

Borland®

Excellence Endures

JBuilder 9 EJB 开发新功能

使用 Borland JBuilder 9 Enterprise

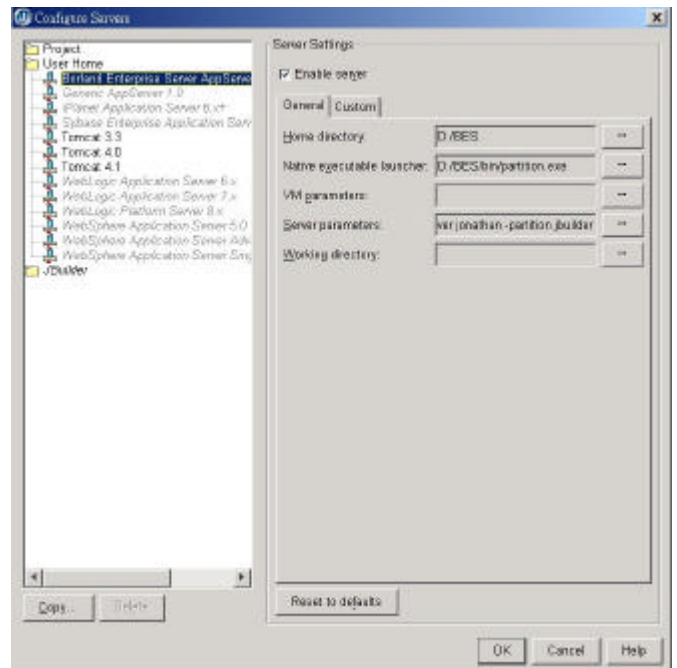
Introduction

本篇文章将针对如何通过新一代的 JBuilder 9 对 EJB 开发所新增的支持功能，如 Session Façade，Data Transfer Object(Value Object)，及 Struts Client 作一介绍，经由新一代的 JBuilder 来开发上述技术的组件，可以说是非常简单，让我们又再一次体会到 JBuilder 的强大功能；同时本文也将就如何在 JBuilder 环境下，开发 EJB 组件作一简单介绍，希望让未接触过 EJB 的读者也能尝试练习一下。

JBuilder 应用服务器设置

对于如何在 JBuilder 中开发 J2EE 应用程序，首先第一步要做的是即是设置好你所要使用的应用服务器，由于 JBuilder 9 Enterprise 版本中已附有 Borland 的应用服务器--Borland Enterprise Server 的开发授权，所以本文中为方便起见，将以 Borland Enterprise Server 作为我们的应用服务器平台。对于使用其它应用服务器的用户，事实上，只需将设置服务器的部份，加以置换即可，本文的主要内容，还是可以套用到其它服务器上的 EJB 开发。

首先通过主选菜单上 Tools|Configure Servers... 的选项，选取你所要使用的应用服务器，如下图所示：



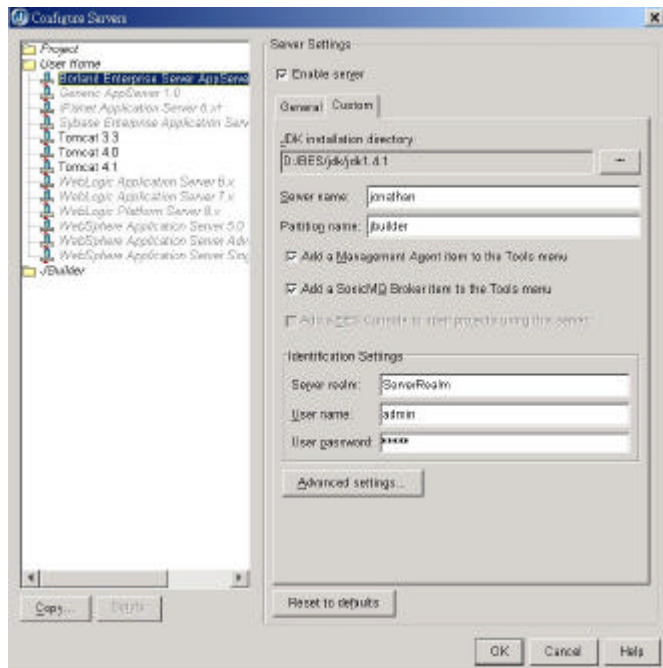
首先点击你所要使用的应用服务器，如“Borland Enterprise Server”，接着勾选“Enable server”，记得设置你

目录

Introduction	1
JBuilder 应用服务器设置	1
开发 EJB Project	2
生成 EJB 模块	2
开发 Entity Bean	3
使用 JBuilder DTO/Façade 精灵	4
创建 EAR 档案	7
测试 EJB 项目	8

white paper

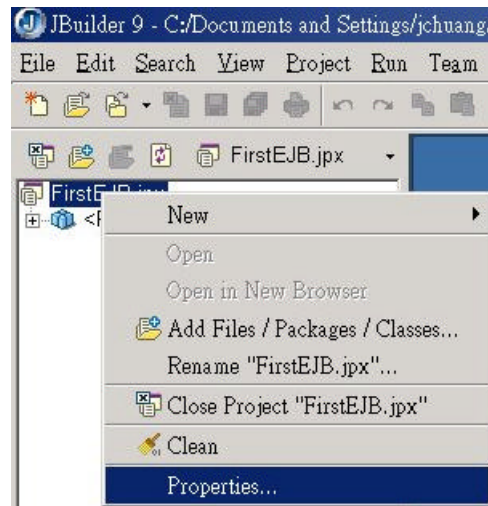
的应用服务器所安装的路径，通过“General”项目下的 Home directory 可加以设置，其它如 VM，Server parameters，Working directory 等项目，则是针对在 JBuilder IDE 环境内，如何激活应用服务器同时将项目内开发的 EJB 组件部署至何处的设置，在这里我们可以使用 JBuilder 所预设的值即可。接下来，我们可切换至“Custom”页面，如下图所示：



