weblogic.xml 部署描述符元素

本文档提供 WebLogic Server 特定的部署描述符 weblogic.xml 中的元素的完整参考。如果您的 Web 应用程序不包含 weblogic.xml 部署描述符，WebLogic Server 会自动选择部署描述符元素的默认值。要查看 weblogic.xml 的 Schema，请转至 <http://www.bea.com/ns/weblogic/90/weblogic-web-app.xsd>。

下列部分描述 weblogic.xml 部署描述符中根元素 <weblogic-web-app> 下可定义的复杂部署描述符元素：

* + [description](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1019570)

* + [weblogic-version](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1040928)

* + [security-role-assignment](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1040908)

* + [run-as-role-assignment](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1057621)

* + [reference-descriptorGroup](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1068615)

* + [session-descriptor](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1071982)

* + [jsp-descriptor](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1038491)

* + [auth-filter](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1067853)

* + [container-descriptor](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1067857)

* + [charset-params](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1062822)

* + [virtual-directory-mapping](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1039396)

* + [url-match-map](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1039479)

* + [security-permission](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1073709)

* + [context-root](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1073750)

* + [wl-dispatch-policy](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1073535)

* + [servlet-descriptor](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1073622)

* + [work-manager](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1063118)

* + [logging](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1063199)

* + [library-ref](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1060332)

* + [向后兼容性标志](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1075210)

* + [Web 容器全局配置](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html" \l "wp1075244)

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>

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>

|  |  |
| --- | --- |
| **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** |
| <role-name> | 必需 | 指定安全角色的名称。 |
| <principal-name> | 如果未定义 <externally-defined>，则此元素必需。 | 指定安全领域内定义的委托人的名称。可以使用多个 <principal-name> 元素向一个角色映射多个委托人。有关安全领域的详细信息，请参阅 [确保 WebLogic 安全性](http://edocs.weblogicfans.net/wls/docs92/secmanage/index.html)。 |
| <externally-defined> | 如果未定义 <principal-name>，则此元素必需。 | 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。 |

|  |  |
| --- | --- |
| **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。 |

|  |  |
| --- | --- |
| **注意：** | 假设角色名是“role\_xyz”，请考虑以下使用场景 |

* 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>

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

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

reference-descriptorGroup

这个 weblogic.xml 部署描述符指 reference-descriptorGroup，它属于 weblogic-j2ee-xsd 文件的一部分。将使用 reference-descriptorGroup 的以下子元素

|  |  |  |
| --- | --- | --- |
| **元素名** | **默认值** | **值** |
| 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。请参阅 [resource-description](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html#wp1071513)。 |
| 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。请参阅 [resource-env-description](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html#wp1071776)。 |
| ejb-reference-description |  | 请参阅 [ejb-reference-description](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html#wp1071536)。 |
| service-reference-description |  | 请参阅 [service-reference-description](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html#wp1071864)。 |

weblogic.xml 部署描述符。

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

.

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

.

ejb-reference-description

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

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

.

service-reference-description

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

|  |  |  |
| --- | --- | --- |
| **元素** | **必需/ 可选** | **描述** |
| <service-ref-name> |  |  |
| <wsdl-url> |  |  |
| <call-property> |  | <call-property> 元素具有下列子元素：   * <name> * <value> |
| <port-info> |  | <port-info> 元素具有下列子元素：   * <port-name> * <stub-property> * <call-property> |

