Web Beans: Java

##################Java###

Gavin King
JSR-299#####
###########

Pete Muir
Web Beans (JSR-299####) ###
###########

David Allen
######: Nicola Benaglia, Francesco Milesi
######: Gladys Guerrero
###########
####: Eun-Ju Ki,
###########
######: Terry Chuang
###########
######: Sean Wu
Kava##

æ³¨é##

JSR-299æ##è¿#å°#å##å-#ä»#"Web Beans"æ#¹ä¸º"Javaä¸#ä¸#æ##å##ä¾#èµ#æ³¨å#¥"ã##å##è##æ##å##ä»#ç#¶ä½¿ç#¨"Web Beans"å##ç§°ï¼#å##è##å®#ç#°ä»#ç#¶ä½¿ç#¨"Web Beans RI"å##ç§°ã##å#¶ä»#ç##æ##æ¡£ï¼#å##å®¢ï¼#è®ºå##å¸#å-#ç-#å#¯è#½ä½¿ç#¨æ#°ç##å#½å##ï¼#å##æ#¬æ299å##è##å®#ç#°ç##å##å-#- "Web Beans"ã##

ç##æ##è¿#ç##ä¸#ä¾#å##è#½ç¼ºå¤±äº#(ä¾#å¦#ç##äº§è##å##ï¼#å®#ç#°(realization)ï¼#å¼#æ-¥äº#ä»¶ï¼#Java EEèµ#æº#ç##XMLæ# å°#)

I. ä½¿ç#¨å#·å¤#ä¸#ä¸#æ##ç##å¯¹è±¡

The Web Beans (JSR-299)###Java EE##########################Web Beans######JavaBeans###JavaBeans###Java##########################Java EE########Web Beans######

- ###################### ##########
- ##############,
- ## #### ##########
- ################################*(decorator)*#######################

##

- ##############
- ##############
- ########
- #############
- ############
- ############################
- ###################################
- ##################

##Web Bean#############Web Bean####################Web Bean#######################Web Bean##################Web Bean##

############################

- ######################
- ####################
- ################

######Web Beans##################Web Beans#################################XML, ##########################Web Bean##Java####################Web Beans############################

Web Beans################Java EE##########

- ###JavaBeans,
- ###EJB, #
- ###Servlet#

Web Beans##################Java EE#####################Web Beans#######Web Beans############Web Bean###

Web Beans#####Seam, Guice#Spring########Java#########Web Beans###################Seam##Spring############XML, #Guice####Web#########

######Web Beans###JCP###########Java EE##Web Beans###############EJB#Java SE####

Web Beans##

####################Web Bean####################Web Beans##Web Bean#################Web Bean##

1.1. #####Web Bean

####################Java#######Web Bean#######JavaBean, ####EJB3###Bean####Web Bean############JavaBean#EJB####Web Beans################Web Beans#XML##########Web Bean##########Web Bean##############################

############Java##

```
public class SentenceParser {
   public List<String
> parse(String text) { ... }
}
```

################Bean#####Bean################################

```
@Stateless
public class SentenceTranslator implements Translator {
   public String translate(String sentence) { ... }
}
```

`Translator######`

```
@Local
public interface Translator {
   public String translate(String sentence);
}
```

#########Java####################Web Bean#######

```
public class TextTranslator {

   private SentenceParser sentenceParser;
```

```
    private Translator sentenceTranslator;

    @Initializer
    TextTranslator(SentenceParser sentenceParser, Translator sentenceTranslator) {
        this.sentenceParser = sentenceParser;
        this.sentenceTranslator = sentenceTranslator;
    }

    public String translate(String text) {
        StringBuilder sb = new StringBuilder();
        for (String sentence: sentenceParser.parse(text)) {
            sb.append(sentenceTranslator.translate(sentence));
        }
        return sb.toString();
    }

}
```

#########Web Bean#Servlet##EJB######### `TextTranslator###`:

```
@Initializer
public setTextTranslator(TextTranslator textTranslator) {
    this.textTranslator = textTranslator;
}
```

############Web Bean##############

```
TextTranslator tt = manager.getInstanceByType(TextTranslator.class);
```

#######`TextTranslator`##################Web Bean##########################Web Bean################`@Initializer`#####

######### `@Initializer`########## `@Initializer`#######Web Bean##############Web Bean#########Web Bean#################Web Beam#########Web Bean################

##########Web Bean############Web Bean#####################`Translator`#####`SentenceTranslator` EJB######Web Bean#########`UnsatisfiedDependencyException`#########`Translator`###Web Bean#########`AmbiguousDependencyException`###

1.2. ###Web Bean#

Web Bean######

##Web Bean#################Web Bean###Java#########################Web Bean###########Web Bean#######Web Bean###########Web Bean### ########Web Bean#########Web Bean########

########"###"##########Web Beans####################Bean####################Bean###############Servlet####Bean######Web Bean#########Web Bean#################################Web Bean######

###Web Bean#####################Bean#####################################Web Bean#######

- ##Web Bean##########
- #######Web Bean##########

####Web Bean####################Web Bean################################Web Bean###################################Web Bean################################Web Bean########

######################Web Bean########Web Bean##########################

##########################Web Bean##########################Web Bean#########################Web Bean##########Web Bean###################Web Bean#########

- ################
- ##############

###################################Web Bean####Web Bean########Web Bean#######Web Bean###################Web Bean########*# 4.2 # "####"*#####

###########Web Bean######Web Bean#######Servlet######Bean#151;#######, ###################Web Bean####

######################

##Web Bean###

- ######API##
- ############
- ####
- ######
- ######Web Bean###

- #########
- ##Web Bean##

###########Web Bean###########

1.2.1. API############

Web Bean############Web Bean#################"##"############Web Bean##########

- ##API####
- ######

##API#################Web Bean###EJB##Bean#API### `@Local` ####Bean################################API################

############################`@BindingType`################API## `PaymentProcessor`#####`@CreditCard`#

```
@CreditCard PaymentProcessor paymentProcessor
```

################################`@Current`#

########Web Bean############Web Bean####API#################Web Bean###

###Web Bean########`@CreditCard`######API##`PaymentProcessor`######Web Bean#################

```
@CreditCard
public class CreditCardPaymentProcessor
   implements PaymentProcessor { ... }
```

####Web Bean################################`@Current`#

Web Bean##############################Web Bean############### *# 4 #* ##########

1.2.2.

######################Web Bean################################`@Mock`#`@Staging`##`@AustralianTaxLaw`##############Web Bean################################Web Bean#########

##Web Bean##########`@Production`################################Web Bean#######`@Production`#

###############`SentenceTranslator` Web Bean#####"####":

```
@Mock
public class MockSentenceTranslator implements Translator {
   public String translate(String sentence) {
      return "Lorem ipsum dolor sit amet";
   }
}
```

#################`@Mock`######## `MockSentenceTranslator`######`@Mock`###Web Bean#

#####*# 4.2 # "####"*################

1.2.3.

#######Web Bean############Web Bean##Web Bean############################

#######Web####### #####Web Bean#

```
@SessionScoped
public class ShoppingCart { ... }
```

#######Web Bean###############################

######Web Bean##########################Web Bean##

#####*# 5 # ######*########

1.2.4. Web Bean###########

##Web Bean##############Web Bean################Web Bean############

```
@SessionScoped @Named("cart")
public class ShoppingCart { ... }
```

############JSF##JSP#######Web Bean#

```
<h:dataTable value="#{cart.lineItems}" var="item">
   ....
```

```
</h:dataTable
>
```

#######Web Bean#####Web Bean########

```
@SessionScoped @Named
public class ShoppingCart { ... }
```

#######Web Bean######`shoppingCart`###############

1.2.5.

Web Bean##EJB3#########Web Bean#######POJO##########Web Bean################EJB Bean###Web Bean##

####`@Interceptors` ###########

```
@SessionScoped
@Interceptors(TransactionInterceptor.class)
public class ShoppingCart { ... }
```

#####################################

```
@SessionScoped @Transactional
public class ShoppingCart { ... }
```

7 # ###*## 8 #* ###########Web Bean########

1.3. ##########Web Bean#

#######JavaBean, EJB###Java######Web Bean###################Web Bean#

1.3.1. ###Web Bean

Web Bean##########Java##########Web Bean, ######

- #####EE#############EJB###Servlet####JPA###
- ###############
- ##############

- ##################@Initializer###

########JavaBean#########Web Bean#

#####Web Bean#################Web Bean###API#############API###

1.3.2. ###Web Bean

######EJB3#####Bean####Bean#####Web Bean#####Bean##Web Beans#################151;###########Web Bean###############

#####Web Bean###################################Web Bean#API###################Web Bean#API#####EJB Bean###Bean#########Bean############API###

######Bean###################@Destructor########Web Bean############Bean############################Web Bean######

```
@Stateful @SessionScoped
public class ShoppingCart {

   ...

   @Remove
   public void destroy() {}

}
```

###########Web Bean#####Web Bean#######EJB#################

- ##############
- ####
- ##########Bean#########Bean#####
- ####Web######
- ##########

##################Web Bean##################Web Bean#####

##Web Bean################Web Bean############EJB3.1########################Web Bean###EJB#

####################Web Bean################################EJB# @Stateless/@Stateful/@Singleton###############

##

###Web Bean###########EJB#################`@Stateless`, `@Stateful` ## `@Singleton`#

1.3.3.