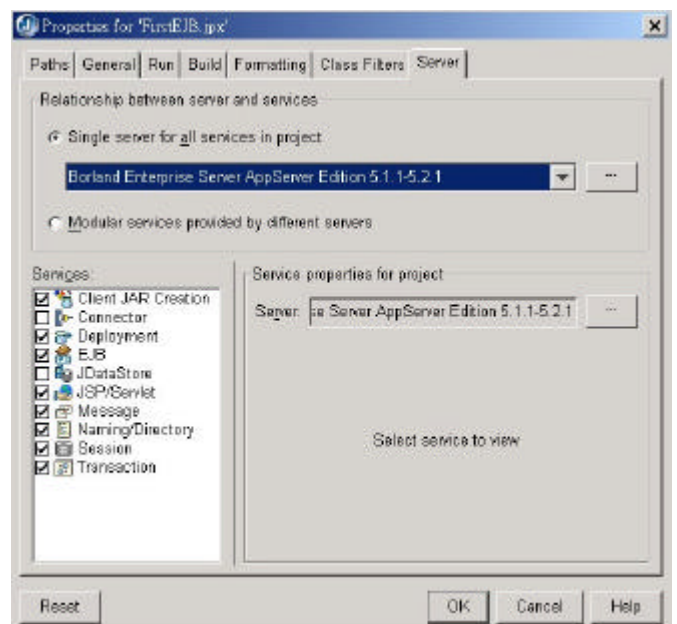
用户可针对 Borland Enterprise Server 所要使用的 JDK 加以设置，同时可选定是否要将“Management Agent”及“SonicMQ Broker”两个项目加入 JBuilder 的 Tools 选单中。最下方所显示的“Identification Settings”则是当我们要连接至应用服务器时所需要的 Username/Password。做到这里，我们已大致完成了应用服务器的设置。接下来，我们就可以开始开发我们的项目。

开发 EJB Project

首先，通过 File 选单下的 New Project...，产生我们要开发的项目，给它一个名称，好比叫“FirstEJB”，接下来，记得通过鼠标右键点击选择 Project Pane 中的“FirstEJB.jpx”，并选取“Properties...”如下图，



接下来，JBuilder 会出现下面的窗口，

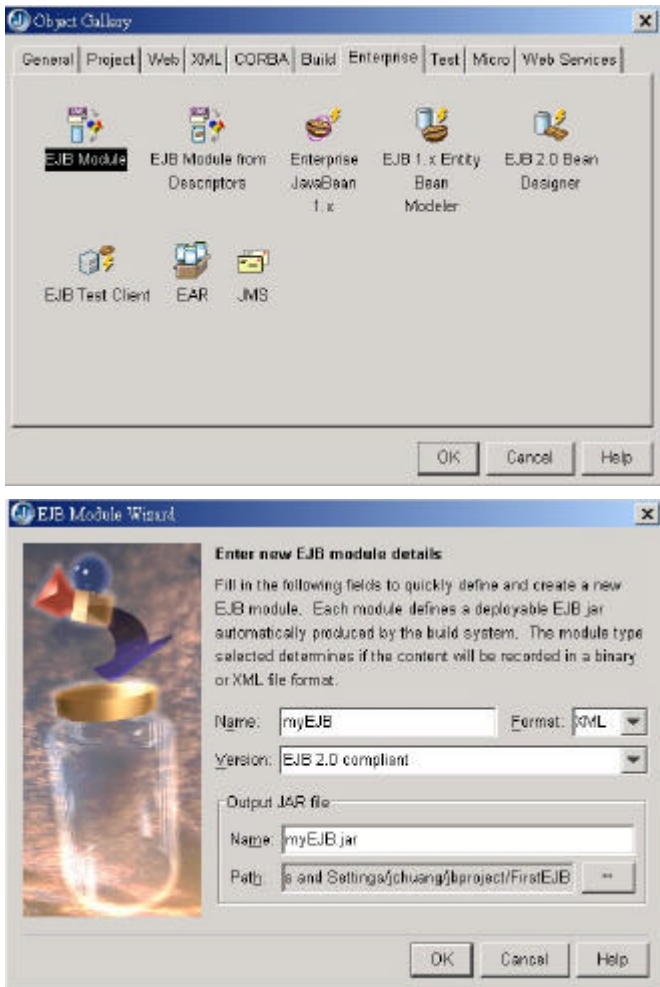


请点击选择 Server 页面，并选取 Single server for all services in project，同时通过其下的选单---选取我们所要使用的应用服务器---Borland Enterprise Server，这个步骤不要忘了，否则你将无法使用 JBuilder 所提供的各项 EJB 精灵。设置完毕之后，我们就可以开始开发了。

