



Java is a trademark of Sun Microsystems, Inc.

JavaOneSM

JavaTM Platform, Enterprise
Edition 5 and 6:
Eclipse and NetBeansTM IDE
Tooling Offering

TS-5055

Ludovic Champenois, Sun
Philippe Ombredanne, NexB

Agenda

- > Java EE 5 versus Java EE 6
- > GlassFish v3
- > NetBeans GlassFish Support
- > Eclipse GlassFish Support

Overall Goal For Java EE 6

- > Make the platform:
 - Easier to use
 - More flexible, adaptable
 - Easier to learn
 - Easier to evolve going forward

Java EE Timeline



Profiles

Java EE 6

EJB Lite
Restful WS
Web Beans
Extensibility

Java EE 6 Web Profile

Ease of
Development

Java EE 5

Ease of
Development
Annotations
EJB 3.0
Persistence API
New and
Updated
Web Services

Web
Services

J2EE 1.4

Web Services,
Management,
Deployment,
Async.
Connector

Robustness

J2EE 1.3 CMP, Connector

Architecture

Enterprise
Java
Platform

J2EE 1.2

Servlet, JSP,
EJB, JMS
RMI/IIOP

JPE
Project

Major new Features in Java EE 6

> Profiles

- targeted bundles of technologies

> Pruning

- Make some technologies optional
 - CMP, JAX-RPC...

> Extensibility

- Embrace open source libraries and frameworks
- Zero-configuration, drag-and-drop for web frameworks, web.xml fragments

Ease Of Development

- > Ongoing concern
- > This time focus is the web tier
- > Lots of opportunities in other areas, e.g. EJB
- > General principles:
 - Annotation-based programming model
 - Traditional API for advanced users
 - Reduce or eliminate need for deployment descriptors
 - Get technologies to work together well

Servlet 3.0 Highlights

- > Annotation-based programming model
 - `@WebServlet` `@ServletFilter` etc.
- > Modular web.xml descriptor:
 - `WEB-INF/lib/mylibrary.jar` → `META-INF/web-fragment.xml`
- > Annotations and web fragments are merged
- > Programmatic API for dynamic registration of servlets
- > Async APIs
 - Useful for Comet, chat rooms, long waits

EJB 3.1 Highlights

- > Singleton beans: `@Singleton`
- > No interface view: one source file per bean
- > Calendar timers:
 - `@Schedule(dayOfWeek="Mon,Wed")`
 - Non-persistent timers (tied to a JVM)
- > Async business methods: `@Asynchronous`
 - Methods must return void or a `Future<T>`
- > Global JNDI names for beans
 - `java:global/(app)/(module)/(bean)#(interface)`

Session Bean with Local Business Interface...OLD...

HelloBean Client

```
@EJB
private Hello h;

...

h.sayHello();
```

<<interface>

com.acme.Hello

String sayHello()



com.acme.HelloBean

```
public String
sayHello()
{ ... }
```

Optional Local Business Interfaces

- > Sometimes local business interface isn't needed
- > Better to completely remove interface from developer's view than to generate it
- > Result : “no-interface” view
 - Just a bean class
 - public bean class methods exposed to client
 - Same behavior and client programming model as Local view
 - Client acquires an EJB component *reference* instead of calling `new ()`

Session Bean with “No-interface” View

```
@Stateless
public class HelloBean {

    public String sayHello(String msg) {
        return "Hello " + msg;
    }

}
```

No-interface View Client

```
@EJB HelloBean h;
```

```
...
```

```
h.sayHello("bob");
```

Web/EJB Application in Java™ EE Platform 5

foo.ear

foo_web.war

WEB-INF/web.xml
WEB-INF/classes/
com/acme/FooServlet.class
WEB-INF/classes
com/acme/Foo.class

foo_ejb.jar

com/acme/FooBean.class
com/acme/Foo.class

OR

foo.ear

lib/foo_common.jar

com/acme/Foo.class

foo_web.war

WEB-INF/web.xml
WEB-INF/classes/
com/acme/FooServlet.class

foo_ejb.jar

com/acme/FooBean.class

Web/EJB Application in Java™ EE Platform 6

foo.war

WEB-INF/classes/
com/acme/FooServlet.class

WEB-INF/classes/
com/acme/FooBean.class

EJB “Lite”

