



A Quick Tour of the CDI Landscape

Reza Rahman

Expert Group Member, Java EE 6 and EJB 3.1

Resin EJB 3.1 Lite Container Developer

Author, EJB 3 in Action

reza@caucho.com

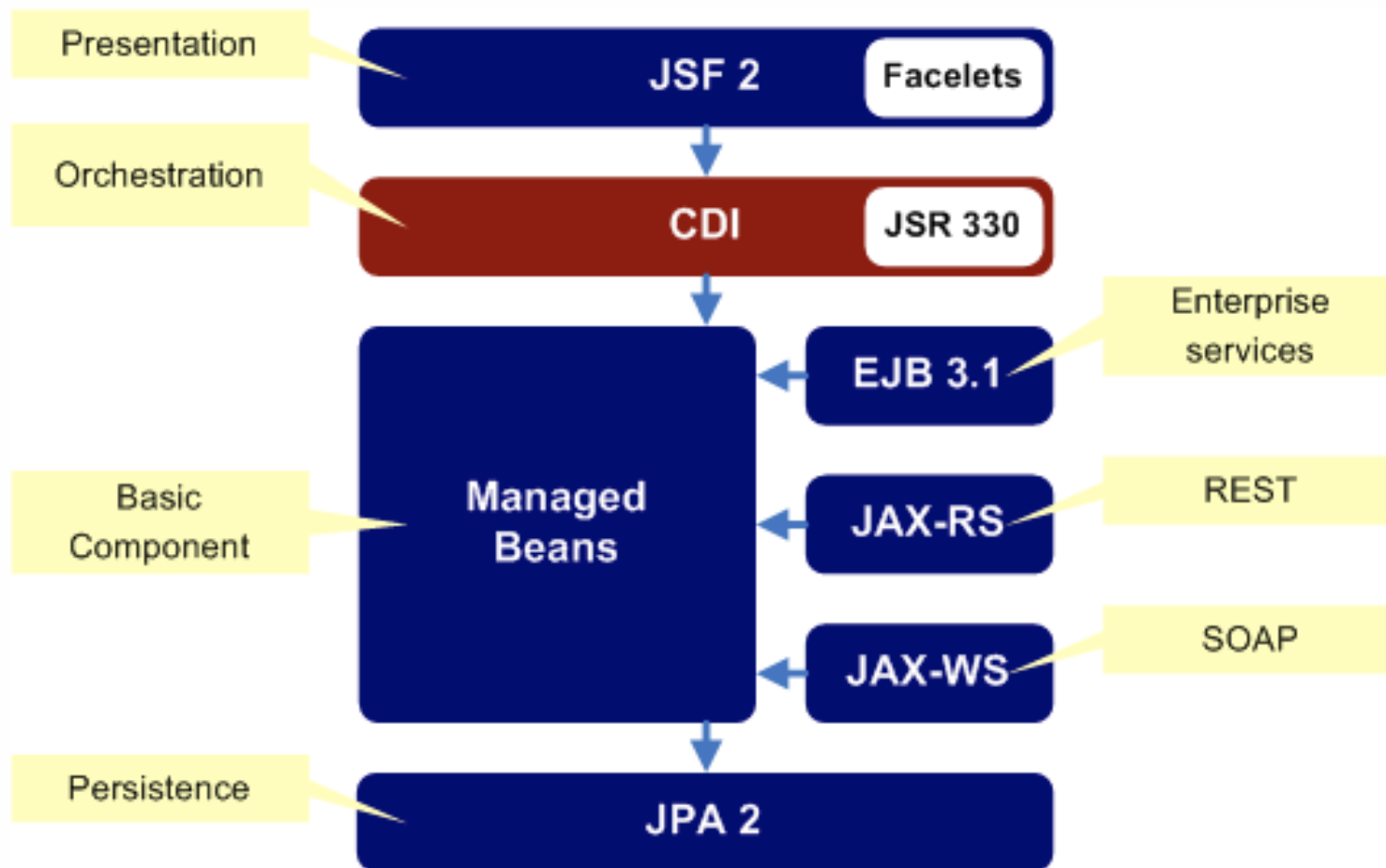


Contexts and Dependency Injection for Java EE (CDI)

- **Next-generation dependency injection for Java EE**
- **Synthesizes best ideas from Seam 2, Guice and Spring**
- **Many innovative features on its own right**
- **Focus on loose-coupling, Java-centric type-safety, annotations, expressiveness and ease-of-use**
- **Makes Java EE much more flexible, testable, pluggable and extensible**
- **Large ecosystem rapidly evolving around CDI**



CDI and Java EE





CDI Features

- **Basic dependency injection**
 - **@Inject, @Qualifier, @Stereotype, @Alternative, Instance, @All, @Any**
- **Component naming**
 - **@Named**
- **Context management**
 - **@Dependent, @RequestScoped, @SessionScoped, @ConversationScoped, @ApplicationScoped, @Scope**



CDI Features

- **Custom Object Factories**
 - **@Produces, @Disposes, InjectionPoint**
- **Lightweight Events**
 - **Event, @Observes**
- **Interceptors/Decorators**
 - **@Interceptor, @InterceptorBinding, @AroundInvoke, InvocationContext, @Decorator, @Delegate**
- **Portable extensions SPI**



CDI Ecosystem

Implementations



Weld



CanDI



OpenWebBeans

Portable Extensions



Tools



JBoss Tools



NetBeans



IntelliJ IDEA



CDI Implementations





Weld

- **CDI reference implementation and compatibility test kit from JBoss**
- **Weld reference excellent documentation for CDI**
- **Evolved from Seam 2 core**
- **Included in GlassFish and JBoss AS**
- **Can be used with Tomcat, Jetty**
- **Can be used in standalone Java SE**



CanDI



- **Caucho implementation included in the Resin application server**
- **CanDI forms the core of Resin itself**
- **Compact, schema-driven, type-safe XML configuration**
- **CDI XML used to configure Resin itself**
- **Use of annotations like @TransactionAttribute, @Schedule, @Asynchronous, @RolesAllowed, @RunAs, @Lock, @Startup and @Remote outside EJB in plain CDI managed beans**