########Web Bean###############Web Bean##############Web Bean##Web Bean#########

```
@ApplicationScoped
public class Generator {

   private Random random = new Random( System.currentTimeMillis() );

   @Produces @Random int next() {
      return random.nextInt(100);
   }

}
```

##################################Web Bean###

```
@Random int randomNumber
```

#####################################API#######################API###

####################

```
@Produces @RequestScoped Connection connect(User user) {
   return createConnection( user.getId(), user.getPassword() );
}
```

####################

```
void close(@Disposes Connection connection) {
   connection.close();
}
```

###################Web Bean######

#####*# 6 #* #################

1.3.4. JMS##

#####JMS##############Web Bean#############################JMS########Web Bean##########################*# 13.4 # “JMS##”*#####JMS###

JSF Web####

##########################JSF############/##############Web Bean####################

```
@Named @RequestScoped
public class Credentials {

    private String username;
    private String password;

    public String getUsername() { return username; }
    public void setUsername(String username) { this.username = username; }

    public String getPassword() { return password; }
    public void setPassword(String password) { this.password = password; }

}
```

##Web Bean####JSF###login###

```
<h:form>
    <h:panelGrid columns="2" rendered="#{!login.loggedIn}">
        <h:outputLabel for="username"
>Username:</h:outputLabel>
        <h:inputText id="username" value="#{credentials.username}"/>
        <h:outputLabel for="password"
>Password:</h:outputLabel>
        <h:inputText id="password" value="#{credentials.password}"/>
    </h:panelGrid>
    <h:commandButton value="Login" action="#{login.login}" rendered="#{!login.loggedIn}"/>
    <h:commandButton value="Logout" acion="#{login.logout}" rendered="#{login.loggedIn}"/>
</h:form
>
```

##############Web Bean##########Web Bean################User#######Web Bean#

```
@SessionScoped @Named
public class Login {
```

```
    @Current Credentials credentials;
    @PersistenceContext EntityManager userDatabase;

    private User user;

    public void login() {

        List<User
> results = userDatabase.createQuery(
            "select u from User u where u.username=:username and u.password=:password")
            .setParameter("username", credentials.getUsername())
            .setParameter("password", credentials.getPassword())
            .getResultList();

        if ( !results.isEmpty() ) {
            user = results.get(0);
        }

    }

    public void logout() {
        user = null;
    }

    public boolean isLoggedIn() {
       return user!=null;
    }

    @Produces @LoggedIn User getCurrentUser() {
        return user;
    }

}
```

`@LoggedIn`########

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD})
@BindingType
public @interface LoggedIn {}
```

########Web Bean##############

```
public class DocumentEditor {

    @Current Document document;
    @LoggedIn User currentUser;
    @PersistenceContext EntityManager docDatabase;

    public void save() {
        document.setCreatedBy(currentUser);
        docDatabase.persist(document);
    }

}
```

##############Web Bean########################Web Bean######

##Web Beans#####

Web Bean#####*Seam##* [http://seamframework.org/WebBeans]#######*the downloads page* [http://seamframework.org/Download]###########

Web Beans RI########`webbeans-numberguess` (#########Bean#WAR####)#`webbeans-translator` (######Bean#EAR####)############

- ###Web Bean#######
- JBoss AS 5.0.0.GA, #
- Apache Tomcat 6.0.X##
- Ant 1.7.0.

3.1. ##JBoss AS 5

###Web Beans#########JBoss AS 5#######*jboss.org* [http://www.jboss.org/jbossas/downloads/]##JBoss AS 5.0.0.GA, ########

```
$ cd /Applications
$ unzip ~/jboss-5.0.0.GA.zip
```

###*seamframework.org* [http://seamframework.org/WebBeans]##Web Beans###############

```
$ cd ~/
$ unzip ~/webbeans-1.0.0.ALPHA1.zip
```

#########Web Beans JBoss######`jboss-as/build.properties`###`jboss.home`######

```
jboss.home=/Applications/jboss-5.0.0.GA
```

############Ant 1.7.0###`ANT_HOME`########

```
$ unzip apache-ant-1.7.0.zip
$ export ANT_HOME=~/apache-ant-1.7.0
```

##################Maven#####Web Beans#EJB3#

```
$ cd webbeans-1.0.0.ALPHA1/jboss-as
$ ant update
```

##################

##

###############

- `ant restart` - #exploded######
- `ant explode` - ###########exploded########
- `ant deploy` - ###jar########
- `ant undeploy` - ##########
- `ant clean` - ####

#####(numberguess)###

```
$ cd examples/numberguess
ant deploy
```

JBoss AS:

```
jboss.home=/Applications/jboss-5.0.0.GA
```

##

#####Windows########`run.bat` ###

################*http://localhost:8080/webbeans-numberguess*#

Web Bean####################################WAR#############Beans#########EAR#########EJB######B

```
$ cd examples/traslator
```

```
ant deploy
```

##########http://localhost:8080/webbeans-translator#

3.2. ##Apache Tomcat 6.0

###seamframework.org [http://seamframework.org/WebBeans]##Web Beans###############

```
$ cd /Applications
$ unzip ~/jboss-5.0.0.GA.zip
```

###seamframework.org [http://seamframework.org/WebBeans]##Web Beans###############

```
$ cd ~/
$ unzip ~/webbeans-1.0.0.ALPHA1.zip
```

#########Web Beans JBoss######`jboss-as/build.properties`###`jboss.home`######

```
jboss.home=/Applications/jboss-5.0.0.GA
```

##

###############

- `ant restart` - #exploded######
- `ant explode` - ###########exploded########
- `ant deploy` - ###jar########
- `ant undeploy` - ##########
- `ant clean` - ####

######(numberguess)###

```
$ cd examples/traslator
ant deploy
```

##Tomcat:

```
jboss.home=/Applications/jboss-5.0.0.GA
```

##

#####Windows########`run.bat` ###

################*http://localhost:8080/webbeans-numberguess*#

3.3. ##Glassfish

##

3.4.

###################1#100###################################

######Web Beans######Facelete JSF##########WAR#############

################`WEB-INF/`###########`WebContent`########faces-config.xml#####JSF##Faceletes#

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="1.2"
              xmlns="http://java.sun.com/xml/ns/javaee"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/
javaee/web-facesconfig_1_2.xsd">

   <application>
      <view-handler
>com.sun.facelets.FaceletViewHandler</view-handler>
   </application>

</faces-config
>
```

######`web-beans.xml`############Web Beans###

####### `web.xml`#

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/
web-app_2_5.xsd">

  <display-name
>Web Beans Numbergues example</display-name>

  <!-- JSF -->
                                                    (1)
  <servlet>
    <servlet-name
>Faces Servlet</servlet-name>
    <servlet-class
>javax.faces.webapp.FacesServlet</servlet-class>
                                                    (2)
    <load-on-startup
>1</load-on-startup>
  </servlet>

  <servlet-mapping>
                                                    (3)
    <servlet-name
>Faces Servlet</servlet-name>
    <url-pattern
>*.jsf</url-pattern>
  </servlet-mapping>
                                                    (4)
  <context-param>
    <param-name
>javax.faces.DEFAULT_SUFFIX</param-name>
                                                    (5)
    <param-value
>.xhtml</param-value>
  </context-param>

  <session-config>
    <session-timeout
>10</session-timeout>
  </session-config>
```

```
</web-app
>
```

1. Enable and load the JSF servlet
2. Configure requests to `.jsf` pages to be handled by JSF
3. Tell JSF that we will be giving our source files (facelets) an extension of `.xhtml`
4. Configure a session timeout of 10 minutes

##

Whilst this demo is a JSF demo, you can use Web Beans with any Servlet based web framework.

Let's take a look at the Facelet view:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:ui="http://java.sun.com/jsf/facelets"
   xmlns:h="http://java.sun.com/jsf/html"
   xmlns:f="http://java.sun.com/jsf/core"
   xmlns:s="http://jboss.com/products/seam/taglib">


 <ui:composition template="template.xhtml">                                    1
   <ui:define name="content">
     <h1
>Guess a number...</h1>

     <h:form id="NumberGuessMain">                                             2
       <div style="color: red">
         <h:messages id="messages" globalOnly="false"/>
           <h:outputText id="Higher" value="Higher!" rendered="#{game.number gt game.guess
 and game.guess ne 0}"/>
         <h:outputText id="Lower" value="Lower!" rendered="#{game.number lt game.guess and
 game.guess ne 0}"/>
       </div>


       <div>                                                                   3
         I'm thinking of a number between #{game.smallest} and #{game.biggest}.
         You have #{game.remainingGuesses} guesses.
```

```
        </div>

        <div>
          Your guess:                                                    4
          <h:inputText id="inputGuess"
                       value="#{game.guess}"
                       required="true"
                       size="3"

                       disabled="#{game.number eq game.guess}">          5
            <f:validateLongRange maximum="#{game.biggest}"
                                 minimum="#{game.smallest}"/>

          </h:inputText>                                                 6
          <h:commandButton id="GuessButton"
                           value="Guess"
                           action="#{game.check}"
                           disabled="#{game.number eq game.guess}"/>
        </div>
        <div>
                      <h:commandButton id="RestartButton" value="Reset" action="#{game.reset}"
immediate="true" />
        </div>
      </h:form>
    </ui:define>
  </ui:composition>
</html
>
```

1. Facelets is a templating language for JSF, here we are wrapping our page in a template which defines the header.
2. There are a number of messages which can be sent to the user, "Higher!", "Lower!" and "Correct!"
3. As the user guesses, the range of numbers they can guess gets smaller - this sentance changes to make sure they know what range to guess in.
4. This input field is bound to a Web Bean, using the value expression.
5. A range validator is used to make sure the user doesn't accidentally input a number outside of the range in which they can guess - if the validator wasn't here, the user might use up a guess on an out of range number.
6. And, of course, there must be a way for the user to send their guess to the server. Here we bind to an action method on the Web Bean.

