

# ViewControllerMapper

## Purpose

To support the ViewController pattern Shale provides a mechanism for automatic 1:1 association of managed beans with JSF views. The [ViewControllerMapper](#) interface provides the mapping between a given JSF view id and managed bean name in your faces-config.xml (maybe you've already seen those warnings "No ViewController found for view id ..."). (For details on the ViewController pattern look into the [Javadoc](#)!)

## Default Implementation

The DefaultViewControllerMapper maps view ids to managed bean names as documented in its [Javadoc](#).

**ATTENTION:** When your view ids have dots in its name (e.g. "tiles.like.file.name.jsp"), the resulting managed bean name is illegal from JSF's point of view. The resulting managed bean name will be "tiles.like.file.name". It can be used without problems for defining the corresponding managed bean in your faces-config.xml. The problem is: it can never be used as a value binding expression. The reason: JSF's default VariableResolver interprets the first '.' in a value binding expression as separator between the scoped object (e.g. a managed bean) and the referenced property or method. So in this small example the VariableResolver would look for a managed bean called "tiles" which is not defined - boom!

## Custom Implementation

To modify the mapping between view ids and managed bean names you can replace the [DefaultViewControllerMapper](#) with an own implementation. As Craig explained [here](#), you have to add the mapper to the application scope using [ServletContextListener](#). In the 'contextInitialized' method put the following code:

```
ServletContext context = event.getServletContext();
context.setAttribute(org.apache.shale.view.Constants.VIEW_MAPPER, new MyViewControllerMapper());
```