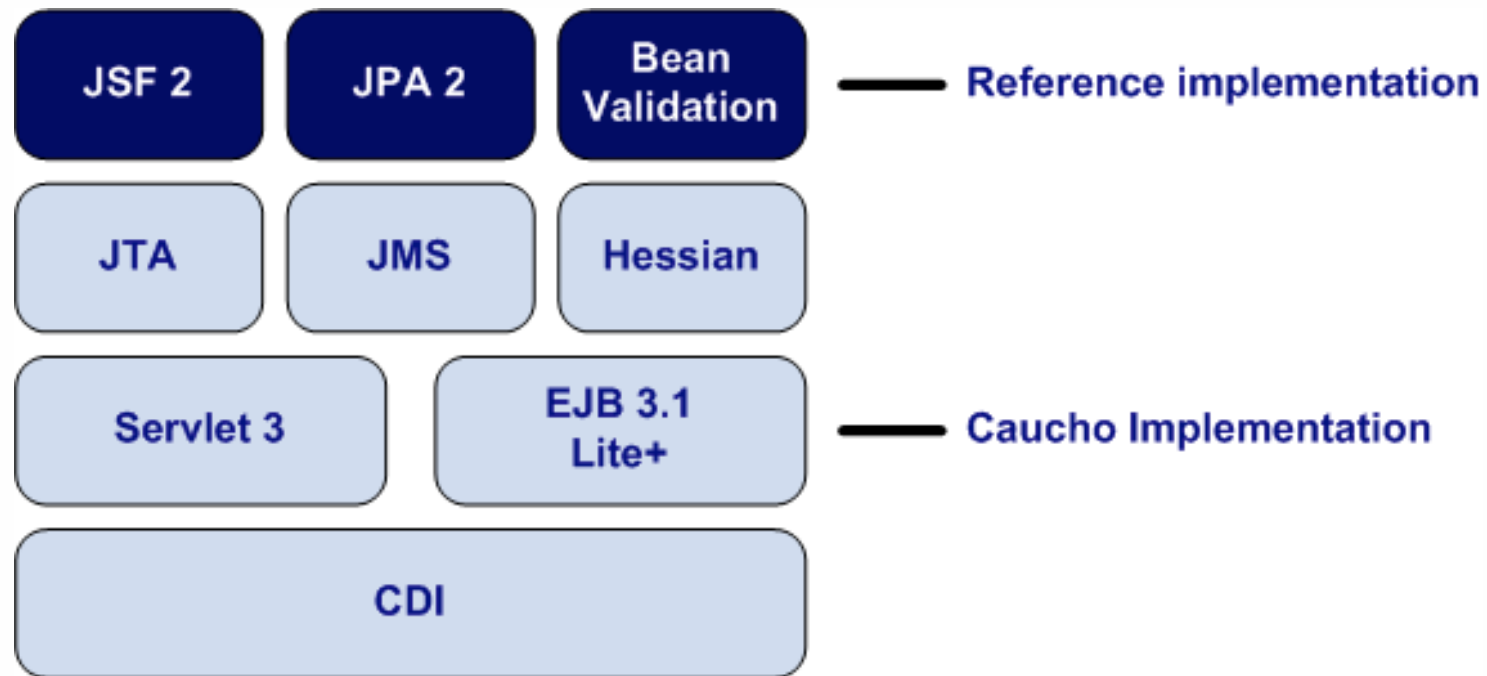


CanDI

- **@Pooled, @TransactionScoped, @ThreadScoped
custom scopes**
- **Embedded container based testing framework
integrating CDI, EJB, JPA with JUnit**
- **Integrating CDI with PHP**
- **Portable extensions in progress for JDBC, iBATIS, Struts
2, Quartz**
- **Focus on performance**



CanDI and Resin





OpenWebBeans



- **Apache implementation of CDI included in Geronimo**
- **Closely related to Apache OpenEJB**
- **Great integration with Tomcat along with OpenEJB**
- **Extended injection of JMS objects (sessions, connection factories)**
- **Runs in standalone Java SE**
- **Portable extensions for major Apache projects**