######4#####################@Random################

```
@Target( { TYPE, METHOD, PARAMETER, FIELD })
@Retention(RUNTIME)
@Documented
@BindingType
public @interface Random {}
```

#####@MaxNumber###############

```
@Target( { TYPE, METHOD, PARAMETER, FIELD })
@Retention(RUNTIME)
@Documented
@BindingType
public @interface MaxNumber {}
```

Generator########(producer)################################

```
@ApplicationScoped
public class Generator {

  private java.util.Random random = new java.util.Random( System.currentTimeMillis() );

  private int maxNumber = 100;

  java.util.Random getRandom()
  {
    return random;
  }

  @Produces @Random int next() {
    return getRandom().nextInt(maxNumber);
  }

  @Produces @MaxNumber int getMaxNumber()
  {
    return maxNumber;
  }

}
```

#####Generator##############################

#######Web Bean###### `Game` #

############
`@Named`############EL########JSF#####Bean##

```
package org.jboss.webbeans.examples.numberguess;


import javax.annotation.PostConstruct;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.webbeans.AnnotationLiteral;
import javax.webbeans.Current;
import javax.webbeans.Initializer;
import javax.webbeans.Named;
import javax.webbeans.SessionScoped;
import javax.webbeans.manager.Manager;

@Named
@SessionScoped
public class Game
{
   private int number;

   private int guess;
   private int smallest;
   private int biggest;
   private int remainingGuesses;

   @Current Manager manager;

   public Game()
   {
   }

   @Initializer
   Game(@MaxNumber int maxNumber)
   {
      this.biggest = maxNumber;
   }

   public int getNumber()
   {
      return number;
```

```
  }

  public int getGuess()
  {
    return guess;
  }

  public void setGuess(int guess)
  {
    this.guess = guess;
  }

  public int getSmallest()
  {
    return smallest;
  }

  public int getBiggest()
  {
    return biggest;
  }

  public int getRemainingGuesses()
  {
    return remainingGuesses;
  }

  public String check()
  {
    if (guess
>number)
    {
      biggest = guess - 1;
    }
    if (guess<number)
    {
      smallest = guess + 1;
    }
    if (guess == number)
    {
      FacesContext.getCurrentInstance().addMessage(null, new FacesMessage("Correct!"));
    }
    remainingGuesses--;
    return null;
```

```
  }

  @PostConstruct
  public void reset()
  {
    this.smallest = 0;
    this.guess = 0;
    this.remainingGuesses = 10;
    this.number = manager.getInstanceByType(Integer.class, new AnnotationLiteral<Random
>(){});
  }

}
```

3.4.1.

#Tomcat###################WebBean######Web#########`WEB-INF/lib`####################jar# `webbeans-tomcat.jar`#######Tomcat###Web Bean#

##

##########JSF#EL, ########(`jsr250-api.jar`)########Java EE###########

########`web.xml`######Tomcat#Servlet########Web Bean###############

```
<listener>
  <listener-class
>org.jboss.webbeans.environment.tomcat.Listener</listener-class>
</listener
>
```

3.5.

######################

########EAR#####EJBs###Beans#################

##

EJB3.1#Java EE 6####WAR####EJBs, ############

##########EAR#######webbeans-translator-ear####Maven########application.xml#jboss-app.xml###

```
<plugin>
  <groupId
>org.apache.maven.plugins</groupId>
  <artifactId
>maven-ear-plugin</artifactId>
  <configuration>
    <modules>
      <webModule>
        <groupId
>org.jboss.webbeans.examples.translator</groupId>
        <artifactId
>webbeans-translator-war</artifactId>
        <contextRoot
>/webbeans-translator</contextRoot>
      </webModule>
    </modules>
    <jboss>
      <loader-repository
>webbeans.jboss.org:loader=webbeans-translator</loader-repository>
    </jboss>
  </configuration>
</plugin
>
```

###########-####################URL(*http://localhost:8080/webbeans-translator*)#######JBoss AS############

##

######Maven#############META-INF/jboss-app.xml#

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://java.sun.com/xml/ns/javaee"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://
java.sun.com/xml/ns/javaee/application_5.xsd"
             version="5">
  <display-name
>webbeans-translator-ear</display-name>
```

```
  <description
>Ear Example for the reference implementation of JSR 299: Web Beans</
description>

  <module>
    <web>
      <web-uri
>webbeans-translator.war</web-uri>
      <context-root
>/webbeans-translator</context-root>
    </web>
  </module>
  <module>
    <ejb
>webbeans-translator.jar</ejb>
  </module>
</application
>
```

########WAR###############`faces-config.xml`###Facelets######`WebContent/WEB-INF`##`web.xml`###JSF##Web Beans####Servlet#####

##############facelet###################################

```
<h:form id="NumberGuessMain">

  <table>
    <tr align="center" style="font-weight: bold" >
      <td>
        Your text
      </td>
      <td>
        Translation
      </td>
    </tr>
    <tr>
      <td>
      <h:inputTextarea id="text" value="#{translator.text}" required="true" rows="5" cols="80" />
      </td>
      <td>
        <h:outputText value="#{translator.translatedText}" />
      </td>
    </tr>
```

```
  </table>
  <div>
    <h:commandButton id="button" value="Translate" action="#{translator.translate}"/>
  </div>

</h:form
>
```

####################################

########EJB###webbeans-translator-ejb##src/main/resources/META-INF###############web-beans.xml############Web Beans####ejb-jar.xml#Web Beans####EJB##############ejb-jar.xml###########Web Beans#EJB##########

################################Beans, SentanceParser#TextTranslator#######Beans#TanslatorControllerBean#SentenceTranslator#######Web Beans#####################

SentanceParser#TextTranslator######Beans#TextTranslator#########

```
public class TextTranslator {
  private SentenceParser sentenceParser;
  private Translator sentenceTranslator;

  @Initializer
  TextTranslator(SentenceParser sentenceParser, Translator sentenceTranslator)
  {
    this.sentenceParser = sentenceParser;
    this.sentenceTranslator = sentenceTranslator;
```

TextTranslator######Bean#####################-###################################

############UI###Bean##############

```
@Stateful
@RequestScoped
@Named("translator")
public class TranslatorControllerBean implements TranslatorController
{

  @Current TextTranslator translator;
```

##Bean##########getter#setter###

##Bean######Bean########remove###

```
@Remove
public void remove()
{

}
```

Web Beans######bean#######remove##################

Web Beans###*http://www.seamframework.org/WebBeans/Development*#

#############-bug#########################

####

Web Beans###############

########

```
public class Checkout {

   private final ShoppingCart cart;

   @Initializer
   public Checkout(ShoppingCart cart) {
      this.cart = cart;
   }

}
```

#######

```
public class Checkout {

   private ShoppingCart cart;

   @Initializer
   void setShoppingCart(ShoppingCart cart) {
      this.cart = cart;
   }

}
```

########

```
public class Checkout {

   private @Current ShoppingCart cart;

}
```

#Web Bean######################

- ###Web Bean#####Web Bean########Web Bean####
- ####Web Bean########Web bean#########
- ###Web Bean#######Web Bean#######
- ### ### `@PostConstruct` ############

EJB Beans##############EJB##EJB############Web Bean######

#########`@Current`## ### ####################################

#############

```
@Produces Checkout createCheckout(ShoppingCart cart) {
   return new Checkout(cart);
}
```

############## # 9 # ##########disposal#######destructor#############

Web Beans######### ########################Web Bean##Web Bean###Web Bean############################## `UnsatisfiedDependencyException` #### `AmbiguousDependencyException` ###

############Web Beans#####API######

- ###### #### ###########
- ############## ##### ###################################
- ######### #####API###########API################

#####Web Bean###########Web Bean#

4.1.

##########API#####Web Bean########################Web Bean########### `PaymentProcessor` ####

```
@PayByCheque
public class ChequePaymentProcessor implements PaymentProcessor {
   public void process(Payment payment) { ... }
}
```

```
@PayByCreditCard
```

```
public class CreditCardPaymentProcessor implements PaymentProcessor {
    public void process(Payment payment) { ... }
}
```

`@PayByCheque` # `@PayByCreditCard`

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
@BindingType
public @interface PayByCheque {}
```

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
@BindingType
public @interface PayByCreditCard {}
```

####Web Bean#################Web Bean######

######

```
@PayByCheque PaymentProcessor chequePaymentProcessor;
@PayByCreditCard PaymentProcessor creditCardPaymentProcessor;
```

##########

```
@Initializer
public void setPaymentProcessors(@PayByCheque PaymentProcessor
 chequePaymentProcessor,
                    @PayByCreditCard PaymentProcessor creditCardPaymentProcessor) {
  this.chequePaymentProcessor = chequePaymentProcessor;
  this.creditCardPaymentProcessor = creditCardPaymentProcessor;
}
```

##########

```
@Initializer
public Checkout(@PayByCheque PaymentProcessor chequePaymentProcessor,
            @PayByCreditCard PaymentProcessor creditCardPaymentProcessor) {
```

```
  this.chequePaymentProcessor = chequePaymentProcessor;
  this.creditCardPaymentProcessor = creditCardPaymentProcessor;
}
```

4.1.1.

###########

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
@BindingType
public @interface PayBy {
   PaymentType value();
}
```

###############

```
@PayBy(CHEQUE) PaymentProcessor chequePaymentProcessor;
@PayBy(CREDIT_CARD) PaymentProcessor creditCardPaymentProcessor;
```

#####Web Bean########################## `@NonBinding` ###

4.1.2.

##################

```
@Asynchronous @PayByCheque PaymentProcessor paymentProcessor
```

###################Web Bean########

4.1.3.

################

```
@Produces
@Asynchronous @PayByCheque
PaymentProcessor createAsyncPaymentProcessor(@PayByCheque PaymentProcessor
 processor) {
   return new AsynchronousPaymentProcessor(processor);
```

```
}
```

4.1.4.

Web Beans##### `@Current` ##############################Web Bean########

################`@Current` #

- ################################
- ###Web Bean######Web Bean######################

4.2.

###Web Beans#### #### ###########Web Beans###Web Beans##################

############### `@Mock` ####################################Web Beans#

```
@Retention(RUNTIME)
 @Target({TYPE, METHOD})
 @DeploymentType
 public @interface Mock {}
```

################Web Beans######

```
public class ExternalPaymentProcessor {

   public void process(Payment p) {
      ...
   }

}
```

####Web Bean######################## `@Production` #####

##

```
@Mock
public class MockPaymentProcessor implements PaymentProcessor {

   @Override
   public void process(Payment p) {
```

```
        p.setSuccessful(true);
    }

}
```

##Web Bean#########################

4.2.1.

