Oracle® OpenSSO Update 2 发行说明

Beta

文件号码 821–2840
2010 年 7 月

ORACLE

101014@24999

目录

前言

《Oracle OpenSSO 8.0 Update 2 发行说明》提供有关下载和安装 OpenSSO Update 2 软件的信息。此文档还包含有关自 OpenSSO Update 1 发行版本以来对软件所做更改的信息。

目标读者

这些发行说明旨在供已经安装并部署 Oracle OpenSSO 8.0 的企业管理员和开发者使用。您应该先熟悉核心产品文档中介绍的概念和步骤。

相关文档

这些发行说明作为以下 URL 上提供的核心 Oracle OpenSSO 8.0 产品文档的补充：http://docs.sun.com/app/docs/coll/1767.1。

相关的第三方 Web 站点引用

本文档引用了第三方 URL 以提供其他相关信息。

注 – Oracle 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Oracle 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的、名义上造成的或连带产生的实际或名义损坏或损失，Oracle 概不负责，也不承担任何责任。

文档、支持和培训

有关其他资源，请参见以下 Web 站点：

- 文档 (http://docs.sun.com)
- 支持 (http://www.oracle.com/us/support/systems/index.html)
- 培训 (http://education.oracle.com) – 单击左侧导航栏中的 Sun 链接。

Oracle 欢迎您提出意见

Oracle 欢迎您就其文档的质量及用途提出意见和建议。如果您发现任何错误或有其他任何改进建议，请转至 http://docs.sun.com 并单击 "Feedback"（反馈）。请给出文档的标题、文件号码以及章节号和页码（如果适用）。如果您需要回复，请告知我们。

Oracle Technology Network (http://www.oracle.com/technetwork/index.html) 上提供与 Oracle 软件相关的一系列资源：

- 在 Discussion Forums (http://forums.oracle.com) 上讨论技术问题和解决方案。
- 通过 Oracle By Example (http://www.oracle.com/technology/obe/start/index.html) 获得有关实际操作的逐步教程。
- 下载样例代码 (http://www.oracle.com/technology/sample_code/index.html)。

印刷约定

下表介绍了本文档中使用的印刷约定。

表 P–1 印刷约定

字样	含义	示例
AaBbCc123	命令、文件和目录的名称，以及计算机屏幕输出	编辑 .login 文件。 使用 ls -a 列出所有文件。 machine_name% you have mail.
AaBbCc123	您键入的内容，与计算机屏幕输出的显示不同	machine_name% **su** Password:
aabbcc123	占位符：将用实际名称或值替换	用于删除文件的命令为 rm *filename*。

表 P–1 印刷约定 （续）

字样	含义	示例
AaBbCc123	书名、新术语和要强调的术语	请阅读用户指南中的第 6 章。 **高速缓存**是指在本地存储的副本。 请**勿**保存文件。 **注意**：某些强调项在联机查看时显示为粗体。

命令中的 Shell 提示符示例

下表列出默认的 UNIX 系统提示符和 Oracle Solaris 操作系统中所含 Shell 的超级用户提示符。请注意，命令示例中显示的默认系统提示符会根据 Oracle Solaris 发行版不同而有所差异。

表 P–2 Shell 提示符

Shell	提示符
Bash shell、Korn shell 和 Bourne shell	`$`
用于超级用户的 Bash shell、Korn shell 和 Bourne shell	`#`
C shell	`machine_name%`
用于超级用户的 C shell	`machine_name#`

◆ ◆ ◆ 第 1 章

1 关于 OpenSSO 8.0 Update 2

本章包含以下主题：

OpenSSO 8.0 Update 2 中的新增功能

OpenSSO 8.0 Update 2 提供了安全令牌服务和 OpenSSO Fedlet 的增强功能。

安全令牌服务增强功能

安全令牌服务现在提供以下新功能：

- 支持 TokenType 以生成特定的 Web 服务提供者安全令牌。
- 对于以 X509 安全令牌和用户名安全令牌作为请求者，同时支持非对称和传输绑定。
- 使用用户名通过 SSL 配置 OpenSSO STS 时，通过用户名安全令牌实施 SSL/传输绑定。
- 为使用 useKey 作为 Web 服务客户端公钥的非对称 KeyType 分发 SAML 密钥持有者安全令牌，及分发 Web 服务客户端 X509 安全令牌。
- WSDL 根据安全令牌配置动态进行更新。
- 支持通过 Web 服务提供者公钥进行加密。
- 在将静态用户名密码存储在配置存储库前对其进行加密。
- 通过 WS-Trust 请求支持将用户名令牌作为代表安全令牌。

- 支持分发 SAML 持有者令牌。
- 新的 Web 服务安全验证模块 WSSAuth 支持摘要密码验证。
- 新的 OAMAuth 验证模块通过将 Oracle Access Manager 与 OpenSSO 配合使用，支持单点登录。

有关详细信息，请参见第 3 章，使用安全令牌服务。

Fedlet 增强功能

Fedlet 现在提供了以下新功能：

- 在 .NET Fedlet 中支持加密
- 在 .NET Fedlet 中支持签名
- .NET Fedlet 现在支持单点注销
- .NET Fedlet 支持服务提供者启动的单点登录和工件
- 在 .NET Fedlet 中支持多个身份认证提供者和身份认证提供者搜索
- 在 Fedlet 的属性和配置文件内提供版本信息
- 新密码服务提供商接口实现
- 支持属性查询
- 支持单点注销

有关详细信息，请参见第 4 章，使用 Oracle OpenSSO Fedlet。

OpenSSO 8.0 Update 2 的硬件和软件要求

请参见《Sun OpenSSO Enterprise 8.0 Update 1 Release Notes》中的“Hardware and Software Requirements For OpenSSO Enterprise 8.0 Update 1”

支持新 Web 容器

OpenSSO 8.0 Update 2 支持在《Sun OpenSSO Enterprise 8.0 Update 1 Release Notes》中的“Support for New Web Containers”中介绍的 Web 容器以及下列新 Web 容器：

- Oracle WebLogic Server 10g 发行版本 3 (10.3)

OpenSSO 8.0 Update 2 的问题和解决方法

CR 6959610： 在生产环境中 OpenSSO 8.0 Update 2 样例应删除

OpenSSO 8.0 Update 2 样例可能会导致潜在的安全问题。

解决方法： 如果在生产环境中部署 OpenSSO 8.0 Update 2， 请删除样例以避免出现任何潜在的安全问题。

CR 6964648： WebLogic Server 10.3.3 需要新 Java 安全权限

如果在启用安全管理器的情况下在 Oracle WebLogic Server 10.3.3 上部署 OpenSSO 8.0 Update 2， 则需要其他 Java 安全权限。

解决方法： 将以下权限添加到 WebLogic Server 10.3.3 `weblogic.policy` 文件中：

```
permission java.lang.RuntimePermission "getClassLoader";
```

CR 6939443： 在 WebLogic Server 10.3.x 上通过 LDAP 检查或 OCSP 检查进行证书验证失败

由于早期版本的 Oracle WebLogic Server（如版本 10.3.0 和 10.3.1）中存在的问题， 在启用 LDAP 检查或 OSCP 检查的情况下进行证书验证会失败。

解决方法： 此问题已在 WebLogic Server 10.3.3 中修复。要将证书验证与 LDAP 检查或 OSCP 检查配合使用， 请将 OpenSSO Update 2 和 WebLogic Server 10.3.3 一起使用。

CR 6967026：配置程序无法从 GlassFish 2.1.x 连接到启用了 LDAPS 的目录服务器实例

如果将 GlassFish Enterprise Server v2.1.1 或 v2.1.2 作为 OpenSSO 8.0 Update 2 Web 容器进行部署，则配置程序无法连接到启用了 LDAPS 的目录服务器实例。

解决方法：要使用启用了 LDAPS 的目录服务器并使用 GlassFish 作为 Web 容器，请部署 GlassFish Enterprise Server v2.1。

CR 6948937：在 WebLogic Server 10.3.3 管理控制台中激活 OpenSSO 8.0 Update 2 导致发生异常

如果在 WebLogic Server 10.3.3 管理控制台中部署 OpenSSO 8.0 Update 2 (`opensso.war`) 并单击“启动”以允许 OpenSSO 8.0 Update 2 开始接收请求，在启动了 WebLogic Server 域的控制台中会抛出异常。

注意：启动 OpenSSO 8.0 Update 2 后，它会保持启动状态，并且在停止 OpenSSO 8.0 Update 2 并将其重新启动之前不会再次抛出异常。

解决方法：将 OpenSSO 8 Update 2 `opensso-client-jdk15.war` 文件中的 `saaj-impl.jar` 文件复制到 WebLogic Server 10.3.3 配置的 `endorsed` 目录中，如下所示：

1. 停止 Oracle WebLogic Server 10.3.3 域。
2. 如有必要，解压缩 OpenSSO 8.0 Update 2 `opensso.zip` 文件。
3. 创建一个临时目录并将 *zip-root*`/opensso/samples/opensso-client.zip` 文件解压缩到该目录中，其中 *zip-root* 是解压缩 `opensso.zip` 文件的位置。例如：

   ```
   cd zip-root/opensso/samples
   mkdir ziptmp
   cd ziptmp
   unzip ../opensso-client.zip
   ```
4. 创建一个临时目录并从 `opensso-client-jdk15.war` 中解压缩 `saaj-impl.jar` 文件。例如：

   ```
   cd zip-root/opensso/samples/ziptmp/war
   mkdir wartmp
   cd wartmp
   jar xvf ../opensso-client-jdk15.war WEB-INF/lib/saaj-impl.jar
   ```
5. 在 `WEBLOGIC_JAVA_HOME/jre/lib` 目录下创建一个名为 `endorsed` 的新目录（如果 `endorsed` 不存在），其中 `WEBLOGIC_JAVA_HOME` 是 WebLogic Server 配置为要使用的 JDK。
6. 将 `saaj-impl.jar` 文件复制到 `WEBLOGIC_JAVA_HOME/jre/lib/endorsed` 目录中。
7. 启动 WebLogic Server 域。

CR 6959373：运行 updateschema 脚本后 Web 容器需要重新启动

运行 `updateschema.sh` 或 `updateschema.bat` 脚本后，您必须重新启动 OpenSSO 8.0 Update 2 Web 容器。

CR 6961419：运行 updateschema.bat 脚本需要密码文件

`updateschema.bat` 脚本执行若干个 `ssoadm` 命令。因此，在 Windows 系统上运行 `updateschema.bat` 前，请为 `amadmin` 用户创建包含明文用户密码的密码文件。`updateschema.bat` 脚本会提示您输入密码文件的路径。在该脚本终止前，它将删除此密码文件。

OpenSSO 8.0 Update 2 文档

除此文档外，在以下集合中还提供了其他 OpenSSO 8.0 文档：

`http://docs.sun.com/coll/1767.1`

文档问题

OpenSSO 8.0 Update 2 包含以下文档问题：

- 第 15 页中的 “CR 6958580：控制台联机帮助记录不支持的搜索代理”
- 第 15 页中的 “CR 6967006：控制台联机帮助不记录 OAMAuth 和 WSSAuth 验证模块”
- 第 16 页中的 “CR 6953582：Fedlet Java API（应用编程接口）参考应为公共参考”
- 第 16 页中的 “CR 6953579：OpenSSO Fedlet README 文件应记录单点注销功能”

