



| IBM Software Group

# WAS V7 Application Development

**An IBM Proof of Technology**

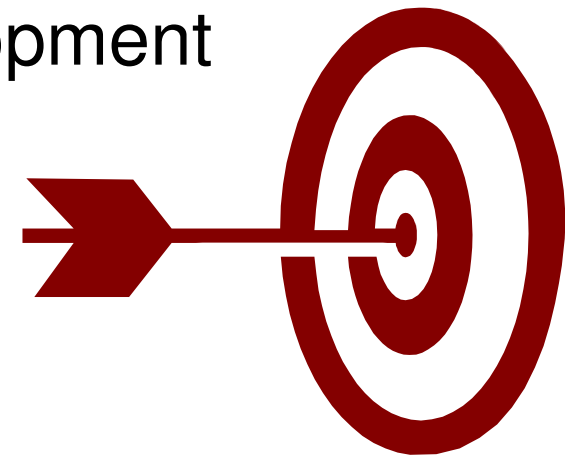


@business on demand.

© 2008 IBM Corporation  
Updated September 28, 2009

# WAS v7 Programming Model Goals

- One word ..... **Simplify**
  - ▶ Simplify the programming model
  - ▶ Simplify application assembly / packaging
  - ▶ Simplify testing
  - ▶ Simplify Object Relational Mapping
  - ▶ Simplify Web Services Development



# Agenda

- Java Enterprise Edition (Java EE) 5
  - ▶ EJB 3.0
  - ▶ JPA
- Web Services
  - ▶ JEE 5
  - ▶ WS\*
- Application Development Tooling
  - ▶ Rational Application Developer (RAD)



| IBM Software Group

# Java EE 5

## An IBM Proof of Technology



@business on demand.

© 2008 IBM Corporation  
Updated September 28, 2009

## WAS V7 supports Java EE 5

- Java annotations (JSR 175)
- Enterprise JavaBeans (EJB) 3.0
- Java Persistence API (JPA)
- Web services (JAX-WS 2.0)
- Web application technologies
  - ▶ JSP 2.1 ( Java Server Pages)
  - ▶ JSF 1.2 ( Java Server Faces)
  - ▶ Servlet 2.5

## @Annotations

- Java EE 5 annotations let you embed resources, dependencies, services, and life-cycle notifications directly in your source code
  - ▶ Replace descriptors for most purposes
  - ▶ Reduce the number of artifacts you have to maintain
  - ▶ Allow application settings to be visible in the component they affect
- Can be used, e.g., when:
  - ▶ Developing Enterprise JavaBeans applications
  - ▶ Defining and using Web services
  - ▶ Mapping Java technology classes to databases

## Annotation based Dependency Injection

- Resource dependencies can be declared using annotations
  - ▶ Inversion-of-Control (IoC) pattern eliminates boilerplate code and lets the container do the work
  - ▶ No longer need to use factories to get objects from JNDI
- Examples:
  - ▶ @EJB ShoppingCart myCart
  - ▶ @Resource DataSource bankDS
- WAS v7 uses dependency injection to avoid clutter, allowing you to focus on business logic...

## Security Annotations and Authorization

- Annotations can be used for role-based authorization
- Provide equivalent functionality to:
  - ▶ `<security-role>`, `<method-permission>`, `<run-as>`, `<exclude-list>`
- Five annotations are supported:
  - ▶ Servlets: **@DeclareRoles** and **@RunAs**
  - ▶ EJB: **@DeclareRoles**, **@DenyAll**, **@PermitAll**, **@RolesAllowed**, **@RunAs**



IBM Software Group

# EJB 3.0

## An IBM Proof of Technology



@business on demand.

© 2008 IBM Corporation  
Updated September 28, 2009

## EJB 3 Programming Model

- Two types of EJBs:
  - ▶ Session beans (stateless and stateful)
  - ▶ Message driven beans
- A complete new persistency model
  - ▶ based on the JPA standard
  - ▶ supersedes EJB 2.x entity beans
- Make it as easy as possible to implement simple things, while keeping complex things possible



### Simplification

## Simplifying EJB

- EJBs are now POJOs
  - ▶ Expose regular business interfaces
  - ▶ No need to implement *javax.ejb.SessionBean*
  - ▶ Interfaces no longer have to implement *EJBObject*
  - ▶ No longer have to throw *RemoteException*
  - ▶ Home interfaces no longer required
  - ▶ EJB implementations no longer required to implement EJB life cycle methods. Use callback annotations instead.

## Example: Simplifying EJB - continued

### EJB 2.1

```
public interface ShoppingCart
    extends EJBObject {
    public int
    someShoppingMethod()
    throws RemoteException;
}
```

### EJB 3.0

```
public interface ShoppingCart
{
    public int
    someShoppingMethod();
}
```

### EJB 2.1

```
Object obj =
    Context.lookup("java:comp/env/ejb/
    MyCartHome");
CartHome theCartHome = (CartHome)
    PortableRemoteObject.narrow(obj,
    CartHome.class);
ShoppingCart myCart =
    theCartHome.create();

myCart.someShoppingMethod();
```

### EJB 3.0

```
@EJB
ShoppingCart myCart;

myCart.someShoppingMethod();
```

# EJB 2.1 Descriptor vs EJB 3.0

## EJB 2.1

```

<ejb-jar>
<enterprise-beans>
  <session>
    <description>Account Transfer Controller EJB</description>
    <display-name>Account Transfer Controller EJB</display-name>
    <ejb-name>AccountTransferController</ejb-name>
    <home>com.mycompany.AccountTransferControllerHome</home>
    <remote>com.mycompany.AccountTransferController</remote>
    <ejb-class>com.mycompany.AccountTransferControllerImpl</ejb-class>
    <session-type>Stateless</session-type>
    <transaction-type>Container</transaction-type>
  </session>
</enterprise-beans>
<assembly-descriptor>
  <container-transaction>
    <method>
      <ejb-name>AccountTransferController</ejb-name>
      <method-name>*</method-name>
    </method>
    <trans-attribute>Required</trans-attribute>
  </container-transaction>
</assembly-descriptor>
</ejb-jar>

```

## EJB 3.0

```

@Stateless
public class AccountTransferControllerBean
    implements AccountTransferController {

    public void transfer(Account src, Account, dest, float amount)
        throws InsufficientFundsException
    { ... }

}

```

# Annotations vs EJB Deployment Descriptor

- EJB deployment descriptors are now optional
  - ▶ Used to override information specified in the annotations.
  - ▶ Can be sparsely populated. No entries needed except to override defaults.
  - ▶ Can be used as an alternative to annotations or in combination.
- Annotations assume the obvious – but allow specification of non-default values.
  - ▶ Configuration by Exception
- Annotations to allow specification of meta-data, making deployment descriptors optional
  - ▶ @Stateless                      stateless session bean
  - ▶ @Stateful                        stateful session bean
  - ▶ @MessageDriven                message-driven bean
  - ▶ @Entity                          JPA entity

## WAS V7 EJB Backward Compatibility

- EJBs 1 and 2 are still supported by WAS v7 EJB container
- To call an older bean from an EJB 3.0 bean:
  - ▶ Inject the home interface

```
@EJB  
private CustomerHome customer;
```

- To call an EJB 3.0 bean from an older bean:
  - ▶ Add an EJB reference to lookup the business interface

```
Customer3 customer =  
(Customer3) ctx.lookup("java:comp/env/Customer3");
```

- Or look up the home interface and use the bean just as you would an older bean
  - The EJB 3.0 bean must have a home interface, and a component (EJBObject/EJBLocalObject based) interface



| IBM Software Group

# Java Persistence API (JPA)

**An IBM Proof of Technology**



@business on demand.