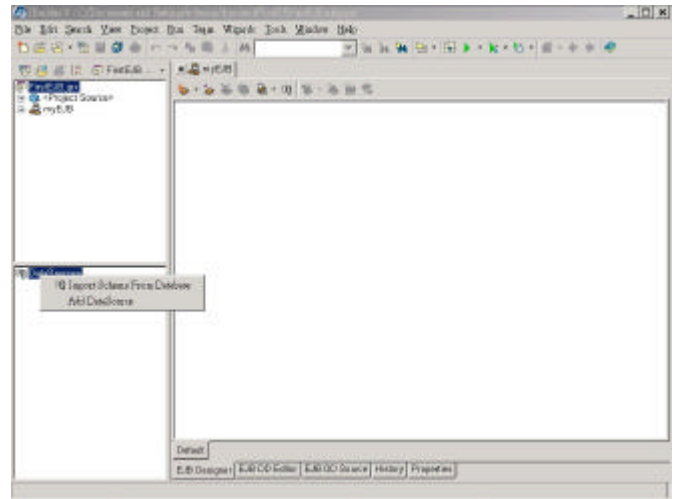
生成 EJB 模块

我们第一步要做的是生成项目所需要的“EJB Module”，请点击选择“File|New...”调出 JBuilder 的 Object Gallery，并点击选择至“Enterprise”页面，第一个精灵即为我们要调用的 EJB 模块(module)精灵，请双击这个项目，接着我们可通过下面窗口来加以设置，



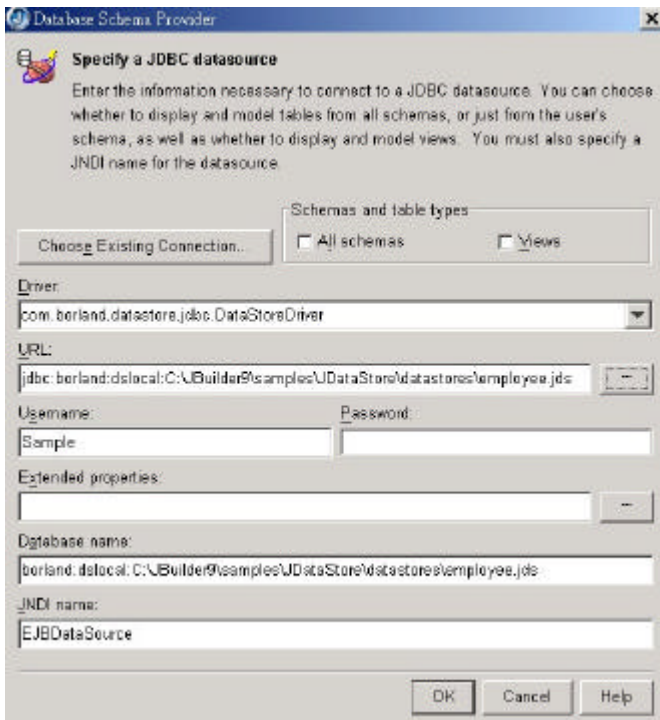


我们可以给它一个名称，如“myEJB”，这个名称即是之后，JBuilder 将为你项目所产生的 JAR 文件名称，也是我们将用来部署至应用服务器的压缩档（.jar 文件），接下来，请设置我们开发所要使用的 EJB 版本，好比“EJB 2.0 compliant”，接下来请按下“OK”按键。之后，JBuilder 的环境将出现 EJB 设计家（EJB Designer）的接口，如下图所示：

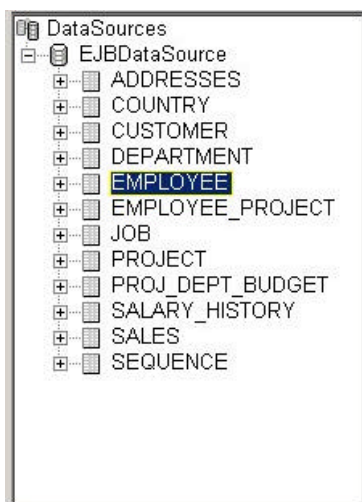


开发 Entity Bean

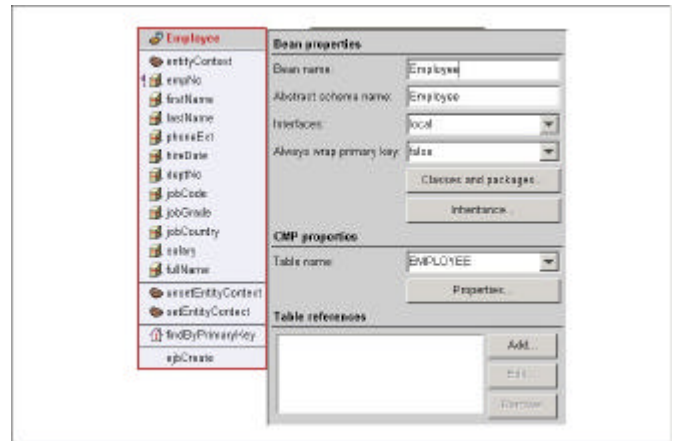
首先我们要开发一个 Entity Bean，请按右键点击选择 JBuilder 左下角的“DataSources”，由跳出的选单中，选取“Import Schema From Database”，此种做法是由已设计好的数据库中取得 Entity Bean 所要对应的 Table 资料，JBuilder 亦支持由开发人员通过“Add DataSource”的第二种做法来手动生成数据库里的 Table。在这里，为节省篇幅，我们以存取一个 JBuilder 环境中内含的 JDataStore 数据库为例子，来写我们自己的 Entity Bean。接下来，在 Database Schema Provider 的窗口内请设置我们所要使用的 JDBC Driver 为 “com.Borland.datastore.jdbc.DataStoreDriver”，同时将其 URL 指向 [JBuilderHomeDirectory]\samples\JdataStore\datastores\employee.jds” 档案，Username 请设为“Sample”，Password 为空，最后请将 JNDI name 设为如“EJBDataSource”。接着就可按下 OK 按键结束这个窗口



