



目 录

目 录.....	2
第 5 章 WebLogic 配置相关文件.....	4
5.1 启动服务相关的几个文件.....	4
5.1.1 startWebLogic.cmd/startWebLogic.sh.....	4
5.1.2 startManagedWebLogic.cmd/startManagedWebLogic.sh.....	4
5.1.3 setDomainEnv.cmd/setDomainEnv.sh.....	4
5.1.4 stopWebLogic.cmd/stopWebLogic.sh.....	4
5.1.5 stopManagedWebLogic.cmd/stopManagedWebLogic.sh.....	5
5.2 系统配置文件 config.xml.....	5
5.2.1 关于 WebLogic 管理和 config.xml 文件概述.....	5
5.2.2 何时去编辑 config.xml.....	5
5.2.3 config.xml 文件里的内容.....	6
5.3 属性文件 weblogic.xml.....	6
5.3.1 说明.....	6
5.3.2 可配置的属性详解.....	6
description.....	6
weblogic-version.....	6
security-role-assignment.....	6
run-as-role-assignment.....	7
resource-description.....	8
resource-env-description.....	8
ejb-reference-description.....	8
service-reference-description.....	9
session-descriptor.....	9
jsp-descriptor.....	14
auth-filter.....	16
charset-params.....	16
input-charset.....	17
charset-mapping.....	17
virtual-directory-mapping.....	17
url-match-map.....	18
security-permission.....	19
context-root.....	19
wl-dispatch-policy.....	19
servlet-descriptor.....	20
work-manager.....	20
logging.....	22
library-ref.....	24
5.3.2.1 Backwards Compatibility Flags.....	25
5.3.2.2 Web Container Global Configuration.....	25
5.4 属性文件 web.xml.....	25
5.4.1 说明.....	25
5.4.2 可配置的属性详解.....	25
icon.....	25
display-name.....	26
description.....	26
context-param.....	26
filter.....	26
filter-mapping.....	27
listener.....	27
servlet.....	27
servlet-mapping.....	28

session-config.....	28
mime-mapping.....	29
welcome-file-list.....	29
error-page.....	29
taglib.....	30
resource-env-ref.....	30
resource-ref.....	30
security-constraint.....	31
web-resource-collection.....	31
auth-constraint.....	32
user-data-constraint.....	32
login-config.....	32
form-login-config.....	33
security-role.....	33
env-entry.....	33
ejb-ref.....	34
ejb-local-ref.....	34
5.5 日志文件.....	35
5.5.1 域日志(domain.log).....	35
5.5.2 server.log.....	35
5.5.3 access.log.....	35

Beijing Landing Technologies

第 5 章 WebLogic 配置相关文件

在梳理完 WebLogic 相对比较庞大的主要目录结构后，我们对其繁复的配置文件做一个归纳整理。

5.1 启动服务相关的几个文件

WebLogic 的启动是通过启动文件来完成的，包括启动管理服务器(startWebLogic)、启动被管服务器(startManagedWebLogic)、设置域环境(setDomainEnv)、关闭管理服务器(stopWebLogic)、关闭被管服务器(stopManagedWebLogic)等。

5.1.1 startWebLogic.cmd/startWebLogic.sh

1. 说明

startWebLogic.cmd (Windows 环境下的启动文件)

startWebLogic.sh (Linux/Unix 环境下的启动文件)

2. 文件位置

/WLS_HOME/user_projects/domains/base_domain (默认安装目录，可更改)

启动文件的最外层目录，调用

/WLS_HOME/user_projects/domains/base_domain/bin (默认安装目录，可更改) 目录下的 startWebLogic.cmd/startWebLogic.sh。

5.1.2 startManagedWebLogic.cmd/startManagedWebLogic.sh

1. 说明

startManagedWebLogic.cmd (Windows 环境下的启动文件)

startManagedWebLogic.sh (Linux/Unix 环境下的启动文件)

2. 文件位置

/WLS_HOME/user_projects/domains/base_domain/bin

5.1.3 setDomainEnv.cmd/setDomainEnv.sh

1. 说明

setDomainEnv.cmd (Windows 环境下的启动文件)

setDomainEnv.sh (Linux/Unix 环境下的启动文件)

启动管理和被管服务器时的参数信息记录在此文件里。例如：设置堆内存的最小值和最大值-Xms256m -Xmx512m, Sun JDK 和 HP JDK 的 perm 区大小-XX:PermSize=48m -XX:MaxPermSize=128m。