session-descriptor

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

|  |  |  |
| --- | --- | --- |
| **元素名** | **默认值** | **值** |
| 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。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，您也不希望毫无必要地占用服务器。  该元素可以由 web.xml 中的 session-timeout 元素（分钟）替代。 |
| invalidation-interval-secs | 60 | 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。  默认值为 60 秒。 |
| sharing-enabled | false | 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。  如果在 Web 应用程序级别打开该元素，将忽略该元素。 |
| debug-enabled | false | 启用 HTTP 会话的调试功能。  默认值为 false。 |
| id-length | 52 | 设置会话 ID 的大小。  最小值为 8 字节，最大值为 Integer.MAX\_VALUE。  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。  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。有关详细信息，请参阅“开发 WebLogic Server 的 Web 应用程序”中的 [URL 重写和无线访问协议](http://edocs.weblogicfans.net/wls/docs92/webapp/sessions.html#wap)。 |
| 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 | 设置内存/复制会话的最大限制。  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 weblogic.servlet.SessionCreationException。此功能适用于复制内存中会话和非复制内存中会话。  要对内存中 servlet 会话的使用加以限制，您可以在 max-in-memory-sessions 元素中设置限制。  默认值为 –1（无限制）。 |
| 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 false 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 JSESSIONID。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 | 定义会话跟踪 cookie 路径。  如果未设置此特性，则此特性默认为 /（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。 |
| cookie-domain | null | 指定 cookie 有效的域。例如，将 cookie-domain 设置为 .mydomain.com 会向 \*.mydomain.com 域中的所有服务器返回 cookie。  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。  如果不设置此特性，则此特性默认为发出 cookie 的服务器。  有关详细信息，请参阅 Sun Microsystems 的 Servlet 规范中的 Cookie.setDomain()。 |
| cookie-comment | null | 指定标识 cookie 文件中会话跟踪 cookie 的注释。 |
| cookie-secure | false | 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。  如果希望使用此特性，则应禁用 url-rewriting-enabled 元素。 |
| cookie-max-age-secs | -1 | 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。  默认值为 –1（无限制）  有关 cookie 的详细信息，请参阅[使用会话和会话持久性](http://edocs.weblogicfans.net/wls/docs92/webapp/sessions.html)。 |
| persistent-store-type | memory | 将持久性存储方法设置为以下某个选项：   * memory - 禁用持久性会话存储。 * replicated - 与 memory 相同，但会话数据将在群集服务器之间复制。 * replicated\_if\_clustered – 如果 Web 应用程序部署于群集服务器上，则会复制生效的 persistent-store-type。否则，memory 为默认值。 * sync-replication-across-cluster – 复制将在群集内同步发生。 * async-replication-across-cluster – 复制将在群集内异步发生。 * file - 使用基于文件的持久性（另请参阅 [persistent-store-dir](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html#wp1061967)）。 * jdbc - 使用数据库存储持久性会话。（另请参阅 [persistent-store-pool](http://edocs.weblogicfans.net/wls/docs92/webapp/weblogic_xml.html#wp1062057)。） * cookie – 所有会话数据都存储于用户浏览器的 cookie 中。 |
| persistent-store-cookie-name | WLCOOKIE | 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。  有关详细信息，请参阅[使用基于 Cookie 的会话持久性](http://edocs.weblogicfans.net/wls/docs92/webapp/sessions.html#wp138913)。 |
| persistent-store-dir | session\_db | 指定基于文件的持久性所使用的存储目录  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。  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，默认存储不能在群集服务器间共享。  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群集化。但是，您需要手工创建此目录。 |
| persistent-store-pool | None | 指定要用于持久性存储的 JDBC 连接缓冲池的名称。 |
| persistent-store-table | wl\_servlet\_sessions | 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。  当您选择数据库表名称而非默认值时，将使用 persistent-store-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：本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。 |

