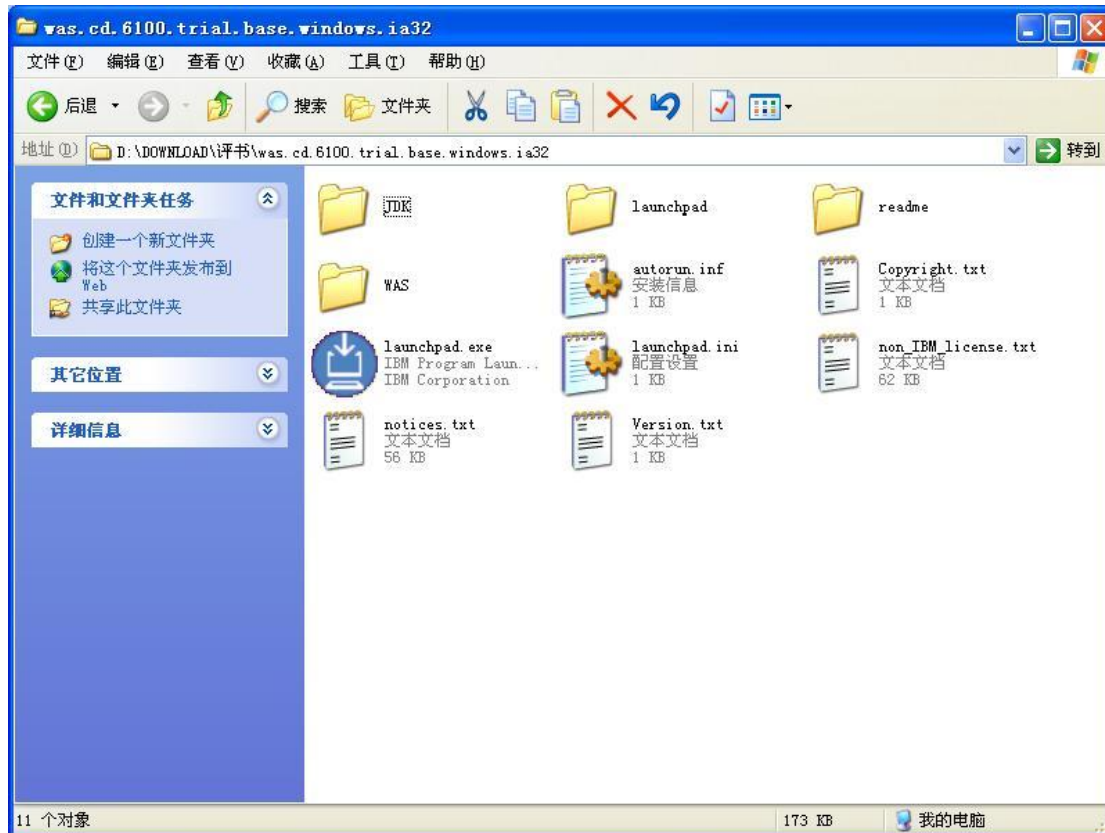


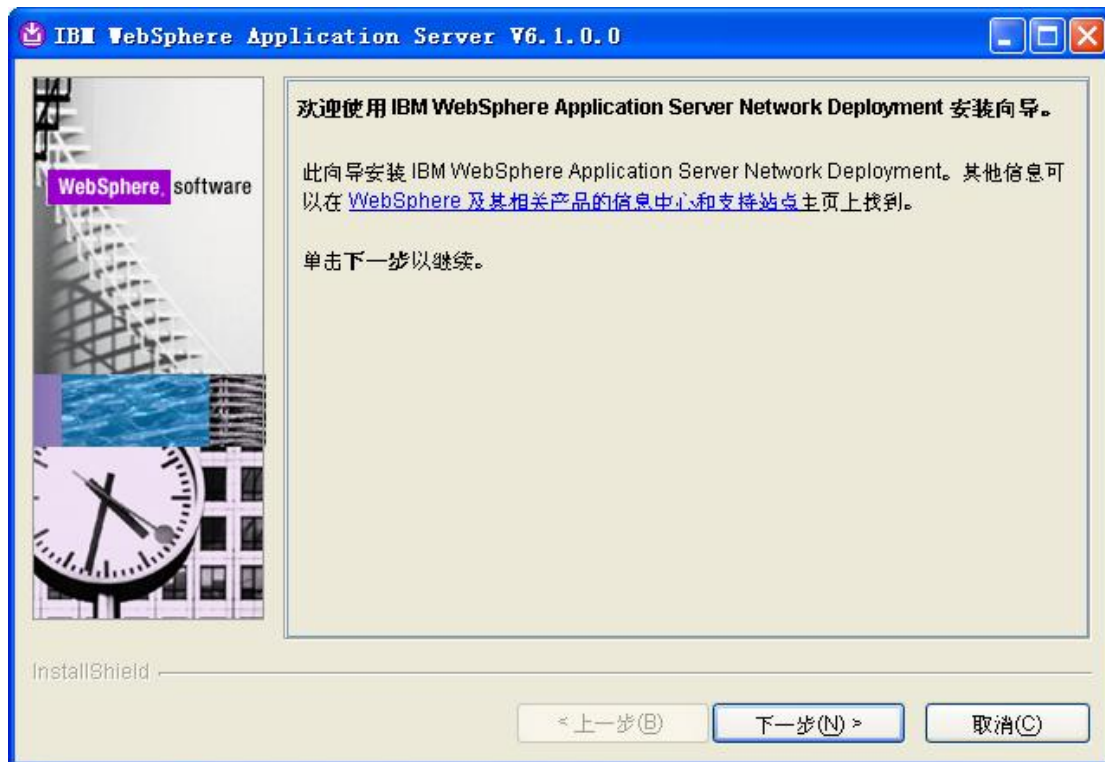
## WebSphere 的安装、部署项目和数据源的配置

### 一、WebSphere 安装步骤

#### 1、打开安装目录



#### 2、双击安装文件 launchpad.exe



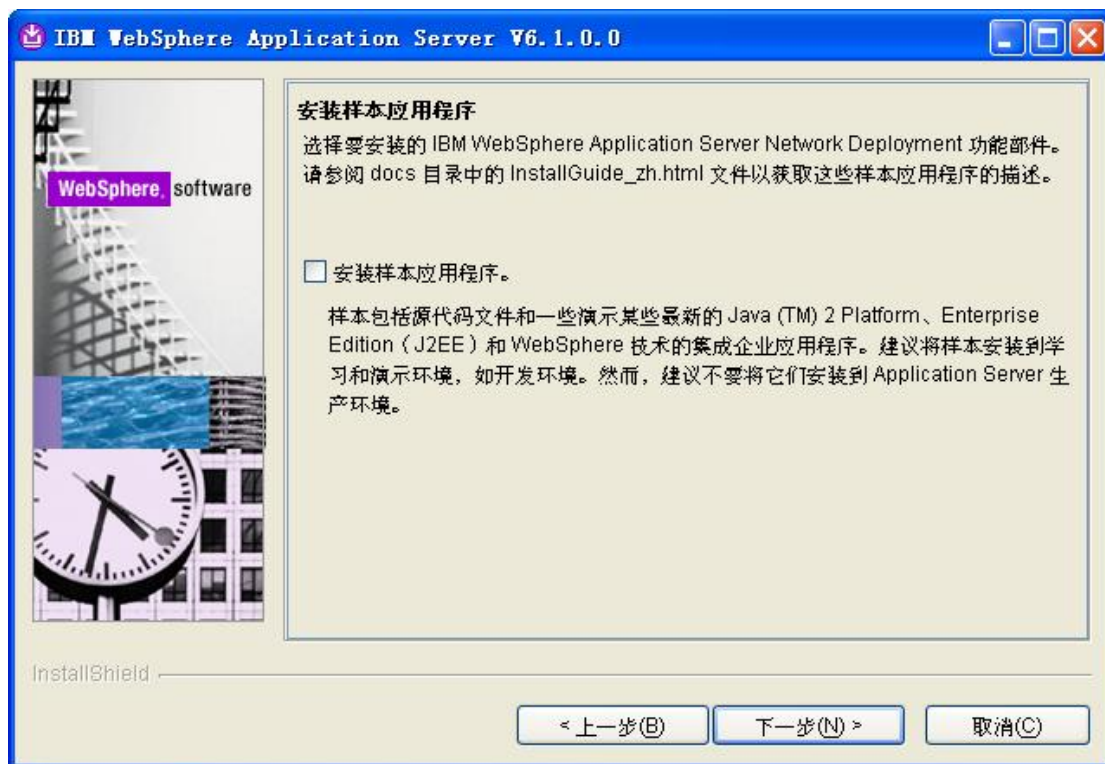
3、单击“下一步”按钮



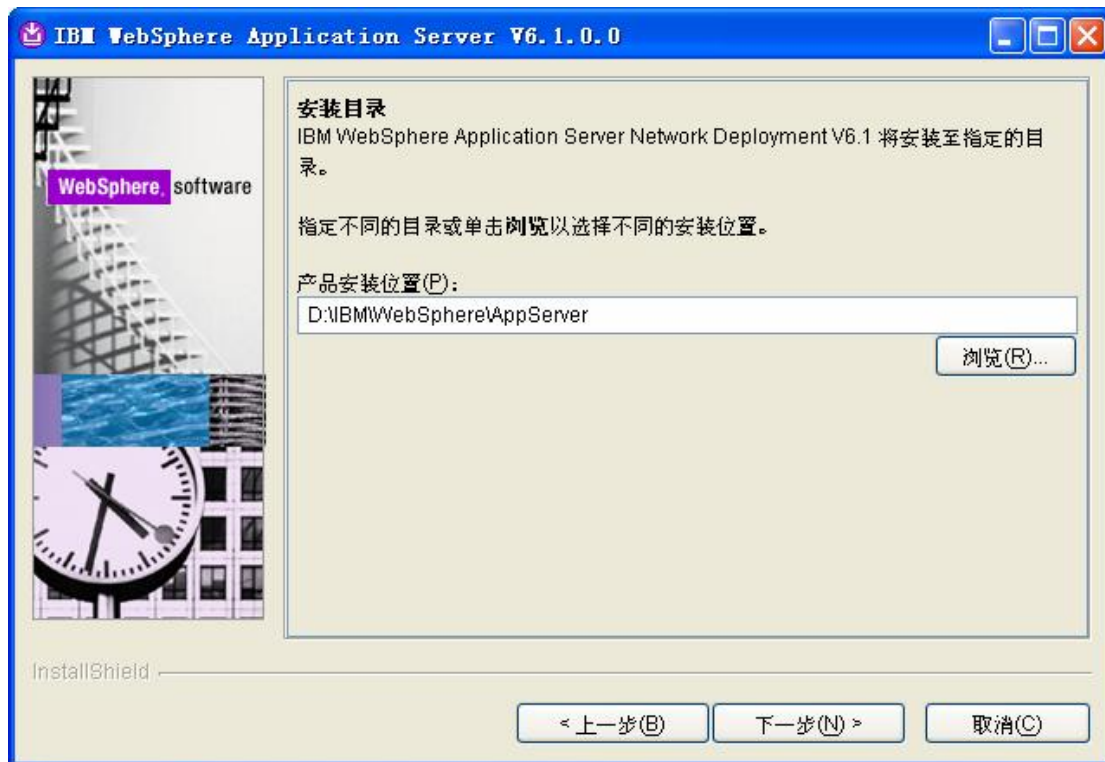
4、单击“我既接受 IBM 条款也接受非 IBM 条款。”，再单击“下一步”按钮。



