Geronimo v1.1 User Requirements

> **Stale Content**
>
> The content of this page is growing stale and may or may not contain relevant, useful or correct information.

Here are the User Requirements as identified from the Geronimo User mailing list in April 2006.

(X) indicates the number of requests for the item.

Specifications/Functionality

- JDK 5 for Geronimo **(12)**
- JEE 5 **(4)**
  - Enterprise JavaBeans 3.0 **(5)**
  - JAX-WS support **(4)**
  - Java Servlet 2.5
  - Java ServerPages 2.1
  - Java ServerPages Standard Tag Library
  - Java Persistence Architecture (is this part of Java EE?)
  - Java Transaction API (JTA)
  - J2EE Management
  - J2EE Connector Architecture
- GBean improvements (doc, lifecycle, dependencies, dep. injection, ..) **(2)**
- Dynamic Queries **(2)**
- javax.persistence
- annotated session beans.
- Support for JSR-168 (Portlet API)
- ServiceMix
- Maven2 support
- service/daemon wrapper
- Remove the requirement of the openejb-jar.xml??
- Better db tools in the admin console
- Configuration management, possibility to make a production version without some current modules (e.g. OpenEJB or ActiveMQ), there are end users who don't have the resources (memory, disk) to run Geronimo fully equipped and they don't need every feature of the J2EE stack
- Continued support of Jetty
- First class HttpSession clustering.
- More integration with Apache mod_JK/mod_ajp . It would be nice if a request would continue through a pool until it landed on a server with that webapp.
- Mass deployment tools that allow the server 'cloning' and rollout mentioned during Geronimos initial days.
- Global JNDI ENC
- Integrate or provide as addons, things like Struts Frameworks, JSF, Pluto, etc.

Tools

- NetBeans support **(4)**
- JDK 5 for launching and running the Geronimo Eclipse plugin **(2)**
- Eclipse plugin improvement (it is really good but think it could be better)
- Eclipse mini-roadmap (from Sachin)
  - run resources directly from the workspace, so the ear isn't built and re-packaged on every publish
  - More control over the runtime/server wizards, publish process, and server management
  - Continue to build up the UI so we have a complete set of "form editors" for the G deployment plans
  - ability to see changes reflected live in something like the Common Navigator as you modify your plans
  - full synchronization between the source view and views using the model
  - copy/paste, undo/redo support
  - There's also plans in WTP to improve the Server Tools Framework to make it easier to have more control over what defines your runtime.

- Improved IDE integration, not only with Eclipse, but also include support for Netbeans, IntelliJ, etc. . .

Usability

- Deployment **(6)**
  - Less deployment requirements (simpler plans, more defaults, etc.)
  - Simplifying deployment (some means to generate geronimo deployment plans?
  - Easier way to deploy EAR Files
  - More application validation at deployment
  - Better redeployment to prevent requests from failing if they hit that server during the redeploy.
- More powerful text configuration
- Migration path from Tomcat to Geronimo
- Shortcuts for building web services
- Fully functional Console (Automate all tasks including creation of deployment plans) Focus on ease of use cause that is what general people need. I need my application running. I don't want to worry about the 1001 catches and work around or procedures.

Process

- More frequent releases incorporating more user feedback (small releases more often vs. big releases only 4 times a year) **(7)**
- Geronimo Certified Partner Program (including Jetspeed-2 as a member). **(2)**
- Need to get JIRAs assigned to owners quickly (too many unassigned JIRAs)
- Resolve dual wiki situation (moin-moin, confluence)

Documentation (11)

- Tutorials **(4)**
  - How to use Geronimo with: Apache Axis, WSS4J, ActiveMQ
- Cookbooks
- Better documentation on deployment descriptors
- More detailed documentation about the architecture and Gbeans **(2)**
- A browsable table describing where to find all plans, etc. for each deployed component or service.
- More documentaion on deploying EAR's, WAR's, EJB's, RAR's, classloading and dependencies with that apps)
- Better javadoc
- More and more documentations and easily available at one single place

Examples (4)

- Examples for everything
- More documentation/examples for me should be more explicative models of the basic openejb-jar.xml and ejb-jar.xml, explaining which tag points where or what
  - Session beans, entity beans(BMP and CMP)
  - Message-Driven-Bean.