

Struts 2006

An Embarrassment of Riches



Struts University Series




Abstract

Apache Struts is a hotbed of activity.

Struts Classic 1.3, Struts Shale,
Struts OverDrive, Struts Ti.

Why so many frameworks? How are they different? Why are they all called Struts? Which is the best choice for my next project?

In this session, we step back and look at Struts through a wide-angle lens.






The Story So Far

- Struts 0.5 – May 2000 - **Prototype** release
 - Struts 1.0 – Jun 2001 - **Initial** release
 - Struts 1.1 – Jun 2003 - **Commons** release
 - Struts 1.2 – Dec 2004 - **Wildcard** release
 - Struts 1.3 – xxx 2006 - **Subproject** release
- 




The Upcoming 1.3 Release

- Divided project into subprojects
 - Apps, Core, EL, Extras, Site, Taglibs, Tiles
 - Faces
 - Scripting (new)
 - Flow (new)
 - Shale (new)
 - Subprojects are the "unit of release"
 - Seven subprojects from 1.2 are bundled as "Struts Action Library"
 - Complete Maven build
- 



Struts Action 1.3

- Composable Request Processor
 - Arbitrary configuration properties
 - Command and Catalog elements
 - "Extends" attribute
 - "isCommitted" Exception Handling
 - Postback Actions
 - Wildcard ActionConfig properties
- 



Request Processor for 1.2

```
public void process(HttpServletRequest request,
                    HttpServletResponse response)
    throws IOException, ServletException {

    processLocale(request, response);

    if (!processRoles(request, response, mapping)) {
        return;
    }

    /// ...


    ActionForward forward =
        processActionPerform(request, response,
                               action, form, mapping);

    processForwardConfig(request, response, forward);
}
```




Composable Request Processor for 1.3

```
<chain name="process-action">
  <command className= "...SelectLocale" />
  <command className= "...AuthorizeAction" />
  <!-- ... -->
  <command className= "...CreateAction" />
  <command className= "...ExecuteAction" />
</chain>
```






Arbitrary Configuration Attributes

```
<action path="/EditSubscription"  
        extends="Editor">  
    <set-property key="foo" value="bar"  
/>  
</action>
```

```
public ActionForward execute(  
    ActionMapping mapping,  
    ActionForm form,  
    HttpServletRequest request,  
    HttpServletResponse response)  
    throws Exception {  
    string foo = getProperty("foo");  
    // ...
```





Catalog and Command elements

catalog - Name of commons-chain catalog for command.

command - Name of commons-chain command which should be looked up and executed as part of servicing this request.

Available for both the Controller and any Action Mapping

```
<controller
    inputForward="true"
    catalog="Foo"
    command="FooBar"
/>
```



"extends" attribute

```
<struts-config>
  <form-beans>
    <form-bean name="registrationForm"
      type="org.apache.struts.action.DynaValidatorForm">
      <form-property name="firstName" type="java.lang.String" />
      <form-property name="lastName" type="java.lang.String" />
    </form-bean>


    <form-bean name="managerRegistrationForm"
      type="org.apache.struts.action.DynaValidatorForm"
      extends="registrationForm">
      <form-property name="department" type="java.lang.String"
    />
    </form-bean>
  </form-beans>
  ...
</struts-config>
```



"isCommitted" Exception Handling

```
<exception
  key="GlobalExceptionHandler.default"
  type="java.lang.Exception"
  path="/ErrorPage.jsp">
  <set-property key="INCLUDE_PATH"
value="/error.jsp" />
</exception>
```

```
<exception
  key="GlobalExceptionHandler.default"
  type="java.lang.Exception"
  path="/ErrorPage.jsp">
  <set-property key="SILENT_IF_COMMITTED"
value="true" />
</exception>
```





PostBack Actions

The action attribute of the Form taglib is now optional.
If omitted, the original URI of the original request is used.

```
<html:form  
    onsubmit="return validateLogonForm(this);">
```






Wildcard ActionConfig properties

Wildcards (introduced in 1.2) can now be used in the properties of an ActionConfig.

