Oracle® OpenSSO Update 2 版本說明

Beta

文件號碼：821–2841
2010 年 7 月

ORACLE

101014@24999

目錄

前言

《Oracle OpenSSO 8.0 Update 2 版本說明》中提供了有關下載和安裝 OpenSSO Update 2 軟體的資訊。本文件中還包含了有關自 OpenSSO Update 1 發行版本以來的軟體變更的資訊。

本書適用對象

這些版本說明旨在供已安裝和部署 Oracle OpenSSO 8.0 的企業管理員和開發人員使用。您應該先熟悉核心產品文件中介紹的概念和程序。

相關書籍

這些版本說明是以下 URL 上提供的核心 Oracle OpenSSO 8.0 產品文件的補充：http://docs.sun.com/app/docs/coll/1767.1。

相關的協力廠商網站參考

本文件中提供了協力廠商 URL 以供參考，另亦提供其他相關的資訊。

備註 – Oracle 不保證本文件中提到的協力廠商網站可用。對於此類網站或資源中提供的或透過它們獲得的任何內容、廣告、產品或其他材料，Oracle 並不表示認可，也不承擔任何責任。對於因使用或依靠此類網站或資源中提供的或透過它們獲得的任何內容、產品或服務而造成的或連帶產生的任何實際或名義上的損壞或損失，Oracle 概不負責，也不承擔任何責任。

文件、支援與訓練

請造訪以下網站以獲得其他資源：

- 文件 (http://docs.sun.com)
- 支援 (http://www.oracle.com/us/support/systems/index.html)
- 訓練 (http://education.oracle.com) – 按一下左側瀏覽位址列中的 [Sun] 連結。

Oracle 歡迎您提出寶貴意見

Oracle 歡迎您對其文件的品質和作用提出寶貴的意見和建議。如果您發現任何錯誤或有任何其他改進建議，請前往 http://docs.sun.com 並按一下 [Feedback] (建議)。指明文件的標題和文件號碼，以及具體章節和頁碼 (如果有)。如果您需要回覆，請告訴我們。

Oracle Technology Network (http://www.oracle.com/technetwork/index.html) 上提供了與 Oracle 軟體相關的一系列資源：

- 在 Discussion Forums (http://forums.oracle.com) 上討論技術問題和解決方案。
- 從 Oracle By Example (http://www.oracle.com/technology/obe/start/index.html) 上獲得有關實際操作的逐步自學課程。
- 下載範例代碼 (http://www.oracle.com/technology/sample_code/index.html)。

印刷排版慣例

下表說明本書在印刷排版上的慣例。

表 P–1 印刷排版慣例

字型	含義	範例
`AaBbCc123`	指令、檔案與目錄的名稱，以及螢幕畫面輸出	編輯您的 `.login` 檔案。 使用 `ls -a` 可列出所有檔案。 `machine_name% you have mail.`
`AaBbCc123`	您鍵入的內容，與螢幕畫面輸出相對	`machine_name%` **`su`** **密碼:**
aabbcc123	預留位置：替代成實際名稱或值	移除檔案的指令爲 `rm` *filename*。

表 P–1 印刷排版慣例 (續)

字型	含義	範例
AaBbCc123	書籍標題、新術語及要強調的術語	請閱讀《使用者指南》中的第 6 章。 **快取**是儲存在本機的副本。 請**勿**儲存此檔案。 **備註**：某些強調的項目在線上會以粗體顯示。

指令中的 Shell 提示範例

下表中列出了預設 UNIX 系統提示和 Oracle Solaris 作業系統中包含的 Shell 超級使用者提示。請注意，指令範例中顯示的預設系統提示會隨 Oracle Solaris 發行版本的不同而不同。

表 P–2 Shell 提示

Shell	提示
Bash shell、Korn shell 和 Bourne shell	`$`
適用於超級使用者的 Bash shell、Korn shell 和 Bourne shell	`#`
C shell	`machine_name%`
適用於超級使用者的 C shell	`machine_name#`

◆ ◆ ◆ 第 1 章

1 關於 OpenSSO 8.0 Update 2

本章包含以下主題：

OpenSSO 8.0 Update 2 中的新增功能

OpenSSO 8.0 Update 2 中提供了安全性代表字元服務和 OpenSSO Fedlet 的增強功能。

安全性代表字元服務的增強功能

現在，安全性代表字元服務提供以下新功能：

- 支援 TokenType 以產生特定 Web 服務提供者安全性代表字元。
- 對於以 X509 和使用者名稱安全性代表字元作爲請求者，同時支援非對稱和傳輸連結。
- 使用使用者名稱透過 SSL 配置 OpenSSO STS 時，透過使用者名稱安全性代表字元實現 SSL/傳輸連結。
- 爲使用 useKey 作爲 Web 服務用戶端公開金鑰的非對稱 KeyType 頒發 SAML 金鑰所有者安全性代表字元，及頒發 Web 服務用戶端 X509 安全性代表字元。
- WSDL 根據安全性代表字元配置動態更新。
- 支援透過 Web 服務提供者公開金鑰進行加密。
- 在將靜態使用者名稱密碼儲存到配置存放區之前對其進行加密。
- 支援透過 WS-Trust 請求將使用者名稱代表字元作爲代理安全性代表字元。

- 支援頒發 SAML 載送程式代表字元。
- 新的 Web 服務安全性認證模組 WSSAuth 支援摘要密碼驗證。
- 新的 OAMAuth 認證模組透過將 Oracle Access Manager 與 OpenSSO 配合使用，支援單次登入。

如需詳細資訊，請參閱第 3 章 「使用安全性代表字元服務」。

Fedlet 的增強功能

現在，Fedlet 提供以下新功能：

- 在 .NET Fedlet 中支援加密
- 在 .NET Fedlet 中支援簽名
- 現在，.NET Fedlet 支援單次登出
- .NET Fedlet 支援服務提供者啓動的單次登入和小工具
- 在 .NET Fedlet 中支援多個識別提供者和識別提供者探索
- 在 Fedlet 的特性和配置檔案中提供版本資訊
- 新密碼服務提供者介面實作
- 支援屬性查詢
- 支援單次登出

如需詳細資訊，請參閱第 4 章 「使用 Oracle OpenSSO Fedlet」。

OpenSSO 8.0 Update 2 的硬體和軟體需求

請參閱「Sun OpenSSO Enterprise 8.0 Update 1 Release Notes」中的「Hardware and Software Requirements For OpenSSO Enterprise 8.0 Update 1」

新 Web 容器支援

OpenSSO 8.0 Update 2 支援在「Sun OpenSSO Enterprise 8.0 Update 1 Release Notes」中的「Support for New Web Containers」中介紹的 Web 容器，以及下列新 Web 容器：

- Oracle WebLogic Server 10g Release 3 (10.3)

OpenSSO 8.0 Update 2 的問題和解決方法

CR 6959610：在生產環境中應移除 OpenSSO 8.0 Update 2 範例

OpenSSO 8.0 Update 2 範例可能會導致潛在的安全性問題。

解決方法：如果在生產環境中部署 OpenSSO 8.0 Update 2，請移除範例，以免發生任何潛在的安全性問題。

CR 6964648：WebLogic Server 10.3.3 需要新的 Java 安全性權限

如果您在啓用了安全性管理員的情況下在 Oracle WebLogic Server 10.3.3 上部署 OpenSSO 8.0 Update 2，將需要其他 Java 安全性權限。

解決方法：將以下權限增加到 WebLogic Server 10.3.3 `weblogic.policy` 檔案中：

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

CR 6939443：在 WebLogic Server 10.3.x 上透過 LDAP 檢查或 OCSP 檢查進行憑證認證失敗

由於 10.3.0 和 10.3.1 等舊版 Oracle WebLogic Server 中存在的問題，導致在啓用 LDAP 檢查或 OSCP 檢查的情況下進行憑證認證時會失敗。

解決方法：該問題在 WebLogic Server 10.3.3 中已得到解決。若要將憑證認證與 LDAP 檢查或 OCSP 檢查配合使用，請將 OpenSSO Update 2 與 WebLogic Server 10.3.3 配合使用。

CR 6967026：配置程式無法透過 GlassFish 2.1.x 連線到啓用了 LDAPS 的目錄伺服器實例

如果將 GlassFish Enterprise Server v2.1.1 或 v2.1.2 部署爲 OpenSSO 8.0 Update 2 Web 容器，配置程式將無法連線到啓用了 LDAPS 的目錄伺服器實例。

解決方法：若要使用啓用了 LDAPS 的目錄伺服器並使用 GlassFish 作爲 Web 容器，請部署 GlassFish Enterprise Server v2.1。

CR 6948937：在 WebLogic Server 10.3.3 管理主控台中啓用 OpenSSO 8.0 Update 2 導致發生異常

如果在 WebLogic Server 10.3.3 管理主控台中部署 OpenSSO 8.0 Update 2 (`opensso.war`)，並按一下 [啓動] 以允許 OpenSSO 8.0 Update 2 開始接收請求，在啓動 WebLogic Server 網域的主控台中會拋出異常。

備註：啓動 OpenSSO 8.0 Update 2 後，它會保持啓動狀態，並且在停止 OpenSSO 8.0 Update 2 並重新啓動之前不會再次拋出異常。

解決方法：將 OpenSSO 8 Update 2 `opensso-client-jdk15.war` 檔案中的 `saaj-impl.jar` 檔案複製到 WebLogic Server 10.3.3 配置的 `endorsed` 目錄中，具體作業如下：

1. 停止 Oracle WebLogic Server 10.3.3 網域。
2. 視需要解壓縮 OpenSSO 8.0 Update 2 `opensso.zip` 檔案。
3. 建立一個暫存目錄並將 *zip-root*`/opensso/samples/opensso-client.zip` 檔案解壓縮到該目錄中，此處的 *zip-root* 是您解壓縮 `opensso.zip` 檔案的位置。例如：

   ```
   cd zip-root/opensso/samples
   mkdir ziptmp
   cd ziptmp
   unzip ../opensso-client.zip
   ```
4. 建立一個暫存目錄並從 `opensso-client-jdk15.war` 中擷取 `saaj-impl.jar` 檔案。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` 目錄下建立一個名爲 `endorsed` 的新目錄 (如果 `endorsed` 不存在)，此處的 `WEBLOGIC_JAVA_HOME` 是 WebLogic Server 配置爲要使用的 JDK。
6. 將 `saaj-impl.jar` 檔案複製到 `WEBLOGIC_JAVA_HOME/jre/lib/endorsed` 目錄中。
7. 啓動 WebLogic Server 網域。

CR 6959373：執行 updateschema 程序檔後，需要重新啓動 Web 容器

執行 `updateschema.sh` 或 `updateschema.bat` 程序檔後，您必須重新啓動 OpenSSO 8.0 Update 2 Web 容器。

CR 6961419：執行 updateschema.bat 程序檔需要密碼檔案

`updateschema.bat` 程序檔會執行若干個 `ssoadm` 指令。因此，在 Windows 系統上執行 `updateschema.bat` 之前，請爲 `amadmin` 使用者建立一個包含純文字使用者密碼的密碼檔案。`updateschema.bat` 程序檔會提示您提供密碼檔案的路徑。該程序檔會在終止之前移除該密碼檔案。