- > **Small** subset of EJB 3.1 API for use in Web Profile
- > Broaden the availability of EJB technology
 - Without losing portability
- > Same exact Lite application can be deployed to Web Profile and Full Profile
 - Thanks to simplified .war packaging

“Lite” vs. Full Functionality

Lite

- > Local Session Beans
- > Annotations / ejb-jar.xml
- > CMT / BMT
- > Declarative Security
- > Interceptors
- > (Also requires JPA 2.0 API / JTA 1.1 API)

Full = Lite + the following:

- > Message Driven Beans
- > EJB Web Service Endpoints
- > RMI-IIOP Interoperability
- > 2.x / 3.x Remote view
- > 2.x Local view
- > Timer Service
- > CMP / BMP

EJB Component Testing

- > It's too hard to test EJB components, especially the Local view
 - Forced to go through Remote facade or Web tier
 - Separate JVM™ instances needed for server and client
- > Support for running in Java SE exists, but...
 - Not present in all implementations
 - No standard API for bootstrapping, component discovery, shutdown etc.

New Features

- > Singletons
- > Application startup / shutdown callbacks
- > Calendar-based timer expressions
- > Automatic timer creation
- > Simple Asynchrony

Singletons

- > New session bean component type
 - One singleton bean instance per application per JVM
 - Provides easy sharing of state within application
 - Designed for instance-level concurrent access
- > Lots in common with stateless / stateful beans
 - Client views (No-interface , Local, Remote, Web Service)
 - CMT / BMT
 - Container services: timer service, injection, etc.
 - Method authorization

Startup / Shutdown Callbacks

```
@Singleton
@Startup
public class StartupBean {

    @PostConstruct
    private void onStartUp() {
        ...
    }

    @PreDestroy
    private void onShutdown() {
        ...
    }
}
```

Timer Service Features

- > Calendar-based timeout expressions
- > Automatic timer creation
- > Non-persistent timers
- > “Cron”-like semantics with improved syntax
- > Named attributes
 - second, minute, hour (default = “0”)
 - dayOfMonth, month, dayOfWeek, year (default = “*”)

Calendar Based Timeouts

// The last Thursday in November at 2 p.m.

(hour="14", dayOfMonth="Last Thu", month="Nov")

// Every weekday morning at 3:15 a.m.

(minute="15", hour="3", dayOfWeek="Mon-Fri")

// Every five minutes

(minute="*/5", hour="*")

For ex:

// Callback the 1st of each month at 8 a.m.

```
@Schedule(hour="8", dayOfMonth="1")
```

```
void sendMonthlyBankStatements() {...}
```

Java EE 6: Summary

- > Smaller, more agile platform: profiles, pruning, extensibility
- > Ease of development still a major focus area
- > Component specs ready for public review
- > Open source implementation in GlassFish v3

Agenda

- > Java EE 6
- > GlassFish v3
- > NetBeans Glassfish Support
- > Eclipse GlassFish Support

GlassFish v3

- > Java EE 5 = GlassFish v2
 - > GlassFish v2.1: better, faster, more scalable
 - > GlassFish v3 Prelude
 - Web Tier Only, OSGi Based, some EE 6 previews
 - EclipseLink Bundled
 - Multiple Containers (jRuby, Groovy, Phobos)
 - > GlassFish v3 = Java EE 6
 - Work in Progress
 - OSGi Based
 - Equinox And Felix
- All these Application Servers have Eclipse Integration

Agenda

- > Java EE 6
- > GlassFish v3
- > NetBeans GlassFish Support
- > Eclipse GlassFish Support

NetBeans 6.7

- > Bundles with GlassFish v2.1 and v3 Prelude
- > Can download and register latest GlassFish v3 builds
- > Stellar Java EE 5 support
 - Web, Web Services, Jersey, JPAs, EJBs, EARs
- > Allow Java EE 6 development when GF v3 is used

NetBeans 6.7 and JAX RS support

- > Complete dev env for
 - Creating JPAs from DB
 - Creating JAX-RS RESTful from JPAs
 - Test the Resources
- > Demo

NetBeans 6.7 and GlassFish v3

- > JPA possible in Web Applications
- > Servlet 3.0: pure annotation based
- > Demo

Agenda

- > Java EE 6
- > GlassFish v3
- > NetBeans GlassFish Support
- > Eclipse GlassFish Support

