# WebLogic中间件

websphere jboss

## WebLogic介绍、安装

### 什么是中间件

中间件(middleware)是基础软件的一大类, 属于可复用软件的范畴. 顾名思义,中间件处于操作系统软件与用户的应用软件的中间.

中间件在操作系统、网络和数据库之上，应用软件的下层,总的作用是为处于自己上层的应用软件提供运行与开发的环境,帮助用户灵活、高效地开发和集成复杂的应用软件.在众多关于中间件的定义中,比较普遍被接受的是IDC表述的:**中间件是一种独立的系统软件或服务程序,分布式应用软件借助这种软件在不同的技术之间共享资源, 中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信**

### BEA公司介绍

著名的Java 中间件软件公司，美国NASDAQ上市公司，其中间件市场份额一度IBM还要高。2008年初被Oracle收购, 客户选择BEA来帮助其将已有的企业软件应用，从笨拙的、冗余的、遗留的客户机/服务器架构体系，发展成为响应灵敏的、成熟的Web基础结构

### 什么是WebLogic

WebLogic是美国Oracle公司出品的一个application server确切的说是一个基于JAVAEE架构的中间件，BEA WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。

### WebLogic与Tomcat比较

|  |  |
| --- | --- |
| **Weblogic特点** | **Tomcat特点** |
| WLS全面支持J2EE的标准规范和其他标准规范,Web Service, SSL, xml,EJB等 | Tomcat 为WEB容器，只支持部分J2EE标准, 应用局限性强,不支持EJB |
| 完善的售后支持 | 没有售后支持, 看社区与论坛 |
| 集群机制, 支持分布式的应用 | 需要结合第三方插件/应用 |
| Web控制台进行组件、JDBC、管理和配置 | 差 |
| 较好的支持热部署（开发模式下） | 差 |
| 需要费用 | 开源免费 |

### 什么是负载均衡/故障转移

在集群系统中. 来自客户的请求可以进行平均分配. 把相应的进程分发给与之共同承担任务的服务器，从而不影响应用进程的运行, 大多数中间件都支持负载均衡，实现负载均衡大大降低了系统的崩溃现象，从而减少对企业带来的损失.

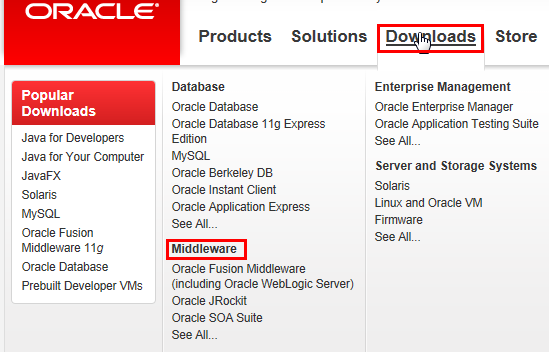
故障转移：软硬件出现故障，能够有其它相关的软硬件设备来承接相关的工作任务，以保障系统正常工作。

### 什么分布式

什么是分布式计算？所谓分布式计算是一门计算机科学，它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算机进行处理，最后把这些计算结果综合起来得到最终的结果。分布式网络存储技术是将数据分散的存储于多台独立的机器设备上。分布式网络存储系统采用可扩展的系统结构，利用多台存储服务器分担存储负荷，利用位置服务器定位存储信息，不但解决了传统集中式存储系统中单存储服务器的瓶颈问题，还提高了系统的可靠性、可用性和扩展性