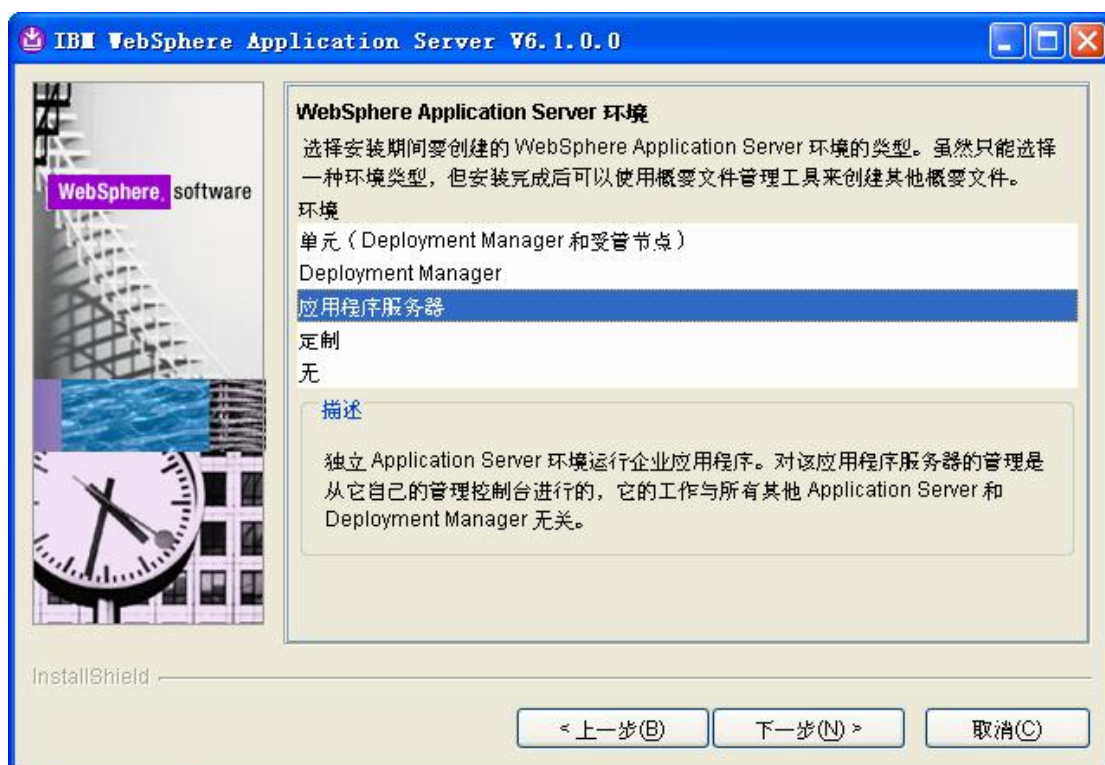
5、系统必备软件检查通过后，单击“下一步”按钮。



6、保持默认，单击“下一步”按钮。

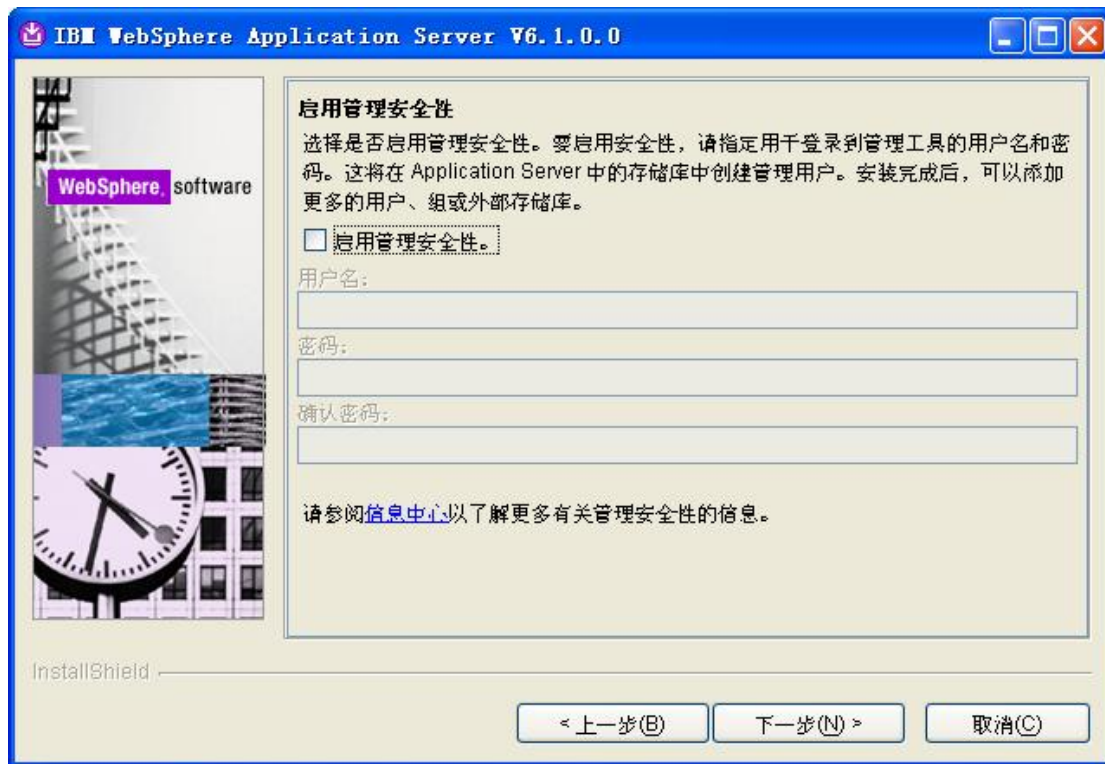


7、指定产品安装位置，建议安装在根目录下，单击“下一步”按钮。

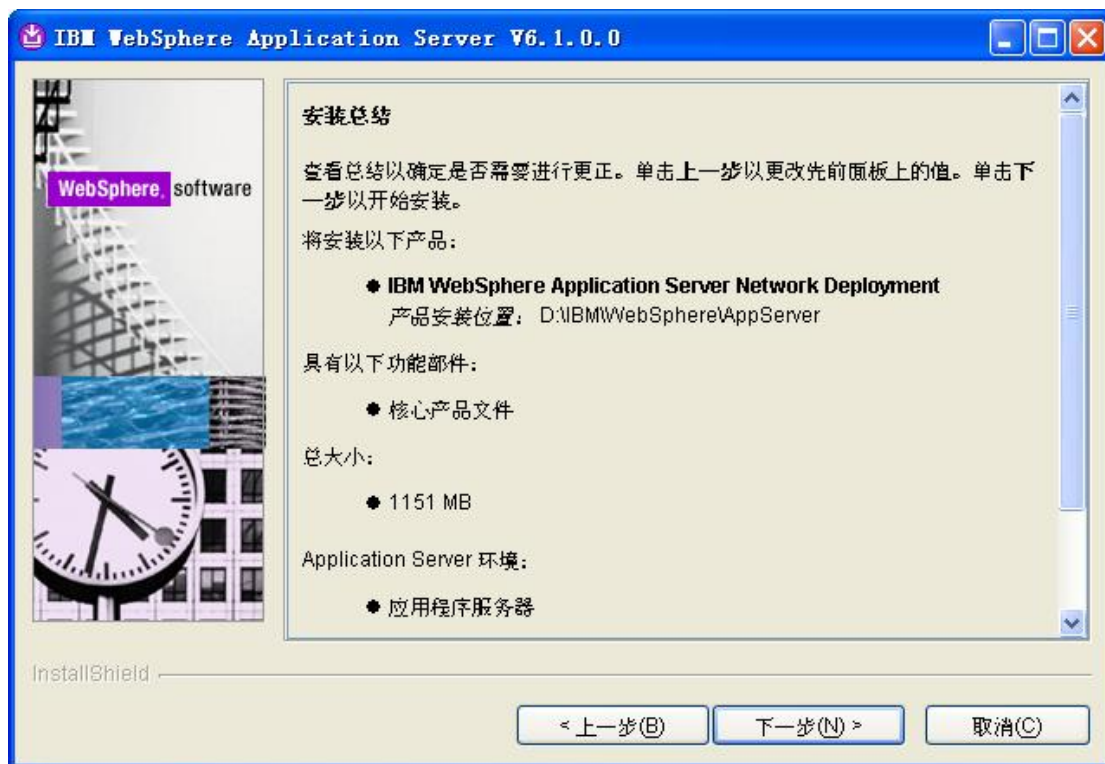


8、WebSphere Application Server 环境，选择“应用程序服务器”，单击“下一步”按钮。

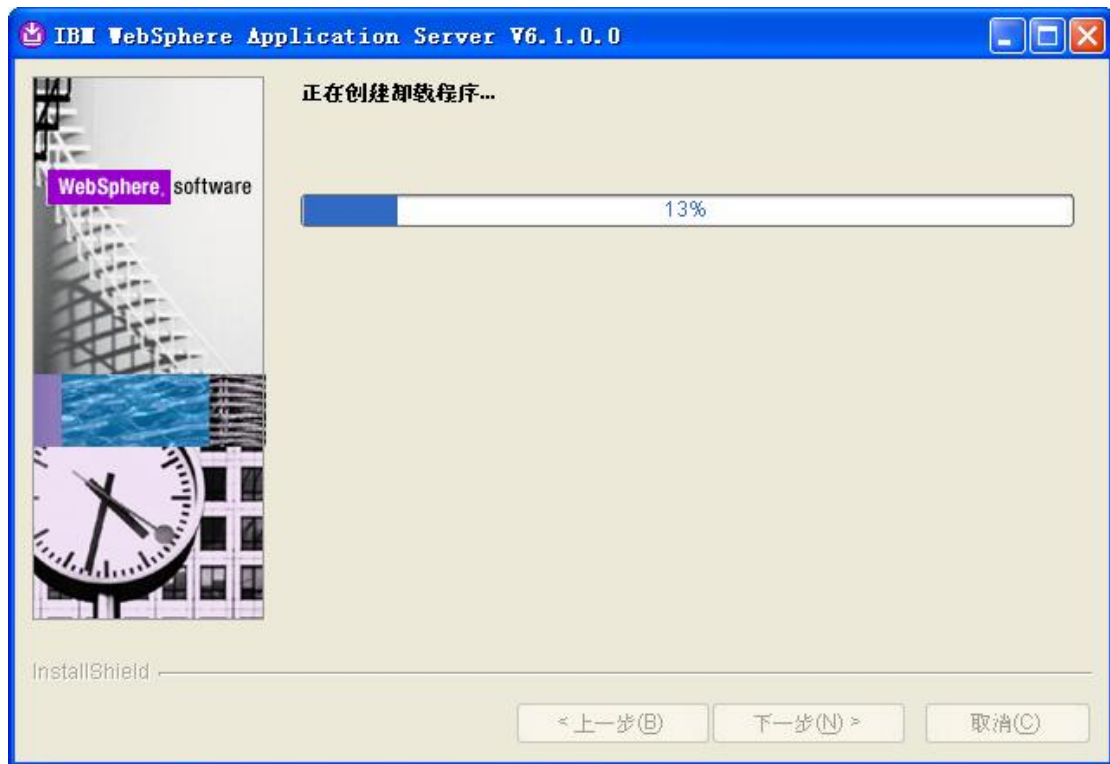




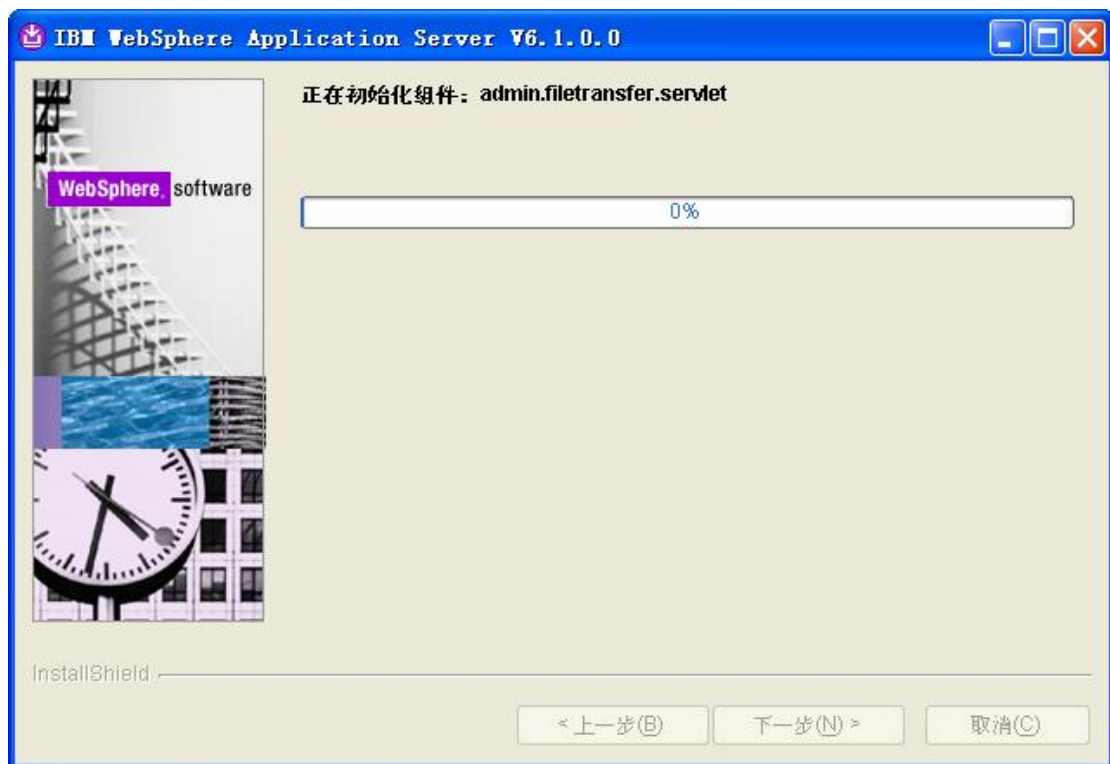
9、取消选择“启用管理安全性”，单击“下一步”按钮。



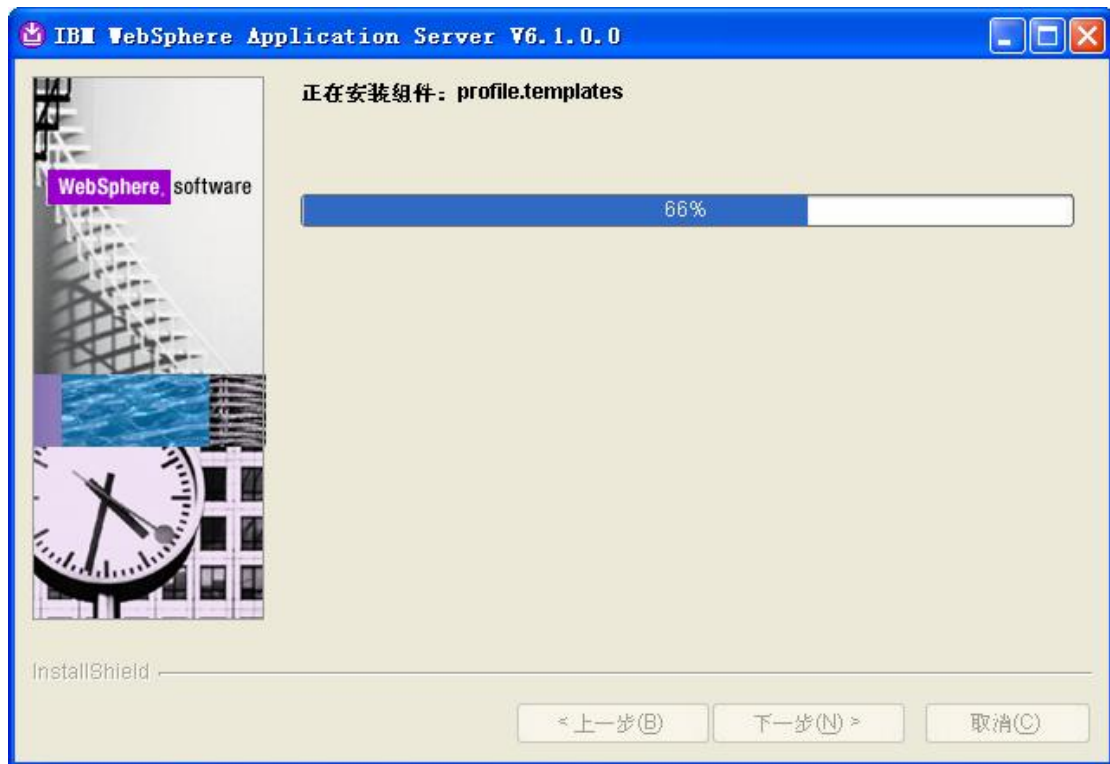
10、显示安装总结，单击“下一步”按钮。



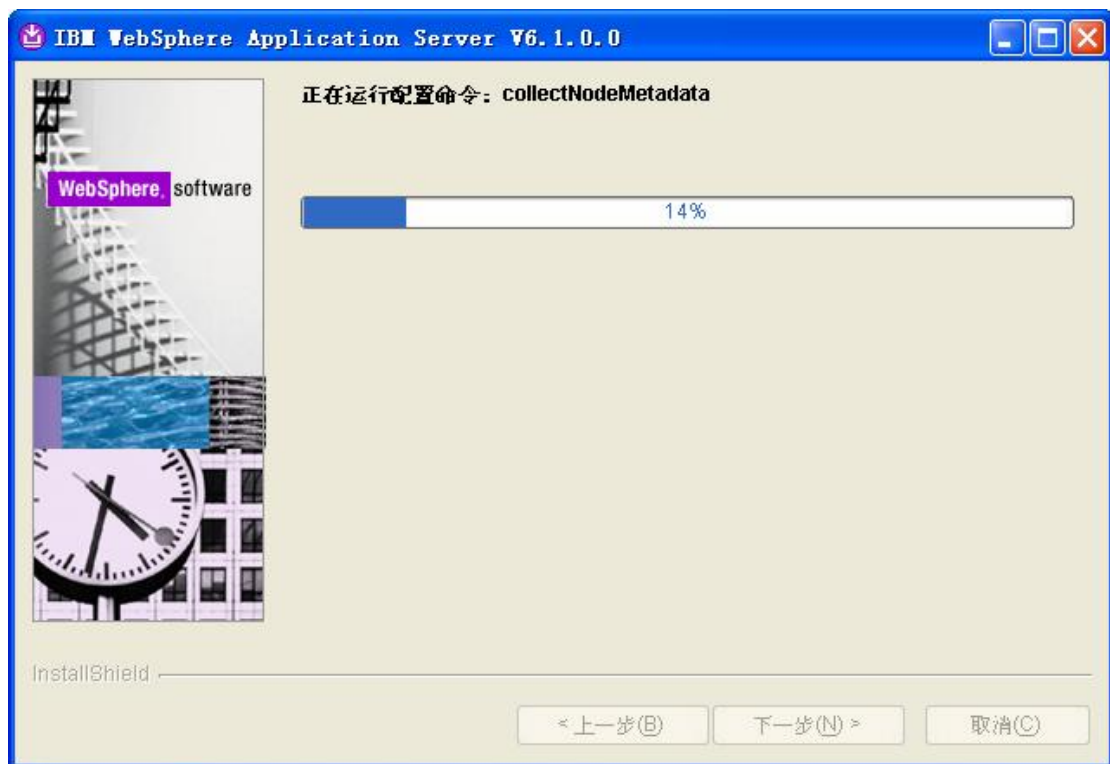
11、安装程序“正在创建卸载程序。。。”



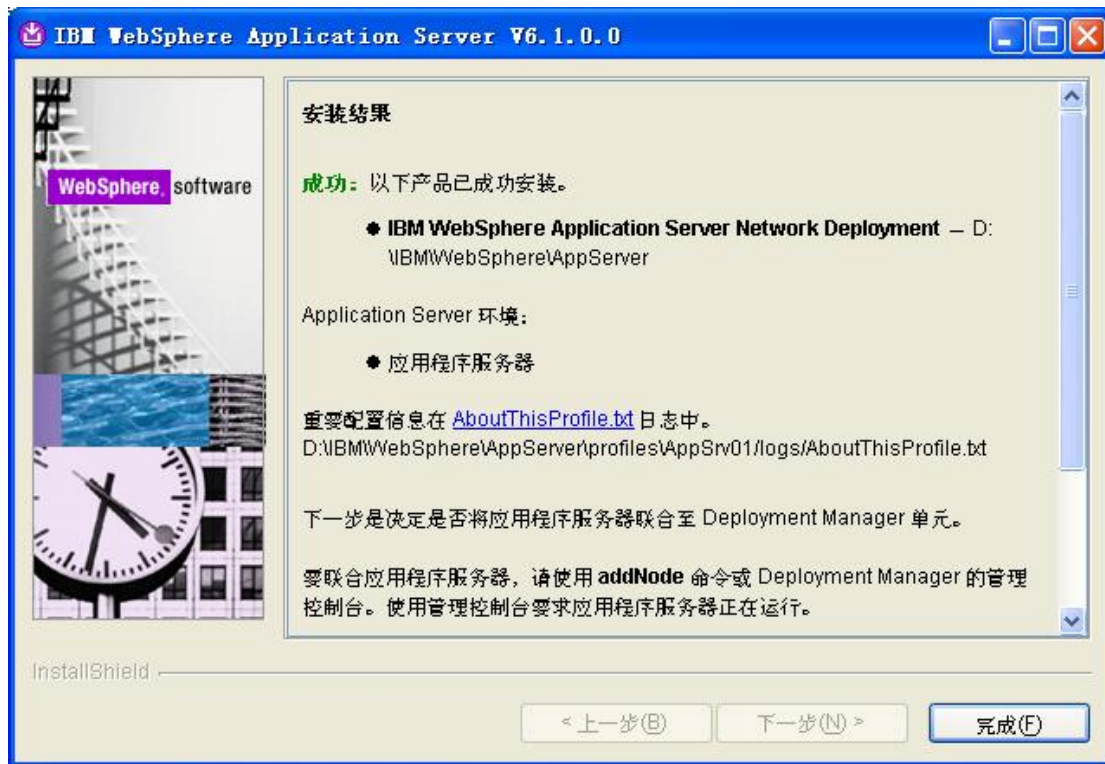
12、安装程序“正在初始化组件:”



13、安装程序“正在安装组件”



14、安装程序“正在运行配置命令”。



15、安装完成，单击“完成”按钮。



## 二、部署项目

要部署项目需要将项目打包成.war 文件



视图：所有任务

■ 欢迎

■ 指导性活动

■ 服务器

■ 应用程序

■ 企业应用程序

■ 安装新的应用程序

■ 资源

■ 安全性

■ 环境

■ 系统管理

■ 用户和组

■ 监视和调整

■ 故障诊断

■ 服务集成

■ UDDI

企业应用程序

准备应用程序安装

指定要上载并要安装的 EAR、WAR、JAR 或 SAR 模块。

新应用程序的路径

本地文件系统

完整路径

E:\workspace\JndiWeb\JndiWeb.war

浏览...

远程文件系统

完整路径

浏览...

上下文根

myweb

仅用于独立 Web 模块（.war 文件）和 SIP 模块（.sar 文件）

您想如何安装此应用程序？

仅当需要其他信息时才提示。

显示所有的安装选项和参数。

下一步

取消

注：

完整路径：将要安装布署的应用程序位置(E:\workspace\JndiWeb\JndiWeb.war)

上下文根：站点目录 myweb 如：http://localhost:9080/myweb/

安装新的应用程序

指定用于安装企业应用程序和模块的选项。

→ 步骤 1: 选择安装选项

步骤 2 将模块映射至服务器

★ 步骤 3 将资源引用映射至资源

★ 步骤 4 为 Web 模块映射虚拟主机

步骤 5 总结

选择安装选项

指定可用于准备和安装您的应用程序的各种选项。

预编译 JavaServer Pages 文件

安装应用程序的目录

分发应用程序

使用二进制配置

部署企业 bean

应用程序名

JndiWeb\_war

为资源创建 MBean

启用类重新装入

重新装入时间间隔，以秒为单位

部署 Web Service

验证输入关闭 / 警告 / 失败

警告

进程嵌入式配置

安装新的应用程序

指定用于安装企业应用程序和模块的选项。

步骤 1 选择安装选项

→ 步骤 2: 将模块映射至服务器

★ 步骤 3 将资源引用映射至资源

★ 步骤 4 为 Web 模块映射虚拟主机

步骤 5 总结

将模块映射至服务器