New: GlassFish + Eclipse Bundle

GlassFish Tools Bundle For Eclipse 1.0

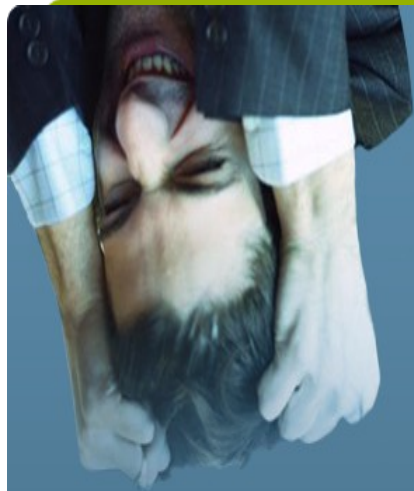
Release Candidate Build...



(c) Copyright Eclipse contributors and others, 2000,2009. All rights reserved. Java and all Java related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries, or both. Eclipse is a trademark of the Eclipse Foundation, Inc.

GlassFish Tools Bundle for Eclipse

- > <http://download.java.net/glassfish/eclipse>
 - V1.0 today, with support available
- > Or Standalone Plugin
 - <http://glassfishplugins.dev.java.net>



Eclipse Java EE IDE

GlassFish v2.1, v3prelude
JDK1.6 (optional)

GlassFish Eclipse Plugins

Registration

Installer

Eclipse

User

Workspace:

- > v2.1 domain
- > v3 domain
- > JavaDB config
- > Java EE projects
- > User settings



GlassFish Tools Bundle For Eclipse Key Features

- > Out of the box Installer with all Java EE Eclipse Standard Features and JDK and GlassFish servers
- > GlassFish Registration Wizard
- > GlassFish v2.1 & v3 Prelude automatic domain creation/configuration
- > JavaDB configuration, JDBC resource wizard
- > Start, Stop, Deploy, undeploy, Debug(JSP/Java) (v2, v3)
- > Deploy on Save: Default for v3
- > HTTP Monitoring preconfigured

GlassFish Tools Bundle For Eclipse Key Features

- > All Sun DTDs registered for validation/code completion
- > All Java EE APIs registered for code completion/JavaDoc
- > GlassFish Log integrated into Eclipse IDE console
- > GlassFish Update Center & Admin Console Integration
- > All v2, v3p DocBooks integrated in Help (no need for Internet)
- > GlassFish Web Properties in Help Menu (The Aquarium, Support,...)

GlassFish Tools Bundle For Eclipse Key Features

- > Update Centers Integration
- > Eclipse Update Center for Eclipse bits and the GlassFish Plugin
- > In future releases:
 - Metro JAX-WS Eclipse Plugin
 - JavaFX ,Maven Plugin
 - WebSynergy ,GlassFish ESB
 - Better MySQL integration

