

Oracle ERP最佳技术实践
E-BUSINESS SUITE
XX100 –应用技术
从JSP到FOP打印

Author: 黄建华Jianhua.Huang
MSN: huajhua@hotmail.com
Creation Date: February 23, 2006
Last Updated: December 14, 2006
Document Ref: <Document Reference Number>
Version: DRAFT 1A

Approvals:

<Approver 1>

<Approver 2>

Document Control

Change Record

Date	Author	Version	Change Reference
23-Feb-06	Jianhua.Huang	Draft 1a	No Previous Document

Reviewers

Name	Position

Distribution

Copy No.	Name	Location
1	Library Master	Project Library
2		Project Manager
3		
4		

Note To Holders:

If you receive an electronic copy of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a hard copy of this document, please write your name on the front cover, for document control purposes.

Contents

Document Control	ii
1. Development Configuration开发配置和验证	1
1.1. 了解EBS服务器目录	1
1.2. 简单了解JServ	1
1.3. 配置JDeveloper OA Extension	2
1.4. 先睹为快：一个简单例子.创建Workspace和Project	5
1.5. 先睹为快：一个简单例子.创建java类和jsp页面	7
1.6. 先睹为快：一个简单例子.发布	10
1.7. 小结	11
2. 一个完整的JSP查询例子	12
2.1. 开发准备	12
2.2. 定义DTO类	13
2.3. 定义业务逻辑类、数据访问类	14
2.4. 编写JSP页面	15
2.5. 模式转换：JSP中使用类	17
3. 在Applet中使用Apache FOP	19
3.1. 了解Apache FOP	19
3.2. 编写Applet	19
3.3. 打包、签名Applet	22
3.4. 测试Applet	22
4. 整合Applet和JSP	24
5. Open and Closed Issues for this Deliverable	25
Open Issues	25
Closed Issues	25

1. Development Configuration 开发配置和验证

1.1. 了解EBS服务器目录

\$OA_HTML

\$OA_HTML是所有EBS Web文件所在目录，最常见的是HTML文件、图片文件、JSP文件、Applet对应的Jar文件。

我们开发的jsp等相关文件要用ftp上传到这里；通常我们会再创建一个客户化文件夹，比如cux，方便文件管理。

如何具体确定\$OA_HTML？

用applmgr用户（如gappdev）Telnet登录应用服务器，执行echo \$OA_HTML可以看到具体目录位置，方便在ftp中定位。

\$COMMON_TOP/_pages/_oa_html

第一次运行，Jserv自动将JSP转成Servlet，然后Cache到内存。

首先是转成Java源文件，然后编译成Class文件，都放在\$COMMON_TOP/_pages/_oa_html下，当然如果jsp放在\$OA_HTML的子目录下，这里也有子目录。

一个执行过的jsp文件会对应这里的一个java文件和一个class文件，有时候查错会把这里的java文件拿下来看，因为如果报错，出错行指的是java文件的行，而不是jsp文件的行。

\$JAVA_TOP

JSP中可以引用标准Java类库，也可以引用其他Java类库，客户化开发的类库统一放在\$JAVA_TOP下，按目录组织。

\$OA_MEDIA

\$OA_MEDIA通常存放图片。

1.2. 简单了解JServ

调整autoreload.classes、developer_mode和debug_mode参数

N: \$IAS_ORACLE_HOME/ Apache/Jserv/etc/zone.properties

developer_mode指示JSP运行期是否自动重编译任何在上次执行后被修改了的jsp文件，原理是检查JSP源文件的时间标志。开发环境设为“true”，否则要重启Apache才能使新上载的jsp文件生效；该功能会大大降低性能。

autoreload.classes指示是否自动载入servlet的class文件，不是指jsp中import的class。

debug_mode用来显示调试信息，比如Java堆栈记录。开发环境也建议设为“true”。

正式环境这几个参数应该被设为“false”。

设完重启Apache生效。

参考资料: *Auto-reloading Classes in class folder*

In most cases unless specified otherwise the Tomcat context is set to autoreload classes which have been modified. However following things need to be kept in mind about autoreload -

- 1. Only classes in the folder WEB-INF/classes will be reloaded*
- 2. These classes will be reloaded only if a servlet or a JSP file using them is called. In particular, changes to classes other than the servlet you are requesting do not trigger class reloads -- you will need to restart Tomcat to reflect changes in those classes.*
- 3. Beans do not get reloaded. If you have beans that you call in servlets the beans will not get reloaded unless you restart tomcat. The only other way I figure to get them to reload is to build in functionality to allow them to be called as servlets and then everytime you change the bean call it at least once as a servlet*

