Oracle® OpenSSO Update 2-Versionsinformationen

Beta

Teilenr.: 821–2835
Juli 2010

101013@24999

Inhalt

Vorwort

Die Oracle OpenSSO 8.0 Update 2-Versionsinformationen enthalten Informationen zum Herunterladen und Installieren der OpenSSO Update 2-Software. Dieses Dokument enthält sich Informationen zu Änderungen in der Software, die seit der Veröffentlichung von OpenSSO Update 1 stattgefunden haben.

Zielgruppe dieses Buchs

Diese Versionsinformationen richten sich an Unternehmensadministratoren und Entwickler, die Oracle SSO 8.0 bereits installiert und bereitgestellt haben. Sie müssen mit den Konzepten und Verfahren vertraut sein, die in der Dokumentation des Kernprodukts beschrieben sind.

Verwandte Bücher

Diese Versionsinformationen ergänzen die Kerndokumentation von Oracle OpenSSO 8.0 unter folgender URL: http://docs.sun.com/app/docs/coll/1767.1.

Verweise auf Drittanbieter-Websites

In dieser Dokumentation wird auf URLs von Drittanbietern verwiesen, über die zusätzliche relevante Informationen zur Verfügung gestellt werden.

Hinweis – Oracle haftet nicht für die Verfügbarkeit der Websites Dritter, die in diesem Dokument erwähnt werden. Oracle unterstützt keine Inhalte, Werbung, Produkte oder sonstige Materialien, die auf oder über solche Websites oder Ressourcen verfügbar sind, und übernimmt keine Verantwortung oder Haftung dafür. Oracle übernimmt keine Verantwortung oder Haftung für tatsächliche oder angebliche Schäden oder Verluste, die durch den Gebrauch von oder in Verbindung mit derartigen Inhalten, Gütern oder Diensten entstanden sind, die auf diesen oder durch diese Websites oder Ressourcen verfügbar sind.

Dokumentation, Support und Schulung

Auf den folgenden Websites finden Sie zusätzliche Ressourcen:

- Dokumentation (`http://docs.sun.com`)
- Support (`http://www.oracle.com/us/support/systems/index.html`)
- Training (`http://education.oracle.com`) – Klicken Sie auf den Sun-Link in der linken Navigationsleiste.

Sagen Sie uns Ihre Meinung!

Oracle freut sich über Ihre Anmerkungen und Vorschläge zur Qualität und Nützlichkeit der Dokumentation. Falls Sie Fehler finden oder andere Verbesserungsvorschläge haben, besuchen Sie `http://docs.sun.com` und klicken auf Feedback. Geben Sie den Titel und die Teilenummer der Dokumentation mit dem Kapitel, dem Abschnitt und der Seitennummer an, wenn sie verfügbar sind. Wenn Sie eine Antwort wünschen, teilen Sie dies bitte mit.

Oracle Technology Network (`http://www.oracle.com/technetwork/index.html`) bietet eine Reihe von Ressourcen zur Oracle-Software:

- Diskutieren Sie technische Probleme und Lösungen im Diskussionsforum (`http://forums.oracle.com`).
- Rufen Sie die praktischen, schrittweisen Tutorials ab über Oracle By Example (`http://www.oracle.com/technology/obe/start/index.html`).
- Laden Sie denMustercoder (`http://www.oracle.com/technology/sample_code/index.html`) herunter.

Typografische Konventionen

In der folgenden Tabelle sind die in diesem Handbuch verwendeten typografischen Konventionen aufgeführt.

TABELLE P–1 Typografische Konventionen

Schriftart	Bedeutung	Beispiel
`AaBbCc123`	Die Namen von Befehlen, Dateien, Verzeichnissen sowie Bildschirmausgabe.	Bearbeiten Sie Ihre `.login`-Datei. Verwenden Sie `ls -a`, um eine Liste aller Dateien zu erhalten. `system% Sie haben eine neue Nachricht.`
`AaBbCc123`	Von Ihnen eingegebene Zeichen (im Gegensatz zu auf dem Bildschirm angezeigten Zeichen)	`system%` **`su`** `Password:`

TABELLE P–1 Typografische Konventionen *(Fortsetzung)*

Schriftart	Bedeutung	Beispiel
aabbcc123	Platzhalter: durch einen tatsächlichen Namen oder Wert zu ersetzen	Geben Sie zum Löschen einer Datei den Befehl `rm` *Dateiname* ein.
AaBbCc123	Buchtitel, neue Ausdrücke; hervorgehobene Begriffe	Lesen Sie hierzu Kapitel 6 im *Benutzerhandbuch*.
		Ein *Cache* ist eine lokal gespeicherte Kopie.
		Diese Datei *nicht* speichern.
		Hinweis: Einige hervorgehobene Begriffe werden online fett dargestellt.

Shell-Eingabeaufforderungen in Befehlsbeispielen

Die folgende Tabelle zeigt die standardmäßigeUNIX-System- und Superuser-Eingabeaufforderungen für Shells, die im Oracle Solaris Betriebssystem enthalten sind. Beachten Sie, dass die standardmäßige System-Eingabeaufforderung, die in den Befehlsbeispielen angezeigt werden, abhängig von der jeweiligen Oracle Solaris-Version variieren.

TABELLE P–2 Shell-Eingabeaufforderungen

Shell	Eingabeaufforderung
Bash-Shell, Korn-Shell und Bourne-Shell	`$`
Bash-Shell, Korn-Shell und Bourne-Shell für Superuser	`#`
C-Shell	`system%`
C-Shell für Superuser	`system#`

◆ ◆ ◆ KAPITEL 1

1 Info zu OpenSSO 8.0 Update 2

In diesem Kapitel werden die folgenden Themen behandelt.

Neues in OpenSSO 8.0 Update 2

OpenSSO 8.0 Update 2 bietet Verbesserungen im Sicherheitstoken-Dienst und dem OpenSSO Fedlet.

Verbesserungen im Sicherheitstoken-Dienst

Der Sicherheitstoken-Dienst enthält jetzt die folgenden neuen Funktionen:

- Unterstützt TokenType zum Generieren eine spezifischen Sicherheitstokens eines Webdienstanbieters.
- Unterstützt eine asymmetrische und eine Transportverknüpfung für X509 und Benutzername-Sicherheitstokens als Anforderer.
- Setzt SSL/Transport-Verknüpfung mit einem Benutzername-Sicherheitstoken um, wenn OpenSSO STS mit einem Benutzernamen über SSL konfiguriert ist.
- Gibt SAML-Schlüsselinhaber-Sicherheitstoken für asymmetrischen KeyType mit useKey als öffentlichen Schlüssel des Webdienst-Clients und dem Sicherheitstoken X509 des Webdienst-Clients aus.
- WSDL wird dynamisch auf Basis der Sicherheitstoken-Konfiguration aktualisiert.
- Unterstützt Verschlüsselung durch den öffentlichen Schlüssel des Webdienstanbieters.

- Verschlüsselt das statische Benutzernamekennwort, bevor es im Konfigurationsspeicher gespeichert wird.
- Unterstützt UserName-Token anstelle eines Sicherheitstokens über eine WS-Trust-Anforderung.
- Unterstützt Ausgabe von SAML-Übermittlungstokens.
- Das neue Webdienst-Sicherheitsauthentifizierungsmodul WSSAuth unterstützt die Verarbeitung der Kennwortvalidierung.
- Das neue OAMAuth-Authentifizierungsmodul ermöglicht mithilfe des Oracle Access Manager mit OpenSSO das einmalige Anmelden.

Weitere Informationen finden Sie in Kapitel 3, „Verwenden des Sicherheitstoken-Diensts“.

Fedlet-Verbesserungen

Das Fedlet umfasst jetzt die folgenden neuen Funktionen:

- Unterstützt Verschlüsselung im .NET-Fedlet.
- Unterstützt Anmeldung im .NET-Fedlet.
- .NET-Fedlet unterstützt jetzt die einmalige Abmeldung.
- .NET-Fedlet bietet vom Dienstanbieter initiierte einmalige Anmeldung und Artefakt-Unterstützung.
- Unterstützt mehrere Identity-Anbieter und Erkennung von Identity-Anbietern im .NET-Fedlet.
- Bietet Versionsinformationen in Eigenschafts- und Konfigurationsdateien für das Fedlet.
- Neue Kennwort-SPI-Implementierung.
- Unterstützt Attributsabfrage.
- Unterstützt einmalige Abmeldung.

Weitere Informationen finden Sie in Kapitel 4, „Arbeiten mit dem Oracle OpenSSO Fedlet“.

Hardware- und Software-Anforderungen für OpenSSO 8.0 Update 2

Siehe „Hardware and Software Requirements For OpenSSO Enterprise 8.0 Update 1“ in *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*

Unterstützung für neue Webcontainer

OpenSSO 8.0 Update 2 unterstützt die Webcontainer, die in den „Support for New Web Containers“ in *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes* beschrieben werden und die folgenden neuen Webcontainer:

- Oracle WebLogic Server 10g Version 3 (10.3)

Probleme und Problemumgehungen in OpenSSO 8.0 Update 2

CR 6959610: Muster von OpenSSO 8.0 Update 2 sollten in einer Produktionsumgebung entfernt werden.

Die Muster von OpenSSO 8.0 Update 2 können potenzielle Sicherheitsprobleme verursachen.

Problemumgehung Wenn Sie OpenSSO 8.0 Update 2 in einer Produktionsumgebung bereitstellen, entfernen Sie die Muster, um potenzielle Sicherheitsprobleme zu vermeiden.

CR 6964648: Für WebLogic Server 10.3.3 werden neue Java-Sicherheitsgenehmigungen benötigt.

Wenn Sie OpenSSO 8.0 Update 2 auf Oracle WebLogic Server 10.3.3 bei aktivierem Sicherheitsmanager bereitstellen, ist eine zusätzliche Java-Sicherheitsgenehmigung erforderlich.

Problemumgehung Fügen Sie die folgende Genehmigung in die WebLogic Server 10.3.3-Datei `weblogic.policy` ein:

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

CR 6939443: Zertifikatauthentifizierung mit LDAP-Prüfung oder OCSP-Prüfung schlägt auf WebLogic Server 10.3.x fehl.

Aufgrund eines Problems in früheren Version von Oracle WebLogic Server, z. B. 10.3.0 und 10.3.1, schlägt die Zertifikatauthentifizierung mit LDAP-Prüfung oder OSCP-Prüfung fehl.

Problemumgehung Dieses Problem wurde in WebLogic Server 10.3.3 behoben. Für die Verwendung der Zertifikatauthentifizierung mit LDAP-Prüfung oder OSCP-Prüfung verwenden Sie OpenSSO Update 2 mit WebLogic Server 10.3.3.

CR 6967026: Konfigurator kann über GlassFish 2.1.x keine Verbindung zur LDAPS-aktivierten Directory-Server-Instanz herstellen.

Wenn GlassFish Enterprise Server v2.1.1 oder v2.1.2 als OpenSSO 8.0 Update 2-Webcontainer bereitgestellt wird, kann der Konfigurator keine Verbindung zu einer LDAPS-aktivierten Directory-Server-Instanz herstellen.

Problemumgehung Für die Verwendung eines LDAPS-aktivierten Directory-Servers mit GlassFish als Webcontainer stellen Sie GlassFish Enterprise Server v2.1 bereit.

CR 6948937: Das Aktivieren von OpenSSO 8.0 Update 2 in WebLogic Server 10.3.3-Administrationskonsolen verursacht Ausnahmen.

Wenn Sie OpenSSO 8.0 Update 2 (`opensso.war`) in der WebLogic Server 10.3.3-Administrationskonsole bereitstellen und auf Start klicken, um zuzulassen, dass OpenSSO 8.0 Update 2 Anforderungen erhalten kann, werden in der Konsole, in der die WebLogic Server-Domäne gestartet wurde, Ausnahmen ausgelöst.

Hinweis: Wenn Sie OpenSSO 8.0 Update 2 gestartet haben, bleibt es weiterhin gestartet und werden keine neuen Ausnahmen ausgelöst, bis OpenSSO 8.0 Update 2 angehalten und anschließend neu gestartet wird.

Problemumgehung Kopieren Sie die Datei `saaj-impl.jar` aus der OpenSSO 8 Update 2-Datei `opensso-client-jdk15.war` in das WebLogic Server 10.3.3-Konfigurationsverzeichnis `endorsed`. Gehen Sie dazu folgendermaßen vor:

1. Halten Sie die Oracle WebLogic Server 10.3.3-Domäne an.
2. Dekomprimieren Sie bei Bedarf die OpenSSO 8.0 Update 2-Datei `opensso.zip`.
3. Erstellen Sie ein temporäres Verzeichnis und dekomprimieren Sie die Datei *zip-root*`/opensso/samples/opensso-client.zip` in diesem Verzeichnis, wobei *zip-root* den Pfad angibt, in dem Sie die Datei `opensso.zip` dekomprimiert haben. Beispiel:

   ```
   cd zip-root/opensso/samples
   mkdir ziptmp
   cd ziptmp
   unzip ../opensso-client.zip
   ```

4. Erstellen Sie ein temporäres Verzeichnis und extrahieren Sie die Datei `saaj-impl.jar` aus `opensso-client-jdk15.war`. Beispiel:

   ```
   cd zip-root/opensso/samples/ziptmp/war
   mkdir wartmp
   cd wartmp
   jar xvf ../opensso-client-jdk15.war WEB-INF/lib/saaj-impl.jar
   ```

5. Erstellen Sie ein neues Verzeichnis mit dem Namen `endorsed` im Verzeichnis `WEBLOGIC_JAVA_HOME/jre/lib` (wenn `endorsed` nicht bereits vorhanden ist), wobei `WEBLOGIC_JAVA_HOME` das JDK ist, zu dessen Verwendung WebLogic Server konfiguriert ist.
6. Kopieren Sie die Datei `saaj-impl.jar` in das Verzeichnis `WEBLOGIC_JAVA_HOME/jre/lib/endorsed`.
7. Starten Sie die WebLogic Server-Domäne.