© 2008 IBM Corporation  
Updated September 28, 2009

# The Object Relational Mapping Challenge

- Persistence is a challenge in EJB 2.1
  - ▶ Complex programming model
  - ▶ Entity beans considered too heavyweight
  - ▶ EJB-QL considered too limited
- Other approaches have emerged in response
  - ▶ Custom frameworks or plain JDBC™
  - ▶ Object-relational mapping (ORM) solutions
    - Hibernate
    - OpenJPA
    - Toplink (now EclipseLink)
    - Others

## Java Persistence API (JPA)

- JPA is a standard persistence and object-relational mapping (ORM) framework for Java
  - ▶ Part of the EJB 3.0 specification (JSR220)
  - ▶ Provided by the `javax.persistence` package
- Enables persisting POJOs to a relational database
  - No deployment code or abstract classes
  - No interfaces required
  - ▶ Metadata is specified with Java annotations or in XML deployment descriptors
- JPA is available in both Java SE and Java EE environments

# Entity Bean vs JPA

## EJB 2.1 Entity Bean

```
public abstract class CustomerBean implements
    Customer, EntityBean {
    public CustomerBean() { }
    public abstract String getName();
    public abstract void setName(String n);
    public abstract int getAmountSpent();
    public abstract void setAmountSpent(int
        amount);
    private EntityContext ctx;

    public String ejbCreateByName(String) throws
        EJBException { }
    public void setEntityContext (EntityContext
        theCtx) throws EJBException {
        ctx = theCtx; }
    public void unsetEntityContext() throws
        EJBException {
        ctx = null; }
}
```

## JPA

```
@Entity @Table (name="CUSTS")
public class Customer implements Serializable {
    public Customer() { }
    @Id
    public String getName() { return name; }
    public void setName(String n) { name = n; }
    public int getAmountSpent() { return
        amountSpent; }
    public void setAmountSpent(int a) {
        amountSpent = a; }
    private String name;
    private int amountSpent;
}
```



| IBM Software Group

# Web Services

## An IBM Proof of Technology



@business on demand.

© 2008 IBM Corporation  
Updated September 28, 2009

## Web services: Java EE 5 - Standards

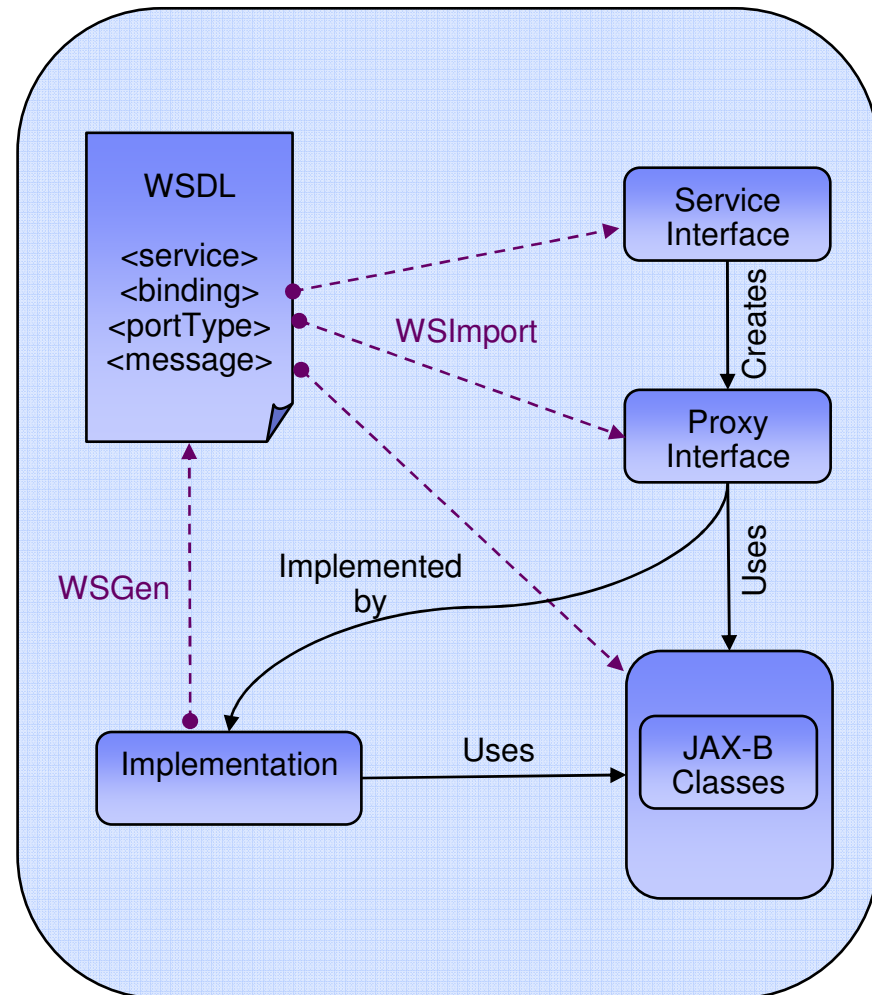
- Java API for XML Web services (JAX-WS)
  - ▶ Supersedes JAX-RPC
  - ▶ Simplified Web service development
  - ▶ Standardized programming model for using WS-Addressing
- JAXB 2.1 for simpler XML to Java mappings, including improved support for XML type substitution
- StAX 1.0 for high-performance streaming parsing

## JAX-WS Web Services

- Support for generic XML/HTTP as a protocol binding
  - ▶ Create clients and providers that do not use SOAP for their wire level message format
- Asynchronous support for clients
  - ▶ Using either a polling or callback mechanism
- Supports Message Transmission Optimization Mechanism (MTOM)
  - ▶ Improved method for sending binary attachments
- As in JAX-RPC, JAX-WS provides a mechanism for application handlers
  - ▶ Insert or retrieve data from a message as it moves through the Web services engine

# JAX-WS Web Services Tools

- **WSImport**
  - ▶ for top down development
    - Create a Web service form a WSDL
    - JAX-WS equivalent of WSDL2Java
- **WSGen**
  - ▶ for bottom up development
    - Create a Web service from Java code
    - JAX-WS equivalent of Java2WSDL
- **Java SE 6**