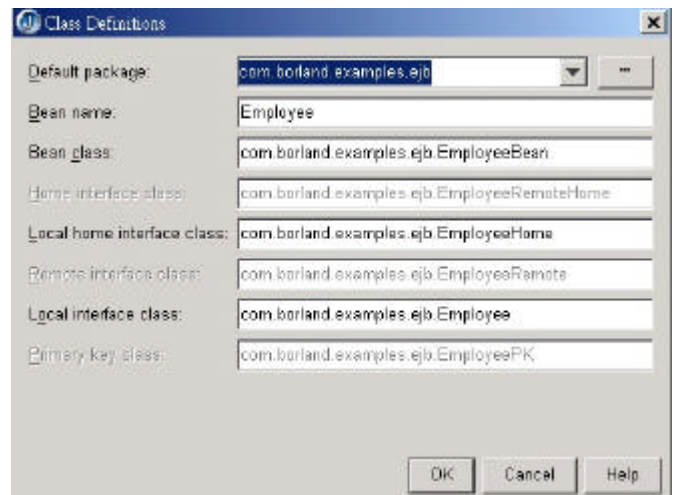
。结束后，如果可以成功连接上数据库，我们应可在 DataSource 内看到如以下的画面：



请按右键选取"EMPLOYEE" Table，通过跳出的选单中选取"Create CMP 2.0 Entity Bean"项目，我们便可在 EJB 设计家中看到我们新产生的 Entity Bean 了，用户可按鼠标点击选择新产生的 Bean 来加以修改其属性，如下图：



例如，我们可以将 Employee Entity Bean package 加以更换，只须按下 Classes and packages 按键，然后键入如" com.borland.examples.ejb"

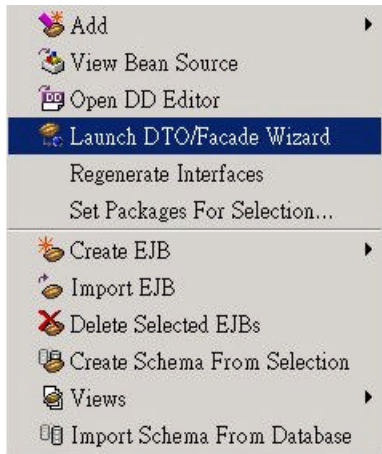


做到这里，我们可以略微检查一下 JBuilder 为我们所产生的原始程序代码，我们可以按右键点击选择我们新生成的 Employee Entity Bean，而后选取" View Bean Source" 来转换至 JBuilder 的 Source Code Editor 来查看其程序代码。

使用 JBuilder DTO/Façade 精灵

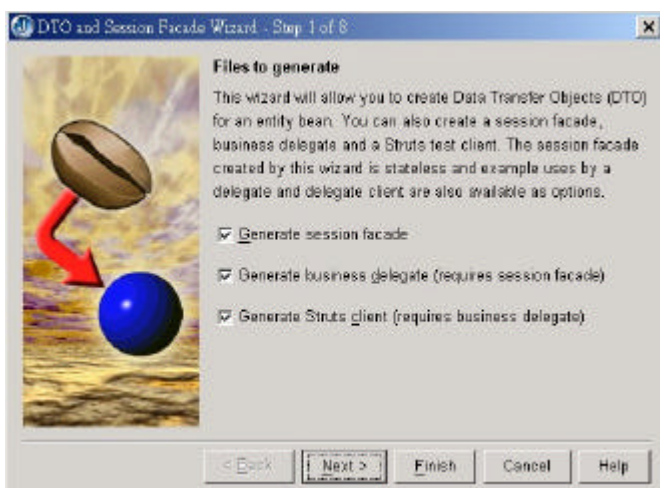
在检查完后，让我们继续我们的开发，请再一次按右键点击选择我们的 Employee Entity Bean，同时通过选单选取" Launch DTO/Façade Wizard" 选项，如下图：





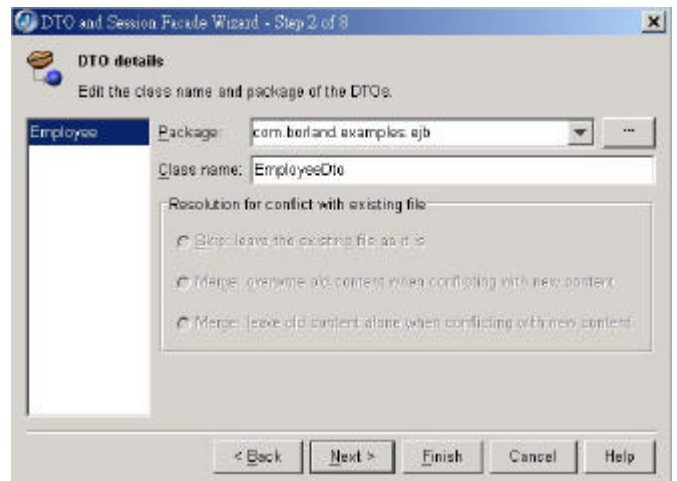
这个精灵必须以上述的方式来调出，目前 JBuilder 并没有在别的地方有提供上述的连接，此一精灵的功能为 JBuilder 9 所新增的功能，目的即是为帮助开发人员产生一般 J2EE Design Pattern 中所常用到的“Session Faade”及“Data Transfer Object”又称“Value Object”模式，在这里，我们不多花费额外的篇幅来介绍这两种模式，有兴趣的读者，不妨参考相关的书籍介绍。不过要请大家记得的是，此一精灵必须是在 EJB 模块内有已定义好的 Entity Bean 下，才可加以调出，因为它需要有 Entity Bean 的定义，作为它的输入。另外在这里，我们只用一个 Entity Bean 作为例子，事实上，如果你的 Entity Bean 和其它的 Entity Bean 之间有关联的话，DTO/Facade 精灵也可以帮你反应出来，同时产生相对类别及程序代码。