指定应用程序中所包含的模块的安装目标，例如，应用程序服务器或应用程序服务器集群。可将模块安装在同一台应用程序服务器上务器上。也可以指定某些 Web 服务器为目标，这些 Web 服务器将作为路由器来路由对此应用程序的请求。每个 Web 服务器的 cfg.xml 是基于通过路由的应用程序生成的。

集群和服务端：  
WebSphere:cell=39c71c176c9c4eeNode01Cell,node=39c71c176c9c4eeNode01,server=server1  
WebSphere:cell=39c71c176c9c4eeNode01Cell,node=127.0.0.1-node,server=webServer1

☒

☐

选择	模块	URI	服务器
<input type="checkbox"/>	JndiWeb.war	JndiWeb.war,WEB-INF/web.xml	WebSphere:cell=39c71c176c9c4eeNode01Cell,node=39c71c176c9c4e

上一步

下一步

取消

☒

☐

选择	模块	EJB	URI	资源引用	目标资源 JNDI 名称	登录配置
<input type="checkbox"/>	JndiWeb.war		JndiWeb.war,WEB-INF/web.xml	jdbc/aaa	<div><div>jdbc/asdf</div><div>浏览...</div></div>	资源权限： 容器 认证方法： 无

上一步

下一步

取消

注：  
目标资源 JNDI 名称：jdbc/asdf 将在后面配置，jdbc/asdf 须和应用程序一致，如：  
context.lookup(“jdbc/asdf”);

## 安装新的应用程序

指定用于安装企业应用程序和模块的选项。

步骤 1 选择安装选项

步骤 2 将模块映射至服务器

步骤 3 将资源引用映射至资源



→ 步骤 4: 为 Web 模块映射虚拟主机

步骤 5 总结

### 为 Web 模块映射虚拟主机

指定安装应用程序中包含的 Web 模块的虚拟主机。可将 Web 模块安装在同一个虚拟主机上或分散安装在几个主机上。

☒ 应用多个映射

<div><div></div><div></div></div>		
选择	Web 模块	虚拟主机
<input type="checkbox"/>	JndiWeb.war	default_host ▾

上一步

下一步

取消

注：JndiWeb.war 安装在默认主机上 default\_host

## 安装新的应用程序

指定用于安装企业应用程序和模块的选项。

步骤 1 选择安装选项

步骤 2 将模块映射至服务器

步骤 3 将资源引用映射至资源

步骤 4 为 Web 模块映射虚拟主机


→ 步骤 5: 总结

### 总结

#### 安装选项总结

选项	值
预编译 JavaServer Pages 文件	否
安装应用程序的目录	
分发应用程序	是
使用二进制配置	否
部署企业 bean	否
应用程序名	JndiWeb_war
为资源创建 MBean	是
启用类重新装入	否
重新装入时间间隔，以秒为单位	
部署 Web Service	否
验证输入关闭/警告/失败	警告
进程嵌入式配置	否

应用程序构建标识	Unknown
允许将包含内容分派给远程资源	否
允许处理来自远程资源的包含内容	否
单元/节点/服务器	<a href="#">单击此处</a>

 应用程序模块未映射至 Web 服务器。每个 Web 服务器的插件配置文件（plugin-cfg.xml）是根据映射至它的应用程序模块生成的，因此 Web 服务器不会将请求传递至此应用程序。要更改此选项，选择将模块映射至服务器步骤。

上一步

完成

取消

JndiWeb.war 安装完成

正在安装...