Web Beans############ `@Production` # `@Standard`##########################Web Beans########## `web-beans.xml` ############################

################################### `@Mock` ######

```
<WebBeans>
    <Deploy>
        <Standard/>
        <Production/>
        <test:Mock/>
    </Deploy>
</WebBeans
>
```

###Web Bean################### `@Production`# `@Standard` # `@Mock` ###Web Beans#

`@Standard` ##Web Beans#######Web Bean##################Web Bean################

`@Production` ############Web Beans################

4.2.2.

###############Web Bean########### `ExternalPaymentProcessor` ## `MockPaymentProcessor` ########################

```
@Current PaymentProcessor paymentProcessor
```

#####Web Bean## `PaymentProcessor` ##Web Bean#

######################### ### ############ `web-beans.xml` ################## `@Mock` # `@Production` #################

############Web Bean###############API###############Web Beans###################Web Bean###################################Web Bean###### `MockPaymentProcessor` ###

########################
"###"###XML######Web Bean######XML######Web Bean###############################XML#######Web Beans####################Web Bean###########

4.2.3.

######################

- `@Mock` # `@Staging` ###########
- `@AustralianTaxLaw` #######Web Beans#######
- `@SeamFramework`, `@Guice` #####Web Bean###############
- `@Standard` ##Web Bean#######Web Bean#######

##############...

4.3.

############API###Web Bean#########################Web Bean###############Web Bean#

###########`UnsatisfiedDependencyException` ## `AmbiguousDependencyException`#

`UnsatisfiedDependencyException`############API###Web Bean############ # ##########API###########Web Bean######

#####
`AmbiguousDependencyException`#################API#############################Web Bean#################################Web Bean#########################
`AmbiguousDependencyException` #

##Web Bean#################

4.4.

###Web Bean#############Web bean######

#############Web Bean###########Web Bean###########Web Bean#################################Web Bean###

############Web Bean#########Web Bean###Web Bean##########Web Bean#######

#####Web Bean##### `@Dependent` ####Web Bean#########################Web Bean##### #### ##########Web Bean#####################################Web Bean#########################Web Bean#####Web Bean#

#######Java########Java#####Web Bean#########################Web Bean######## `UnproxyableDependencyException` ###

###Java#####Web Bean######

- ### `final` ###### `final` #####
- ################
- ########

`UnproxyableDependencyException` ####################################Web Bean### `@Dependent`

4.5. ##########Web Bean

############ `Manager` #####

```
@Current Manager manager;
```

`Manager` ###############Web Bean######

```
PaymentProcessor p = manager.getInstanceByType(PaymentProcessor.class);
```

################ `AnnotationLiteral` #############Java############

```
PaymentProcessor p = manager.getInstanceByType(PaymentProcessor.class,
                                               new AnnotationLiteral<CreditCard
>(){});
```

###################### `AnnotationLiteral` ###### ################

```
abstract class CreditCardBinding
   extends AnnotationLiteral<CreditCard
>
   implements CreditCard {}
```

```
PaymentProcessor p = manager.getInstanceByType(PaymentProcessor.class,
                                    new CreditCardBinding() {
                                       public void value() { return paymentType; }
                                    } );
```

4.6. ##########@Resource# @EJB # @PersistenceContext

###Web Beans####EJB###############@PostConstruct, @PreDestroy, @PrePassivate # @PostActivate#

###Web Bean### @PostConstruct # @PreDestroy ###

#######Web Bean#####f @Resource, @EJB # @PersistenceContext #####Java EE###EJB#JPA##########Web Bean##### @PersistenceContext(type=EXTENDED) #

@PostConstruct #################

4.7. InjectionPoint

########## @Dependent ###Web Bean # ###################################

- Logger###############
- ##HTTP############################
- ########################

@Dependent ###Web Bean###### InjectionPoint

#########################

```
Logger log = Logger.getLogger(MyClass.class.getName());
```

###############JDK# Logger ###############

```
class LogFactory {

  @Produces Logger createLogger(InjectionPoint injectionPoint) {
    return Logger.getLogger(injectionPoint.getMember().getDeclaringClass().getName());
  }

}
```

#########

```
@Current Logger log;
```

##################HTTP###############

```
@BindingType
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
public @interface HttpParam {
   @NonBinding public String value();
}
```

#################

```
@HttpParam("username") String username;
@HttpParam("password") String password;
```

################

```
class HttpParams

   @Produces @HttpParam("")
   String getParamValue(ServletRequest request, InjectionPoint ip) {
      return request.getParameter(ip.getAnnotation(HttpParam.class).value());
   }

}
```

(## `HttpParam` ##### `value()` ##Web Bean########### `@NonBinding.` ##)

Web Bean############ `InjectionPoint` ###Web Bean#

```
public interface InjectionPoint {
   public Object getInstance();
   public Bean<?> getBean();
   public Member getMember():
   public <T extends Annotation
> T getAnnotation(Class<T
> annotation);
   public Set<T extends Annotation
```

```
> getAnnotations();
}
```

######

############## ##########Web Bean######Web Bean########################Web Bean#####Web Beans##########

- ####Web Bean##########
- ####Web Bean#######
- ###########Web Bean#####

###############Web Bean#`CurrentUser`#######`HttpSession`###########Web Bean#######`CurrentUser`###############`CurrentUser`##################

5.1.

Web Bean##

```
@Retention(RUNTIME)
@Target({TYPE, METHOD})
@ScopeType
public @interface ClusterScoped {}
```

##`Context`##

#####Web Bean###############Web Bean####

```
@ClusterScoped
public class SecondLevelCache { ... }
```

##########Web Bean######

5.2.

Web Beans##########

- `@RequestScoped`
- `@SessionScoped`
- `@ApplicationScoped`
- `@ConversationScoped`

####Web Beans#Web###

- ##Servlet######################
- ##JSF################

##################

- #EJB#########
- #EJB#####
- ##########Bean####
- #Web Service#####

##########Web Bean#####################Web Bean############`ContextNotActiveException`###

#################Java EE######################################

5.3.

Web Beans###(Conversation)###############(Session)###

- ##############
- ###JSF###########Web###########

###

##########JSF###

5.3.1.

Web Beans########Web Bean######JSF#############Web Bean##########

```
@Current Conversation conversation;
```

############################## `begin()`##############################`end()`###

###############Web Bean##########

```
@ConversationScoped @Stateful
```

```
public class OrderBuilder {

   private Order order;
   private @Current Conversation conversation;
   private @PersistenceContext(type=EXTENDED) EntityManager em;

   @Produces public Order getOrder() {
      return order;
   }

   public Order createOrder() {
      order = new Order();
      conversation.begin();
      return order;
   }

   public void addLineItem(Product product, int quantity) {
      order.add( new LineItem(product, quantity) );
   }

   public void saveOrder(Order order) {
      em.persist(order);
      conversation.end();
   }

   @Remove
   public void destroy() {}

}
```

Web Bean####`Conversation`##################Web Bean###############

5.3.2.

########JSF faces########JSF########faces##############################

########faces##################################Web Beans################`cid`#############`Conversation`########Web Beans###`conversation`#

###############

```
<a href="/addProduct.jsp?cid=#{conversation.id}"
>Add Product</a
```

```
>
```

Web Bean##POST-then-redirect############## "##"##########Web Bean#########URL##########

5.3.3.

Web Bean###Web Bean#################################Web Bean##########################

`Conversation`####################Web Bean###########

```
conversation.setTimeout(timeoutInMillis);
```

5.4.

##########Web Beans##############################Web Bean######

#####Web Bean#######`@Dependent`:

```
public class Calculator { ... }
```

################Web Bean########Web Bean################Web Bean######Web Beans############Web Beans################Web Bean#########

###Web Bean######################

Web Bean###########Java###EJB Bean#############EJB Bean#############Web Bean#####

5.4.1. `@New`##

###`@New`####################Web Bean#################

```
@New Calculator calculator;
```

##Web Bean##########`@Dependent`######`@New`#API###`Calculator`####`Calculator`#######`@Standard`#

###`Calculator`##########################

```
@ConversationScoped
```

```
public class Calculator { ... }
```

#################### Calculator###

```
public class PaymentCalc {

    @Current Calculator calculator;
    @New Calculator newCalculator;

}
```

calculator#########Calculator#####newCalculator######Calculator####################PaymentCa

############################

#####

###############Web Bean#####################################Web Beans######Web Beans################# *12 # ##XML##Web Bean*########

#####

> A Web Beans producer method acts as a source of objects to be injected, where:
>
> - the objects to be injected are not required to be instances of Web Beans,
> - the concrete type of the objects to be injected may vary at runtime or
> - the objects require some custom initialization that is not performed by the Web Bean constructor

For example, producer methods let us:

- expose a JPA entity as a Web Bean,
- expose any JDK class as a Web Bean,
- define multiple Web Beans, with different scopes or initialization, for the same implementation class, or
- vary the implementation of an API type at runtime.

In particular, producer methods let us use runtime polymorphism with Web Beans. As we've seen, deployment types are a powerful solution to the problem of deployment-time polymorphism. But once the system is deployed, the Web Bean implementation is fixed. A producer method has no such limitation:

```
@SessionScoped
public class Preferences {

   private PaymentStrategyType paymentStrategy;

   ...

   @Produces @Preferred
   public PaymentStrategy getPaymentStrategy() {
      switch (paymentStrategy) {
         case CREDIT_CARD: return new CreditCardPaymentStrategy();
         case CHEQUE: return new ChequePaymentStrategy();
         case PAYPAL: return new PayPalPaymentStrategy();
```

```
            default: return null;
        }
    }

}
```

Consider an injection point:

```
@Preferred PaymentStrategy paymentStrat;
```

This injection point has the same type and binding annotations as the producer method, so it resolves to the producer method using the usual Web Beans injection rules. The producer method will be called by the Web Bean manager to obtain an instance to service this injection point.

6.1.

############ `@Dependent`###Web Bean###`PaymentStrategy`###

###################### `@SessionScoped` ###

```
@Produces @Preferred @SessionScoped
public PaymentStrategy getPaymentStrategy() {
    ...
}
```

################### `PaymentStrategy` ######################################

6.2.