CR 6959373: Webcontainer muss nach dem Ausführen des Skripts `updateschema` neu gestartet werden.

Wenn Sie den Skript `updateschema.sh` oder `updateschema.bat` ausgeführt haben, müssen Sie den OpenSSO 8.0 Update 2-Webcontainer neu starten.

CR 6961419: Das Ausführen des Skripts `updateschema.bat` erfordert eine Kennwortdatei.

Der Skript `updateschema.bat` führt mehrere `ssoadm`-Befehle aus. Daher müssen Sie vor dem Ausführen von `updateschema.bat` auf Windows-Systemen eine Kennwortdatei erstellen, die den Kennwortbenutzer in Klartext für den Benutzer `amadmin` enthält. Der Skript `updateschema.bat` fordert Sie auf, den Pfad zur Kennwortdatei einzugeben. Bevor der Skript beendet wird, entfernt er die Kennwortdatei.

OpenSSO 8.0 Update 2-Dokumentation

Als Ergänzung zu diesem Dokument ist weitere OpenSSO 8.0-Dokumentation in der folgenden Sammlung verfügbar:

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

Probleme mit der Dokumentation

OpenSSO 8.0 Update 2 enthält die folgenden Probleme mit der Dokumentation:

CR 6958580: Online-Hilfedokumente der Konsole unterstützten keine Discovery Agents.

Die Online-Hilfe der Administrationskonsole von OpenSSO 8.0 Update 2 dokumentiert Discovery Agents, auch wenn diese Agenten nicht unterstützt werden.

Problemumgehung Keine. Ignorieren Sie die Informationen über Discovery Agents in der Online-Hilfe.

CR 6967006 Online-Hilfe der Konsole dokumentiert keine OAMAuth- und WSSAuth-Authentifizierungsmodule.

Die Online-Hilfe der OpenSSO 8.0 Update-Admininistrationskonsole dokumentiert die Oracle Access Manager- (OAM) und Web Services Security-(WSS)-Authentifizierungsmodule nicht.

Problemumgehung Informationen in diesen Authentifizierungsmodulen finden Sie in Kapitel 3, „Verwenden des Sicherheitstoken-Diensts“

CR 6953582: Fedlet Java API-Referenz muss öffentlich sein.

Die öffentliche Referenz für Fedlet Java API ist als Bestandteil der Oracle OpenSSO 8.0 Update 2 Java API Reference erhältlich, die in der folgenden Dokumentationssammlung verfügbar ist: `http://docs.sun.com/coll/1767.1`.

Hinweis: OpenSSO 8.0 Update 2 unterstützt nicht die Methode `getPolicyDecisionForFedlet`, auch wenn sich diese Methode in der Java API-Reference befindet.

CR 6953579: OpenSSO Fedlet README-Datei muss die Funktion für die einmalige Abmeldung dokumentieren.

Die Fedlet README-Dateien dokumentieren die Funktion für die einmalige Abmeldung nicht.

Problemumgehung Für Oracle OpenSSO 8.0 Update 2 ist die Fedlet-Funktion für die einmalige Abmeldung in Kapitel 4, „Arbeiten mit dem Oracle OpenSSO Fedlet“ dokumentiert.

Zusätzliche Informationen und Ressourcen

Weitere nützliche Informationen und Ressourcen finden Sie an folgenden Stellen:

- „Abschreibungsbenachrichtigungen und -ankündigungen“ auf Seite 17
- „Problemmeldungen und Feedback“ auf Seite 18
- „Zugriffsfunktionen für Personen mit Behinderungen“ auf Seite 18
- „Verwandte Websites von Drittanbietern“ auf Seite 18
- Oracle Advanced Customer Services für Systeme:

  `http://www.oracle.com/us/support/systems/advanced-customer-services/index.html`
- Softwareprodukte: `http://www.oracle.com/us/sun/sun-products-map-075562.html`
- SunSolve: `http://sunsolve.sun.com/`
- Sun Developer Network (SDN): http://developers.sun.com/
- Sun Developer Services:`http://developers.sun.com/services/`

Abschreibungsbenachrichtigungen und -ankündigungen

- Die Service Management Service-(SMS)-APIs (`com.sun.identity.sm` -Paket) und das SMS-Modell sind in künftigen OpenSSO-Versionen nicht enthalten.
- Das UNIX-Authentifizierungsmodul und der UNIX-Authentifizierungs-Helper (`amunixd`) sind in künftigen OpenSSO-Versionen nicht enthalten.
- Laut den Sun Java System Access Manager 7.1-Versionshinweisen sind das Access Manager-Paket `com.iplanet.am.sdk`, das allgemein als Access Manager SDK (AMSDK) bezeichnet wird, und alle verwandten APIs und XML-Vorlagen, nicht in künftigen OpenSSO-Versionen enthalten.

  Wenn AMSDK entfernt wird, werden die Option Legacy Mode und Support ebenfalls entfernt.

  Zurzeit stehen Migrationsoptionen zur Verfügung, die in Zukunft wahrscheinlich nicht mehr angeboten werden. Oracle Identity Manager bietet Bereitstellungslösungen, die Sie anstelle von AMSDK verwenden können. Weitere Informationen zu Identity Manager

finden Sie unter http://www.oracle.com/products/middleware/identity-management/identity-manager.html.

Problemmeldungen und Feedback

Wenn Sie Fragen zu oder Probleme mit OpenSSO 8.0 Update 2 oder einer folgenden Patch-Version haben, wenden Sie sich an Support Resources unter http://sunsolve.sun.com/.

Auf dieser Website finden Sie Verknüpfungen zur Knowledge Base, zum Online Support Center, zum ProductTracker und auch zu Wartungsprogrammen und Kontaktinformationen für den Kundendienst. Wenn Sie Hilfe zu einem Problem angeben, übermitteln Sie bitte die folgenden Informationen:

- Beschreibung des Problems, einschließlich der Situation, in der das Problem aufgetreten ist und die damit verbundenen Auswirkungen auf den Betriebsablauf.
- Rechnertyp, Version des Betriebssystems, Webcontainer und Version, JDK-Version und OpenSSO-Version, einschließlich der Patches und anderer Software, die sich auf das Problem auswirken kann.
- Schritte zum Reproduzieren des Problems
- Sämtliche Fehlerprotokolle oder Kernspeicherauszüge.

Zugriffsfunktionen für Personen mit Behinderungen

Um Zugriffsfunktionen zu erhalten, die seit der Herausgabe dieser Medien veröffentlicht wurden, lesen Sie die Section 508-bezogenen Product Assessments (Produktbewertungen), die von Sun auf Anfrage zur Verfügung gestellt werden.

Weitere Informationen zum Engagement von Oracle zur Steigerung der Zugänglichkeit finden Sie unter http://www.oracle.com/index.html.

Verwandte Websites von Drittanbietern

In dieser Dokumentation wird auf URLs von Drittanbietern verwiesen, über die zusätzliche relevante Informationen zur Verfügung gestellt werden.

Hinweis – Oracle haftet nicht für die Verfügbarkeit der Websites Dritter, die in diesem Dokument erwähnt werden. Oracle unterstützt keine Inhalte, Werbung, Produkte oder sonstige Materialien, die auf oder über solche Websites oder Ressourcen verfügbar sind, und übernimmt keine Verantwortung oder Haftung dafür. Oracle übernimmt keine Verantwortung oder Haftung für tatsächliche oder angebliche Schäden oder Verluste, die durch den Gebrauch von oder in Verbindung mit derartigen Inhalten, Gütern oder Diensten entstanden sind, die auf diesen oder durch diese Websites oder Ressourcen verfügbar sind.

◆ ◆ ◆ KAPITEL 2

Installieren von OpenSSO 8.0 Update 2

In diesem Kapitel werden die folgenden Themen behandelt.

Überblick zur Installation von OpenSSO 8.0 Update 2

OpenSSO 8.0 Update 2 ist als Patch **TBS** verfügbar.

Bevor Sie OpenSSO 8.0 Update 2 (oder folgende Patches) installieren, prüfen Sie die Informationen zu neuen Funktionen, Hardware- und Softwareanforderungen und Problemen und Problemumgehungen in diesem Dokument.

OpenSSO 8.0 Update 2 enthält die Datei `opensso.war`, die Sie mithilfe dieser Methoden installieren können:

- **Patchen einer bestehenden OpenSSO 8.0-Bereitstellung**: Verwenden Sie das Dienstprogramm `ssopatch` in Update 2, um eine bestehende OpenSSO 8.0-Bereitstellung zu patchen, wie in diesem Kapitel beschrieben.

Hinweis - Oracle unterstützt nur das Patching von OpenSSO 8.0-Versionen. Beispielsweise wird das Patching von OpenSSO 8.0 mit OpenSSO 8.0 Update 2 unterstützt.

- **Installieren einer neuen OpenSSO 8.0 Update 2-Bereitstellung** Installieren und konfigurieren Sie die OpenSSO 8.0 Update 2-Datei `opensso.war`, wie beschrieben im *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.
- **Erstellen einer neuen spezialisierten WAR-Datei**: Verwenden Sie den Skript `createwar`, um mithilfe der Update 2-Datei `opensso.war` eine der folgenden neuen WAR-Dateien zu erstellen:
  - OpenSSO-Administrationskonsole, nur mit WAR
  - WAR des verteilten Authentication-UI-Servers
  - WAR nur für OpenSSO-Server, ohne die Administrationskonsole
  - IDP-Erkennungsdienst-WAR

    Weitere Informationen finden Sie in Kapitel 4, „Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File“ in *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.
- **Patchen einer bestehenden spezialisierten OpenSSO WAR-Datei**: Verwenden Sie das Dienstprogramm `ssopatch` in Update 2, um eine bestehende spezialisierte OpenSSO 8.0 WAR-Datei zu erstellen, wie beschrieben in Kapitel 23, „Patching OpenSSO Enterprise 8.0“ in *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

Hinweis – Wenn Sie Access Manager 7.1 oder Access Manager 7 2005Q4 ausführen und Sie einen Uprade auf Update 2 ausführen möchten, führen Sie die folgenden Schritte aus:

1. Rüsten Sie Access Manager 7.x auf OpenSSO 8.0 auf, wie beschrieben im *Sun OpenSSO Enterprise 8.0 Upgrade Guide*.
2. Wenden Sie das Update 2-Patch an, wie in diesem Kapitel beschrieben.

OpenSSO 8.0 Update 2-Patches

Sun veröffentlicht regelmäßig Patches für OpenSSO 8.0 Update 2. Prüfen Sie hier regelmäßig, ob Informationen zu diesen Patches hier vorliegen.

Planen des Patchvorgangs

▼ So planen Sie den Patchvorgang für OpenSSO 8.0:

1 Lesen Sie den „Überblick zum Dienstprogramm `ssopatch`“ auf Seite 23.

2. **Installieren Sie das Patch-Dienstprogramm für Ihre Plattform, wie beschrieben in „Installieren des Dienstprogramms ssopatch“ auf Seite 24.**

3. **Rufen Sie Informationen zur bestehenden WAR-Datei ab, um zu ermitteln, ob die bestehende WAR-Datei angepasst oder geändert wurde, wie beschrieben in „Vergleichen einer OpenSSO WAR-Datei mit ihrem internen Manifest“ auf Seite 26.**

4. **Vergleichen Sie die bestehende WAR-Datei und die Update 2-WAR-Datei, um die Dateien angepasst, aktualisierte Dateien in der neuen WAR-Datei und die zwischen den beiden WAR-Versionen hinzugefügten oder gelöschten Dateienan die ursprüngliche WAR zurückzuschicken, wie beschrieben in „Vergleichen von zwei OpenSSO WAR-Dateien“ auf Seite 27.**

5. **Sichern und archivieren Sie die bestehende OpenSSO-WAR-Datei, wie beschrieben in „Sichern einer OpenSSO WAR-Datei“ auf Seite 25.**

6. **Patchen Sie die OpenSSO-WAR-Datei, wie beschrieben in „Patching einer neuen OpenSSO WAR-Datei“ auf Seite 28.**

7. **Führen Sie den Skript `updateschema` aus, wie beschrieben in „Ausführen des Skripts `updateschema`“ auf Seite 31.**

   Hinweis - Wenn Sie eine spezialisierte WAR-Datei patchen, die Sie aus einer `opensso.war` generiert haben, z. B. einen WAR nur für OpenSSO-Server, ohne die Administrationskonsole, verteilten Authentifizierungs-UI-Server oder eine IDP-Erkennungsdienst-WAR, beachten Sie „Patching einer spezialisierten OpenSSO WAR“ auf Seite 30.

Überblick zum Dienstprogramm `ssopatch`

Das Dienstprogramm `ssopatch` ist ein Java-Befehlszeilen-Dienstprogramm, das auf auf Solaris- und Linux-Systemen als`ssopatch` und auf Windows als `ssopatch.bat` verfügbar ist.