OpenSSO 8.0 Update 2 文件

除本文件之外，在以下集合中還提供了其他 OpenSSO 8.0 文件：

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

文件問題

OpenSSO 8.0 Update 2 具有以下文件相關的問題：

- 第 15 頁的 「CR 6958580：主控台線上說明中介紹了不受支援的探索代理程式」
- 第 15 頁的 「CR 6967006：主控台線上說明中未介紹 OAMAuth 和 WSSAuth 認證模組」
- 第 16 頁的 「CR 6953582：Fedlet Java API 參照應爲公開參照」
- 第 16 頁的 「CR 6953579：OpenSSO Fedlet README 檔案應介紹單次登出功能」

CR 6958580：主控台線上說明中介紹了不受支援的探索代理程式

OpenSSO 8.0 Update 2 管理主控台線上說明中介紹了探索代理程式，儘管這些代理程式不受支援。

解決方法：無。忽略線上說明中有關探索代理程式的資訊。

CR 6967006：主控台線上說明中未介紹 OAMAuth 和 WSSAuth 認證模組

OpenSSO 8.0 Update 1 管理主控台線上說明中未介紹 Oracle Access Manager (OAM) 和 Web 服務安全性 (WSS) 認證模組。

解決方法：如需有關這兩個認證模組的資訊，請參閱第 3 章 「使用安全性代表字元服務」

CR 6953582：Fedlet Java API 參照應為公開參照

Fedlet Java API 公開參照作爲 Oracle OpenSSO 8.0 Update 2 Java API 參照的一部分提供，Oracle OpenSSO 8.0 Update 2 Java API 參照在以下文件集合中提供：`http://docs.sun.com/coll/1767.1`。

備註：OpenSSO 8.0 Update 2 不支援 `getPolicyDecisionForFedlet` 方法，儘管此方法存在於 Java API 參照中。

CR 6953579：OpenSSO Fedlet README 檔案應介紹單次登出功能

Fedlet README 檔案未介紹單次登出功能。

解決方法：對於 Oracle OpenSSO 8.0 Update 2，Fedlet 單次登出功能在第 4 章 「使用 Oracle OpenSSO Fedlet」中介紹。

附加資訊和資源

您還可以在以下位置找到其他有用的資訊和資源：

- 第 16 頁的 「棄用內容通知和公告」
- 第 17 頁的 「如何報告問題和提供建議」
- 第 17 頁的 「適用於殘障人士的協助工具功能」
- 第 17 頁的 「相關的協力廠商網站」

- Oracle 進階客戶系統服務：

  `http://www.oracle.com/us/support/systems/advanced-customer-services/index.html`
- 軟體產品：`http://www.oracle.com/us/sun/sun-products-map-075562.html`
- SunSolve：`http://sunsolve.sun.com/`
- Sun 開發人員網路 (SDN)：http://developers.sun.com/
- Sun 開發人員服務：`http://developers.sun.com/services/`

棄用內容通知和公告

- 服務管理服務 (SMS) API (`com.sun.identity.sm` 套裝軟體) 和 SMS 模型將不會包括在將來的 OpenSSO 發行版本中。
- Unix 認證模組和 Unix 認證說明程式 (`amunixd`) 將不會包括在將來的 OpenSSO 發行版本中。

- 《Sun Java System Access Manager 7.1 版本說明》中指出，Access Manager `com.iplanet.am.sdk` 套裝軟體 (通常稱爲 Access Manager SDK [AMSDK]) 及所有相關的 API 和 XML 範本將不會包括在將來的 OpenSSO 發行版本中。

  因此，在移除 AMSDK 時也會移除「舊有模式」選項 和支援。

  現在不提供遷移選項，預計將來也不會提供。Oracle 識別管理員提供使用者佈建解決方案，可用以代替 AMSDK。如需有關識別管理員的詳細資訊，請參見 `http://www.oracle.com/products/middleware/identity-management/identity-manager.html`。

如何報告問題和提供建議

如果您有關於 OpenSSO 8.0 Update 2 或後續修補程式版本的問題，請造訪支援資源網站：`http://sunsolve.sun.com/`。

此網站中包含一些連結，透過這些連結可以造訪知識庫、線上支援中心和產品追蹤器，還可瞭解維護方案和支援連絡號碼。如要您要請求獲得關於某個問題的幫助，請提供以下資訊：

- 問題描述，包括問題發生的時間及其對您作業的影響
- 電腦類型、作業系統版本、Web 容器及版本、JDK 版本和 OpenSSO 版本，包括可能與問題相關的任何修補程式或其他軟體
- 重現問題的步驟
- 任何錯誤記錄或核心傾印

適用於殘障人士的協助工具功能

若要獲得自本媒體發佈以來所發行的協助工具功能，請查閱 Section 508 產品評估 (可以透過請求獲得) 來確定哪些版本最適合部署易存取解決方案。

如需 Oracle 對協助工具的支援的相關資訊，請參見 `http://www.oracle.com/index.html`。

相關的協力廠商網站

本文件中提供了協力廠商 URL 以供參考，另亦提供其他相關的資訊。

備註 – Oracle 不保證本文件中提到的協力廠商網站可用。對於此類網站或資源中提供的或透過它們獲得的任何內容、廣告、產品或其他材料，Oracle 並不表示認可，也不承擔任何責任。對於因使用或依靠此類網站或資源中提供的或透過它們獲得的任何內容、產品或服務而造成的或連帶產生的任何實際或名義上的損壞或損失，Oracle 概不負責，也不承擔任何責任。

◆ ◆ ◆ 第 2 章

安裝 OpenSSO 8.0 Update 2

本章包含以下主題：

OpenSSO 8.0 Update 2 安裝概述

OpenSSO 8.0 Update 2 作爲修補程式 **TBS** 提供。

在您安裝 OpenSSO 8.0 Update 2 (或後續修補程式) 之前，請閱讀本文件中有關新功能、硬體和軟體需求以及問題和解決方法的資訊。

OpenSSO 8.0 Update 2 中包含一個 `opensso.war` 檔案，您可以使用以下方法進行安裝：

- **修補現有的 OpenSSO 8.0 部署**：使用 Update 2 中的 `ssopatch` 公用程式修補現有的 OpenSSO 8.0 部署，如本章中所述。

  備註 - Oracle 僅支援修補 OpenSSO 8.0 發行版本。例如，支援使用 OpenSSO 8.0 Update 2 修補 OpenSSO 8.0。

- **安裝新的 OpenSSO 8.0 Update 2 部署**：安裝並配置 OpenSSO 8.0 Update 2 `opensso.war` 檔案，如「Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide」中所述。
- **建立新的專用 WAR 檔案**：使用 `createwar` 程序檔從 Update 2 `opensso.war` 檔案建立以下新 WAR 檔案之一：
  - 「僅 OpenSSO 管理主控台」WAR
  - 「已經發行的認證 UI 伺服器」WAR
  - 「僅 OpenSSO 伺服器」WAR，不含管理主控台
  - 「IDP 探索服務」WAR

  如需相關資訊，請參閱「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」。
- **修補現有的專用 OpenSSO WAR 檔案**：使用 Update 2 中的 `ssopatch` 公用程式修補現有的專用 OpenSSO 8.0 WAR 檔案，如「Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide」中的第 23 章「Patching OpenSSO Enterprise 8.0」中所述

備註 – 如果您執行的是 Access Manager 7.1 或 Access Manager 7 2005Q4，並且想要升級到 Update 2，請執行以下步驟：

1. 將 Access Manager 7.x 升級到 OpenSSO 8.0，如「Sun OpenSSO Enterprise 8.0 Upgrade Guide」中所述。
2. 套用 Update 2 修補程式，如本章中所述。

OpenSSO 8.0 Update 2 修補程式

Sun 會定期發行 OpenSSO 8.0 Update 2 的修補程式。如需有關這些修補程式的資訊，請定期到此處查閱。

計劃修補作業

▼ 計劃針對 OpenSSO 8.0 的修補作業的步驟：

1 **閱讀第 21 頁的「ssopatch 公用程式概述」。**

2 **為您的平台安裝修補公用程式，如第 22 頁的「安裝 ssopatch 公用程式」中所述。**

3 **取得有關您的現有 WAR 檔案的資訊，以確定現有 WAR 檔案是否已自訂或修改，如第 24 頁的「將 OpenSSO WAR 檔案與其內部清單進行比較」中所述。**

4. 將現有 WAR 檔案與 Update 2 WAR 檔案進行比較，以傳回在原始 WAR 中自訂的檔案、在新 WAR 檔案中更新的檔案，以及在兩個 WAR 版本之間增加或刪除的檔案，如第 24 頁的「比較兩個 OpenSSO WAR 檔案」中所述。
5. 備份和歸檔現有 Opensso WAR 檔案，如第 22 頁的「備份 OpenSSO WAR 檔案」中所述。
6. 修補您的 OpenSSO WAR 檔案，如第 25 頁的「修補 OpenSSO WAR 檔案」中所述。
7. 執行 `updateschema` 程序檔，如第 28 頁的「執行 updateschema 程序檔」中所述。

   備註 - 如果修補從 `opensso.war` 產生的專用 WAR 檔案 (例如「僅 OpenSSO 伺服器」WAR、「僅 OpenSSO 管理主控台」WAR、「已經發行的認證 UI 伺服器」WAR 或「IDP 探索服務」WAR)，請參閱第 27 頁的「修補專用 OpenSSO WAR」。

ssopatch 公用程式概述

`ssopatch` 公用程式是一個 Java 指令行公用程式，該公用程式在 Solaris 和 Linux 系統上作爲 `ssopatch` 提供，在 Windows 上作爲 `ssopatch.bat` 提供。

備註 - `ssopatch` 在 OpenSSO 8.0 Update 2 中的語法與在 OpenSSO 8.0 發行版本中相比已經發生了很大的變化。如需有關新語法的資訊，請參閱第 28 頁的「執行 updateschema 程序檔」。

`ssopatch` 修補公用程式執行以下功能：

- 將 OpenSSO WAR 與其原始清單進行比較，以確定 WAR 檔案是否已自訂或修改
- 比較兩個 OpenSSO WAR 檔案，以確定兩個檔案之間的差異，包括對原始 WAR 檔案進行的任何自訂，以及新 WAR 檔案中的任何變更
- 針對產生新的已修補 OpenSSO WAR 檔案所需的檔案建立一個臨時區域

下載並解壓縮 OpenSSO 8.0 Update 2 ZIP 檔案 (`opensso_80U2.zip`) 後，修補公用程式及相關檔案將包括在 *zip-root*`/opensso/tools` 目錄中的 `ssoPatchTools.zip` 檔案中，此處的 *zip-root* 是您解壓縮 `opensso_80U2.zip` 的位置。