############### `CreditCardPaymentStrategy` #####Java# `new` ##

################################Web Bean###

```
@Produces @Preferred @SessionScoped
public PaymentStrategy getPaymentStrategy(CreditCardPaymentStrategy ccps,
                                          ChequePaymentStrategy cps,
                                          PayPalPaymentStrategy ppps) {
    switch (paymentStrategy) {
        case CREDIT_CARD: return ccps;
        case CHEQUE: return cps;
```

```
      case PAYPAL: return ppps;
      default: return null;
   }
}
```

CreditCardPaymentStrategy ########Web Bean##########################"##"############Bug#################Web Bean################## "##"######Web Bean##########################Web bean##############

###############Bug####### CreditCardPaymentStrategy######################Web Bean######################### @Dependent ## @RequestScoped#

################ @New #####

6.3. ########## @New

#############

```
@Produces @Preferred @SessionScoped
public PaymentStrategy getPaymentStrategy(@New CreditCardPaymentStrategy ccps,
                                          @New ChequePaymentStrategy cps,
                                          @New PayPalPaymentStrategy ppps) {
   switch (paymentStrategy) {
      case CREDIT_CARD: return ccps;
      case CHEQUE: return cps;
      case PAYPAL: return ppps;
      default: return null;
   }
}
```

######## CreditCardPaymentStrategy ##
Preferences ######

II. å¼#å##æ#¾è#¦å##ç##ä»£ç

Web Bean############################

- ###############
- ##################
- ##############Web Bean########

##############################API###

###Web Beans###############

Web Bean####################

- ######################
- ##################
- ######################

############

###

Web Beans###EJB3.0#########################

- ##Web Bean#################Bean#
- Web Beans######################Web Bean##

EJB###########

- ########
- ############

#Web Bean######Web Bean#######

```
public class TransactionInterceptor {
    @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }
}
```

####################

```
public class DependencyInjectionInterceptor {
    @PostConstruct public void injectDependencies(InvocationContext ctx) { ... }
}
```

################################

7.1.

############Web Beans######################## ####### ######Web Beans######

```
@InterceptorBindingType
@Target({METHOD, TYPE})
@Retention(RUNTIME)
public @interface Transactional {}
```

################ `ShoppingCart` ##########

```
@Transactional
public class ShoppingCart { ... }
```

##############################

```
public class ShoppingCart {
   @Transactional public void checkout() { ... }
}
```

7.2.

##EJB###### `@Interceptor` # `@Transactional####`

```
@Transactional @Interceptor
public class TransactionInterceptor {
   @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }
}
```

###Web Beans########Web Beans##########################

```
@ApplicationScoped @Transactional @Interceptor
public class TransactionInterceptor {

   @Resource Transaction transaction;

   @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }

}
```

####################

7.3.

######## `web-beans.xml` ##### ## #######

```
<Interceptors>
   <tx:TransactionInterceptor/>
</Interceptors
>
```

###########

###XML##########

- ##
- ######################

########## `TransactionInterceptor` #############

```
<Interceptors>
   <sx:SecurityInterceptor/>
   <tx:TransactionInterceptor/>
</Interceptors
>
```

###################

7.4.

######## `@Transactional` #############

```
@InterceptorBindingType
@Target({METHOD, TYPE})
@Retention(RUNTIME)
public @interface Transactional {
   boolean requiresNew() default false;
}
```

Web Beans### `requiresNew` ############### `TransactionInterceptor` # `RequiresNewTransactionInterceptor` #######

```
@Transactional(requiresNew=true) @Interceptor
public class RequiresNewTransactionInterceptor {
   @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }
}
```

############ `RequiresNewTransactionInterceptor` #

```
@Transactional(requiresNew=true)
public class ShoppingCart { ... }
```

################################### `requiresNew` ################ `@NonBinding` ###

```
@InterceptorBindingType
@Target({METHOD, TYPE})
@Retention(RUNTIME)
public @interface Secure {
   @NonBinding String[] rolesAllowed() default {};
}
```

7.5.

############################Web Bean############## `TransactionInterceptor` # `SecurityInterceptor` ######Web Bean##

```
@Secure(rolesAllowed="admin") @Transactional
public class ShoppingCart { ... }
```

##

```
@Transactional @Secure @Interceptor
public class TransactionalSecureInterceptor { ... }
```

########################## `checkout()` ####

```
public class ShoppingCart {
   @Transactional @Secure public void checkout() { ... }
}
```

```
@Secure
public class ShoppingCart {
   @Transactional public void checkout() { ... }
}
```

```
@Transactionl
public class ShoppingCart {
   @Secure public void checkout() { ... }
}
```

```
@Transactional @Secure
public class ShoppingCart {
   public void checkout() { ... }
}
```

7.6.

Java##

```
public @interface Action extends Transactional, Secure { ... }
```

#######Web Beans######Java## ############Web Bean#################

```
@Transactional @Secure
@InterceptorBindingType
@Target(TYPE)
@Retention(RUNTIME)
public @interface Action { ... }
```

`@Action` ###Web Bean##### `TransactionInterceptor` # `SecurityInterceptor` ###### `TransactionalSecureInterceptor`

7.7. `@Interceptors`

#######Web Bean###EJB##### `@Interceptors` ######

```
@Interceptors({TransactionInterceptor.class, SecurityInterceptor.class})
public class ShoppingCart {
   public void checkout() { ... }
}
```

##############

- #################
- ####################
- ############ # ##################

##########Web Bean##########

###

##Java#####################################

########################Java###

################

```
public interface Account {
   public BigDecimal getBalance();
   public User getOwner();
   public void withdraw(BigDecimal amount);
   public void deposit(BigDecimal amount);
}
```

################Web Beans## `Account`
##

#########Web Bean################## `@Decorator` ###

```
@Decorator
public abstract class LargeTransactionDecorator
      implements Account {

   @Decorates Account account;

   @PersistenceContext EntityManager em;

   public void withdraw(BigDecimal amount) {
      account.withdraw(amount);
      if ( amount.compareTo(LARGE_AMOUNT)
>0 ) {
         em.persist( new LoggedWithdrawl(amount) );
      }
   }

   public void deposit(BigDecimal amount);
      account.deposit(amount);
      if ( amount.compareTo(LARGE_AMOUNT)
>0 ) {
         em.persist( new LoggedDeposit(amount) );
      }
   }
```

```
}
```

######Web Beans###

8.1.

######### #### ##########################Web Bean#############################

#################### `Account` ###Web Beans#

```
@Decorates Account account;
```

################################Web Beans#

```
@Decorates @Foreign Account account;
```

####################Web Bean##

- #####API############
- #################

######################### `InvocationContext.proceed()` #######

8.2.

`web-beans.xml`

```
<Decorators>
   <myapp:LargeTransactionDecorator/>
</Decorators
>
```

#############`<Interceptors>`##############

- ##################################
- ######################

########################

##

Web Beans#########Web Beans#################*(producers)*########Web Bean######## *###(observers)* ##############################

- ##############################
- #########"###"################
- ###################################

9.1.

#########Web Bean############`@Observes`######

```
public void onAnyDocumentEvent(@Observes Document document) { ... }
```

##"###"########Web Beans##

```
@BindingType
@Target({PARAMETER, FIELD})
@Retention(RUNTIME)
public @interface Updated { ... }
```

#########################

```
public void afterDocumentUpdate(@Observes @Updated Document document) { ... }
```

################## # ################ ## ##

########################Web Beans###########

```
public void afterDocumentUpdate(@Observes @Updated Document document, User user) { ... }
```

9.2.

############### ########

```
@Observable Event<Document
> documentEvent
```

`@Observable#########` `@Dependent` `#########@Standard` #Web Bean#####Web Bean######

######### `Event` ### `fire()` ############ #### #

```
documentEvent.fire(document);
```

############################Java##########################

- ##################
- ########

Web Bean##Web Bean###############`fire()` ############

###"###"####################### `fire()` ###

```
documentEvent.fire( document, new AnnotationLiteral<Updated
>(){} );
```

`AnnotationLiteral` ####################Java#######

######################

- ##################
- #####`fire()`####################

######################

```
@Observable @Updated Event<Document
> documentUpdatedEvent
```

####### `Event` ##

- ##################
- ##### `fire()` ###################

9.3.

######################## `Observer` ####### `observe()` ##################

```
documentEvent.observe( new Observer<Document
>() { public void notify(Document doc) { ... } } );
```

###################################### `observe()` ###

```
documentEvent.observe( new Observer<Document
>() { public void notify(Document doc) { ... } },
                                    new AnnotationLiteral<Updated
>(){} );
```

9.4.

################

```
@BindingType
@Target({PARAMETER, FIELD})
@Retention(RUNTIME)
public @interface Role {
   RoleType value();
}
```

################

```
public void adminLoggedIn(@Observes @Role(ADMIN) LoggedIn event) { ... }
```

##

```
@Observable @Role(ADMIN) Event<LoggedIn
> LoggedInEvent;}}
```

################################## `AnnotationLiteral` ######

```
abstract class RoleBinding
```

```
   extends AnnotationLiteral<Role
>
   implements Role {}
```

################ `fire()` #

```
documentEvent.fire( document, new RoleBinding() { public void value() { return user.getRole();
 } } );
```

9.5.

##############

```
@Observable @Blog Event<Document
> blogEvent;
...
if (document.isBlog()) blogEvent.fire(document, new AnnotationLiteral<Updated
>(){});
```

########################

```
public void afterBlogUpdate(@Observes @Updated @Blog Document document) { ... }
```

```
public void afterDocumentUpdate(@Observes @Updated Document document) { ... }
```

```
public void onAnyBlogEvent(@Observes @Blog Document document) { ... }
```

```
public void onAnyDocumentEvent(@Observes Document document) { ... }}}
```

9.6.