Hinweis - Die Syntax für `ssopatch` in OpenSSO 8.0 Update 2 hat sich seit der Veröffentlichung von OpenSSO 8.0 beträchtlich geändert. Die neue Syntax finden Sie unter „Ausführen des Skripts `updateschema`“ auf Seite 31.

Das Patch-Dienstprogramm `ssopatch` führt diese Funktionen aus:

- Vergleicht eine OpenSSO WAR mit dem ursprünglichen Manifest, um zu ermitteln, ob die WAR-Datei angepasst oder verändert wurde.
- Vergleichen Sie zwei OpenSSO WAR-Dateien, um die Unterschiede zwischen zwei Dateien zu ermitteln, einschließlich der Anpassungen an der ursprünglichen WAR-Datei und den Änderungen in der neuen WAR-Datei.
- Generiert einen Staging-Bereich der Dateien, die zum Generieren einer neuen gepatchten OpenSSO WAR-Datei benötigt werden.

Wenn Sie die OpenSSO 8.0 Update 2-ZIP-Datei (opensso_80U2.zip) heruntergeladen und dekomprimiert haben, sind die Patch-Dienstprogramme und verwandten Dateien in der Datei ssoPatchTools.zip im Verzeichnis *zip-root*/opensso/tools verfügbar, wobei *zip-root* das Verzeichnis ist, in dem Sie opensso_80U2.zip dekomprimiert haben.

Das Dienstprogramm ssopatch verwendet eine Manifestdatei, um den Inhalt einer bestimmten OpenSSO WAR-Datei zu ermitteln. Eine Manifestdatei ist eine ASCII-Textdatei, die Folgendes enthält:

- Eine Zeichenfolge, die die spezifische Version der OpenSSO WAR-Datei kennzeichnet.
- Alle einzelnen Dateien in der OpenSSO WAR-Datei mit den Prüfsummenangaben für jede Datei.

Die Manifestdatei wird üblicherweise OpenSSO.manifest genannt und wird im Verzeichnis META-INF der OpenSSO WAR-Datei gespeichert.

Das Dienstprogramm ssopatch sendet seine Ergebnis an die Standardausgabe (stdout). Sie können auf Wunsch die ssopatch-Ausgabe erfassen, indem Sie die Ausgabe an eine Datei weiterleiten. Wenn ssopatch erfolgreich abgeschlossen wird, gibt es einen Null-Beendigungscode (0) zurück. Wenn Fehler auftreten, gibt ssopatch einen Beendigungscode zurück, der ungleich null ist.

Installieren des Dienstprogramms ssopatch

Vor dem Installieren des Dienstprogramms ssopatch:

- Laden Sie die OpenSSO 8.0 Update 2-ZIP-Datei herunter und dekomprimieren Sie sie (opensso_80U2.zip).
- Stellen Sie den Umgebungsvariablenpunkt JAVA_HOME auf JDK 1.5 oder höher ein.

So installieren Sie das Dienstprogramm ssopatch:

1. Machen Sie die Datei ssoPatchTools.zip im Verzeichnis *zip-root*/opensso/tools ausfindig, wobei *zip-root* das Verzeichnis ist, in dem Sie opensso_80U2.zip dekomprimiert haben.
2. Erstellen Sie ein neues Verzeichnis, um die Datei ssoPatchTools.zip zu dekomprimieren. Zum Beispiel: ssopatchtools.
3. Dekomprimieren Sie die Datei ssoPatchTools.zip im neuen Verzeichnis.
4. Wenn Sie das Dienstprogramm ssopatch über ein anderes Verzeichnis als das aktuelle Verzeichnis ausführen möchten, ohne den vollständigen Pfad anzugeben, fügen Sie das Dienstprogramm in die Variable PATH ein.

In der folgenden Tabelle werden die Dateien in ssoPatchTools.zip beschrieben.

Datei oder Verzeichnis	Beschreibung
README	Readme-Datei, die ssopatch beschreibt.
/lib	Erforderliche ssopatch-JAR-Dateien.
/patch	updateschema- und updateschema.bat-Skripte und verwandte XML-Dateien
/resources	Erforderliche Eigenschaftendateien
ssopatch und ssopatch.bat	Dienstprogramme für Solaris-, Linux- und Windows-Systeme

Sichern einer OpenSSO WAR-Datei

Sichern Sie zunächst Ihre vorhandene OpenSSO WAR-Datei und die Konfigurationsdaten:

- Kopieren Sie die bestehende OpenSSO WAR-Datei an einen sicheren Speicherort. Falls Sie Update 2 aus irgendeinem Grund aufgeben müssen, können Sie die Sicherungskopie der WAR-Datei erneut bereitstellen.
- Sichern Sie die Konfigurationsdaten, wie beschrieben in Kapitel 15, „Backing Up and Restoring Configuration Data“ in *Sun OpenSSO Enterprise 8.0 Administration Guide*.

Ausführen des Dienstprogramms ssopatch

Zum Ausführen des Dienstprogramms ssopatch gehen Sie folgendermaßen vor:

```
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]
```

wobei folgende Optionen gelten:

- `-war-file|-o` gibt einen Pfad zu einer WAR-Datei an (z. B. `opensso.war`), die zuvor bereitgestellt wurde.
- `-manifest|-m` gibt den Pfad zur Manifestdatei an, die Sie erstellen möchten. Die Manifestdatei wird über die WAR-Datei generiert, die durch `-war-file|-o` angegeben wird, wenn diese Option eingestellt wird.
- `-war-file-compare|-c` gibt einen Pfad zu einer WAR-Datei an, die mit der WAR-Datei verglichen wird, die durch `-war-file|-o` angegeben wird.
- `-staging|-s` gibt einen Pfad zum Staging-Bereich an, in den die Dateien aus einer OpenSSO WAR geschrieben werden.
- `-locale|-l` gibt das zu verwendende Gebietsschema an. Wenn diese Option nicht angegeben ist, verwendet `ssopatch` das standardmäßige Systemgebietsschema.
- `-override|-r` überschreibt die Revisionsprüfung für die beiden WAR-Dateien. Durch Prüfen der Revision werden die Versionen der WAR-Dateien ermittelt. Der Vorgang wird nur fortgesetzt, wenn die Versionen kompatibel sind. Diese Option ermöglicht es Ihnen, diese Prüfung zu überschreiben.

  Standardwert ist false (Revisionsprüfung wird ausgeführt).
- `-overwrite|-w` überschreibt die Dateien im bestehenden Staging-Bereich. Standardwert ist false (Dateien werden nicht überschrieben).

Vergleichen einer OpenSSO WAR-Datei mit ihrem internen Manifest

Verwenden Sie dieses Verfahren, um zu ermitteln, ob eine OpenSSO WAR-Datei angepasst oder geändert wurde, seit sie heruntergeladen wurde.

Das Dienstprogramm `ssopatch` generiert eine neue interne Manifestdatei und vergleicht anschließend dieses interne Manifest mit dem Manifest, das in der ursprünglichen OpenSSO WAR-Datei im Verzeichnis `META-INF` gespeichert ist.

So vergleichen Sie eine OpenSSO WAR-Datei mit ihrem internen Manifest:

1. Führen Sie `ssopatch` aus, um die OpenSSO WAR-Datei mit dem internen Manifest zu vergleichen. Beispiel:

```
./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
```

In diesem Beispiel werden diese Änderungen in der ursprünglichen WAR-Datei dargestellt:

- `images/login-origimage.jpg` befindet sich in `opensso.war` , wurde jedoch nicht im ursprünglichen Manifest gefunden.
- `images/login-backimage.jpg` wurde in `opensso.war` gegenüber dem ursprünglichen Manifest angepasst.
- Die Datei `WEB-INF/classes/amConfigurator.properties` wurde in `opensso.war` gegenüber dem ursprünglichen Manifest angepasst.

Vergleichen von zwei OpenSSO WAR-Dateien

Verwenden Sie dieses Verfahren, um zwei WAR-Dateien zu vergleichen, um die Dateien anzuzeigen, mit denen der folgende Vorgang ausgeführt wurde:

- Angepasst in einer ursprünglichen OpenSSO WAR
- Aktualisiert in einer neuen OpenSSO WAR-Datei
- Hinzugefügt oder gelöscht zwischen den beiden OpenSSO WAR-Versionen

So vergleichen Sie zwei OpenSSO WAR-Dateien:

1. Führen Sie `ssopatch` aus, um die beiden WAR-Dateien zu vergleichen. In diesem Beispiel wird die Option `-override` verwendet, um die Revisionsprüfung zwischen den beiden WAR-Dateien zu überschreiben:

```
./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
```

In diesem Beispiel werden die Dateien angezeigt, die in einer neuen WAR-Datei aktualisiert und angepasst wurden.

Patching einer neuen OpenSSO WAR-Datei

Verwenden Sie dieses neue Verfahren, um einen neuen Staging-Bereich zu erstellen, in dem eine ursprüngliche WAR-Datei mit einer neuen WAR-Datei zusammengeführt wird.

In diesem Vorgang werden die Manifeste für jede WAR-Datei verglichen und anschließend Folgendes angezeigt:

- Dateien, die in der ursprünglichen WAR-Datei angepasst wurden.
- Dateien, die in einer neuen WAR-Datei aktualisiert wurden.
- Dateien, die zwischen den beiden WAR-Dateiversionen hinzugefügt oder entfernt wurden.

`ssopatch` kopiert anschließend die entsprechenden Dateien in ein Staging-Verzeichnis, in dem Sie Anpassungen hinzufügen müssen, bevor Sie die neue, gepatchte WAR-Datei erstellen und bereitstellen.

So erstellen Sie einen Staging-Bereich zum Patchen einer OpenSSO WAR-Datei:

1. `ssopatch` verändert zwar nicht Ihre ursprüngliche `opensso.war`-Datei, jedoch wird empfohlen, dass sie diese Datei sichern, falls Sie die gepatchte `opensso.war`-Datei rückgängig machen möchten.
2. Führen Sie `ssopatch` aus, um den Staging-Bereich zu erstellen. Beispiel:

```
./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
```

In diesem Beispiel ist `/tmp/staging` der Staging-Bereich, in den `ssopatch` die Dateien kopiert.

Aktualisieren Sie die Dateien wie gewünscht im Staging-Bereich und verwenden Sie dazu die Ergebnisses der vorherigen Schritts.

Verwenden Sie die folgende Tabelle, um die Aktion zu ermitteln, die Sie unter Umständen an jeder Datei ausführen müssen, bevor Sie eine neue gepatchte WAR-Datei generieren.

ssopatch-Ergebnisse	**Erklärung und Aktion erforderlich**
`Datei nicht in ursprünglicher WAR` *Dateiname*.	Die angegebene Datei existiert nicht in der ursprünglichen WAR-Datei, sondern in der letzten Version der WAR-Datei. **Aktion**: Keine
`Datei in neuer WAR` *Dateiname* aktualisiert.	Die angegebene Datei existiert in der ursprünglichen und neuen WAR-Datei und wurde in der neuesten Version der WAR-Datei aktualisiert. In der ursprünglichen WAR-Datei sind keine Anpassungen erfolgt. **Aktion**: Keine
`Datei ist benutzerdefiniert`*Dateiname*	Die angegebene Datei existiert in beiden WAR-Dateien, wurde in der ursprünglichen Version der WAR-Datei angepasst, wurde jedoch nicht in der neuesten Version der WAR-Datei aktualisiert. **Aktion**: Keine
`Erfordert möglicherweise manuelle Benutzerdefinition` *Dateiname*	Die angegebene Datei existiert in beiden WAR-Dateien, wurde in der ursprünglichen Version der WAR-Datei angepasst und wurde in der neuesten Version der WAR-Datei aktualisiert. **Aktion**: Wenn die Anpassungen in der Datei stattfinden sollen, müssen Sie sie manuell in der neuen aktualisierten Datei im Staging-Verzeichnis hinzufügen.
`Datei war in ursprünglicher WAR-Datei benutzerdefiniert, konnte aber in neuer WAR-Datei nicht gefunden werden.`	Die Datei existierte in der ursprünglichen WAR-Datei, befindet sich jedoch nicht in der neuen WAR. **Aktion**: Keine

Nächste Schritte

1. Erstellen Sie eine neue OpenSSO WAR-Datei mithilfe der Dateien im Staging-Bereich. Beispiel:

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

   wobei`/patched/opensso.war` die neue gepatchte OpenSSO WAR-Datei ist.
2. Stellen Sie die Datei `/patched/opensso.war` im Webcontainer über die ursprüngliche Bereitstellungs-URL bereit. Zum Beispiel `/opensso`

Änderungen an der OpenSSO-Konfiguration Eine neue OpenSSO WAR-Datei enthält unter Umständen Konfigurationsänderungen, die in der ursprünglichen WAR-Datei nicht enthalten waren. Konfigurationsänderungen werden für jedes Patch gesondert dokumentiert. Weitere

Informationen zu den Konfigurationsänderungen finden Sie in der Patch-Dokumentation und den *Sun OpenSSO Enterprise 8.0, Versionshinweise*. (Die Versionszeichenfolge in der OpenSSO-Manifestdatei wird geändert, auch wenn die neue WAR-Datei keine Konfigurationsänderungen enthält.)