Size of GlassFish Tools Bundle for Eclipse

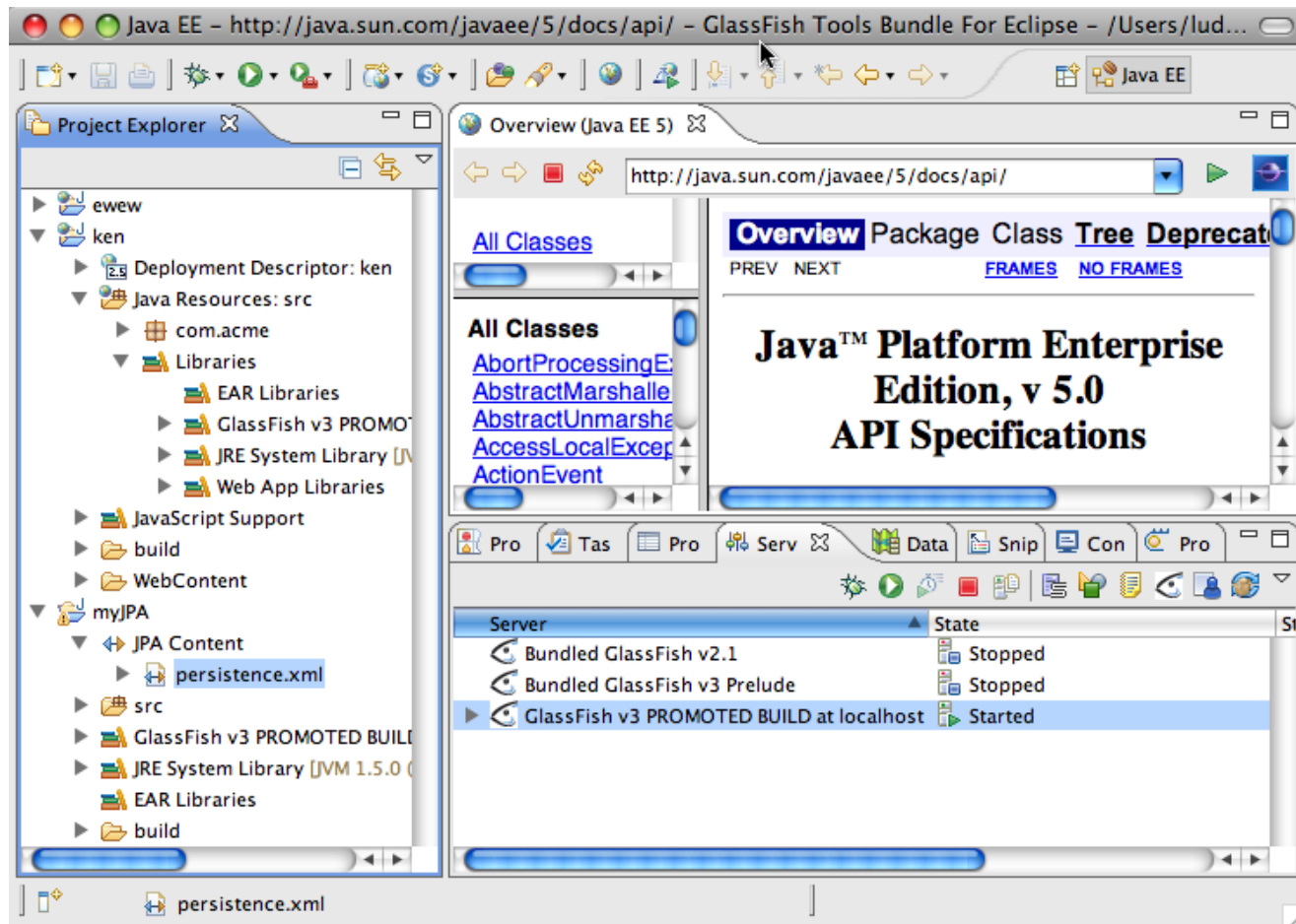
Components in the Installer	Size (total=375Mb)
• Eclipse	164Mb
• GlassFish v2.1	87Mb
• GlassFish v3 Prelude	29Mb
• GlassFish Plugin for Eclipse	10Mb (includes javadoc and help books)
• Registration/Configuration	0.5Mb
• JDK 1.6u12	85Mb

- > JBoss Dev Studio 2.0 RC2 +EAP = 617Mb (No JDK)
- > Oracle Workshop for WebLogic 10.2 = 748Mb (No JDK)