jsp-descriptor

jsp-descriptor 元素指定 JSP 编译器的配置参数列表。下表描述可在 jsp-descriptor 元素中定义的元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需/ 可选** | **描述** |
| page-check-seconds | 1 | 设置 WebLogic Server 检查 JSP 文件的时间间隔（秒），以便查看 JSP 文件是否已更改以及是否需要重新编译。如果已发生更改，还会检查依赖关系并递归重新加载。   * 值 -1 表示永不检查页面。该值为生产环境中的默认值。 * 值 0 表示总是检查页面。 * 值 1 表示每秒检查一次页面。该值为开发环境中的默认值。   在生产环境下，很少发生 JSP 更改，可考虑根据您的调整要求将 pageCheckSeconds 的值更改为 60 或更大。 |
| precompile | false | 如果设置为 true，当部署或重新部署 Web 应用程序时，或启动 WebLogic Server 时，WebLogic Server 会自动预编译所有已修改的 JSP。 |
| precompile-continue | false | 如果设置为 true，即使编译期间其中某些 JSP 失败，WebLogic Server 也会继续预编译所有已修改的 JSP。仅当 precompile 设置为 true 时才生效。 |
| keepgenerated | false | 在 JSP 编译过程中保存作为中间步骤生成的 Java 文件。除非此参数设置为 true，否则编译完成后将删除所有中间 Java 文件。 |
| verbose | true | 设置为 true 时，将向浏览器、命令提示和 WebLogic Server 日志文件中输出调试信息。 |
| working-dir | 内部生成的目录 | WebLogic Server 用于保存为 JSP 生成的 Java 和编译类文件的目录的名称。 |
| print-nulls | null | 设置为 false 时，此参数确保表达式结果为“null”时将输出“ ”。 |
| backward-compatible | true | 设置为 true 时，支持向后兼容。 |
| encoding | 平台的默认编码 | 指定 JSP 页面中使用的默认字符集。使用标准 [Java character set names](http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.html)。  如果没有设置，则此特性默认为平台编码。  JSP 页面指令（包含于 JSP 代码中）会替代此设置。例如：  <%@ page contentType="text/html; charset=custom-encoding"%> |
| package-prefix | jsp\_servlet | 指定所有 JSP 页面将编译至的包的前缀。 |
| exact-mapping | true | 设置为 true 时，只要第一次请求 JSP，新创建的 JspStub 即将准确地映射至该请求。如果 exactMapping 设置为 false，则 Web 应用程序容器将为 JSP 生成不准确的 url 映射，exactMapping 允许提供 JSP 页面的路径信息。 |
| default-file-name | true | WebLogic Server 用于保存为 JSP 生成的 Java 和编译类文件的默认文件的名称。 |
| rtexprvalue-jsp-param-name | false | 允许在 jsp:param 标记的 name 特性中使用运行时表达式值。它默认设置为 false。 |

auth-filter

auth-filter 元素指定身份验证筛选器 HttpServlet 类。

|  |  |
| --- | --- |
| **注意：** | 当前版本中不赞成使用该元素。而改用 servlet 身份验证筛选器。 |

container-descriptor

<container-descriptor> 元素指定影响 Web 应用程序行为的参数列表。

check-auth-on-forward

如果要求对从 servlet 或 JSP 转发的请求进行身份验证，请添加 <check-auth-on-forward/> 元素。如果不要求重新身份验证，请省略此标记.例如：

<container-descriptor>  
    <check-auth-on-forward/>  
</container-descriptor>

|  |  |
| --- | --- |
| **注意：** | 作为最佳实践，BEA 不建议启用 check-auth-on-forward 属性。 |

filter-dispatched-requests-enabled

<filter-dispatched-requests-enabled> 元素控制是否向已调度的请求应用筛选器。默认值为 false。

|  |  |
| --- | --- |
| **注意：** | 由于 2.4 servlet 向后兼容 2.3 servlet（根据 2.4 规范），当 WebLogic Server 检测到 2.3 描述符元素时，<filter-dispatched-requests-enabled> 元素默认为 true。 |

redirect-with-absolute-url

<redirect-with-absolute-url> 元素控制 javax.servlet.http.HttpServletResponse.SendRedirect() 方法是否使用相对或绝对 URL 进行重定向。如果您使用的是代理 HTTP 服务器，而且不希望该 URL 转换为非相对链接，请将该元素设置为 false。

默认行为是将 URL 转换为非相对链接。

重定向中使用的用户可读数据。

index-directory-enabled

<index-directory-enabled> 元素控制在找不到合适的索引文件的情况下是否自动生成 HTML 目录列表。

默认值为 false（不生成目录）。值为 true 或 false。

index-directory-sort-by

<index-directory-sort-by> 元素定义 weblogic.servlet.FileServlet 生成的目录列表的排序顺序。有效的 sort-by 值为 NAME、LAST\_MODIFIED 和 SIZE。默认 sort-by 值为 NAME。

servlet-reload-check-secs

<servlet-reload-check-secs> 元素定义 WebLogic Server 是否执行 servlet 检查以查看 servlet 是否已更改，如果已更改，是否重新加载。

* + 值 -1 表示永不检查 servlet。该值为生产环境中的默认值。
  + 值 0 表示总是检查 servlet。
  + 值 1 表示每秒检查一次 servlet。该值为开发环境中的默认值。

控制台中指定的值将始终优先于手工指定的值。

resource-reload-check-secs

使用 <resource-reload-check-secs> 元素针对 Web 应用程序范围内资源路径中发现的缓存资源执行元数据缓存。该参数标识 WebLogic Server 检查资源是否发生修改的频率，如果已修改，则重新加载。