Wenn Sie die gepatchte Version rückgängig machen möchten, heben Sie die Bereitstellung der gepatchten WAR-Datei auf und bringen die ursprüngliche WAR-Datei erneut aus.

Erstellen einer OpenSSO WAR-Manifestdatei

Eine OpenSSO-Manifestdatei ist eine Textdatei, die alle einzelnen Dateien in einer WAR-Datei für eine spezifische Version kennzeichnet, mit Prüfsummenangaben für jede Datei.

Verwenden Sie dieses Verfahren, um eine Manifestdatei zu erstellen, die in einer spezialisierten OpenSSO WAR enthalten sein kann, z. B. einem ausschließlichen OpenSSO-Server, einer ausschließlichen Administrationskonsole, einem verteilten Authentifizierungs-UI-Server oder einer IDP-Erkennungsdienst-WAR.

So erstellen Sie eine OpenSSO WAR-Manifestdatei

1. Führen Sie `ssopatch` aus, um die OpenSSO-Manifestdatei zu erstellen. Beispiel:

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

   wobei `opensso.war` eine bestehende OpenSSO WAR-Datei ist.

   Das Dienstprogramm `ssopatch` erstellt eine neue Manifestdatei mit dem Namen `manifest` im Verzeichnis `/tmp`.

2. Kopieren Sie diese neue Manifestdatei in das Verzeichnis `META-INF` in der Datei `opensso.war`, damit die WAR-Datei gepatcht werden kann. Beispiel:

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

Patching einer spezialisierten OpenSSO WAR

Wenn Sie zuvor eine spezialisierte OpenSSO WAR erstellt haben, z. B. einen ausschließlichen OpenSSO-Server, eine ausschließliche Administrationskonsole, einen verteilten Authentifizierungs-UI-Server oder eine IDP-Erkennungsdienst-WAR, können Sie sie mit dem Dienstprogramm `ssopatch` patchen.

So patchen Sie eine spezialisierte OpenSSO WAR

1. Erstellen Sie eine Manifestdatei für Ihre spezialisierte OpenSSO WAR, wie beschrieben in „Erstellen einer OpenSSO WAR-Manifestdatei“ auf Seite 30.

   Hinweis: Erstellen Sie die Manifestdatei auf Basis der ursprünglichen OpenSSO 8.0-Datei `opensso.war`, wie sie von Sun geliefert wurde, bevor Sie Benutzerdefinitionen vornehmen. Wenn das Manifest nach den Benutzerdefinitionen erstellt wird, verwendet `ssopatch` möglicherweise die Dateien aus Update 2 und nicht Ihre Benutzerdefinitionen. In diesem Fall müssen Sie die Benutzerdefinitionen nach dem Patchen erneut vornehmen.
2. Generieren Sie die spezialisierte OpenSSO WAR über die OpenSSO 8.0 Update 2-Datei `opensso.war`, wie beschrieben in Kapitel 4, „Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File“ in *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.
3. Verwenden Sie die alten und neuen WAR-Dateien mithilfe des Dienstprogramms `ssopatch`.
4. Generieren eines Staging-Bereichs für eine neue spezialisierte WAR-Datei, wie beschrieben in „So erstellen Sie einen Staging-Bereich zum Patchen einer OpenSSO WAR-Datei:“ auf Seite 28.
5. Erneutes Bereitstellen der neuen spezialisierten WAR-Datei.

Ausführen des Skripts `updateschema`

Nach dem Ausführen von `ssopatch` führen Sie auf Solaris- oder Linux-Systemen `updateschema.sh` und auf Windows-Systemen `updateschema.bat` aus. Mit diesem Skript wird die OpenSSO-Serverversion aktualisiert, werden die neuen Standard-Servereigenschaften hinzugefügt, die neuen Attributschemata hinzugefügt, die für Bug Fixes und Verbesserungen benötigt werden. Sie müssen `updateschema` ausführen, um die Serverversion zu aktualisieren.

Vorbereitung

- Der Skript `updateschema.sh` oder `updateschema.bat` benötigt die Update 2-Version (oder höher) des Befehlszeilen-Dienstprogramms `ssoadm`. Führen Sie daher vor dem Ausführen dieses Skripts das Update 2-Administrations-Tool aus, wie beschrieben in Kapitel 3, „Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools“ in *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.
- Der Skript `updateschema.bat` führt mehrere `ssoadm`-Befehle aus. Daher müssen Sie vor dem Ausführen von `updateschema.bat` auf Windows-Systemen eine Kennwortdatei erstellen, die den Kennwortbenutzer in Klartext für den Benutzer `amadmin` enthält. Der Skript `updateschema.bat` fordert Sie auf, den Pfad zur Kennwortdatei einzugeben. Bevor der Skript beendet wird, entfernt er die Kennwortdatei.

So führen Sie den Skript `updateschema` aus:

1. Wechseln Sie zum Verzeichnis *patch-tools*, wobei *patch-tools* der Pfad ist, unter dem Sie `ssoPatchTools.zip` dekomprimiert haben.
2. Führen Sie `updateschema.sh` oder `updateschema.bat` aus. Z. B. auf Solaris-Systemen:

   ```
   ./updateschema.sh
   ```

3. Geben Sie die folgenden Informationen ein, wenn der Skript Sie dazu auffordert:
   - Vollständiger Pfad zum Dienstprogramm `ssoadm` (schließt `ssoadm`selbst aus). Zum Beispiel: `/opt/ssotools/opensso/bin`
   - `amadmin`-Passwort

   Das Skript `updateschema.sh` oder `updateschema.bat` schreibt Nachrichten oder Fehler in die Standardausgabe.
4. Starten Sie den OpenSSO 8.0 Update 2-Webcontainer erneut auf.

Rückgängig machen einer Patch-Installation

Wenn Sie die Patch-Installation rückgängig machen möchten, stellen Sie einmalig die ursprüngliche Datei `opensso.war` (oder eine spezialisierte WAR-Datei) wieder bereit.

◆ ◆ ◆ KAPITEL 3

Verwenden des Sicherheitstoken-Diensts

Als vertrauenswürdiger Autoritätsdienst gibt der OpenSSO-Sicherheitstoken-Dienst Sicherheits-Token aus und validiert sie. Als Webdienste-Sicherheitsanbieter kommuniziert der Sicherheitstoken-Dienst gesichert zwischen dem Webdienst-Client und dem OpenSSO STS-Dienst selbst. Seit Open SSO 8.0 Update 2 sind zahlreiche Verbesserungen am Sicherheitstoken-Dienst erfolgt.

In diesem Kapitel werden die folgenden Themen behandelt.

Hinzufügen eines WSSAuth-Authentifizierungsmoduls

Das Webdienst-Sicherheitsauthentifizierungsmodul ermöglicht es OpenSSO, einen Benutzernamen mit einem gesammelten Passwort zu validieren, das als Authentifizierungs-Token erhalten wurde und in einer Dienstanforderung vom Webdienst-Client an einen Webdienstanbieter enthalten war.

▼ So fügen Sie eine neue Instanz eines Webdienst-Sicherheitsauthentifizierungsmoduls hinzu:

1 **Klicken Sie in der Registerkarte Access Manager auf die Unterregisterkarte Authentication (Authentifizierung).**

2 **Im Abschnitt Module Instances (Modulinstanzen) klicken Sie auf New (Neu).**

3 **Geben Sie im Feld Name einen Namen für diese Instanz des WSSAuth-Authentifizierungsmoduls ein.**

4 **Wählen Sie als Typ WSSAuth aus.**

5 **Konfigurieren Sie die Instanz des WSSAuth-Authentifizierungsmoduls.**

▼ So konfigurieren Sie eine Instanz des WSSAuth-Authentifizierungsmoduls:

1 **Klicken Sie in der Registerkarte Access Manager auf die Unterregisterkarte Authentication (Authentifizierung).**

2 **Klicken Sie im Abschnitt Module Instances (Modulinstanzen) auf den Namen der WSSAuth-Authentifizierungsinstanz, die Sie konfigurieren möchten.**

3 **Geben Sie Werte für die Attribute WSSAuth Authentication Module Instance Realm (WSSAuth-Authentifizierungsmodul-Instanzbereich) ein.**

In der folgenden Tabelle werden eine Liste und Beschreibungen der Attribute dargestellt, die Sie konfigurieren können.

Benutzersuchattribut	Wird noch entwickelt
Benutzerbereich	Wird noch entwickelt
Benutzerpasswortattribut	Wird noch entwickelt
Authentication Level	Wird noch entwickelt

Hinzufügen eines OAMAuth-Authentifizierungsmoduls

Das Oracle-Authentifizierungsmodul ermöglicht es OpenSSO, einen Administrator in OpenSSO zu authentifizieren und einmalig anzumelden, der sich zuvor bei Oracle Access Manager authentifiziert hat. Der Administrator muss keine Berechtigungsnachweise in OpenSSO angeben.

▼ So fügen Sie eine neue Instanz eines Oracle-Authentifizierungsmoduls hinzu:

1. **Klicken Sie in der Registerkarte Access Manager auf die Unterregisterkarte Authentication (Authentifizierung).**
2. **Klicken Sie im Abschnitt Modules Instances auf New.**
3. **Geben Sie im Feld Name einen Namen für diese Oracle-Authentifizierungsmodulinstanz ein.**
4. **Wählen Sie als Typ OAMAuth aus.**
5. **Klicken Sie auf „OK".**
6. **Konfigurieren Sie die Instanz des OAMAuth-Authentifizierungsmoduls.**

▼ So konfigurieren Sie eine Instanz eines Oracle-Authentifizierungsmoduls:

1. **Klicken Sie in der Registerkarte Access Manager auf die Unterregisterkarte Authentication (Authentifizierung).**
2. **Klicken Sie im Abschnitt Module Instances (Modulinstanzen) auf den Namen der OAMAuth-Authentifizierungsinstanz, die Sie konfigurieren möchten.**
3. **Geben Sie Werte für die Attribute Oracle Authentication Module Instance Realm (Oracle-Authentifizierungsmodul-Instanzbereich) ein.**

   In der folgenden Tabelle werden eine Liste und Beschreibungen der Attribute dargestellt, die Sie konfigurieren können.

Kopzeilenname des Remote-Benutzers	Wird noch entwickelt
Zugelassene Kopfzeilenwerte	Die Liste Current Values (Aktuelle Werte) zeigt To Be Developed (Wird noch entwickelt) an. ■ Zum Hinzufügen eines Kopfzeilenwerts in der Liste geben Sie im Feld New Value (Neuer Wert) To Be Developed ein und klicken auf Add (Hinzufügen).

	■ Um einen Eintrag aus der Liste Current Values zu entfernen, wählen Sie den Eintrag aus und klicken auf Remove (Entfernen).
Authentifizierungsebene	Wird noch entwickelt

Generieren von Sicherheitstoken

Oracle OpenSSO Security Token Service (OpenSSO STS) stellt eine Vertrauensstellung zwischen einem Webdienst-Client und einem Webdienstanbieter her und vermittelt anschließend das Vertrauen zwischen ihnen. Kann der Webdienst Tokens vertrauen, die von lediglich einer Entität ausgegeben wurden? OpenSSO STS? Statt mit mehreren Clients kommunizieren zu müssen. Auf diese Weise reduziert OpenSSO STS den Aufwand zur Vertrauenspunktverwaltung.

In den folgenden Abschnitten werden Anweisungen zum Ermitteln Ihrer Sicherheitstoken-Anforderungen und zum entsprechenden Konfigurieren des Sicherheitstoken-Dienstes zum Generieren und Validieren von Sicherheitstoken dargestellt.

Registrieren eines Webdienstanbieters in OpenSSO STS

Wenn Sie ein neues Webdienstanbieter-Sicherheitsagentenprofil hinzufügen, wird der Webdienstanbieter automatisch in OpenSSO STS registriert. In den folgenden Abschnitten werden mehr Details dargestellt:

Wenn Sie einen Webdienstanbieter in OpenSSO STS registriert haben, können Sie OpenSSO STS dazu konfigurieren, Webclient-Sicherheitstokens zu generieren, die der Webdienstanbieter akzeptieren kann.

Anfordern eines Webdienst-Clientsicherheits-Tokens von OpenSSO STS

Bevor Sie den Sicherheitstoken-Dienst zum Generieren von Webclient-Sicherheitstoken konfigurieren, müssen Sie ermitteln, welche Art von Sicherheitstoken der Webdienstanbieter benötigt. OpenSSO STS unterstützt Liberty Alliance Project-Sicherheitstoken und Web Services-Interoperability Basic Security Profile-Sicherheitstoken.

Prozessablauf beim Generieren von Sicherheitstokens

Wenn Sicherheit mithilfe von Liberty Alliance Project-Tokens aktiviert ist, sendet der HTTP-Client oder der Browser über den Webdienst-Client eine Zugriffsanforderung an den Webdienstanbieter. Ein Webdienste-Sicherheitsagent leitet die Anforderung an den OpenSSO STS-Authentifizierungsdienst weiter. Wenn der Liberty Alliance Project-Sicherheitsmechanismus aktiv ist, gibt ein HTTP-Sicherheitsagent die Weiterleitung aus. Wenn WS-IBS-Sicherheit verwendet wird, gibt ein SOAP-Sicherheitsagent die Weiterleitung aus.

