

# GlassFish V3 Config and Admin

Lloyd Chambers October 19, 2007



All images ©2004-2007 Lloyd L Chambers. All Rights Reserved.

# Glassfish V3 Config changes

---

- Low-level Config API is proposed for replacement with an “injection” based approach.
- GlassFish V3 modules obtain configuration without knowledge of where it comes from.
- Performance is a key goal.
- Tools to generate XML elements from configuration class.
- Class vs interface, thread-safety issues, etc.

# Impacts — Validator

---

- Validation code must be understood and possibly reworked and/or reimplemented.
- How to validate configuration for 3rd-party modules?
- Annotations might allow auto-validation of individual fields.
- More complex validation requires “hook”
- Validation spanning XML elements is going to need some thinking to fit into new scheme.

# Impacts—`com.sun.appserv:category=config`

---

- Now is the time to eliminate these MBeans:  
`com.sun.appserv:category=config`
- Some of them contain logic and operations, which needs to be understood and moved into AMX.
- An entire layer of MBeans gone = faster, smaller admin.
- GUI and CLI teams should migrate remaining code to use AMX, instead of `category=config` MBeans.
- GUI and CLI teams to document what code cannot be migrated due to lack of AMX APIs.

# Impacts—AMX (1 / 4)

---

- AMX will need some work, but the existing Delegate implementation should make it feasible to “port” them onto the new Config API with minimal effort.
- The AMX Loader will need to use the new Config API directly, just as it (already) does for its implementation against V2 Config API.
- `createAbcConfig()` and `removeAbcConfig()` implementations will need to be updated for new API.  
[ Example: `createHTTPListenerConfig()` ]
- ...

# Impacts—AMX (2/4)

---

- AMX to support configuration provided by user-supplied modules. The existing Container APIs can be used for this purpose eg `getContaineerJ2EETypes()`, `getContaineerSet(j2eeType)`, etc.
- Configuration elements should be required to have consistent fields eg “name” for the name. **Some standards will be needed for consistent support.**
- The `j2eeType` portion of the `ObjectName` will become dynamic, based on XML element type.

# Impacts—AMX (3 / 3)

---

- AMX to support runtime and/or monitoring MBeans provided by user-supplied modules.
- Modules will need some kind of `ObjectName` support in order to maintain a useful search/traversal ability.
- There should be a formal API for a module (or any GlassFish code) to obtain the **server's MBeanServer**, not just the Platform one (the might or might not be the same!).

# Impacts—GUI

---

- AMX should provide 90% compatibility.
- ACTION: Find all code that uses anything but AMX. Migrate it to AMX if feasible, document required facilities if not available in AMX.
- Don't assume in-process execution.
- ACTION: document any implicit assumptions about bulk set, pseudo-transactional changes, etc.
- GUI will need to devise generic support of an arbitrary number of user-supplied modules using AMX facilities.



# Impacts—CLI

---

- AMX should provide 90% compatibility.
- Issues are more or less the same as for GUI.
- Devising generic syntax for working with user-supplied modules.

# Impacts—Remote servers

---

- Mechanism for pushing config changes to remote servers must be understood and (possibly) reimplemented.

-

# Configuration changes—semantics

---

- What does it mean to change configuration?  
Immediate/synchronous? Deferred/asynchronous?
- Exclusive access and/or locking across changes?
- Requiring (or not) restart? (How to distinguish whether any particular value is actually in effect).

# Q&A

---