接下来，就让我们来看一下这个精灵，有那些步骤：

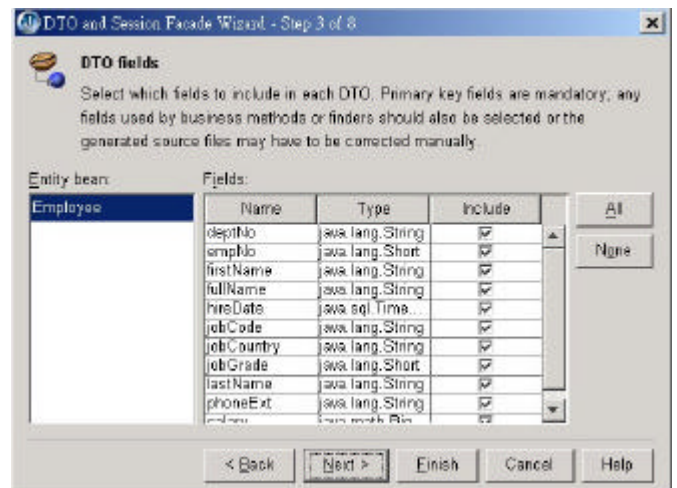


在第一个步骤中，有三个选项可供选择，同时在选项上方也有一简单说明，第一个选项为“Generate Session facade”即帮我们产生 Session Faade 模式所需之程序代

码，第二个选项为“Generate Business delegate”，只是前提是要有 Session Faade 的 Session Bean 存在，通过此选项可帮我们产生一额外类别称为 Business Delegate，我们可通过这个类别来帮我们存取 Session Faade 类别，第三个选项则是可以帮我们自动产生一个基于 Struts Framework 的客户存取程序。在这里我们将三个选项一并加以勾选。

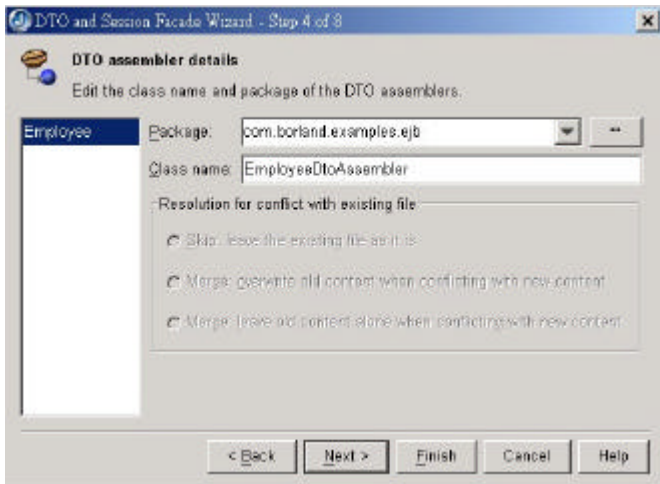


在第二个步骤中，我们可指明所要产生的 Data Transfer Object 的类别名称及其 package 名称。好比这里我们保留了类别名称，不过将 package 改为“com.borland.examples.ejb”。