### 下载WebLogic

* 打开Oracle官方网站: <http://www.oracle.com/index.html>



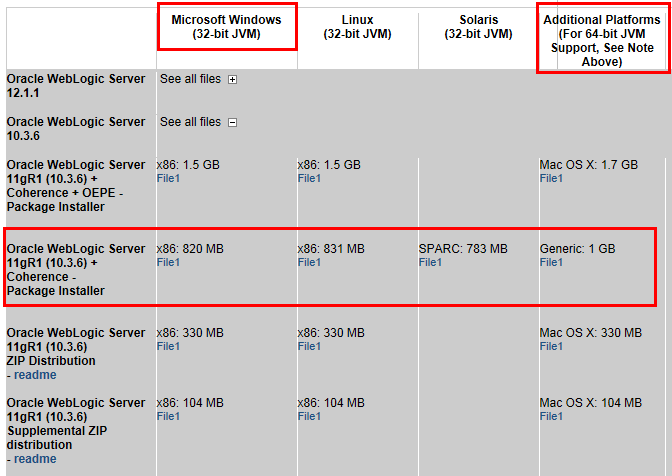
* 在中间件中找到Weblogic产品
* Mycat



* 下载最稳定的版本10.3.6



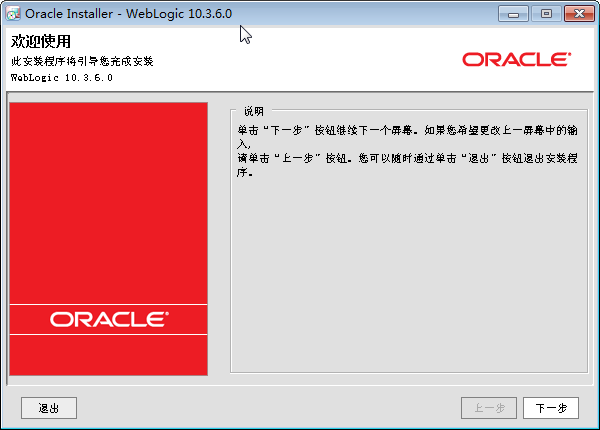
* 根据自己的系统平台选择下载32或者64版本



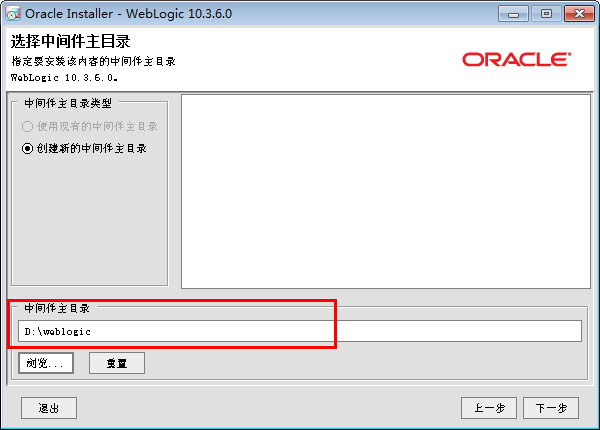
1. Oracle WebLogic Server 11gR1 (10.3.4) + Coherence + OEPE Package Installer – 此安装包为离线完整安装包, 可安装包括 Oracle WebLogic Server 组件, 以及 Oracle Coherence 和Oracle Enterprise Pack for Eclipse开发工具.
2. Oracle WebLogic Server 11gR1 (10.3.4) + Coherence Package Installer – 此安装包为离线完整安装包, 可安装包括 Oracle WebLogic Server 组件, 以及 Oracle Coherence
3. Oracle WebLogic Server 11gR1 (10.3.4) ZIP Installer 此ZIP压缩包格式仅用于WebLogic Server 开发. 它只包含了安装和运行WebLogic服务器的必要文件, 此包也不包含 Oracle Coherence 或者Oracle Enterprise pack for Eclipse

### WebLogic安装

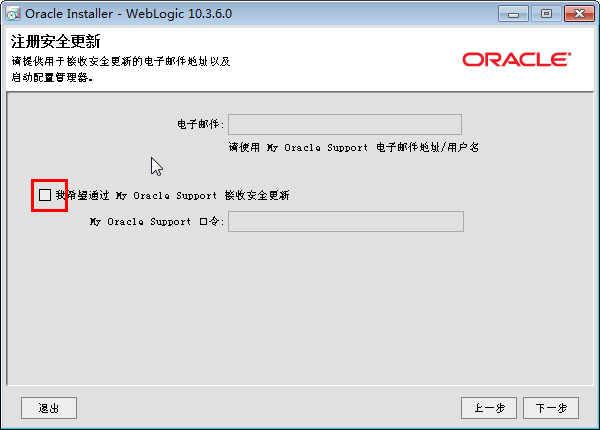
1. 执行：wls1036\_win32.exe 安装比较简单，典型的windows安装，一路"下一步"，即可安装截图如下