**如果应用程序中有 Enterprise Bean，则 EJB 部署进程可能需要几分钟。请在进程完成之后才保存配置。**

查看 Deployment Manager 或部署有应用程序的服务器上的 SystemOut.log，以获取发生 EJB 部署进程时有关此部署进程的特定信息。

ADMA5016I: 安装 JndiWeb\_war 已启动。

ADMA5067I: 应用程序 JndiWeb\_war 的资源验证成功完成。

ADMA5058I: 已使用部署目标的版本验证应用程序和模块版本。

ADMA5005I: 在 WebSphere Application Server 存储库中配置了应用程序 JndiWeb\_war。

ADMA5053I: 创建了已安装的任选软件包的库引用。

ADMA5005I: 在 WebSphere Application Server 存储库中配置了应用程序 JndiWeb\_war。

ADMA5001I: 应用程序二进制文件保存在 E:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\wstemp\92668751\workspace\cells\39c71c176c9c4eeNode01Cell\applications\JndiWeb\_war.ear\JndiWeb\_war.ear 中

ADMA5005I: 在 WebSphere Application Server 存储库中配置了应用程序 JndiWeb\_war。

SECJ0400I: 成功地用 appContextIDForSecurity 信息更新了应用程序 JndiWeb\_war。

ADMA5011I: 清除应用程序 JndiWeb\_war 的 temp 目录已完成。

ADMA5013I: 应用程序 JndiWeb\_war 安装成功。

应用程序 JndiWeb\_war 安装成功。

要启动应用程序，请先把更改保存到主配置。

已更改了您的本地配置。您可以：

直接

- [保存](#)到主配置。
- 保存或放弃之前
- [查看](#)更改。

要使用已安装的应用程序，请单击“管理应用程序”按钮。

[管理应用程序](#)

保存到主配置

### 三、WebSphere 的数据库连接池、数据源的配置

JDBC—两层结构应用 WebSphere

数据库连接特点:

- 客户端程序在启动时打开数据库连接，在退出程序时关闭数据库连接
- 在整个程序运行中，每个客户端始终占用一个数据库连接，即使在大量没有数据库操作的空闲时间，从而造成数据库连接的使用效率低下
- 客户端数量大时，数据库资源消耗大、性能急剧下降

JDBC—三层结构应用

数据库连接特点:

- 采用连接池管理



- （好处）需要进行数据库操作时，才从连接池申请一个连接，数据库操作完毕，连接立即释放到连接池中，以供其他用户使用
- 共享较少的数据库连接，提高了数据库连接的使用效率
- 节省建立连接的时间（不需客户建立）

## JDBC—连接池创建步骤

- WebSphere 环境变量：驱动程序路径
- 创建 JDBC 驱动的提供者
- 创建数据源
- 创建连接认证
- 数据源与连接认证关联
- 测试连接

## JDBC—oracle 连接池示例

- 连接方式（Oracle，2 种方式）
  - thin（通过连接字符的创连接到服务器端，必须安装 Oracle 客户端程序）
  - oci（客户端只需有 JDBC 驱动程序就行）
- classes12.jar（9i 前） / ojdbc14.jar（9i 后）（）

第一步：设置 WebSphere 变量 主要是数据库驱动程序的.jar 文件

视图：|所有任务

■ 欢迎

☐ 指导性活动

☐ 服务器

☐ 应用程序

☐ 资源

☐ 安全性

☐ 环境
 

■ 虚拟主机

■ 更新全局 Web 服务器插

■ WebSphere 变量

■ 共享库

■ 复制域

☐ 命名

☐ 系统管理

☐ 用户和组

☐ 监视和调整

☐ 故障诊断

☐ 服务集成

☐ UDDI

websphere 变量

WebSphere 变量

WebSphere 变量

使用此页面来定义替换变量。这些变量指定某些系统定义的值（例如，文件系统根目录）的间接程度。变量有作用域级别，即服务器、节点、节点组、节点组或节点。某个作用域的值可与其他级别的值不同。当变量具有冲突的作用域值时，粒度较细的值会覆盖作用域级别高一些的值。因此，服务器变量覆盖节点变量，节点变量覆盖集群变量，集群变量覆盖单元变量。

☐ 作用域：=所有作用域
 

作用域指定资源定义可视的级别。要获得有关作用域的内容以及如何工作的详细信息，[请参阅作用域设置帮助](#)

所有作用域

☐ 首选项
 

新建

删除

☐

☐

☐

☐

选择	名称	值	作用域
<input type="checkbox"/>	<a href="#">APP_INSTALL_ROOT</a>	\${USER_INSTALL_ROOT}/installedApps	节点=39c71c176c9c4e
<input type="checkbox"/>	<a href="#">CONNECTJDBC JDBC DRIVER PATH</a>		节点=39c71c176c9c4e
<input type="checkbox"/>	<a href="#">CONNECTOR_INSTALL_ROOT</a>	\${USER_INSTALL_ROOT}/installedConnectors	节点=39c71c176c9c4e
<input type="checkbox"/>	<a href="#">DB2390 JDBC DRIVER PATH</a>		节点=39c71c176c9c4e
<input type="checkbox"/>	<a href="#">DB2UNIVERSAL JDBC DRIVER NATIVEPATH</a>		节点=39c71c176c9c4e

**WebSphere 变量**

使用此页面来定义替换变量。这些变量指定某些系统定义的值（例如，文件系统根目录）的间接程度。变量有作用域级别，即服务器、节点、集群或单元。某个作用域的值可与其他级别的值不同。当变量具有冲突的作用域值时，粒度较细的值会覆盖作用域级别高一些的值。因此，服务器变量覆盖节点变量，节点变量覆盖集群变量，集群变量覆盖单元变量。

☐ 作用域: =所有作用域

作用域指定资源定义可视的级别。要获得有关作用域的内容以及如何工作的详细信息，[请参阅作用域设置帮助](#)

所有作用域

田 首选项

<div> <div>新建</div> <div>删除</div> </div>			
<div> <div> <div></div> <div></div> <div></div> <div></div> </div> </div>			
选择	名称	值	作用域
<input type="checkbox"/>	<a href="#">APP_INSTALL_ROOT</a>	\${USER_INSTALL_ROOT}/installedApps	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">CONNECTJDBC JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">CONNECTOR_INSTALL_ROOT</a>	\${USER_INSTALL_ROOT}/installedConnectors	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">DB2390 JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">DB2UNIVERSAL JDBC DRIVER NATIVEPATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">DB2UNIVERSAL JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">DB2 JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">DEPLOY_TOOL_ROOT</a>	\${WAS_INSTALL_ROOT}/deploytool/itp	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">DERBY JDBC DRIVER PATH</a>	\${WAS_INSTALL_ROOT}/derby/lib	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">DRIVER_PATH</a>	\${WAS_INSTALL_ROOT}	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">INFORMIX JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">JAVA_HOME</a>	E:/Program Files/IBM/WebSphere/AppServer/java	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">JVM_CACHE</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">LOG_ROOT</a>	\${USER_INSTALL_ROOT}/logs	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">MQ_INSTALL_ROOT</a>	\${WAS_INSTALL_ROOT}/lib/WMQ	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">MSSQLSERVER JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input checked="" type="checkbox"/>	<a href="#">ORACLE JDBC DRIVER PATH</a>	E:\oracle\ora921\jdbc\lib	节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">OS400_NATIVE JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">OS400_TOOLBOX JDBC DRIVER PATH</a>		节点=39c71c176c9c4eeNode01
<input type="checkbox"/>	<a href="#">SERVER_LOG_ROOT</a>	\${LOG_ROOT}/server1	节点=39c71c176c9c4eeNode01, 服务器=server1

点击设置 ORACLE\_JDBC\_DRIVER\_PATH

WebSphere 变量

WebSphere 变量 > ORACLE\_JDBC\_DRIVER\_PATH

使用此页面来定义替换变量。这些变量指定某些系统定义的值（例如，文件系统根目录）的间接程度。变量有作用域级别，即服务器、节点、集群或单元。某个作用域的值可与其他级别的值不同。当变量具有冲突的作用域值时，粒度较细的值会覆盖作用域级别高一些的值。因此，服务器变量覆盖节点变量，节点变量覆盖集群变量，集群变量覆盖单元变量。

配置

常规属性

\* 名称

ORACLE\_JDBC\_DRIVER\_PATH

值

E:\oracle\ora921\jdbc\lib

描述

The directory that contains the Oracle thin or oci8 JDBC Driver.

应用 确定 复位 取消

注：值为 Oracle 安装目录下jdbc\lib,也就是驱动程序.jar 文件所有的目录

第二步：JDBC 提供程序



## JDBC 提供程序

使用此页面来编辑 JDBC 提供程序的属性。JDBC 提供程序对象包括特定的 JDBC 驱动程序实现类，以便访问环境中的特定供应商数据库。在 [guided activity](#) 中学习有关此任务的更多内容。指导性活动提供任务步骤列表和关于主题的更多信息。

作用域: 单元=39c71c176c9c4eeNode01Cell, 节点=39c71c176c9c4eeNode01, 服务器=server1

作用域指定资源定义可视的级别。要获得有关作用域的内容以及如何工作的详细信息，[请参阅作用域设置帮助](#)

节点=39c71c176c9c4eeNode01, 服务器=server1

所有作用域  
单元=39c71c176c9c4eeNode01Cell  
节点=39c71c176c9c4eeNode01  
节点=39c71c176c9c4eeNode01, 服务器=server1  
节点=127.0.0.1-node

选择	名称	作用域	描述
<input type="checkbox"/>	<a href="#">Derby JDBC Provider</a>	节点=39c71c176c9c4eeNode01, 服务器=server1	Derby embedded non-XA JDBC Provider
<input type="checkbox"/>	<a href="#">Samples Derby JDBC Provider (XA)</a>	节点=39c71c176c9c4eeNode01, 服务器=server1	Built-in Derby JDBC Provider (XA)

总数 2

注：选择 JDBC 提供程序作用域，详细作用域设置本文档在最后面有说明

## 新建

### 创建新的 JDBC 提供程序

#### → 步骤 1: 创建新的 JDBC 提供程序

步骤 2: 输入数据库类路径信息

步骤 3: 总结

#### 创建新的 JDBC 提供程序

设置 JDBC 提供程序的基本配置值，该提供程序包含访问数据库所需的特定供应商 JDBC 驱动程序实现类。向导会填写名称和描述字段，但您也可输入不同的值。

作用域

cells:39c71c176c9c4eeNode01Cell:nodes:39c71c176c9c4eeNode01:servers:server1

\* 数据库类型

Oracle

\* 提供程序类型

Oracle JDBC Driver

\* 实施类型

连接池数据源

\* 名称

Oracle JDBC Driver

描述

Oracle JDBC Driver

下一步

取消



注：数据库类型为 Oracle  
实施类型：连接池数据源

创建新的 JDBC 提供程序

创建新的 JDBC 提供程序

步骤 1: 创建新的 JDBC 提供程序

→ 步骤 2: 输入数据库类路径信息

步骤 3: 总结

输入数据库类路径信息

设置表示 JDBC 驱动程序类文件的环境变量，WebSphere(R) Application Server 使用它们来定义 JDBC 提供程序。此向导页显示文件名，您仅提供这些文件的目录位置。在输入 JDBC 驱动程序文件位置时，请使用完整的目录路径。例如： /home/db2inst1/sqllib/java（在 Linux(TM) 上）。如果您指定了值，则可单击“下一步”以接受该值。