# JAX RPC vs JAX-WS

## JAX-RPC 1.1 Code

```
public interface StockQuote
    extends Remote {
    public float getQuote(String sym)
        throws RemoteException;
    }

    public class QuoteBean implements
    {
        public float getQuote(String sym)
        { ... }
    }
```

## JAX-WS 2.0 Code

### @WebService

```
public interface StockQuote {
    public float getQuote(String sym);
}
```

### @WebService

```
public class QuoteBean implements
    StockQuote {
    @WebMethod
    public float getQuote(String sym)
    { ... }
}
```

## WAS v7 WS\* Standards

- Support for OASIS standards
  - ▶ Emphasize interoperability (with, for example .NET):
- WS-I profiles:
  - ▶ WS Reliable Secure Profile
  - ▶ Basic Profile 1.2 and 2.0
- WS-ReliableMessaging
- WS-SecureConversation
- WS-Trust

## Additional WAS V7 WS\* Standards

- WS-Transaction
- WS-SecurityPolicy
- Kerberos Token Profile
- W3C WS-Policy
- W3C WS-Addressing Metadata
- W3C SOAP 1.2
- W3C MTOM and XOP
- WS-MetadataExchange

# WAS v7 Standards

- WAS v7 Standards

- ▶ [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rovr\\_specs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rovr_specs.html)

- Web Services Standards

- ▶ <http://publib.boulder.ibm.com/infocenter/radhelp/v7r5/index.jsp?topic=/com.ibm.webservice.wsfp.doc/topics/cwsfpstandards.html>



| IBM Software Group

# Rational Application Developer (RAD)

**An IBM Proof of Technology**



@business on demand.

© 2008 IBM Corporation  
Updated September 28, 2009

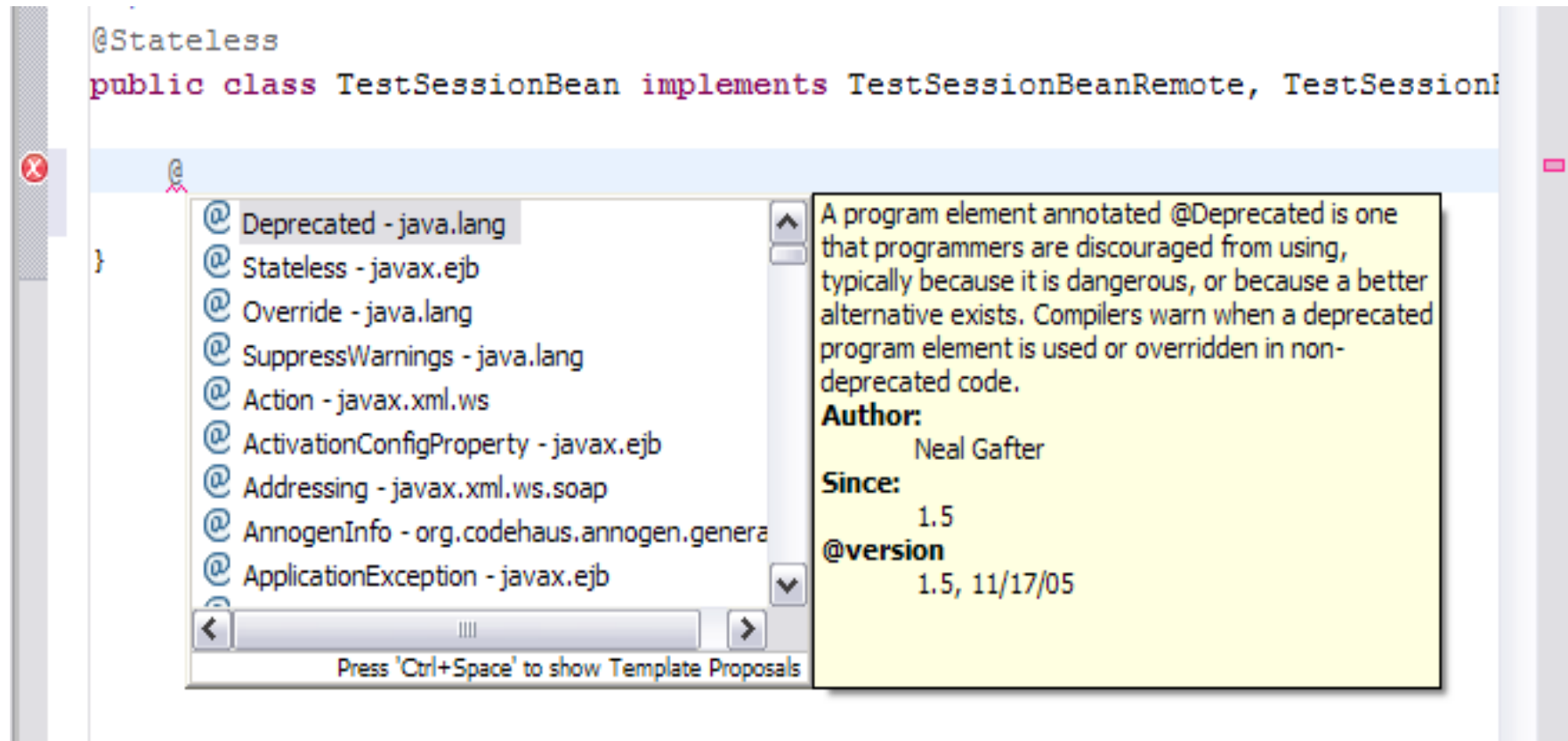
## RAD 7.5: Background

- Rational Application Developer (RAD) 7.5
  - ▶ Based on Eclipse 3.4
- IBM's IDE for WAS 7
  - ▶ Has built-in WAS 7 server for testing purposes
- RAD 7.5 can also create & test older, J2EE 1.4, 1.3, and 1.2 applications