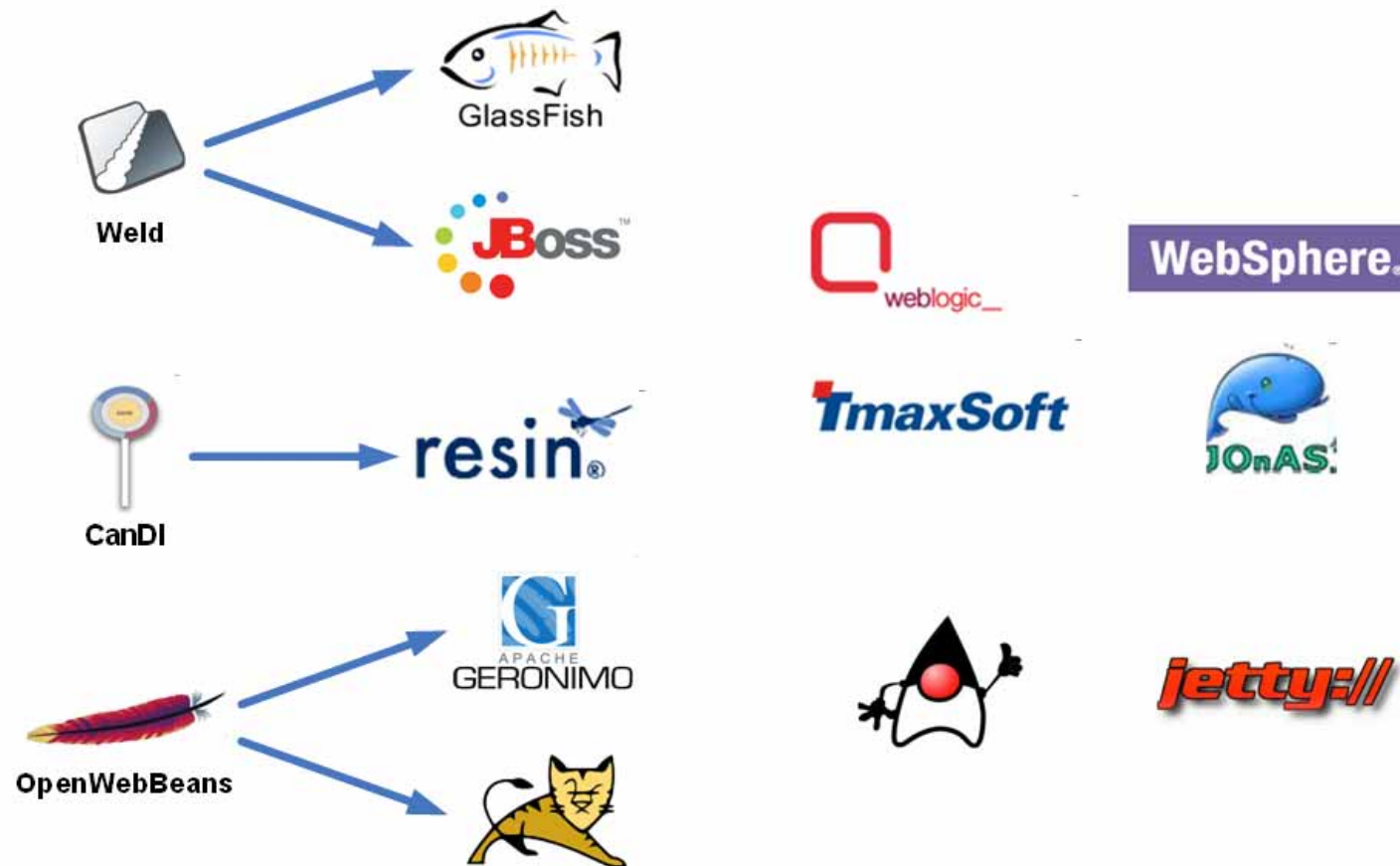


Java EE 6 Implementations





Java EE 6 Application Servers





GlassFish v3



- **Open source Java EE 6 reference implementation**
- **Fully backed by Oracle**
- **Earliest Java EE 6 runtime**
- **Web Profile as well as Full Platform supported**
- **OSGi based modular architecture**
- **Excellent pooling, clustering and administration features**
- **Embedded container for testing**



JBoss AS 6



- **Most widely used open source Java application server**
- **Recently passed Java EE 6 Web Profile TCK**
- **Java EE 6 Full Profile implementation on the way**
- **Track record for innovation and thought leadership**
- **Pooling, caching, clustering, admin console**
- **HornetQ excellent JMS implementation**



Resin 4



- **Focus on fast, lightweight, reliable open source Java EE 6 Web Profile server**
- **Scheduling, asynchronous processing, Hessian remoting, message-driven beans and lightweight JMS also included**
- **HTTP server, proxy caching, load-balancing, pooling, clustering, caching, cloud support, security framework, SSL, administration**
- **Running PHP on Java**



Geronimo 3

- **Apache licensed open source application server**
- **Basis for IBM WebSphere Community Edition**
- **Geronimo 3 aimed to pass Java EE 6 Web Profile TCK shortly with a Full Profile implementation to follow**
- **OSGi, clustering, administrative console**
- **Includes Tomcat/Jetty, MyFaces, OpenWebBeans, OpenEJB, OpenJPA, ActiveMQ, CXF/Axis, Derby**
- **TomEE Tomcat centric Java EE 6 Web Profile offering**

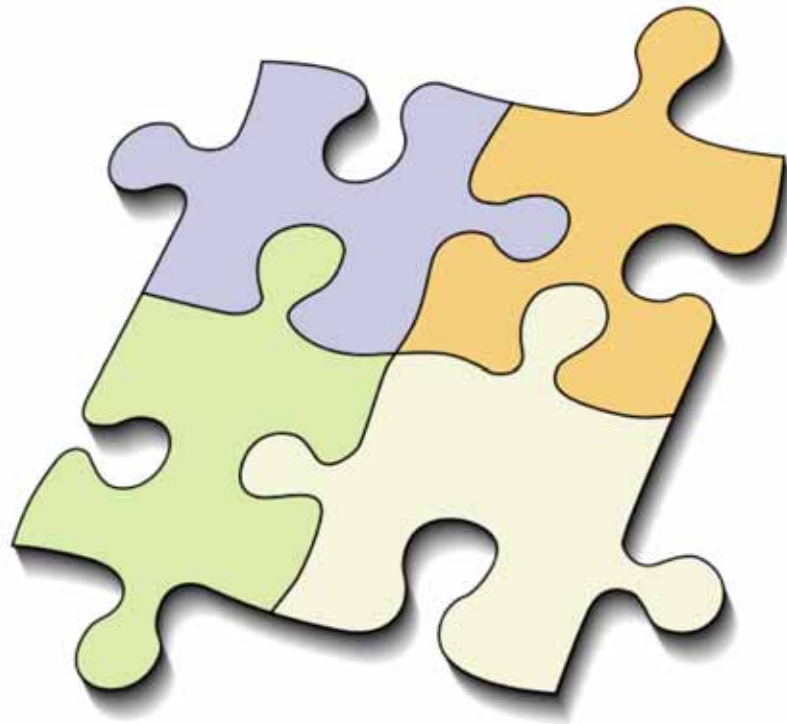


More on the Way...

TmaxSoft JEUS	Korean application server; JEUS 7 one of the earliest to get Java EE 6 certified; still in beta stage
WebLogic	Next release, WebLogic Server 11g R2, will support both Java EE 6 Full and Web Profile; planned for calendar year 2011
WebSphere	WebSphere 8 beta previews Java EE 6
JOnAS	Open source application server from OW2 consortium; JOnAS 5.2 implementing some Java EE 6 features; JOnAS 6 will support Java EE 6
Slwpas	Tomcat based open source Java EE 6 Web Profile offering leveraging Apache projects from Turkish company MechSoft; not certified yet