类路径:

另存为 WebSphere 变量 `${ORACLE_JDBC_DRIVER_PATH}` 的 `ojdbc14.jar` 的目录位置

上一步

下一步

取消

注：定位 JDBC 驱动程序类文件的环境变量，  
类路径：`${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar`  
`ORACLE_JDBC_DRIVER_PATH` 前面我们已经在 WebSphere 变量中设置

创建新的 JDBC 提供程序

创建新的 JDBC 提供程序

步骤 1: 创建新的 JDBC 提供程序

步骤 2: 输入数据库类路径信息

→ 步骤 3: 总结

总结

操作总结:

选项	值
作用域	cells:39c71c176c9c4eeNode01Cell:nodes:39c71c176c9c4eeNode01
JDBC 提供程序名称	Oracle JDBC Driver
描述	Oracle JDBC Driver
类路径	<code>\${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar</code>
<code>\${ORACLE_JDBC_DRIVER_PATH}</code>	
本机路径	
实现类名	<code>oracle.jdbc.pool.OracleConnectionPoolDataSource</code>

上一步

完成

取消

注：JDBC 提供程序设置完成

田 首选项

新建 删除			
   			
选择	名称 ◇	作用域 ◇	描述 ◇
<input type="checkbox"/>	<a href="#">Derby JDBC Provider</a>	节点 =39c71c176c9c4eeNode01,服 务器=server1	Derby embedded non-XA JDBC Provider
<input checked="" type="checkbox"/>	<a href="#">Oracle JDBC Driver</a>	节点 =39c71c176c9c4eeNode01,服 务器=server1	Oracle JDBC Driver
<input type="checkbox"/>	<a href="#">Samples Derby JDBC Provider (XA)</a>	节点 =39c71c176c9c4eeNode01,服 务器=server1	Built-in Derby JDBC Provider (XA)
总数 3			

第三步：数据源

资源

- Scheduler
- 对象池管理器
- JMS
- JDBC
  - JDBC 提供程序
  - 数据源
  - 数据源 (WebSphere Application Server)
- 资源适配器
- Asynchronous beans
- 高速缓存实例
- 邮件
- URL
- 资源环境

数据源

使用此页面来编辑与所选 JDBC 提供程序相关联的数据源设置。数据源对象为应用程序提供用于访问数据库的连接。在 [guided activity](#) 中有关此任务的更多内容。指导性活动提供任务步骤列表和关于主题的更多一般信息。

作用域: 单元=39c71c176c9c4eeNode01Cell, 节点=39c71c176c9c4eeNode01, 服务器=server1

作用域指定资源定义可见的级别。要获得有关作用域的内容以及如何工作的详细信息，[请参阅作用域设置帮助](#)

节点=39c71c176c9c4eeNode01, 服务器=server1

首选项

新建 删除 测试连接 管理状态...

选择	名称	JNDI 名称	作用域	提供程序	描述	类
<input type="checkbox"/>	<a href="#">Default Datasource</a>	DefaultDatasource	节点=39c71c176c9c4eeNode01, 服务器=server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input type="checkbox"/>	<a href="#">PLANTSDB</a>	jdbc/PlantsByWebSphereDataSource	节点=39c71c176c9c4eeNode01, 服务器=server1	Samples Derby JDBC Provider (XA)	Data source for the Plants by WebSphere entity beans	

总数 2

注：选择作用域，新建数据源

创建数据源

创建数据源

→ 步骤 1: 输入基本数据源信息

步骤 2: 选择 JDBC 提供程序

步骤 3: 输入数据源的特定于数据库的属性

步骤 4: 总结

输入基本数据源信息

设置数据源的基本配置值以便与 JDBC 提供程序关联。数据源提供应用程序服务器和数据库之间的物理连接。

要求：如果应用程序基于 Enterprise JavaBeans(TM) (EJB) 1.0 规范或 Java (TM) Servlet 2.2 规范，则使用数据源（WebSphere(R) Application Server V4）控制台页面。

作用域  
cells:39c71c176c9c4eeNode01Cell:nodes:39c71c176c9c4eeNode01:servers:server1

\* 数据源名  
thin datasource

\* JNDI 名称  
jdbc/asdf

组件管理的认证别名和 XA 恢复认证别名  
选择组件管理的认证别名。如果 JDBC 提供程序支持 XA，则所选认证别名也将设置为 XA 恢复认证别名。如果选择 [创建新的 J2C 认证别名](#)，则此向导将被取消。

(无)  
(无)  
39c71c176c9c4eeNode01Cell/samples

下一步 取消

注：数据源名为 thin datasource(名称用户自定义)

JNDI 名称：这个就是应用程序连接数据库用的 JNDI 名称，要和应用程序里对应 Context.lookup(“jdbc/asdf”); 这里我们用的 jdbc/asdf

组件管理的认证别名和 XA 恢复认证别名：如果前面我们已经创建了可直接选取这里我们将”创建新的 J2c 认证别名”。



第四步：创建连接认证



注：新建

**数据源 > JAAS - J2C 认证数据 > 新建**

指定供 Java(TM) 2 连接器安全性使用的一组用户标识和密码。

**配置**

**常规属性**

\* 别名

scott

\* 用户标识

scott

\* 密码

•••••

描述

应用

确定

复位

取消

用户标识：数据库登录用户名

密码：数据库登录密码

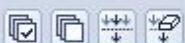
**数据源 > JAAS - J2C 认证数据**

指定供 Java(TM) 2 连接器安全性使用的一组用户标识和密码。

田 首选项

新建

删除



选择	别名	用户标识	描述
<input checked="" type="checkbox"/>	<a href="#">39c71c176c9c4eeNode01/scott</a>	scott	
<input type="checkbox"/>	<a href="#">39c71c176c9c4eeNode01Cell/samples</a>	samples	JAAS Alias for WebSphere Samples
总数 2			

注：创建完成。回到数据源

第五步：数据源与连接认证关联

创建数据源

创建数据源

→ 步骤 1: 输入基本数据源信息

步骤 2: 选择 JDBC 提供程序

步骤 3: 输入数据源的特定于数据库的属性

步骤 4: 总结

输入基本数据源信息

设置数据源的基本配置值以便与 JDBC 提供程序关联。数据源提供应用程序服务器和数据库之间的物理连接。

要求：如果应用程序基于 Enterprise JavaBeans(TM) (EJB) 1.0 规范或 Java(TM) Servlet 2.2 规范，则使用数据源 (WebSphere(R) Application Server V4) 控制台页面。

作用域  
cells:39c71c176c9c4eeNode01Cell:nodes:39c71c176c9c4eeNode01:servers:server1

\* 数据源名  
thin\_datasource

\* JNDI 名称  
jdbc/asdf

组件管理的认证别名和 XA 恢复认证别名

选择组件管理的认证别名。如果 JDBC 提供程序支持 XA，则所选认证别名也将设置为 XA 恢复认证别名。如果选择 [创建新的 J2C 认证别名](#)，则此向导将被取消。

39c71c176c9c4eeNode01/scott  
(无)  
39c71c176c9c4eeNode01Cell/samples  
39c71c176c9c4eeNode01/scott

下一步 取消

创建数据源

创建数据源

步骤 1: 输入基本数据源信息

→ 步骤 2: 选择 JDBC 提供程序

步骤 3: 输入数据源的特定于数据库的属性

步骤 4: 总结

选择 JDBC 提供程序

指定 JDBC 提供程序以支持此数据源。

☐ 创建新的 JDBC 提供程序

☒ 选择现有的 JDBC 提供程序

选择...  
选择...  
Derby JDBC Provider  
Samples Derby JDBC Provider (XA)  
Oracle JDBC Driver

上一步 下一步 取消

注：  
如果没有创建 JDBC 提供程序，请创建新的 JDBC 提供程序，我们前面已经创建了 JDBC 提供程序 Oracle JDBC Driver。选取 Oracle JDBC Driver

创建数据源

创建数据源

步骤 1: 输入基本数据源信息

步骤 2: 选择 JDBC 提供程序

→ 步骤 3: 输入数据源的特定于数据库的属性

步骤 4: 总结

输入数据源的特定于数据库的属性

设置这些特定于数据库的属性，数据库供应商 JDBC 驱动程序需要这些属性以支持通过此数据源进行管理的连接。

\* URL

\* 数据存储器 helper 类名

☒ 将此数据源用于容器管理的持久性 (CMP)

上一步

下一步

取消

注:

URL: jdbc:oracle:thin@Localhost:Port:SID

创建数据源

创建数据源

步骤 1: 输入基本数据源信息

步骤 2: 选择 JDBC 提供程序

步骤 3: 输入数据源的特定于数据库的属性

→ 步骤 4: 总结

总结

操作总结:
 

选项	值
作用域	cells:39c71c176c9c4eeNode01Cell:nodes:39c71c176c9c4eeNode01:servers:se
数据源名	thin datasource
JNDI 名称	jdbc/asdf
组件管理的认证别名	39c71c176c9c4eeNode01/scott
选择现有的 JDBC 提供程序	Oracle JDBC Driver
实现类名	oracle.jdbc.pool.OracleConnectionPoolDataSource
URL	jdbc:oracle:thin:@127.0.0.1:1521:abc
数据存储器 helper 类名	com.ibm.websphere.rsadapter.OracleDataStoreHelper
将此数据源用于容器管理的持久性 (CMP)	true

上一步

完成

取消

注: 数据源创建完成, 保存设置



第六步：测试连接

数据源

使用此页面来编辑与所选 JDBC 提供程序相关联的数据源设置。数据源对象为应用程序提供用于访问数据库的连接。在 [guided activity](#) 有关此任务的更多内容。指导性活动提供任务步骤列表和关于主题的更多一般信息。

作用域: 单元=39c71c176c9c4eeNode01Cell, 节点=39c71c176c9c4eeNode01, 服务器=server1

作用域指定资源定义可视的级别。要获得有关作用域的内容以及如何工作的详细信息，[请参阅作用域设置帮助](#)

节点=39c71c176c9c4eeNode01, 服务器=server1

田 首选项

新建 删除 测试连接 管理状态...

选择	名称	JNDI 名称	作用域	提供程序	描述	类
<input type="checkbox"/>	<a href="#">Default Datasource</a>	DefaultDataSource	节点=39c71c176c9c4eeNode01, 服务器=server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input type="checkbox"/>	<a href="#">PLANTSDB</a>	jdbc/PlantsByWebSphereDataSource	节点=39c71c176c9c4eeNode01, 服务器=server1	Samples Derby JDBC Provider (XA)	Data source for the Plants by WebSphere entity beans	
<input checked="" type="checkbox"/>	<a href="#">thin datasource</a>	jdbc/asdf	节点=39c71c176c9c4eeNode01, 服务器=server1	Oracle JDBC Driver	New JDBC Datasource	

总数 3

数据源

田 消息

在节点 39c71c176c9c4eeNode01 上的服务器 server1 上，对数据源 thin datasource 执行的测试连接操作是成功的。

注：数据源测试连接成功。

企业应用程序测试

测试代码：

```
<%@ page contentType="text/html; charset=GB2312"%>
<%@ page import="java.sql.*"%>
<%@ page import="javax.sql.DataSource"%>
<%@ page import="javax.naming.*"%>

<html>
  <head>
    <title>WebSphere 测试</title>
```

```

</head>
<body>
    <font size="12" color="red" align="center">WebSphere 数据源</font>
    <br/>
    <%
        Context initCtx = new InitialContext();
        //从 Context 中 lookup 数据源。(相当于已经获得了数据库连接)
        DataSource ds = null;
        Connection con = null;
        Statement stmt = null;
        try
        {
            ds = (DataSource) initCtx.lookup("jdbc/asdf");
            if(ds != null)
            {
                con=ds.getConnection();
            }
            else
            {
                System.out.println("-----获取 JNDI 连接失败!!");
            }

            String sql="select * from scott.dept";

            stmt=con.createStatement();
            java.sql.ResultSet rs=stmt.executeQuery(sql);
            System.out.println(sql);
            while(rs.next())
            {
                out.println(rs.getString(1));
                out.println("-----");
                out.println(rs.getString(2));
                out.println("<br/>");
            }
            rs.close();
        }
        catch(Exception ex)
        {
            out.println(ex.toString());
        }

    %>
</body>