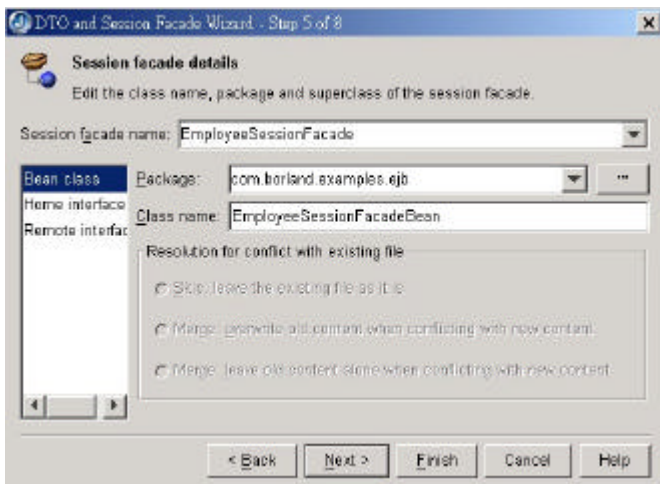


在第三个步骤中，我们可设置 DTO 对象所要对应的数据域位，这里我们以 Domain Value Object 的方式设计，所以请将所有的字段都加以勾选。

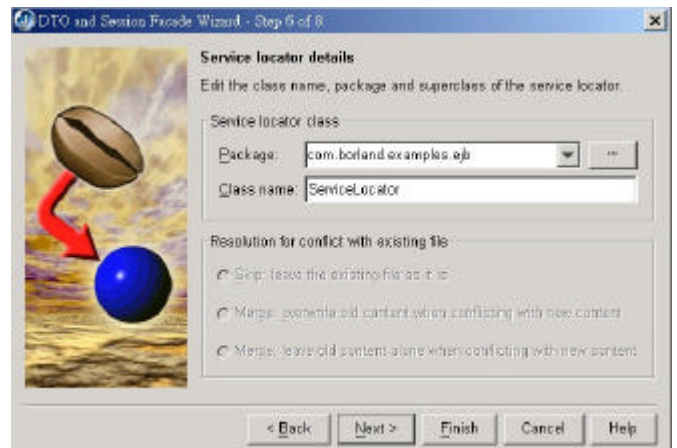




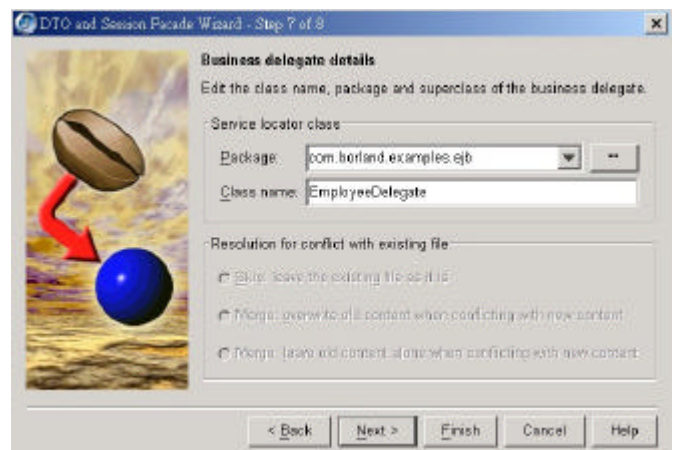
在接下来的步骤中，主要是要帮我们产生另一额外辅助类别，此类别的功用即是类似于 Value Object Factory 的功能，是要帮我们产生 Data Transfer Object 实例之用，一样的，我们可针对其 Package 及 Class name 加以变更。这里我们一样保留了类别名称，不过将 package 改为 "com.borland.examples.ejb"。



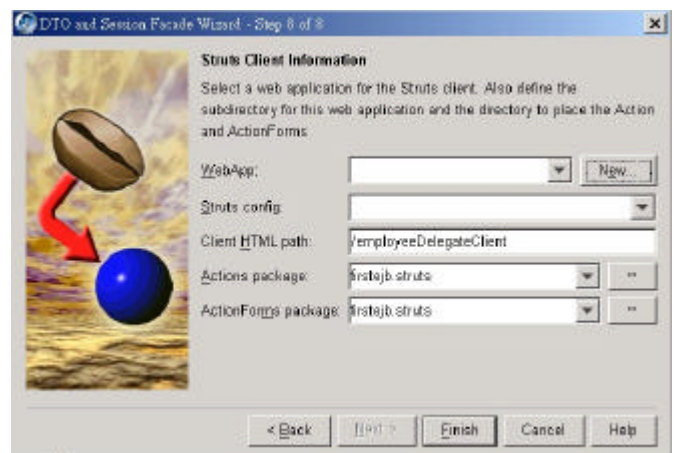
在第五个步骤中，主要是针对 Session Façade 模式所要产生的 Session Bean 其相关信息加以设置，如类别名称及 package 名称，不过在这里要提醒大家的是，如果要更换 package 名称，则记得点击选择 Home Interface 及 Remote Interface 项目，并将其相对 package 名称一并更换，以免到时产生出来的 EJB Bean 程序代码不在同一路径下。



在第六个步骤中，我们可设置一个所谓的 Service Locator 类别的名称及 Package，此一类别主要为一辅助类别，其所提供的服务为帮我们查找所要调用的 EJB Bean Home reference。

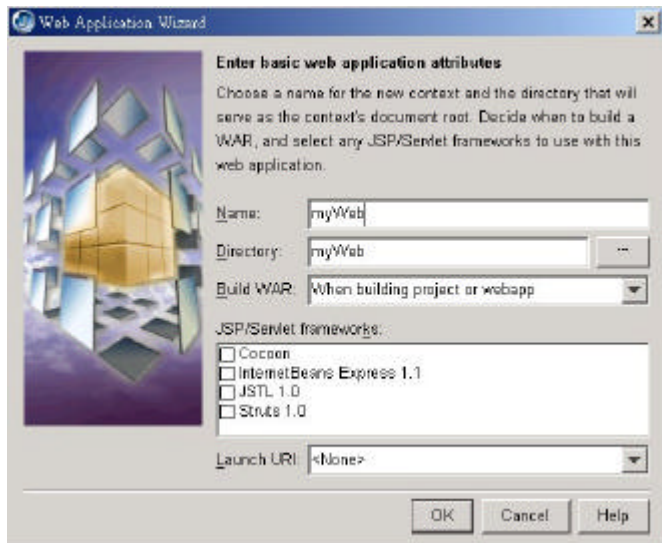


第七个步骤中，所产生的为一 Java 辅助类别，此类别主要的工作即为我们 Session Façade 的代理人，故其方法的内容都是转而调用实际 Session Façade 的相对方法。