参考资料: *developer_mode*

Set this flag to false to instruct the Oracle JSP container to not routinely compare the timestamp of the page implementation class to the timestamp of the .jsp source file when a page is requested. With developer_mode set to true, the Oracle JSP container checks every time to see if the source has been modified since the page implementation class was generated. If that is the case, the JSP container retranslates the page. With developer_mode set to false, the JSP container will check only upon the initial request for the page or application. For subsequent requests, it will simply re-execute the generated page implementation class.

This flag also affects dynamic class reloading for JavaBeans and other support classes called by a JSP page. With developer_mode set to true, the Oracle JSP container checks to see if such classes have been modified since being loaded by the Oracle JSP class loader

Log文件

N: \$IAS_ORACLE_HOME/Apache/Jserv/logs

通常文件名是mod_jserv.log，如果遇到错误，可以查看这里的log文件。

1.3. 配置JDeveloper OA Extension

常见的同类工具

Dreamweaver: 做Web开发（HTML/CSS/ASP/PHP/JSP）非常方便，我常用它来生成HTML代码和CSS代码

JCreator: 做简单的Java开发

Eclipse: 流行的Java开发工具

JBuilder: Borland公司出品的IDE

JDeveloper: Oracle公司出品的IDE

工具选择

推荐直接使用开发OAF的JDeveloper版本: Oracle9i JDeveloper with OA Extension。
为以后做OAF开发打下基础。

下载安装

从Metalink下载Patch 4045639, 解压即可, 位置任意; 不需要安装, 也不需要原版本的Jdeveloper。假定解压到D:\Software\Oracle\JDev9iOAExtension, 后面的章节将以这个目录为基础。

目录	描述
jdevdoc\javadoc\fwk	OA Framework 的 Java 文档
jdevdoc\javadoc\ao1j	AOL/J 的 Java 文档
jdevdoc\javadoc\bc4j	BC4J 的 Java 文档
jdevdoc\javadoc\uix	UIX 的 Java 文档
jdevdoc\toolbox	OA Framework ToolBox 指南/实验文档
jdevdoc\devguide	OA Framework 的开发员手册
jdevbin\	扩展的Oracle 9i JDeveloper可执行和OA Framework类库
jdevhome\	OA Framework ToolBox指南的源文件和开发工作区