* + 值 -1 表示永不重新加载。该值为生产环境中的默认值。
  + 值 0 表示始终重新加载。
  + 值 1 表示每秒重新加载一次。该值为开发环境中的默认值。

通过管理控制台为此参数指定的值具有优先级。

single-threaded-servlet-pool-size

<single-threaded-servlet-pool-size> 元素定义用于 SingleThreadMode 实例缓冲池的缓冲池的大小。默认值为 5。

|  |  |
| --- | --- |
| **注意：** | 此版本中不赞成使用 SingleThreadMode 实例缓冲池。 |

session-monitoring-enabled

如果将 <session-monitoring-enabled> 元素设置为 true，则允许为会话创建运行时 Mbean。如果设置为默认值 false，则不创建运行时 Mbean。控制台中指定的值优先于手工设置的值。

save-sessions-enabled

<save-sessions-enabled> 元素控制重新部署或取消部署期间是否清除会话数据。它影响内存会话和复制会话。将值设置为 true 表示保存会话数据。设置为 false 则表示重新部署或取消部署 Web 应用程序时将破坏会话数据。默认值为 false。

prefer-web-inf-classes

如果将 <prefer-web-inf-classes> 元素设置为 true，则将优先加载位于 Web 应用程序的 WEB-INF 目录中的类，然后再加载应用程序或系统类加载器中的类。默认值为 false。控制台中指定的值优先于手工设置的值。

default-mime-type

<default-mime-type> 元素的默认值为 null。该元素允许用户为未映射扩展名的内容类型指定默认 mime 类型。

client-cert-proxy-enabled

<client-cert-proxy-enabled> 元素的默认值为 true。设置为 true 时，WebLogic Server 会将标识证书从客户端传递至后端服务器。同时，将通知 WebLogic Server 是否允许或放弃传入的 WL-Proxy-Client-Cert 头。

代理服务器插件将每一标识证书在 WL-Proxy-Client-Cert 头中进行编码，然后将其传递至后端 WebLogic Server 实例。每一 WebLogic Server 实例从该头中获取证书信息，确保其来自于安全来源并使用该信息对用户进行身份验证。对于后台 WebLogic Server 实例，此参数必须设置为 true（在群集/服务器级别上或 Web 应用程序级别上）。

如果将该元素设置为 true，请使用 weblogic.security.net.ConnectionFilter 以确保每一 WebLogic Server 实例仅接受与正在运行代理服务器插件的计算机的连接。如果指定 true 却不使用连接筛选器，则将形成潜在的安全漏洞，因为 WL-Proxy-Client-Cert 头是可以被欺骗的。

relogin-enabled

<relogin-enabled> 元素是向后兼容的参数。如果用户已登录，并尝试访问其不具有访问权限的资源，则会发生“禁止”(403) 响应。

allow-all-roles

在 Web 应用程序的 web.xml 描述符中定义的 security-constraints 元素中，auth-constraint 元素指明允许访问该资源集合的用户角色。此处的 role-name = "\*" 是一种简洁的语法，用于表示 Web 应用程序中的所有角色。在过去的版本中，role-name = "\*" 被视为在领域中定义的所有用户/角色。

此 allow-all-roles 元素是向后兼容的开关，可以用于恢复原有行为。默认行为是允许 Web 应用程序中定义的所有角色。weblogic-xml 中指定的值优先于 WebAppContainerMBean 中定义的值。

native-io-enabled

在使用已隐式注册为默认 servlet 的 weblogic.servlet.FileServlet 来提供静态文件的过程中，如果要使用本地 I/O，请将 native-io-enabled 设置为 true。（默认值为 false。）native-io-enabled 元素仅适用于 Windows。

minimum-native-file-size

minimum-native-file-size 元素仅适用于 native-io-enabled 设置为 true 时。它设置使用本地 I/O 时的最小文件大小。如果要提供的文件大于此值，则使用本地 I/O。如果不设置此值，则所使用的默认值为 4K。

disable-implicit-servlet-mapping

当 disable-implicit-servlet-mappings 标志设置为 true 时，Web 应用程序容器不创建内部 servlet（\*.jsp 和 \*.class 等）的隐式映射；仅使用默认 servlet 映射。关闭隐式 servlet 映射的一个典型使用情况是配置 HttpClusterServlet 或 HttpProxyServlet 时。