```

</html>

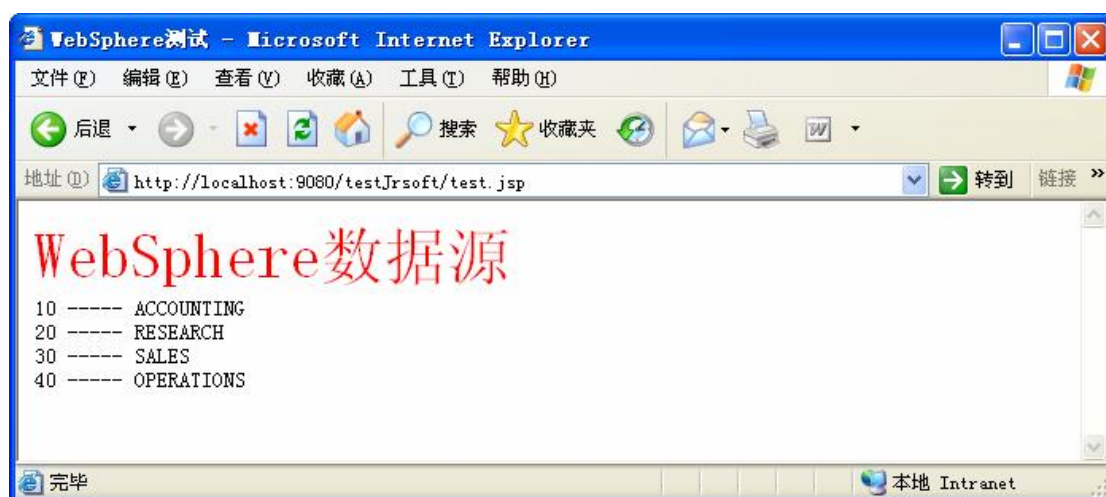


注：启动企业应用程序



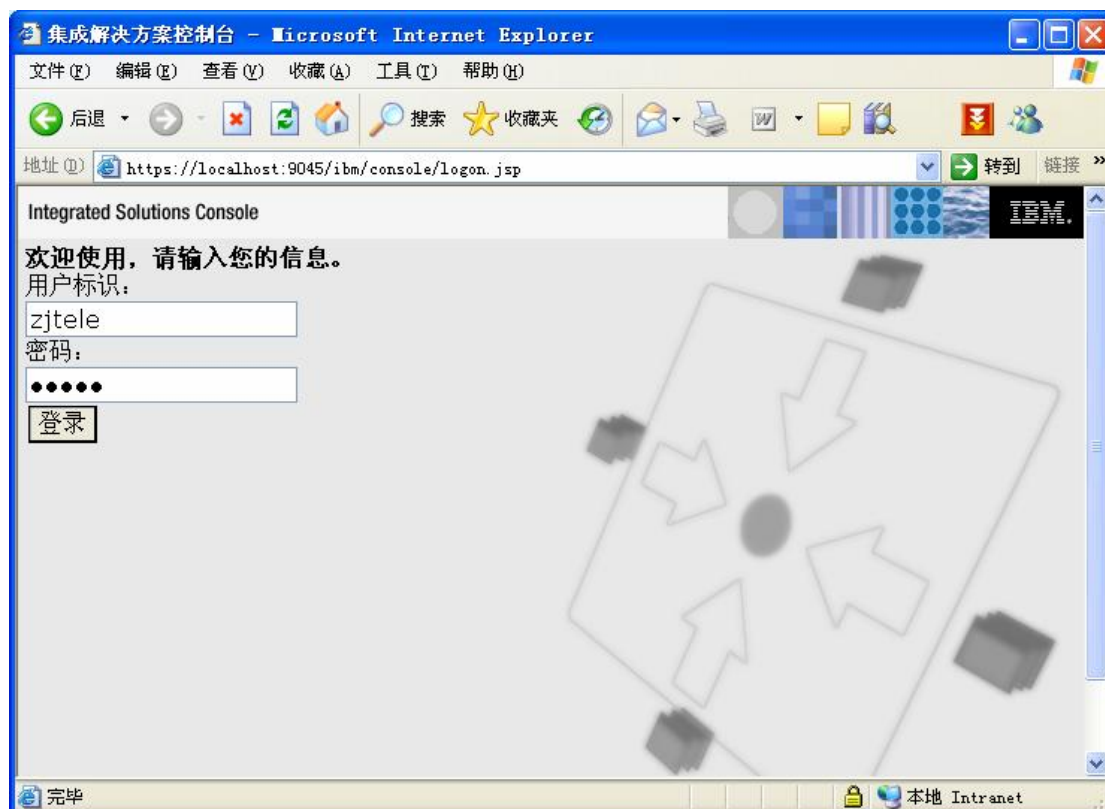
注：企业应用程序启动成功

打开浏览器输入：<http://localhost:9080/testJrsoft/test.jsp>



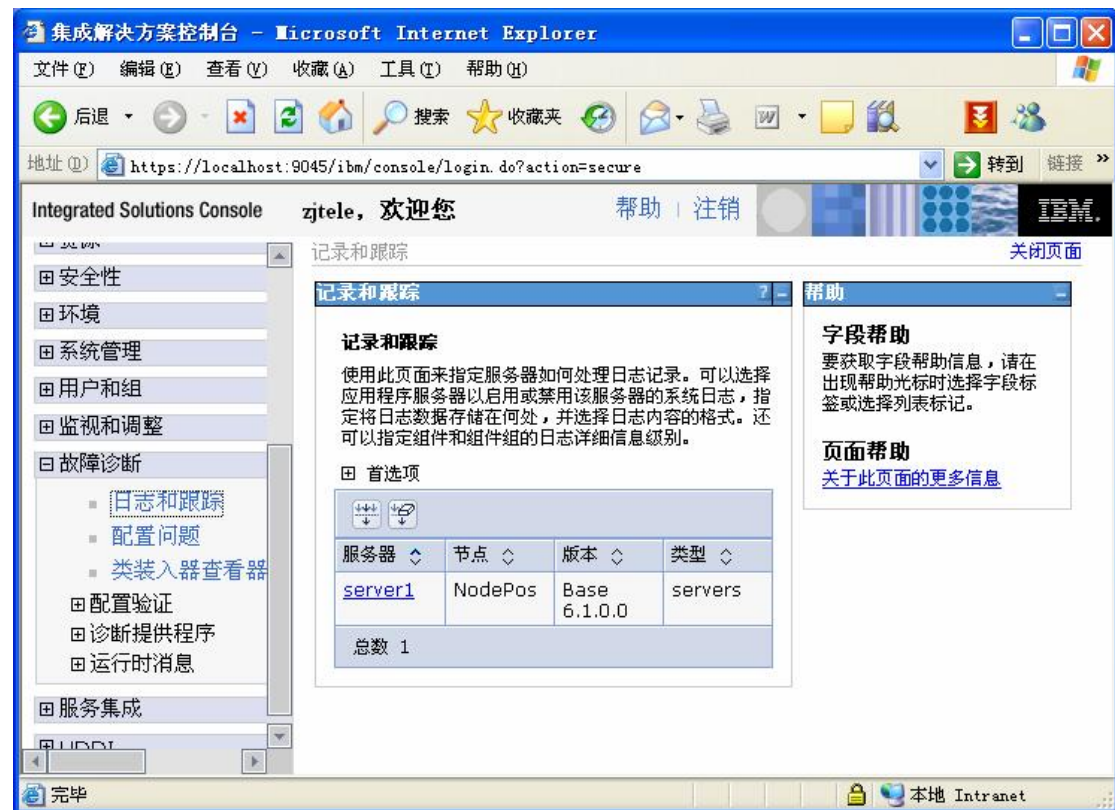
#### 四. Websphere 的控制台输出的设置

1. WAS6 按提示，安装完成之后，从程序组选择概要文件管理，创建新的概要文件。
2. 从安装的端口进入概要文件管理控制台：<https://localhost:9062/ibm/console/>  
跳转的 <https://localhost:9045/ibm/console/> 为安全管理控制台端口，是因为在安装的时候启用了安全管理，需要输入用户名标识才可以登录进行管理

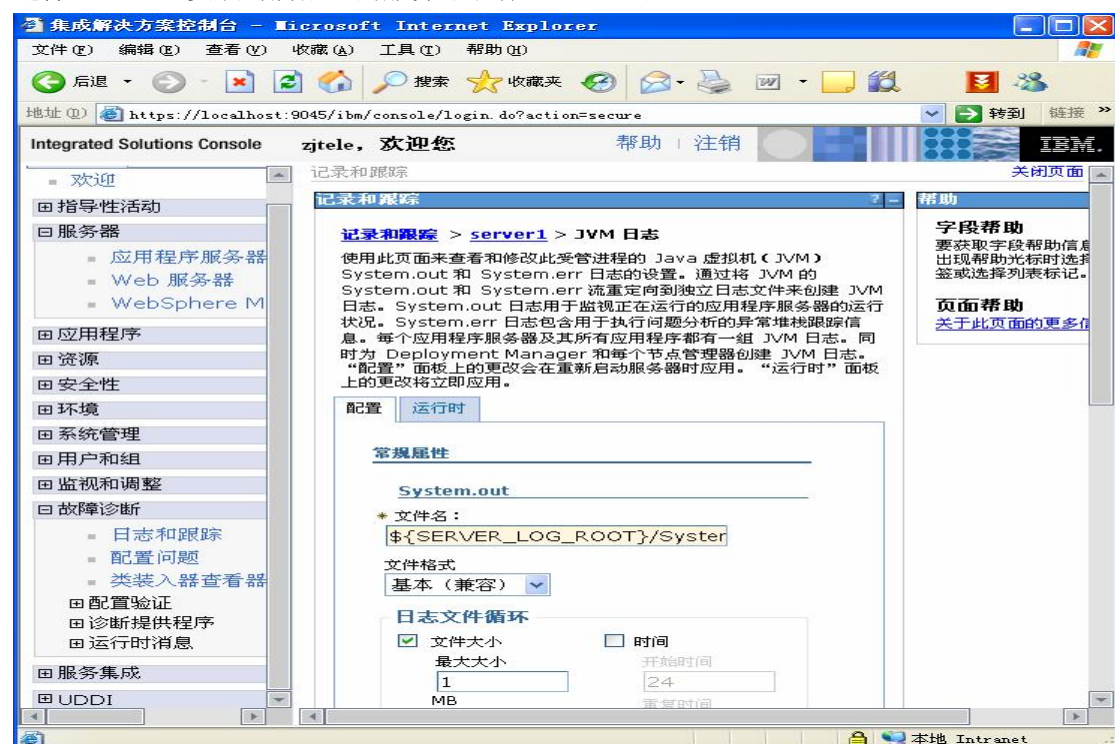


### 3. 设置日志输出选项：文件记录——>控制台

故障诊断——>日志和跟踪



选择 server1，安装时所配置的服务器名称——JVM 日志



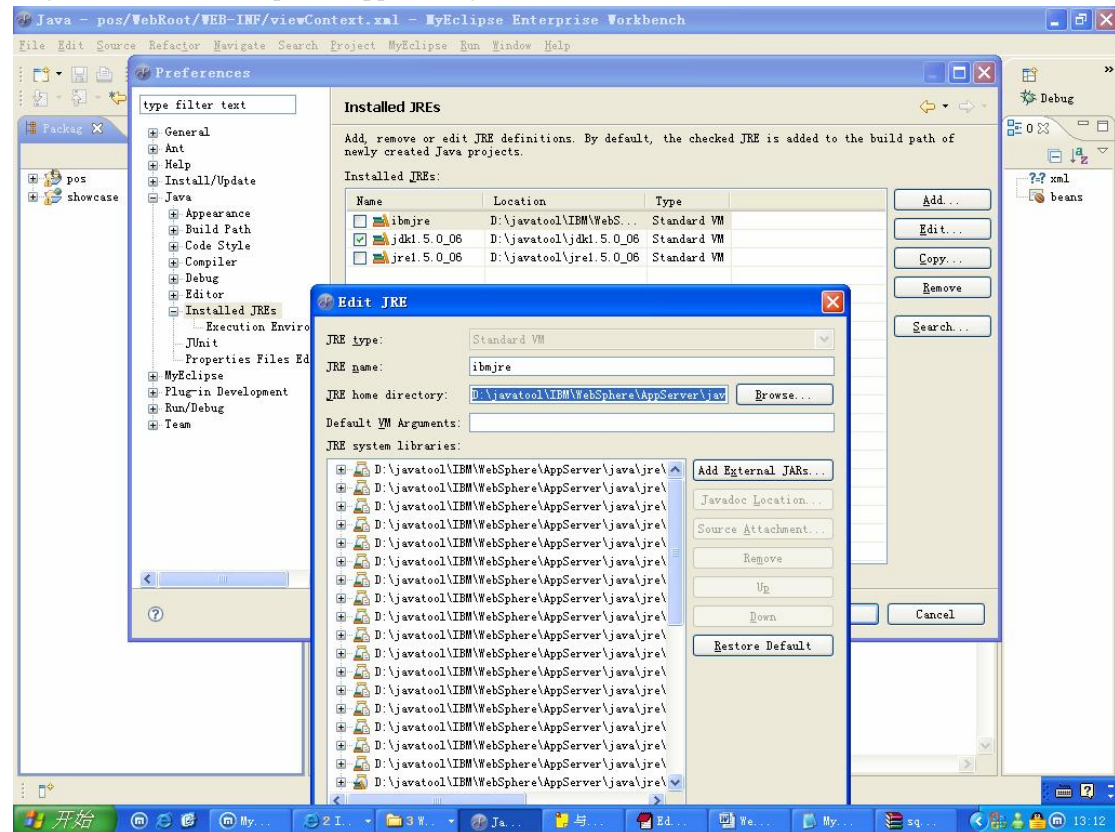


修改`${SERVER_LOG_ROOT}/SystemOut.log` 和 `${SERVER_LOG_ROOT}/SystemErr.log` 为 console