设置Windows环境变量JDEV_USER_HOME

N: 我的电脑/右键/属性/高级/环境变量/用户变量的新建

变量名: JDEV_USER_HOME

变量值: D:\Software\Oracle\JDev9iOAExtension\jdevhome\jdev

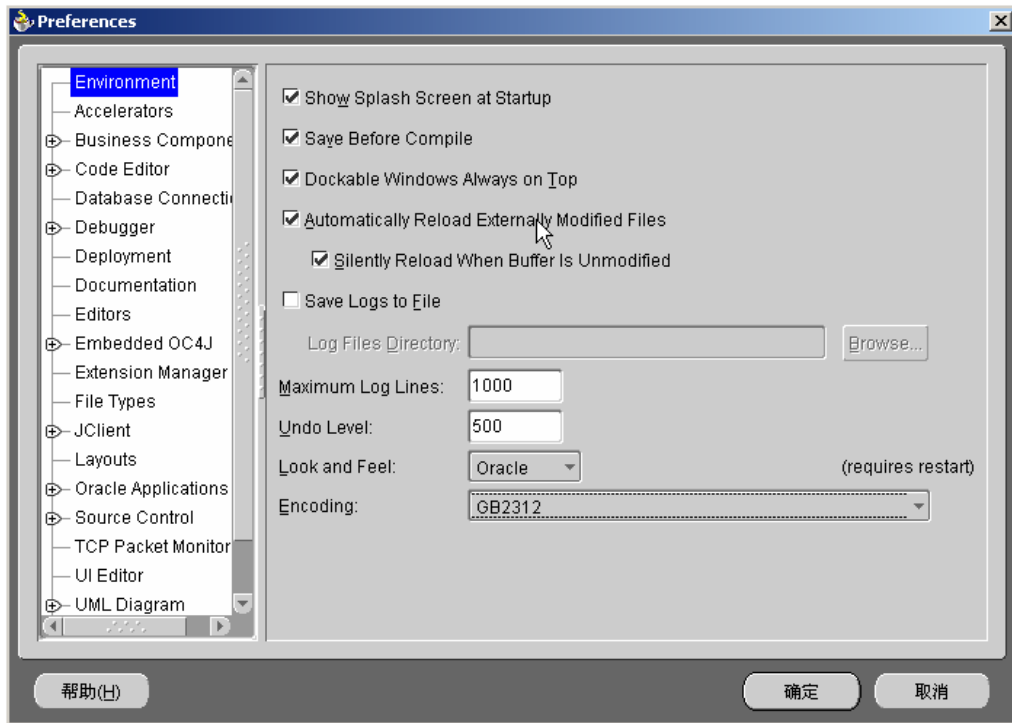
为JDeveloper主程序jdevw.exe创建快捷方式

为了方便快速启动, 为 jdevbin\jdev\bin\jdevw.exe 创建快捷方式到桌面

设置默认字符集

N: Tools/Preferences/Environment

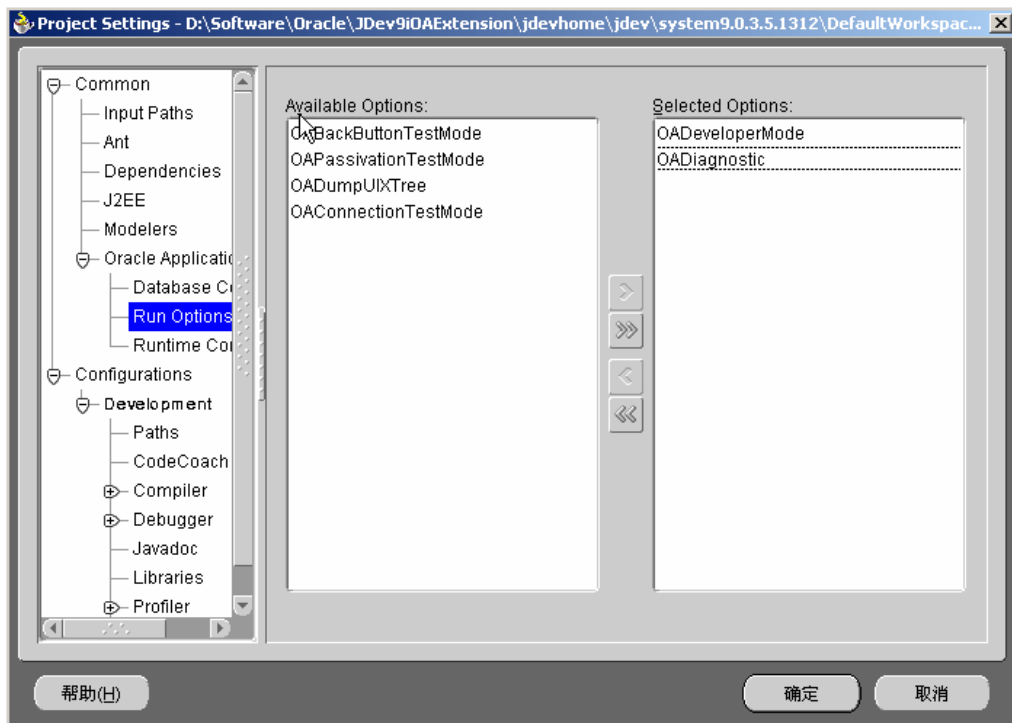
要根据EBS字符集修改; 这里我改成GB2312。我们新建的jsp页面, 头部的charset将默认取这个值, 如<%@ page contentType="text/html;charset=GB2312"%>。