CR 6958580：控制台联机帮助记录不支持的搜索代理

OpenSSO 8.0 Update 2 管理控制台联机帮助记录搜索代理，即使这些代理不受支持。

解决方法：无。忽略联机帮助中有关搜索代理的信息。

CR 6967006：控制台联机帮助不记录 OAMAuth 和 WSSAuth 验证模块

OpenSSO 8.0 Update 1 管理控制台联机帮助不记录 Oracle Access Manager (OAM) 和 Web 服务安全 (WSS) 验证模块。

解决方法：有关这些验证模块的信息，请参见第 3 章，使用安全令牌服务

CR 6953582：Fedlet Java API（应用编程接口）参考应为公共参考

Fedlet Java API（应用编程接口）公共参考作为 Oracle OpenSSO 8.0 Update 2 Java API（应用编程接口）参考的一部分提供，Oracle OpenSSO 8.0 Update 2 Java API（应用编程接口）参考在以下文档集合中提供：`http://docs.sun.com/coll/1767.1`。

注意：OpenSSO 8.0 Update 2 不支持 `getPolicyDecisionForFedlet` 方法，即使该方法存在于 Java API（应用编程接口）参考中。

CR 6953579：OpenSSO Fedlet README 文件应记录单点注销功能

Fedlet README 文件不记录单点注销功能。

解决方法：对于 Oracle OpenSSO 8.0 Update 2，Fedlet 单点注销功能记录在第 4 章，使用 Oracle OpenSSO Fedlet中。

附加信息和资源

您也可以在以下位置查找其他有用的信息和资源：

- 第 16 页中的 “过时通知和公告”
- 第 17 页中的 “如何报告问题和提供反馈”
- 第 17 页中的 “供残障人士使用的辅助功能”
- 第 17 页中的 “相关的第三方 Web 站点”
- Oracle 高级客户系统服务：

  `http://www.oracle.com/us/support/systems/advanced-customer-services/index.html`
- 软件产品：`http://www.oracle.com/us/sun/sun-products-map-075562.html`
- SunSolve：`http://sunsolve.sun.com/`
- Sun 开发者网络 (SDN)：http://developers.sun.com/
- Sun 开发者服务：`http://developers.sun.com/services/`

过时通知和公告

- 在将来的 OpenSSO 发行版本中，将不再包含服务管理服务 (Service Management Service, SMS) API（应用编程接口）（`com.sun.identity.sm` 软件包)和 SMS 模型。
- 在将来的 OpenSSO 发行版本中，将不再包含 Unix 验证模块和 Unix 验证帮助应用程序 (`amunixd`)。
- 《Sun Java System Access Manager 7.1 发行说明》中指明，Access Manager `com.iplanet.am.sdk` 软件包（一般称为 Access Manager SDK [AMSDK]）和所有相关的 API（应用编程接口）和 XML（可扩展标记语言）模板都不再包含在将来的 OpenSSO 发行版本中。

因此，删除 AMSDK 后，也将删除“传统模式”选项和支持。

现在不提供迁移选项，预计将来也不提供。Oracle Identity Manager 提供用户置备解决方案，可用于取代 AMSDK。有关 Identity Manager 的详细信息，请参见 http://www.oracle.com/products/middleware/identity-management/identity-manager.html。

如何报告问题和提供反馈

如果您有关于 OpenSSO 8.0 Update 2 或后续修补程序发行版本的疑问或问题，请访问支持资源网站 http://sunsolve.sun.com/。

此站点上有一些链接，通过这些链接可以访问知识库、联机支持中心、Product Tracker，还可了解维护方案以及用于联系支持部门的电话号码。如果您要请求获得关于某一问题的帮助，请提供以下信息：

- 问题描述，包括问题发生的时间及其对操作的影响
- 计算机类型、操作系统版本、Web 容器和版本、JDK 版本和 OpenSSO 版本，包括可能影响问题的任何修补程序或其他软件
- 重现问题的步骤
- 所有错误日志或核心转储

供残障人士使用的辅助功能

要获得自此介质发布后所发布的辅助功能，请查看 Section 508 产品评估（可以通过请求获得）来确定哪些版本最适合部署易用解决方案。

有关 Oracle 对辅助功能的支持的信息，请参见 http://www.oracle.com/index.html。

相关的第三方 Web 站点

本文档引用了第三方 URL 以提供其他相关信息。

注 – Oracle 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Oracle 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Oracle 概不负责，也不承担任何责任。

◆ ◆ ◆ 第 2 章

2 安装 OpenSSO 8.0 Update 2

本章包含以下主题：

OpenSSO 8.0 Update 2 安装概述

OpenSSO 8.0 Update 2 可作为修补程序 **TBS** 提供。

在安装 OpenSSO 8.0 Update 2（或后续修补程序）前，请查看此文档中有关新功能、硬件和软件要求、问题和解决方法的信息。

OpenSSO 8.0 Update 2 包含一个 `opensso.war` 文件，您可使用以下方法进行安装：

- **修补现有 OpenSSO 8.0 部署**：使用 Update 2 中的 `ssopatch` 实用程序修补现有的 OpenSSO 8.0 部署，如本章中所述。

  注意 - Oracle 仅支持修补 OpenSSO 8.0 发行版本。例如，支持使用 OpenSSO 8.0 Update 2 修补 OpenSSO 8.0。

- **安装新 OpenSSO 8.0 Update 2 部署**：安装和配置 OpenSSO 8.0 Update 2 `opensso.war` 文件，如《Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide》中所述。

- **创建专用的新 WAR 文件**：使用 `createwar` 脚本从 Update 2 `opensso.war` 文件创建以下新 WAR 文件之一：
  - 仅 OpenSSO 管理控制台 WAR
  - Distributed Authentication UI Server WAR
  - 仅 OpenSSO 服务器 WAR，无管理控制台
  - IDP 搜索服务 WAR

  有关信息，请参见《Sun OpenSSO Enterprise 8.0 Update 1 Release Notes》中的第 4 章 “Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File”。
- **修补现有的专用 OpenSSO WAR 文件**：使用 Update 2 中的 `ssopatch` 实用程序修补现有的专用 OpenSSO 8.0 WAR 文件，如《Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide》中的第 23 章 “Patching OpenSSO Enterprise 8.0”中所述

注 – 如果您运行的是 Access Manager 7.1 或 Access Manager 7 2005Q4 并且要升级到 Update 2，请执行以下步骤：

1. 将 Access Manager 7.x 升级到 OpenSSO 8.0，如《Sun OpenSSO Enterprise 8.0 Upgrade Guide》中所述。
2. 应用 Update 2 修补程序，如本章中所述。

OpenSSO 8.0 Update 2 修补程序

Sun 定期发布 OpenSSO 8.0 Update 2 的修补程序。有关这些修补程序的信息，请定期到这里检查。

计划修补操作

▼ 计划针对 OpenSSO 8.0 的修补操作

1 **阅读第 21 页中的“ssopatch 实用程序概述”。**

2 **为您的平台安装此修补实用程序，如第 22 页中的“安装 ssopatch 实用程序”中所述。**

3 **获取有关现有 WAR 文件的信息，以确定现有 WAR 文件是否已经进行自定义或已经修改，如第 24 页中的“将 OpenSSO WAR 文件与其内部清单进行比较”中所述。**

4 **将现有的 WAR 文件与 Update 2 WAR 文件进行比较，以返回在原始 WAR 中进行自定义的文件、在新 WAR 文件中更新的文件以及两个 WAR 版本之间添加或删除的文件，如第 24 页中的“比较两个 OpenSSO WAR 文件”中所述。**

5 备份并归档现有的 Opensso WAR 文件，如第 22 页中的“备份 OpenSSO WAR 文件”中所述。

6 修补 OpenSSO WAR 文件，如第 25 页中的“修补 OpenSSO WAR 文件”中所述。

7 运行 updateschema 脚本，如第 28 页中的“运行 updateschema 脚本”中所述。

注意 - 如果您修补的是通过 opensso.war 生成的专用 WAR 文件（如仅 OpenSSO 服务器 WAR、仅管理控制台 WAR、Distributed Authentication UI Server 或 IDP 搜索服务 WAR），请参见第 27 页中的“修补专用的 OpenSSO WAR”。

ssopatch 实用程序概述

ssopatch 实用程序是 Java 命令行实用程序，在 Solaris 和 Linux 系统上以 ssopatch 提供，在 Windows 上以 ssopatch.bat 提供。

注意 - ssopatch 在 OpenSSO 8.0 Update 2 中的语法与在 OpenSSO 8.0 发行版本中相比已经进行相当大的更改。有关新语法，请参见第 28 页中的“运行 updateschema 脚本”。

ssopatch 修补实用程序执行以下功能：

- 将 OpenSSO WAR 与其原始清单进行比较，以确定 WAR 文件是否已经进行自定义或已经修改
- 比较两个 OpenSSO WAR 文件，以确定两个文件之间的差异，包括对原始 WAR 文件进行的任何自定义以及在新 WAR 文件中进行的任何更改
- 为生成新修补的 OpenSSO WAR 文件所需的文件生成一个临时区域

下载并解压缩 OpenSSO 8.0 Update 2 ZIP 文件 (opensso_80U2.zip) 后，修补实用程序和相关文件位于 *zip-root*/opensso/tools 目录中的 ssoPatchTools.zip 文件中，其中 *zip-root* 是解压缩 opensso_80U2.zip 的位置。

ssopatch 实用程序使用清单文件确定特定 OpenSSO WAR 文件的内容。清单文件是包含以下内容的 ASCII（美国信息交换标准代码）文本文件：

- 标识 OpenSSO WAR 文件特定版本的字符串
- OpenSSO WAR 文件中的所有单个文件，以及每个文件的校验和信息

清单文件通常命名为 OpenSSO.manifest 并存储在 OpenSSO WAR 文件的 META-INF 目录中。

ssopatch 实用程序将其结果发送到标准输出 (stdout)。如果愿意，也可以通过将输出重定向到文件捕获 ssopatch 输出。如果 ssopatch 成功完成，将返回零 (0) 退出代码。如果发生错误，ssopatch 将返回非零退出代码。

安装 ssopatch 实用程序

在安装 ssopatch 实用程序前，请执行以下操作：

- 下载并解压缩 OpenSSO 8.0 Update 2 ZIP 文件 (opensso_80U2.zip)。
- 将 JAVA_HOME 环境变量设置为指向 JDK 1.5 或更高版本。

安装 ssopatch 实用程序

1. 在 *zip-root*/opensso/tools 目录中找到 ssoPatchTools.zip 文件，其中 *zip-root* 是解压缩 opensso_80U2.zip 的位置。
2. 创建一个新目录以解压缩 ssoPatchTools.zip 文件。例如：ssopatchtools
3. 将 ssoPatchTools.zip 文件解压缩到此新目录中。
4. 如果要从当前目录之外的其他目录运行 ssopatch 实用程序而又不提供完整路径，请将此实用程序添加到 PATH 变量中。

下表介绍 ssoPatchTools.zip 中的文件。

文件或目录	说明
README	说明 ssopatch 的自述文件
/lib	所需的 ssopatch JAR 文件
/patch	updateschema 和 updateschema.bat 脚本及相关的 XML（可扩展标记语言）文件
/resources	所需的属性文件
ssopatch 和 ssopatch.bat	用于 Solaris、Linux 和 Windows 系统的实用程序