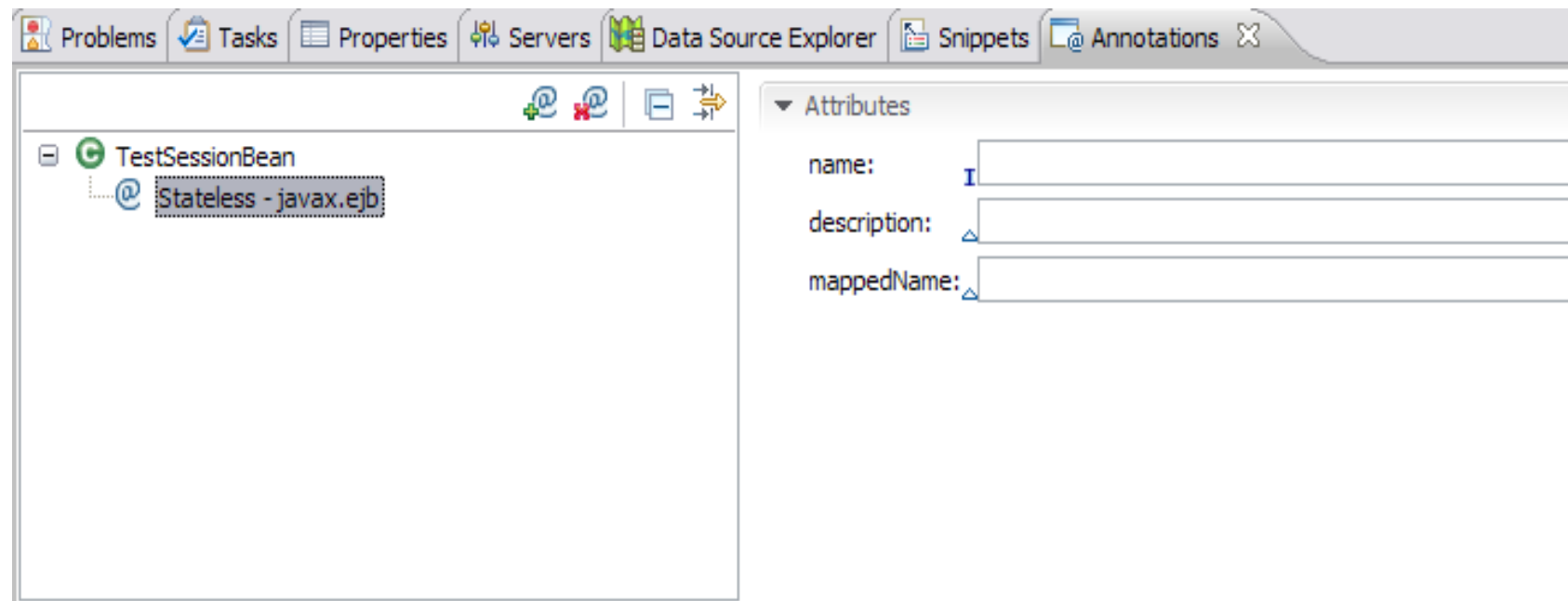
## Annotation support in RAD 7.5

- While editing Java files annotations can be added using the content assist function
  - ▶ Type '@' then hit **ctrl+space** to display a list of available annotations including annotations for
    - EJB 3.0
    - JAX-WS
    - JPA
- The editor provides validation for annotations while you type
- Annotations are also able to be edited, added, and removed using the new Annotation view

# Content assist helps add annotations



# The Annotation view



## RAD 7.5: Creating an EJB 3.0 project

- The EJB project creation wizard now defaults to EJB 3.0 support
- Older EJBs are still supported and available as options in the drop down list
- You cannot create
  - ▶ EJB 1.1, 2.0, 2.1 bean in an EJB 3.0 Project
  - ▶ EJB 3.0 bean in a non 3.0 compliant project

## New Session Bean generated code

```
TestSessionBean.java X
package com.duck.ejb.session.slide;

import javax.ejb.Stateless;

/**
 * Session Bean implementation class for: TestSessionBean
 */
@Stateless
public class TestSessionBean implements TestSessionBeanRemote, TestSessionBeanLocal {
}
```

- When the stateless bean is created, an annotation `@Stateless` is added to mark it as a stateless bean
- The local and remote interfaces are created and the bean implements them automatically

# Example: MDB generated by wizard

```
MessageBean.java X
package com.test.mdb;

import javax.ejb.ActivationConfigProperty;
import javax.ejb.MessageDriven;
import javax.jms.Message;
import javax.jms.MessageListener;

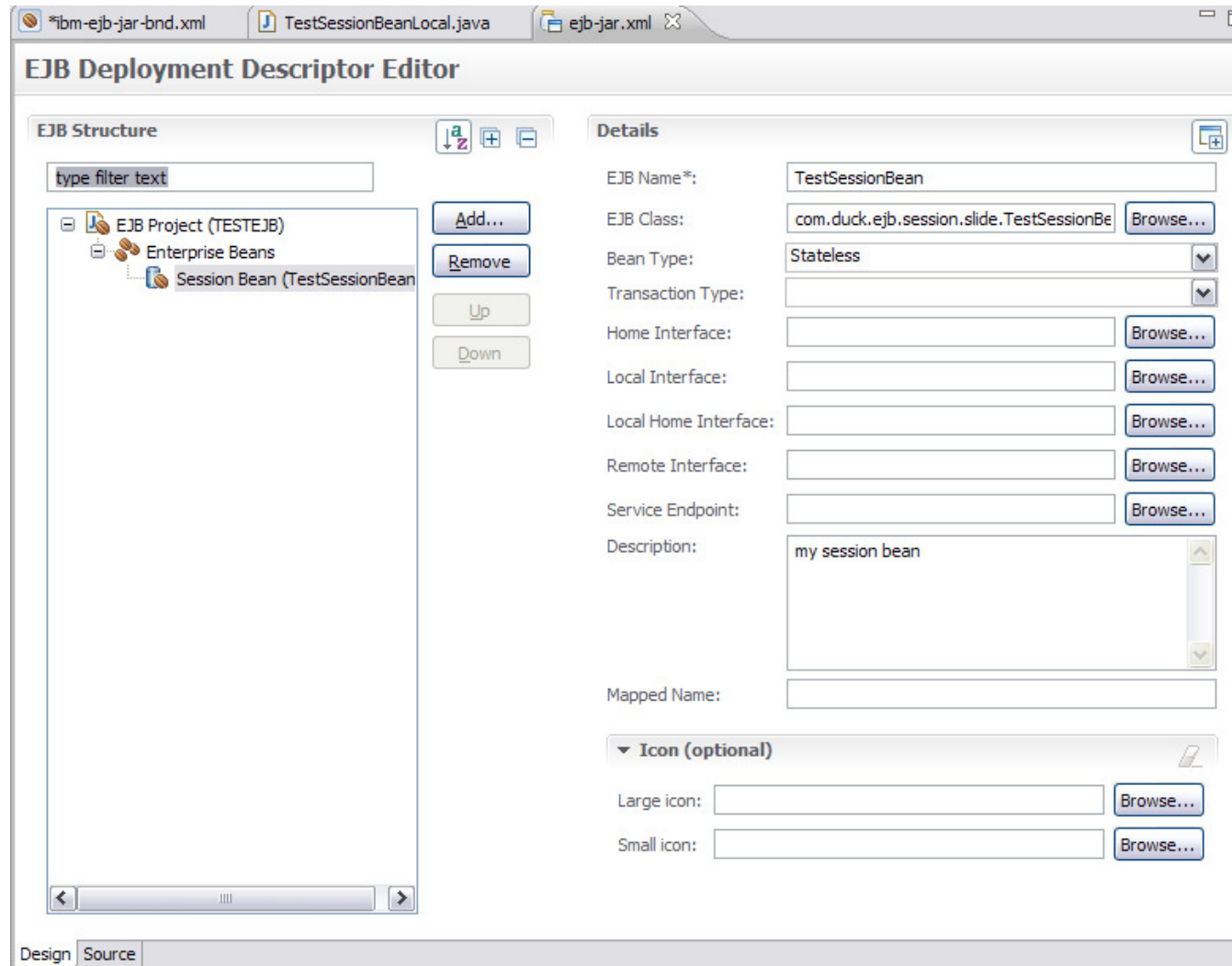
/**
 * Message-Driven Bean implementation class for: MessageBean
 *
 */
@MessageDriven(
    activationConfig = { @ActivationConfigProperty(
        propertyName = "destinationType", propertyValue = "javax.jms.Topic"
    ) })
public class MessageBean implements MessageListener {

    /**
     * @see MessageListener#onMessage(Message)
     */
    public void onMessage(Message message) {
        // TODO Auto-generated method stub
    }
}
```