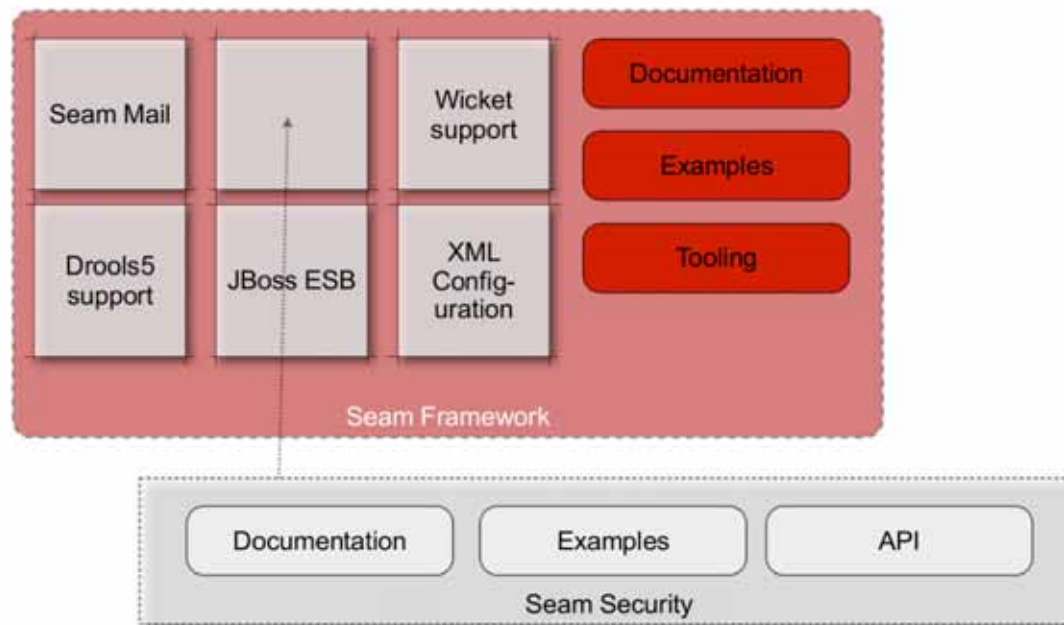
Portable Extensions





Seam 3

- Set of CDI portable extensions independent of Weld/JBoss AS
- Each module developed separately
- There are Seam “umbrella” releases





Seam Modules

Module	Description
Solder	General enhancements to CDI
XML Configuration	XML configuration for managed beans
Persistence	Transactions and persistence for non-EJB managed beans
Faces	JSF enhancements
Servlet	Servlet enhancements
JMS	JMS integration
REST	REST enhancements
Remoting	JavaScript remoting of managed beans



Seam Modules

Module	Description
Catch	Exception handling framework
International	Internationalized/localized locale, time, messages
Security	Higher level security API for Java EE
Mail	JavaMail integration
Cron	CDI based scheduling
Document	Document generation (PDF, Excel)
Seam 2	Seam 2 backwards compatibility
Spring	Spring interoperability



Seam Modules

Module	Description
Wicket	Wicket Integration
GWT	GWT integration
Drools	Drools integration
jBPM	jBPM integration
JBoss ESB	JBoss ESB integration
Clouds	Cloud frameworks integration (JClouds, Infinispan)



Apache MyFaces Codi

- **Set of CDI portable extensions from Apache**
- **Somewhat MyFaces/JSF centric, aims to provide a friendly “toolbox” for development with Java EE 6**
- **Modular, configurable and extensible**
- **Closely related to Apache OpenWebBeans**



Using JSF Project Stage

```
<context-param>  
    <param-name>javax.faces.PROJECT_STAGE</param-name>  
    <param-value>Development</param-value>  
</context-param>
```

```
@Inject private ProjectStage projectStage;
```

```
...
```

```
boolean isDevProjectStage =  
    ProjectStage.Development.equals(projectStage);
```

```
@Alternative
```

```
@ProjectStageActivated(ProjectStage.UnitTest.class)
```

```
public class TestServiceMockImpl implements Service {  
    ...  
}
```



Injecting Loggers

```
public class DefaultAccountDao
    implements AccountDao {
    @Inject
    private Logger logger;

    @PersistenceContext
    private EntityManager entityManager;

    public void addAccount(Account account) {
        ...
    }
    ...
}
```



CODI Conversations

```
@Named @ConversationScoped
// @ConversationGroup(Wizard1.class)
public class CodiConversationBean {
    @Inject private Conversation conversation;
    ...
    public void finishWorkflow() {
        ...
        conversation.close();
        ...
    }
}
```



Other CODI Scopes

@Named @ViewScoped

```
public class MyViewScopedBean {  
    ...  
}
```

@Named @WindowScoped

```
public class MyWindowScopedBean {  
    @Inject private Conversation conversation;  
    ...  
}
```



Enhanced JSF Injection

@Advanced

@FacesValidator(...)

```
public class DependencyInjectionAwareValidator
    implements Validator {
    @Inject FacesContext facesContext;
    @Inject ValidationService validationService;
    ...
    public void validate(FacesContext facesContext,
        UIComponent component, Object value)
        throws ValidatorException {
        Violation violation =
            validationService.validate(value);
        ...
    }
}
```



JSF Life-Cycle Listeners

```
public void onPreRender(@Observes
    @BeforePhase(RENDER_RESPONSE)
    PhaseEvent phaseEvent) {
    ...
}

public void onFacesRequestStart(@Observes
    @BeforeFacesRequest FacesContext facesContext) {
    ...
}
```



Injecting Bean Validation Objects

```
@Inject @Advanced
```

```
private Validator validator;
```

```
@Inject @Advanced
```

```
private ValidatorFactory validatorFactory;
```



Injecting into Validators

```
public class UsernameValidator implements
    ConstraintValidator<Username, String> {
    @Inject private UserService userService;

    public boolean isValid(String userName,
        ConstraintValidatorContext
            constraintValidatorContext) {
        return userService.validateUserName(userName);
    }
}
```



CODI Transactions

```
public class DefaultAccountService
    implements AccountService {
    @Inject
    private AccountDao accountDao;

    @Transactional
    public void addAccount(Account account) {
        accountDao.addAccount(account);
    }
    ...
}
```



Scripting (JSR-223) Integration

```
@Inject @ScriptLanguage(JavaScript.class)
private ScriptExecutor scriptExecutor;
...
return scriptExecutor.eval("10 + 4", Double.class);
```

```
@Inject @ScriptLanguage(JavaScript.class)
private ScriptBuilder scriptBuilder;
...
return scriptBuilder.script("x + y").namedArgument(
    "x", a).namedArgument("y", b).eval(Double.class);
```

```
<h:outputText value="#{sExec.js['[a:#{bean1.result},
    b:#{bean2.result}'] ['a + b']}"/>
```



Cargo JUnit Test

```
@RunWith(JUnit4WithCargo.class)
public class DemoTestCase extends AbstractCargoTest {
    @Test
    public void doSomeTest() throws Exception {
        SimplePageInteraction interaction = new
            SimplePageInteraction(getTestConfiguration())
                .with(Pages.Page1.class).with(
                    Pages.Page1.class).start(
                        Pages.Page1.class);
        interaction.useForm("form1").setValue(
            "form1:input1", "1").click("form1:button1");
        interaction.checkState(Pages.Page2.class)
            .checkTextValue("output1", "1");
    }
}
```