1. 选择要安装的目录建议安装到非系统盘



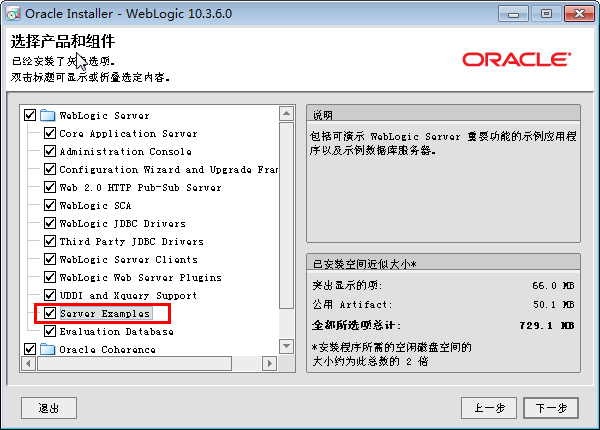
1. 取消自动更新选项



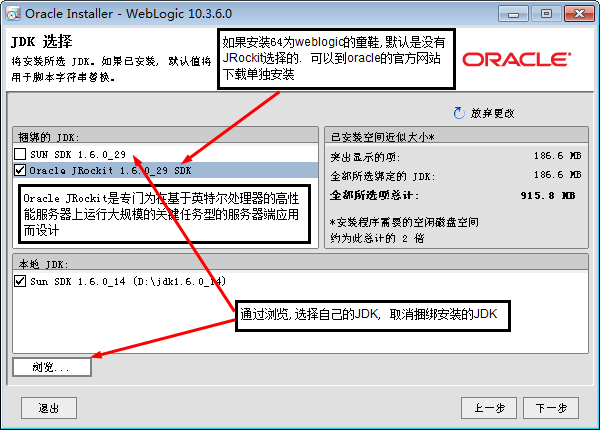
1. 选择定制安装



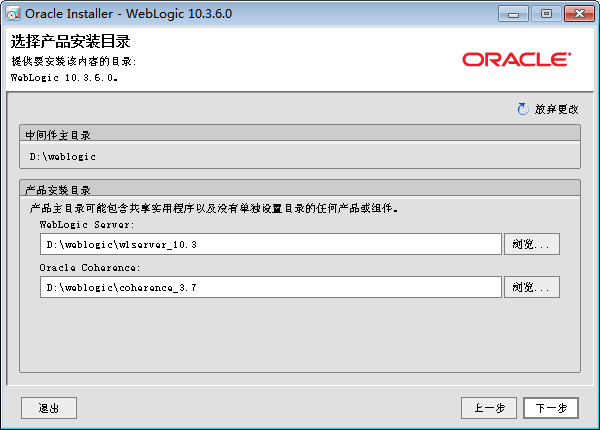
1. 选中安装Server Exampes



1. 选择自己的JDK, 和安装JRockit(可选)



1. 选择组件的产品安装目录



1. 安装服务与设置端口号



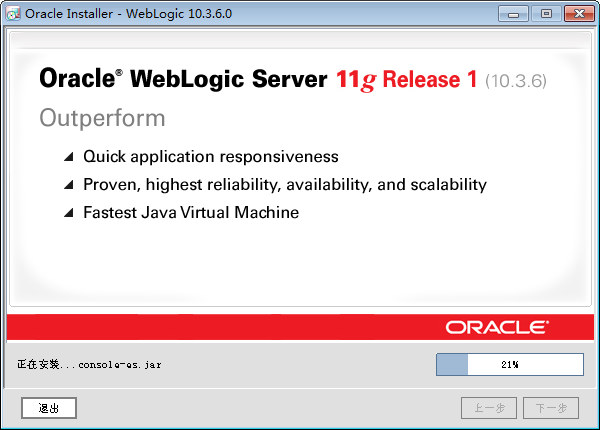
1. 选择是否创建快捷方式



1. 前面所设置的安装列表



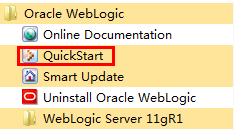
1. 安装进行中........



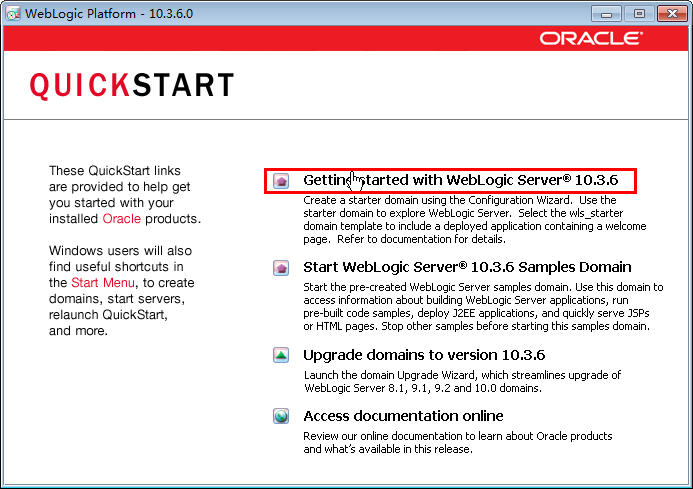
1. 安装完成, 也可以选择QuickStart方式启动,或者正常启动都可以



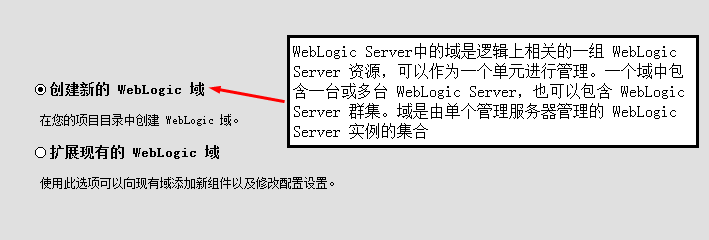
1. 通过快捷方式找到Weblogic菜单



1. 快速启动webLogic向导



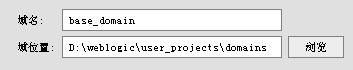
1. 创建一个WebLogic域



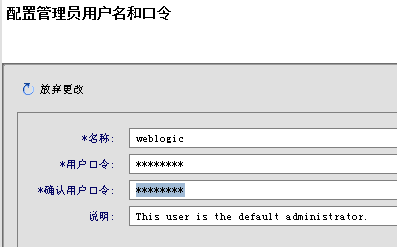
1. 选择创建的域支持那些产品服务



1. 定义域的名称与存放位置, 可以修改自己感兴趣的名称



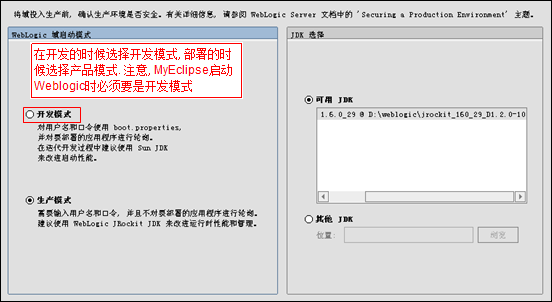
1. 配置管理员的用户名和口令



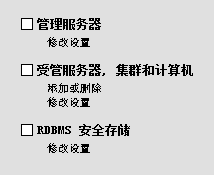
1. 选择weblogic域的启动模式

开发模式：该模式启用自动部署  
生产模式：该模式关闭自动部署(MyEcipse版本不支持产品模式)