##
`Category` ########

```
public void refreshCategoryTree(@AfterTransactionSuccess @Observes CategoryUpdateEvent
 event) { ... }
```

############

- `@AfterTransactionSuccess` ##################
- `@AfterTransactionFailure` ################
- `@AfterTransactionCompletion` ################
- `@BeforeTransactionCompletion` ################

##########Web Beans###

####################JPA#######

```
@ApplicationScoped @Singleton
public class Catalog {

   @PersistenceContext EntityManager em;

   List<Product
> products;

   @Produces @Catalog
   List<Product
> getCatalog() {
      if (products==null) {
         products = em.createQuery("select p from Product p where p.deleted = false")
            .getResultList();
      }
      return products;
   }

}
```

`Product`############################ `Product`

`Product` #Web Bean##########

```
@Stateless
public class ProductManager {
```

```
    @PersistenceContext EntityManager em;
    @Observable Event<Product
> productEvent;

    public void delete(Product product) {
        em.delete(product);
        productEvent.fire(product, new AnnotationLiteral<Deleted
>(){});
    }

    public void persist(Product product) {
        em.persist(product);
        productEvent.fire(product, new AnnotationLiteral<Created
>(){});
    }

    ...

}
```

Catalog

```
@ApplicationScoped @Singleton
public class Catalog {

    ...

    void addProduct(@AfterTransactionSuccess @Observes @Created Product product) {
        products.add(product);
    }

    void addProduct(@AfterTransactionSuccess @Observes @Deleted Product product) {
        products.remove(product);
    }

}
```

III. æ##å¤§ç¨#åº¦å#°ä½¿ç#¨å¼ºç±»å##

Web Bean##########Web Bean##Java##########

#Web Bean###################################### # ##"#####" # ########################

################IDE###

Web Beans###################

- `@Asynchronous`,
- `@Mock`,
- `@Secure` or
- `@Updated`,

############

- `asyncPaymentProcessor`,
- `mockPaymentProcessor`,
- `SecurityInterceptor` or
- `DocumentUpdatedEvent`.

###

Web Beans###

##Web Bean#XML##########XML######Web Bean#####XML#####XML####Java##########XML################Java##########

########Web Bean###

##

##Web Bean###

########################Web Bean##Web Bean###########

###############

- ##########
- ##########
- ####Web Bean#####
- ##Web Bean##################
- ##########

###################Web Bean######Web Bean###

##Web Bean################

##########Java############MVC#########

```
@Retention(RUNTIME)
@Target(TYPE)
@Stereotype
public @interface Action {}
```

######Web Bean####

```
@Action
public class LoginAction { ... }
```

10.1.

#############Web Bean##########################@WebTier ###################Web###########Web Bean##################

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
```

```
@WebTier
@Stereotype
public @interface Action {}
```

##############################

```
@Dependent @Mock @Action
public class MockLoginAction { ... }
```

##################################

10.2.

########################Web Bean################Web Bean##############

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
@WebTier
@Stereotype(supportedScopes=RequestScoped.class)
public @interface Action {}
```

#################Web Bean########Web Bean#################

##################Web Bean###############

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
@WebTier
@Stereotype(requiredTypes=AbstractAction.class)
public @interface Action {}
```

############### `AbstractAction`#Web Bean#################

10.3.

####################Web Bean###

```
@Retention(RUNTIME)
```

```
@Target(TYPE)
@RequestScoped
@Transactional(requiresNew=true)
@Secure
@WebTier
@Stereotype
public @interface Action {}
```

######################

10.4.

################Web Bean####Web Bean###Web Bean#############JSP## `@Named` ###

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
@Transactional(requiresNew=true)
@Secure
@Named
@WebTier
@Stereotype
public @interface Action {}
```

`LoginAction` ####### `loginAction` Web Bean##.

10.5.

#######Web Bean##############`@Interceptor` # `@Decorator`#

Web Bean#############

```
@Named
@RequestScoped
@Stereotype
@Target({TYPE, METHOD})
@Retention(RUNTIME)
public @interface Model {}
```

#######JSF###############Web Bean### `@Model` ########Web Bean##JSF###Bean#######JSF#######Web Bean#

##

######Web Bean################## ## ##API############Web Bean###########PaymentProcessor######

```
@CreditCard @Stateless
public class CreditCardPaymentProcessor
      implements PaymentProcessor {
   ...
}
```

#########################Web Bean#####PaymentProcessor ###

```
@CreditCard @Stateless @Staging
public class StagingCreditCardPaymentProcessor
      implements PaymentProcessor {
   ...
}
```

####################StagingCreditCardPaymentProcessor#####AsyncPaymentProcessor##############@S

```
@CreditCard PaymentProcessor ccpp
```

StagingCreditCardPaymentProcessor

##################

- ######Web Bean###############API###
- ######Web Bean###################
- ######Web Bean###########Web Bean###########
- #####Web Bean############################

###################Web Bean#################################

Web Bean###

11.1.

##########Web Bean###################Web Bean###

- #####Web Bean########
- ##########Web Bean, #########Web Bean############Web Bean, ##########Web Bean###
- #### `@Specializes` ###

```
@Stateless @Staging @Specializes
public class StagingCreditCardPaymentProcessor
      extends CreditCardPaymentProcessor {
   ...
}
```

#########Web Bean########

11.2.

#######

- #############`@Specializes`###Web Bean#####
- ###Web Bean#######`@Specializes`###Web Bean#####
- ############################`@Specializes`###Web Bean######

#########`CreditCardPaymentProcessor` ##### `@CreditCard` `#StagingCreditCardPaymentProcessor`###

####Web Bean#######

- #####API##### `@Specializes` ###Web Bean#API########Bean################
- ## `@Specializes` ###Web Bean########################
- ##########Web Bean####

#############Web Bean#################

#############`@Specializes`###Web Bean############################

##XML##Web Bean

##################Web Bean####################Web Bean######:

- ########################
- ##Web Bean############

###########Web Bean########

- ###########
- ##XML###Web Bean#

######XML###Java###########Web Bean##################Java###########Web Bean############XML################XML##########Web Bean#

##############XML#####XML,
##XML########################Java#######XML###############IDE#########################

12.1. ##Web Bean#

####Java##Web Bean########XML########## `urn:java:`####Java####### `com.mydomain.myapp` #######XML### `urn:java:com.mydomain.myapp`#

######Java###################XML##########Java##

####XML######`<util:Date/>`### `java.util.Date`

```
<WebBeans xmlns="urn:java:javax.webbeans"
      xmlns:util="urn:java:java.util">

   <util:Date/>

</WebBeans
>
```

####`Date` #######web Bean############### `Date` ##########Web Bean####

```
@Current Date date
```

12.2. ##Web Bean####

########Web Bean#########################

```
<myapp:ShoppingCart>
   <SessionScoped/>
   <myfwk:Transactional requiresNew="true"/>
   <myfwk:Secure/>
</myapp:ShoppingCart
>
```

######################

```
<util:Date>
   <Named
>currentTime</Named>
</util:Date>

<util:Date>
   <SessionScoped/>
   <myapp:Login/>
   <Named
>loginTime</Named>
</util:Date>

<util:Date>
   <ApplicationScoped/>
   <myapp:SystemStart/>
   <Named
>systemStartTime</Named>
</util:Date
>
```

`@Login` # `@SystemStart` ########

```
@Current Date currentTime;
@Login Date loginTime;
@SystemStart Date systemStartTime;
```

#####Web Bean###########

```
<myapp:AsynchronousChequePaymentProcessor>
   <myapp:PayByCheque/>
   <myapp:Asynchronous/>
</myapp:AsynchronousChequePaymentProcessor
>
```

###############Web Beans############Web Bean######

```
<myfwk:TransactionInterceptor>
   <Interceptor/>
   <myfwk:Transactional/>
</myfwk:TransactionInterceptor
>
```

12.3. ##Web Bean##

TODO!

12.4. #####Web Beans

Web Beans################Web Bean####

```
<myapp:System>
   <ApplicationScoped/>
   <myapp:admin>
      <myapp:Name>
         <myapp:firstname
>Gavin</myapp:firstname>
         <myapp:lastname
>King</myapp:lastname>
         <myapp:email
>gavin@hibernate.org</myapp:email>
      </myapp:Name>
   </myapp:admin>
</myapp:System
>
```

`<Name>` `##########` `@Dependent` `###` `Name` `######`Web Bean################Web Bean##################################

####################Web Bean XML#########Java##########################

12.5.

#########XML######Java##################################Web Beans################

```
<WebBeans xmlns="urn:java:javax.webbeans"
        xmlns:myapp="urn:java:com.mydomain.myapp"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:java:javax.webbeans http://java.sun.com/jee/web-beans-1.0.xsd
                          urn:java:com.mydomain.myapp http://mydomain.com/xsd/myapp-1.2.xsd">

   <myapp:System>
      ...
   </myapp:System>

</WebBeans
>
```

####XML##########Web Bean#######################Java########XML###

IV. Web Beansså##Java EEç##æ##ç³»ç»#

Web Bean##########Web Bean##Web Bean##############Java EE###

#######Web Bean######EJB#JSF, #####EJB#####JSF###########Web Bean################################Web###########Java EE########Java#######################Web Bean##Java EE####

################Web Bean#####Java EE##################Web Bean########SPI################SPI###

Java EE##

Web Beans########Java EE####Web Beans ####Java EE###JPA############JSF#JSP##################################Servlet#####Bean#########Web Bean#

13.1. #Java EE#######Web Bean#

#########Web Beans#####`@Resource`, `@EJB`#`@PersistenceContext`###Java EE##################################

```
@Transactional @Interceptor
public class TransactionInterceptor {

    @Resource Transaction transaction;

    @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }

}
```

```
@SessionScoped
public class Login {

    @Current Credentials credentials;
    @PersistenceContext EntityManager userDatabase;

    ...

}
```

##########Web Beans###Java EE#`@PostConstruct`#`@PreDestroy`###`@PostConstruct`###### #########