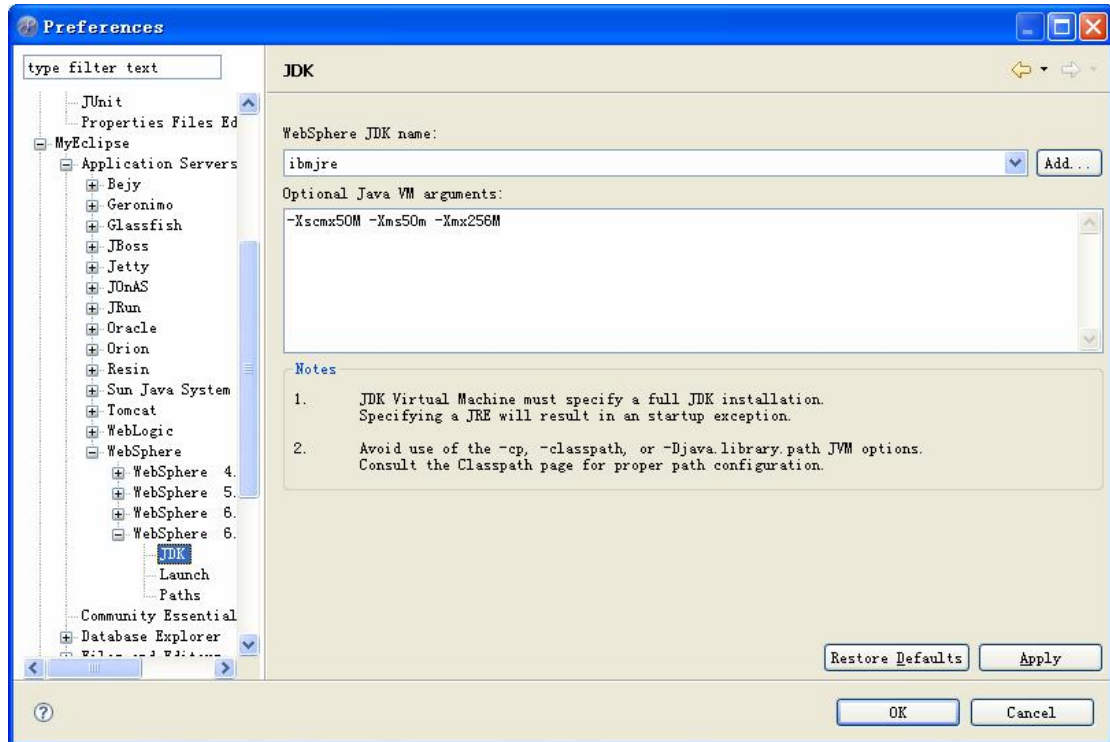
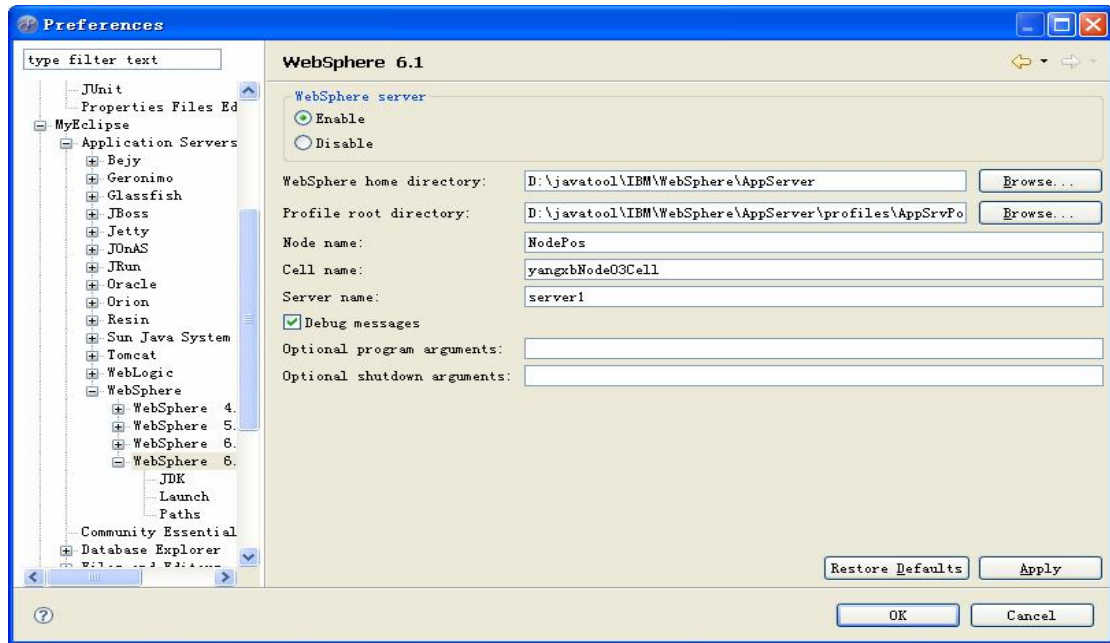
## 五、在 Myeclipse 下集成 Websphere 服务器

➤ 版本: Eclipse 3.3, MyEclipse 6.0.0.M1

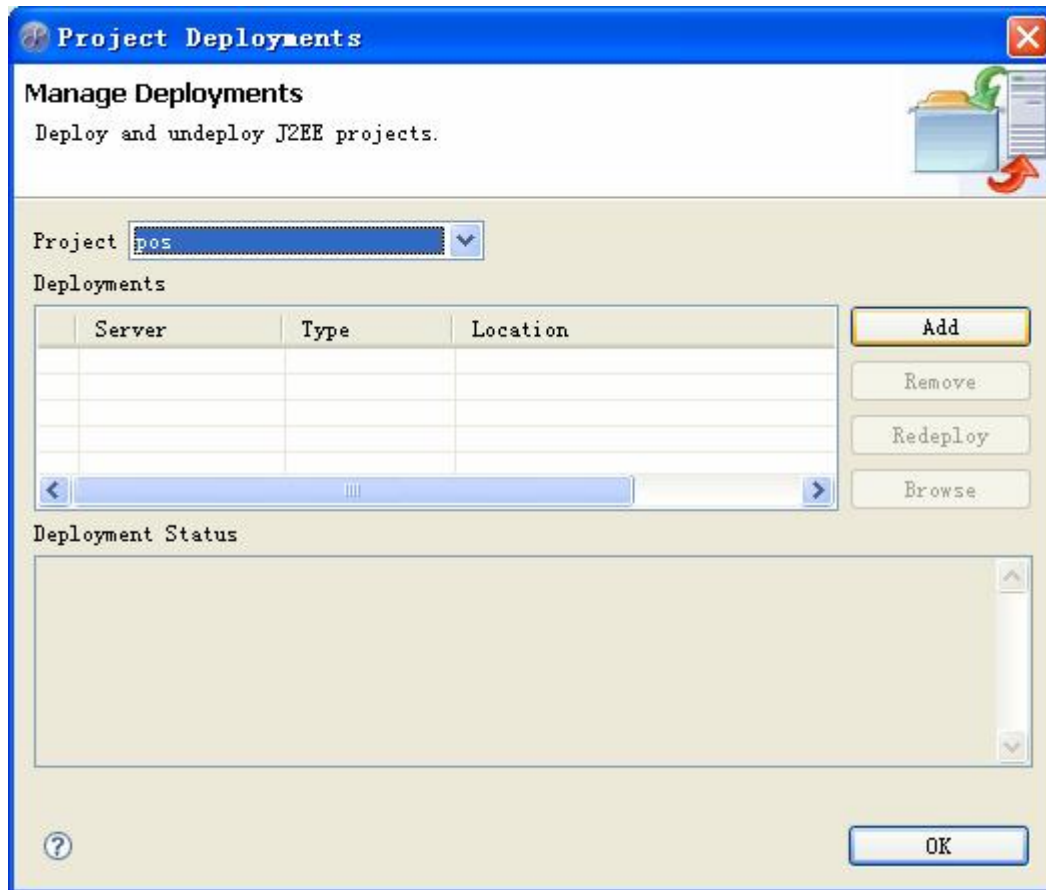
1. 启动 MyEclipse 进入 Windows – Preferences – Java – InstallJREs 添加 ibmjre 指向 `D:\javatool\IBM\WebSphere\AppServer\java`