备份 OpenSSO WAR 文件

开始之前，请备份现有的 OpenSSO WAR 文件和配置数据：

- 将现有的 OpenSSO WAR 文件复制到一个安全的位置。以后，如果由于某种原因需要回退 Update 2，可以重新部署此 WAR 文件的备份副本。
- 备份配置数据，如《Sun OpenSSO Enterprise 8.0 Administration Guide》中的第 15 章 “Backing Up and Restoring Configuration Data”中所述。

运行 ssopatch 实用程序

要运行 ssopatch 实用程序，请按如下所述进行使用：

```
ssopatch
--help|-?
[--locale|-l]

ssopatch
--war-file|-o
[--manifest|-m]
[--locale|-l]

ssopatch
--war-file|-o
--war-file-compare|-c
[--staging|-s]
[--locale|-l]
[--override|-r]
[--overwrite|-w]
```

其中，选项包括：

- `-war-file|-o` 指定以前部署的 WAR 文件（如 `opensso.war`）的路径。
- `-manifest|-m` 指定要创建的清单文件的路径。清单文件基于 `-war-file|-o`（如果提供此选项）指示的 WAR 文件生成。
- `-war-file-compare|-c` 指定要与 `-war-file|-o` 指示的 WAR 文件进行比较的 WAR 文件的路径。
- `-staging|-s` 指定将 OpenSSO WAR 中的文件写入其中的临时区域的路径。
- `-locale|-l` 指定要使用的语言环境。如果未指定此选项，`ssopatch` 将使用默认的系统语言环境。
- `-override|-r` 忽略两个 WAR 文件的版本检查。版本检查用于确定 WAR 文件的版本，并且只有版本兼容时才能继续。通过此选项，您可以忽略此检查。

  默认值为 false（执行版本检查）。
- `-overwrite|-w` 覆盖现有临时区域中的文件。默认值为 false（不覆盖文件）。

将 OpenSSO WAR 文件与其内部清单进行比较

使用此过程可以确定自下载 OpenSSO WAR 文件以来，该文件是否已进行自定义或已经修改。

ssopatch 实用程序可生成新的内部清单文件，然后将此内部清单与原始 OpenSSO WAR 文件内的 META-INF 目录中存储的清单进行比较。

将 OpenSSO WAR 文件与其内部清单进行比较

1. 运行 ssopatch 以将 OpenSSO WAR 文件与其内部清单进行比较。例如：

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Comparing manifest of Internal (Enterprise 8.0 Build 6(200810311055))
against /zip-root/opensso/deployable-war/opensso.war (generated-200905050855)
File not in original war (images/login-origimage.jpg)
File updated in new war (images/login-backimage.jpg)
File updated in new war (WEB-INF/classes/amConfigurator.properties)
Differences: 3
```

以下示例表明对原始 WAR 文件进行的这些更改：

- images/login-origimage.jpg 存在于 opensso.war 中，但未包含在原始清单中。
- 相对于原始清单，images/login-backimage.jpg 已在 opensso.war 中进行自定义。
- 相对于原始清单，WEB-INF/classes/amConfigurator.properties 文件已在 opensso.war 中进行自定义。

比较两个 OpenSSO WAR 文件

使用此过程可以比较两个 WAR 文件，以显示已经执行以下操作的文件：

- 已在原始 OpenSSO WAR 中进行自定义
- 已在新 OpenSSO WAR 文件中进行更新
- 已在两个 OpenSSO WAR 版本间添加或删除

比较两个 OpenSSO WAR 文件

1. 运行 ssopatch 以比较两个 WAR 文件。在以下示例中，使用了 -override 选项来忽略两个 WAR 文件间的版本检查：

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
-c /u1/opensso/deployable-war/opensso.war --override
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Original manifest: Enterprise 8.0 Build 6(200810311055)
```

```
New manifest: Enterprise 8.0 Update 2 Build 6.1(200904300525)
Versions are compatible
Generating Manifest for: /u1/opensso/deployable-war/opensso.war
Comparing manifest of /zip-root/opensso/deployable-war/opensso.war
(generated-200905050919) against
    /u1/opensso/deployable-war/opensso.war (generated-200905050920)
File updated in new war(WEB-INF/classes/amClientDetection_en.properties)
File updated in new war(WEB-INF/classes/fmSAMLConfiguration_fr.properties)
...
Differences: 1821
Customizations: 3
```

此示例显示已在新 WAR 文件中更新和自定义的文件。

修补 OpenSSO WAR 文件

使用此过程可以创建新的临时区域，在其中原始 WAR 文件将与新 WAR 文件进行合并。

此操作可比较每个 WAR 文件的清单，然后显示：

- 原始 WAR 文件中自定义的文件
- 新 WAR 文件中更新的文件
- 两个 WAR 文件版本间添加或删除的文件

`ssopatch` 然后会将相应的文件复制到一个临时目录中，您必须先在该目录中添加任何自定义内容，然后再创建和部署新修补的 WAR。

创建临时区域以修补 OpenSSO WAR 文件

1. 尽管 `ssopatch` 不会修改原始 `opensso.war` 文件，但建议您备份此文件，以供您在需要回退修补的 `opensso.war` 文件时使用。
2. 运行 `ssopatch` 以创建临时区域。例如：

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
  -c /u1/opensso/deployable-war/opensso.war --override -s /tmp/staging
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Original manifest: Enterprise 8.0 Build 6(200810311055)
New manifest: Enterprise 8.0 Update 2 Build 6.1(200904300525)
Versions are compatible
Generating Manifest for: /u1/opensso/deployable-war/opensso.war
Comparing manifest of /zip-root/opensso/deployable-war/opensso.war
    (generated-200905051031) against /u1/opensso/deployable-war/opensso.war
    (generated-200905051032)
File was customized in original, but not found in new war.
Staging area using original war version (samples/saml2/sae/header.jsp)
File was customized in original, but not found in new war.
Staging area using original war version
    (WEB-INF/template/opends/config/upgrade/config.ldif.4517)
File was customized in original, but not found in new war.
```