Der OpenSSO STS-Authentifizierungsdienst ermittelt den Sicherheitsmechanismus, der beim Webdienstanbieter registriert ist, und ruft die entsprechenden Sicherheitstoken ab. Nach der erfolgreichen Authentifizierung stellt der Webdienst-Client einen SOAP-Nachrichtentext bereit, während der SOAP-Sicherheitsagent auf Seite des Webdienst-Clients die Sicherheitskopfzeilen und ein Token einfügt. Die Nachricht wird anschließend entfernt, bevor die Anforderung an den WSP gesendet wird.

Der SOAP-Sicherheitsagent auf Seite des Webdienstanbieters überprüft die Signatur und den Sicherheitstoken in der SOAP-Anforderung, bevor er die Anforderung an den Webdienstanbieter selbst weiterleitet. Der Webdienstanbieter verarbeitet sie anschließend und gibt eine vom SOAP-Sicherheitsagenten signierte Antwort an den Webdienst-Client zurück. Der SOAP-Sicherheitsagent auf Seite des Webdienst-Clients überprüft anschließend die Signatur, bevor er die Antwort an den Webdienst-Client weiterleitet.

In der folgenden Tabelle werden eine Liste und kurze Beschreibungen von Tokens dargestellt, die für Liberty Alliance Project-Transaktionen unterstützt werden.

TABELLE 3–1 Requestor Tokens - Liberty Alliance Project

Token	Erfüllt diese Anforderungen
X.509	■ Der gesicherte Webdienst verwendet eine Public Key Infrastructure (Infrastruktur mit öffentlichen Schlüsseln), in der der Webdienst-Client einen öffentlichen Schlüssel als Mittel zum Ermitteln des Anforderers zur Verfügung stellt, und mit der der Webdienstanbieter authentifiziert wird. ■ Der gesicherte Webdienst verwendet eine Public Key Infrastructure (Infrastruktur mit öffentlichen Schlüsseln), in der der Webdienst-Client einen öffentlichen Schlüssel als Mittel zum Ermitteln des Anforderers zur Verfügung stellt, und mit der der Webdienstanbieter authentifiziert wird.

TABELLE 3–1 Requestor Tokens - Liberty Alliance Project *(Fortsetzung)*

BearerToken	■ Der gesicherte Webdienst verwendet die SAML-Bearer-Token-Bestätigungsmethode von Security Assertion Markup Language (SAML). ■ Der Webdienst-Client stellt eine SAML-Behauptung mit Informationen zum öffentlichen Schlüssel zur Verfügung, um den Anforderer gegenüber dem Webdienstanbieter zu authentifizieren. ■ Eine zweite Signatur bindet die Behauptung an die SOAP-Nachricht. ■ Die zweite Signaturbindung verwendet Regeln, die vom Liberty Alliance Project verwendet wurden.
SAML-Token	■ Der gesicherte Webdienst verwendet die SAML-Schlüsselinhaber-Bestätigungsmethode. ■ Der Webdienst-Client fügt eine SAML-Behauptung und eine digitale Signatur in eine SOAP-Kopfzeile ein. ■ Ein Absenderzertifikat oder öffentlicher Schlüssel wird ebenfalls mit der Signatur zur Verfügung gestellt. ■ Der Versand wird mithilfe von Regeln verarbeitet, die vom Liberty Alliance Project definiert wurden.

In den folgenden Tabellen werden eine Liste und kurze Beschreibungen von Tokens dargestellt, die für WS-IBS-Transaktionen unterstützt werden.

TABELLE 3–2 Anforderer-Tokens - WS-IBS

Token	**Erfüllt diese Anforderungen**
Benutzername	■ Der gesicherte Webdienst erfordert einen Benutzernamen, ein Passwort und optional eine Signierung für die Anforderung. ■ Der Webdienst-Verbraucher stellt ein Benutzernametoken als Mittel zum Identifizieren des Anforderers zur Verfügung. ■ Der Webdienstverbraucher stellt ein Passwort, gemeinsames Geheimnis oder ein Passwortäquivalent zum Authentifizieren der Identität gegenüber dem Webdienstanbieter zur Verfügung.
X.509	Der gesicherte Webdienst verwendet eine PKI (Public Key Infrastructure, in der der Webdienst-Verbraucher einen öffentlichen Schlüssel als Mittel zum Ermitteln des Anforderers und Abschließen des Authentifizierung gegenüber dem Webdienstanbieter.

TABELLE 3–2 Anforderer-Tokens - WS-IBS *(Fortsetzung)*

SAML-Schlüsselinhaber	■ Der gesicherte Webdienst verwendet die SAML-Schlüsselinhaber-Bestätigungsmethode. ■ Der Webdienst-Client stellt eine SAML-Behauptung mit Informationen zum öffentlichen Schlüssel zur Verfügung, um den Anforderer gegenüber dem Webdienstanbieter zu authentifizieren. ■ Eine zweite Signatur bindet die Behauptung an die SOAP-Payload.
SAML-SenderVouches	■ Der gesicherte Webdienst verwendet die SAML-Sender-Vouches-Bestätigungsmethode. ■ Der Webclient-Verbraucher fügt eine SAML-Behauptung und eine digitale Signatur in eine SOAP-Kopfzeile ein. Ein Absenderzertifikat oder öffentlicher Schlüssel wird ebenfalls mit der Signatur zur Verfügung gestellt.

Verwenden der Matrix zum Generieren von Sicherheitstokens

Verwenden Sie die Matrix zum Generieren von Sicherheitstoken als Hilfe beim Konfigurieren von OpenSSO STS zum Generieren von Webdienst-Client-Sicherheitstokens, die der Webdienstanbieter benötigt. Machen Sie zuerst in der letzten Spalte mit dem Titel OpenSSO STS Output Token (OpenSSO STS Ausgabetoken) eine Beschreibung ausfindig, die den Anforderungen des Webdienstanbieter-Tokens entspricht. Verwenden Sie anschließend die Parameterwerte in der gleichen Zeile, wenn Sie den Sicherheitstoken-Dienst konfigurieren. Die "Legende der Matrix zum Generieren von Tokens" enthält Informationen zu den Tabellenkopfzeilen und den verfügbaren Optionen. Im Abschnitt 5.2.3 "So konfigurieren Sie den Sicherheitstoken-Dienst" finden Sie detaillierte Konfigurationsanweisungen. Allgemeine Informationen zur Webdienstsicherheit und die entsprechende Terminologie finden Sie unter:

- 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

In der Matrix zum Generieren von Sicherheitstoken werden häufig verwendete Sicherheitstokendienst-Parametereinstellungen und die Typen der Sicherheitstoken zusammengefasst, die OpenSSO STS auf Basis dieser Einstellungen generiert.

TABELLE 3–3 Matrix zum Generieren von Sicherheitstokens

Zeile	Sicherheitsbindung auf Nachrichtenebene	Webdienst-Client-Typ	Key Type	OnBehalfOf Token	Use Key	OpenSSO STS-Ausgabetoken
1	Asymmetrisch	X509	Inhaber	Ja	Nein	SAML-Inhaber, kein Beweisschlüssel

TABELLE 3–3 Matrix zum Generieren von Sicherheitstokens *(Fortsetzung)*

2	Asymmetrisch	Benutzername	Inhaber	Ja	Nein	SAML-Inhaber, kein Beweisschlüssel
3	Asymmetrisch	X509	Inhaber	Nein	Nein	SAML-Inhaber, kein Beweisschlüssel
4	Asymmetrisch	Benutzername	Inhaber	Nein	Nein	SAML-Inhaber, kein Beweisschlüssel
5	Asymmetrisch	X509	Symmetrisch	Ja	Nein	SAML-Schlüsselinhaber, symmetrischer Beweisschlüssel
6	Asymmetrisch	Benutzername	Symmetrisch	Ja	Nein	SAML-Schlüsselinhaber, symmetrischer Beweisschlüssel
7	Asymmetrisch	X509	Symmetrisch	Nein	Nein	SAML Schlüsselinhaber, Symmetrisch
8	Asymmetrisch	Benutzername	Symmetrisch	Nein	Nein	SAML-Schlüsselinhaber, symmetrischer Beweisschlüssel
9	Asymmetrisch	X509	Asymmetrisch	Nein	Öffentlicher Schlüssel des Webdienstclients	SAML-Schlüsselinhaber, asymmetrischer Beweisschlüssel
10	Asymmetrisch	X509	Oracle-hergestellt für SAML-Absendernachweise	Ja	Nein	SAML-Absendernachweise, kein Beweisschlüssel
11	Asymmetrisch	Benutzername	Oracle-hergestellt für SAML-Absendernachweise	Ja	Nein	SAML-Absendernachweise, kein Beweisschlüssel
12	Asymmetrisch	X509	Oracle-hergestellt für SAML-Absendernachweise	Nein	Nein	FEHLER
13	Asymmetrisch	Benutzername	Oracle-hergestellt für SAML-Absendernachweise	Nein	Nein	FEHLER
14	Transport	Benutzername	Inhaber	Ja	Nein	SAML-Inhaber, kein Beweisschlüssel

TABELLE 3–3 Matrix zum Generieren von Sicherheitstokens *(Fortsetzung)*

15	Transport	Benutzername	Inhaber	Nein	Nein	SAML-Inhaber, kein Beweisschlüssel
16	Transport	Benutzername	Symmetrisch	Ja	Nein	SAML-Schlüsselinhaber Symmetrisch
17	Transport	Benutzername	Symmetrisch	Nein	Nein	SAML-Schlüsselinhaber symmetrischer Beweisschlüssel
18	Transport	Benutzername	Oracle-herstellerspezifisch für SAML-Absendernachweise	Ja	Nein	SAML-Absendernachwe kein Beweisschlüssel
19	Transport	Benutzername	Oracle-herstellerspezifisch für SAML-Absendernachweise	Nein	Nein	FEHLER
20	Asymmetrisch	Benutzername	Asymmetrisch	Nein	Öffentlicher Schlüssel des Webdienstclients	FEHLER
21	Transport	Benutzername	Asymmetrisch	Nein	Öffentlicher Schlüssel des Webdienstclients	FEHLER
22	Asymmetrisch	X509	Asymmetrisch	Ja	Nein	FEHLER
23	Asymmetrisch	Benutzername	Asymmetrisch	Ja	Nein	FEHLER
24	Transport	Benutzername	Asymmetrisch	Ja	Nein	FEHLER
25	Asymmetrisch	X509	Asymmetrisch	Nein	Nein	SAML-Schlüsselinhaber asymmetrischer Beweisschlüssel
26	Asymmetrisch	X509	Nein	Nein	Nein	SAML-Schlüsselinhaber asymmetrischer Beweisschlüssel
27	Asymmetrisch	Benutzername	Nein	Nein	Nein	SAML-Schlüsselinhaber symmetrischer Beweisschlüssel
28	Transport	Benutzername	Nein	Nein	Nein	SAML-Schlüsselinhaber symmetrischer Beweisschlüssel

Probleme und Problemumgehungen im Sicherheitstoken-Dienst

Wird noch entwickelt

Probleme und Problemumgehungen in der Konfiguration

Wird noch entwickelt

Dokumentations-Errata

Wird noch entwickelt

◆ ◆ ◆ KAPITEL 4

Arbeiten mit dem Oracle OpenSSO Fedlet

Dieser Abschnitt enthält die folgenden Informationen zum Oracle OpenSSO Fedlet:

Informationen zum Oracle OpenSSO Fedlet

Das Oracle OpenSSO Fedlet ist ein Lightweight-Dienstanbieter-Implementierung, die mit Java oder der .NET-Dienstanbieteranwendung bereitgestellt werden kann, sodass die Anwendung mit einem Identity-Anbieter, z. B. Oracle OpenSSO 8.0 Update 2, mithilfe des SAMLv2-Protokolls kommunizieren kann. Abhänging von der Plattform hat das Fedlet zwei Versionen:

- Das Java Fedlet wurde zuerst in OpenSSO 8.0 veröffentlicht. Informationen finden Sie in Kapitel 5, „Using the OpenSSO Enterprise Fedlet to Enable Identity Federation“ in *Sun OpenSSO Enterprise 8.0 Deployment Planning Guide*.
- Das .NET Fedlet wurde in OpenSSO 8.0 Update 1 veröffentlicht. Informationen finden Sie in Kapitel 10, „Using the ASP.NET Fedlet with OpenSSO Enterprise 8.0 Update 1“ in *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.

In Oracle OpenSSO 8.0 Update 2 ist das Fedlet in folgender Form verfügbar:

- Nachdem Sie die OpenSSO 8.0 Update 2-ZIP-Datei dekomprimiert haben, sind das Java Fedlet und .NET Fedlet in der folgenden Datei verfügbar:

  zip-root`/opensso/fedlet/fedlet-unconfigured.zip`, wobei *zip-root* der Pfad ist, an dem Sie die Datei Oracle OpenSSO 8.0 Update 2 ZIP dekomprimiert haben.

- Wenn Sie Oracle OpenSSO 8.0 Update 2 installiert haben, können Sie das Java Fedlet mithilfe des Arbeitsablaufs Create Fedlet (Fedlet erstellen) unter Common Tasks (Allgemeine Aufgaben) in der OpenSSO 8.0-Administrationskonsole erstellen.

Anforderungen für das Oracle OpenSSO Fedlet

Für das Fedlet gelten die folgenden Anforderungen:

- Von Oracle OpenSSO 8.0 Update 2 unterstützter Webcontainer, wenn Sie beansichtigen, `fedlet.war` oder eine Java-Dienstanbieteranwendung bereitzustellen, die im Fedlet integriert ist. Siehe „Hardware- und Software-Anforderungen für OpenSSO 8.0 Update 2“ auf Seite 12
- Microsoft Internet Information Server (IIS) 7.0 und höher, wenn Sie beabsichtigen, das .NET Fedlet bereitzustellen.
- JDK 1.6.x und höher.