`ssopatch` 公用程式使用清單檔案來確定特定 OpenSSO WAR 檔案包含的內容。清單檔案是一個包含以下內容的 ASCII 文字檔：

- 一個指示 OpenSSO WAR 檔案的具體版本的字串
- OpenSSO WAR 檔案中的所有單個檔案，以及每個檔案的總和檢查資訊

清單檔案通常被命名爲 `OpenSSO.manifest`，並儲存在 OpenSSO WAR 檔案的 `META-INF` 目錄中。

`ssopatch` 公用程式將其結果傳送給標準輸出 (`stdout`)。如果願意，您也可以透過將輸出重新導向至檔案來擷取 `ssopatch` 輸出。如果 `ssopatch` 成功完成，其將傳回零 (`0`) 結束代碼。如果發生錯誤，`ssopatch` 將傳回非零的結束代碼。

安裝 ssopatch 公用程式

在您安裝 ssopatch 公用程式之前，請先完成以下作業：

- 下載並解壓縮 OpenSSO 8.0 Update 2 ZIP 檔案 (opensso_80U2.zip)。
- 將 JAVA_HOME 環境變數設定爲指向 JDK 1.5 或更高版本。

安裝 ssopatch 公用程式的步驟：

1. 在 *zip-root*/opensso/tools 目錄中找到 ssoPatchTools.zip 檔案，此處的 *zip-root* 是您解壓縮 opensso_80U2.zip 的位置。
2. 建立一個新目錄以解壓縮 ssoPatchTools.zip 檔案。例如：ssopatchtools
3. 將 ssoPatchTools.zip 檔案解壓縮到該新目錄中。
4. 如果想要不提供完整路徑而從不同於其目前目錄的其他目錄執行 ssopatch 公用程式，請將該公用程式增加到 PATH 變數中。

下表中描述了 ssoPatchTools.zip 中的檔案。

檔案或目錄	描述
README	描述 ssopatch 的讀我檔案
/lib	所需的 ssopatch JAR 檔案
/patch	updateschema 和 updateschema.bat 程序檔及相關的 XML 檔案
/resources	所需的特性檔案
ssopatch 和 ssopatch.bat	適用於 Solaris、Linux 和 Windows 系統的公用程式

備份 OpenSSO WAR 檔案

開始之前，請備份現有的 OpenSSO WAR 檔案和配置資料：

- 將現有的 OpenSSO WAR 檔案複製到一個安全的位置。然後，如果出於某些原因需要復原 Update 2，您可以重新部署 WAR 檔案的備份副本。
- 備份配置資料，如「Sun OpenSSO Enterprise 8.0 Administration Guide」中的第 15 章「Backing Up and Restoring Configuration Data」中所述。

執行 ssopatch 公用程式

若要執行 ssopatch 公用程式，請遵循如下用法：

```
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]
```

其中的選項說明如下：

- `-war-file|-o` 指定先前部署的 WAR 檔案 (例如 `opensso.war`) 的路徑。
- `-manifest|-m` 指定您要建立的清單檔案的路徑。清單檔案將從 `-war-file|-o` (如果提供此選項) 指示的 WAR 檔案產生。
- `-war-file-compare|-c` 指定要與 `-war-file|-o` 指示的 WAR 檔案進行比較的 WAR 檔案的路徑。
- `-staging|-s` 指定要將 OpenSSO WAR 中的檔案寫入其中的臨時區域的路徑。
- `-locale|-l` 指定要使用的語言環境。如果未指定該選項，`ssopatch` 將使用預設的系統語言環境。
- `-override|-r` 將忽略這兩個 WAR 檔案的修訂版檢查。修訂版檢查可確定 WAR 檔案的修訂版，並且僅當修訂版相容時，該檢查才能繼續。此選項允許您忽略該檢查。

  預設爲 false (執行修訂版檢查)。
- `-overwrite|-w` 將覆寫現有臨時區域內的檔案。預設爲 false (不覆寫檔案)。

將 OpenSSO WAR 檔案與其內部清單進行比較

使用此程序可確定 OpenSSO WAR 檔案自下載以來是否已被自訂或修改。

ssopatch 公用程式會產生一個新的內部清單檔案，然後將此內部清單與儲存在原始 OpenSSO WAR 檔案的 META-INF 目錄中的清單進行比較。

將 OpenSSO WAR 檔案與其內部清單進行比較的步驟：

1. 執行 ssopatch 以將 OpenSSO WAR 檔案與其內部清單進行比較。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
```

此範例顯示了對原始 WAR 檔案的以下變更：

- images/login-origimage.jpg 包含在 opensso.war 中，但未在原始清單中找到。
- 相對於原始清單，images/login-backimage.jpg 已在 opensso.war 中自訂。
- 相對於原始清單，WEB-INF/classes/amConfigurator.properties 檔案已在 opensso.war 中自訂。

比較兩個 OpenSSO WAR 檔案

使用此程序可比較兩個 OpenSSO WAR 檔案，以顯示已經執行以下作業的檔案：

- 已在原始 OpenSSO WAR 中自訂
- 已在新 OpenSSO WAR 檔案中更新
- 已在兩個 OpenSSO WAR 版本之間增加或刪除

比較兩個 OpenSSO WAR 檔案的步驟：

1. 執行 ssopatch 以比較兩個 WAR 檔案。在此範例中，使用 -override 選項來忽略兩個 WAR 檔案之間的修訂版檢查：

```
./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
```

此範例顯示了已在新 WAR 檔案中更新和自訂的檔案。

修補 OpenSSO WAR 檔案

使用此程序可建立新的臨時區域，在其中原始 WAR 檔案將與新 WAR 檔案合併在一起。

此作業將比較每個 WAR 檔案的清單，然後顯示：

- 在原始 WAR 檔案中自訂的檔案
- 在新 WAR 檔案中更新的檔案
- 在兩個 WAR 檔案版本之間增加或移除的檔案

然後，`ssopatch` 會將相應的檔案複製到暫存目錄中，於此您必須先增加任何自訂，然後再建立和部署新的已修補 WAR。

建立臨時區域以修補 OpenSSO WAR 檔案的步驟：

1. 雖然 `ssopatch` 不會修改原始 `opensso.war` 檔案，但我們建議您備份此檔案，以供您在需要復原已修補的 `opensso.war` 檔案時使用。
2. 執行 `ssopatch` 以建立臨時區域。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
```

在此範例中，`/tmp/staging` 是 `ssopatch` 複製檔案的臨時區域。

使用上一步驟的結果依需要更新臨時區域中的檔案。

使用下表來確定在產生新的已修補 WAR 檔案之前可能需要對每個檔案執行的動作。

ssopatch 結果	**所需的說明和動作**
`File not in original war` *filename*	指示的檔案在原始 WAR 檔案中不存在，但在最新版本的 WAR 檔案中存在。 **動作**：無
`File updated in new war` *filename*	指示的檔案在原始 WAR 檔案和新 WAR 檔案中均存在，並且已在最新版本的 WAR 檔案中更新。在原始 WAR 檔案中未執行任何自訂。 **動作**：無
`File customized` *filename*	指示的檔案在原始 WAR 檔案和新 WAR 檔案中均存在，並且已在原始版本的 WAR 檔案中自訂，但未在最新版本的 WAR 檔案中更新。 **動作**：無
`May require manual customization` *filename*	檔案在原始 WAR 檔案和新 WAR 檔案中均存在，並且已在原始版本的 WAR 檔案中自訂及在最新版本的 WAR 檔案中更新。 **動作**：如果需要檔案中的自訂，您必須手動將這些自訂增加到暫存目錄中的新的已更新檔案中。
`File was customized in original, but not found in new war`	檔案在原始 WAR 檔案中存在，但在新 WAR 檔案中不存在。 **動作**：無

後續步驟

1. 從臨時區域中的檔案建立一個新的 OpenSSO WAR 檔案。例如：

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

   此處的 `/patched/opensso.war` 是新的已修補 OpenSSO WAR 檔案
2. 使用原始部署 URI 將 `/patched/opensso.war` 檔案重新部署到 Web 容器。例如，`/opensso`

OpenSSO 配置變更。新的 OpenSSO WAR 檔案可能包含原始 WAR 檔案中未包含的配置變更。將單獨爲每個修補程式記錄任何配置變更 (如果有)。請參閱修補程式文件和「Sun OpenSSO Enterprise 8.0 版本說明」，以瞭解有關任何配置變更的詳細資訊。(即使新 WAR 檔案中沒有任何配置變更，OpenSSO 清單檔案中的版本字串也會變更。)

如果您需要復原您的已修補版本，請解除部署已修補的 WAR 檔案，然後重新部署您的原始 WAR 檔案。

建立 OpenSSO WAR 清單檔案

OpenSSO 清單檔案是透過每個檔案的總和檢查資訊識別特定發行版本的 WAR 檔案中所有單個檔案的文字檔。

使用此程序可建立您可以包括在專用 OpenSSO WAR (例如「僅 OpenSSO 伺服器」WAR、「僅 OpenSSO 管理主控台」WAR、「已經發行的認證 UI 伺服器」WAR 或「IDP 探索服務」WAR) 中的清單檔案。

建立 OpenSSO WAR 清單檔案的步驟：

1. 執行 `ssopatch` 以建立 OpenSSO 清單檔案。例如：

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

   此處的 `opensso.war` 是現有的 OpenSSO WAR 檔案。

   `ssopatch` 公用程式會在 `/tmp` 目錄中建立一個名爲 `manifest` 的新清單檔案。
2. 若要允許修補 WAR 檔案，請將此新清單檔案複製到 `opensso.war` 檔案的 `META-INF` 目錄中。例如：

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

修補專用 OpenSSO WAR

如果您之前建立了專用 OpenSSO WAR (例如「僅 OpenSSO 伺服器」WAR、「僅 OpenSSO 管理主控台」WAR、「已經發行的認證 UI 伺服器」WAR 或「IDP 探索服務」WAR)，您可以使用 `ssopatch` 公用程式對其進行修補。

修補專用 OpenSSO WAR 的步驟：

1. 爲您的專用 OpenSSO WAR 建立一個清單檔案，如第 27 頁的 「建立 OpenSSO WAR 清單檔案」中所述。

備註：請在進行任何自訂之前，以原始 OpenSSO 8.0 `opensso.war` (與 Sun 最初提供時相同) 爲基礎建立清單檔案。如果在自訂之後建立清單，`ssopatch` 可能會使用 Update 2 中的檔案而非您的自訂，因此在修補之後您將需要重新進行自訂。

2. 從 OpenSSO 8.0 Update 2 `opensso.war` 檔案產生專用 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` 公用程式比較您的舊 WAR 檔案與新 WAR 檔案。
4. 爲新的專用 WAR 檔案產生臨時區域，如第 25 頁的 「建立臨時區域以修補 OpenSSO WAR 檔案的步驟：」中所述。
5. 重新部署新的專用 WAR 檔案。

執行 updateschema 程序檔

執行 `ssopatch` 之後，請在 Solaris 或 Linux 系統上執行 `updateschema.sh`，或在 Windows 系統上執行 `updateschema.bat`。該程序檔將更新 OpenSSO 伺服器版本、新增預設伺服器特性、增加 Update 2 中的錯誤修復和增強功能所需的新屬性模式。若要更新伺服器版本，您必須執行 `updateschema`。