#################Web Beans### `@PersistenceContext(type=EXTENDED)`

13.2. #Servlet####Web Bean

#Java EE 6#####Servlet#####Web Bean#########Web Beans#####Web Bean###########

```
public class Login extends HttpServlet {
```

```
    @Current Credentials credentials;
    @Current Login login;

    @Override
    public void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        credentials.setUsername( request.getAttribute("username") ):
        credentials.setPassword( request.getAttribute("password") ):
        login.login();
        if ( login.isLoggedIn() ) {
            response.sendRedirect("/home.jsp");
        }
        else {
            response.sendRedirect("/loginError.jsp");
        }
    }

}
```

Web Beans######Servlet###############HTTP######`Credentials` # `Login` ###

13.3. #####Bean#####Web Bean

Web Beans##########EJB#####EJB##Web Bean#####################JNDI######`@EJB`########Bean####Web Beans##################Web Beans#######Bean##

######Web Beans##########Bean##

```
@Transactional @MessageDriven
public class ProcessOrder implements MessageListener {

    @Current Inventory inventory;
    @PersistenceContext EntityManager em;

    public void onMessage(Message message) {
        ...
    }

}
```

######Web Beans#####################################Bean############################## `@RequestScoped` # `@ApplicationScoped` ###Web Beans#

##Web Beans##########

13.4. JMS##

##JMS##############################Queue#####`Queue`, `QueueConnectionFactory`, `QueueConnection`, `QueueSession` # `QueueSender`######Topic##`Topic`, `TopicConnectionFactory`, `TopicConnection`, `TopicSession` # `TopicPublisher`###################################

Web Beans################## `web-beans.xml`##############################

```
<Queue>
   <destination
>java:comp/env/jms/OrderQueue</destination>
   <connectionFactory
>java:comp/env/jms/QueueConnectionFactory</connectionFactory>
   <myapp:OrderProcessor/>
</Queue
>
```

```
<Topic>
   <destination
>java:comp/env/jms/StockPrices</destination>
   <connectionFactory
>java:comp/env/jms/TopicConnectionFactory</connectionFactory>
   <myapp:StockPrices/>
</Topic
>
```

############### `Queue`, `QueueConnection`, `QueueSession` ## `QueueSender`############`Topic`, `TopicConnection`, `TopicSession` ## `TopicPublisher`#

```
@OrderProcessor QueueSender orderSender;
@OrderProcessor QueueSession orderSession;

public void sendMessage() {
   MapMessage msg = orderSession.createMapMessage();
   ...
```

```
    orderSender.send(msg);
}
```

```
@StockPrices TopicPublisher pricePublisher;
@StockPrices TopicSession priceSession;

public void sendMessage(String price) {
    pricePublisher.send( priceSession.createTextMessage(price) );
}
```

####JMS##########Web Bean######

13.5.

Web Beans#################JAR, EJB-JAR##WAR####Web Beans#########################Web Beans#####`META-INF` ## `WEB-INF`######### `web-beans.xml`##################`web-beans.xml` ############Web Beans#

##Java SE#Web Beans##########EJB#######EJB################## `web-beans.xml` #####

##Web Bean

Web Beans###################################Web Bean###SPI###################Web Bean#######Web Bean###########

- ###########
- ##########Spring, Seam, GWT##Wicket##
- ##Web Bean#########

##Web Bean###### `Manager` ###

14.1. `Manager`

`Manager` #################Web Bean#################

```
public interface Manager
{

  public <T
> Set<Bean<T
>
> resolveByType(Class<T
> type, Annotation... bindings);

  public <T
> Set<Bean<T
>
> resolveByType(TypeLiteral<T
> apiType,
      Annotation... bindings);

  public <T
> T getInstanceByType(Class<T
> type, Annotation... bindings);

  public <T
> T getInstanceByType(TypeLiteral<T
> type,
      Annotation... bindings);

  public Set<Bean<?>
> resolveByName(String name);
```

```
   public Object getInstanceByName(String name);

   public <T
> T getInstance(Bean<T
> bean);

   public void fireEvent(Object event, Annotation... bindings);

   public Context getContext(Class<? extends Annotation
> scopeType);

   public Manager addContext(Context context);

   public Manager addBean(Bean<?> bean);

   public Manager addInterceptor(Interceptor interceptor);

   public Manager addDecorator(Decorator decorator);

   public <T
> Manager addObserver(Observer<T
> observer, Class<T
> eventType,
         Annotation... bindings);

   public <T
> Manager addObserver(Observer<T
> observer, TypeLiteral<T
> eventType,
         Annotation... bindings);

   public <T
> Manager removeObserver(Observer<T
> observer, Class<T
> eventType,
         Annotation... bindings);

   public <T
> Manager removeObserver(Observer<T
> observer,
         TypeLiteral<T
> eventType, Annotation... bindings);

   public <T
```

```
> Set<Observer<T
>
> resolveObservers(T event, Annotation... bindings);

  public List<Interceptor
> resolveInterceptors(InterceptionType type,
        Annotation... interceptorBindings);

  public List<Decorator
> resolveDecorators(Set<Class<?>
> types,
        Annotation... bindings);

}
```

############## Manager ###

```
@Current Manager manager
```

14.2. Bean

Bean #####Web Bean########Web Bean##### Manager ### Bean

```
public abstract class Bean<T> {

   private final Manager manager;

   protected Bean(Manager manager) {
      this.manager=manager;
   }

   protected Manager getManager() {
      return manager;
   }

   public abstract Set<Class> getTypes();
   public abstract Set<Annotation> getBindingTypes();
   public abstract Class<? extends Annotation> getScopeType();
   public abstract Class<? extends Annotation> getDeploymentType();
   public abstract String getName();

   public abstract boolean isSerializable();
```

```
    public abstract boolean isNullable();

    public abstract T create();
    public abstract void destroy(T instance);

}
```

Bean ###### Manager.addBean() ##############Web Bean#######Web Bean, ##WebBean, ######JMS######Web Bean############ Bean ###################Web Bean##

Web Bean######## Bean ######## # ####

14.3. Context

Context #####Web Bean#########################

```
public interface Context {

    public Class<? extends Annotation> getScopeType();

    public <T> T get(Bean<T> bean, boolean create);

    boolean isActive();

}
```

######### Context ##Web Bean#################Wicket###############

###

##Web Bean#######################

###Web Bean#####Web Bean###############100##http://jcp.org/en/jsr/detail?id=299###

Web Bean######http://seamframework.org/WebBeans################Web Bean#########http://in.relation.to#############################

V. Web Beans Reference

Web Beans is the reference implementation of JSR-299, and is used by JBoss AS and Glassfish to provide JSR-299 services for Java Enterprise Edition applications. Web Beans also goes beyond the environments and APIs defined by the JSR-299 specification and provides support for a number of other environments (such as a servlet container such as Tomcat, or Java SE) and additional APIs and modules (such as logging, XSD generation for the JSR-299 XML deployment descriptors).

If you want to get started quickly using Web Beans with JBoss AS or Tomcat and experiment with one of the examples, take a look at *# 3 # ##Web Beans#####*. Otherwise read on for a exhaustive discussion of using Web Beans in all the environments and application servers it supports, as well the Web Beans extensions.

Application Servers and environments supported by Web Beans

16.1. Using Web Beans with JBoss AS

No special configuration of your application, beyond adding either `META-INF/beans.xml` or `WEB-INF/beans.xml` is needed.

If you are using JBoss AS 5.0.1.GA then you'll need to install Web Beans as an extra. First we need to tell Web Beans where JBoss is located. Edit `jboss-as/build.properties` and set the `jboss.home` property. For example:

```
jboss.home=/Applications/jboss-5.0.1.GA
```

Now we can install Web Beans:

```
$ cd webbeans-$VERSION/jboss-as
$ ant update
```

##

A new deployer, `webbeans.deployer` is added to JBoss AS. This adds supports for JSR-299 deployments to JBoss AS, and allows Web Beans to query the EJB3 container and discover which EJBs are installed in your application.

Web Beans is built into all releases of JBoss AS from 5.1 onwards.

16.2. Glassfish

TODO

16.3. Tomcat (or any plain Servlet container)

Web Beans can be used in Tomcat 6.0.

##

Web Beans doesn't support deploying session beans, injection using `@EJB`, or `@PersistenceContext` or using transactional events on Tomcat.

Web Beans should be used as a web application library in Tomcat. You should place `webbeans-tomcat.jar` in `WEB-INF/lib`. `webbeans-tomcat.jar` is an "uber-jar" provided for your convenience. Instead, you could use its component jars:

- `jsr299-api.jar`
- `webbeans-api.jar`
- `webbeans-spi.jar`
- `webbeans-core.jar`
- `webbeans-logging.jar`
- `webbeans-tomcat-int.jar`
- `javassist.jar`
- `dom4j.jar`

You also need to explicitly specify the Tomcat servlet listener (used to boot Web Beans, and control its interaction with requests) in `web.xml`:

```
<listener>
  <listener-class
>org.jboss.webbeans.environment.servlet.Listener</listener-class>
</listener
>
```

Tomcat has a read-only JNDI, so Web Beans can't automatically bind the Manager. To bind the Manager into JNDI, you should add the following to your `META-INF/context.xml`:

```
<Resource name="app/Manager"
        auth="Container"
        type="javax.inject.manager.Manager"
        factory="org.jboss.webbeans.resources.ManagerObjectFactory"/>
```

and make it available to your deployment by adding this to `web.xml`:

```
<resource-env-ref>
  <resource-env-ref-name>
    app/Manager
  </resource-env-ref-name>
  <resource-env-ref-type>
    javax.inject.manager.Manager
  </resource-env-ref-type>
</resource-env-ref>
```

Tomcat only allows you to bind entries to `java:comp/env`, so the Manager will be available at `java:comp/env/app/Manager`

Web Beans also supports Servlet injection in Tomcat. To enable this, place the `webbeans-tomcat-support.jar` in `$TOMCAT_HOME/lib`, and add the following to your `META-INF/context.xml`:

```
<Listener className="org.jboss.webbeans.environment.tomcat.WebBeansLifecycleListener" />
```

16.4. Java SE

Apart from improved integration of the Enterprise Java stack, Web Beans also provides a state of the art typesafe, stateful dependency injection framework. This is useful in a wide range of application types, enterprise or otherwise. To facilitate this, Web Beans provides a simple means for executing in the Java Standard Edition environment independently of any Enterprise Edition features.