```
Staging area using original war version
    (WEB-INF/template/opends/config/upgrade/schema.ldif.4517)
Differences: 1813
Customizations: 0
```

在此示例中，`/tmp/staging` 是 `ssopatch` 复制文件的临时区域。

使用上一步骤的结果，根据需要更新临时区域中的文件。

使用下表确定在生成新修补的 WAR 文件前您可能需要对每个文件采取的操作。

ssopatch 结果	**说明和所需操作**
`File not in original war` *filename*	指示的文件未存在于原始 WAR 文件中，但存在于最新版本的 WAR 文件中。 **操作**：无
`File updated in new war` *filename*	指示的文件存在于原始 WAR 文件和新 WAR 文件中，并且已在最新版本的 WAR 文件中进行了更新。在原始 WAR 文件中未进行任何自定义。 **操作**：无
`File customized` *filename*	指示的文件存在于两个 WAR 文件中，并且已在原始版本的 WAR 文件中进行了自定义，但未在最新版本的 WAR 文件中进行更新。 **操作**：无
`May require manual customization` *filename*	文件存在于两个 WAR 文件中，并且已在原始版本的 WAR 文件中进行了自定义，在最新版本的 WAR 文件中进行了更新。 **操作**：如果需要文件中的自定义内容，您必须手动将其添加到临时目录中的新更新的文件中。
`File was customized in original, but not found in new war`	文件存在于原始 WAR 文件中，但未在新的 WAR 中。 **操作**：无

后续步骤

1. 基于临时区域中的文件创建新 OpenSSO WAR 文件。例如：

   ```
   cd /tmp/staging
   jar cvf /patched/opensso.war *
   ```

   其中，`/patched/opensso.war` 是新修补的 OpenSSO WAR 文件
2. 使用原始的部署 URI，将 `/patched/opensso.war` 文件重新部署到 Web 容器。例如，`/opensso`

OpenSSO 配置更改。新 OpenSSO WAR 文件可能包含未存在于原始 WAR 文件中的配置更改。将单独为每个修补程序记录任何配置更改（如果有）。有关任何配置更改的详细信息，请查看修补程序文档和《Sun OpenSSO Enterprise 8.0 发行说明》。（即使在新 WAR 文件中没有配置更改，OpenSSO 清单文件中的版本字符串也会更改。）

如果需要回退修补的版本，请取消部署修补的 WAR 文件，然后重新部署原始 WAR 文件。

创建 OpenSSO WAR 清单文件

OpenSSO 清单文件是一个文本文件，用于标识特定发行版本的 WAR 文件中的所有单个文件，并包含每个文件的校验和信息。

使用此过程可以创建可在专用 OpenSSO WAR（如仅 OpenSSO 服务器 WAR、仅管理控制台 WAR、Distributed Authentication UI Server 或 IDP 搜索服务 WAR）中包含的清单文件。

创建 OpenSSO WAR 清单文件

1. 运行 `ssopatch` 以创建 OpenSSO 清单文件。例如：

   ```
   ./ssopatch -o zip-root/opensso/deployable-war/opensso.war --manifest /tmp/manifest
   ```

   其中，`opensso.war` 是现有的 OpenSSO WAR 文件。

   `ssopatch` 实用程序可在 `/tmp` 目录中创建名为 `manifest` 的新清单文件。
2. 要允许修补 WAR 文件，请将此新清单文件复制到 `opensso.war` 文件内的 `META-INF` 目录中。例如：

   ```
   mkdir META-INF
   cp /tmp/manifest META-INF
   jar uf opensso.war META-INF/manifest
   ```

修补专用的 OpenSSO WAR

如果您以前创建了专用 OpenSSO WAR（如仅 OpenSSO 服务器 WAR、仅管理控制台 WAR、Distributed Authentication UI Server 或 IDP 搜索服务 WAR），那么可以使用 `ssopatch` 实用程序对其进行修补。

修补专用的 OpenSSO WAR

1. 为专用的 OpenSSO WAR 创建清单文件，如第 27 页中的 “创建 OpenSSO WAR 清单文件”中所述。

注意：请在进行任何自定义之前，以原始 OpenSSO 8.0 `opensso.war`（与 Sun 最初提供时相同）为基础创建清单文件。如果在进行自定义之后创建清单，`ssopatch` 可能使用 Update 2 中的文件，而非经过自定义的文件，因此您需要在修补之后重新执行自定义。

2. 基于 OpenSSO 8.0 Update 2 `opensso.war` 文件生成专用的 OpenSSO WAR，如《Sun OpenSSO Enterprise 8.0 Update 1 Release Notes》中的第 4 章 "Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File"中所述。
3. 使用 `ssopatch` 实用程序将旧的 WAR 文件与新的 WAR 文件进行比较。
4. 为新的专用 WAR 文件生成一个临时区域，如第 25 页中的 "创建临时区域以修补 OpenSSO WAR 文件"中所述。
5. 重新部署新的专用 WAR 文件。

运行 updateschema 脚本

运行 `ssopatch` 后，请在 Solaris 或 Linux 系统上运行 `updateschema.sh`，或在 Windows 系统上运行 `updateschema.bat`。此脚本会更新 OpenSSO 服务器版本，添加新的默认服务器属性、添加 Update 2 中的错误修复和增强功能所需的新属性模式。必须运行 `updateschema` 才能更新服务器版本。

准备工作

- `updateschema.sh` 或 `updateschema.bat` 脚本需要 Update 2 版本（或更高版本）的 `ssoadm` 命令行实用程序。因此，在运行此脚本前，请先安装 Update 2 管理工具，如《Sun OpenSSO Enterprise 8.0 Update 1 Release Notes》中的第 3 章 "Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools"中所述。
- `updateschema.bat` 脚本执行若干个 `ssoadm` 命令。因此，在 Windows 系统上运行 `updateschema.bat` 前，请为 `amadmin` 用户创建包含明文用户密码的密码文件。`updateschema.bat` 脚本会提示您输入密码文件的路径。在该脚本终止前，它将删除此密码文件。

运行 updateschema 脚本

1. 更改到 *patch-tools*`/patch` 目录，其中 *patch-tools* 是解压缩 `ssoPatchTools.zip` 的位置。
2. 运行 `updateschema.sh` 或 `updateschema.bat`。例如，在 Solaris 系统上：

   ```
   ./updateschema.sh
   ```

3. 当脚本提示您时，请提供以下信息：
   - `ssoadm` 实用程序的完整路径（不包括 `ssoadm` 本身）。例如：`/opt/ssotools/opensso/bin`

- amadmin 密码

updateschema.sh 或 updateschema.bat 脚本会向标准输出中写入任何消息或错误。

4. 重新启动 OpenSSO 8.0 Update 2 Web 容器。

回退修补程序安装

如果需要回退修补程序安装，只需重新部署原始的 opensso.war 文件（或专用的 WAR 文件）。

◆ ◆ ◆ 第 3 章

3 使用安全令牌服务

作为可信赖的权威服务，OpenSSO 安全令牌服务用于分发和验证安全令牌。作为 Web 服务安全提供者，安全令牌服务可确保 Web 服务客户端与 OpenSSO STS 服务本身之间通信的安全。自 OpenSSO 8.0 Update 2 以来，已经为安全令牌服务新增了许多增强功能。

本章包含以下主题：

添加 WSSAuth 验证模块

Web 服务安全验证模块使 OpenSSO 可以通过作为验证令牌接收并包含在从 Web 服务客户端到 Web 服务提供者的服务请求中的摘要密码验证用户名。

▼ 添加新的 Web 服务安全验证模块实例

1. **在“Access Manager”选项卡中，单击“验证”子选项卡。**
2. **在“模块实例”部分，单击“新建”。**
3. **在“名称”字段中，键入此 WSSAuth 验证模块实例的名称。**
4. **对于“类型”，选择“WSSAuth”。**

5 **配置 WSSAuth 验证模块实例。**

▼ 配置 WSSAuth 验证模块实例

1 **在“Access Manager”选项卡中，单击“验证”子选项卡。**

2 **在“模块实例”部分，单击您要配置的 WSSAuth 验证模块实例的名称。**

3 **为 WSSAuth 验证模块实例领域属性提供值。**

下表列出您可以配置的属性及其描述。

用户搜索属性	待编写
用户领域	待编写
用户密码属性	待编写
验证级别	待编写

添加 OAMAuth 验证模块

Oracle 验证模块可让 OpenSSO 使先前通过了 Oracle Access Manager 验证的管理员通过 OpenSSO 的验证，并使该管理员可以单点登录到 OpenSSO。该管理员无需提供 OpenSSO 凭据。

▼ 添加新的 Oracle 验证模块实例

1 **在“Access Manager”选项卡中，单击“验证”子选项卡。**

2 **在“模块实例”部分，单击“新建”。**

3 **在“名称”字段中，键入此 Oracle 验证模块实例的名称。**

4 **对于“类型”，选择“OAMAuth”。**

5 **单击“确定”。**

6 **配置 OAMAuth 验证模块实例。**

▼ 配置 Oracle 验证模块实例

1 **在“Access Manager”选项卡中，单击“验证”子选项卡。**

2 **在“模块实例”部分，单击您要配置的 OAMAuth 验证模块实例的名称。**

3 **为 Oracle 验证模块实例领域属性提供值。**

下表列出您可以配置的属性及其描述。

远程用户标头名称	待编写
允许的标头值	“当前值”列表显示“待编写” ■ 要在列表中添加标头值，请在“新值”字段中键入“待编写”，然后单击“添加”。 ■ 要从“当前值”列表中删除条目，请选择该条目，然后单击“删除”。
验证级别	待编写

生成安全令牌

Oracle OpenSSO 安全令牌服务 (OpenSSO STS) 可在 Web 服务客户端与 Web 服务提供者之间建立信任关系，然后在它们之间维持这种信任。Web 服务可信任仅由一个实体 (OpenSSO STS) 分发的令牌，而不必与多个客户端进行通信。通过这种方式，OpenSSO STS 可大大降低信任点管理的系统开销。

以下部分将提供有关以下内容的说明：如何确定安全令牌需求，如何配置安全令牌服务以生成安全令牌并验证其是否满足这些需求。

将 Web 服务提供者注册到 OpenSSO STS

当添加新的 Web 服务提供者安全代理配置文件时，Web 服务提供者会自动注册到 OpenSSO STS。有关详细信息，请参见下面的部分：

将某 Web 服务提供者注册到 OpenSSO STS 后，可以配置 OpenSSO STS 以生成该 Web 服务提供者可接受的 Web 客户端安全令牌。

从 OpenSSO STS 请求 Web 服务客户端安全令牌

您必须先确定 Web 服务提供者需要哪种类型的安全令牌，然后才能配置安全令牌服务以生成 Web 客户端安全令牌。OpenSSO STS 支持 Liberty Alliance Project 安全令牌和 Web 服务互操作性基本安全配置文件安全令牌。

安全令牌生成过程流

使用 Liberty Alliance Project 令牌启用安全性时，HTTP（超文本传输协议）客户端（即浏览器）会通过 Web 服务客户端向 Web 服务提供者发送访问请求。Web 服务安全代理会将该请求重定向至 OpenSSO STS 验证服务。如果采用 Liberty Alliance Project 安全机制，HTTP（超文本传输协议）安全代理会发出重定向命令。如果使用 WS-IBS 安全，SOAP 安全代理会发出重定向命令。

OpenSSO STS 验证服务会确定 Web 服务提供者注册的安全机制，并检索相应的安全令牌。成功进行验证后，Web 服务客户端会提供 SOAP 消息体，而 Web 服务客户端上的 SOAP 安全代理会插入安全标头和一个令牌。然后，在将请求发送给 WSP 之前，会对该消息进行签名。

Web 服务提供者端的 SOAP 安全代理会先验证 SOAP 请求中的签名和安全令牌，然后再将该请求转发给 Web 服务提供者自身。Web 服务提供者然后将处理该请求并将 SOAP 安全代理签名的响应返回给 Web 服务客户端。然后，Web 服务客户端上的 SOAP 安全代理会验证签名，之后再将该响应转发给 Web 服务客户端。

下表列出 Liberty Alliance Project 事务支持的令牌及其简短描述。

表 3–1 请求者令牌 - Liberty Alliance Project

令牌	满足以下要求
X.509	■ 安全 Web 服务使用公钥基础结构 (Public Key Infrastructure, PKI)，在该基础结构中，Web 服务客户端提供公钥作为识别请求程序以及向 Web 服务提供者进行验证的方式。 ■ 安全 Web 服务使用公钥基础结构 (Public Key Infrastructure, PKI)，在该基础结构中，Web 服务客户端提供公钥作为识别请求程序以及向 Web 服务提供者进行验证的方式。
持有者令牌	■ 安全 Web 服务使用安全声明标记语言 (Security Assertion Markup Language， SAML) SAML 持有者令牌确认方法。 ■ WSC 为 SAML 声明提供公钥信息，以此作为向 Web 服务提供者验证请求程序的方式。 ■ 第二个签名将该声明绑定到 SOAP 消息。 ■ 第二个签名绑定使用 Liberty Alliance Project 定义的规则。

表 3–1 请求者令牌 - Liberty Alliance Project （续）

SAML 令牌	■ 安全 Web 服务使用 SAML 密钥持有者确认方法。 ■ WSC 将 SAML 声明和数字签名添加到 SOAP 标头中。 ■ 该签名还会随附一个发件人证书或公钥。 ■ 将使用 Liberty Alliance Project 定义的规则来处理发送。

下表列出 WS-IBS 事务支持的令牌及其简短描述。

表 3–2 请求者令牌 - WS-IBS

令牌	满足以下要求
用户名	■ 安全 Web 服务需要用户名、密码及已签名的请求（可选）。 ■ Web 服务使用者提供用户名令牌作为识别请求程序的方式 ■ Web 服务使用者提供密码、共享密码或等效密码以向 Web 服务提供者验证身份。
X.509	安全 Web 服务使用 PKI（公钥基础结构），在该基础结构中，Web 服务使用者提供公钥作为识别请求程序以及完成向 Web 服务提供者进行的验证的方式。
SAML 密钥持有者	■ 安全 Web 服务使用 SAML 密钥持有者确认方法。 ■ Web 服务使用者为 SAML 声明提供公钥信息，以此作为向 Web 服务提供者验证请求程序的方式。 ■ 第二个签名将该声明绑定到 SOAP 有效载荷。
SAML 发件人担保	■ 安全 Web 服务使用 SAML 发件人担保确认方法。 ■ Web 服务使用者将 SAML 声明和数字签名添加到 SOAP 标头中。该签名还会随附一个发件人证书或公钥。

使用安全令牌生成矩阵

使用安全令牌生成矩阵可以帮助您配置 OpenSSO STS 以生成 Web 服务提供者所需的 Web 服务客户端安全令牌。首先，在名为“OpenSSO STS 输出令牌”的最后一列中，找到满足 Web 服务提供者令牌要求的描述。然后，使用同一行中的参数值来配置安全令牌服务。“令牌生成矩阵图例”提供有关表标题和可用选项的信息。有关详细的配置说明，请参见第 5.2.3 部分“配置安全令牌服务”。有关 Web 服务安全性和相关术语的一般信息，请参见：

- http://www.oracle.com/technology/tech/standards/pdf/security.pdf
- http://download.oracle.com/docs/cd/E15523_01/web.1111/b32511/intro_security.htm#CDDHHG

安全令牌生成矩阵汇总了常用的安全令牌服务参数设置和 OpenSSO STS 基于这些设置生成的安全令牌的类型。

表 3–3 安全令牌生成矩阵

行	消息级别安全绑定	Web 服务客户端令牌	KeyType	代表令牌	使用密钥	OpenSSO STS 输出令牌
1	非对称	X509	持有者	是	否	SAML 持有者，无证明密钥
2	非对称	用户名	持有者	是	否	SAML 持有者，无证明密钥
3	非对称	X509	持有者	否	否	SAML 持有者，无证明密钥
4	非对称	用户名	持有者	否	否	SAML 持有者，无证明密钥
5	非对称	X509	对称	是	否	SAML 密钥持有者，对称证明密钥
6	非对称	用户名	对称	是	否	SAML 密钥持有者，对称证明密钥
7	非对称	X509	对称	否	否	SAML 密钥持有者，对称
8	非对称	用户名	对称	否	否	SAML 密钥持有者，对称证明密钥
9	非对称	X509	非对称	否	Web 服务客户端公钥	SAML 密钥持有者，非对称证明密钥
150	非对称	X509	Oracle 专属 SAML 发件人担保	是	否	SAML 发件人担保，无证明密钥
11	非对称	用户名	Oracle 专属 SAML 发件人担保	是	否	SAML 发件人担保，无证明密钥

表 3–3 安全令牌生成矩阵 （续）

12	非对称	X509	Oracle 专属 SAML 发件人担保	否	否	错误
13	非对称	用户名	Oracle 专属 SAML 发件人担保	否	否	错误
14	传输	用户名	持有者	是	否	SAML 持有者，无证明密钥
15	传输	用户名	持有者	否	否	SAML 持有者，无证明密钥
16	传输	用户名	对称	是	否	SAML 密钥持有者，对称
17	传输	用户名	对称	否	否	SAML 密钥持有者，对称证明密钥
18	传输	用户名	Oracle 专属 SAML 发件人担保	是	否	SAML 发件人担保，无证明密钥
19	传输	用户名	Oracle 专属 SAML 发件人担保	否	否	错误
20	非对称	用户名	非对称	否	Web 服务客户端公钥	错误
21	传输	用户名	非对称	否	Web 服务客户端公钥	错误
22	非对称	X509	非对称	是	否	错误
23	非对称	用户名	非对称	是	否	错误
24	传输	用户名	非对称	是	否	错误
25	非对称	X509	非对称	否	否	SAML 密钥持有者，非对称证明密钥

表 3–3 安全令牌生成矩阵 （续）

26	非对称	X509	否	否	否	SAML 密钥持有者，非对称证明密钥
27	非对称	用户名	否	否	否	SAML 密钥持有者，对称证明密钥
28	传输	用户名	否	否	否	SAML 密钥持有者，对称证明密钥

安全令牌服务问题和解决方法

待编写

配置问题和解决方法

待编写

文档勘误表

待编写

◆ ◆ ◆ 第 4 章

使用 Oracle OpenSSO Fedlet

本部分提供关于 Oracle OpenSSO Fedlet 的以下信息：

关于 Oracle OpenSSO Fedlet

Oracle OpenSSO Fedlet 是轻量级服务提供者 (Service Provider, SP) 实现，可随 Java 或 .NET 服务提供者应用程序一起部署，使该应用程序可以使用 SAMLv2 协议与 Oracle OpenSSO 8.0 Update 2 等身份认证提供者 (Identity Provider, IDP) 进行通信。Fedlet 具有两个版本，具体取决于您的平台：

- Java Fedlet 最初发布于 OpenSSO 8.0 中。有关信息，请参见《Sun OpenSSO Enterprise 8.0 Deployment Planning Guide》中的第 5 章 “Using the OpenSSO Enterprise Fedlet to Enable Identity Federation”。
- .NET Fedlet 发布于 OpenSSO 8.0 Update 1 中。有关信息，请参见《Sun OpenSSO Enterprise 8.0 Update 1 Release Notes》中的第 10 章 “Using the ASP.NET Fedlet with OpenSSO Enterprise 8.0 Update 1”。

在 Oracle OpenSSO 8.0 Update 2 中，Fedlet 可按如下方式获取：

- 解压缩 OpenSSO 8.0 Update 2 ZIP 文件后，可在以下文件中找到 Java Fedlet 和 .NET Fedlet：

  zip-root`/opensso/fedlet/fedlet-unconfigured.zip`，其中 *zip-root* 是解压缩 Oracle OpenSSO 8.0 Update 2 ZIP 文件的位置。
- 安装 Oracle OpenSSO 8.0 Update 2 后，您可以使用“常见任务”下的“创建 Fedlet”工作流在 OpenSSO 8.0 管理控制台中创建 Java Fedlet。

Oracle OpenSSO Fedlet 的要求

Fedlet 具有以下要求：

- Oracle OpenSSO 8.0 Update 2 支持的 Web 容器（如果您计划部署 `fedlet.war`）或与 Fedlet 集成的 Java 服务提供者应用程序。请参见第 12 页中的 “OpenSSO 8.0 Update 2 的硬件和软件要求”。
- Microsoft Internet Information Server (IIS) 7.0 及更高版本（如果您计划部署 .NET Fedlet）
- JDK 1.6.x 及更高版本

Oracle OpenSSO Fedlet 配置

本部分说明如何使用服务提供者应用程序对 Fedlet 进行初始配置：

- 第 40 页中的 “配置 Java Fedlet”
- 第 42 页中的 “配置 .NET Fedlet”

完成对 Fedlet 的初始配置后，请继续进行您要执行的任何其他配置。需要注意以下事项：

- 如果修改 Fedlet `sp.xml` 文件，必须将此文件重新导入到身份认证提供者中。
- 如果在服务提供者端进行了其他 Fedlet 配置更改，请将此信息告知身份认证提供者管理员，以便可在身份认证提供者端进行所需的配置更改。

▼ 配置 Java Fedlet

1 **在身份认证提供者端，生成身份认证提供者的 XML（可扩展标记语言）元数据，并将该元数据保存在一个名为 `idp.xml` 的文件中。**

对于 Oracle OpenSSO 8.0 Update 2，请使用 `exportmetadata.jsp`。例如：