Plain JUnit Test

```
@RunWith(JUnit4.class)
public class BidServiceTest extends AbstractTest {
    @Inject private BidService bidService;
    ...
    @Test
    public void testGetBid() {
        Bid bid = bidService.getBid(1L);
        assertNotNull(bid);
    }
    ...
}
```



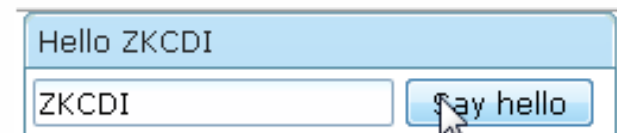
ZK Framework CDI Integration

- **ZK is Ajax-based event-driven RIA engine with a rich set of XHTML and XUL components**
- **ZUML mark-up language for creating UIs**
- **ZK Provides powerful CDI integration, including remote contexts (+ custom scopes)**



ZUML Markup

```
<?page title="HelloWorld"
      contentType="text/html; charset=UTF-8"?>
<?variable-resolver class=
      "org.zkoss.zkplus.cdi.DelegatingVariableResolver"?>
<zk>
  <window id="helloWindow" title="HelloWorld"
        border="normal" width="250px"
        apply="${helloWorld}">
    <textbox id="guestName" width="150px" />
    <button id="sayHelloBtn" label="Say hello"/>
  </window>
</zk>
```





ZK CDI Managed Bean

@Named

@SessionScoped

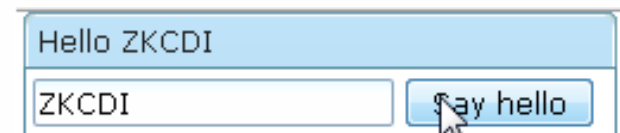
```
public class HelloWorld extends GenericComposer  
    implements Serializable {
```

```
    @Inject @ComponentId("guestName") Textbox guestName;
```

```
    @Inject @ComponentId("sayHelloBtn") Button sayHelloBtn;
```

```
    @Inject @ComponentId("helloWindow") Window helloWindow;
```

```
    public void sayHello(@Observes  
        @Events("sayHelloBtn.onClick") MouseEvent evt) {  
        helloWindow.setTitle("Hello "  
            + guestName.getValue());  
    }  
}
```





Tools





Arquillian

- **Embedded or remote (or managed) container integration with JUnit/TestNG**
- **Container life-cycle management, micro-deployments, injection into unit tests**
- **Supports multiple containers and aims to be container-independent**
- **Ideal for cross-vendor unit testing**
- **Integrates with JSFUnit**



Arquillian Supported Containers

Container	Mode
JBoss AS 5	Remote, managed
JBoss AS 6	Remote, managed, embedded
GlassFish 3	Remote, embedded
Tomcat 6	Embedded
Jetty 7	Embedded
Weld SE	Embedded
OpenWebBeans	Embedded
OpenEJB	Embedded



Arquillian Unit Test

```
@RunWith(Arquillian.class)
```

```
public class BidServiceTest {
```

```
    @Deployment
```

```
    public static WebArchive createDeployment() {
```

```
        return ShrinkWrap.create(WebArchive.class, "test.war")
```

```
            .addClasses(BidService.class, DefaultBidService.class,  
                        BidDao.class, DefaultBidDao.class, Bid.class)
```

```
            .addWebResource(EmptyAsset.INSTANCE, "beans.xml")
```

```
            .addWebResource("test-persistence.xml",  
                            "classes/META-INF/persistence.xml");
```

```
    }
```

```
    @Inject private BidService bidService;
```

```
    ...
```

```
    @Test
```

```
    public void testGetBid() {
```

```
        Bid bid = bidService.getBid(1L);
```

```
        assertNotNull(bid);
```

```
    }
```

```
    ...
```

```
}
```



Forge

- **Command-line RAD code-generation tool for Java EE**
- **Based on Maven**
- **Aimed at fast bootstrap and refactoring**
- **Highly pluggable and extensible**
- **Complements JBoss Tools CDI/Java EE 6 support**



Forge Plug-in Example

```
public class ExamplePlugin implements Plugin {
    @Command(name="perform")
    public void exampleCommand(
        @Option(name="one") String one,
        @Option(name="two") String two,
        @Option String three,
        @Option String four,
        PipeOut out) {
        out.println(">> first option equals: " + one);
        out.println(">> second option equals: " + two);
        out.println(">> third option equals: " + three);
        out.println(">> fourth option equals: " + four);
    }
}
```