## EJB 3.0 deployment descriptor

- The EJB deployment descriptor is now optional with EJB 3.0
- The deployment descriptor is only generated into the project if the option to generate a descriptor was selected during the creation of the EJB 3.0 project
  - ▶ As an alternative you may right click on the EJB module and select Java EE > Generate Deployment Descriptor Stub
- The `ejb-jar.xml` file is available to be edited with the deployment descriptor editor
  - ▶ This editor allows developers to add, remove, and modify EJB descriptions and properties

# Example: deployment descriptor



## Entity Beans

- Entity Bean support has changed for EJB 3.0.
  - ▶ Entity Beans are based on the EJB 2.x specification
  - ▶ WAS currently does not support 2.x Entity beans in a EJB 3.0 Jar
- The Java Persistence API (JPA) was introduced with the EJB 3.0 specification as an option to 2.x Entity beans
  - ▶ JPA provides a simplified model for working with persistent data

## JPA development tools overview

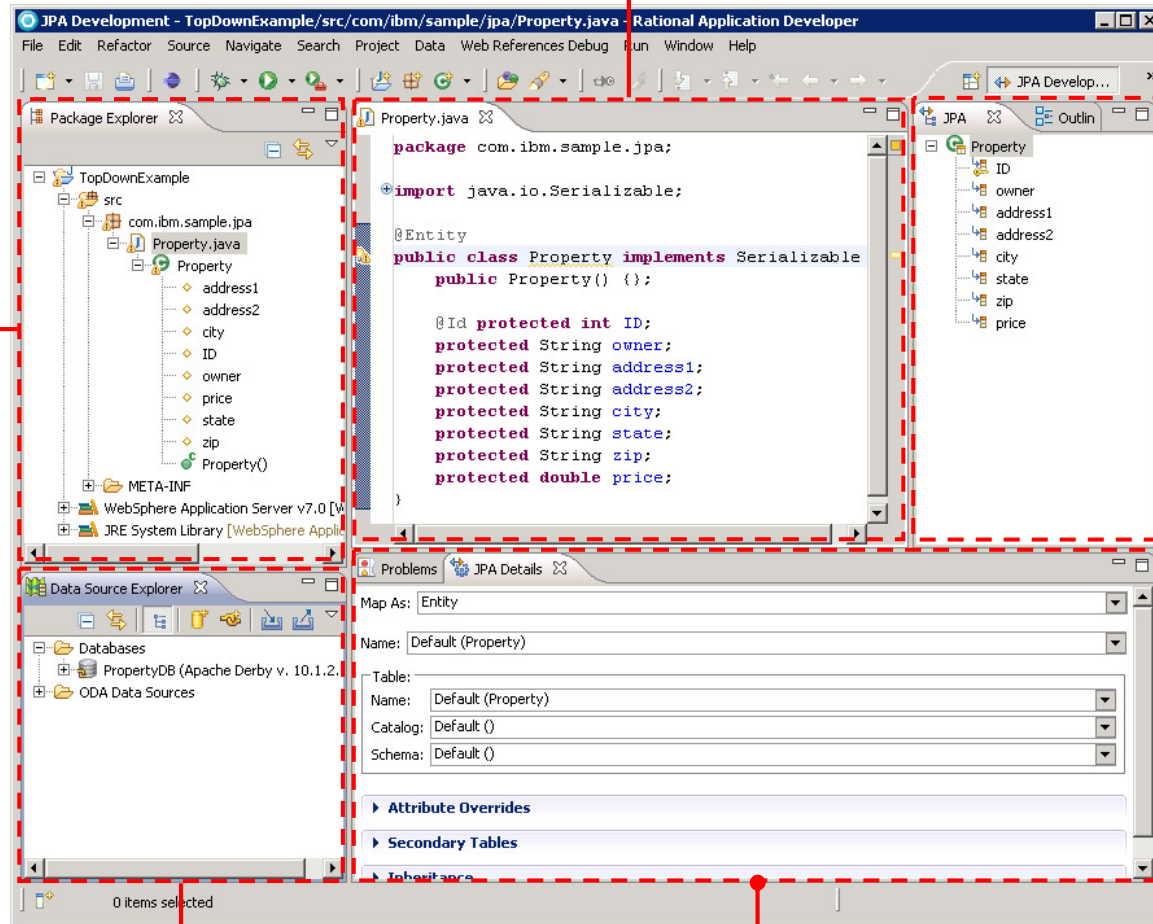
- New JPA projects and JPA views
  - ▶ Add JPA support to existing projects by enabling the JPA facet
- Tools for working with JPA annotations
  - ▶ Java editor provides validation, content assist and quick fixes for JPA annotations
  - ▶ Modify annotation properties in JPA details view or annotations view
- JPA XML editors (persistence.xml, orm.xml)
- Wizards for creating and initializing object-relational mappings
  - ▶ Create tables from entity classes (top-down)
  - ▶ Create new entity beans from existing database tables (bottom-up)