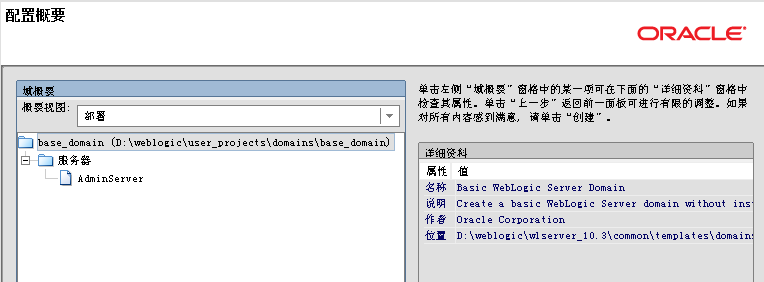
生产模式是为保证速度快放在指定目录下，他的包不会自动更新。开发模式是比较自由的，它保证开发灵活性，随便把它扔哪里都会自动更新，这样会影响速度。一般地开发好的产品，都给客户用产品模式部署



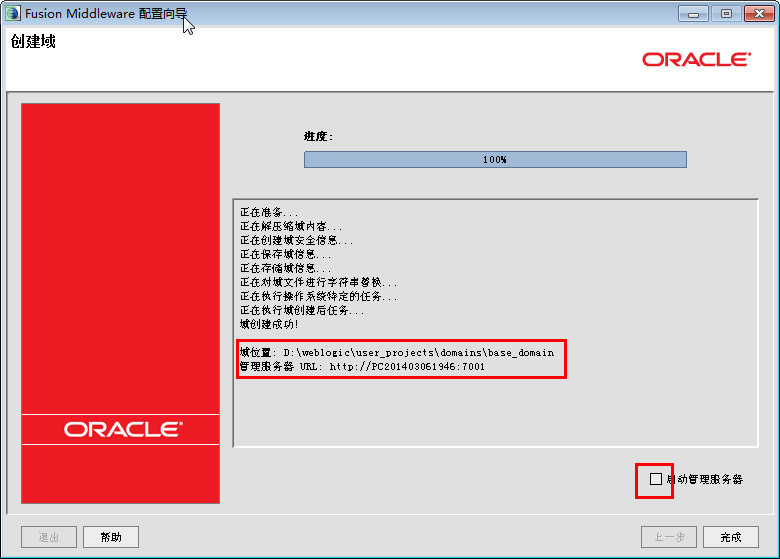
1. 选择可选配置



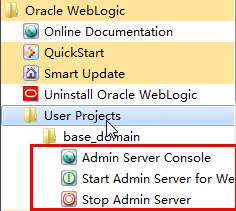
1. 显示了配置概要



1. 创建域的相关信息

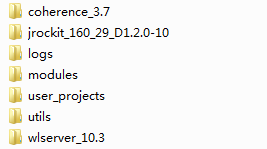


1. WebLogic启动,停止,管理页面登录



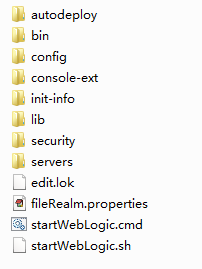
## Weblogic目录结构

### WebLogic基本目录介绍



* Coherence：在可靠的、高度可伸缩的对等集群协议之上提供了复制的、分布式的（分区的）数据管理和缓存服务
* Logs：系统日志目录
* Modules: weblogic运行时需要的jar文件
* user\_projects：定义域存放的位置
* Utils：用户可以存放的第三方工具包
* wlserver\_10.3：weblogic的主目录

### User\_projects目录介绍



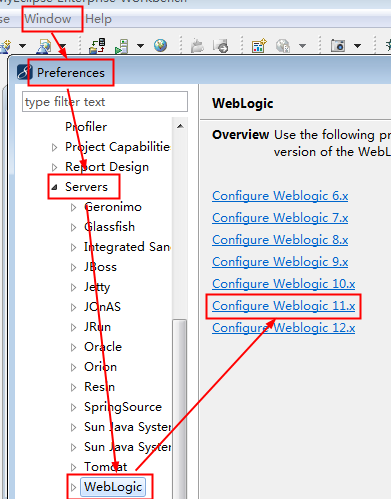
* Autodeploy：当Weblogic服务实例与开发模式运行时,此目录下应用程序(ear, war)等将被自动展开服务
* Bin：存放域中的可执行文件
* Config：本域相关的配置文件
* Console-ext：控制台信息, 只应用于adminServer
* Lib：存放库文件
* Security：安全相关
* Servers：包含域的所有服务
* StartWebLogic.cmd 调用bin下面的StartWebLogic.cmd
* StartWebLogic.sh：调用bin下面 StartWebLogic.sh (Linux下)