默认值为 false。

optimistic-serialization

如果 optimistic-serialization 处于打开状态，并将请求发送到所有 Servlet 上下文中，WebLogic Server 在收到 getAttribute(name) 时将不对上下文和请求特性进行序列化-除序列化。

这意味着必须确保将 Web 应用程序公用特性的作用域设置为共同的父类加载器（以应用程序为作用域）；如果两个 Web 应用程序不属于同一应用程序，则将这些特性置于系统类路径中。

当 optimistic-serialization 处于关闭状态（默认值）时，WebLogic Server 将在 getAttribute(name) 时序列化-除序列化上下文和请求特性，以免可能引发 ClassCastExceptions。

也可以在 [WebAppContainerMBean](http://e-docs.bea.com/wls/docs92/wlsmbeanref/mbeans/WebAppContainerMBean.html) 的域级别中指定优化序列化值，这也适用于所有 Web 应用程序。如果在 weblogic.xml 中指定该值，则该值将替代域级别上的值。

默认值为 false。

monitoring-attribute-name

可以通过监视 ID 标识 HTTP 会话。默认情况下，给定 HTTP 会话的监视 ID 是随机字符串，为确保安全，它并不同于会话 ID。可以在 weblogic.xml 部署描述符的 session-descriptor 中设置 monitoring-attribute-name 元素，然后将某会话特性设置为所定义的 monitoring-attribute-name，从而配置监视 ID。然后，该会话特性值的 toString() 将用作监视 ID。

monitoring-attribute-name元素适用于标记不同会话的会话运行时信息。例如，如果您的“用户名”特性是唯一的，可以将其设置为“用户名”。

WebAppComponentRuntimeBean.getSessionIds() 方法返回具有该名称的会话特性值的数组。如果没有设置，则返回随机生成的字符串的数组。

charset-params

<charset-params> 元素用于定义非 unicode 操作的代码集行为。例如：

<charset-params>

<input-charset>

<resource-path>/\*</resource-path>

<java-charset-name>UTF-8</java-charset-name>

</input-charset>

</charset-params>

input-charset

使用 <input-charset> 元素定义用于读取 GET 和 POST 数据的字符集。例如：

<input-charset>  
    <resource-path>/foo</resource-path>  
    <java-charset-name>SJIS</java-charset-name>  
</input-charset>

有关详细信息，请参阅[确定 HTTP 请求的编码](http://edocs.weblogicfans.net/wls/docs92/webapp/configurejsp.html#wp162180)。

下表描述可在 <input-charset> 元素中定义的元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需/ 可选** | **描述** |
| <resource-path> | 必需 | 如果某请求的 URL 中包含此路径，则将指示 WebLogic Server 使用 <java-charset-name> 指定的 Java 字符集。 |
| <java-charset-name> | 必需 | 指定要使用的 Java 字符集。 |

.

charset-mapping

使用 <charset-mapping> 元素将 IANA 字符集名称映射至 Java 字符集名称。例如：

<charset-mapping>  
    <iana-charset-name>Shift-JIS</iana-charset-name>  
    <java-charset-name>SJIS</java-charset-name>  
</charset-mapping>

有关详细信息，请参阅[将 IANA 字符集映射到 Java 字符集](http://edocs.weblogicfans.net/wls/docs92/webapp/configurejsp.html#wp158086)。

下表描述可在 <charset-mapping> 元素中定义的元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需/ 可选** | **描述** |
| <iana-charset-name> | 必需 | 指定 IANA 字符集名称，该名称将映射至 <java-charset-name> 元素指定的 Java 字符集。 |
| <java-charset-name> | 必需 | 指定要使用的 Java 字符集。 |

.

virtual-directory-mapping

使用 virtual-directory-mapping 元素指定文档根，取代某些特定请求（例如图像请求）的 Web 应用程序的默认文档根。一组 Web 应用程序的所有图像可以存储于一个位置上，不需要复制到使用这些图像的每个 Web 应用程序的文档根中。对于传入的请求，如果已经指定虚拟目录，则 servlet 容器将首先在该虚拟目录中搜索所请求的资源，然后再在 Web 应用程序的原始文档根下搜索。这定义了两个位置上具有同一文档时的搜索优先级。