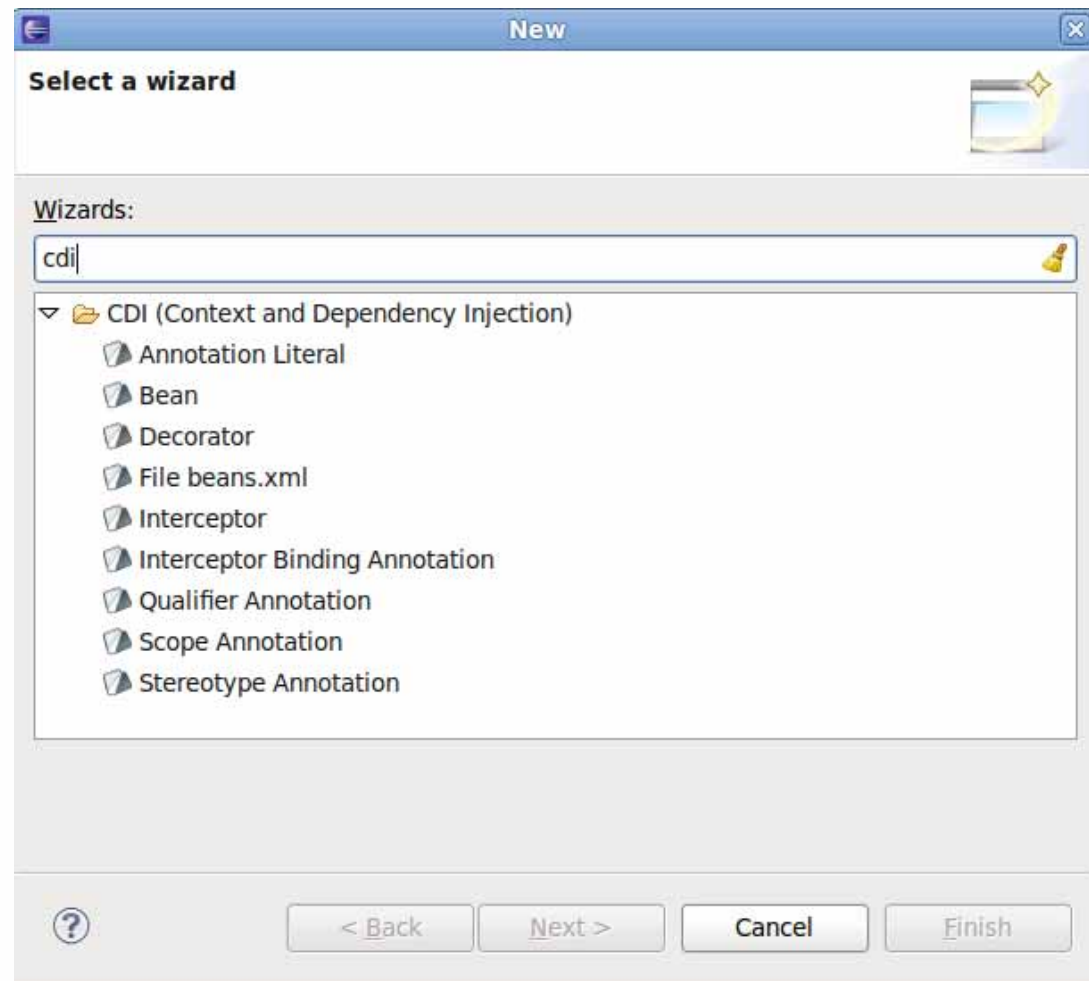


JBoss Tools

- **A set of Eclipse plug-ins for JBoss technologies like CDI, Seam, JSF, Facelets, Hibernate, JBoss AS, Drools, jBPM and JBoss ESB**
- **Provides excellent support for CDI (which Eclipse currently does not)**
- **CDI wizards, validation, quick-fix, auto-completion, navigation**

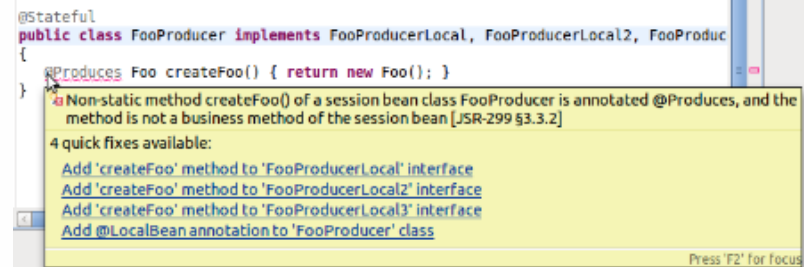
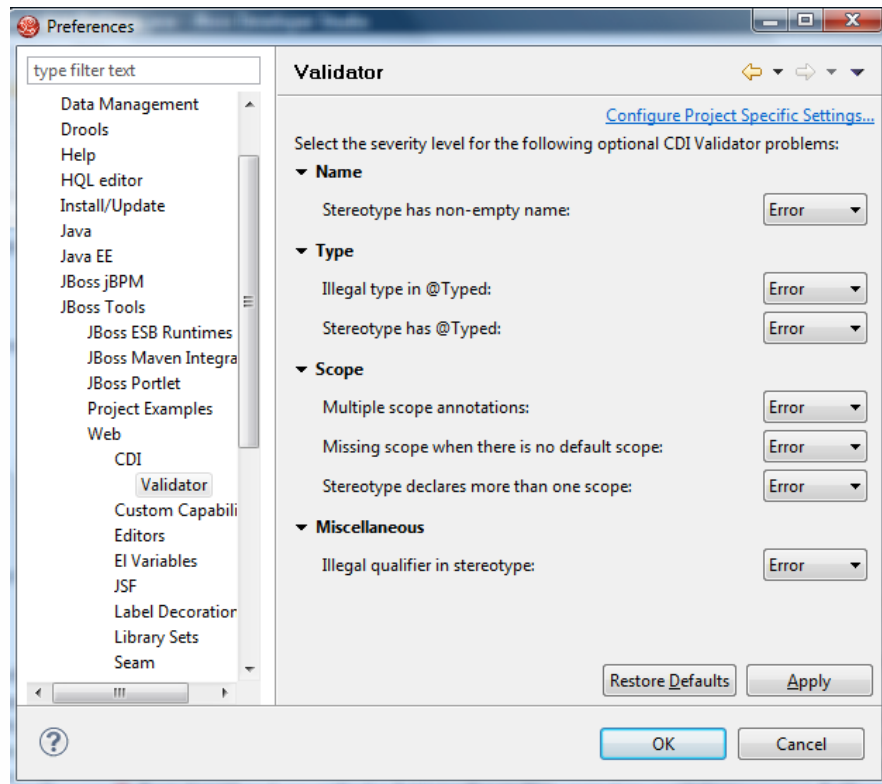


CDI Wizards





CDI Validation





CDI Code Navigation

```

1 package org.jboss.jsr299.tck.tests.jbt.resolution;
2
3 import javax.inject.Inject;
4
5 public class InjectionInstance {
6
7     @Inject String multipleBroken;
8
9
10
11 }
    
```

Context menu for `@Inject String multipleBroken;`:

- Open Declaration
- Open @Inject Bean BeanWithStaticProducerMethod.getString()
- Show All Eligible Beans...

```

@Inject @Any @NonRuntimeBindingType Event<String> stringEventWithAnyAndNonRuntimeBindingType;
@Inject @NonRuntimeBindingType Event<String> stringEventWithOnlyNonRuntimeBindingType;

public void fireEvent()
{
    stringEvent.fire("event");
}
    
```

Context menu for `stringEvent`:

- Open Declaration
- Open Injected Bean
- Open CDI Observer Methods

```

@Produces
{
    ...
}
    
```

Context menu for `@Produces`:

- Open Declaration
- Open Bound Disposer Method dispose



Eclipse

- **Some Java EE 6 support (EJB 3.1, Servlet 3, Facelets, JPA 2) but not complete**
- **CDI not supported at all**
- **GlassFish plug-in includes a little better support for Java EE 6, but not much**
- **Using JBoss Tools with Eclipse good choice for CDI developers for the time-being**