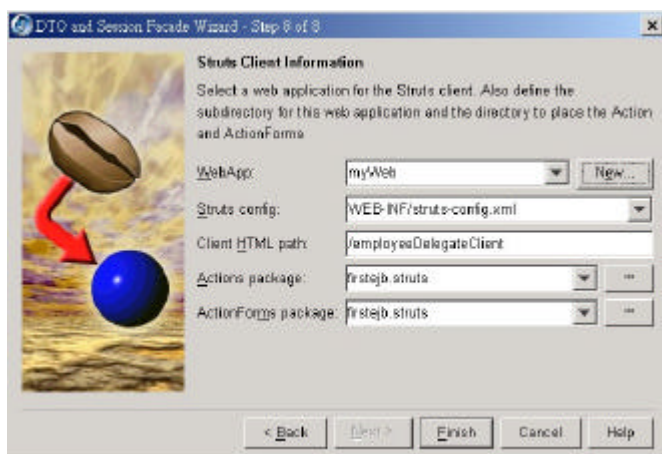


在第八个步骤中，主要是帮我们产生 Struts 的客户端程序，首先我们要先产生一个 Web Application 用来置放我

们的 Struts 组件，请点击选择 WebApp 旁的 New 按钮以调出 JBuilder 内的 Web Application Wizard。如下图：



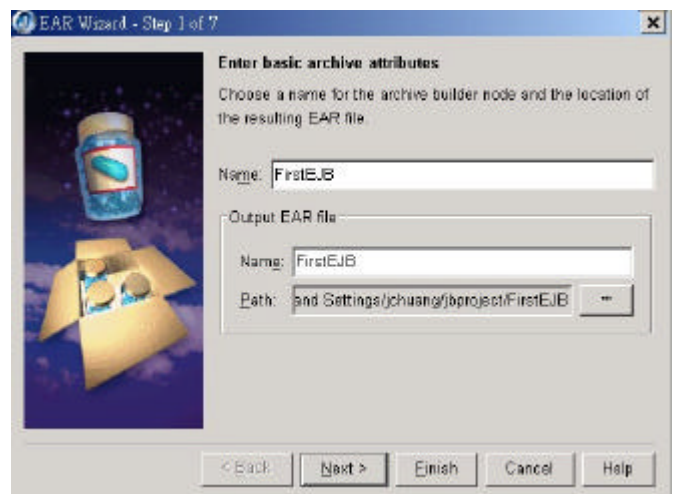
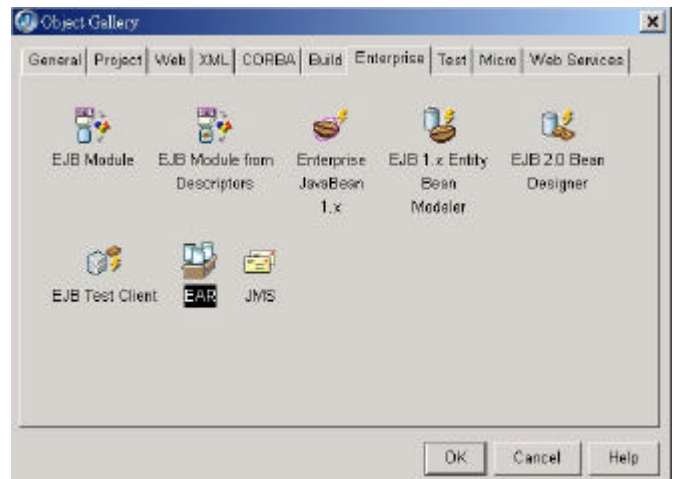
给我们的 Web Application 一个名称，如“myWeb”，之后按“OK”，结束窗口。JBuilder 便返回原先的精灵窗口，如下图：



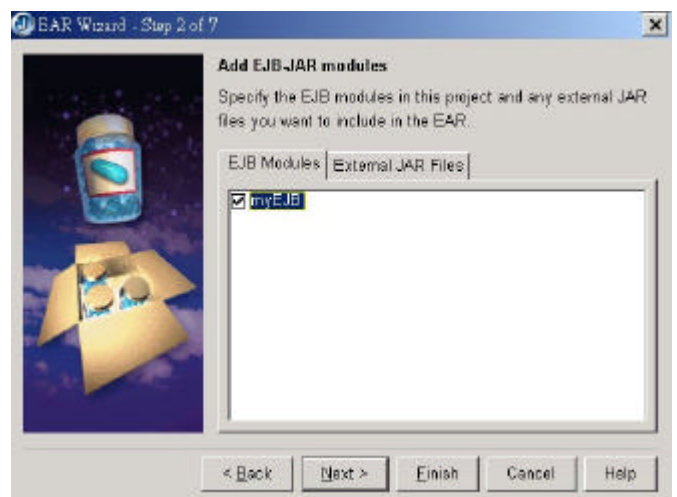
在经过了这么一系列的步骤，我们总算是大功告成，可以按下“Finish”按钮，结束 DTO and Session Façade 精灵窗口。

创建 EAR 档案

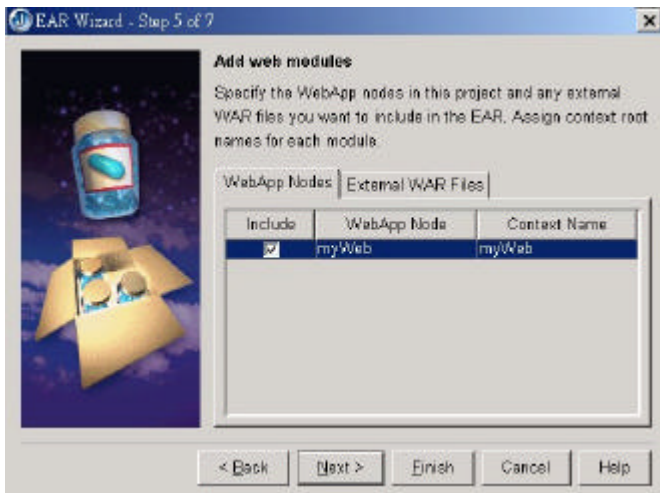
不过我们的工作还没有完成，为了要能部署我们的 EJB 组件，根据 J2EE 标准的要求，我们必须要把 Web Application 及 EJB module 加以打包成所谓的 Enterprise Archive file，也就是所谓的 EAR 档案，在 JBuilder 中，我们可以通过 Object Gallery 中的 EAR 精灵来帮我们完成这样的工作。