# JPA Development perspective

Display Java classes, work with deployment descriptors in the main panel

The **Package Explorer** view shows project structure

Work with data connections in the **Data Source Explorer** view

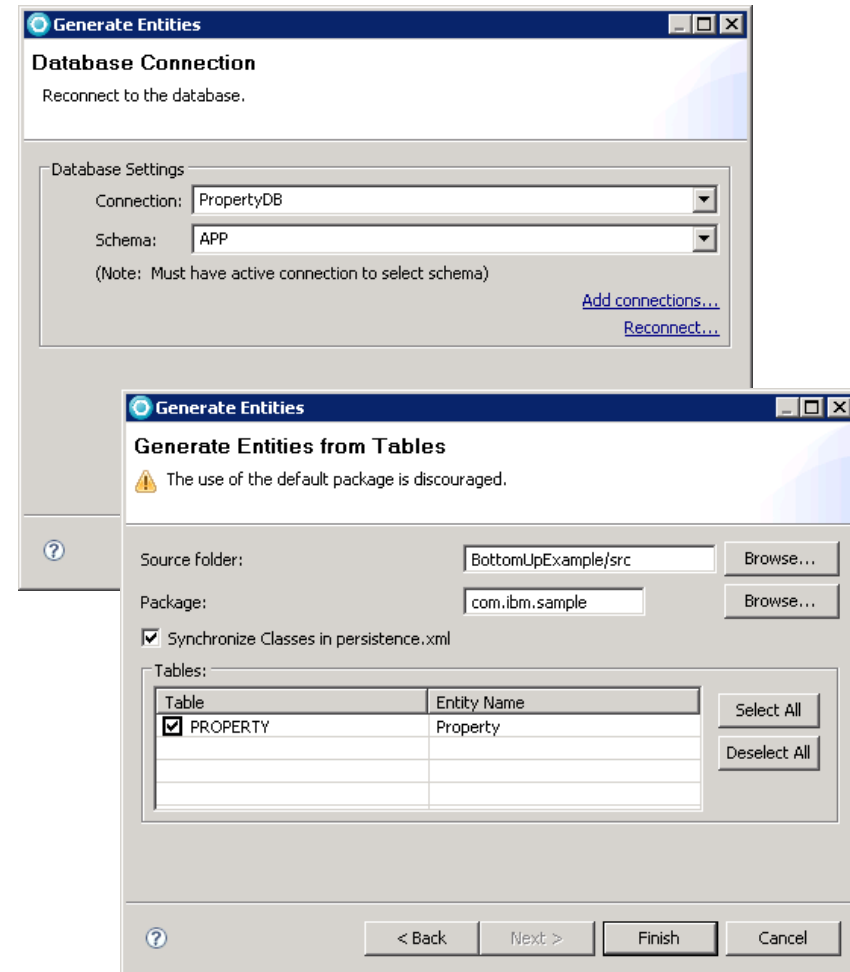


**JPA Structure** view provides a tree display of the data in a JPA class

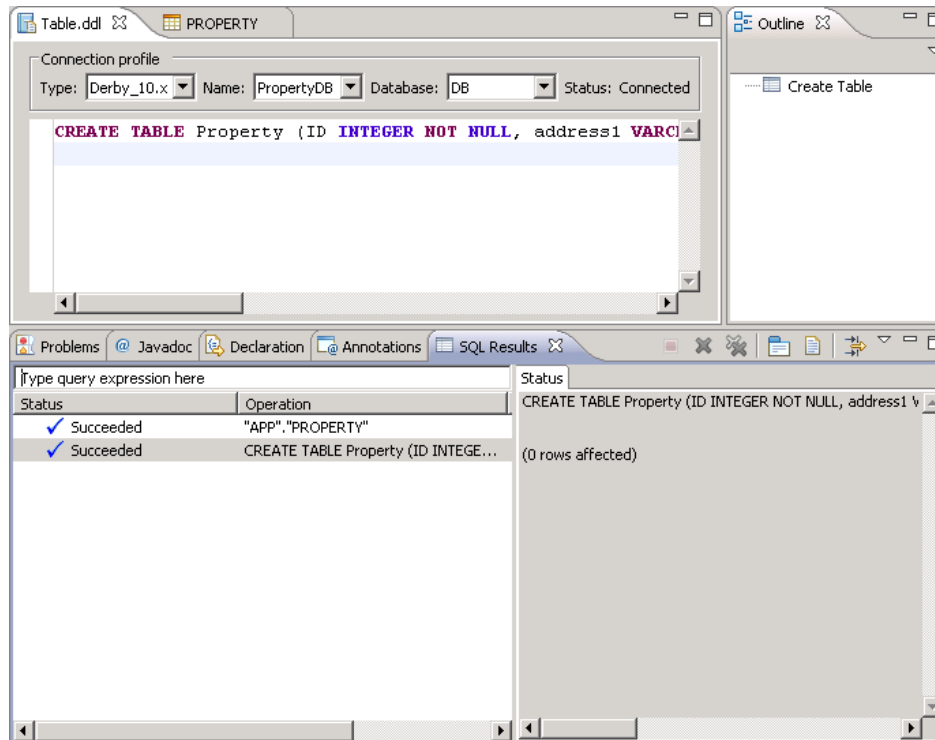
Highlight a JPA construct in the JPA Structure view to see configuration details in the **JPA Details** view

# Creating entities from database tables

- Configure a database connection in the project
  - ▶ Database needs to contain a table to use for generating an entity
- Right-click on the project in the Project Explorer and select **JPA Tools > Generate Entities**
- Choose the appropriate connection and schema in the **Database Connection** panel
- Select the table to use to create an entity
- Tools automatically generate a class based on the table structure
  - ▶ Includes data members and accessor methods



# Creating a database table from an entity

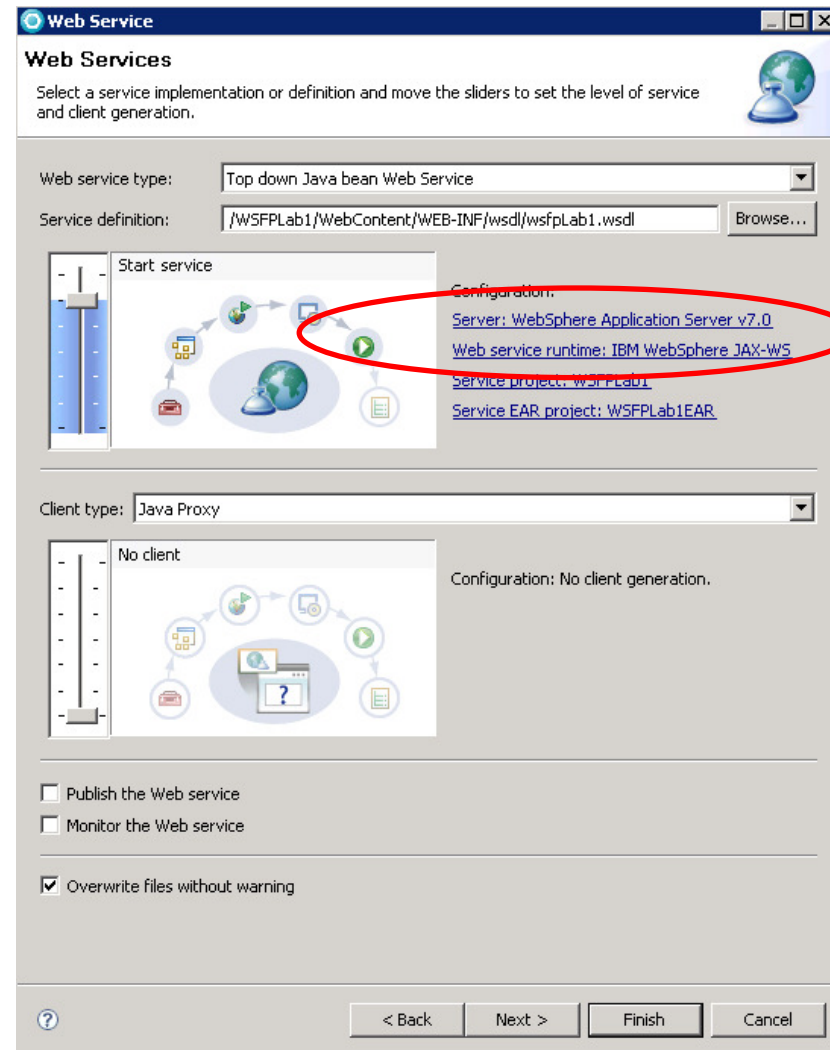


- Use JPA entities to create database tables top-down
  - ▶ In the Package Explorer view, right-click the JPA project and select **JPA Tools > Generate DDL...**
  - ▶ The JPA tools create a file named **Table.ddl** in the **META-INF** directory of your JPA project
  - ▶ Double-click Table.ddl to open it
  - ▶ Configure the connection properties using the dropdown menus at the top of the editor
  - ▶ Right-click in the editor and select **Execute All**
  - ▶ Check the SQL Results view to see if the table generation was successful