This makes it easier to pass multiple request-time values to the Action without complicating the "parameter" attribute.


```
<action path="/Save*"
        name="Save{1}"
        extends="BaseSave">
    <set-properties key="Save" value="{1}"/>
</action>
```






New Project Organization

Struts Action Library

- **Apps** - Example applications
 - **Core** - Controller and Validator
 - **EL** - The JSTL EL extended JSP taglibs
 - **Extras** - Optional Actions and Plugins
 - **Site** - The top-level Struts website
 - **Taglibs** - The original Struts JSP taglibs
 - **Tiles** - The Tiles templating framework
- 



New Project Organization

- New Directions, New Subprojects
 - **Shale** - New framework built on JSF
 - **Tiles** - A standalone Tiles release with no Struts dependencies
 - New Directions, New Whiteboards
 - **OverDrive** - Chain-based business framework (C#, for now)
 - **Ti** - Action-based MVC2 framework focusing on simplicity and development speed
- 



New Project Organization

- Building on Struts Action
 - **Scripting** - Scripted Actions using BSF
 - **Flow** - Continuation-based process flow
 - **Faces** - Utilize Struts Actions via JSF






Struts Scripting

Implement Actions with JavaScript, BeanShell, Groovy, Jython, etc.

```
<action path="/logoff"  
    type="org.apache.struts.bsf.ScriptAction"  
    parameter="/WEB-INF/scripts/Logoff.bsh">  
  
    <forward name="success" path="/index.jsp"/>  
</action>
```






Struts Flow

Introduces server-side, Continuations-based JavaScript flows

```
function login() {
  while (struts.sessionScope["curUser"] == null) {


    forwardAndWait("loginForm", {"error" : error});
    user = struts.param["user"];
    passwd = struts.param["passwd"];
    if (userManager.login(user, passwd)) {
      struts.sessionScope["curUser"] = user;
    } else error = "Invalid login, please try again";


  }
}
```






Struts Shale

- Web framework based on JSF
 - Created by Craig McClanahan
 - Provides a "Front Controller" layer
 - Like Struts Action
 - Shale and Action are both first-class citizen of Apache Struts
 - One project, two frameworks
- 




Struts Ti (phase 2)

- Exploring how to simplify development of MVC web applications
 - Prefers convention over configuration
 - Including features like annotations and automatic compilation
 - Builds on WebWork, Beehive, and Spring
 - Still in the design phase although working code and examples are available
- 



Future of Action 1.x

- Apache Struts is a set of products, each with its own label and lifecycle
 - Struts Action 1.x line has been evolutionary not revolutionary
 - Deprecate, Release, Remove
 - Same codebase, same major release
- 

Struts 2006

WebWork Merger




Struts University Series



Abstract

Apache Struts,

the leading web application framework for Java,
and **Open Symphony WebWork,**
a leader in technical innovation,
are working to merge their
communities and codebases.





The Story So Far

- WebWork 1.0 – Mar 2002 – **Initial** release
 - WebWork 1.2 – Aug 2002 – **OS** release
 - WebWork 1.4 – Nov 2003
 - WebWork 2.0 – Feb 2004 – **XWork** release
 - WebWork 2.1 – Sep 2005
 - WebWork 2.2 – Dec 2005 (Beta 4)
- 




WebWork joining Struts

- October 2005
 - Java Web Alignment Group
 - November 2005
 - Mailing list announcements
 - December 2005
 - ApacheCon announcement
- 




Migration

- Well-paved migration paths
 - Tools
 - Examples
 - Dual Processors, Shared Resources
- 



Migration

- Tools for source migration
 - Work in progress
 - Examples for developer migration
 - MailReader
 - IBATIS JPetstore
 - and others
- 




Migration

- Dual processors / Shared resources
 - *.do – Struts Action 1.x
 - *.action – Struts Action 2.x
- What breaks? -- *nothing!*





Brave New World

- POJO forms and actions
 - HTTP independence
 - UI Tags
 - Explicit AJAX support using Dojo
 - Direct support for Action Chaining
 - Continuations
 - Extensible response types
 - Quick Start
- 




Ti RoadMap

Phase 1


- WebWork 2.2 codebase,
- Struts Action 1.x migration tools,
- Commons Chain integration.

Phase 2

- Zero XML configuration
 - Annotated stateful POJO controllers
 - Dialog / workflow support
- 




Glossary

- Actions – Actions
 - ActionForms – Action properties
 - Or POJOs
 - XWork Interceptors – Chain Commands
 - Interceptor Stack – Request Processor
 - Per Action
- 



Final Thoughts

- Apache Struts is both evolving and plunging ahead
 - Struts Action 1.x will be actively developed and supported for a long time
 - Offering both Shale and Action future-proofs Apache Struts
 - Bottom line: use what works for you.
- 



Struts University Series