開始之前

- `updateschema.sh` 或 `updateschema.bat` 程序檔需要 Update 2 版本 (或更高版本) 的 `ssoadm` 指令行公用程式。因此，在您執行此程序檔之前，請先安裝 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` 程序檔會執行若干個 `ssoadm` 指令。因此，在 Windows 系統上執行 `updateschema.bat` 之前，請爲 `amadmin` 使用者建立一個包含純文字使用者密碼的密碼檔案。`updateschema.bat` 程序檔會提示您提供密碼檔案的路徑。該程序檔會在終止之前移除該密碼檔案。

執行 updateschema 程序檔的步驟：

1. 變更到 *patch-tools*`/patch` 目錄，此處的 *patch-tools* 是您解壓縮 `ssoPatchTools.zip` 的位置。
2. 執行 `updateschema.sh` 或 `updateschema.bat`。例如，在 Solaris 系統上：

   ```
   ./updateschema.sh
   ```

3. 當程序檔提示您時，請提供以下資訊：
   - `ssoadm` 公用程式的完整路徑 (不包括 `ssoadm` 本身)。例如：`/opt/ssotools/opensso/bin`

- amadmin 密碼

updateschema.sh 或 updateschema.bat 程序檔會將任何訊息或錯誤寫入標準輸出。

4. 重新啓動 OpenSSO 8.0 Update 2 Web 容器。

復原修補程式安裝

如果您需要復原您的修補程式安裝，只需重新部署原始 opensso.war 檔案 (或專用 WAR 檔案) 即可。

◆ ◆ ◆ 第 3 章

3 使用安全性代表字元服務

作爲一種可靠的授權服務，OpenSSO 安全性代表字元服務可頒發和驗證安全性代表字元。作爲 Web 服務安全性提供者，安全性代表字元服務可確保 Web 服務用戶端與 OpenSSO STS 服務本身之間的通訊的安全。自 OpenSSO 8.0 Update 2 以來，在安全性代表字元服務的功能上進行了諸多增強。

本章包含以下主題：

增加 WSSAuth 認證模組

Web 服務安全性認證模組可讓 OpenSSO 透過作爲認證代表字元接收並包含在從 Web 服務用戶端至 Web 服務提供者的請求中的摘要密碼驗證使用者名稱。

▼ 新增 Web 服務安全性認證模組實例的步驟：

1. **在 [Access Manager] 標籤中，按一下 [認證] 子標籤。**
2. **在 [模組實例] 區段中，按一下 [新增]。**
3. **在 [名稱] 欄位中，鍵入此 WSSAuth 認證模組實例的名稱。**
4. **對於 [類型]，請選擇 [WSSAuth]。**

5 **配置該 WSSAuth 認證模組實例。**

▼ 配置 WSSAuth 認證模組實例的步驟：

1 **在 [Access Manager] 標籤中，按一下 [認證] 子標籤。**

2 **在 [模組實例] 區段中，按一下您要配置的 WSSAuth 認證模組實例的名稱。**

3 **設定該 WSSAuth 認證模組實例的範圍屬性的值。**

下表中列出了您可以配置的屬性及其描述。

使用者搜尋屬性	待編寫
使用者範圍	待編寫
使用者密碼屬性	待編寫
認證層級	待編寫

增加 OAMAuth 認證模組

Oracle 認證模組可讓 OpenSSO 使先前通過了 Oracle Access Manager 認證的管理員通過 OpenSSO 的認證，並爲該管理員啓用 OpenSSO 的單次登入。該管理員將無需提供 OpenSSO 憑證。

▼ 新增 Oracle 認證模組實例的步驟：

1 **在 [Access Manager] 標籤中，按一下 [認證] 子標籤。**

2 **在 [模組實例] 區段中，按一下 [新增]。**

3 **在 [名稱] 欄位中，鍵入此 Oracle 認證模組實例的名稱。**

4 **對於 [類型]，請選擇 [OAMAuth]。**

5 **按一下 [確定]。**

6 **配置該 OAMAuth 認證模組實例。**

▼ 配置 Oracle 認證模組實例的步驟：

1. **在 [Access Manager] 標籤中，按一下 [認證] 子標籤。**
2. **在 [模組實例] 區段中，按一下您要配置的 OAMAuth 認證模組實例的名稱。**
3. **設定該 Oracle 認證模組實例的範圍屬性的值。**

   下表中列出了您可以配置的屬性及其描述。

遠端使用者標頭名稱	待編寫
允許的標頭值	[目前的值] 清單中顯示 [待編寫] ■ 若要將某標頭值增加到該清單中，請在 [新的值] 欄位中鍵入 [待編寫]，然後按一下 [增加]。 ■ 若要從 [目前的值] 清單中移除項目，請選取該項目，然後按一下 [移除]。
認證層級	待編寫

產生安全性代表字元

Oracle OpenSSO 安全性代表字元服務 (OpenSSO STS) 會在 Web 服務用戶端與 Web 服務提供者之間建立信任關係，然後在它們之間維護該信任關係。Web 服務可以信任僅由一個實體「OpenSSO STS」頒發的代表字元，而不必與多個用戶端進行通訊。透過這種方式，OpenSSO STS 可大大降低信任點管理的系統開銷。

下面幾個小節將爲您介紹如何確定安全性代表字元需求，以及如何配置安全性代表字元服務來產生安全性代表字元並驗證其是否符合這些需求。

將 Web 服務提供者註冊到 OpenSSO STS

當您新增 Web 服務提供者安全性代理程式設定檔時，系統會自動將 Web 服務提供者註冊到 OpenSSO STS。請參閱以下小節以獲得更多詳細資訊：

將 Web 服務提供者註冊到 OpenSSO STS 後，便可以配置 OpenSSO STS 以產生該 Web 服務提供者可以接受的 Web 用戶端安全性代表字元。

從 OpenSSO STS 請求 Web 服務用戶端安全性代表字元

您必須先確定 Web 服務提供者需要哪種類型的安全性代表字元，然後才能配置安全性代表字元服務以產生 Web 用戶端安全性代表字元。OpenSSO STS 支援 Liberty Alliance Project 安全性代表字元和 Web 服務互通性基本安全性設定檔安全性代表字元。

安全性代表字元產生處理流程

如果使用 Liberty Alliance Project 代表字元啓用安全性，HTTP 用戶端 (即瀏覽器) 會透過 Web 服務用戶端向 Web 服務提供者傳送存取請求。Web 服務安全性代理程式會將該請求重新導向至 OpenSSO STS 認證服務。如果 Liberty Alliance Project 安全性機制已可用，HTTP 安全性代理程式會執行重新導向。如果使用 WS-IBS 安全性，則 SOAP 安全性代理程式會執行重新導向。

OpenSSO STS 認證服務會確定 Web 服務提供者所註冊的安全性機制，然後擷取相應的安全性代表字元。成功認證之後，Web 服務用戶端會提供一個 SOAP 訊息內文，而 Web 服務用戶端一端的 SOAP 安全性代理程式會插入安全性標頭和代表字元。然後，在將請求傳送給 WSP 之前，將對該訊息進行簽名。

Web 服務提供者一端的 SOAP 安全性代理程式會先驗證 SOAP 請求中的簽名和安全性代表字元，然後將該請求轉寄給 Web 服務提供者自身。然後，Web 服務提供者將處理該請求，並將經 SOAP 安全性代理程式簽名的回應傳回給 Web 服務用戶端。然後，Web 服務用戶端一端的 SOAP 安全性代理程式會先驗證該簽名，之後再將該回應轉寄給 Web 服務用戶端。

下表中列出了 Liberty Alliance Project 交易支援的代表字元及其簡短描述。

表 3–1 請求者代表字元 - Liberty Alliance Project

代表字元	符合以下要求
X.509	■ 安全 Web 服務使用公開金鑰基礎架構 (PKI)，在該基礎架構中，Web 服務用戶端提供公開金鑰作爲識別請求程式以及向 Web 服務提供者進行認證的方式。 ■ 安全 Web 服務使用公開金鑰基礎架構 (PKI)，在該基礎架構中，Web 服務用戶端提供公開金鑰作爲識別請求程式以及向 Web 服務提供者進行認證的方式。

表 3–1 請求者代表字元 - Liberty Alliance Project *(續)*

載送程式代表字元	■ 安全 Web 服務使用安全性指定標記語言 (SAML) SAML 載送程式代表字元確認方法。 ■ WSC 爲 SAML 指定提供公開金鑰資訊，以此作爲向 Web 服務提供者認證請求程式的方式。 ■ 第二個簽名將該指定連結至 SOAP 訊息。 ■ 第二個簽名連結使用 Liberty Alliance Project 定義的規則。
SAML 代表字元	■ 安全 Web 服務使用 SAML 金鑰所有者確認方法。 ■ Web 服務用戶將一個 SAML 指定和一個數位簽名增加到 SOAP 標頭中。 ■ 該簽名還會隨附一個寄件者憑證或公開金鑰。 ■ 將使用 Liberty Alliance Project 定義的規則來處理傳送。

下表中列出了 WS-IBS 交易支援的代表字元及其簡短描述。

表 3–2 請求者代表字元 - WS-IBS

代表字元	符合以下要求
使用者名稱	■ 安全 Web 服務需要使用者名稱、密碼及已簽名的請求 (可選)。 ■ Web 服務用戶提供使用者名稱代表字元作爲識別請求程式的方式。 ■ Web 服務用戶提供密碼、共用密碼或等效密碼來向 Web 服務提供者認證身份。
X.509	安全 Web 服務使用 PKI (公開金鑰基礎架構)，在該基礎架構中，Web 服務用戶提供公開金鑰作爲識別請求程式以及完成向 Web 服務提供者進行認證的方式。
SAML 金鑰所有者	■ 安全 Web 服務使用 SAML 金鑰所有者確認方法。 ■ Web 服務用戶爲 SAML 指定提供公開金鑰資訊，以此作爲向 Web 服務提供者認證請求程式的方式。 ■ 第二個簽名將該指定連結至 SOAP 有效負載。
SAML 寄件者擔保	■ 安全 Web 服務使用 SAML 寄件者擔保確認方法。 ■ Web 服務用戶將一個 SAML 指定和一個數位簽名增加到 SOAP 標頭中。該簽名還會隨附一個寄件者憑證或公開金鑰。

使用安全性代表字元產生矩陣

使用安全性代表字元產生矩陣協助您配置 OpenSSO STS 以產生 Web 服務提供者所需的 Web 服務用戶端安全性代表字元。首先，在標題爲「OpenSSO STS 輸出代表字元」的最後一欄中，找到符合 Web 服務提供者代表字元需求的描述。然後，使用同一列中的參數值配置安全性代表字元服務。「代表字元產生矩陣圖例」提供有關表格標題和可

