

Release Plan 2.0.0

- Struts 2.0.0 is preparing a "early adopter" build.
- Codebase seems feature-complete.
- Documentation issues remain.
- Example issues remain.
- Assembly/distribution issues remain.

✔ For resolved issues, see [Struts 2.0.0 DONE](#) in the issue tracker.

🔧 For pending issues (if any) see [Struts 2.0.0 TODO](#) in the issue tracker.

⚠ For known issues, see [Struts 2.0.1 TODO](#)

Overall Status

Initial rename of packages and configuration files	Complete	✔
Pass Web`Work2 codebase through the Incubator	Complete	✔
Setup Nightly Builds	Pending	🔧
Update WW2 documentation for Struts 2	Complete	✔
Utilize Cookbook format for Showcase and add Struts-Examples use cases		🔧
Remove deprecated members (IoC components)	Complete	✔
Deprecate selected legacy API members (e.g. Around`Interceptor, "do" clause)	Complete	✔
Update Javadoc to match new package and configuration naming	Complete	✔
Implement selected new features and Struts 1 features (infra)	Complete	✔
Add Struts 1 and WW2 Migration Guides	Initial draft added	🔧
Develop routine for exporting wiki to HTML/PDF	Complete	✔
Develop Maven assembly to create distribution	Complete	✔
Second documentation review	Complete	✔
Complete standard release plan	Complete	✔
<i>Struts 2.0.0 is intended as a "development" distribution only</i>		

Documentation Status

Step 1 - Setup new Confluence instance	Complete	✔
Step 2 - Initial pass to update nomenclature and remove obsolete material	Complete	✔
Step 3 - Update "snippets" to reference ASF repository	Complete	✔
Step 4 - Complete tutorials and other "TODO" sections	Complete	✔

Additional Background

- [Struts 2.0.0 omnibus ticket](#)
- [Issue Tracker](#)
- [Struts Ti Proposal](#)
- [Migration Issues](#)
- [Struts 2 Rough Spots](#)
- [Public API Whiteboard](#)
- [Retroweaver](#)

Release Manager

The release manager is **Ted Husted**.

- See also [Creating and Signing a Distribution](#)

Special Issues

Struts 2.0 is based on the WebWork 2 codebase. Essentially, Struts 2.0 will be the technical equivalent of WebWork 2.3. (The WebWork 2 codebase was accepted into the Struts project through the [Apache Incubator](#).)

For Struts 2.0, the core goals are to repackage the codebase as an Apache Struts product and to provide migration resources for Struts 1 and WW 2 developers. A few legacy API elements will be removed, and a few new features will be added.

The Struts 2.0 release is following the [Struts Ti Proposal](#). Struts 2.0 is Phase 1 of the proposal. A subsequent Struts release series (either Struts 2.1 or Struts 3.0) will address Phase 2 of the proposal.

Struts 2.0.0 is targeting August 2006 as a relaxed distribution date.

The platform for Struts 2.0.x is Java 1.5, with Java 1.4 compatibility provided by [RetoTranslator](#).

See also [Struts 2](#).

New Features Under Consideration

This initial list of new features is taken from the [Struts 2 Rough Spots](#) page.

Review and Refinement

- Separate classes that most applications need to use. Isolate classes that the framework uses internally or that are rarely used by applications. (See Rough Spot #3.)
- Distinguish between the published public versus private APIs. (See Rough Spot #4.)
- Utilize properties (methods) instead of protected or public fields (). (See #13.)
- Review exception handling. Some are still being swallowed rather than rethrown. (See #20 and [WW-652@ww](#))

Modifications

- Simplify TextProvider. (See #15.)
- Modify ParametersInterceptor to sort parameter names by depth (using bucket sort) and then map them in that order (shallowest first). (See #21.)
- Use the strategy pattern to swap in different implementations of interfaces like ValidationAware. (See Tim, NTH #1.)

Deprecations and Removals

- Deprecate or remove the AroundInterceptor. (See #7.)
- Eliminate "do" clause for Action alias lookups. Scan Actions at load time for known aliases, and fail fast.
- Eliminate "default" alias (which relied on "doAlais").

Architecture

- Address the confusing issue of the validation/workflow lifecycle and different methods. (See Patrick #4.)

Configuration

- Add support for type aliasing (a la Spring and iBATIS) to reduce or eliminate need to use fully-qualified class names, over and over. The fully-qualified type is declared once, and then a simple type "nickname" can be used instead. (See #23.)
- Combine configuration DTDs so that Actions, Messages, and Validators can be defined in a single file. (See Ted, #1.)
- Use values on the stack to override/define configuration options. (See Tim, NTH #3.)

OGNL

- On the OGNL value stack, rename #request for request parameters and #requestAttributes for request attributes. (See #17 and [WW-785@ww](#).)
- Work with OGNL to add Java 5 support. (See #31.)
- Work with OGNL to make it a clear superset of JSTL. (See Gabe #4.)

Struts 1 Features to Implement

- Wildcard Action Mappings (Done in XWork 2)
- ~~DynaForm as Model Object (including [LazyDynaForms](#))~~
- Extends for action mappings (See Tim, NTH #2.)
- Change the default submit method for the form tag from GET to POST (See Ted, #2.)
- [Set field focus](#)

Outstanding Issue Review

[Struts 2.0.0 Issues](#)

Testing Checklist

Testing Summary

#	Description	Completed
1.	Run Unit Test targets	✔ Done
2.	Play test bundled applications	\$(STATUS)

TODO: A Silenium test suite for the applications would be great!

Test Build Distribution Checklist (A)

See also [Creating and Signing Releases](#)

#	Description	Completed
A1.	Setup new JIRA version level or update release on Roadmap	✔ Done
A2.	Tag release in svn: \${STRUTS_2_0_0}	✔ Done (r447072)
A3.	Update POM version level and run Distribution Target	✔ Done
A4.	Create Sums and Sign Distributions	✔ Done
A5.	Upload Distribution to <code>people.apache.org/builds/struts/2.0.0/</code>	✔ Done
A6.	Deploy JAR to Apache Java-Repository	✔ Done
	<code>\$ mvn deploy -P pre-assembly</code>	
A7.	Post release-quality vote on dev@ lists	\$(STATUS)

Vote (A)

\$(PMC_MEMBER)	\$(GRADE)
}	}

- If release vote fails, including for a lack of quorum (3 binding +1s), remove from builds folder.
- If a release vote passes (Alpha, Beta or GA), the release may be announced and mirrored, at the PMC's discretion.
- If vote is GA, goto Checklist (C).

Point Release Distribution Checklist (B)

B1	Update "Downloads" page on website and Test Downloads	\$(STATUS)
----	---	------------

Vote (B)

If initial vote is Alpha or Beta, voting can continue until a GA or "withdraw" vote passes, or there is a subsequent release.

\$(PMC_MEMBER)	\$(GRADE)
}	}

General Availability Distribution Checklist (C)

	Description	Completed
--	-------------	-----------

C1.	Copy Distribution to Mirrored Directories	\$(STATUS)
C2.	After 24 hours, update "Downloads" page on website	\$(STATUS)
C3.	Post an announcement to lists and website	\$(STATUS)