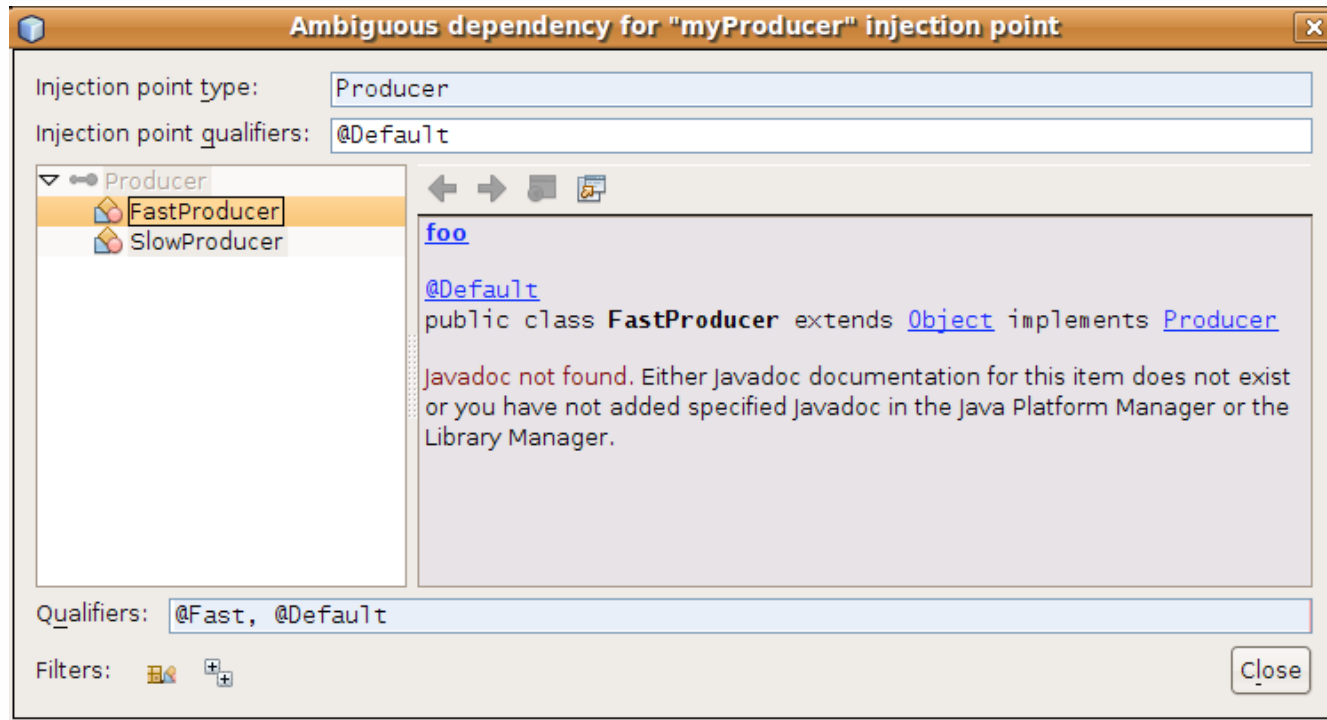


NetBeans

- **Excellent support for CDI and Java EE 6**
- **CDI wizards, validation, quick-fix, auto-completion, navigation, dependency graph**



CDI Validation



```
@Inject
```

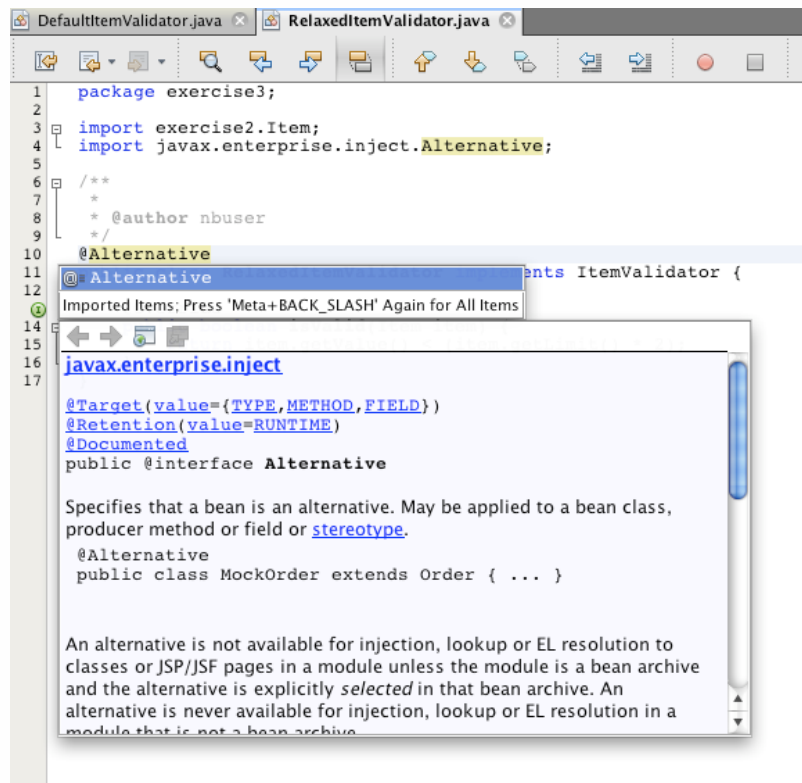
```
public Producer myProducer;
```

Choose element eligible for injection

FastProducer

SlowProducer

CDI Auto-Completion



```

5
6 package newpackage;
7     cannot find symbol
8     symbol: class NewQualifier
9
10    @NewQualifier
11
12    Create Qualifier NewQualifier in newpackage
13
14 }

```

```

1  <?xml version='1.0' encoding='UTF-8' ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3  <html xmlns="http://www.w3.org/1999/xhtml"
4      xmlns:h="http://java.sun.com/jsf/html">
5      <h:head>
6          <title>Facelet Title</title>
7      </h:head>
8      <h:body>
9          Hello from Facelets
10         <br/>
11         Message is: #{messageServerBean.message}
12         <br/>
13         Message Server Bean: #{messageServerBean}
14     </h:body>
15 </html>
16

```



CDI Navigation

```

6  package newpackage;
7
8  import javax.enterprise.event.Event;
9  import javax.enterprise.inject.Any;
10 import javax.inject.Inject;
11
12
13
14 public class NewClass {
15
16     @Inject @Any Event<EventObject> event
17 }
18