设置默认Project属性：调式信息

N: Project/Default Project Settings.../Common/Oracle Applications/Run Options

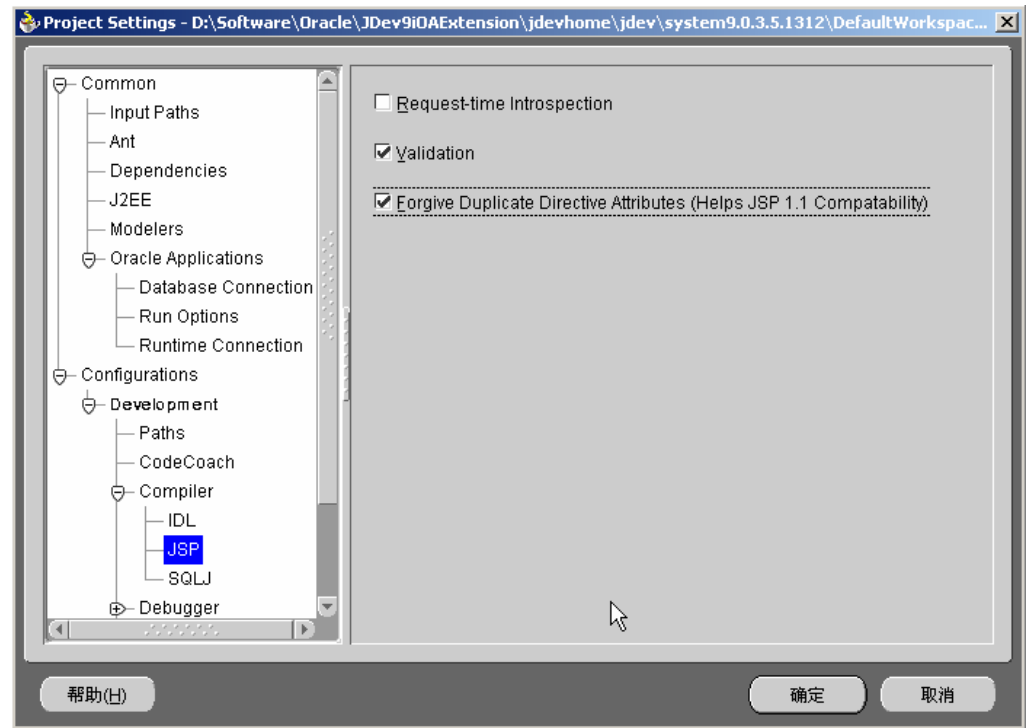
添加OA Diagnostic



设置默认Project属性：Forgive Duplicate Directive Attributes

N: Project/Default Project Settings.../Configurations/Development/Compiler/JSP