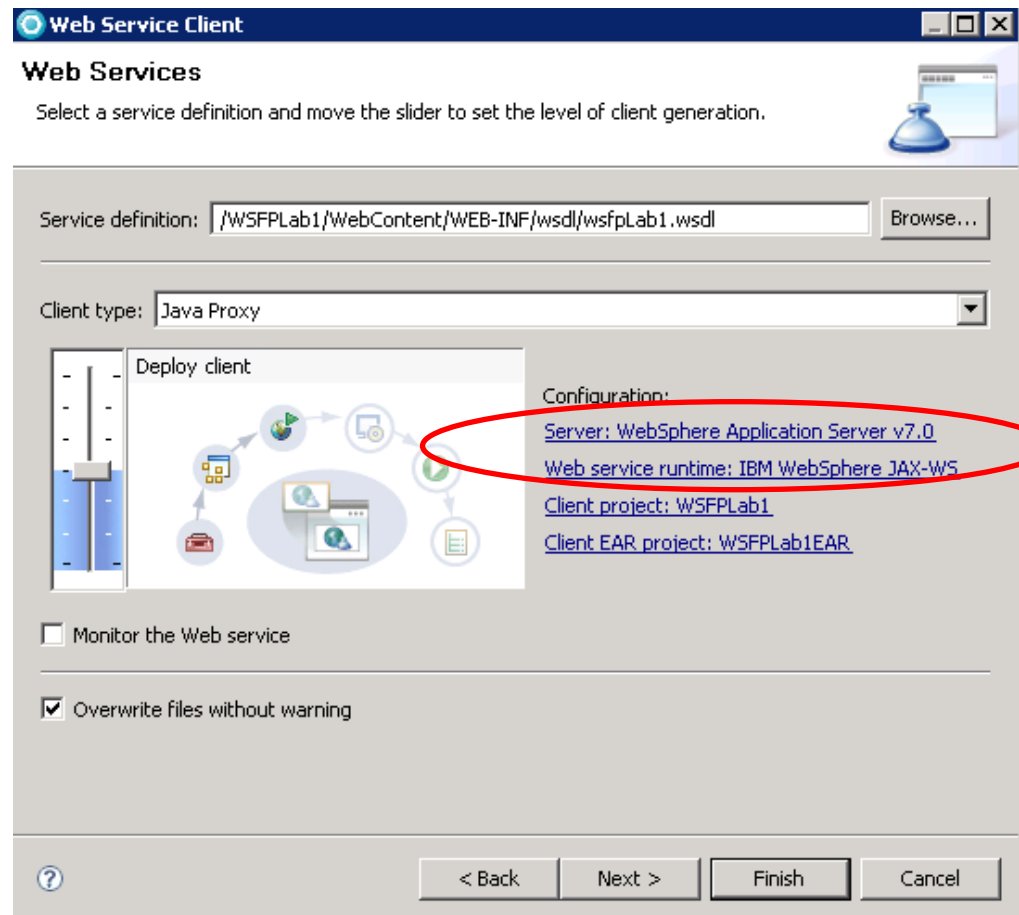
## RAD 7.5: Web services

- JAX-WS development wizards
  - ▶ Top down
  - ▶ Bottom up
  - ▶ Client generation wizards
- Policy Set support