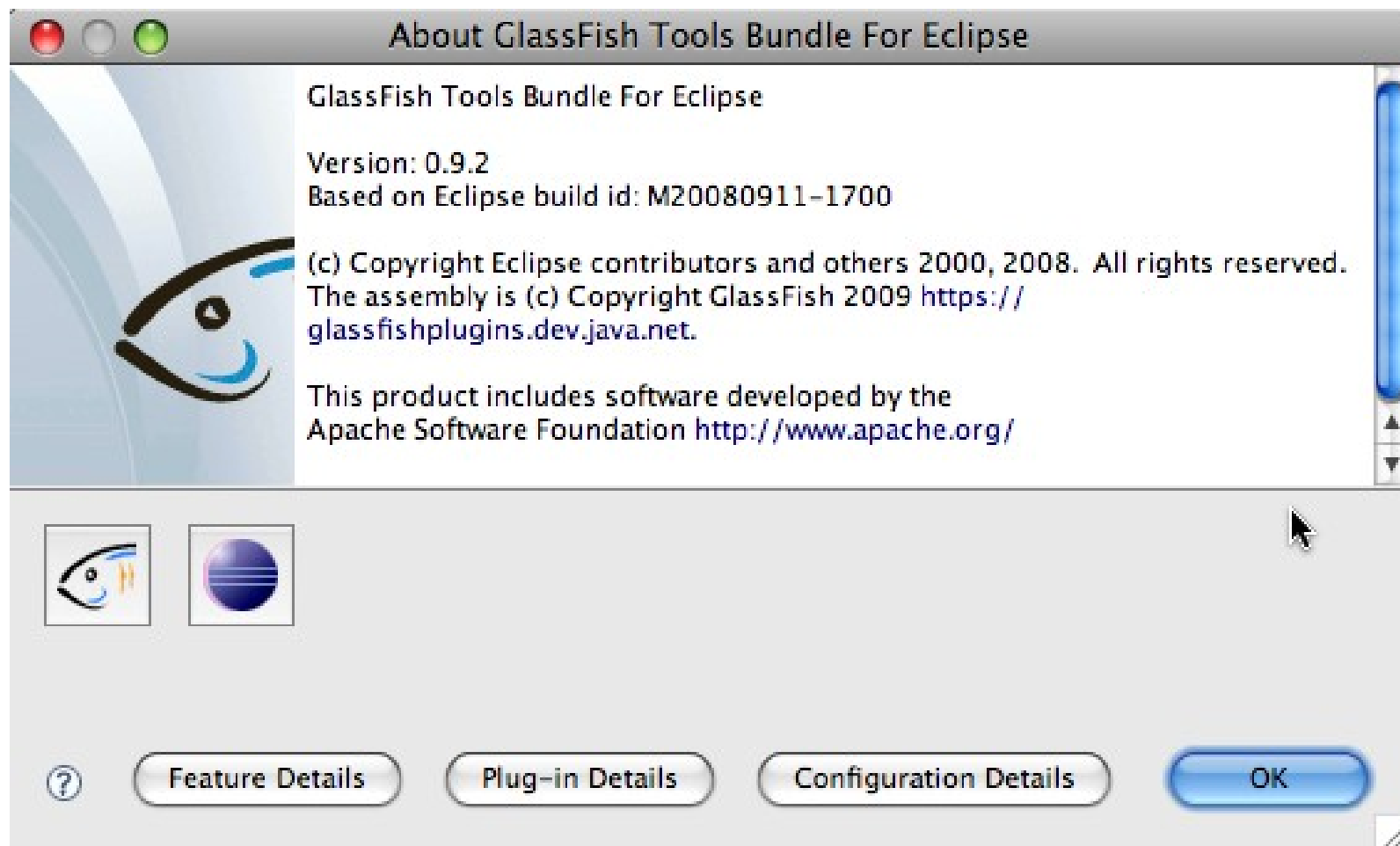
Collaboration work with EasyEclipse

- > Re-use the open source build loop for distros
 - Installers on MacOS X, Windows, Linux, OpenSolaris
- > Help in assembling a complete IDE
 - Using Eclipse.org, Glassfish and community plug-ins
- > Why doing a distros and installers?
 - Batteries included approach
 - Yet minimalist essential set of features to get going
 - <http://blog.hantsuki.org/2009/03/20/eclipse-and-installers/>
- > Tight work with core WTP and JPA committers
 - Bug fixing in P2 compatibility