用選項的資訊。請參閱第 5.2.3 節「配置安全性代表字元服務的步驟」，以獲得詳細的配置說明。如需有關 Web 服務安全性及相關術語的一般資訊，請參閱：

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

安全性代表字元產生矩陣彙總了常用的安全性代表字元服務參數設定及 OpenSSO STS 根據這些設定產生的安全性代表字元類型。

表 3–3 安全性代表字元產生矩陣

列	訊息層級安全性連結	Web 服務用戶端代表字元	KeyType	代理代表字元	使用金鑰	OpenSSO STS 輸出代表字元
1	非對稱	X509	載送程式	是	否	SAML 載送程式，無證明金鑰
2	非對稱	使用者名稱	載送程式	是	否	SAML 載送程式，無證明金鑰
3	非對稱	X509	載送程式	否	否	SAML 載送程式，無證明金鑰
4	非對稱	使用者名稱	載送程式	否	否	SAML 載送程式，無證明金鑰
5	非對稱	X509	對稱	是	否	SAML 金鑰所有者，對稱證明金鑰
6	非對稱	使用者名稱	對稱	是	否	SAML 金鑰所有者，對稱證明金鑰
7	非對稱	X509	對稱	否	否	SAML 金鑰所有者，對稱
8	非對稱	使用者名稱	對稱	否	否	SAML 金鑰所有者，對稱證明金鑰
9	非對稱	X509	非對稱	否	Web 服務用戶端公開金鑰	SAML 金鑰所有者，非對稱證明金鑰

表 3–3 安全性代表字元產生矩陣 *(續)*

10	非對稱	X509	Oracle 專屬 SAML 寄件者擔保	是	否	SAML 寄件者擔保，無證明金鑰
11	非對稱	使用者名稱	Oracle 專屬 SAML 寄件者擔保	是	否	SAML 寄件者擔保，無證明金鑰
12	非對稱	X509	Oracle 專屬 SAML 寄件者擔保	否	否	錯誤
13	非對稱	使用者名稱	Oracle 專屬 SAML 寄件者擔保	否	否	錯誤
14	傳輸	使用者名稱	載送程式	是	否	SAML 載送程式，無證明金鑰
15	傳輸	使用者名稱	載送程式	否	否	SAML 載送程式，無證明金鑰
16	傳輸	使用者名稱	對稱	是	否	SAML 金鑰所有者，對稱
17	傳輸	使用者名稱	對稱	否	否	SAML 金鑰所有者，對稱證明金鑰
18	傳輸	使用者名稱	Oracle 專屬 SAML 寄件者擔保	是	否	SAML 寄件者擔保，無證明金鑰
19	傳輸	使用者名稱	Oracle 專屬 SAML 寄件者擔保	否	否	錯誤
20	非對稱	使用者名稱	非對稱	否	Web 服務用戶端公開金鑰	錯誤
21	傳輸	使用者名稱	非對稱	否	Web 服務用戶端公開金鑰	錯誤
22	非對稱	X509	非對稱	是	否	錯誤
23	非對稱	使用者名稱	非對稱	是	否	錯誤

表 3–3 安全性代表字元產生矩陣 (續)

24	傳輸	使用者名稱	非對稱	是	否	錯誤
25	非對稱	X509	非對稱	否	否	SAML 金鑰所有者，非對稱證明金鑰
26	非對稱	X509	否	否	否	SAML 金鑰所有者，非對稱證明金鑰
27	非對稱	使用者名稱	否	否	否	SAML 金鑰所有者，對稱證明金鑰
28	傳輸	使用者名稱	否	否	否	SAML 金鑰所有者，對稱證明金鑰

安全性代表字元服務的問題和解決方法

待編寫

配置問題和解決方法

待編寫

文件勘誤表

待編寫

◆ ◆ ◆ 第 4 章

使用 Oracle OpenSSO Fedlet

本節提供關於 Oracle OpenSSO Fedlet 的以下資訊：

關於 Oracle OpenSSO Fedlet

Oracle OpenSSO Fedlet 是一個輕量型服務提供者 (SP) 實作，可隨 Java 或 .NET 服務提供者應用程式一起部署，讓該應用程式可以使用 SAMLv2 協定與 Oracle OpenSSO 8.0 Update 2 等識別提供者 (IDP) 進行通訊。Fedlet 有兩個版本，可根據您的平台進行選擇：

- Java Fedlet 最初發行於 OpenSSO 8.0 中。如需相關資訊，請參閱「Sun OpenSSO Enterprise 8.0 Deployment Planning Guide」中的第 5 章「Using the OpenSSO Enterprise Fedlet to Enable Identity Federation」。
- .NET Fedlet 發行於 OpenSSO 8.0 Update 1 中。如需相關資訊，請參閱「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 按以下方式提供：

- 解壓縮 OpenSSO 8.0 Update 2 ZIP 檔案後，可在以下檔案中找到 Java Fedlet 和 .NET Fedlet：

  zip-root`/opensso/fedlet/fedlet-unconfigured.zip`，此處的 *zip-root* 是您解壓縮 Oracle OpenSSO 8.0 Update 2 ZIP 檔案的位置。
- 安裝 Oracle OpenSSO 8.0 Update 2 後，可以使用 [共用作業] 下面的 [建立 Fedlet] 工作流程在 OpenSSO 8.0 管理主控台中建立 Java Fedlet。

Oracle OpenSSO Fedlet 的需求

Fedlet 需要以下項目：

- Oracle OpenSSO 8.0 Update 2 支援的 Web 容器 (如果您打算部署 `fedlet.war`) 或與 Fedlet 整合在一起的 Java 服務提供者應用程式。請參閱第 12 頁的 「OpenSSO 8.0 Update 2 的硬體和軟體需求」。
- Microsoft Internet Information Server (IIS) 7.0 及更高版本 (如果您打算部署 .NET Fedlet)
- JDK 1.6.x 及更高版本

Oracle OpenSSO Fedlet 配置

本節介紹如何透過服務提供者應用程式對 Fedlet 進行初始配置：

- 第 40 頁的 「配置 Java Fedlet 的步驟：」
- 第 42 頁的 「配置 .NET Fedlet 的步驟：」

完成 Fedlet 的初始配置後，請繼續執行您需要的任何其他配置。下面是幾條注意事項：

- 如果修改了 Fedlet `sp.xml` 檔案，必須將該檔案重新匯入識別提供者。
- 如果在服務提供者一端進行了其他 Fedlet 配置變更，請將此資訊告知識別提供者管理員，以便可以在識別提供者一端進行所需的配置變更。

▼ 配置 Java Fedlet 的步驟：

1. **在識別提供者一端，為識別提供者產生 XML 中介資料，並將該中介資料儲存到名為 `idp.xml` 的檔案中。**

   對於 Oracle OpenSSO 8.0 Update 2，請使用 `exportmetadata.jsp`。例如：

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

2. **在服務提供者一端，解壓縮 Fedlet ZIP 檔案 (如有必要)。**

3. **建立 Fedlet 主目錄，這是 Fedlet 於其中讀取它的中介資料、信任圈及配置特性檔案的目錄。**

   預設位置是執行 Fedlet Web 容器的使用者的主目錄 (由 `user.home` JVM 特性指示) 下的 Fedlet 子目錄。例如，如果該主目錄爲 `/home/webservd`，則 Fedlet 主目錄爲：

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

   若要變更 Fedlet 預設主目錄，請將 JVM 執行時 `com.sun.identity.fedlet.home` 特性的值設定爲所需的位置。例如：

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

之後，Fedlet 將會從 /export/fedlet/conf 目錄中讀取它的中介資料、信任圈及配置檔案。

4 **將以下檔案從 Java Fedlet java/conf 目錄複製到 Fedlet 主目錄：**

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

5 **在 Fedlet 主目錄中，重新命名您所複製的檔案，並將 -template 從每個名稱中刪除。**

6 **在您複製到 Fedlet 主目錄並進行重新命名的檔案中，替代下表中所示的標記：**

標記	替代成
FEDLET_COT	遠端識別提供者和 Java Fedlet 服務提供者應用程式所屬的信任圈 (COT) 的名稱。
FEDLET_ENTITY_ID	Java Fedlet 服務提供者應用程式的 ID (名稱)。例如：fedletsp
FEDLET_PROTOCOL	Java Fedlet 服務提供者應用程式 (例如 fedlet.war) Web 容器的協定。例如：https
FEDLET_HOST	Java Fedlet 服務提供者應用程式 (例如 fedlet.war) Web 容器的主機名稱。例如：fedlet-host.example.com
FEDLET_PORT	Java Fedlet 服務提供者應用程式 (例如 fedlet.war) Web 容器的連接埠號碼。例如：80
FEDLET_DEPLOY_URI	Java Fedlet 服務提供者應用程式的 URL。例如：http://fedletsp.example.com/myFedletApp
IDP_ENTITY_ID	遠端識別提供者的 ID (名稱)。例如：openssoidp
備註：如果 Fedlet 服務提供者或識別提供者實體 ID 包含百分號 (%) 或逗號 (,)，則必須先對這些字元進行換碼，然後才能在 fedlet.cot 檔案中替代它。例如，將「%」變更爲「%25」，將「,」變更爲「%2C」。	

7 **將 FedletConfiguration.properties 檔案從 Java Fedlet java/conf 目錄複製到 Fedlet 主目錄。**

8 **將識別提供者標準中介資料 XML 檔案 (在步驟 1 中產生) 複製到 Fedlet 主目錄。此檔案必須命名為 idp.xml。**

9 **將 Java Fedlet XML 中介資料檔案 (`sp.xml`) 匯入識別提供者。**

對於 Oracle OpenSSO 8.0 Update 2，請在 OpenSSO 8.0 管理主控台中，使用 [共用作業] 下的 [註冊遠端服務提供者] 工作流程匯入 Java Fedlet 服務提供者中介資料，並將 Java Fedlet 服務提供者增加到信任圈中。

接下來的步驟 根據您的需求，繼續對 Java Fedlet 進行其他配置。

▼ 配置 .NET Fedlet 的步驟：

1 **在識別提供者一端，為識別提供者產生 XML 中介資料，並將該中介資料儲存到名為 `idp.xml` 的檔案中。**

對於 Oracle OpenSSO 8.0 Update 2，請使用 `exportmetadata.jsp`。例如：

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

2 **在服務提供者一端，解壓縮 Fedlet ZIP 檔案 (如有必要)。**

3 **將以下檔案從 .NET Fedlet `asp.net/conf` 資料夾複製到應用程式的 `App_Data` 資料夾中：**

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

4 **在 `App_Data` 資料夾中，重新命名您所複製的檔案，並將 `-template` 從每個名稱中刪除。**

5 **在您複製到 `App_Data` 資料夾並進行重新命名的檔案中，替代下表中所示的標記：**

標記	替代成
`FEDLET_COT`	遠端識別提供者和 .NET Fedlet 服務提供者應用程式所屬的信任圈 (COT) 的名稱。
`FEDLET_ENTITY_ID`	.NET Fedlet 服務提供者應用程式的 ID (名稱)。例如：`fedletsp`
`FEDLET_DEPLOY_URI`	.NET Fedlet 服務提供者應用程式的 URL。例如：`http://fedletsp.example.com/myFedletApp`
`IDP_ENTITY_ID`	遠端識別提供者的 ID (名稱)。例如：`openssoidp`

6 **將識別提供者標準中介資料 XML 檔案 (在步驟 1 中產生) 複製到應用程式的 `App_Data` 資料夾中。此檔案必須命名為 `idp.xml`。**