2. 文件位置

/WLS_HOME/user_projects/domains/base_domain/bin

5.1.4 stopWebLogic.cmd/stopWebLogic.sh

1. 说明

stopWebLogic.cmd (windows 环境下的启动文件)

stopWebLogic.sh(Linux/Unix 环境下的启动文件)

2. 文件位置

/WLS_HOME/user_projects/domains/base_domain/bin

5.1.5 stopManagedWebLogic.cmd/stopManagedWebLogic.sh

1. 说明

stopManagedWebLogic.cmd(windows 环境下的启动文件)

stopManagedWebLogic.sh(Linux/Unix 环境下的启动文件)

2. 文件位置

/WLS_HOME/user_projects/domains/base_domain/bin

5.2 系统配置文件 config.xml

系统配置文件 config.xml 内存放着域内所有配置信息，该文件的存放位置为：
/WLS_HOME/user_projects/domains/base_domain/config/config.xml(默认安装目录，其中域的名字以及路径可任意选择)。

5.2.1 关于 WebLogic 管理和 config.xml 文件概述

WebLogic 的管理和配置服务是基于 Sun 公司的 Java™ Management Extensions (JMX) API 的。创建 config.xml 文件的目的是存储管理对象的变更信息，以便在确保在 weblogic 重新启动时这些对象仍然可用。

通常应使用管理控制台来配置 WebLogic Server 的管理对象和服务以及允许 WebLogic Server 来维持 config.xml 文件。每次通过管理控制台或其他 weblogic server 工具修改 config.xml 文件，weblogic 都会把他归档到一个旧文件中，你可以配置 WebLogic 保存的归档文件的个数。

虽然 config.xml 是一个良好的 XML 文档，您可以使用文本编辑器修改，你应该把它看作一个数据库，你只会在特殊情况下直接更新。该文件并不是一个正式的 XML 内容文档，它实际上是一个库，其中每个 XML 元素包含了一个在 WebLogic Server 管理对象在内存中实例的请求保存的数据。

5.2.2 何时去编辑 config.xml

只有在很少的情况下您会选择直接修改文件而不是通过管理控制台。在这些情况下您要保证所的改变遵循最小化明确化。您不能试图通过写一个新的 config.xml 文件来创建一个新的 WebLogic 配置。

警告：您不能在 WebLogic 运行期间修改 config.xml，因为 WebLogic 会定期重写 config.xml 文件。您的更改将会丢失并且根据您平台的不同可能会导致 WebLogic 服务的失败。在任何情况下您应该先保留一份 config.xml 的副本，再去修改文件。

这里有一些情况适合您直接修改 config.xml 文件：

- 如果您要部署多个 WebLogic Server 实例，您可以“克隆”一个 config.xml 文件并编辑新的服务器上为每个值；
- 如果您有一台服务器上定义一个对象，并希望将其复制到另一台服务器，您可以复制在 config.xml 文件中定义的 xml 元素到另一个 config.xml 文件中；

- 为了帮助您更好的解决问题，WebLogic 的技术支持可能会建议您在 config.xml 文件中设置一些不能通过管理控制台设定的属性；
- 一些第三方应用程序文件可能需要您修改 config.xml 文件。

5.2.3 config.xml 文件里的内容

config.xml 文件中包含了一系列的 xml 元素。其中域元素的顶层元素，域中所有元素都是域元素的子集。域元素包括服务器、集群、应用等子元素，这些元素可能还有其自己的子元素。例如：服务器的子元素包含 Web 服务器，SSL 和日志等，应用程序的子元素包含 EJB 组件和 Web 应用程序组件等。

每个元素都有一个或多个可配置的属性。一个属性有一个配置 API 中相应的属性。例如，服务器元素有一个 ListenPort 属性，同样，`weblogic.management.configuration.ServerMBean` 类有一个 ListenPort 属性。可配置的属性是可读可写，也就是 `ServerMBean` 的 `getListenPort()` 和 `setListenPort()` 方法。

所有属性的值都必须加引号。Boolean 属性的值只能是“true”或“false”。整数值不能包含逗号或小数点，但如果属性允许负值，可以有一个前置的减号。

5.3 属性文件 weblogic.xml

5.3.1 说明

WebLogic Server 允许通过设置 weblogic 应用程序扩展描述符 (weblogic.xml) 配置 jsp 容器，该文件通常位于 web 应用程序的 web-inf 目录下。可配置的元素包含在 `</weblogic-web-app></weblogic-web-app>` 根节点下。