示例：

<virtual-directory-mapping>

     <local-path>c:/usr/gifs</local-path>

     <url-pattern>/images/\*</url-pattern>

     <url-pattern>\*.jpg</url-pattern>

</virtual-directory-mapping>

<virtual-directory-mapping>

     <local-path>c:/usr/common\_jsps.jar</local-path>

     <url-pattern>\*.jsp</url-pattern>

</virtual-directory-mapping>

下表描述可在 virtual-directory-mapping 元素中定义的元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需/ 可选** | **描述** |
| <local-path> | 必需 | 指定磁盘上的物理位置。 |
| <url-pattern> | 必需 | 包含映射的 URL 模式。必须遵循 Servlet API 规范的 11.2 部分中指定的规则。 |

WebLogic Server 的虚拟目录映射实现要求您的目录匹配该映射的 url 模式。图像示例要求您在 c:/usr/gifs/images 上创建名为 images 的目录。这允许 servlet 容器在 images 目录下找到适用于多个 Web 应用程序的图像。

url-match-map

使用该元素指定用于 URL 模式匹配的类。WebLogic Server 默认 URL 匹配映射类是基于 J2EE 标准的 weblogic.servlet.utils.URLMatchMap。WebLogic Server 中包含的另外一个实现是 SimpleApacheURLMatchMap，可以通过使用 url-match-map 元素将该实现用作插件。

SimpleApacheURLMatchMap 的规则：

如果将 \*.jws 映射至 JWSServlet，则

http://foo.com/bar.jws/baz 将解析为路径信息为 baz 的 JWSServlet。

按照如下示例在 weblogic.xml 中配置要使用的 URLMatchMap：

<url-match-map>

weblogic.servlet.utils.SimpleApacheURLMatchMap

</url-match-map>

security-permission

security-permission 元素根据安全策略文件语法指定单个安全权限。有关 Sun 的安全权限规范的实现，请参阅以下 URL：

[http://java.sun.com/j2se/1.3/docs/guide/security/PolicyFiles.html#FileSyntax](http://java.sun.com/j2se/1.3/docs/guide/security/PolicyFiles.html" \l "FileSyntax)

放弃可选的 codebase 和 signedBy 子句。

例如：

<security-permission-spec>

     grant { permission java.net.SocketPermission "\*", "resolve" };

</security-permission-spec>

其中：

permission java.net.SocketPermission 是权限类名。

"\*" 表示目标名称。

resolve 表示操作。

context-root

context-root 元素定义该独立 Web 应用程序的上下文根。如果 Web 应用程序不是独立的，而属于某 EAR 的一部分，请在该 EAR 的 META-INF/application.xml 文件中指定上下文根。application.xml 中的 context-root 设置优先于 weblogic.xml 中的 context-root 设置。

注意，该 weblogic.xml 元素仅用于使用两阶段部署模型进行部署时。

确定 Web 应用程序的上下文根的优先级顺序如下：

* 1. 在 application.xml 中检查上下文根；如果找到，则将其用作 Web 应用程序的上下文根。
  2. 如果 application.xml 中未设置上下文根，且 Web 应用程序是作为 EAR 的一部分部署的，则检查 weblogic.xml 中是否定义了上下文根。如果找到，则将其用作 Web 应用程序的上下文根。如果 Web 应用程序是独立部署的，则 application.xml 不参与该过程，而仅从 weblogic.xml 开始 context-root 的确定，如果其中没有定义，则默认使用 URI。
  3. 如果 weblogic.xml 或 application.xml 中没有定义上下文根，则将从 URI 中推断上下文路径，将 URI 中定义的值减去 WAR 后缀作为其名称。例如，如果 URI 为 MyWebApp.war，则上下文根的名称为 MyWebApp。

|  |  |
| --- | --- |
| **注意：** | 不能为 EAR 库中的各个 Web 应用程序分别设置 context-root 元素。只能针对 Web 应用程序库进行设置。 |

wl-dispatch-policy

使用 wl-dispatch-policy 元素，通过标识执行队列名称，从而将 Web 应用程序分配至已配置的执行队列。可以使用 per-servlet-dispatch-policy 元素在单个 servlet 或 jsp 级别上替代该 Web 应用程序级别参数。