7 **將 `Fedlet.dll` 和 `Fedlet.dll.config` 檔案從 .NET Fedlet `asp.net/bin` 資料夾複製到應用程式的 `bin` 資料夾中。**

8 **將 .NET Fedlet XML 中介資料檔案 (`sp.xml`) 匯入識別提供者。**

對於 Oracle OpenSSO 8.0 Update 2，請在 OpenSSO 8.0 管理主控台中，使用 [共用作業] 下的 [註冊遠端服務提供者] 工作流程匯入 .NET Fedlet 服務提供者中介資料，並將 .NET Fedlet 服務提供者增加到信任圈中。

接下來的步驟 根據您的需求，繼續對 .NET Fedlet 進行其他配置。

OpenSSO 8.0 Update 2 中 Fedlet 的新功能

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

Fedlet 版本資訊 (CR 6941387)

Oracle OpenSSO Fedlet 包括版本資訊。擷取 Fedlet 套裝軟體 (ZIP 檔案) 中的檔案之後，檢視下列檔案之一可確定 Fedlet 的版本：

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

Java Fedlet 密碼加密和解密 (CR 6930477)

Java Fedlet 在 `fedlet.war` 檔案中提供了 `fedletEncode.jsp`，以加密 `storepass` 和 `keypass` 密碼。依預設，將爲各 Fedlet 產生不同的加密金鑰。若要變更此加密金鑰，請設定 Fedlet `FederationConfig.properties` 檔案中的 `am.encryption.pwd` 特性。

Java Fedlet 簽名和加密支援

Java Fedlet 支援 XML 簽名驗證和解密已加密的 `assertion` 和 `NameID` 元素及其相應屬性。

▼ 配置 Java Fedlet 以支援簽名和加密的步驟：

1. 使用 **keytool** 公用程式建立名為 **keystore.jks** 的金鑰庫檔案。
2. 將用於簽名的私密金鑰 (及公開憑證 [如果適用]) 和用於加密的私密金鑰 (及公開憑證 [如果適用]) 增加到 **keystore.jks** 檔案中。
3. 建立 **.storepass** 檔案。
4. 將密碼增加到 **.storepass** 檔案。若要加密密碼，請使用 **fedletEncode.jsp**。
5. 建立 **.keypass** 檔案。
6. 將密碼增加到 **.keypass** 檔案。若要加密密碼，請使用 **fedletEncode.jsp**。
7. 如果要使用純文字密碼，請在 **FederationConfig.properties** 檔案中註釋以下行：

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

8. 在 **FederationConfig.properties** 檔案中設定以下屬性的完整路徑，此處的 *path* 為相應檔案的完整路徑：

   ```
   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** 匯出簽名憑證。例如：

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

   該工具會提示您輸入用於存取 `keystore.jks` 的密碼，然後產生憑證。
10. 如果您需要加密憑證，請使用 **keytool** 匯出該憑證，如上一步驟中所述。(也可以將同一憑證用於簽名和加密。)
11. 建立一個 **KeyDescriptor** XML 區塊，並將簽名憑證增加到該區塊中。範例如下，請注意 **KeyDescriptor** 元素的 **use="signing"** 標記：

    ```
    <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 **建立另一個 KeyDescriptor XML 區塊，並將加密憑證增加到該區塊中。範例如下，請注意 KeyDescriptor 元素的 use="encryption" 標記：**

```
<KeyDescriptor use="encryption">
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <X509Data>
          <X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDBnzANBgkqhkiG9w0B
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 檔案中，將包含簽名憑證和加密憑證的 XML 區塊增加到 SPSSODescriptor 元素下面。如需範例 SPSSODescriptor 元素，請參見範例 4–1。**

將 AuthnRequestsSigned 屬性設定爲 true，以將 Java Fedlet 配置爲簽署所有認證請求。

14 **在 Java Fedlet sp-extended.xml 檔案中，設定以下元素的值：**

- signingCertAlias 包含金鑰庫中的 XML 簽名憑證的別名。
- encryptionCertAlias 包含金鑰庫中的 XML 加密憑證的別名。

15 **若要強制 Java Fedlet 服務提供者加密的內容，請在 sp-extended.xml 檔案中將以下屬性設定為 true：**

- wantAssertionEncrypted
- wantNameIDEncrypted
- wantAttributeEncrypted

16 **若要強制 Java Fedlet 服務提供者簽署的內容及計劃簽署的內容，請將以下屬性設定為 true：**

- idp.xml 檔案中的 wantAuthnRequestsSigned 告訴 Fedlet 要簽署的內容。

- sp.xml 檔案中的 AuthnRequestsSigned 和 WantAssertionsSigned 告訴識別提供者 Fedlet 計劃簽署的內容。
- sp-extended.xml 檔案中的 wantArtifactResponseSigned 告訴 Fedlet 要簽署的內容。
- sp-extended.xml 檔案中的 wantPOSTResponseSigned
- sp-extended.xml 檔案中的 wantLogoutRequestSigned
- sp-extended.xml 檔案中的 wantLogoutResponseSigned

如果識別提供者需要簽署特定訊息，請將 idp-extended.xml 檔案中的相應屬性設定爲 true。例如，wantLogoutRequestSigned 和 wantLogoutResponseSigned。

備註 – 如果在 sp-extended.xml 檔案中設定了屬性，請將此資訊告知識別提供者管理員，以便可以在識別提供者中進行必要的配置變更。

---

17 重新啓動 Java Fedlet Web 容器。

18 將 Java Fedlet sp.xml 檔案匯入識別提供者。

範例 4–1 Java Fedlet 範例 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
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDBnzANBgkqhkiG9w0B
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 屬性查詢支援 (CR 6930476)

Java Fedlet 支援 SAMLv2 屬性查詢，以針對特定的身份屬性值查詢諸如 Oracle OpenSSO 8.0 Update 2 等識別提供者。您可以配置 Fedlet 以簽署查詢和加密查詢。簽名對於發出 Fedlet 查詢是必需的，而加密是可選的。

▼ 配置 Java Fedlet 以支援屬性查詢的步驟：

1. 啓用 XML 簽名以簽署屬性查詢，如第 43 頁的 「Java Fedlet 簽名和加密支援」中所述。

2. 將於先前步驟中產生的憑證增加到 Fedlet `sp.xml` 檔案中的 `RoleDescriptor` 元素中。在以下範例中，有兩個供您在其中貼上憑證的 `KeyDescriptor` 標記。一個用於簽名，另一個用於加密。如果不啓用加密，則不需要 `KeyDescriptor use="encryption"` 標記。

```
<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` 檔案中，指定 `signingCertAlias` 屬性的值，並指定 `encryptionCertAlias` 屬性 (如果已配置) 的值。**

如果您計劃配置識別提供者以加密指定，也請加密 `NameID` 元素。因此，必須將 `wantNameIDEncrypted` 屬性的值設定爲 `true`。將 XML 代碼增加到 `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>
```

在此範例中，`test` 是範例金鑰的別名。

4 **將 Java Fedlet 中介資料檔案 (`sp.xml`) 匯入識別提供者。**

此外，在識別提供者中執行其他配置步驟，以支援 Fedlet 屬性查詢。

請求和回應的 .NET Fedlet 加密和解密 (CR 6939005)

.NET Fedlet 可以加密外寄 XML 請求和解密針對 NameID、Attribute 和 Assertion 元素的內送回應。

▼ 配置 .NET Fedlet 以加密和解密請求和回應的步驟：

1 **使用 Microsoft Management Console 的憑證嵌入式管理單元將您的 X.509 憑證匯入本機電腦帳號內的個人資料夾中。若要使用此嵌入式管理單元，請參閱以下 Microsoft 文章：**

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

2 **透過檢視 [特性] 對話方塊並輸入值，為此憑證指定一個友好的名稱。(儲存此值以在步驟 4 中使用。)**

3. 為 Internet Information Server (IIS) 使用的使用者帳號設定相應的權限，以允許其讀取該憑證，如 Microsoft 文章中所述。例如：

   a. 在憑證嵌入式管理單元中，瀏覽至 [Action] (動作) > [All Tasks] (所有作業) > [Manage Private Keys] (管理私密金鑰)。

   b. 為執行 IIS 的使用者帳號 (通常為 NETWORK SERVICE) 指定「允許讀取」權限。

4. 在 .NET Fedlet 的擴展中介資料檔案 (`sp-extended.xml`) 中，指定在步驟 2 中指定的友好名稱作為 `encryptionCertAlias` 屬性的值。例如：

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

5. 在 .NET Fedlet 的服務提供者中介資料檔案 (`sp.xml`) 中，增加用於加密金鑰的 `KeyDescriptor`。

   使用先前使用過的 Microsoft Management Console 憑證嵌入式管理單元匯出要包括在 `KeyDescriptor` XML 區塊中的 Base64 編碼的憑證公開金鑰。此 `KeyDescriptor` 必須爲 `SPSSODescriptor` 內的第一個子元素。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. 重新啓動與您的 .NET 應用程式關聯的應用程式集區。

接下來的步驟 若要測試此配置，請使用範例應用程式。此外，請設定以下屬性以加密請求和解密來自包含所配置中介資料的適當變更的識別提供者的回應：

- Assertion：將 `sp-extended.xml` 中介資料檔案中的 `wantAssertionEncrypted` 屬性設定爲 `true`，以讓 .NET Fedlet 解密來自識別提供者的內送回應中的 `EncryptedAssertion` 元素。

- Attribute：將 sp-extended.xml 中介資料檔案中的 wantAssertionEncrypted 屬性設定爲 true，以讓 .NET Fedlet 解密來自識別提供者的內送回應中的 EncryptedAssertion 元素。
- NameID：將idp-extended.xml 中介資料檔案中的 wantNameIDEncrypted 屬性設定爲 true，以讓 .NET Fedlet 加密外寄請求中的 NameID 元素。在 sp-extended.xml 中設定這一屬性，以讓 .NET Fedlet 解密來自識別提供者的內送回應中的 EncryptedID 元素。

請求和回應的 .NET Fedlet 簽名 (CR 6928530)

.NET Fedlet 支援簽署外寄 XML 請求 (例如 Authn 請求) 和登出請求。

▼ 配置 .NET Fedlet 以簽署請求和回應的步驟：

1. **使用 Microsoft Management Console 的憑證嵌入式管理單元將您的 X.509 憑證匯入本機電腦帳號內的個人資料夾中。若要使用此嵌入式管理單元，請參閱以下 Microsoft 文章：**

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

2. **透過檢視 [特性] 對話方塊並輸入值，為此憑證指定一個友好的名稱。(儲存此值以在步驟 4 中使用。)**

3. **為 Internet Information Server (IIS) 使用的使用者帳號設定相應的權限，以允許其讀取該憑證，如 Microsoft 文章中所述。例如：**

   a. **在憑證嵌入式管理單元中，瀏覽至 [Action] (動作) > [All Tasks] (所有作業) > [Manage Private Keys] (管理私密金鑰)。**

   b. **為執行 IIS 的使用者帳號 (通常為 NETWORK SERVICE) 指定「允許讀取」權限。**

4. **在 .NET Fedlet 的擴展中介資料檔案 (sp-extended.xml) 中，指定在步驟 2 中指定的友好名稱作為 signingCertAlias 屬性的值。例如：**