如果JSP页面包含另外一个JSP页面，这两个页面中都Import了某个相同的类，则Build该JSP页面会报“**Error(XXX): 错误, 属性定义了两次: language**”。解决办法是选中Forgive Duplicate Directive Attributes (Helps JSP 1.1 Compatibility)。



1.4. 先睹为快：一个简单例子.创建Workspace和Project

启动Oracle9i JDeveloper

直接运行jdevbin\jdev\bin\jdevw.exe即可。

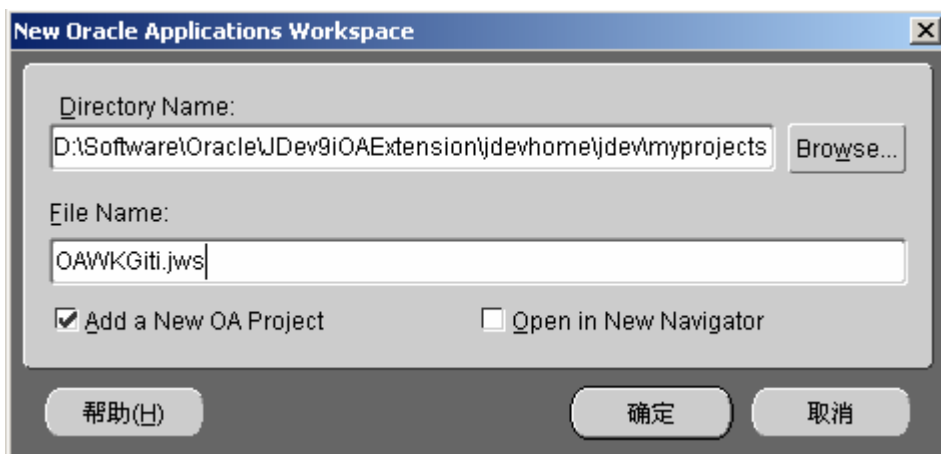
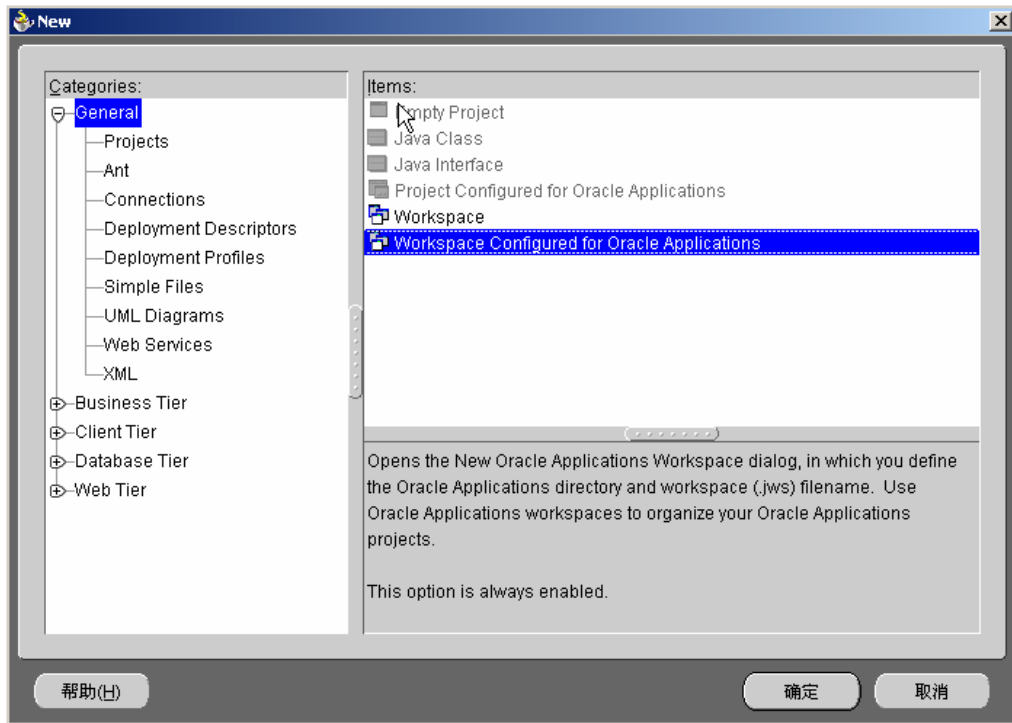
创建Workspace