```
http://opensso-idp.example.com:8080/opensso/saml2/jsp/exportmetadata.jsp
```

2 **在服务提供者端，解压缩 Fedlet ZIP 文件（如有必要）。**

3 **创建 Fedlet 主目录，这是 Fedlet 读取其元数据、信任环和配置属性文件的目录。**

默认位置是运行 Fedlet Web 容器的用户的主目录（由 `user.home` JVM 属性表示）下的 Fedlet 子目录。例如，如果此主目录为 `/home/webservd`，则 Fedlet 主目录为：

```
/home/webservd/fedlet
```

要更改默认的 Fedlet 主目录，请将 JVM 运行时 `com.sun.identity.fedlet.home` 属性的值设置为所需的位置。例如：

```
-Dcom.sun.identity.fedlet.home=/export/fedlet/conf
```

Fedlet 之后将从 /export/fedlet/conf 目录读取其元数据、信任环和配置文件。

4 **将以下文件从 Java Fedlet java/conf 目录复制到 Fedlet 主目录：**

- sp.xml-template
- sp-extended.xml-template
- idp-extended.xml-template
- fedlet.cot-template

5 **在 Fedlet 主目录中，重命名所复制的文件并将 -template 从每个名称中删除。**

6 **在复制到 Fedlet 主目录并进行重命名的文件中，替换下表中显示的标记：**

标记	替换为
FEDLET_COT	远程身份认证提供者和 Java Fedlet 服务提供者应用程序所属的信任环 (Circle Of Trust, COT) 的名称。
FEDLET_ENTITY_ID	Java Fedlet 服务提供者应用程序的 ID（名称）。例如：fedletsp
FEDLET_PROTOCOL	Java Fedlet 服务提供者应用程序（如 fedlet.war）的 Web 容器的协议。例如：https
FEDLET_HOST	Java Fedlet 服务提供者应用程序（如 fedlet.war）的 Web 容器的主机名。例如：fedlet-host.example.com
FEDLET_PORT	Java Fedlet 服务提供者应用程序（如 fedlet.war）的 Web 容器的端口号。例如：80
FEDLET_DEPLOY_URI	Java Fedlet 服务提供者应用程序的 URL。例如：http://fedletsp.example.com/myFedletApp
IDP_ENTITY_ID	远程身份认证提供者的 ID（名称）。例如：openssoidp

注意：如果 Fedlet 服务提供者或身份认证提供者实体 ID 包含百分号 (%) 或逗号 (,)，则必须先对该字符进行转义，然后才能在 fedlet.cot 文件中对其进行替换。例如，将 "%" 更改为 "%25"，将 "," 更改为 "%2C"。

7 **将 FedletConfiguration.properties 文件从 Java Fedlet java/conf 目录复制到 Fedlet 主目录。**

8 **将身份认证提供者标准元数据 XML（可扩展标记语言）文件（通过步骤 1 获得）复制到 Fedlet 主目录。此文件必须命名为 idp.xml。**

9 **将 Java Fedlet XML（可扩展标记语言）元数据文件 (sp.xml) 导入到身份认证提供者中。**

对于 Oracle OpenSSO 8.0 Update 2，请在 OpenSSO 8.0 管理控制台中，使用“常见任务”下的“注册远程服务提供者”工作流导入 Java Fedlet 服务提供者元数据，并将 Java Fedlet 服务提供者添加到信任环中。

接下来的操作 根据您的要求，对 Java Fedlet 继续进行任何其他配置。

▼ 配置 .NET Fedlet

1. **在身份认证提供者端，生成身份认证提供者的 XML（可扩展标记语言）元数据，并将该元数据保存在一个名为 `idp.xml` 的文件中。**

   对于 Oracle OpenSSO 8.0 Update 2，请使用 `exportmetadata.jsp`。例如：

   ```
   http://opensso-idp.example.com:8080/opensso/saml2/jsp/exportmetadata.jsp
   ```

2. **在服务提供者端，解压缩 Fedlet ZIP 文件（如有必要）。**

3. **将以下文件从 .NET Fedlet `asp.net/conf` 文件夹复制到您的应用程序的 `App_Data` 文件夹：**

   - `sp.xml-template`
   - `sp-extended.xml-template`
   - `idp-extended.xml-template`
   - `fedlet.cot-template`

4. **在 `App_Data` 文件夹中，重命名所复制的文件并将 `-template` 从每个名称中删除。**

5. **在复制到 `App_Data` 文件夹并进行重命名的文件中，替换下表中显示的标记：**

标记	替换为
`FEDLET_COT`	远程身份认证提供者和 .NET Fedlet 服务提供者应用程序所属的信任环 (Circle Of Trust, COT) 的名称。
`FEDLET_ENTITY_ID`	.NET Fedlet 服务提供者应用程序的 ID（名称）。例如：`fedletsp`
`FEDLET_DEPLOY_URI`	.NET Fedlet 服务提供者应用程序的 URL。例如：`http://fedletsp.example.com/myFedletApp`
`IDP_ENTITY_ID`	远程身份认证提供者的 ID（名称）。例如：`openssoidp`

6. **将身份认证提供者标准元数据 XML（可扩展标记语言）文件（通过步骤 1 获得）复制到您的应用程序的 `App_Data` 文件夹中。此文件必须命名为 `idp.xml`。**

7. **将 `Fedlet.dll` 和 `Fedlet.dll.config` 文件从 .NET Fedlet `asp.net/bin` 文件夹复制到应用程序的 `bin` 文件夹中。**

8. **将 .NET Fedlet XML（可扩展标记语言）元数据文件 (`sp.xml`) 导入到身份认证提供者中。**

   对于 Oracle OpenSSO 8.0 Update 2，请在 OpenSSO 8.0 管理控制台中，使用“常见任务”下的“注册远程服务提供者”工作流导入 .NET Fedlet 服务提供者元数据，并将 .NET Fedlet 服务提供者添加到信任环中。

接下来的操作 根据您的要求，对 .NET Fedlet 继续进行任何其他配置。

OpenSSO 8.0 Update 2 中 Fedlet 的新功能

Oracle OpenSSO 8.0 Update 2 提供了有关 Fedlet 的以下新功能：

Fedlet 版本信息 (CR 6941387)

Oracle OpenSSO Fedlet 包含版本信息。解压缩 Fedlet 软件包（ZIP 文件）中的文件后，通过查看以下文件之一可以确定 Fedlet 的版本：

- Java Fedlet： `java/conf/FederationConfig.properties`
- .NET Fedlet： `asp.net/bin/Fedlet.dll.config`

Java Fedlet 密码加密和解密 (CR 6930477)

Java Fedlet 在 `fedlet.war` 文件中提供了 `fedletEncode.jsp`，以对 `storepass` 和 `keypass` 密码进行加密。默认情况下，将为每个 Fedlet 生成不同的加密密钥。要更改此加密密钥，请设置 Fedlet `FederationConfig.properties` 文件中的 `am.encryption.pwd` 属性。

Java Fedlet 支持签名和加密

Java Fedlet 支持 XML（可扩展标记语言）签名验证和已加密 `assertion` 和 `NameID` 元素及其相应属性的解密。

▼ 配置 Java Fedlet 以支持签名和加密

1. **使用 `keytool` 实用程序创建名为 `keystore.jks` 的密钥库文件。**
2. **将用于签名的私钥（和公共证书，如果适用）和用于加密的私钥（和公共证书，如果适用）添加到 `keystore.jks` 文件中。**

3 创建 **.storepass** 文件。

4 将密码添加到 **.storepass** 文件中。要对密码加密，请使用 **fedletEncode.jsp**。

5 创建 **.keypass** 文件。

6 将密码添加到 **.keypass** 文件中。要对密码加密，请使用 **fedletEncode.jsp**。

7 如果使用的是明文密码，请注释掉 **FederationConfig.properties** 文件中的以下行：