### Wlserver\_10.3

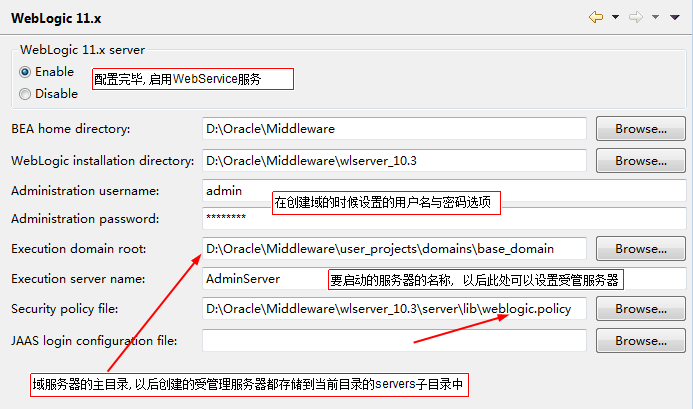
* Common:由产品组件共享的文件,包括计算机上运行的所有WebLogic域所公用的环境属性, 创建域提供Configuration Wizard和WLST脱机使用的模板JAR文件
* Server：WebLogic Server程序文件
* Uninstall:卸载webLogic产品目录

## MyEclipse配置WebLogic

### 开启Weblogic服务



### 配置Weblogic与Tomcat相同



### 项目部署到WebLogic中

* 目前Weblogic版本仅仅支持J2EE5.0 因此要把web.xml中的配置文件改成2.5规范

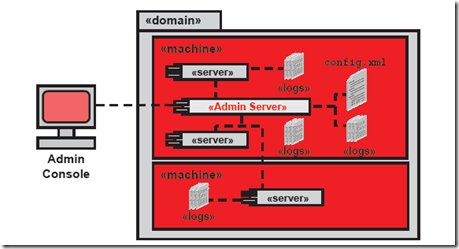
|  |
| --- |
| <web-app version=*"2.5"*  xmlns=*"http://java.sun.com/xml/ns/javaee"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*  *http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*> |

* 如果是开发模式，则可以测试热部署的执行情况

## Weblogic概念介绍

### 域Domain

WebLogic Server中的域是逻辑上相关的一组 WebLogic Server 资源，可以作为一个单元进行管理。一个域中包含一台或多台 WebLogic Server，也可以包含 WebLogic Server 群集。域是由单个管理服务器管理的 WebLogic Server 实例的集合



### Managed Server

被管理服务器也是一个WebLogic Server的实例，他从Admin Server那里获取配置信息。通常在Managed Server上部署自己的服务、组件或应用。基于性能考虑，Managed Server维护着一份只读的域配置文件，当服务启动的时候，会从Admin Server上同步配置文件信息，如果配置文件发生了改变，Admin Server也会将改变推送到Managed Server上

### Administrator Server

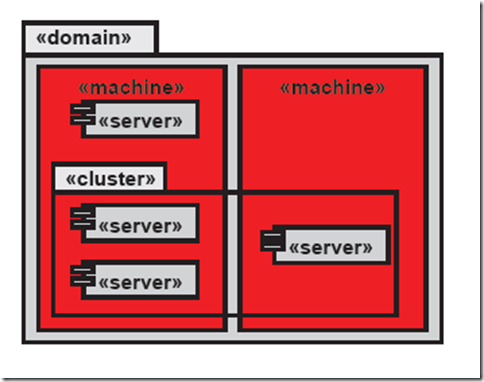
管理服务器是控制整个域配置的中心操作节点，管理服务器维护着整个域 Domain 的配置并将配置分配到每个被管理服务器 Managed Server 中,每个域中都必须有一个Administration Server

### 计算机Machine

是指运行WebLogic Server实例的物理机器，通常与部属在其上的Managed Server相关连。Machine用于Node Manager（一个运行在物理服务器上的进程)

### Cluster集群

集群是一组WebLogic Server的组合，集群是为了满足服务的高可用（High Avilability）以及可扩展（Load Balancing）需求而出现的。需要注意的是，集群中的WebLogic Server必须版本号一致。Cluster需要注意的几个问题：不能够跨Domain、Cluster中的服务器必须在同一个域中、Domain中可以有多个Cluster



### 启动与服务相关配置

WebLogic的启动是通过启动文件来完成的,不同的服务定义在不通的启动配置文件中,都在域的bin目录中

* StartWebLogic：启动管理服务器
* StartManagedWebLogic：启动被管理服务器
* StopWebLogic：关闭管理服务器
* StopManagerWebLogic：关闭被管服务器