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

5. **在 .NET Fedlet 的服務提供者中介資料檔案 (sp.xml) 中，增加用於簽署金鑰的 KeyDescriptor。**

   使用先前使用過的 Microsoft Management Console 憑證嵌入式管理單元匯出要包括在 KeyDescriptor XML 區塊中的 Base64 編碼的憑證公開金鑰。此 KeyDescriptor 必須爲 SPSSODescriptor 內的第一個子元素。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 **重新啓動與您的 .NET 應用程式關聯的應用程式集區。**

.NET Fedlet 單次登出 (CR 6928528 和 CR 6930472)

.NET Fedlet 支援識別提供者啓動的單次登出和服務提供者啓動的單次登出。爲實作單次登出，.NET Fedlet 範例應用程式將 `logout.aspx` 和 `spinitiatedslo.aspx` 檔案包括在 `asp.net/SampleApp` 資料夾中。若要瞭解 Fedlet 單次登出功能如何工作，請部署 .NET Fedlet 範例應用程式。

▼ 配置 .NET Fedlet 服務提供者應用程式單次登出的步驟：

1 **如果您尚未配置 .NET Fedlet，請依照 `Readme` 檔案中說明的步驟進行作業。**

2 **將 `logout.aspx` 和 `spinitiatedslo.aspx` 檔案複製到您的 .NET 應用程式的公共內容中。**

3 **對您的應用程式的配置檔案進行以下變更：**

- 在 `sp.xml` 檔案中，確保 `logout.aspx` 檔案的路徑指向您的應用程式的該檔案的正確位置。
- 在 `idp.xml` 檔案中 (或識別提供者的配置過程中)，確保 `spinitiatedslo.aspx` 檔案的路徑指向您的應用程式的該檔案的正確位置。

4 **如果您希望簽署登出請求和登出回應，請將 `sp-extended.xml` 和 `idp-extended.xml` 檔案中的以下屬性設定為 `true`：**

- `wantLogoutRequestSigned`
- `wantLogoutResponseSigned`

5 **將 Fedlet 服務提供者中介資料檔案 (`sp.xml`) 匯入服務提供者。**

此外，請通知識別提供者管理員您已爲 Fedlet 服務提供者配置了單次登出，以便可以對識別提供者配置進行所需的任何其他變更。

.NET Fedlet 服務提供者啓動的單次登入 (CR 6928525)

.NET Fedlet 支援 SAMLv2 服務提供者啓動的單次登入 (SSO)。此外，還必須支援小工具，以允許 .NET Fedlet 接收小工具，然後透過 SOAP 使用頒發識別提供者的小工具解析服務進行解析。

.NET Fedlet 範例應用程式顯示了配置單次登入的方式。爲應用程式安裝必要的小工具後，需要提供特定的 URI，以在識別提供者成功進行認證後接收包含 SAMLv2 回應的 HTTP POST。以下代碼範例顯示如何在 .NET 應用程式中擷取此資訊：

範例 4–2 在 .NET Fedlet 應用程式中擷取 `AuthnResponse` 的代碼範例

```
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)
}
```

如果應用程式收到 SAMLv2 回應，將使用指定資訊填寫 `authnResponse` 物件。範例應用程式顯示如何從該物件中擷取屬性和主體資訊。

.NET Fedlet 支援多個識別提供者和探索服務 (CR 6928524)

.NET Fedlet 支援多個識別提供者和識別提供者探索服務。

在某些部署中，您可能需要爲 .NET Fedlet 配置多個識別提供者 (如 Oracle OpenSSO 8.0 Update 2)。針對您要增加的每個附加識別提供者執行如下作業。

▼ 配置 .NET Fedlet 以支援多個識別提供者的步驟：

1. **取得附加識別提供者的 XML 中介資料檔案。**
2. **將附加識別提供者的中介資料檔案命名為 `idp`*n*`.xml`，此處的 *n* 是您正在增加的識別提供者。例如，將第二個識別提供者檔案命名為 `idp2.xml`，將第三個識別提供者檔案命名為 `idp3.xml`，依此類推。此程序使用 `idp2.xml` 作為檔案名稱。**

3. **將步驟 2 中的 `idp2.xml` 檔案複製到應用程式的 `App_Data` 資料夾中。**

4. **將這個新識別提供者增加到 .NET Fedlet 信任圈中。**

   將新識別提供者增加到現有信任圈的步驟：

   在應用程式的 `App_Data` 資料夾中的 `fedlet.cot` 檔案中，將新的 IDP 實體 ID (由 `idp2.xml` 中介資料檔案中的 `entityID` 屬性指示) 附加至 `sun-fm-trusted-providers` 屬性的值，使用逗號 (,) 分隔。

   將新識別提供者增加到新信任圈的步驟：

   a. **在應用程式的 `App_Data` 資料夾中建立名為 `fedlet2.cot` 的新檔案。使用現有 `fedlet.cot` 作為範本，但將 `cot-name` 屬性的值變更為新信任圈的名稱 (例如 `cot2`)。包括新識別提供者實體 ID 和 Fedlet 實體 ID 作為 `sun-fm-trusted-providers` 屬性的值，使用逗號 (,) 分隔這兩個實體 ID。**

   b. **在 `sp-extended.xml` 檔案中，將新信任圈的名稱增加到 `cotlist` 屬性的值中。例如，對於名為 `cot2` 的信任圈：**

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

5. **在應用程式的 `App_Data` 資料夾中，建立一個新的 `idp2-extended.xml` 檔案作為新識別提供者的擴展中介資料。使用現有 `idp-extended.xml` 檔案作為範本，但將 `entityID` 變更為新識別提供者的實體 ID。如果為該識別提供者建立了新信任圈，請將 `cotlist` 屬性的值變更為該信任圈的名稱。請確保該附加識別提供者是一個遠端識別提供者。**

6. **重新啓動與您的 Fedlet .NET 應用程式關聯的應用程式集區。**

7. **必須將 Fedlet 中介資料 XML 檔案 (`sp.xml`) 匯入該附加識別提供者並增加到該識別提供者實體所屬的信任圈中。將 `sp.xml` 檔案匯入識別提供者，或將該檔案提供給您的識別提供者管理員進行匯入。**

.NET Fedlet 支援識別提供者探索服務 (CR 6928524)

在該分析藍本中，爲 .NET Fedlet 配置了同屬一個信任圈的多個識別提供者，而您希望將 Fedlet 配置爲使用識別提供者探索服務來確定偏好的識別提供者。

必須爲用於 .NET Fedlet 的識別提供者配置該探索服務。如需有關在 Oracle OpenSSO 8.0 Update 2 中配置識別提供者探索服務的資訊，請參見以下文件集合：`http://docs.sun.com/coll/1767.1`。

▼ 將 .NET Fedlet 配置為使用識別提供者探索服務的步驟：

1. **在 .NET Fedlet `fedlet.cot` 檔案中，將 `sun-fm-saml2-readerservice-url` 特性設定為 SAMLv2 讀取器服務 URL。例如：**

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

2. **重新啓動與您的 .NET Fedlet 應用程式關聯的應用程式集區。**

關於 Oracle OpenSSO Fedlet 的一般問題和解決方法

待編寫

文件勘誤表

Fedlet Java API 參照在 Oracle OpenSSO 8.0 Update 2 Java API 參照中提供，Oracle OpenSSO 8.0 Update 2 Java API 參照包含在以下文件集合中：`http://docs.sun.com/coll/1767.1`

備註 – 在 OpenSSO 8.0 Update 2 發行版本中，不支援 `getPolicyDecisionForFedlet` 方法。

◆ ◆ ◆ 第 5 章

5 整合 OpenSSO 8.0 Update 2 與 Oracle Access Manager

本章介紹如何使用 OpenSSO 8.0 Update 2 和 Oracle Access Manager 10g 或 11g 實作單次登入。該資訊是「Sun OpenSSO Enterprise 8.0 Integration Guide」中的第 3 章「Integrating Oracle Access Manager」 中包含的概念資訊的補充。該使用案例透過採用 Oracle Access Manager 階段作業提供受 OpenSSO 保護的應用程式的單次登入體驗。配置的 OpenSSO 認證模組會根據該 Oracle Access Manager 階段作業產生 OpenSSO 階段作業。

整合步驟概述

1. 第 55 頁的 「開始之前」
2. Unpacking the Integration Bits
3. Building source files for Oracle Access Manager in OpenSSO
4. 第 58 頁的 「(可選) 在 Oracle Access Manager 中爲 OpenSSO 建立認證方案」
5. 第 59 頁的 「使用 Oracle Access Manager 和 Oracle OpenSSO STS 配置單次登入」
6. 第 61 頁的 「測試單次登入的步驟：」
7. 第 61 頁的 「(可選) 將 Oblix 認證方案安裝到 Oracle Access Manager 中」

開始之前

在您嘗試安裝 OpenSSO 8.0 Update 2 以與 Oracle Access Manager 整合之前，請確保您具有以下元件的存取權限：

opensso.zip	該 zip 檔案包含安裝和配置 OpenSSO 8.0 Update 2 所需的 opensso.war 檔案、整合原始碼、配置檔案及其他工具。
OpenSSO 代理程式	當受 OpenSSO 保護的應用程式確實可以使用 Oracle Access Manager 建立的認證階段作業時，將使用 OpenSSO 代理程式。

Oracle Access Manager 10g 或 11g	從 Oracle 網站下載 Oracle Access Manager。請參見 Oracle Fusion Middleware 11gR1 Software Downloads 頁面。
Oracle WebGate 10g 或 11g	爲 OpenSSO 和 Oracle Webgate 都支援的容器下載 Oracle Webgate。現在，Sun Web Server 7.x 是唯一一個這兩種產品都支援的容器。請參見 Oracle Fusion Middleware 11gR1 Software Downloads 頁面。
Oracle Access Manager SDK 10g 或 11g	下載 Oracle Access Manager。編譯和建立用於 Oracle Access Manager 整合的 OpenSSO 認證模組時，需要使用 SDK。 請參見 Oracle Fusion Middleware 11gR1 Software Downloads 頁面。
OpenSSO C-SDK 2.2	(可選) 在 Oracle Access Manager 自身中建立認證模組以產生 OAM 階段作業時，需要使用 OpenSSO C-SDK。從 OpenSSO 角度來說，這可能不是一個常見的使用案例。請參閱「Sun OpenSSO Enterprise 8.0 C API Reference for Application and Web Policy Agent Developers」中的「Where is the C SDK?」

整合資料各部分解說

`opensso/integrations/oracle` 目錄中包含用於編譯和建立自訂認證模組及其他外掛程式的來源和配置。請參閱「Sun OpenSSO Enterprise 8.0 Integration Guide」中的第 3 章「Integrating Oracle Access Manager」，以獲得使用案例選項及相關資訊。下表彙總了 `opensso/integrations/oracle` 目錄下的檔案及各檔案的描述。