5.3.2 可配置的属性详解

description

description 元素是 Web 应用程序的文字描述。

weblogic-version

weblogic-version 元素表示该 Web 应用程序（在根元素 `<weblogic-web-app>` 下定义）即将部署至的 WebLogic Server 的版本。该元素仅表示版本信息，WebLogic Server 并不使用该元素。

security-role-assignment

security-role-assignment 元素声明 Web 应用程序安全角色与 WebLogic Server 中一个或多个委托人之间的映射，如以下示例所示。

```
<security-role-assignment>
  <role-name>PayrollAdmin</role-name>
  <principal-name>Tanya</principal-name>
  <principal-name>Fred</principal-name>
  <principal-name>system</principal-name>
</security-role-assignment>
```

示例 5-1

还可以使用它将给定角色标记为外部定义的角色，如以下示例所示：

```
<security-role-assignment>
```

```
<role-name>roleadmin</role-name>
<externally-defined/>
</security-role-assignment>
```

示例 5-2

注意：在 `<security-role-assignment>` 元素中，必须定义 `<principal-name>` 或 `<externally-defined>`，不能两者都忽略。

下表描述可在 `security-role-assignment` 元素中定义的元素：

元素	必需/可选	描述
<code><role-name></code>	必须	指定安全角色的名称。
<code><principal-name></code>	如果未定义 <code><externally-defined></code> ，则此元素必须。	指定安全领域内定义的委托人的名称。可以使用多个 <code><principal-name></code> 元素向一个角色映射多个委托人。
<code><externally-defined></code>	如果未定义 <code><principal-name></code> ，则此元素必须。	指定在安全领域内全局定义某特定安全角色；WebLogic Server 使用该安全角色作为委托人名称，而无需在全局领域内查找委托人。如果还在其他位置上定义了该安全角色及其委托人名称的映射，则该元素将用作表示性占位符。

表 5-1

注意：

- 如果您不定义 `security-role-assignment` 元素及其子元素，则 Web 应用程序容器会将该角色名隐式映射为委托人名称，并记录一条警告。如果不定义映射，则 EJB 容器不部署该模块。如果在 `weblogic.xml` 中将 “`role_xyz`” 映射至用户 “`joe`”，则 `role_xyz` 成为本地角色。如果将 `role_xyz` 指定为外部定义的角色，则它成为全局角色（它指领域级别上定义的角色）。
- 如果不定义 `security-role-assignment` 元素，`role_xyz` 成为本地角色，且 Web 应用程序容器创建其隐式映射，并记录一条警告。

run-as-role-assignment

`run-as-role-assignment` 元素将 `web.xml` 中的 `run-as` 角色名（`servlet` 元素的子元素）映射为系统中的有效用户名。对于给定 `servlet`，该值将被 `servlet-descriptor` 中的 `run-as-principal-name` 元素替代。如果给定角色名中没有指定 `run-as-role-assignment`，则 Web 应用程序容器使用 `security-role-assignment` 中定义的第一个 `principal-name`。

以下示例说明如何使用 `run-as-role-assignment` 元素：

```

<run-as-role-assignment>
  <role-name>RunAsRoleName</role-name>
  <run-as-principal-name>joe</run-as-principal-name>
</run-as-role-assignment>
    
```

示例 5-3

下表描述可在 run-as-role-assignment 元素中定义的元素：

元素	必须/可选	描述
<role-name>	必须	指定安全角色的名称。
<run-as-principal-name>	必须	指定委托人的名称。

表 5-2

resource-description

resource-description 元素用于将服务器资源的 JNDI 名映射至 WebLogic Server 中的 EJB 资源引用。

下表描述可在 resource-description 元素中定义的元素：

元素	必须/可选	描述
<res-ref-name>	必须	指定资源引用的名称。
<jndi-name>	必须	指定资源的 JNDI 名。

表 5-3

resource-env-description

resource-env-description 元素将 ejb-jar.xml 部署描述符中声明的 resource-env-ref 映射至其代表的服务器资源的 JNDI 名。

下表描述可在 resource-env-description 元素中定义的元素：

元素	必须/可选	描述
<res-env-ref-name>	必须	指定资源环境引用的名称。
<jndi-name>	必须	指定资源环境引用的 JNDI 名。

表 5-4

ejb-reference-description

下表描述可在 ejb-reference-description 元素中定义的元素：

元素	必须/可选	描述
<ejb-ref-name>	必须	指定 Web 应用程序中使用的 EJB 引用的名称。
<jndi-name>	必须	指定引用的 JNDI 名。