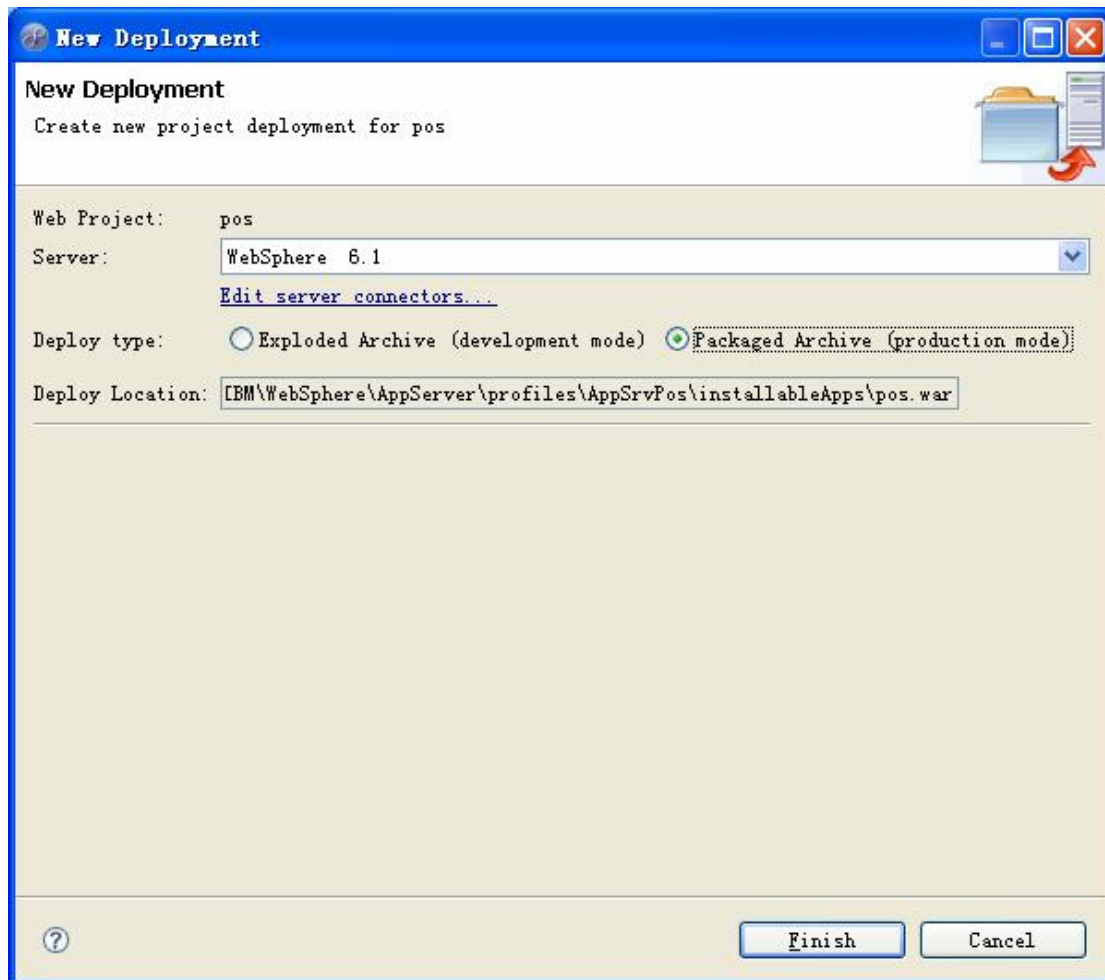


2. 设定 MyEclipse – ApplicationServer – WebSphrer 6.1

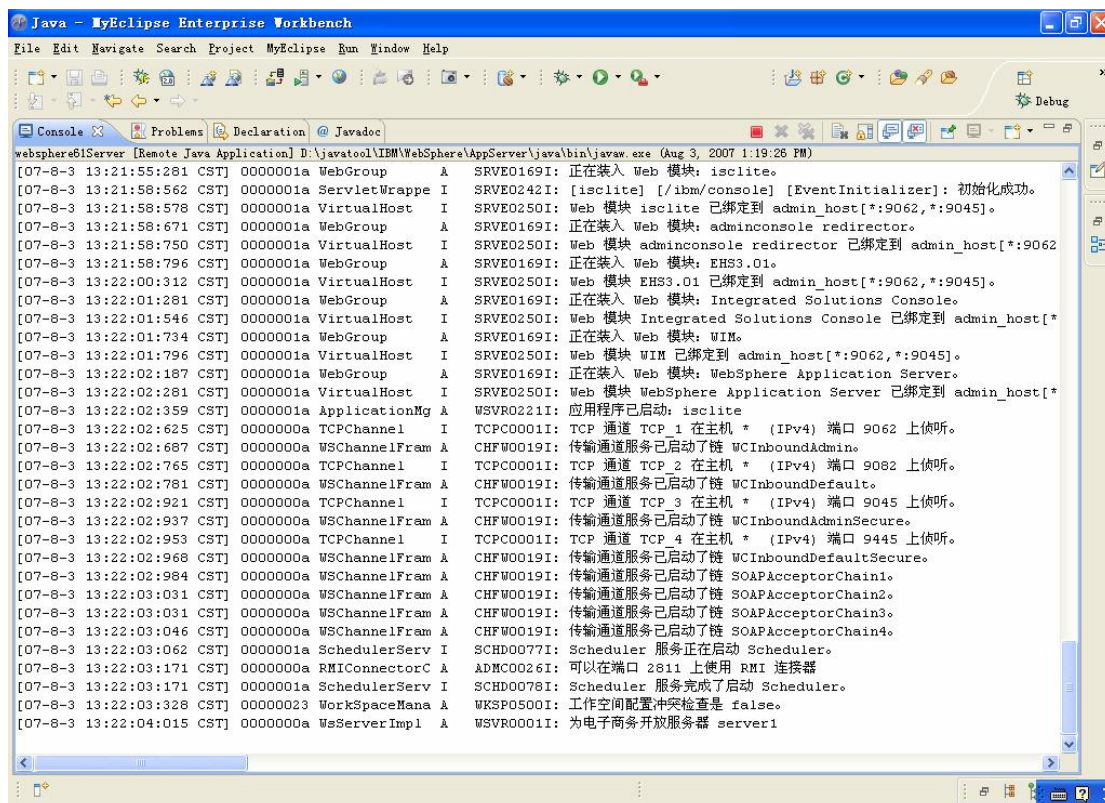


### 3. 建立发布项目路径



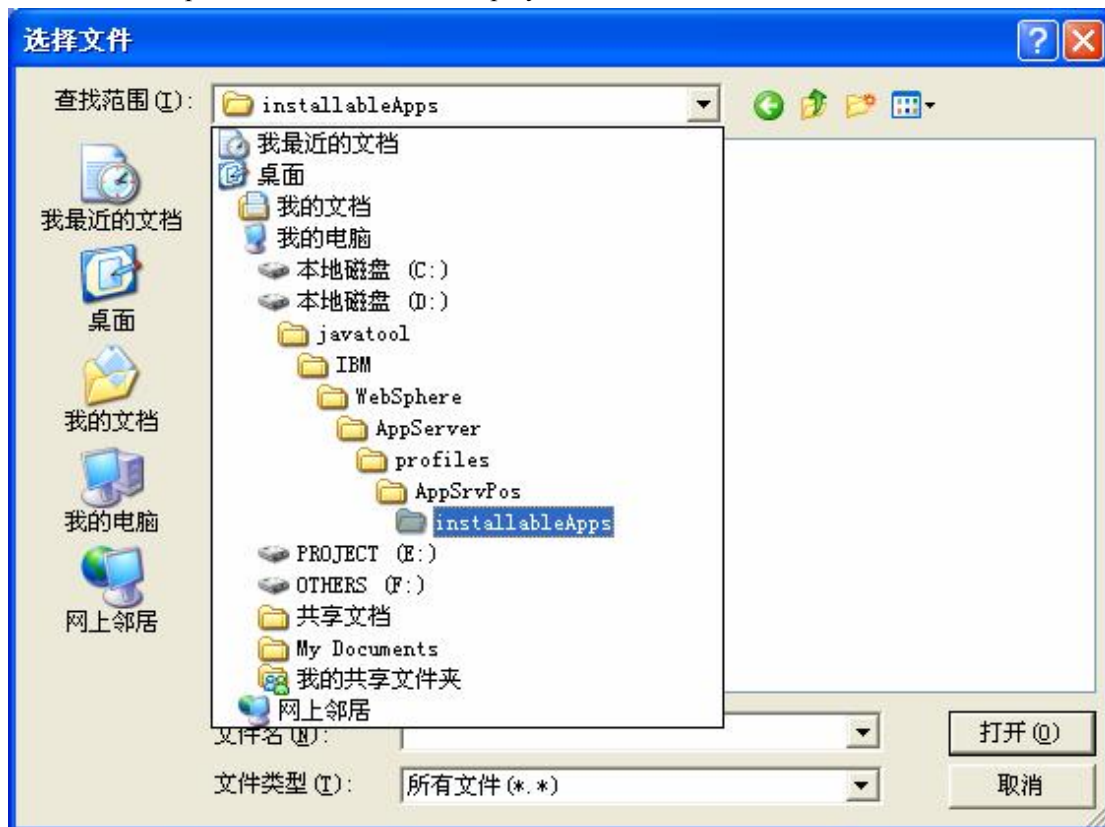


#### 4. 停止服务器并在 MyEclipse 里启动 WebSphere 6.1 server1

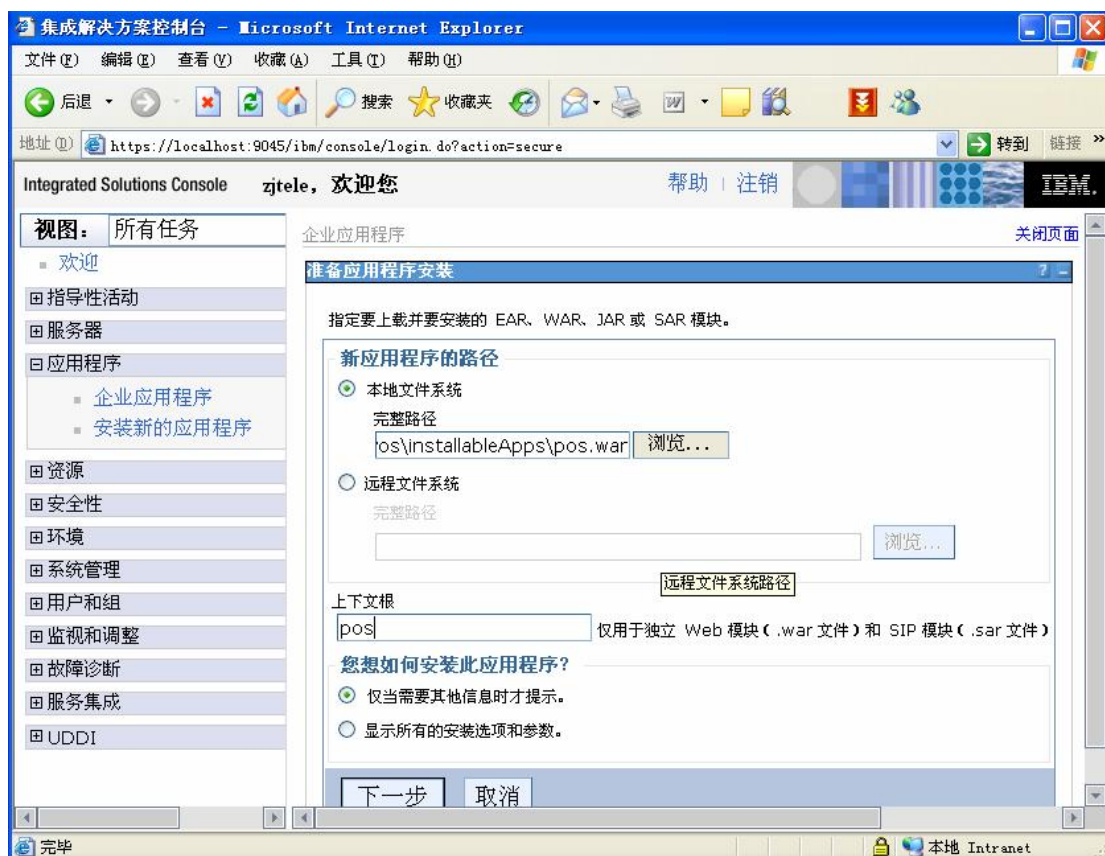




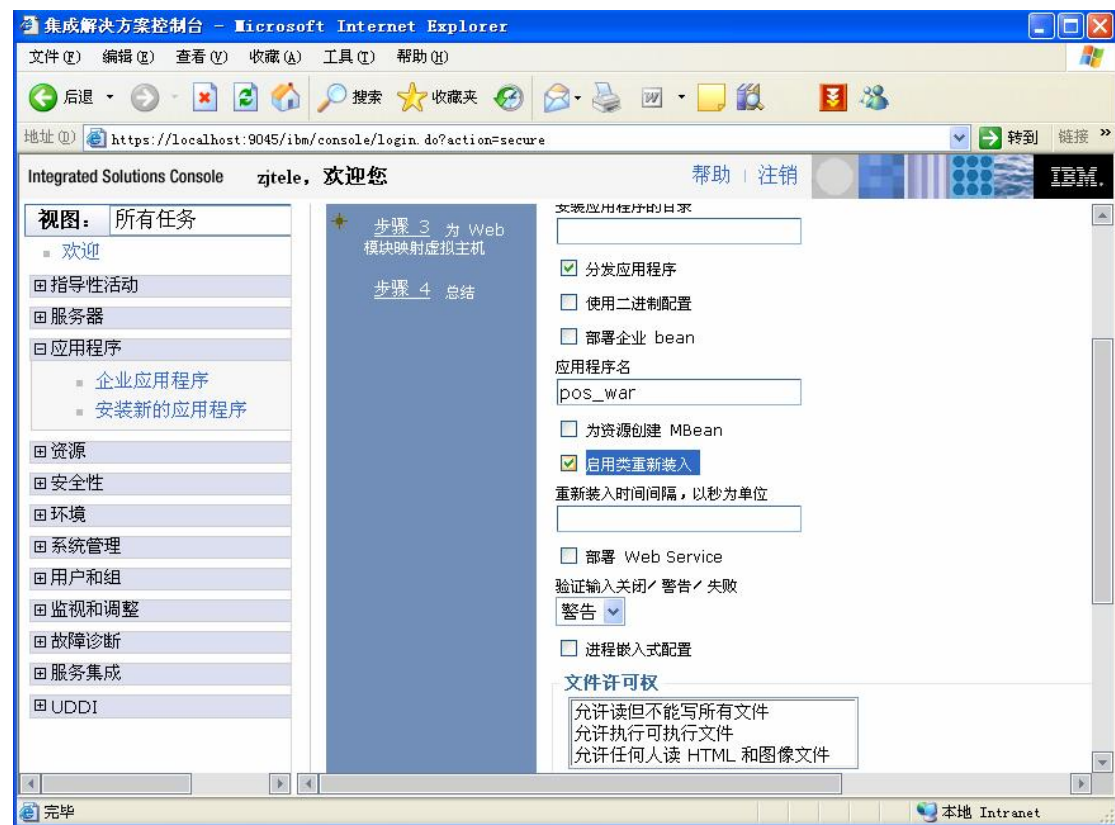
5. 进入 WebSphere 控制管理平台安装 Deploy 的企业应用程序，浏览本地文件



设定上下文



启用类重新装入



完成并发布让其启动

6. 打开浏览器浏览网页

7. 附加说明：在配置成功之后，删除第 3 步建立的项目发布，然后选择 Explored Archive 方式发布到 ibm 的 installedApp 目录下，那么在 MyEclipse 里面就可以进行实时调试进行开发了。