系统配置文件config.xml 存放这域内所有的配置信息,该文件的存放在域中的config目录的config.xml中,修改后端口信息就保存在此文件中

### WebLogic生命周期

* SHUTDOWN 状态下，WebLogic Server 实例已配置但处于非活动状态
* STARTING 状态期间，由于执行"启动"、"管理模式启动"或"以待机模式启动"命令，WebLogic Server 实例从 SHUTDOWN 转换为 STANDBY
* STANDBY 状态的服务器实例不处理任何请求，它的常用监听端口处于关闭状态。管理端口处于打开状态，可以接受将服务器实例转换为 RUNNING 或 SHUTDOWN 状态的生命周期命令
* 在 ADMIN 状态下，WebLogic Server 启动并运行，但仅用于管理操作，您可以在此状态下执行服务器级和应用程序级管理任务
* 在此转换状态中，WebLogic Server 执行将其从 STANDBY 或 ADMIN 状态转换为 RUNNING 状态所需要的操作
* 在 RUNNING 状态下，WebLogic Server 处于完全工作状态，可以向客户端提供服务并作为一个完整的群集成员运行

## 创建被管服务器、MSI、节点管理器

### 创建一个被管理服务

* 启动WebServie主服务,登录到后台控制器



* 填写受管理服务的相关信息



* 创建好的受管理服务,默认停止的,要启动它



* 通过：StartManagedWebLogic.cmd 被管理服务名 [http://主机名:](http://AdminService主机名:)主服务端口号来启动被管理服务在base\_domain\servers\ 会新增加一个Server-0服务
* 注意如果被管服务器启动提示内存不足,则可以配置启动内存：单击服务器--->
* 服务器启动-->参数 -Xms512m -Xmx512m -XX:MaxPermSize=512m

### MSI介绍

* Managed Server Independence受管服务器独立性：受管服务器拥有一份域配置信息的副本.当受管服务器启动的时候,它会从管理服务器获取到配置信息.如果受管服务器启动的时候不能连上管理服务器. 就使用本地的缓存配置信息启动.这就是MSI模式. MSI模式默认开启的
* 如何开启MSI功能：选中”受管服务器”--->”优化”--->”高级”---->”受管服务器独立性”
* 在MSI模式下, 被管服务器：自动寻找本地config目录下的config.xml文件.寻找本地security目录下的SerializedSystemlni.dat文件和boot.Properties文件

### 使用WLST工具

* WebLogic Scripting Tool (WSLT)是一个命令脚本工具, 可以用来创建、管理、监控WebLogic域.它基于Jython语法编写的可以在：
* WL\_HOME\common\bin\wlst.cmd 中启动WLST,也可以在开始菜单启动WLST
* Online模式(使用WLST连接正在运行的管理服务器)：管理活动域的配置、查看域中的性能数据、管理安全数据(例如添加删除用户等......)
* 使用WLST连接被管理服务器,但是不能通过被管理的服务器修改配置
* Offline模式(使用WLST没有连接管理服务器)：可以创建域模板、创建域、扩展活动域等等
* netstat -a -n // 查看网络端口
* help('all') // 查看所有命令帮助
* help('connect') // 查看connect命令的帮助
* connect ('weblogic','weblogic123','t3://localhost:7001')
* connect ('weblogic','weblogic123')
* disconnect() ls() 查看 exit()
* State('AdminServer'): 目前是RUNNING状态
* 挂起：suspend() 恢复：suspend() 关闭：shutdown()

## 集群配置

### 创建两台被管理服务器Server-0,Server-1

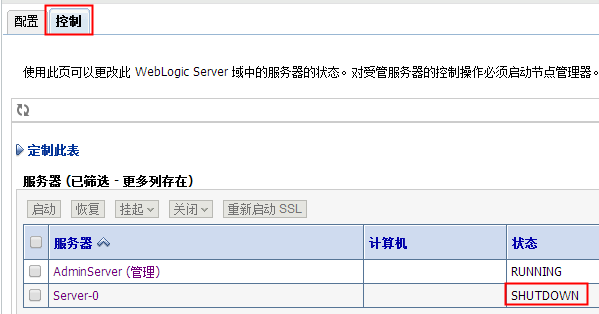
* 选中”服务器”--->新建



* 设置服务器名称、IP地址

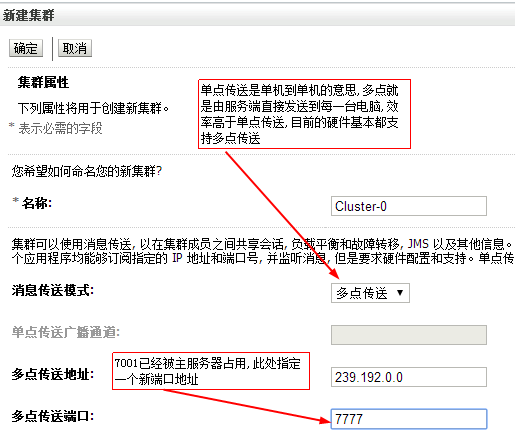


* 默认是SHUTDOWN默认,可以通过命令启动被管服务器



### 创建集群

* 新创建一个集群, 配置集群的名称与传送端口

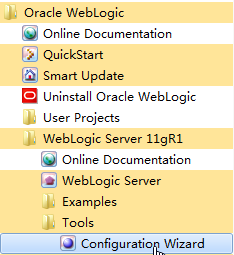


* 把Server-0,1添加到集群中



### 创建代理服务器Proxy(必须要先创建代理服务器)

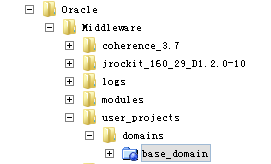
* 首先在”程序菜单”打开代理服务器向导 Configuraction Wizard



