Release Plan 2.0.2

- Struts 2.0.2 is preparing a "release candidate" build.
- Documentation issues remain as to migrating.
- Assembly/distribution issues remain.

✅ For resolved issues, see [Struts 2.0.2 DONE](#) in the issue tracker.

💡 For known and pending issues (if any) see [Struts 2.0.2 TODO](#) in the issue tracker.

Overall Status

[Setup Nightly Builds](#)	**Pending**	💡
[Utilize Cookbook format for Showcase and add Struts-Examples use cases](#)		💡
[Add Struts 1 and WW2 Migration Guides](#)	**Initial draft added**	💡
Complete standard release plan	**Pending**	💡

Additional Background

- [Struts 2.0.2 omnibus ticket](#)
- [Struts Ti Proposal](#)
- [Migration Issues](#)
- [Struts 2 Rough Spots](#)
- [Public API Whiteboard](#)

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 2.2 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.

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

See also [Struts 2](#).

New Features Under Consideration

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

Some new features were added in [Struts 2.0.0](#).

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.)

Configuration

- Add support 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.)
  - An alternative to this issue is adding a default=action-class-ref so that packages can designate a default class other than ActionSupport.
- 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

- Extends for action mappings (See Tim, NTH #2.)
- Set field focus

Testing Checklist

Testing Summary

#	Description	Completed
1.	Run Unit Test targets	
2.	Play test bundled applications	

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

Test Build Distribution Checklist (A)

💡 See also Creating and Signing a Distribution

#	Description	Completed
A1.	Setup new JIRA version level for next distribution	
A2.	Update POM version and tag release in svn: ${STRUTS_2_0_1}	
A3.	Run Distribution Target	
A4.	Create Sums and Sign Distributions	
A5.	Upload Distribution to `people.apache.org/builds/struts/2.0.0/`	
A6.	Deploy JAR to Apache Java-Repository	
A7.	Post release-quality vote on dev@ lists	

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 "Downooads" page on website and Test Downloads	

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.

struts2-all

${PMC_MEMBER}	${GRADE}

struts2-api

${PMC_MEMBER}	${GRADE}

struts2-core

${PMC_MEMBER}	${GRADE}

Plugins

struts2-config-browser-plugin

${PMC_MEMBER}	${GRADE}

struts2-jasperreports-plugin

${PMC_MEMBER}	${GRADE}

struts2-jfreechart-plugin

${PMC_MEMBER}	${GRADE}

struts2-jsf-plugin

${PMC_MEMBER}	${GRADE}

struts2-pell-multipart-plugin

${PMC_MEMBER}	${GRADE}

struts2-plexus-plugin

${PMC_MEMBER}	${GRADE}

struts2-quickstart

${PMC_MEMBER}	${GRADE}

struts2-sitemesh-plugin

${PMC_MEMBER}	${GRADE}

struts2-sitegraph

${PMC_MEMBER}	${GRADE}

struts1-plugin

${PMC_MEMBER}	${GRADE}

struts2-tiles-plugin

${PMC_MEMBER}	${GRADE}

General Availability Distribution Checklist (C)

	Description	Completed
C1.	Copy Distribution to Mirrored Directories	
C2.	After 24 hours, update "Downloads" page on website	
C3.	Post an announcement to lists and website	