

# Skinning

Skinning in the framework can be done more than one way. We will show how to use two skins called "html" and "wml", and we'll be working with the following directory structure:

```
/WEB-INF
  /web.xml
/html
  /index.jsp
  /Register.jsp
/wml
  /index.jsp
  /Register.jsp
/index.jsp
```

## Classic Approach

Simply place all actions in the default namespace so that they are accessible from any URL path. When you create your views, place them in the sub-directory that corresponds with the skin's identifier.

Your action configuration would look like this (simplified, without defined interceptors):

```
<package name="default">
  <action name="registration" class="x.actionset.Register">
    <result name="success" type="dispatcher">
      <param name="location">Register.jsp</param>
    </result>
    <interceptor-ref name="defaultStack" />
  </action>
</package>
```

If a user requested <http://yoursite/html/register.action>, he would see the JSP located at /html/Register.jsp.

## Namespace Defined

If you require the use of namespaces, you can do the following:

Simplified configuration example:

```
<package name="user" extends="default">
  <action name="register" class="x.x.actionset.Register">
    <result name="success" type="dispatcher">
      <param name="location">Register.jsp</param>
    </result>

    <interceptor-ref name="defaultStack" />
  </action>
</package>

<package name="user-html" extends="user" namespace="/user/html" />
<package name="user-wml" extends="user" namespace="/user/wml" />
```

The last two package definitions extend the first package, changing only the namespace. The view result defined in the "register" action has a relative path. Because of this, you'll get the same behavior as the Classic Approach, but with the security of knowing that ONLY those two paths can be accessed for the action, instead of ANY path.