servlet-descriptor

使用 servlet-descriptor 元素聚合 servlet 特定的元素。

下表描述可在 servlet-descriptor 元素中定义的元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需/ 可选** | **描述** |
| <servlet-name> | 必需 | 将 servlet 名称指定为 web.xml 部署描述符文件的 servlet 元素中定义的 servlet 名称。 |
| <run-as-principal-name> | 可选 | 包含针对 web.xml 部署描述符中所定义的 run-as-role-name 的委托人名称。 |
| <init-as-principal-name> | 可选 | 相当于 servlet 的 init 方法的 run-as-principal-name。此处指定的标识应为系统内的有效用户名。如果未指定 init-as-principal-name，则容器将使用 run-as-principal-name 元素。 |
| <destroy-as-principal-name> | 可选 | 相当于 servlet 的 destroy 方法的 run-as-principal-name。此处指定的标识应为系统内的有效用户名。如果未指定 destroy-as-principal-name，则容器将使用 run-as-principal-name 元素。 |
| <dispatch-policy> | 可选 | *不赞成使用此元素。*用于通过标识执行队列名称来向某个已配置的执行队列分配给定的 servlet。该设置将替代 wl-dispatch-policy 定义的 Web 应用程序级别调度策略。 |

work-manager

work-manager 元素是 <weblogic-web-app> 元素的子元素。可以在 work-manager 元素中定义下列元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需 可选** | **描述** |
| name | 必需 | 指定工作管理器的名称。 |
| response-time-request-class / fair-share-request-class / context-request-class / request-class-name | 可选 | 可以从以下四个元素之间选择一个：  response-time-request-class – 定义应用程序的响应时间请求类。响应时间（毫秒）由特性 goal-ms 定义。增量为 ((目标  T) Cr)/R，其中 T 指平均线程使用时间，R 指达到率，Cr 指确定响应时间目标优先于公平共享的系数。  fair-share-request-class – 定义公平共享请求类。公平共享由默认共享的属性百分比定义。因此，默认值为 100。增量为 Cf/(P R T)，其中 P 指百分比，R 指达到率，T 指平均线程使用时间，Cf 指公平共享优先级低于响应时间目标的系数。  context-request-class – 定义上下文类。上下文由将上下文信息（如当前用户或其角色、cookie 或工作区域字段）映射到已命名的服务类的多个案例定义。  request-class-name – 定义请求类名称。 |
| min-threads-constraint / min-threads-constraint-name | 可选 | 可以从以下两个元素之间选择一个：  min-threads-constraint - 用来保证服务器向受约束工作集的请求分配的线程数，以避免死锁。默认值为零。例如，对于复制更新请求，最少线程值如果为一（可以从对等方同步调用），则该值非常有用。  min-threads-constraint-name - 定义 min-threads-constraint 元素的名称。 |
| max-threads-constraint / max-threads-constraint-name | 可选 | 可以从以下两个元素之间选择一个：  max-threads-constraint - 限制执行来自受约束工作集的请求的并发线程数量。默认值为无限制。例如，假设约束被定义为最多具有 10 个线程，并且由 3 个入口点共享。调度逻辑可确保执行来自三个入口点组合的请求的线程不超过 10 个。  max-threads-constraint-name - 定义 max-threads-constraint 元素的名称。 |
| capacity / capacity-name | 可选 | 可以从以下两个元素之间选择一个：  capacity - 可以定义约束并将其应用于入口点集（称为受约束的工作集）。只有达到该容量时服务器才开始拒绝请求。默认值为零。请注意，容量包括受约束的工作集中已经排队或正在执行的所有请求。此约束主要用于像 JMS 之类的子系统，这些子系统执行自己的流控制。此约束与全局队列阈值无关。  capacity-name – 定义 capacity 元素的名称。 |