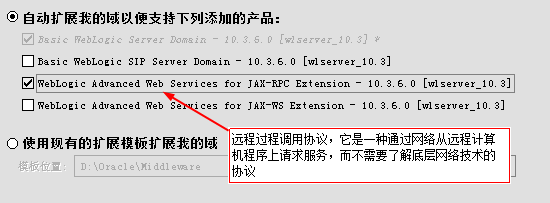
* 选择扩展现有的Weblogic域, 我们要给被集群的服务器创建一个代理程序



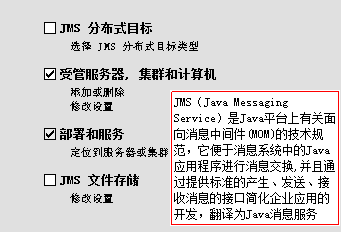
* 选中要扩展的域,此处选中base\_doamin



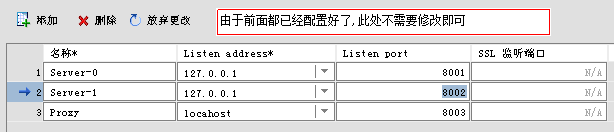
* 选中代理的方式 JAX-RPC, 就是基于XML的远程过程调用协议, JAX-WS是基于XML的WebService服务. 此处我们需要通过XML配置被代理服务器的IP信息, 然后代理服务器吧请求获取后交给被代理服务器.