Oracle OpenSSO Fedlet-Konfiguration

In diesem Abschnitt wird beschrieben, wie das Fedlet zuerst mit einer Dienstanbieteranwendung konfiguriert wird:

- „So konfigurieren Sie das Java-Fedlet“ auf Seite 44
- „So konfigurieren Sie das .NET-Fedlet.“ auf Seite 46

Wenn Sie die erste Konfiguration für das Fedlet abgeschlossen haben, fahren Sie mit weiteren Konfigurationen fort, die Sie wünschen. Folgende Aspekte sind zu beachten:

- Wenn Sie die Fedlet-Datei `sp.xml` ändern, müssen Sie diese Datei erneut in den Idenitätsanbieter importieren.
- Wenn Sie weitere Änderungen an der Fedlet-Konfiguration auf Seite des Dienstanbieters vornehmen, teilen Sie diese Informationen dem Identity-Anbieteradministrator mit, damit die erforderlichen Konfigurationsänderungen auf der Seite des Idenitätsanbieters erfolgen können.

▼ So konfigurieren Sie das Java-Fedlet

1 **Generieren Sie auf der Seite des Identity-Anbieters die XML-Metadaten für den Identity-Anbieter und speichern Sie die Metadaten in einer Datei mit dem Namen `idp.xml`.**

Für Oracle OpenSSO 8.0 Update 2 verwenden Sie `exportmetadata.jsp`. Beispiel:

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

2 **Auf der Seite des Dienstanbieters dekomprimieren Sie bei Bedarf die Fedlet-ZIP-Datei.**

3. **Erstellen Sie das Fedlet-Homeverzeichnis. Hierbei handelt es sich um das Verzeichnis, in dem das Fedlet seine Metadaten, seinen Vertrauenskreis und die Konfigurationseigenschaftendateien liest.**

   Die Standardposition ist das Fedlet-Unterverzeichnis unter dem Home-Verzeichnis des Benutzers, der den Fedlet-Webcontainer ausführt (angegeben durch die JVM-Eigenschaft `user.home`). Wenn dieses Home-Verzeichnis beispielsweise `/home/webservd` ist das Fedlet-Homeverzeichnis

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

   Zum Ändern des Standard-Fedlet-Home-Verzeichnisses stellen Sie den Wert der JVM-Laufzeiteigenschaft `com.sun.identity.fedlet.home` auf den gewünschten Speicherort ein. Beispiel:

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

   Das Fedlet liest anschließend seine Metadaten, den Vertrauenskreis und die Konfigurationsdateien aus dem Verzeichnis `/export/fedlet/conf`.

4. **Kopiere die folgenden Dateien aus dem Java-Fedlet-Verzeichnis `java/conf` in das Fedlet-Home-Verzeichnis:**

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

5. **Im Fedlet-Home-Verzeichnis benennst du die kopierten Dateien um und legst für jeden Namen `-template` ab.**

6. **In den Dateien, die du im Fedlet-Home-Verzeichnis kopiert und umbenannt hast, ersetzt du die Tags, wie in der nächsten Tabelle dargestellt:**

Tag	Ersetzen mit
`FEDLET_COT`	Name des Vertrauenskreises (Circle of Trust, COT), dessen Mitglieder der Remote-Identity-Anbieter und die Java-Fedlet-Dienstanbieteranwendung sind.
`FEDLET_ENTITY_ID`	ID (Name) der Java-Fedlet-Dienstanbieteranwendung. Beispiel: `fedletsp`
`FEDLET_PROTOCOL`	Protokoll des Webcontainers für die Java-Fedlet-Dienstanbieteranwendung (z. B. `fedlet.war`). Z. B.: `https`
`FEDLET_HOST`	Hostname des Webcontainers für die Java-Fedlet-Dienstanbieteranwendung (z. B. `fedlet.war`). Z. B.: `fedlet-host.example.com`

Tag	Ersetzen mit
FEDLET_PORT	Anschlussnummer des Webcontainers für die Java-Fedlet-Dienstanbieteranwendung (z. B. `fedlet.war`). Z. B. `80`
FEDLET_DEPLOY_URI	URL der Java-Fedlet-Dienstanbieteranwendung Z. B.: `http://fedletsp.example.com/myFedletApp`
IDP_ENTITY_ID	ID (Name) des Remote-Identity-Anbieters Z. B.: `openssoidp`

Hinweis: Wenn der Fedlet-Dienstanbieter oder die Identity-Anbieterentität-ID ein Prozentzeichen (%) oder Komma (,) enthalten, müssen Sie das Zeichen verlassen, bevor Sie es in der Datei `fedlet.cot` ersetzen. Ändern Sie z. B. ?%“ in ?%25“ und ?,“ in ?%2C“.

7 Kopieren Sie die Datei `FedletConfiguration.properties` aus dem Java-Fedlet-Verzeichnis `java/conf` in das Fedlet-Home-Verzeichnis.

8 Kopieren Sie die Idenitätsanbieter-Standardmetadaten-XML-Datei (aus Schritt 1) in das Fedlet-Home-Verzeichnis. Diese Datei muss mit `idp.xml` bezeichnet werden.

9 Importieren Sie die Java-Fedlet-XML-Metadatendatei `sp.xml`) in den Identity-Anbieter.

Für Oracle OpenSSO 8.0 Update 2 verwenden Sie den Arbeitsablauf Register Remote Service Provider unter Common Tasks in der OpenSSO 8.0-Administrationskonsole, um die Java-Fedlet-Dienstanbietermetadaten zu importieren und den Java-Fedlet-Dienstanbieter in einen Vertrauenskreis aufzunehmen.

Nächste Schritte Abhängig von Ihren Anforderungen fahren Sie mit der weiteren Konfiguration für das Java-Fedlet fort.

▼ So konfigurieren Sie das .NET-Fedlet.

1 Generieren Sie auf der Seite des Identity-Anbieters die XML-Metadaten für den Identity-Anbieter und speichern Sie die Metadaten in einer Datei mit dem Namen `idp.xml`.

Für Oracle OpenSSO 8.0 Update 2 verwenden Sie `exportmetadata.jsp`. Beispiel:

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

2 Auf der Seite des Dienstanbieters dekomprimieren Sie bei Bedarf die Fedlet-ZIP-Datei.

3 Kopieren Sie die folgenden Dateien aus dem .NET-Fedlet-Ordner `asp.net/conf` in den `App_Data`-Ordner Ihrer Anwendung.

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

4. **Im Ordner `App_Data` benennen Sie die Dateien um, die Sie kopiert haben, und legen `-template` für jeden Namen ab.**

5. **In den Dateien, die Sie im Ordner `App_Data` kopiert und umbenannt haben, ersetzen Sie die Tags, wie in der nächsten Tabelle dargestellt.**

Tag	Ersetzen mit
`FEDLET_COT`	Name des Vertrauenskreises (Circle of Trust, COT), dessen Mitglieder der Remote-Identity-Anbieter und die Java-Fedlet-Dienstanbieteranwendung sind.
`FEDLET_ENTITY_ID`	ID (Name) der Java-Fedlet-Dienstanbieteranwendung. Beispiel: `fedletsp`
`FEDLET_DEPLOY_URI`	URL der Java-Fedlet-Dienstanbieteranwendung. Z. B.: `http://fedletsp.example.com/myFedletApp`
`IDP_ENTITY_ID`	ID (Name) des Remote-Identity-Anbieters Z. B.: `openssoidp`

6. **Kopieren Sie die Identity-Anbieter-Standardmetadaten-XML-Datei (aus Schritt 1) in den Ordner `App_Data` Ihrer Anwendung. Diese Datei muss mit `idp.xml` bezeichnet werden.**

7. **Kopieren Sie die Datei `Fedlet.dll` und `Fedlet.dll.config` aus dem .NET-Fedlet-Ordner `asp.net/bin` in den Ordner `bin` der Anwendung.**

8. **Importieren Sie die .NET-Fedlet-XML-Metadatendatei (`sp.xml`) in den Identity-Anbieter.**

   Für Oracle OpenSSO 8.0 Update 2 verwenden Sie den Arbeitsablauf Register Remote Service Provider unter Common Tasks in der OpenSSO 8.0-Administrationskonsole, um die .NET-Fedlet-Dienstanbietermetadaten zu importieren und den .NET-Fedlet-Dienstanbieter in einen Vertrauenskreis aufzunehmen.

Nächste Schritte Abhängig von Ihren Anforderungen fahren Sie mit der weiteren Konfiguration für das .NET-Fedlet fort.

Neue Funktionen für das Fedlet in OpenSSO 8.0 Update 2

Oracle OpenSSO 8.0 Update 2 enthält die folgenden neuen Funktionen für das Fedlet:

Fedlet-Versionsinformationen (CR 6941387)

Das Oracle OpenSSO-Fedlet enthält Versionsinformationen. Wenn Sie die Dateien im Fedlet-Paket (ZIP-Datei) extrahiert haben, ermitteln Sie die Fedlet-Version, indem Sie eine der folgenden Dateien anzeigen:

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

Java-Fedlet-Kennwortverschlüsselung und -entschlüsselung (CR 6930477)

Das Java-Fedlet enthält `fedletEncode.jsp` in der Datei `fedlet.war` zum Verschlüsseln der Kennwörter `storepass` und `keypass`. Für jedes Fedlet wird standardmäßig ein anderer Verschlüsselungsschlüssel generiert. Zum Ändern dieses Verschlüsselungsschlüssels stellen Sie die Eigenschaft `am.encryption.pwd` in der Fedlet-Datei `FederationConfig.properties` ein.

Java-Fedlet-Support zum Signieren und Verschlüsseln

Dieses Java-Fedlet unterstützt die XML-Signaturüberprüfung und Entschlüsselung der verschlüsselten Elemente `assertion` und `NameID` und der entsprechenden Attribute.

▼ So konfigurieren Sie das Java-Fedlet zum Signieren und für die Verschlüsselung:

1. **Erstellen Sie eine Schlüsselspeicherdatei mit dem Namen `keystore.jks` mithilfe des Dienstprogramms `keytool`.**
2. **Fügen Sie den privaten Schlüssel (und gegebenenfalls das öffentliche Zertifikat), der zum Signieren verwendet wird, und den privaten Schlüssel (und gegebenenfalls das öffentliche Zertifikat), der zum Verschlüsseln verwendet wird, in die Datei `keystore.jks` ein.**
3. **Erstellen Sie eine `.storepass`-Datei.**

4. **Fügen Sie das Passwort in die `.storepass`-Datei ein. Zum Verschlüsseln des Kennworts verwenden Sie `fedletEncode.jsp`.**

5. **Erstellen Sie eine `.keypass`-Datei.**

6. **Fügen Sie das Passwort in die Datei `.keypass` ein Zum Verschlüsseln des Kennworts verwenden Sie `fedletEncode.jsp`.**

7. **Wenn Sie Klartextpasswörter verwenden, füllen Sie die folgende Zeile in der Datei `FederationConfig.properties` aus**

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

8. **Legen Sie den vollständigen Pfad für die folgenden Attribute in der Datei `FederationConfig.properties` fest, wobei *path* der vollständige Pfad zur jeweiligen Datei ist:**

   ```
   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. **Verwenden Sie `keytool`, um das Signierzertifikat zu exportieren Beispiel:**

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

   Das Tool fordert Sie auf, das Passwort einzugeben, das zum Zugreifen auf `keystore.jks` verwendet wird und generiert anschließend das Zertifikat.

10. **Wenn Sie ein Verschlüsselungszertifikat benötigen, verwenden Sie `keytool`, um es zu exportieren, wie im vorherigen Schritt dargestellt (Oder verwenden Sie das gleiche Zertifikat zum Signieren und für die Verschlüsselung.)**

11. **Erstellen Sie einen `KeyDescriptor`-XML-Block und fügen Sie das Verschlüsselungszertifikat ein. Notieren Sie sich z B. den Tag `use="signing"` des Elements `KeyDescriptor`.**

    ```
    <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 Erstellen Sie einen weiteren `KeyDescriptor`-XML-Block und fügen Sie das Verschlüsselungszertifikat ein. Notieren Sie sich z. B. den Tag `use="encryption"` des Elements `KeyDescriptor`:

```
<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 Fügen Sie in der Java-Fedlet-Datei `sp.xml` unter dem Element `SPSSODescriptor` die XML-Blöcke mit den Signier- und Verschlüsselungszertifikaten ein. Das Beispiel für ein `SPSSODescriptor`-Element finden Sie unter Beispiel 4–1.

Das Attribut `AuthnRequestsSigned` ist auf `true` festgelegt, sodass das Java-Fedlet alle Authentifizierungsanforderungen konfiguriert.

14 Stellen Sie in der Java-Fedlet-Datei `sp-extended.xml` die Werte für die folgenden Elemente ein.

- `signingCertAlias` enthält das Alias des XML-Signierzertifikats im Schlüsselspeicher.
- `encryptionCertAlias` enthält das Alias des XML-Verschlüsselungszertifikates im Schlüsselspeicher.