```

Navigate		Go to Source	Ctrl+Shift+B
Show Javadoc	Alt+F1	Go to Declaration	
Find Usages	Alt+F7	Go to Test	Ctrl+Shift+T
Call Hierarchy		Go to Super Implementation	
		Go to Implementation	Ctrl+Alt+B
Insert Code...	Alt+Insert	Inspect Members	Ctrl+Shift+F12
Fix Imports	Ctrl+Shift+I	Inspect Hierarchy	Alt+Shift+F12
Refactor			
Format	Alt+Shift+F	Go to Injectable	Ctrl+Alt+J
Run File	Shift+F6	Inspect CDI	Ctrl+Alt+C
Debug File	Ctrl+Shift+F5		
Test File	Ctrl+F6		
Debug Test File	Ctrl+Shift+F6		



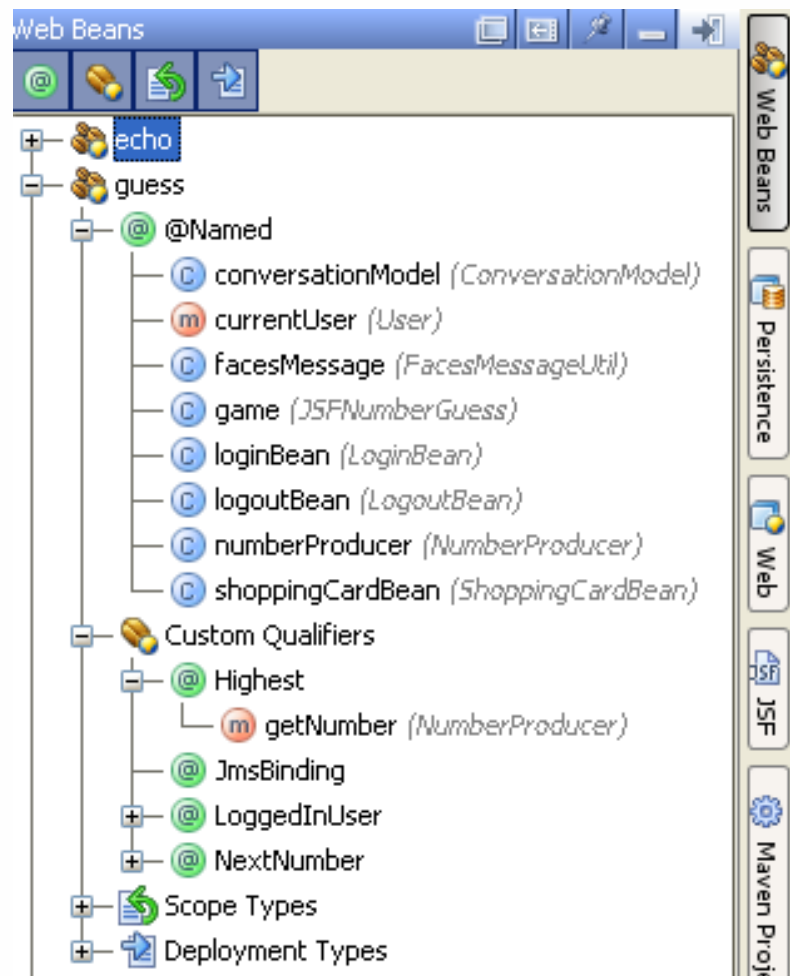


IntelliJ IDEA

- **CDI and Java EE 6 strongly supported**
- **Dependency diagrams, refactoring, navigation, code completion**
- **One of the earliest to support CDI**



CDI Tool Window





CDI Validation

```
public class InjectInMe {
    @Inject
    Credo credos ;
    @Inject
    Credo2 credo2 ;
}
```

Ambiguous dependency: there are multiple beans that match the injection point [more...](#)

Unsatisfied dependency: no bean matches the injection point [more...](#) (Ctrl+F1)

```
@SessionScoped @Named
public class Login implements Serializable {
    @Inject Credentials credentials;
}
```

Injected normal scoped bean is not proxyable [more...](#) (Ctrl+F1)

Unproxyable bean types inconsistency.
Certain legal bean types cannot be proxied by the container: classes which don't have a non-private constructor with no parameters, classes which are declared final or have final methods, primitive types, and array types.

Documentation for Credentials

login

```
org.jboss.weld.examples.login
@RequestScoped
@Named
@Default
public final class Credentials
extends Object
implements Serializable
```



Summary

- **CDI next generation dependency injection for Java EE**
- **Rich ecosystem of implementations, supported runtime platforms, portable extensions and tools rapidly evolving around CDI**
- **Portable extensions for both ease-of-use, Java EE enhancements as well as open source third-party tools**
- **Tools for RAD/code generation, testing as well as traditional IDE based development**
- **More on the way!**



References

CDI

- JSR 299: Contexts and Dependency Injection for Java EE, <http://jcp.org/en/jsr/detail?id=299>
- Weld Reference, <http://docs.jboss.org/weld/reference/1.0.0/en-US/html/>

Implementations

- Weld, the JBoss reference implementation for JSR 299, <http://seamframework.org/Weld>
- CanDI, the JSR 299 implementation for Caucho Resin, <http://caucho.com/projects/candi/>



References

- OpenWebBeans, Apache implementation of JSR 299, <http://openwebbeans.apache.org>

Application Servers

- GlassFish, <http://glassfish.java.net>
- JBoss, <http://www.jboss.org/jbossas>
- Resin, <http://www.caucho.com/products/resin/>
- OpenEJB, <http://openejb.apache.org>



References

Portable Extensions

- Seam 3, <http://seamframework.org>
- Apache MyFaces CODI,
<http://myfaces.apache.org/extensions/cdi/>
- ZK Framework,
http://books.zkoss.org/wiki/Small_Talks/2010/March/Getting_started_with_ZK_CDI

Tools

- Arquillian, <http://www.jboss.org/arquillian>



References

- **Seam Forge,**
<http://seamframework.org/Documentation/SeamForge>
- **JBoss Tools,**
<http://community.jboss.org/en/tools/blog/2011/01/31/overview-of-context-dependency-injection-tools>



Thanks!

Email: reza@caucho.com

Twitter: cauchoresin

Facebook:

[http://www.facebook.com/pages/Caucho-Technology/
8671323790](http://www.facebook.com/pages/Caucho-Technology/8671323790)

Presentation materials: <http://www.caucho.com/articles/>