表 5-5

service-reference-description

下表描述可在 service-reference-description 元素中定义的元素：

元素	描述
<service-ref-name>	
<wsdl-url>	
<call-property>	<call-property> 元素具有下列子元素： <ul style="list-style-type: none"> ■ <name> ■ <value>
<port-info>	<port-info> 元素具有下列子元素： <ul style="list-style-type: none"> ■ <port-name> ■ <stub-property> ■ <call-property>

表 5-6

session-descriptor

session-descriptor 元素定义 servlet 会话的参数：

元素名	默认值	描述
timeout-secs	3600	设置 WebLogic Server 等待会话超时的时间（秒）。默认值为 3600 秒。 在繁忙的站点上，可以通过调整会话超时时间来调整应用程序 Session 所占用的内存大小。尽管您希望为浏览器客户端提供每个完成会话的机会，但如果用户已离开站点或已放弃会话，您也不希望毫

		<p>无必要地占用服务器。</p> <p>该元素可以由 web.xml 中的 session-timeout 元素 (分钟) 替代。</p>
invalidation-interval-secs	60	<p>设置 WebLogic Server 在执行超时会话和无效会话的清理检查与删除旧会话并释放内存之间需要等待的时间 (秒)。使用此元素调整 WebLogic Server 以在高流量站点上获得最佳性能。</p> <p>默认值为 60 秒。</p>
sharing-enabled	false	<p>如果该值在应用程序级别上设置为 true, 则 Web 应用程序能够共享 HTTP 会话。</p> <p>如果在 Web 应用程序级别打开该元素, 将忽略该元素。</p>
debug-enabled	false	<p>启用 HTTP 会话的调试功能。</p> <p>默认值为 false。</p>
id-length	52	<p>设置会话 ID 的大小。</p> <p>最小值为 8 字节, 最大值为 Integer.MAX_VALUE。</p> <p>如果您正在编写 WAP 应用程序, 您必须使用 URL 重写, 因为 WAP 协议不支持 cookie。同时, 某些 WAP 设备限制 URL 的长度 (包括特性) 不得超过 128 个字符, 这限制了使用 URL 重写功能可以传输的数据量。为了给各个特性预留较多空间, 可以使用此特性限制 WebLogic Server 随机生成的会话 ID 的大小。</p> <p>还可以通过设置 WAPEnabled 特性将长度限制为固定的 52 位字符, 且不允许使用特殊字符。</p> <p>http://edocs.weblogicfans.net/wls/docs92/webapp/sessions.html - wap</p> <p>http://edocs.weblogicfans.net/wls/docs92/webapp/sessions.html - wap</p>

tracking-enabled	true	启用 HTTP 请求之间的会话跟踪。
cache-size	1028	设置 JDBC 持久性会话和文件持久性会话的缓存大小。
max-in-memory-sessions	-1	<p>设置内存/复制会话的最大限制。</p> <p>如果不能对内存中 servlet 会话的使用加以限制，那么，随着新会话的持续创建，服务器最终必然引发内存不足。为防止出现此问题，WebLogic Server 将针对所创建会话的数量提供可配置的限制。超出该数量时，每次尝试新建会话时都会引发 <code>weblogic.servlet.SessionCreationException</code>。此功能适用于复制内存中会话和非复制内存中会话。</p> <p>要对内存中 servlet 会话的使用加以限制，您可以在 <code>max-in-memory-sessions</code> 元素中设置限制。</p> <p>默认值为 -1（无限制）。</p>
cookies-enabled	true	默认情况下启用会话 cookie，建议保持此状态，但也可以通过将该属性设置为 <code>false</code> 来禁用它们。您可能会关闭该选项以进行测试。
cookie-name	JSESSIONID	定义会话跟踪 cookie 名称。如果没有设置，则默认值为 <code>JSESSIONID</code> 。可以将其设置为适用于您的应用程序的更具体的名称。
cookie-path	null	<p>定义会话跟踪 cookie 路径。</p> <p>如果未设置此特性，则此特性默认为 <code>/</code>（斜杠），浏览器会向 WebLogic Server 服务的所有 URL 发送 cookie。可以将该路径设置为限制性更强的映射，以便限制浏览器会向其发送 cookie 的请求 URL。</p>
cookie-domain	null	指定 cookie 有效的域。例如，将 <code>cookie-domain</code> 设置为 <code>.mydomain.com</code> 会向 <code>*.mydomain.com</code> 域中的所有服务器