N: File/New/General/Workspace Configured for Oracle Applications

和大部分IDE开发工具一样，JDeveloper也是以“Workspace/Project/具体文件”来组织程序和资源文件的。

Workspace仅仅用来组织Project。

我们选择Workspace Configured for Oracle Applications。



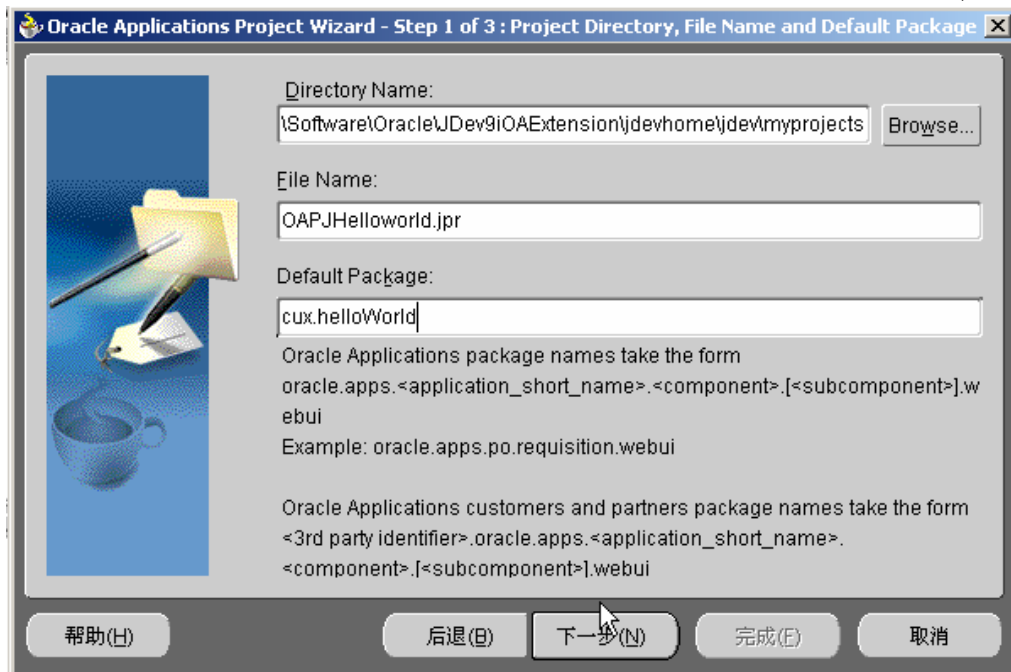
目录自动为myprojects，可以选择其他目录；不用改。

更改文件名为OAWKGiti.jws。

创建Project

N: File/New/General/Project Configured for Oracle Applications

由于上一步选中了Add a New OA Project，接下来会自动出现Project定义向导。



目录自动为myprojects，可以选择其他目录；不用改。更改文件名为OAPJHelloworld.jpr、默认包名为cux.helloWorld。

这个Project仅仅是为了练习，下面我们将正式创建GITI Project。

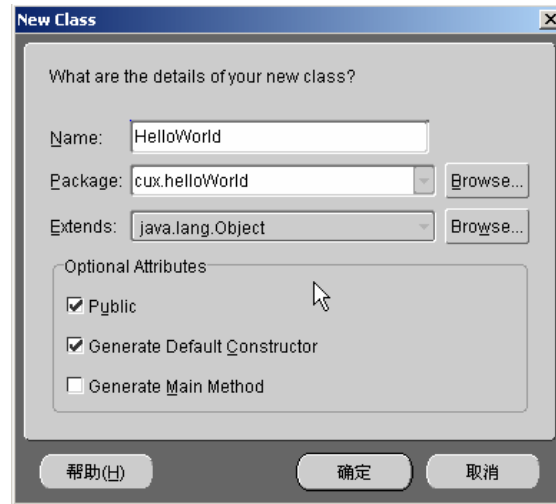
规则：Oracle标准是包名首字母小写，后续单词首字母大写；不用下划线

因为我们实际不是基于Oracle Application Framework开发的，下面的步骤都跳过。有兴趣的可以参考联机帮助《OA Framework ToolBox Tutorial》，这个有Arone整理、翻译过的中文版。