15 Zum Umsetzen der vom Java-Fedlet-Dienstanbieter verschlüsselten Daten stellen Sie die folgenden Attribute in der Datei `sp-extended.xml` auf `true` ein.

- `wantAssertionEncrypted`
- `wantNameIDEncrypted`
- `wantAttributeEncrypted`

16 Um die Daten, die der Java-Fedlet-Dienstanbieter verschlüsselt hat und signieren lassen möchte, stellen Sie die folgenden Attribute auf `true` ein.

- `wantAuthnRequestsSigned` in der Datei `idp.xml` teilt dem Fedlet mit, was verschlüssel werden muss.

- `AuthnRequestsSigned` und `WantAssertionsSigned` in der Datei `sp.xml` teilen dem Identity-Anbieter mit, was das Fedlet signieren will.
- `wantArtifactResponseSigned` in der Datei `sp-extended.xml` teilt dem Fedlet mit, was signiert werden soll.
- `wantPOSTResponseSigned` in der Datei `sp-extended.xml` .
- `wantLogoutRequestSigned` in der Datei `sp-extended.xml` .
- `wantLogoutResponseSigned` in der Datei `sp-extended.xml` .

Wenn der Identity-Anbieter für bestimmte Nachrichten eine Signierung benötigt, stellen Sie in der Datei `idp-extended.xml` `true` ein. Beispielsweise `wantLogoutRequestSigned`und `wantLogoutResponseSigned`.

Hinweis – Wenn Sie in der Datei `sp-extended.xml` Attribute festlegen, teilen Sie diese Informationen dem Identity-Anbieteradministrator mit, sodass er die erforderlichen Änderungen im Identity-Anbieter vornehmen kann.

---

17 Starten Sie den Java-Fedlet-Webcontainer erneut.

18 Importieren Sie die Java-Fedlet-Datei `sp.xml` in den Identity-Anbieter.

Beispiel 4–1 Java-Fedlet-Musterelement `SPSSODescriptor`.

```
<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-Support für Attributabfrage (CR 6930476)

Das Java-Fedlet unterstützt die SAMLv2-Attributabfrage zum Abfragen eines Identity-Anbieters, z. B. Oracle OpenSSO 8.0 Update 2, auf spezifische Identitätsattributwerte. Sie können das Fedlet zum Signieren der Abfrage und Verschlüsseln der Abfrage konfigurieren. Das Signieren ist zum Ausgeben einer Fedlet-Abfrage erforderlich, die Verschlüsselung ist jedoch optional.

▼ So konfigurieren Sie das Java-Fedlet zur Attributsabfrage:

1. **Aktivieren Sie die XML-Signierung zum Signieren der Attributsabfrage, wie beschrieben in „Java-Fedlet-Support zum Signieren und Verschlüsseln“ auf Seite 48.**

2. **Fügen Sie das im vorherigen Schritt generierte Zertifikat in das Element `RoleDescriptor` in der Fedlet-Datei `sp.xml`. Im folgenden Beispiel werden zwei `KeyDescriptor`-Tags dargestellt, in die Sie das Zertifikat einfügen. Eines wird zum Signieren verwendet, das andere zum Verschlüsseln. Wenn Sie die Verschlüsselung nicht aktivieren, ist der `KeyDescriptor use="encryption" tag` nicht erforderlich.**

```
<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 In der Java-Fedlet-Datei `sp-extended.xml` geben Sie den Wert für das Attribut `signingCertAlias` an und (wenn entsprechend konfiguriert) für das Attribut `encryptionCertAlias` an.

Wenn Sie beabsichtigen, den Identity-Anbieter zum Verschlüsseln der Behauptung zu verwenden, verschlüsseln Sie auch das Element `NameID`. Daher muss der Wert des Attributs `wantNameIDEncrypted` auf `true` eingestellt sein. Fügen Sie den XML-Code in das Element `AttributeQueryConfig` ein. Beispiel:

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

In diesem Beispiel ist `test` das Alias für den Musterschlüssel.

4 Importieren Sie die Java-Fedlet-Metadatendatei (`sp.xml`) in den Identity-Anbieter.

Führen Sie auch die zusätzlichen Konfigurationsschritte im Identity-Anbieter aus, um die Attributabfrage für das Fedlet zu unterstützen.

.NET-Fedlet-Verschlüsselung und Entschlüsselung von Abfragen und Antworten (CR 6939005)

Das .NET-Fedlet kann ausgehende XML-Abfragen verschlüsseln und eingehende Antworten auf die Elemente NameID, Attribut und Behauptung abfragen.

▼ So konfigurieren Sie das .NET-Fedlet zum Verschlüsseln und Entschlüsseln von Anforderungen und Antworten:

1. **Importieren Sie mithilfe des Snap-Ins Certificates (Zertifikate) für die Microsoft Management Console das X.509-Zertifikat in den Ordner Personal (Persönlich) im Konto Local Computer (Lokaler Computer). Zur Verwendung dieses Snap-Ins beachten Sie den folgenden Microsoft-Artikel:**

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

2. **Geben Sie einen Anzeigenamen für dieses Zertifikat an, indem Sie den Dialog Eigenschaften aufrufen und einen Wert eingeben. (Speichern Sie diesen Wert für Schritt 4.)**

3. **Stellen Sie die entsprechenden Genehmigungen ein, um den Lesezugriff auf das Zertifikat für das Benutzerkonto zuzulassen, das vom Internet Information Server (IIS) verwendet wird, wie im Microsoft-Artikel beschrieben. Beispiel:**

   a. **Im Snap-In Certificates navigieren Sie zu Action (Aktion), All Tasks (Alle Aufgaben) und anschließend zu Manage Private Keys (Private Schlüssel verwalten).**

   b. **Geben Sie Leseberechtigungen für das Benutzerkonto an, die IIS ausführen (normalerweise NETWORK SERVICE).**

4. **Geben Sie in der erweiterten Metadatendatei (`sp.xml`) des .NET-Fedlets den Anzeigenamen an, der in Schritt 2 als Wert für das Attribut `encryptionCertAlias` angegeben ist. Beispiel:**

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

5. **Fügen Sie in der Dienstanbieter-Metadatendatei (`sp.xml`) des .NET-Fedlets den `KeyDescriptor` für den Verschlüsselungsschlüssel ein.**

   Verwenden Sie das Snap-In Certificates für die Microsoft Management Console, das Sie zuvor verwendet haben, um den öffentlichen Schlüssel Ihres Zertifikates in Base64-Codierung im XML-Block `KeyDescriptor` einzubeziehen. Dieser `KeyDescriptor` muss das erste untergeordnete Element im `SPSSODescriptor` sein. Beispiel:

   ```
   <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 Starten Sie den Anwendungspool erneut, der mit Ihrer .NET-Anwendung verknüpft ist.

Nächste Schritte Verwenden Sie die Musteranwendung, um diese Konfiguration zu testen. Legen Sie auch die folgenden Attribute fest, um mit dem Identity-Anbieter und den entsprechenden Änderungen an den konfigurierten Metadaten Abfragen zu verschlüsseln und Antworten zu entschlüsseln.

- Behauptung: Stellen Sie das Attribut `wantAssertionEncrypted` in der Metadatendatei `sp-extended.xml` auf `true` ein, damit das .NET-Fedlet das Element `EncryptedAssertion` in eingehenden Antworten des Identity-Anbieters entschlüsseln kann.
- Attribut: Stellen Sie das Attribut `wantAttributeEncrypted` in der Metadatendatei `sp-extended.xml` auf `true` ein, damit das .NET-Fedlet das Element `EncryptedAttribute` in eingehenden Antworten vom Identity-Anbieter entschlüsseln kann.
- NameID: Stellen Sie das Attribut `wantNameIDEncrypted` in der Metadatendatei `idp-extended.xml` auf `true` ein, damit das .NET-Fedlet das Element `NameID` in ausgehenden Anforderungen verschlüsseln kann. Stellen Sie das gleiche Attribut in `sp-extended.xml` ein, damit das .NET-Fedlet das Element `EncryptedID` in eingehenden Antworten vom Identity-Anbieter entschlüsselt.

.NET-Fedlet-Signieren von Abfragen und Antworten (CR 6928530)

Das .NET-Fedlet unterstützt das Signieren ausgehender XML-Anforderungen, z. B. `Authn`-Anforderungen und Abmeldungsanforderungen.

▼ So konfigurieren Sie das .NET-Fedlet zum Signieren von Anforderungen und Antworten:

1. **Importieren Sie mithilfe des Snap-Ins Certificates (Zertifikate) für die Microsoft Management Console das X.509-Zertifikat in den Ordner Personal (Persönlich) im Konto Local Computer (Lokaler Computer). Zur Verwendung dieses Snap-Ins beachten Sie den folgenden Microsoft-Artikel:**

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

2. **Geben Sie einen Anzeigenamen für dieses Zertifikat an, indem Sie den Dialog Eigenschaften aufrufen und einen Wert eingeben. (Speichern Sie diesen Wert für Schritt 4.)**

3. **Stellen Sie die entsprechenden Genehmigungen ein, um den Lesezugriff auf das Zertifikat für das Benutzerkonto zuzulassen, das vom Internet Information Server (IIS) verwendet wird, wie im Microsoft-Artikel beschrieben. Beispiel:**

   a. **Im Snap-In Certificates navigieren Sie zu Action (Aktion), All Tasks (Alle Aufgaben) und anschließend zu Manage Private Keys (Private Schlüssel verwalten).**

   b. **Geben Sie Leseberechtigungen für das Benutzerkonto an, die IIS ausführen (normalerweise NETWORK SERVICE).**

4. **Geben Sie in der erweiterten Metadatendatei des .NET-Fedlets (`sp-extended.xml`) den Anzeigenamen an, den Sie in Schritt 2 als Wert für das Attribut `signingCertAlias` angegeben haben. Beispiel:**

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

5. **Fügen Sie in der Dienstanbieter-Metadatendatei des .NET Fedlets (`sp.xml`) den `KeyDescriptor` für den Signierschlüssel ein.**

   Verwenden Sie das Snap-In Certificates für die Microsoft Management Console, das Sie zuvor verwendet haben, um den öffentlichen Schlüssel Ihres Zertifikates in Base64-Codierung im XML-Block `KeyDescriptor` einzubeziehen. Dieser `KeyDescriptor` muss das erste untergeordnete Element im `SPSSODescriptor` sein. Beispiel:

   ```
   <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 Starten Sie den Anwendungspool erneut, der mit Ihrer .NET-Anwendung verknüpft ist.

.NET-Fedlelt-Einzelabmeldung (CR 6928528 und CR 6930472)

Das .NET-Fedlet unterstützt sowohl vom Identity-Anbieter initiierte und vom Dienstanbieter initiierte einmalige Abmeldungen. Zum Implementieren einer einmaligen Abmeldung enthält die .NET-Fedlet-Musteranwendung die Datei `logout.aspx` und `spinitiatedslo.aspx` im Ordner `asp.net/SampleApp`. Um anzuzeigen, wie die Funktion zur einmaligen Abmeldung des Fedlets funktioniert, stellen Sie die .NET-Fedlet-Musteranwendung bereit.

▼ So konfigurieren Sie eine .NET-Fedlet-Service Provider-Anwendung für eine einmalige Abmeldung:

1 Wenn Sie das .NET-Fedlet nicht konfiguriert haben, befolgen Sie die Schritte in der Datei `Readme`.

2 Kopieren Sie die Datei `logout.aspx` und `spinitiatedslo.aspx` in den öffentlichen Inhalt Ihrer .NET-Anwendung.

3 Führen Sie diese Änderungen an den Konfigurationsdateien für Ihre Anwendung aus.

- Achten Sie darauf, dass der Pfad zur Datei `logout.aspx` in der Datei `sp.xml` auf die korrekte Position der Datei für Ihre Anwendung verweist.
- Achten Sie darauf, dass der Pfad zur Datei `spinitiatedslo.aspx` in der Datei `idp.xml` auf die korrekte Position der Datei für Ihre Anwendung verweist.

4 Wenn Sie möchten, dass die Abmeldungsanforderung und Abmeldungsantwort signiert werden, stellen Sie die folgenden Attribute in der Datei `sp-extended.xml` und `idp-extended.xml` auf `true` ein.

- `wantLogoutRequestSigned`
- `wantLogoutResponseSigned`

5 **Importieren Sie die Fedlet-Service Provider-Metadatendatei (`sp.xml`) in den Identity-Anbieter.**

Informieren Sie auch die Identity-Anbieteradministrator, dass Sie die einmalige Abmeldung für den Fedlet-Dienstanbieter konfiguriert haben, sodass alle erforderlichen Änderungen an der Konfiguration des Identity-Anbieters stattfinden können.

.NET-Fedlet-Dienstanbieter-initiierte einmalige Anmeldung (CR 6928525)

Das .NET-Fedlet unterstützt die vom SAMLv2-Dienstanbieter initiierte einmalige Anmeldung (Single Sign-On, SSO). Darüber hinaus wird Artefakt-Support benötigt, damit das .NET-Fedlet ein Artefakt erhalten kann, das anschließend von SOAP mit dem Artifact Resolution Service des ausgebenden Identity-Anbieters aufgelöst wird.

Die .NET-Fedlet-Musteranwendung zeigt, wie Sie die einmalige Anmeldung konfigurieren können. Wenn in Ihrer Anwendung die erforderlichen Artefakte installiert sind, ist eine spezifische IRL erforderlich, um den HTTP POST mit der SAMLv2-Reaktion nach der erfolgreichen Authentifizierung durch den Identity-Anbieter zu erhalten. Im folgenden Codebeispiel wird angezeigt, wie Sie diese Informationen in einer .NET-Anwendung abrufen können.