When executing in the SE environment the following features of Web Beans are available:

- `Simple Web Beans (POJOs)`
- `Typesafe Dependency Injection`
- `Application and Dependent Contexts`
- `Binding Types`
- `Stereotypes`
- `Decorators`
- `(TODO: Interceptors ?)`
- `Typesafe Event Model`

16.4.1. Web Beans SE Module

To make life easy for developers Web Beans provides a special module with a main method which will boot the Web Beans manager, automatically registering all simple Web Beans found on the classpath. This eliminates the need for application developers to write any bootstrapping code. The entry point for a Web Beans SE applications is a simple Web Bean which observes the standard `@Deployed Manager` event. The command line paramters can be injected using either of the following:

```
@Parameters List<String
> params;
@Parameters String[] paramsArray; // useful for compatability with existing classes
```

Here's an example of a simple Web Beans SE application:

```
@ApplicationScoped
public class HelloWorld
{
   @Parameters List<String
> parameters;

   public void printHello( @Observes @Deployed Manager manager )
   {
      System.out.println( "Hello " + parameters.get(0) );
   }
}
```

Web Beans SE applications are started by running the following main method.

```
java org.jboss.webbeans.environments.se.StartMain <args
>
```

If you need to do any custom initialization of the Web Beans manager, for example registering custom contexts or initializing resources for your beans you can do so in response to the `@Initialized Manager` event. The following example registers a custom context:

```
public class PerformSetup
{

   public void setup( @Observes @Initialized Manager manager )
```

```
  {
     manager.addContext( ThreadContext.INSTANCE );
  }
}
```

##

The command line parameters do not become available for injection until the `@Deployed Manager` event is fired. If you need access to the parameters during initialization you can do so via the `public static String getParameters()` method in `StartMain`.

JSR-299 extensions available as part of Web Beans

##

These modules are usable on any JSR-299 implementation, not just Web Beans!

17.1. Web Beans Logger

TODO

17.2. XSD Generator for JSR-299 XML deployment descriptors

TODO

A. #Web Bean##############

###Web Bean##########JBoss AS5############EE#######Glassfish##############Servlet####Tomcat#########EJB3.1###############

##

Web Bean###SE##################################Web Bean##############################Web Bean#######

A.1. Web Bean#####SPI

The Web Beans SPI is located in `webbeans-spi` module, and packaged as `webbeans-spi.jar`. Some SPIs are optional, if you need to override the default behavior, others are required.

###SPI################## `Forwarding` ##

A.1.1. Web Bean###

```
public interface WebBeanDiscovery {
  /**
   * Gets list of all classes in classpath archives with web-beans.xml files
   *
   * @return An iterable over the classes
   */
  public Iterable<Class<?>
> discoverWebBeanClasses();

  /**
   * Gets a list of all web-beans.xml files in the app classpath
   *
   * @return An iterable over the web-beans.xml files
   */
  public Iterable<URL
> discoverWebBeansXml();

}
```

Web Bean##### `web-bean.xml` ################JSR-299################

A.1.2. EJB services

Web Bean######EJB3 Bean##################EJB3##### `ejb-jar.xml` ############EJB######EJBDescriptor#

```
public interface EjbServices
{

  /**
   * Gets a descriptor for each EJB in the application
   *
   * @return The bean class to descriptor map
   */
  public Iterable<EjbDescriptor<?>> discoverEjbs();
```

```
public interface EjbDescriptor<T
> {

  /**
   * Gets the EJB type
   *
   * @return The EJB Bean class
   */
  public Class<T
> getType();

  /**
   * Gets the local business interfaces of the EJB
   *
   * @return An iterator over the local business interfaces
   */
  public Iterable<BusinessInterfaceDescriptor<?>
> getLocalBusinessInterfaces();

  /**
   * Gets the remote business interfaces of the EJB
   *
   * @return An iterator over the remote business interfaces
   */
  public Iterable<BusinessInterfaceDescriptor<?>
> getRemoteBusinessInterfaces();
```

```
  /**
   * Get the remove methods of the EJB
   *
   * @return An iterator over the remove methods
   */
  public Iterable<Method
> getRemoveMethods();

  /**
   * Indicates if the bean is stateless
   *
   * @return True if stateless, false otherwise
   */
  public boolean isStateless();

  /**
   * Indicates if the bean is a EJB 3.1 Singleton
   *
   * @return True if the bean is a singleton, false otherwise
   */
  public boolean isSingleton();

  /**
   * Indicates if the EJB is stateful
   *
   * @return True if the bean is stateful, false otherwise
   */
  public boolean isStateful();

  /**
   * Indicates if the EJB is and MDB
   *
   * @return True if the bean is an MDB, false otherwise
   */
  public boolean isMessageDriven();

  /**
   * Gets the EJB name
   *
   * @return The name
   */
  public String getEjbName();
```

```
}
```

`EjbDescriptor` ###############EJB###################################
`BusinessInterfaceDescriptor` ############EJB###jndi###

The resolution of `@EJB` and `@Resource` is delegated to the container. You must provide an implementation of `org.jboss.webbeans.ejb.spi.EjbServices` which provides these operations. Web Beans passes in the `javax.inject.manager.InjectionPoint` the resolution is for, as well as the `NamingContext` in use for each resolution request.

A.1.3. JPA services

Just as resolution of `@EJB` is delegated to the container, so is resolution of `@PersistenceContext`.

OPEN ISSUE: Web Beans also requires the container to provide a list of entities in the deployment, so that they aren't discovered as simple beans.

A.1.4.

Web Bean RI###JTA########SPI#########(hooks)##`TransactionServices` ############

```
public interface TransactionServices
{
  /**
   * Possible status conditions for a transaction. This can be used by SPI
   * providers to keep track for which status an observer is used.
   */
  public static enum Status
  {
    ALL, SUCCESS, FAILURE
  }

  /**
   * Registers a synchronization object with the currently executing
   * transaction.
   *
   * @see javax.transaction.Synchronization
   * @param synchronizedObserver
   */
  public void registerSynchronization(Synchronization synchronizedObserver);

  /**
   * Queries the status of the current execution to see if a transaction is
   * currently active.
   *
```

```
    * @return true if a transaction is active
    */
   public boolean isTransactionActive();
}
```

Status

javax.transaction.Synchronization ####### registerSynchronization() ###SPI############EJB###JTA######

#####################################isTransactionActive() ###SPI######EJB######JTA######

A.1.5.

Web Bean##############################org.jboss.webbeans.context.api.BeanStore ########################## org.jboss.webbeans.context.api.helpers.ConcurrentHashMapBeanStore #####

A.1.6.

org.jboss.webbeans.bootstrap.api.Bootstrap #####Web Bean##########Web Beans########org.jboss.webbeans.bootstrap.WebBeansBootstrap #######Boostrap# #######SPI##########

The bootstrap is split into phases, bootstrap initialization and boot and shutdown. Initialization will create a manager, and add the standard (specification defined) contexts. Bootstrap will discover EJBs, classes and XML; add beans defined using annotations; add beans defined using XML; and validate all beans.

The bootstrap supports multiple environments. Different environments require different services to be present (for example servlet doesn't require transaction, EJB or JPA services). By default an EE environment is assumed, but you can adjust the environment by calling `bootstrap.setEnvironment()`.

To initialize the bootstrap you call `Bootstrap.initialize()`. Before calling `initialize()`, you must register any services required by your environment. You can do this by calling `bootstrap.getServices().add(JpaServices.class, new MyJpaServices())`. You must also provide the application context bean store.

##initialize()########Bootstrap.getManager()#######

To boot the container you call `Bootstrap.boot()`.

########### Bootstrap.shutdown()###################

A.1.7. JNDI

Web Beans #######JNDI ####################################JNDI############## `org.jboss.webbeans.resources.spi.NamingContext########`

```
public interface NamingContext extends Serializable {

   /**
    * Typed JNDI lookup
    *
    * @param <T
> The type
    * @param name The JNDI name
    * @param expectedType The expected type
    * @return The object
    */
   public <T
> T lookup(String name, Class<? extends T
> expectedType);

   /**
    * Binds an item to JNDI
    *
    * @param name The key to bind under
    * @param value The item to bind
    */
   public void bind(String name, Object value);

}
```

A.1.8.

Web Beans###`org.jbo`
#

```
      public interface ResourceLoader {

   /**
    * Creates a class from a given FQCN
    *
```

```
   * @param name The name of the clsas
   * @return The class
   */
  public Class<?> classForName(String name);

  /**
   * Gets a resource as a URL by name
   *
   * @param name The name of the resource
   * @return An URL to the resource
   */
  public URL getResource(String name);

  /**
   * Gets resources as URLs by name
   *
   * @param name The name of the resource
   * @return An iterable reference to the URLS
   */
  public Iterable<URL
> getResources(String name);

}
```

A.1.9. Servlet

Java EE/Servlet###############Servlet#####Web Bean###############Servlet##JSR-299####

###JSR-299#################Servlet#################Servlet##Servlet###

##Servlet####### `WebBeansManager.injectServlet()`###### `Bootstrap.getManager()`#######

A.2.

###############Web Beans###############

######

######Web Bean##Web Bean###########

Servlet#######

####Web Bean#####Servlet#########Servlet# Web Bean################## `org.jboss.webbeans.servlet.WebBeansListener` #####Servlet####

####Web Bean#####JSF#########JSF# Web Bean###################
`org.jboss.webbeans.servlet.ConversationPropagationFilter`
#####Servlet#################Servlet#####

##Bean###

####Web Beans###EJB############Bean#Web Bean###################
`org.jboss.webbeans.ejb.SessionBeanInterceptor` ########EJB#EJB####

##

########EJB####`SessionBeanInterceptor` ##############

The `webbeans-core.jar`

If you are integrating the Web Beans into an environment that supports deployment of applications, you must insert the `webbeans-core.jar` into the applications isolated classloader. It cannot be loaded from a shared classloader.