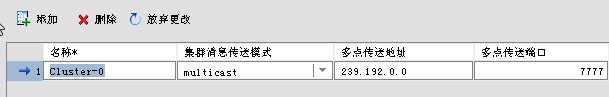
* 选中需要添加的功能 2和3选项



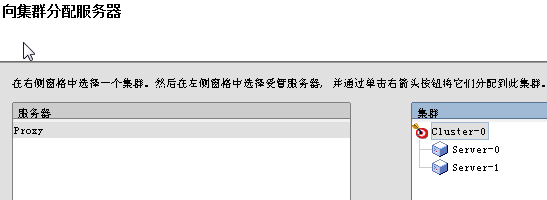
* 如果需要修改端口则可以单击修改



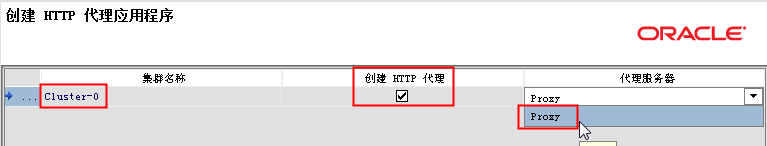
* 如果需要修改集群的配置则单击修改



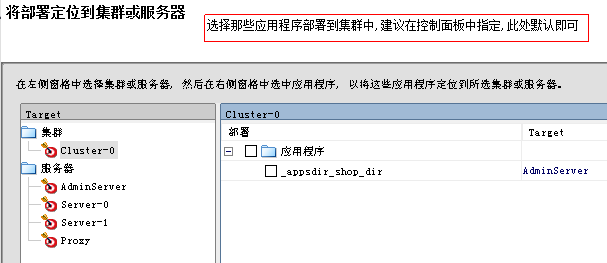
* 配置集群中的受管服务器, 此处如果遵循前面的配置即可



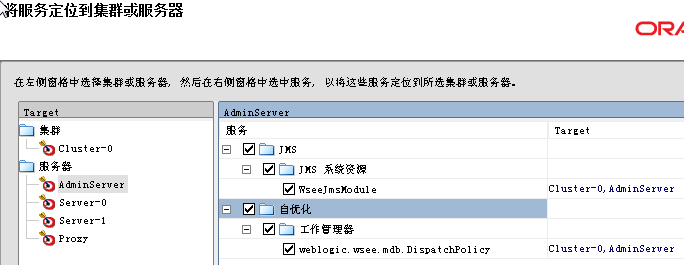
* 创建HTTP代理服务器,对请求进行转发



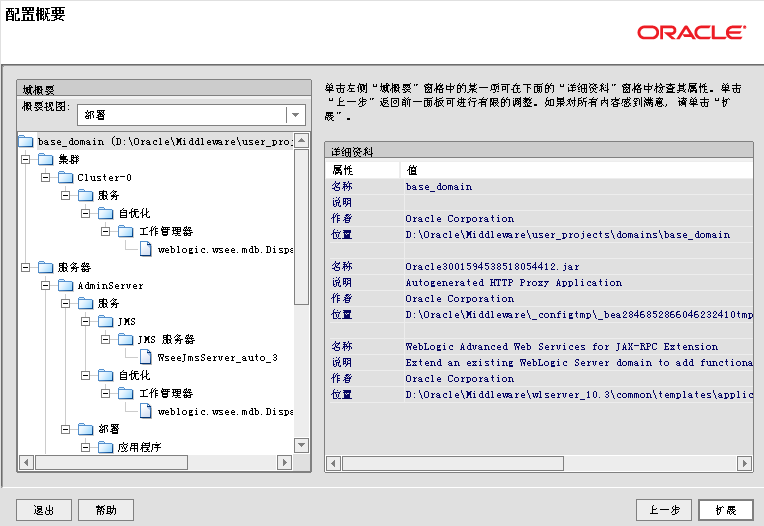
* 询问是否添加计算机，没有计算机，就不用添加
* 选择把项目部署到集群中



* 将服务定位到集群或者服务器



* 显示了前面配置概要



* 最后会自动生成代理项目路径为: user\_projects\domains\base\_domain\apps

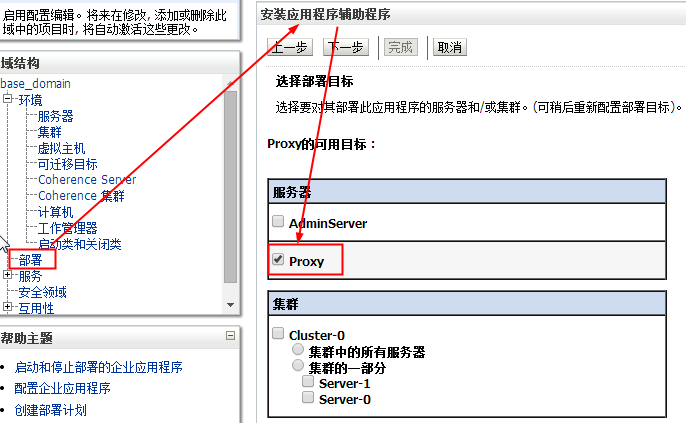
Web.xml中配置如下：

|  |
| --- |
| <!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app\_2\_3.dtd">  <web-app>  <!--HttpClusterServlet来拦截请求,然后转发给被代理的服务器 -->  <servlet>  <servlet-name>HttpClusterServlet</servlet-name>  <servlet-class>weblogic.servlet.proxy.HttpClusterServlet</servlet-class>  <!-- 配置了集群的服务器, 服务器与服务器之间用 |隔开 -->  <init-param>  <param-name>WebLogicCluster</param-name>  <param-value>127.0.0.1:8001|127.0.0.1:8002</param-value>  </init-param>  </servlet>  <!-- 配置了需要拦截的 路径,与后缀,根据情况自行添加 -->  <servlet-mapping>  <servlet-name>HttpClusterServlet</servlet-name>  <url-pattern>/</url-pattern>  </servlet-mapping>  <servlet-mapping>  <servlet-name>HttpClusterServlet</servlet-name>  <url-pattern>\*.jsp</url-pattern>  </servlet-mapping>  <servlet-mapping>  <servlet-name>HttpClusterServlet</servlet-name>  <url-pattern>\*.jspf</url-pattern>  </servlet-mapping>  <servlet-mapping>  <servlet-name>HttpClusterServlet</servlet-name>  <url-pattern>\*.htm</url-pattern>  </servlet-mapping>  <servlet-mapping>  <servlet-name>HttpClusterServlet</servlet-name>  <url-pattern>\*.html</url-pattern>  </servlet-mapping>  </web-app> |

* Weblogic.xml中配置访问项目的前缀

|  |
| --- |
| <!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application 8.1//EN" "http://www.bea.com/servers/wls810/dtd/weblogic810-web-jar.dtd">  <weblogic-web-app>  <!-- 配置访问项目的前缀 -->  <context-root>/项目URL</context-root>  </weblogic-web-app> |

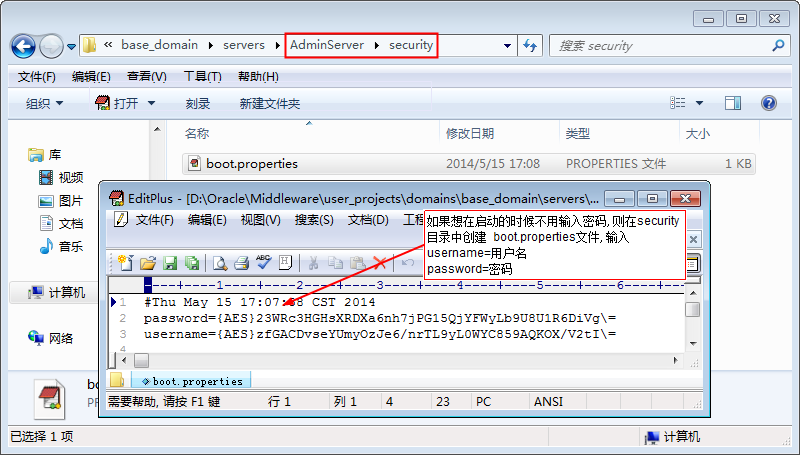
* 最后把生成的项目部署到代理服务器, 测试即可



## 其它主要事项

### 关于密码设置于修改

* 在启动的时候如何没有提示输入密码,则说明密码已经存储到当前服务security目录下



* 如何修改密码