`README.html`	這是您現在正在閱讀的檔案。
`build.xml`	用於在 OpenSSO 中爲 Oracle Access Manager 建立自訂認證模組的 ant 建立檔案。
`config`	在 OpenSSO 中爲 Oracle Access Manager 建立認證模組時需要使用的配置檔案。 ■ `OblixAuthService.xml` Oracle Access Manager 認證模組的認證服務檔案 ■ `OblixAuthModule.xml` Oracle Access Manager 的認證模組回呼。

依預設，該檔案是一個空檔案，但爲執行配置必須提供該檔案。

- `OblixAuth.properties`

  儲存用於認證的國際化金鑰的特性檔案

`lib`

依預設此目錄爲空。此 `lib` 目錄中必須包含以下程式庫才能編譯來源程式庫。

- `jobaccess.jar`

  從 Oracle Access Manager SDK 中複製該檔案。

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

  從 opensso.war 中複製這些檔案

- `servlet.jar` **或** `javaee.jar`

  複製 GlassFish `lib` 目錄。理想情況下，任何包含標準 Java EE 類別 (例如 `javax.servlet.http.Cookie`) 的 JAR 檔案均可。

`source`

包含以下來源檔案的目錄：

- `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`

  此類別是 SAML 服務提供者的 SAML2 外掛程式介面。此類別使用 OpenSSO 階段作業服務對 Oracle Access Manager 進行遠端認證。

oamauth (可選)

此目錄包含用於 OpenSSO 的 Oblix 認證方案的來源檔案。這是一個 C 型認證模組，其利用 OpenSSO C-SDK 進行驗證。

- oam/solaris/authn_api.c

  此檔案爲 OpenSSO 實作 Oblix 自訂認證方案。

- oam/solaris/include/*.h

  編譯認證方案時需要使用的所有標頭檔案。

- oam/solaris/AMAgent.properties

  範例 OpenSSO 代理程式配置檔案。認證方案驗證 OpenSSO 階段作業時需要使用該檔案。

在 OpenSSO 中為 Oracle Access Manager 建立來源檔案

使用 ant 程序檔可建立來源檔案。必須在 PATH 中安裝和配置相容的 ant 程序檔。

▼ 為 Oracle Access Manager 建立來源檔案的步驟：

1. **執行以下指令：**

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

   該指令將在 `$openssozipdir/integrations/oracle/dist` 目錄中建立來源檔案並產生 `fam_oam_integration.jar`。

2. **將認證模組放入 OpenSSO WAR 檔案中。**

   a. **建立一個暫存目錄並解壓縮 `opensso.war`。範例：**

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

      從現在開始，`/export/tmp` 將作爲 WAR 臨時區域使用，並由巨集 `$WAR_DIR` 表示。

   b. **將 `$openssozipdir/integrations/oracle/dist/fam_oam_integration.jar` 複製到 `$WAR_DIR/WEB-INF/lib` 中。**

   c. **將 `$openssozipdir/integrations/oracle/config/OblixAuth.properties` 複製到 `$WAR_DIR/WEB-INF/classes` 中。**

   d. **將 `$openssozipdir/integrations/oracle/config/OblixAuthModule.xml` 複製到 `$WAR_DIR/config/auth/default` 以及 `$WAR_DIR/config/auth/default_en` 目錄中。**

   e. **使用 `$WAR_DIR` 中的 `jar cvf opensso.war` 重新壓縮 `opensso.war`。**

   範例待編寫

(可選) 在 Oracle Access Manager 中為 OpenSSO 建立認證方案

備註：這不是一個常見的使用案例。除非有必要 (例如在 SAML2 服務提供者使用案例中)，否則您不必建立此認證方案。

若要建立 Oblix 認證方案，您必須自訂 `makefile`。此外，由於這是一個 C 型認證模組，其具有作業系統依賴性。

▼ 在 Oracle Access Manager 中為 OpenSSO 建立認證方案的步驟：

開始之前 認證方案檔案位於 `$openssozipdir/integrations/oracle/oamauth/solaris` 目錄下。

1. **下載並配置 OpenSSO C-SDK 2.2 版。**
   `authn_api.c` 檔案包含對 `AMAgent.propeties` 檔案的參照。相應地修改該檔案。

2. **針對您的環境自訂 `makefile`。**
   例如，指定 `gcc` 編譯位置。此外，還請編輯 `LDFLAGS` 以指向您的 OpenSSO C-SDK `lib` 目錄。

3. **執行 `make` 指令。**
   執行 `make` 指令應產生一個 `authn_api.so` 檔案。

使用 Oracle Access Manager 和 Oracle OpenSSO STS 配置單次登入

▼ 使用 Oracle Access Manager 和 Oracle OpenSSO 8.0 Update 2 配置單次登入的步驟：

開始之前：必須已經安裝和配置 Sun Java System Web Server 7.x。請參見 Sun Java System Web Server Documentation Wiki 以獲得 Web 伺服器安裝說明。

1. **在 Sun Java System Web Server 7.x 上安裝 OpenSSO。**

2. **在受支援的容器上安裝 OpenSSO 策略代理程式並配置該代理程式以與 OpenSSO 配合使用。**
   請參閱「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」以獲得安裝說明。

3. **安裝和配置 Oracle Access Manager。**
   請參閱《Oracle Access Manager 安裝指南 10g (10.1.4.3)》

4. **透過 Oracle Access Manager 安裝和配置 Oracle Access Manager SDK。**
   請參閱《Oracle Access Manager 安裝指南 10g (10.1.4.3)》

5 **在安裝 OpenSSO 伺服器的相同 Web 容器中安裝 Oracle Webgate。(Sun Web Server 7.x)**

配置 OpenSSO，使其僅保護 OpenSSO Web 應用程式的 `deployURI/UI/*`。範例：`/opensso/UI/.../*`

如需 Oracle Access Manager 策略、資源及其他配置詳細資訊，請參閱 Oracle Access Manager 管理指南。 取消保護 OpenSSO Enteprise 中的任何其他 URL。這適用於簡單的單次登入整合分析藍本，但策略評估是以完整整合及其他部署依賴關係爲基礎的。

6 **在 OpenSSO 中配置認證模組。**

a. **存取 OpenSSO 主控台。**

瀏覽器會重新導向至 Oracle Access Manager 進行認證。認證成功後，OpenSSO 會顯示一個登入頁面。使用 OpenSSO 管理員使用者名稱和密碼登入。

b. **將 Oracle 認證模組服務 XML 檔案匯入 OpenSSO 配置。**

該認證模組服務可以透過指令行 `ssoadm` 公用程式以及以瀏覽器爲基礎的 ssoadm.jsp 載入。

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

d. **選擇 create-service 選項。**

e. **從 `$openssozipdir/integrations/oracle/config/OblixAuthService.xml` 複製並貼上 XML 檔案，然後按一下 [提交]。**

此作業會將認證模組服務載入到 OpenSSO 配置中。

f. **將認證模組註冊到認證核心服務中。**

核心服務中包含了一系列認證程式。在 `http://host:port/opensso/ssoadm.jsp` 中選擇 `register-auth-module` 選項。輸入 `com.sun.identity.authentication.oblix.OblixAuthModule` 作爲認證模組類別名稱。

g. **確定認證模組已註冊到預設範圍中。**

使用 URL http://host:port/opensso 存取 OpenSSO。在 OpenSSO 主控台中，按一下預設範圍，然後按一下 [認證] 標籤。按一下 [新增] 以建立一個名爲 OblixAuth 的新認證模組。

h. **在 [認證] 標籤上，選取該 OblixAuth 認證模組。**

配置 Oblix SDK 目錄。啓用 [僅檢查遠端使用者標頭]，並將遠端標頭名稱指定爲 `OAM_REMOTE_USER`。根據部署該參數可配置。

7 **(可選) 在 OpenSSO 核心認證服務中啓用 [忽略設定檔] 選項。**

在 OpenSSO 主控台中，前往 [配置] > [核心] > [範圍屬性] > [使用者設定檔]。選擇 [忽略]，然後按一下 [儲存]。

此配置會阻止 OpenSSO 在成功驗證後搜尋現有使用者設定檔。但是，如果 OpenSSO 和 Oracle Access Manager 所使用的使用者儲存庫完全相同，則無需執行此步驟。前往 [管理主控台] -> [配置] -> [核心] -> [範圍屬性] -> [使用者設定檔]。選擇 [忽略]，然後按一下 [儲存]。

8 **編輯 Web 伺服器啓動程序檔，以包含 Oracle Access Manager SDK 共用程式庫。**

更新 `startserv` 程序檔中的 `LD_LIBRARY_PATH`，以包含 `$ACCESSDKDIR/oblix/lib` 中的共用程式庫。

9 **重新啓動包含 OpenSSO 和 Oracle Webgate 的 Sun Web 伺服器。**

10 **將 [Web 代理程式的登入 URL] 的值更新為**
`http://openssohost:openssoport/deployURI/UI/Login?module=OblixAuth`。

測試單次登入的步驟：

存取受 OpenSSO 保護的應用程式中的受保護資源。如果您尚未通過認證，瀏覽器會將您重新導向至 Oracle Access Manager 登入頁面。成功登入後，該應用程式會建立一個 OpenSSO 階段作業，並最終重新導向回受策略代理程式保護的應用程式 URL。您對受保護的應用程式的存取將被允許或拒絕，具體視該策略而定。

(可選) 將 Oblix 認證方案安裝到 Oracle Access Manager 中

如果在驗證 OpenSSO 階段作業時必須產生 Oracle Access Manager 階段作業，此作業將非常有用。請參閱「Sun OpenSSO Enterprise 8.0 Integration Guide」中的第 3 章「Integrating Oracle Access Manager」，以獲得有關相關使用案例的資訊。

Oblix 認證方案顯示爲 C 認證模組，此認證方案使用 OpenSSO C-SDK 2.2 版來驗證 OpenSSO 階段作業。Oblix 中的 OpenSSO 認證方案使用 `AMAgent.properties` 中的 OpenSSO 用戶端一端的配置。在配置認證模組之前， 必須先自訂該檔案。版本說明 指定了該檔案的位置。在配置認證方案之前，必須先將編譯後的 `authn_api.so` 及其他 C-SDK 程式庫複製到 `$OAM_INSTALL_DIR/access/oblix/lib` 目錄中。《「Sun OpenSSO 8.0 整合指南」》中提供了一個範例螢幕截圖，以說明 Oracle 認證方案的配置方式，但這只應作爲參考。如需更多詳細資訊，請參閱最新的 Oracle Access Manager 文件。

整合 OpenSSO 8.0 Update 2 與 Oracle Access Manager

本節介紹如何使用 OpenSSO 8.0 Update 2 和 Oracle Access Manager 10.1.4.0.1 和 11g 版來實作單次登入。該資訊是「Sun OpenSSO Enterprise 8.0 Integration Guide」中的第 3 章「Integrating Oracle Access Manager」 中包含的概念資訊的補充。該使用案例透過採用 Oracle Access Manager 階段作業提供受 OpenSSO 保護的應用程式的單次登入體驗。配置的 OpenSSO 認證模組會根據該 Oracle Access Manager 階段作業產生 OpenSSO 階段作業。