1.5. 先睹为快：一个简单例子.创建java类和jsp页面

创建java类

N: File/New/General/Java Class



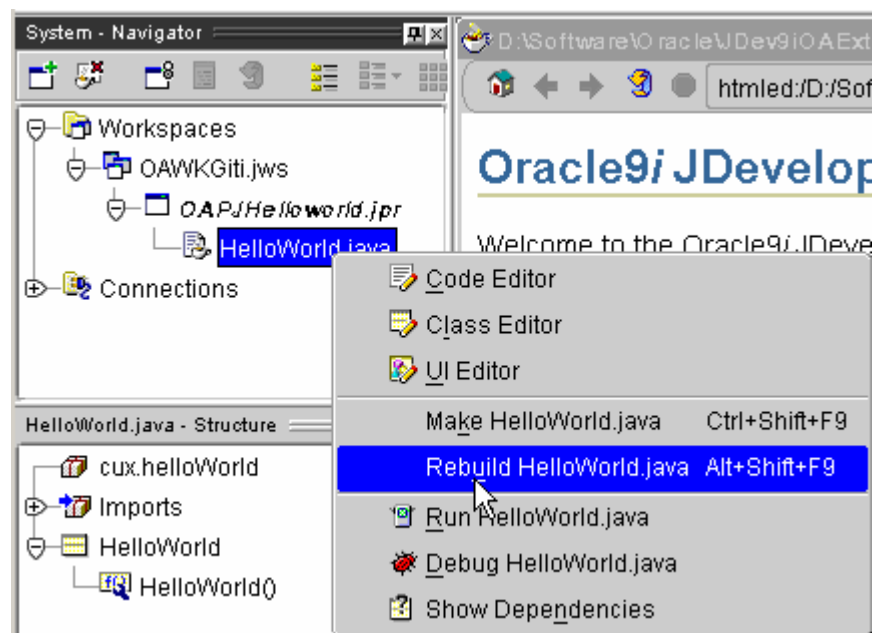
该类名为HelloWorld

规则2: 类名单词首字母大写; 如果应用某种模式, 后面要加模式名称如VO、Factory

编译java类

N: 右键/Rebuild HelloWorld.java

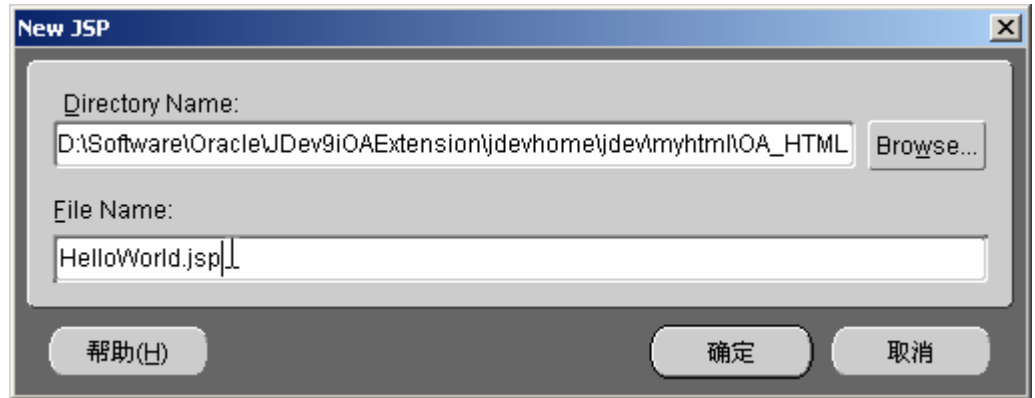
也可以通过菜单、右键代码编辑器来选择编译菜单。



状态栏显示是否由语法等错误。

创建jsp页面

N: File/New/Web Tier/JavaServer Pages (JSP)/JSP Page



目录自动为myhtml\OA_HTML，可以选择其他目录；我们改为myhtml\OA_HTML\cux。

文件名改为HelloWorld.jsp

创建完之后，可以看到文件第一行字符集为我们前面配置的GB2312
`<%@ page contentType="text/html;charset=GB2312"%>`

另外自动生成了以Hello World为Title、显示服务器时间的简单页面。

引用java类（可选，这里仅供演示）

在第二行加入

```
<%@ page import="cux.helloWorld.HelloWorld"%>
```

规则3：尽量引用具体类，不用统配符；上面也可以写成<%@ page import="cux.helloWorld.*"%>，但不推荐

编译jsp页面

N: 右键/Rebuild HelloWorld.jsp

状态栏显示是否由语法等错误。 `Successful compilation: 0 errors, 0 warnings.`

本地运行jsp页面

N: 右键/Run HelloWorld.jsp

可以看到运行结果：



1.6. 先睹为快：一个简单例子.发布

基于OA的Project，其文件组织结构和前面介绍的《了解EBS服务器目录》基本一致。

默认本地文件组织

上面例子产生的所有文件组织如下：

%JDEV_USER_HOME%	
···myhtml	
····OA_HTML	对应服务器\$OA_HTML
····cux	
····HelloWorld.jsp	
····XXXXXX	下面章节FOP打印需要的文件
····OA_MEDIA	对应服务器\$OA_MEDIA
···myprojects	java源文件放在这里，目录对应包名
····OAWKGiti.jws	
····OAPJHelloworld.jpr	
····cux	
····helloWorld	
····HelloWorld.java	
···myclasses	对应服务器\$JAVA_TOP，目录对应包名
····cux	
····helloWorld	
····HelloWorld.class	编译过的class文件
····jsps	对应服务器\$COMMON_TOP/_pages/_oa_html
····_cux	
····_HelloWorld.java	jsp对应的servlet源文件
····_HelloWorld.class	jsp对应的servlet类文件
····_HelloWorld\$_jsp_StaticText.class	jsp页面中的字符串

发布到EBS服务器

只有部分文件需要发布到服务器，以上面例子来说明：

%JDEV_USER_HOME%

···myhtml

·····OA_HTML

·····cux

1、整个文件夹ftp传到\$OA_HTML下

···myclasses

·····cux

2、整个文件夹ftp传到\$JAVA_TOP下

可以看到，没有用到图片，仅需发布两个文件夹的内容

在EBS系统管理员中完成定义

首先定义Function:

名字输入“CUXHLLWLD”

Type选择“SSWA jsp function”；

HTML Call输入\$OA_HTML下的相对路径“cux/HelloWorld.jsp”。

接下来就是定义menu、挂给responsibility、分配给用户；略。

EBS中运行

上面如果定义无误，就可以通过OA菜单打开HelloWorld.jsp，运行结果和本地运行一样。

1.7. 小结

关于如何在EBS环境下配置开发JSP的环境，到这里基本介绍完毕，至于具体的Java、JSP语法和功能这里不再讲解，请务必自行学习。

在EBS环境下开发JSP，还有一个问题需要解决：必须使用EBS的上下文获得数据库连接对象Connection，不可自行配置连接字符串。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/458135113077006045>