在 EAR 精灵中的第一步中，我们可以给它一个名称，如“FirstEJB”，这个名称也就是到时产生出来的 EAR 档名。



在第二个步骤中，请勾选“myEJB”，以便将 EJB 模块包含至 EAR 文件内。



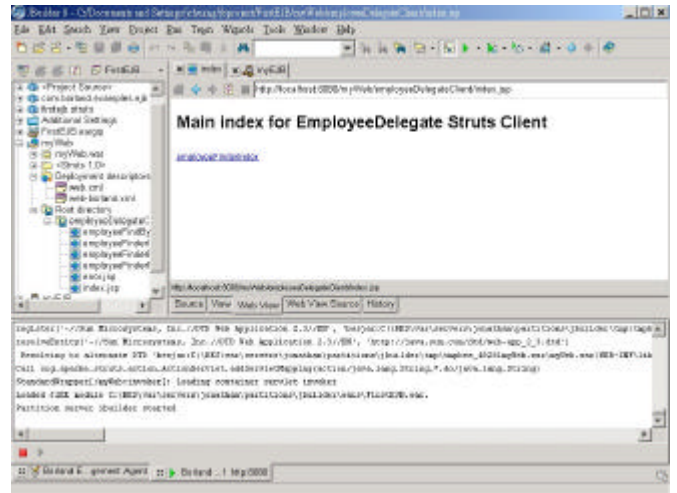
接下来可按” Next”略过数个步骤，直到第五个步骤，请勾选”myWeb”将 Web Application module 加以封包到我们的 EAR 模块内，你可以再往后查看后面几个步骤，不过我们并没有什么需要再改动的，所以可以按下” Finish”结束 EAR 精灵窗口。好了，到这边，我们总算真的完成了我们的 EJB 项目，不过好象一程序代码也没有写，让我们来测试一下，看看我们的 project 可否正常运作。

测试 EJB 项目

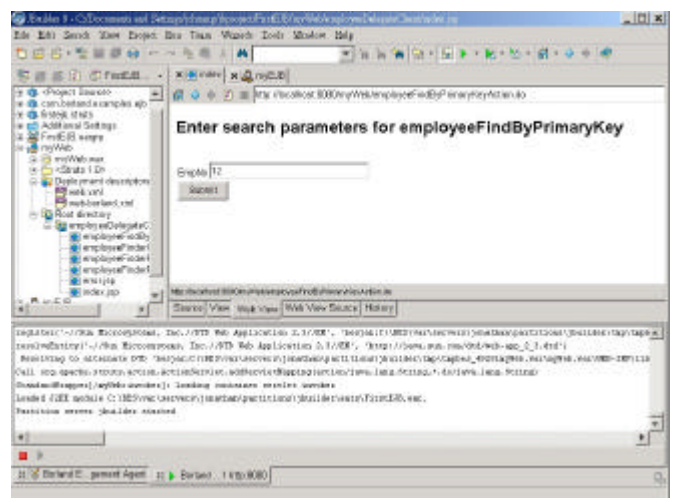
要测试我们的项目，首先请先通过” Project|Make Project...”选项，将我们的项目内的档案加以编辑。如果有任何错误情况，请根据错误讯息，加以修改，不过因为我们并没有撰写任何程序代码，所以即使有错误，多半是 package 名称所造成的路径错误，读者应能轻易发现。

在编辑成功之后，接下来就让我们在 JBuilder 中尝试执行我们的 EJB 项目吧！

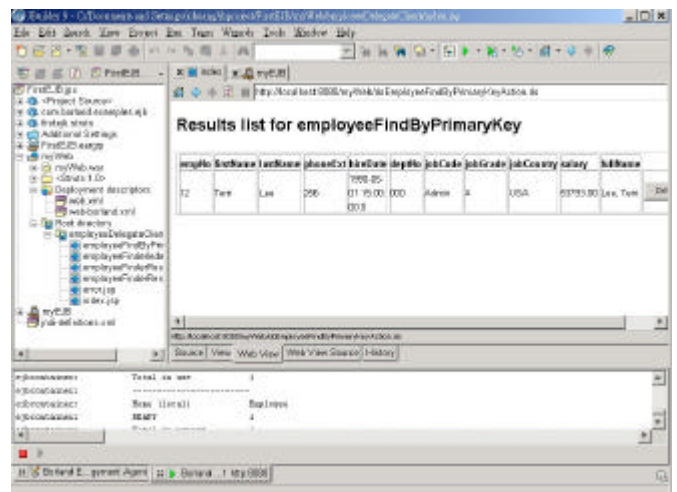
首先，通过” Tools|Borland Enterprise Server Management Agent”选项，激活我们的 Naming Service Agent。接下来，我们可直接按右键点击选择 Project Pane 的 FirstEJB.eargrp 项目，然后选择” Run using defaults”选项，激活我们的 Borland Enterprise Server。等到 Server 激活后，接下来，我们可以展开指到”myWeb\Root directory\employeeDelegateClient 下的 index.jsp，一样按右键点击选择 index.jsp，然后选取”Web Run using defaults”项目，成功的话，读者应可看到如下图的执行情形：



直接点击选择连接，则我们可以看到下面的情形，



请输入一个数字，如” 12”，然后按下 Submit 按键，成功的话，应该是下列的情形：



读者也可以尝试按下页面上的 Detail 按键，看看它的效果如何。最后提醒大家，可以再去检查一下整个项目内容，看看 JBuilder 到底帮我们完成了多少工作。通过了

JBUILDER 9 新增的功能，不仅帮我们产生了相关的程序代码，更帮我们设计了一个可执行的企业级 Java 应用程序。

[全文完](#)

white paper

white paper