```
com.sun.identity.saml.xmlsig.passwordDecoder=
    com.sun.identity.fedlet.FedletEncodeDecode
```

8 在 **FederationConfig.properties** 文件中设置以下属性的完整路径，其中 *path* 是相应文件的完整路径：

```
com.sun.identity.saml.xmlsig.keystore=path/keystore.jks
com.sun.identity.saml.xmlsig.storepass=path/.storepass
com.sun.identity.saml.xmlsig.keypass=path/.keypass
```

9 使用 **keytool** 导出签名证书。例如：

```
keytool -export -keystore keystore.jks -rfc -alias test
```

此工具会提示您输入用于访问 `keystore.jks` 的密码，然后生成证书。

10 如果需要加密证书，请使用 **keytool** 导出证书，如上一步骤所示。（也可以将同一证书用于签名和加密。）

11 创建 **KeyDescriptor XML**（可扩展标记语言）块，然后将签名证书添加到该块中。示例如下，请注意 **KeyDescriptor** 元素的 **use="signing"** 标记：

```
<KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
</KeyDescriptor>
```

12 **创建另一个 KeyDescriptor XML（可扩展标记语言）块，然后将加密证书添加到该块中。示例如下，请注意 KeyDescriptor 元素的 use="encryption" 标记：**

```
<KeyDescriptor use="encryption">
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <X509Data>
          <X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </X509Certificate>
        </X509Data>
      </KeyInfo>
<EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
  <KeySize xmlns="http://www.w3.org/2001/04/xmlenc#">128</KeySize>
</EncryptionMethod>
</KeyDescriptor>
```

13 **在 Java Fedlet sp.xml 文件中，将包含有签名和加密证书的 XML（可扩展标记语言）块添加在 SPSSODescriptor 元素的下面。如需样例 SPSSODescriptor 元素，请参见**示例 4–1。

AuthnRequestsSigned 属性设置为 true，以将 Java Fedlet 配置为对所有验证请求签名。

14 **在 Java Fedlet sp-extended.xml 文件中，设置以下元素的值：**

- signingCertAlias 包含密钥库中 XML（可扩展标记语言）签名证书的别名。
- encryptionCertAlias 包含密钥库中 XML（可扩展标记语言）加密证书的别名。

15 **要实施 Java Fedlet 服务提供者加密的内容，请在 sp-extended.xml 文件中将以下属性设置为 true：**

- wantAssertionEncrypted
- wantNameIDEncrypted
- wantAttributeEncrypted

16 **要实施 Java Fedlet 服务提供者签名的内容和计划签名的内容，请将以下属性设置为 true：**

- idp.xml 文件中的 wantAuthnRequestsSigned，告知 Fedlet 要签名的内容。
- sp.xml 文件中的 AuthnRequestsSigned 和 WantAssertionsSigned，告知身份认证提供者 Fedlet 计划要签名的内容。
- sp-extended.xml 文件中的 wantArtifactResponseSigned，告知 Fedlet 要签名的内容。

- sp-extended.xml 文件中的 wantPOSTResponseSigned
- sp-extended.xml 文件中的 wantLogoutRequestSigned
- sp-extended.xml 文件中的 wantLogoutResponseSigned

如果身份认证提供者需要对特定消息进行签名，请将 idp-extended.xml 文件中的相应属性设置为 true。例如，wantLogoutRequestSigned 和 wantLogoutResponseSigned。

注 – 如果在 sp-extended.xml 文件中设置了属性，请将此信息告知身份认证提供者管理员，以便可以在身份认证提供者中进行必要的配置更改。

---

17 重新启动 Java Fedlet Web 容器。

18 将 Java Fedlet sp.xml 文件导入到身份认证提供者中。

示例 4–1 Java Fedlet 样例 SPSSODescriptor 元素

```
<EntityDescriptor entityID="fedlet"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata">

<SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="false"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<b><KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>

</KeyDescriptor></b>
<b><KeyDescriptor use="encryption">
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <X509Data>
          <X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
```

```
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </X509Certificate>
        </X509Data>
      </KeyInfo>

<EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
<KeySize xmlns="http://www.w3.org/2001/04/xmlenc#">128</KeySize>
</EncryptionMethod>
</KeyDescriptor></b>
<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat
><AssertionConsumerService index="1"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://server.sun.com:7070/fedlet/fedletapplication"/>
</SPSSODescriptor>
</EntityDescriptor>
```

Java Fedlet 支持属性查询 (CR 6930476)

Java Fedlet 支持 SAMLv2 属性查询，以针对特定的身份属性值查询诸如 Oracle OpenSSO 8.0 Update 2 等身份认证提供者。可以将 Fedlet 配置为对查询进行签名和加密。发出 Fedlet 查询必须要进行签名，但加密是可选操作。

▼ 配置 Java Fedlet 以支持属性查询

1. 启用 XML（可扩展标记语言）签名以对属性查询进行签名，如第 43 页中的“Java Fedlet 支持签名和加密”中所述。

2. 将前面步骤中生成的证书添加到 Fedlet `sp.xml` 文件的 `RoleDescriptor` 元素中。在以下示例中，有两个要在其中粘贴证书的 `KeyDescriptor` 标记。一个用于签名，另一个用于加密。如果未启用加密，则不需要 `KeyDescriptor use="encryption"` 标记。

```
<RoleDescriptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:query="urn:oasis:names:tc:SAML:metadata:ext:query"
      xsi:type="query:AttributeQueryDescriptorType"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            --certificate--
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
  </KeyDescriptor>
  <KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            --certificate--
```

```
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    <EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
<xenc:KeySize
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">128</xenc:KeySize>
    </EncryptionMethod>
  </KeyDescriptor>
</RoleDescriptor>
```

3 **在 Java Fedlet `sp-extended.xml` 文件中，指定 `signingCertAlias` 属性的值，并指定 `encryptionCertAlias` 属性（如果已配置）的值。**

如果计划将身份认证提供者配置为对声明进行加密，也请对 `NameID` 元素加密。因此，`wantNameIDEncrypted` 属性的值必须设置为 `true`。将 XML（可扩展标记语言）代码添加到 `AttributeQueryConfig` 元素中。例如：

```
<Attribute name="signingCertAlias">
      <Value>test</Value>
</Attribute>
<Attribute name="encryptionCertAlias">
      <Value>test</Value>
</Attribute>
<Attribute name="wantNameIDEncrypted">
      <Value>true</Value>
</Attribute>
```

在此示例中，`test` 是样例密钥的别名。

4 **将 Java Fedlet 元数据文件 (`sp.xml`) 导入到身份认证提供者中。**

此外，在身份认证提供者中执行其他配置步骤以支持 Fedlet 属性查询。

请求和响应的 .NET Fedlet 加密和解密 (CR 6939005)

.NET Fedlet 可以对外出的 XML（可扩展标记语言）请求进行加密，对接收的针对 NameID、Attribute 和 Assertion 元素的响应进行解密。

▼ 配置 .NET Fedlet 以支持对请求和响应进行加密和解密

1 **使用 Microsoft Management Console 的证书管理单元将 X.509 证书导入到本地计算机帐户内的个人文件夹中。要使用此管理单元，请参见以下 Microsoft 文章：**

`http://msdn.microsoft.com/en-us/library/ms788967.aspx`

2 **通过查看“属性”对话框并输入值，为此证书指定一个易记名称。（保存此值以在步骤 4 中使用。）**

3. 为 Internet Information Server (IIS) 使用的用户帐户设置相应的权限，以允许其读取证书，如 Microsoft 文章中所述。例如：

   a. 在“证书管理单元”中，浏览到“操作”->“所有任务”，然后浏览到“管理私钥”。

   b. 为运行 IIS 的用户帐户（通常为 NETWORK SERVICE）指定允许读取权限。

4. 在 .NET Fedlet 的扩展元数据文件 (`sp-extended.xml`) 中，指定在步骤 2 中指定的易记名称，作为 `encryptionCertAlias` 属性的值。例如：

```
<Attribute name="encryptionCertAlias">
<Value>MyFedlet</Value>
```

5. 在 .NET Fedlet 的服务提供者元数据文件 (`sp.xml`) 中，添加加密密钥的 `KeyDescriptor`。

   使用之前使用过的 Microsoft Management Console 证书管理单元导出以 Base64 编码形式包括在 `KeyDescriptor` XML（可扩展标记语言）块中的证书的公钥。此 `KeyDescriptor` 必须是 `SPSSODescriptor` 中的第一个子元素。例如：

```
<KeyDescriptor use="encryption">
           <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
             <ds:X509Data>
               <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
               </ds:X509Certificate>
             </ds:X509Data>
           </ds:KeyInfo>
           <EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
           <KeySize
xmlns="http://www.w3.org/2001/04/xmlenc#">128</KeySize>
           </EncryptionMethod>
</KeyDescriptor>
```

6. 重新启动与 .NET 应用程序关联的应用程序池。

接下来的操作 要测试此配置，请使用样例应用程序。此外，设置以下属性以加密请求和解密来自包含所配置元数据的适当更改的身份认证提供者的响应：

- Assertion：将 `sp-extended.xml` 元数据文件中的 `wantAssertionEncrypted` 属性设置为 `true`，以让 .NET Fedlet 对接收的来自身份认证提供者的响应中的 `EncryptedAssertion` 元素进行解密。

- Attribute：将 sp-extended.xml 元数据文件中的 wantAttributeEncrypted 属性设置为 true，以让 .NET Fedlet 对接收的来自身份认证提供者的响应中的 EncryptedAttribute 元素进行解密。
- NameID：将 idp-extended.xml 元数据文件中的 wantNameIDEncrypted 属性设置为 true，以让 .NET Fedlet 对外出请求中的 NameID 元素进行加密。在 sp-extended.xml 中设置此相同属性，以让 .NET Fedlet 对接收的来自身份认证提供者的响应中的 EncryptedID 元素进行解密。

请求和响应的 .NET Fedlet 签名 (CR 6928530)

.NET Fedlet 支持对外出 XML（可扩展标记语言）请求（如 Authn 请求）和注销请求签名。

▼ 配置 .NET Fedlet 以支持对请求和响应进行签名：

1. **使用 Microsoft Management Console 的证书管理单元将 X.509 证书导入到本地计算机帐户内的个人文件夹中。要使用此管理单元，请参见以下 Microsoft 文章：**

   http://msdn.microsoft.com/en-us/library/ms788967.aspx

2. **通过查看“属性”对话框并输入值，为此证书指定一个易记名称。（保存此值以在步骤 4 中使用。）**

3. **为 Internet Information Server (IIS) 使用的用户帐户设置相应的权限，以允许其读取证书，如 Microsoft 文章中所述。例如：**

   a. **在“证书管理单元”中，浏览到“操作”->“所有任务”，然后浏览到“管理私钥”。**

   b. **为运行 IIS 的用户帐户（通常为 NETWORK SERVICE）指定允许读取权限。**

4. **在 .NET Fedlet 的扩展元数据文件 (sp-extended.xml) 中，指定在步骤 2 中指定的易记名称，作为 signingCertAlias 属性的值。例如：**

