to it, but this should get you started:

```
ToolManager manager = new ToolManager();
manager.configure("/path/to/my/configuration.xml");
Context context = manager.createContext();
myVelocityEngine.evaluate(context, myOutputWriter, "This is a $text.test", "Test template");
```

Of course, if you are going to be working in a servlet environment and don't want to create and configure tools yourself, then you should use a [VelocityView](#) instance.