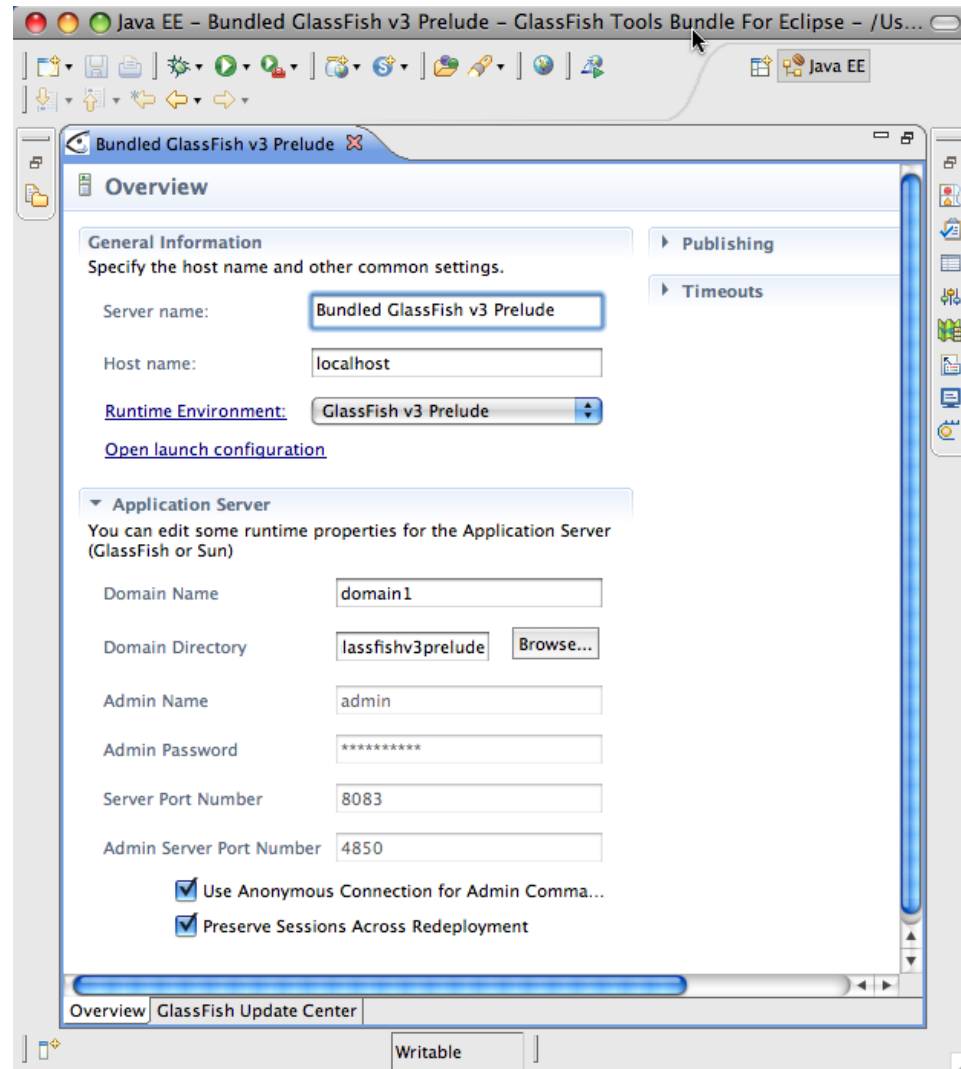
GlassFish Tools Bundle for Eclipse



GlassFish Tools Bundle for Eclipse



GlassFish Tools Bundle for Eclipse



GlassFish Tools Bundle for Eclipse

- ▶ Deployment Descriptor: ewew
- ▼ Java Resources: src
 - ▼ foo
 - ▶ MyServlet.java
 - ▼ Libraries
 - ▼ EAR Libraries
 - ▼ GlassFish v3 Prelude [GlassFish v3 Prelude]
 - ▶ jsf-api.jar - /Applications/GlassFish-Tools-Bu
 - ▶ jsf-impl.jar - /Applications/GlassFish-Tools-E
 - ▶ jstl-impl.jar - /Applications/GlassFish-Tools-
 - ▶ javax.activation.jar - /Applications/GlassFish-
 - ▶ javax.annotation.jar - /Applications/GlassFish
 - ▶ javax.enterprise.deploy.jar - /Applications/Gla
 - ▶ javax.mail.jar - /Applications/GlassFish-Tools
 - ▶ javax.persistence.jar - /Applications/GlassFis
 - ▶ javax.resource.jar - /Applications/GlassFish-7
 - ▶ javax.security.auth.message.jar - /Application
 - ▶ javax.security.jacc.jar - /Applications/GlassFis
 - ▶ javax.servlet.jar - /Applications/GlassFish-To
 - ▶ javax.servlet.jsp.jar - /Applications/GlassFish-
 - ▶ javax.transaction.jar - /Applications/GlassFis
 - ▶ javax.xml.stream.jar - /Applications/GlassFish
 - ▶ javax.servlet.jsp.jstl.jar - /Applications/GlassF
 - ▶ org.eclipse.persistence.antlr.jar - /Application
 - ▶ org.eclipse.persistence.asm.jar - /Application:
 - ▶ org.eclipse.persistence.core.jar - /Application
 - ▶ org.eclipse.persistence.jpa.jar - /Applications,
 - ▶ org.eclipse.persistence.oracle.jar - /Applicatic

For More Information

- > <http://download.java.net/glassfish/eclipse>
- > <http://glassfishplugins.dev.java.net>
- > Blog : <http://weblogs.java.net/blog/ludo/>
- > Reference Implementation : GlassFish project v3
 - <http://glassfish.dev.java.net>



JavaOneSM

Thank You

Ludovic Champenois
Philippe Ombredanne
<http://weblogs.java.net/blog/ludo/>