```
<Attribute name="signingCertAlias">
<Value>MyFedlet</Value>
```

5. **在 .NET Fedlet 的服务提供者元数据文件 (sp.xml) 中，添加签名密钥的 KeyDescriptor。**

   使用之前使用过的 Microsoft Management Console 证书管理单元导出以 Base64 编码形式包括在 KeyDescriptor XML（可扩展标记语言）块中的证书的公钥。此 KeyDescriptor 必须是 SPSSODescriptor 中的第一个子元素。例如：

```
<KeyDescriptor use="signing">
             <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                   <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
```

```
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
</KeyDescriptor>
```

6 **重新启动与 .NET 应用程序关联的应用程序池。**

.NET Fedlet 单点注销（CR 6928528 和 CR 6930472）

.NET Fedlet 支持身份认证提供者启动的单点注销和服务提供者启动的单点注销。要实现单点注销，.NET Fedlet 样例应用程序应将 `logout.aspx` 和 `spinitiatedslo.aspx` 文件包含在 `asp.net/SampleApp` 文件夹中。要了解 Fedlet 单点注销功能如何工作，请部署 .NET Fedlet 样例应用程序。

▼ 配置 .NET Fedlet 服务提供者应用程序以支持单点注销：

1 **如果未配置 .NET Fedlet，请执行 `Readme` 文件中的步骤。**

2 **将 `logout.aspx` 和 `spinitiatedslo.aspx` 文件复制到 .NET 应用程序的公共内容中。**

3 **对您应用程序的配置文件进行以下更改：**

- 在 `sp.xml` 文件中，确保 `logout.aspx` 文件的路径指向您应用程序的该文件的正确位置。
- 在 `idp.xml` 文件中（或在进行身份认证提供者配置期间），确保 `spinitiatedslo.aspx` 文件的路径指向您应用程序的该文件的正确位置。

4 **如果要对注销请求和注销响应进行签名，请将 `sp-extended.xml` 和 `idp-extended.xml` 文件中的以下属性设置为 `true`：**

- `wantLogoutRequestSigned`
- `wantLogoutResponseSigned`

5 **将 Fedlet 服务提供者元数据文件 (`sp.xml`) 导入到身份认证提供者中。**

此外，通知身份认证提供者管理员您已经为 Fedlet 服务提供者配置单点注销，以便可以对身份认证提供者配置进行任何所需的其他更改。

.NET Fedlet 服务提供者启动的单点登录 (CR 6928525)

.NET Fedlet 支持 SAMLv2 服务提供者启动的单点登录 (SSO)。此外，还必须支持工件，以允许 .NET Fedlet 接收工件，然后通过 SOAP 使用分发身份认证提供者的工件解析服务进行解析。

.NET Fedlet 样例应用程序表明如何配置单点登录。为应用程序安装必要的工件后，需要提供特定的 URI，以在身份认证提供者成功进行验证后接收包含 SAMLv2 响应的 HTTP POST。以下代码示例表明如何在 .NET 应用程序中检索此信息：

示例 4–2 在 .NET Fedlet 应用程序中检索 `AuthnResponse` 的代码示例

```
AuthnResponse authnResponse = null;
try
{
      ServiceProviderUtility spu = new ServiceProviderUtility(Context);
      authnResponse = spu.GetAuthnResponse(Context);
}
catch (Saml2Exception se)
{
      // invalid AuthnResponse received
}
catch (ServiceProviderUtilityException spue)
{
      // issues with deployment (reading metadata)
}
```

如果应用程序接收 SAMLv2 响应，将用声明信息填充 `authnResponse` 对象。样例应用程序表明如何从此对象检索属性和主题信息。

.NET Fedlet 支持多个身份认证提供者和搜索服务 (CR 6928524)

.NET Fedlet 支持多个身份认证提供者和身份认证提供者搜索服务。

在某些部署中，您可能想要为 .NET Fedlet 配置多个身份认证提供者（如 Oracle OpenSSO 8.0 Update 2）。对于每个要添加的附加身份认证提供者，请执行以下任务。

▼ 配置 .NET Fedlet 以支持多个身份认证提供者

1. **获取附加身份认证提供者的 XML（可扩展标记语言）元数据文件。**
2. **将附加身份认证提供者元数据文件命名为 `idp`*n*`.xml`，其中 *n* 是您添加的身份认证提供者的序列。例如，将第二个身份认证提供者文件命名为 `idp2.xml`，将第三个身份认证提供者文件命名为 `idp3.xml`，依此类推。此过程使用 `idp2.xml` 作为文件名。**

3. **将步骤 2 中的 `idp2.xml` 文件复制到应用程序的 `App_Data` 文件夹中。**

4. **将此新身份认证提供者添加到 .NET Fedlet 信任环中。**

   将新身份认证提供者添加到现有信任环中：

   在应用程序的 `App_Data` 文件夹的 `fedlet.cot` 文件中，使用逗号 (,) 作为分隔符，将新 IDP 实体 ID（由 `idp2.xml` 元数据文件中的 `entityID` 属性表示）附加到 `sun-fm-trusted-providers` 属性的值。

   将新身份认证提供者添加到新信任环中：

   a. **在应用程序的 `App_Data` 文件夹中，创建一个名为 `fedlet2.cot` 的新文件。使用现有的 `fedlet.cot` 作为模板，但将 `cot-name` 属性的值更改为新信任环的名称（例如 `cot2`）。同时包含新身份认证提供者实体 ID 和 Fedlet 实体 ID 作为 `sun-fm-trusted-providers` 属性的值，两个实体 ID 用逗号 (,) 分隔。**

   b. **在 `sp-extended.xml` 文件中，将新信任环的名称添加到 `cotlist` 属性的值中。例如，对于名为 `cot2` 的信任环：**

      ```
      <Attribute name="cotlist">
      <Value>saml2cot</Value>
      <Value>cot2</Value>
      </Attribute>
      ```

5. **在应用程序的 `App_Data` 文件夹中，创建一个新的 `idp2-extended.xml` 文件作为新身份认证提供者的扩展元数据。使用现有的 `idp-extended.xml` 文件作为模板，但将 `entityID` 更改为新身份认证提供者实体 ID。如果为该身份认证提供者创建了新信任环，则将 `cotlist` 属性的值更改为该信任环的名称。确保附加身份认证提供者是远程身份。**

6. **重新启动与 .NET Fedlet 应用程序关联的应用程序池。**

7. **Fedlet 元数据 XML（可扩展标记语言）文件 (`sp.xml`) 必须导入到附加身份认证提供者中，并添加到身份认证提供者实体所属的同一信任环中。将 `sp.xml` 文件导入到身份认证提供者中，或者将该文件提供给身份认证提供者管理员进行导入。**

.NET Fedlet 支持身份认证提供者搜索服务 (CR 6928524)

在此方案中，为 .NET Fedlet 配置同属一个信任环中的多个身份认证提供者，而您要将 Fedlet 配置为使用身份认证提供者搜索服务来确定首选的身份认证提供者。

必须为与 .NET Fedlet 配合使用的身份认证提供者配置搜索服务。有关在 Oracle OpenSSO 8.0 Update 2 中配置身份认证提供者搜索服务的信息，请参见以下文档集合：`http://docs.sun.com/coll/1767.1`。

▼ 将 .NET Fedlet 配置为使用身份认证提供者搜索服务：

1. 在 .NET Fedlet `fedlet.cot` 文件中，将 `sun-fm-saml2-readerservice-url` 属性设置为 SAMLv2 读取器服务 URL。例如：

   ```
   sun-fm-saml2-readerservice-url=http://discovery.common.com/opensso/saml2reader
   ```

2. 重新启动与 .NET Fedlet 应用程序关联的应用程序池。

Oracle OpenSSO Fedlet 的常见问题和解决方法

待编写

文档勘误表

Fedlet Java API（应用编程接口）参考在 Oracle OpenSSO 8.0 Update 2 Java API（应用编程接口）参考中提供，Oracle OpenSSO 8.0 Update 2 Java API（应用编程接口）参考包含在以下文档集合中：`http://docs.sun.com/coll/1767.1`

注 – `getPolicyDecisionForFedlet` 方法在 OpenSSO 8.0 Update 2 发行版本中不受支持。

◆ ◆ ◆ 第 5 章

5 将 OpenSSO 8.0 Update 2 与 Oracle Access Manager 相集成

本章提供有关使用 OpenSSO 8.0 Update 2 和 Oracle Access Manager 10g 或 11g 实现单点登录的说明。此信息作为《Sun OpenSSO Enterprise 8.0 Integration Guide》中的第 3 章 "Integrating Oracle Access Manager"中所包含概念信息的补充。此使用案例通过采用 Oracle Access Manager 会话提供受 OpenSSO 保护的应用程序的单点登录体验。配置的 OpenSSO 验证模块可基于 Oracle Access Manager 会话生成 OpenSSO 会话。

集成步骤概述

1. 第 55 页中的 "准备工作"
2. Unpacking the Integration Bits
3. Building source files for Oracle Access Manager in OpenSSO
4. 第 58 页中的 "（可选）在 Oracle Access Manager 中生成 OpenSSO 的验证方案"
5. 第 59 页中的 "使用 Oracle Access Manager 和 Oracle OpenSSO STS 配置单点登录"
6. 第 61 页中的 "测试单点登录"
7. 第 61 页中的 "（可选）在 Oracle Access Manager 中安装 Oblix AuthScheme"

准备工作

在尝试安装 OpenSSO 8.0 Update 2 以与 Oracle Access Manager 相集成前，请确保对以下组件具有访问权限：

opensso.zip	此 zip 文件包含安装和配置 OpenSSO 8.0 Update 2 所需的 opensso.war 文件、集成源代码、配置文件和其他工具。
OpenSSO 代理	当受 OpenSSO 保护的应用程序确实可以使用 Oracle Access Manager 建立的验证会话时，将使用 OpenSSO 代理。

Oracle Access Manager 10g 或 11g	从 Oracle Web 站点下载 Oracle Access Manager。请参见 Oracle Fusion Middleware 11gR1 Software Downloads 页面。
Oracle Web Gate 10g 或 11g	为 OpenSSO 和 Oracle Webgate 都支持的容器下载 Oracle Webgate。此时，Sun Web Server 7.x 是这两种产品都支持的唯一容器。请参见 Oracle Fusion Middleware 11gR1 Software Downloads 页面
Oracle Access Manager SDK 10g 或 11g	下载 Oracle Access Manager。编译和生成用于 Oracle Access Manager 集成的 OpenSSO 验证模块时，需要使用 SDK。 请参见 Oracle Fusion Middleware 11gR1 Software Downloads 页面
OpenSSO C-SDK 2.2	（可选）在 Oracle Access Manager 中创建验证模块以生成 OAM 会话时，需要使用 OpenSSO C-SDK。从 OpenSSO 角度来说，这可能不是常见的使用案例。请参见《Sun OpenSSO Enterprise 8.0 C API Reference for Application and Web Policy Agent Developers》中的“Where is the C SDK?”

集成部分分解说明

`opensso/integrations/oracle` 目录包含用于编译和生成自定义验证模块和其他插件的源及配置。有关使用案例选项和相关信息，请参见《Sun OpenSSO Enterprise 8.0 Integration Guide》中的第 3 章 “Integrating Oracle Access Manager”。下表汇总 `opensso/integrations/oracle` 目录下的文件和每个文件的描述。

`README.html`	这是您现在正在阅读的文件。
`build.xml`	用于在 OpenSSO 中生成 Oracle Access Manager 的自定义验证模块的 ant 生成文件
`config`	在 OpenSSO 中创建 Oracle Access Manager 的验证模块时所需的配置文件。 ■ `OblixAuthService.xml` Oracle Access Manager 验证模块的验证服务文件 ■ `OblixAuthModule.xml` Oracle Access Manager 的验证模块回叫。

默认情况下，此文件为空，但为执行配置必须存在该文件。

- `OblixAuth.properties`

  存储用于验证的国际化密钥的属性文件

`lib`

默认情况下，此目录为空。此 `lib` 目录必须包含以下库以编译源库。

- `jobaccess.jar`

  从 Oracle Access Manager SDK 复制此文件。

- `openfedlib.jar`、`amserver.jar` 和 `opensso-sharedlib.jar`

  从 opensso.war 复制这些文件

- `servlet.jar` **或** `javaee.jar`

  复制 GlassFish `lib` 目录。理想情况下，任何具有标准 Java EE 类（如 `javax.servlet.http.Cookie`）的 JAR 文件均可。

`source`

包含以下源文件的目录：

- `com/sun/identity/authentication/oblix/OblixAuthModule.java`
- `com/sun/identity/authentication/oblix/OblixAuthModule.java`
- `com/sun/identity/authentication/oblix/OblixPrincipal.java`
- `com/sun/identity/saml2/plugins/OAMAdapter.java`

  此类是用于 SAML 服务提供者的 SAML2 插件适配器。此类使用 OpenSSO 会话服务对 Oracle Access Manager 进行远程验证。

oamauth（可选）

此目录包含 OpenSSO 的 Oblix 验证方案的源文件。这是一个基于 C 的验证模块，它利用 OpenSSO C-SDK 进行验证。

- oam/solaris/authn_api.c

  此文件可实现 OpenSSO 的 Oblix 自定义验证方案。

- oam/solaris/include/*.h

  编译验证方案所需的所有标头文件。

- oam/solaris/AMAgent.properties

  样例 OpenSSO 代理配置文件。验证方案验证 OpenSSO 会话时需要使用此文件。

在 OpenSSO 中生成 Oracle Access Manager 的源文件

使用 ant 脚本可以生成源文件。必须在 PATH 中安装和配置兼容的 ant 脚本。

▼ 生成 Oracle Access Manager 的源文件

1 **运行以下命令：**