		<p>返回 cookie。</p> <p>域名必须至少由两部分组成。将名称设置为 *.com 或 *.net 是无效的。</p> <p>如果不设置此特性，则此特性默认为发出 cookie 的服务器。</p> <p>有关详细信息，请参阅 Sun Microsystems 的 Servlet 规范中的 Cookie.setDomain()。</p>
cookie-comment	null	指定标识 cookie 文件中会话跟踪 cookie 的注释。
cookie-secure	false	<p>指示浏览器仅通过 HTTPS 连接传回 cookie。这可以确保 cookie ID 是安全的，且仅用于使用 HTTPS 的网站。如果启用此功能，则 HTTP 上的会话 Cookie 将不再起作用。</p> <p>如果希望使用此特性，则应禁用 url-rewriting-enabled 元素。</p>
cookie-max-age-secs	-1	<p>设置客户端上的会话 cookie 的生命周期（秒），会话 cookie 超过该时间即会过期。</p> <p>默认值为 -1（无限制）</p>
persistent-store-type	memory	<p>将持久性存储方法设置为以下某个选项：</p> <ul style="list-style-type: none"> ■ memory - 禁用持久性会话存储； ■ replicated - 与 memory 相同，但会话数据将在集群服务器之间复制； ■ replicated_if_clustered - 如果 Web 应用程序部署于集群服务器上，则会复制生效的 persistent-store-type。否则，memory 为默认值； ■ sync-replication-across-cluster - 复制将在集群内同步发生； ■ async-replication-across-

		<p>cluster - 复制将在集群内异步发生;</p> <ul style="list-style-type: none"> ■ file - 使用基于文件的持久性; ■ jdbc - 使用数据库存储持久性会话; ■ cookie - 所有会话数据都存储于用户浏览器的 cookie 中。
persistent-store-cookie-name	WLCOOKIE	<p>设置基于 cookie 的持久性所使用的 cookie 的名称。WLCOOKIE cookie 中带有会话状态, 不应在 Web 应用程序之间共享。</p>
persistent-store-dir	session_db	<p>指定基于文件的持久性所使用的存储目录</p> <p>确保您有充足的磁盘存储空间 (有效会话数与每个会话大小的乘积)。可以通过查看 persistent-store-dir 下创建的文件来确定会话的大小。注意, 每个会话的大小随序列化会话数据的大小变化而变化。</p> <p>每个服务器实例都有一个默认的不需要配置的持久性文件存储。因此, 如果不指定目录, 会在 <server-name>\data\store\default 目录中自动创建默认存储。但是, 默认存储不能在集群服务器间共享。</p> <p>可以在不同服务器之间共享的目录中创建自定义持久性存储, 从而使文件持久性会话集群化。但是, 您需要手工创建此目录。</p>
persistent-store-pool	None	<p>指定要用于持久性存储的 JDBC 连接缓冲池的名称。</p>
persistent-store-table	wl_servlet_sessions	<p>指定用于存储基于 JDBC 的持久性会话的数据库表名。仅当 persistent-store-type 设置为 jdbc 时, 此项才适用。</p> <p>当您选择数据库表名称而非默认值时, 将使用 persistent-store-</p>

		table 元素。
jdbc-column-name-max-inactive-interval		是 wl_max_inactive_interval 列名的备用名称。这个 jdbc-column-name-max-inactive-interval 元素仅适用于基于 JDBC 的持久性。某些不支持长列名的数据库需要使用该元素。
jdbc-connection-timeout-secs	120	注意：有些版本中不赞成使用该元素。 设置 WebLogic Server 等待 JDBC 连接超时的时间（秒），其中 x 是其间的秒数。
url-rewriting-enabled	true	启用 URL 重写（将会话 ID 编码到 URL 中），如果浏览器中禁用 cookie，它还可以提供会话跟踪。
http-proxy-caching-of-cookies	true	设置为 false 时，WebLogic Server 会在下面的响应中添加如下头信息： "Cache-control: no-cache=set-cookie" 这表示代理缓存没有缓存 cookie。
encode-session-id-in-query-params	false	最新 servlet 规范要求容器将会话 ID 编码在路径参数中。某些 Web 服务器不太支持路径参数。在这样的情况下，应该将 encode-session-id-in-query-params 元素设置为 true。（默认值为 false。）
runtime-main-attribute		用于 ServletSessionRuntimeMBean 中。ServletSessionRuntimeMBean 的 getMainAttribute() 会使用该字符串为关键字而返回会话特性值。 示例：user-name 该元素适用于标记不同会话的会话运行时信息。

表 5-7