

IssuesAndSolutions

This page is meant to be a list of issues that users have encountered while migrating existing Struts Action 1.x application to Webwork, or developing new Webwork-based applications. If you have a solution to the issue that will of course be most helpful, but even if you don't it is worth noting what you encountered none the less in the hopes that someone else can come along and answer it for you, and the rest of us.

How do we set checkboxes false (on uncheck)?

** <http://forums.opensymphony.com/thread.jspa?threadID=23601&tstart=0>

How to set the focus on a form field?

** <http://forums.opensymphony.com/thread.jspa?threadID=23777&tstart=0>

What is the analogy to **ForwardAction**?

<http://forums.opensymphony.com/thread.jspa?messageID=46135됷>

One of the Struts "best practices" is that all requests within an application should go through Struts. So, you should generally never reference a JSP directly for instance. To help with this, Struts provides the ForwardAction, which is more or less an Action that does nothing, it just immediately returns and forwards to the URI as configured. Webwork does not appear to have a direct analogy to this. However, it is easy to get the same effect in a couple of possible ways.

First, you can write a simple Action like so:

```
package com.company.app;

import com.opensymphony.xwork.Action;

public class ForwardAction implements Action {

    public String execute() {
        return SUCCESS;
    }

}
```

Then, configure an Action like this:

```
<action name="justAForward" class="com.company.app.ForwardAction">
    <result name="success">pageToDisplay.jsp</result>
</action>
```

Now you can use that ForwardAction whenever you like, just as you would in Struts. Alternatively, if you don't want to have a whole Action just for that, you can simply have a method in an existing Action (named something other than execute, let's say "gotoJSP" for the sake of argument) and configure an Action like so:

```
<action name="justAForward" class="com.company.app.SomeExistingAction" method="gotoJSP">
    <result name="success">pageToDisplay.jsp</result>
</action>
```

An even easier solution however, as suggested by Ted Husted in the thread referenced below, is to use the ActionSupport class. Your mapping then is just:

```
<action name="justAForward" class="com.opensymphony.xwork.ActionSupport">
    <result name="success">pageToDisplay.jsp</result>
</action>
```

Because the default implementation of execute() in ActionSupport simply returns SUCCESS, and since execute() is called by default when no method is specified in the mapping, this does exactly the same thing as the Action code shown above, without the need for a custom class being introduced.

Here is a reference to the thread discussing this: <http://forums.opensymphony.com/thread.jspa?threadID=23755&tstart=15>