```
cd $openssozipdir/integrations/oracle; ant -f build.xml
```

此命令可在 `$openssozipdir/integrations/oracle/dist` 目录中生成源文件和 `fam_oam_integration.jar`。

2 **将验证模块捆绑到 OpenSSO WAR 文件中。**

a. **创建临时目录并解压缩 WAR 文件 `opensso.war`。示例：**

```
# mkdir /export/tmp
# cd /export/tmp
# jar -xvf opensso.war
```

从现在起，`/export/tmp` 将用作 WAR 临时区域，并用宏 `$WAR_DIR` 表示。

b. **将 `$openssozipdir/integrations/oracle/dist/fam_oam_integration.jar` 复制到 `$WAR_DIR/WEB-INF/lib`。**

c. **将 `$openssozipdir/integrations/oracle/config/OblixAuth.properties` 复制到 `$WAR_DIR/WEB-INF/classes`。**

d. **将 `$openssozipdir/integrations/oracle/config/OblixAuthModule.xml` 复制到 `$WAR_DIR/config/auth/default` 以及 `$WAR_DIR/config/auth/default_en` 目录中。**

e. **使用 `$WAR_DIR` 中的 `jar cvf opensso.war` 重新压缩 WAR 文件 `opensso.war`。**

示例待编写

（可选）在 Oracle Access Manager 中生成 OpenSSO 的验证方案

注意：这不是常见的使用案例。除非有必要（如在 SAML2 服务提供者使用案例中），否则无需生成此验证方案。

要生成 Oblix 验证方案，必须自定义 `makefile`。此外，由于它是基于 C 的验证模块，因此依赖于操作系统。

▼ 在 Oracle Access Manager 中生成 OpenSSO 的验证方案

开始之前 验证方案文件位于 `$openssozipdir/integrations/oracle/oamauth/solaris` 目录下。

1. **下载并配置 OpenSSO C-SDK 2.2 版本。**
   `authn_api.c` 文件包含对 `AMAgent.propeties` 文件的引用。相应地修改此文件。
2. **针对您的环境自定义 `makefile`。**
   例如，指定 `gcc` 编译位置。此外，编辑 `LDFLAGS` 以指向 OpenSSO C-SDK `lib` 目录。
3. **运行 `make` 命令。**
   `make` 命令会生成一个 `authn_api.so` 文件。

使用 Oracle Access Manager 和 Oracle OpenSSO STS 配置单点登录

▼ 使用 Oracle Access Manager 和 Oracle OpenSSO 8.0 Update 2 配置单点登录

准备工作：必须已经安装并配置 Sun Java System Web Server 7.x。有关 Web Server 的安装说明，请参见 Sun Java System Web Server Documentation Wiki。

1. **在 Sun Java System Web Server 7.x 上安装 OpenSSO。**
2. **在支持的容器上安装 OpenSSO 策略代理，并配置该代理以与 OpenSSO 一起使用。**
   有关安装说明，请参见《Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for J2EE Agents》或《Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for Web Agents》《Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for Web Agents》。
3. **安装和配置 Oracle Access Manager。**
   请参见《Oracle Access Manager 安装指南 10g (10.1.4.3)》
4. **通过 Oracle Access Manager 安装和配置 Oracle Access Manager SDK。**
   请参见《Oracle Access Manager 安装指南 10g (10.1.4.3)》

5 **在安装有 OpenSSO 服务器的同一 Web 容器中安装 Oracle Webgate。(Sun Web Server 7.x)**

配置 OpenSSO，以使其仅保护 OpenSSO Web 应用程序的 `deployURI/UI/*`。示例：`/opensso/UI/.../*`

有关 Oracle Access Manager 策略、资源和其他配置详细信息，请查看 Oracle Access Manager 管理指南。 在 OpenSSO Enteprise 中取消对任何其他 URL 的保护。这适用于简单的单点登录集成方案，但策略评估基于的是完整集成和其他部署依存关系。

6 **在 OpenSSO 中配置验证模块。**

a. **访问 OpenSSO 控制台。**

浏览器将重定向到 Oracle Access Manager 以进行验证。进行成功验证后，OpenSSO 将显示“登录”页面。使用 OpenSSO 管理员用户名和密码登录。

b. **将 Oracle 验证模块服务 XML（可扩展标记语言）文件导入到 OpenSSO 配置中。**

该验证模块服务可以通过命令行 `ssoadm` 实用程序以及基于浏览器的 ssoadm.jsp 进行加载。

c. **访问 `http://host:port/opensso/ssoadm.jsp`。**

d. **选择 create-service 选项。**

e. **从 `$openssozipdir/integrations/oracle/config/OblixAuthService.xml` 复制并粘贴 XML（可扩展标记语言）文件，然后单击“提交”。**

该操作会将验证模块服务加载到 OpenSSO 配置中。

f. **将验证模块注册到验证核心服务中。**

核心服务包含一系列验证程序。在 `http://host:port/opensso/ssoadm.jsp` 中选择 `register-auth-module` 选项。输入 `com.sun.identity.authentication.oblix.OblixAuthModule` 作为验证模块类名。

g. **检验验证模块是否已经注册到默认的领域中。**

使用 URL http://host:port/opensso 访问 OpenSSO。在 OpenSSO 控制台中，单击默认领域，然后单击“验证”选项卡。单击“新建”以创建名为 OblixAuth 的新验证模块。

h. **在“验证”选项卡上，选择 OblixAuth 验证模块。**

配置 Oblix SDK 目录。启用“仅检查远程用户标头”，并将远程标头名称指定为 `OAM_REMOTE_USER`。根据部署，此参数是可配置的。

7 **（可选）在 OpenSSO 核心验证服务中启用“忽略配置文件”选项。**

在 OpenSSO 控制台中，转至“配置”>“核心”>“领域属性”>“用户配置文件”。选择“已忽略”，然后单击“保存”。

此配置可禁止 OpenSSO 在成功进行验证后搜索现有的用户配置文件。但是，如果 OpenSSO 和 Oracle Access Manager 使用的用户系统信息库完全相同，那么不需要此步骤。转至“管理控制台”->“配置”->“核心”->“领域属性”->“用户配置文件”。选择“已忽略”，然后单击“保存”。

8 **编辑 Web 服务器启动脚本以包含 Oracle Access Manager SDK 共享库。**

更新 `startserv` 脚本中的 `LD_LIBRARY_PATH` 以包含 `$ACCESSDKDIR/oblix/lib` 中的共享库。

9 **重新启动包含 OpenSSO 和 Oracle Webgate 的 Sun Web Server。**

10 **将“Web 代理的登录 URL”值更新为 `http://openssohost:openssoport/deployURI/UI/Login?module=OblixAuth`。**

测试单点登录

从受 OpenSSO 保护的应用程序访问保护的资源。如果您未通过验证，浏览器会将您重定向至 Oracle Access Manager“登录”页面。成功登录后，将创建一个 OpenSSO 会话，并最终重定向回到受策略代理保护的应用程序 URL。根据策略，将允许或拒绝您访问受保护的应用程序。

（可选）在 Oracle Access Manager 中安装 Oblix AuthScheme

如果在验证 OpenSSO 会话时必须生成 Oracle Access Manager 会话，该操作很有用。有关相关使用案例的信息，请参见《Sun OpenSSO Enterprise 8.0 Integration Guide》中的第 3 章 “Integrating Oracle Access Manager”。

Oblix 验证方案显示为 C 验证模块，此验证方案使用 OpenSSO C-SDK 2.2 版本来验证 OpenSSO 会话。Oblix 中的 OpenSSO 验证方案使用 `AMAgent.properties` 中 OpenSSO 客户端的配置。在配置 验证模块前，必须先自定义此文件。内部版本说明 指定了此文件的位置。在配置验证方案前，必须将编译的 `authn_api.so` 和其他 C-SDK 库复制到 `$OAM_INSTALL_DIR/access/oblix/lib` 目录。《《Sun OpenSSO 8.0 集成指南》》显示一个样例截屏，说明如何配置 Oracle 验证方案，但这只应用作参考。有关详细信息，请参见最新的 Oracle Access Manager 文档。

将 OpenSSO 8.0 Update 2 与 Oracle Access Manager 相集成

本部分提供有关使用 OpenSSO 8.0 Update 2 和 Oracle Access Manager 版本 10.1.4.0.1. 和 11g 实现单点登录的说明。此信息作为《Sun OpenSSO Enterprise 8.0 Integration Guide》中的第 3 章 "Integrating Oracle Access Manager"中所包含概念信息的补充。此使用案例通过采用 Oracle Access Manager 会话提供受 OpenSSO 保护的应用程序的单点登录体验。配置的 OpenSSO 验证模块可基于 Oracle Access Manager 会话生成 OpenSSO 会话。