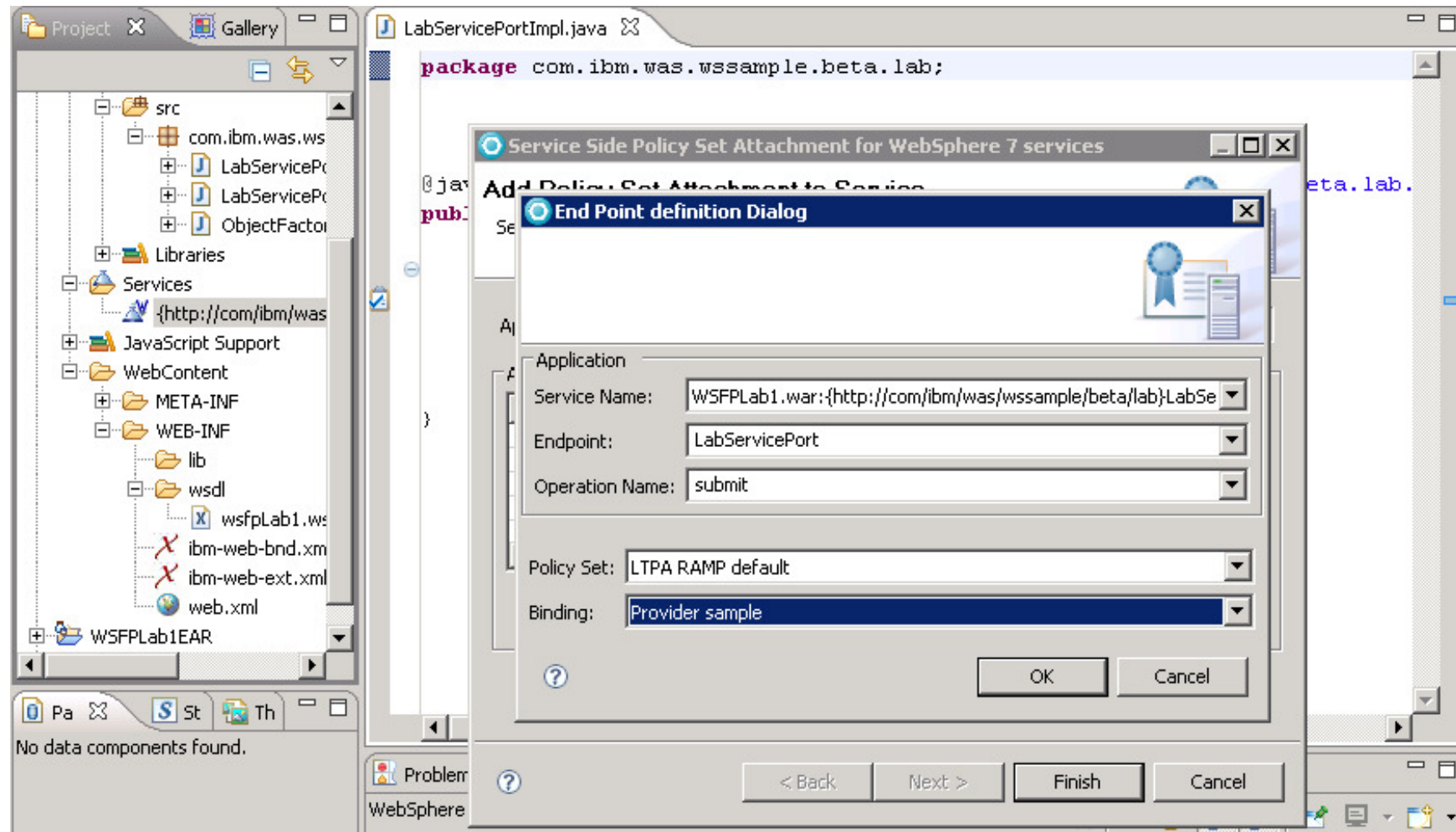
# Web service provider wizard



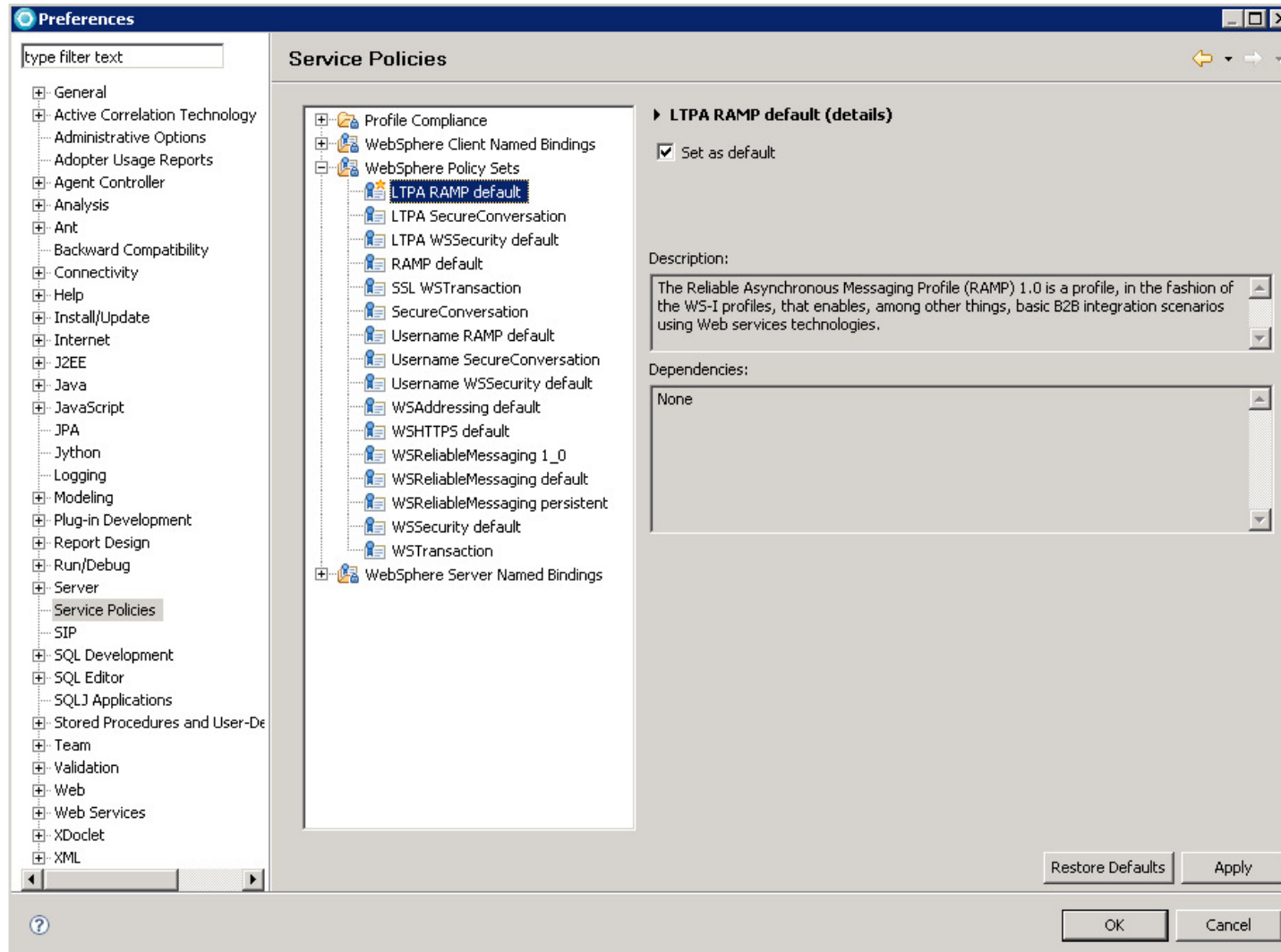
# Web service client wizard



# Policy Set support



# Policy Set support



## WAS v7 Jython Library Support

- WAS v7.0 server added built-in Jython script libraries
- Only available in WAS v7.0 server.
- Jython library support includes:
  - ▶ New preference to the Jython preference page
  - ▶ Content-assist/code completion to the Jython editor for the new script libraries.

# Jython Library -Content assist example

The screenshot shows an IDE window titled '\*NewScript.py'. The code in the editor is:

```
#  
# TODO: enter JYTHON code and save  
#  
wsadminNode = AdminControl.getNode()  
print "Node name" + wsadminNode
```

Below the code, a content assist popup is displayed. The popup has a list of libraries on the left and a detailed description on the right. The selected library is **AdminServerManagement**.

**Admin**

- AdminApplication
- AdminBLA
- AdminJ2C
- AdminJMS
- AdminJDBC
- AdminResources
- AdminServerManagement**
- AdminClusterManagement
- AdminNodeManagement
- AdminNodeGroupManagement
- AdminAuthorizations

**AdminServerManagement**

(This script object is only supported on WebSphere Application Server V7)  
The AdminServerManagement script library provides script procedures that configure, administer, and query server settings.  
The AdminServerManagement script library provides the following script procedures.  
To display detailed information about each script procedure, use the help command for  
the AdminServerManagement script library, specifying the name of the script of

## RAD References

- What's New in RAD 7.5 Technote

- ▶ <http://www-01.ibm.com/support/docview.wss?rs=2042&uid=swg27014208>

- RAD 7.5 InfoCenter

- ▶ <http://publib.boulder.ibm.com/infocenter/radhelp/v7r5/index.jsp>

धन्यवाद  
Hindi

多謝  
Traditional Chinese

ขอบคุณ  
Thai

Спасибо  
Russian

Gracias  
Spanish

---

Thank You  
English

---

شكراً  
Arabic

Merci  
French

Obrigado  
Brazilian Portuguese

Grazie  
Italian

多谢  
Simplified Chinese

Danke  
German

நன்றி  
Tamil

ありがとうございました  
Japanese

감사합니다  
Korean



# Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM                      WebSphere

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.