logging

logging 元素是 <weblogic-web-app> 元素的子元素。可以在 logging 元素中定义下列元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需 可选** | **描述** |
| log-filename | 必需 | 指定日志文件的名称。需要提供该文件名的完整地址。 |
| logging-enabled | 可选 | 指定是否为 ManagedConnectionFactory 或 ManagedConnection 设置了日志编写器。如果将此元素设置为 true，则从 ManagedConnectionFactory 或 ManagedConnection 中生成的输出将发送到由 log-filename 元素指定的文件。  如果不指定此值，则 WebLogic Server 将使用其定义的默认值。  值范围：true | false  默认值：false |
| rotation-type | 可选 | 设置文件滚动类型。  值为 bySize、byName 或 none   * bySize - 当日志文件大小达到在 file-size-limit 中指定的大小时，服务器将把该文件重命名为 *FileName.n*。 * byName - 按照在 file-time-span 中指定的时间间隔，服务器将该文件重命名为 *FileName.n*。在服务器重命名某个文件之后，后续的消息会累积在一个名称在 log-filename 中指定的新文件中。 * none - 消息累积在单个文件中。当文件体积过大时，必须清除它的内容。   默认值：bySize |
| number-of-files-limited | 可选 | 指定是否应该对该服务器实例为存储旧消息而创建的文件数进行限制。（需要指定 bySize 的滚动类型）。服务器达到此限制后，它将覆盖最旧的文件。如果不启用此选项，服务器将无限制地创建新文件，这样就必须在需要时清除这些文件。  如果通过将 number-of-files-limited 设置为 true 启用该选项，则服务器将参考 rotationType 变量来决定如何滚动该日志文件。滚动意味着替换现有文件而不是创建新文件。如果将 number-of-files-limited 指定为 false，则服务器将创建大量日志文件而不是替换同一个日志文件。  值范围：true | false  默认值：false |
| file-count | 可选 | 服务器在滚动日志时创建的日志文件的最大数量。该数目不包括服务器用于存储当前消息的文件。（需要启用 number-of-files-limited。）  默认值：7 |
| file-size-limit | 可选 | 触发服务器将日志消息移动到单个文件的大小。（需要指定 bySize 的滚动类型。）日志文件达到指定的最小大小之后，服务器会在下次检查文件大小时将当前的日志文件重命名为 *FileName.n*，并新建一个用来存储后续消息的文件。  默认值：500 |
| rotate-log-on-startup | 可选 | 指定服务器在其启动周期内是否滚动其日志文件。  值范围：true | false  默认值：true |
| log-file-rotation-dir | 可选 | 指定将存储滚动日志文件的目录的路径。 |
| rotation-time | 可选 | 日志文件基于时间滚动顺序的开始时间的格式为 k:mm，其中 k 为 1-24。（需要指定 byTime 的滚动类型。）在到达指定时间时，服务器会重命名当前的日志文件。之后，服务器将按照在 file-time-span 中指定的间隔重命名日志文件。  如果指定的时刻已过，服务器会立即开始其文件滚动。  默认情况下，滚动周期会立即开始。 |
| file-time-span | 可选 | 服务器将旧的日志消息保存到另一个文件的时间间隔（小时）。（需要指定 byTime 的滚动类型。）  默认值：24 |

library-ref

library-ref 元素引用计划用作当前 Web 应用程序中的 Web 应用程序库的一个库模块。

示例：

<library-ref>

<library-name>WebAppLibraryFoo</library-name>

<specification-version>2.0</specification-version>

<implementation-version>8.1beta</implementation-version>

<exact-match>false</exact-match>

</library-ref>

只有下列子元素与 Web 应用程序相关： library-name、specification-version、implementation-version 以及 exact-match。

可以在 library-ref 元素中定义下列元素。

|  |  |  |
| --- | --- | --- |
| **元素** | **必需 可选** | **描述** |
| library-name | 必需 | 提供用于库模块引用的库名称。默认值为 null。 |
| specification-version | 必需 | 提供用于库模块引用的规范版本。默认值为 0。（为浮点型。） |
| implementation-version | 必需 | 提供用于库模块引用的实现版本。默认值为 null。 |
| exact-match | 必需 | 默认值为 false。 |

向后兼容性标志

本版本中增添了若干向后兼容性标志，允许您恢复 WebLogic Server 9.0 之前的版本中所见的行为。有关这些标志的完整列表和描述，以及有关 Web 应用程序、JSP 和 Servlet 向后兼容性的所有信息，请参阅 [*Upgrading WebLogic Application Environments*](http://edocs.bea.com/common/docs90/upgrade/index.html) 中的 [Compatibility with Previous Releases](http://edocs.bea.com/common/docs90/upgrade/compat.html)。