BEISPIEL 4–2 Codebeispiel zum Abrufen der `AuthnResponse` in einer .NET-Fedlet-Anwendung.

```
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)
}
```

Wenn Ihre Anwendung die SAMLv2-Reaktion erhält, wird das `authnResponse`-Objekt mit den Behauptungsinformationen gefüllt. Die Musteranwendung zeigt an, wie Sie die Attribute und Betreffinformationen von diesem Objekt abrufen.

.NET-Fedlet-Support für mehrere Identity-Anbieter und Erkennungsdienst (CR 6928524)

Das .NET-Fedlet unterstützt mehrere Identity-Anbieter und den Erkennungsdienst des Identity-Anbieters.

In einigen Bereitstellungen empfiehlt es sich, das .NET-Fedlet mit mehreren Identity-Anbietern zu konfigurieren, z. B. Oracle OpenSSO 8.0 Update 2. Führen Sie die folgenden Aufgaben für jeden zusätzlichen Identity-Anbieter aus, den Sie hinzufügen möchten.

▼ So konfigurieren Sie das .NET-Fedlet für mehrere Identity-Anbieter:

1. **Rufen Sie die XML-Metadatendatei vom zusätzlichen Identity-Anbieter ab.**

2. **Benennen Sie die zusätzliche Identity-Anbieter-Metadatendatei als `idp`*n*`.xml`, wobei *n* der Identity-Anbieter ist, den Sie hinzufügen. Benennen Sie die zweite Identity-Anbieterdatei als `idp2.xml`, die dritte als `idp3.xml` usw. In diesem Verfahren wird `idp2.xml` als Dateiname verwendet.**

3. **Kopieren Sie die Datei `idp2.xml` aus Schritt 2 in den Ordner `App_Data` Ihrer Anwendung.**

4. **Fügen Sie diesen neuen Identity-Anbieter in den .NET-Fedlet-Vertrauenskreis hinzu.**

   So fügen Sie den neuen Identity-Anbieter in einen bestehenden Vertrauenskreis hinzu:

   In der Datei `fedlet.cot` im Ordner der Anwendung `App_Data` hängen Sie die neue IDP-Entitäts-ID (gekennzeichnet durch das Attribut `entityID` in der Metadendatei `idp2.xml`) an den Wert des Attributs `sun-fm-trusted-providers`an und verwenden Sie dabei ein Komma (,) als Trennzeichen.

   So fügen Sie den neuen Identity-Anbieter in einen bestehenden Vertrauenskreis hinzu:

   a. **Erstellen Sie eine neue Datei mit dem Namen `fedlet2.cot` im Ordner `App_Data` Ihrer Anwendung. Verwenden Sie die bestehende Datei `fedlet.cot` als Vorlage, aber ändern Sie den Wert des Attributs `cot-name` in den Namen des neuen Vertrauenskreises (z. B. `cot2`). Verwenden Sie die neue Entitäts-ID des Identity-Anbieters und die Fedlet-Entitäts-ID als Wert für das Attribut `sun-fm-trusted-providers` und trennen Sie die beiden Entitäts-IDs mit einem Komma (,).**

   b. **Fügen Sie in der Datei `sp-extended.xml` den neuen Namen des Vertrauenskreises in den Wert des Attributs `cotlist` ein. Beispielsweise für einen Vertrauenskreis mit dem Namen `cot2`.**

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

5. **Erstellen Sie im Ordner `App_Data` Ihrer Anwendung die neue Datei `idp2-extended.xml` als erweiterte Metadaten für den neuen Identity-Anbieter ein. Verwenden Sie die bestehende Datei `idp-extended.xml` als eine Vorlage, aber ändern Sie die `entityID` in die neue Entitäts-ID des Identity-Anbieters. Ändern Sie den Wert für das Attribut `cotlist` in den**

Vertrauenskreisnamen, wenn ein neuer Vertrauenskreis für den Identity-Anbieter erstellt wird. Achten Sie darauf, dass es sich bei dem zusätzlichen Identity-Anbieter um eine Remote-Identity handelt.

6 **Starten Sie den Anwendungspool erneut, der mit Ihrer Fedlet-.NET-Anwendung verknüpft ist.**

7 **Die Fedlet-Metadaten-XML-Datei (`sp.xml`) muss in den zusätzlichen Identity-Anbieter importiert und in den gleichen Vertrauenskreis wie die Identity-Anbieterentität hinzugefügt werden. Importieren Sie die Datei `sp.xml` in den Identity-Anbieter oder übergeben Sie die Datei Ihrem Identity-Anbieter zum Importieren.**

.NET-Fedlet-Support für den Identity-Anbieter-Erkennungsdienst (CR 6928524)

In diesem Szenario wird das .NET-Fedlet wird mehreren Idenitätsanbietern in einem Vertrauenskreis konfiguriert und Sie können das Fedlet dazu konfigurieren, den Identity-Anbieter-Erkennungsdienst zum Ermitteln des bevorzugten Identity-Anbieters zu verwenden.

Der Erkennungsdienst muss für die Identity-Anbieter konfiguriert werden, die Sie mit dem .NET-Fedlet verwenden. Informationen zum Konfigurieren des Identity-Anbieter-Erkennungsdienstes in Oracle OpenSSO 8.0 Update 2 finden Sie in der folgenden Dokumentensammlung: `http://docs.sun.com/coll/1767.1`.

▼ So konfigurieren Sie das.NET-Fedlet für die Verwendung des Identity-Anbieter-Erkennungsdienstes:

1 **Stellen Sie in der .NET-Fedlet-Datei `fedlet.cot` die Eigenschaft `sun-fm-saml2-readerservice-url` auf die SAMLv2-Leserdienst-URL ein. Beispiel:**

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

2 **Starten Sie den Anwendungspool erneut, der mit Ihrer .NET-Fedlet-Anwendung verknüpft ist.**

Allgemeine Probleme und Problemumgehungen für das Oracle OpenSSO Fedlet

Wird noch entwickelt

Dokumentations-Errata

Die Referenz für Fedlet Java API ist in der Oracle OpenSSO 8.0 Update 2-Java-API-Referenz in der folgenden Dokumentationssammlung verfügbar_ `http://docs.sun.com/coll/1767.1`.

Hinweis – Die Methode `getPolicyDecisionForFedlet` wird in der OpenSSO 8.0 Update 2-Version nicht unterstützt.

◆ ◆ ◆ KAPITEL 5

Integrieren von OpenSSO 8.0 Update 2 in Oracle Access Manager

Dieses Kapitel enthält Anweisungen zum Implementieren einer einmaligen Anmeldung mit OpenSSO 8.0 Update 2 und Oracle Access Manager 10g oder 11g. Diese Informationen ergänzen die Konzeptinformationen in Kapitel 3, „Integrating Oracle Access Manager“ in *Sun OpenSSO Enterprise 8.0 Integration Guide*. In diesem Fall wird eine einmalige Abmeldung auf OpenSSO-geschützten Anwendungen mittels einer Oracle Access Manager-Sitzung dargestellt. Das konfigurierte OpenSSO-Authentifizierungsmodul generiert eine OpenSSO-Sitzung auf Basis der Oracle Access Manager-Sitzung.

Übersicht zu den Integrationsschritten

1. „Vorbereitung“ auf Seite 63
2. Entpacken der Integrations-Bits
3. Erstellen für Quelldateien in Oracle Access Manager in OpenSSO
4. „(Optional) Erstellen Sie ein Authentifizierungsschema für OpenSSO in Oracle Access Manager.“ auf Seite 67
5. „Konfigurieren des einmaligen Anmeldens mit Oracle Access Manager und Oracle OpenSSO STS “ auf Seite 68
6. „So testen Sie die einmalige Anmeldung:“ auf Seite 71
7. „(Optional) Installieren des Oblix-Authentifizierungsschema in Oracle Access Manager“ auf Seite 71

Vorbereitung

Stellen Sie sicher, dass Sie Zugriff auf die folgenden Komponenten haben, bevor Sie OpenSSO 8.0 Update 2 zur Integration mit Oracle Access Manager installieren.

opensso.zip	Die ZIP-Datei enthält die Datei opensso.war, den Integrationsquellcode, Konfigurationsdateien

	und andere Tools, die für die Installation und Konfiguration von OpenSSO 8.0 Update 2 benötigt werden.
OpenSSO Agent	Der OpenSSO Agent wird verwendet, wenn eine von OpenSSO geschützte Anwendung die Authentifizierungssitzung verwenden kann, die Oracle Access Manager hergestellt hat.
Oracle Access Manager 10g oder 11g	Laden Sie Oracle Access Manager von der Oracle-Website herunter. Weitere Informationen finden Sie auf der Seite Oracle Fusion Middleware 11gR1 Software Downloads.
Oracle Web Gate 10g oder 11g	Laden Sie Oracle Webgate für einen Container herunter, der von OpenSSO und Oracle Webgate unterstützt wird. Zurzeit ist Sun Web Server 7.x der einzige Container, der von beiden Produkten unterstützt wird. Beachten Sie die Seite Oracle Fusion Middleware 11gR1 Software Downloads.
Oracle Access Manager SDK 10g oder 11g	Laden Sie Oracle Access Manager herunter SDK wird zum Kompilieren und Erstellen der OpenSSO-Authentifizierungsmodule für die Oracle Access Manager-Integration benötigt. Beachten Sie die Seite Oracle Fusion Middleware 11gR1 Software Downloads.
OpenSSO C-SDK 2.2	(Optional) OpenSSO C-SDK wird zum Erstellen eines Authentifizierungsmoduls in Oracle Access Manager verwendet, um eine OAM-Sitzung zu generieren. In OpenSSO tritt dieser Nutzungsfall nicht häufig auf. Siehe „Where is the C SDK?“ in *Sun OpenSSO Enterprise 8.0 C API Reference for Application and Web Policy Agent Developers*

Entpacken der Integrations-Bits

Das Verzeichnis `opensso/integrations/oracle` enthält Quelldaten und Konfigurationen zum Kompilieren und Erstellen von benutzerdefinierten Authentifizierungsmodulen und anderen Plugins. Weitere Informationen zum Verwenden von Optionen in Nutzungsfällen und verwandte Informationen finden Sie unter Kapitel 3, „Integrating Oracle Access Manager“ in

Sun OpenSSO Enterprise 8.0 Integration Guide. In der folgenden Tabelle werden die Dateien unter dem Verzeichnis `opensso/integrations/oracle` und Beschreibungen für jede Datei zusammengefasst.

`README.html`	Hierbei handelt es sich um die Datei, die Sie jetzt lesen.
`build.xml`	Eine Ant-Builddatei zum Erstellen eines benutzerdefinierten Authentifizierungsmoduls für Oracle Access Manager in OpenSSO.
`Konfig`	Konfigurationsdateien, die zum Erstellen eines Authentifizierungsmoduls für Oracle Access Manager in OpenSSO benötigt werden. ■ `OblixAuthService.xml` Authentifizierungsdienstdateien für das Oracle Access Manager-Authentifizierungsmodul ■ `OblixAuthModule.xml` Authentifizierungsmodul-Rückmeldungen für Oracle Access Manager. Dies ist standardmäßig eine leere Datei, die jedoch zu Konfigurationszwecken vorhanden sein muss. ■ `OblixAuth.properties` Eigenschaftendatei, in der Internationalisierungsschlüssel für die Authentifizierung gespeichert sind.
`lib`	Dieses Verzeichnis ist standardmäßig leer. Das Verzeichnis `lib` muss die folgenden Bibliotheken enthalten, um die Quelldateien zu kompilieren. ■ `jobaccess.jar` Kopieren Sie diese Datei aus Oracle Access Manager SDK. ■ `openfedlib.jar`, `amserver.jar` und `opensso-sharedlib.jar` Kopieren Sie diese Dateien aus opensso.war. ■ `servlet.jar oder javaee.jar` Kopieren Sie das GlassFisch-Verzeichnis `lib`. Im Idealfall ist jede JAR-Datei mit standardmäßigen Java EE-Klasse, z. B. `javax.servlet.http.Cookie` geeignet.
`source`	Verzeichnis, das die folgenden Quelldateien enthält:

- `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`

  Diese Klasse ist ein SAML2-Pluginadapter für SAML-Dienstanbieter. Diese Klasse übernimmt die Remote-Authentifizierung in Oracle Access Manager mithilfe des OpenSSO-Sitzungsdienstes.

oamauth (optional) Dieses Verzeichnis enthält Quelldateien für das Oblix-Authentifizierungsschema für OpenSSO. Dies ist ein C-basiertes Authentifizierungsmodul, das OpenSSO C-SDK zur Validierung nutzt.

- oam/solaris/authn_api.c

  Diese Datei implementiert das Oblix-Authentifizierungsschema für OpenSSO.

- oam/solaris/include/*.h

  Alle Kopfzeilendateien, die zum Kompilieren des Authentifizierungsschemas benötigt werden.

- oam/solaris/AMAgent.properties

  Muster-OpenSSO-Agent-Konfigurationsdatei. Das Authentifizierungsschema benötigt sie zum Validieren der OpenSSO-Sitzung.

Erstellen von Quelldateien für Oracle Access Manager in OpenSSO

Verwenden Sie den Ant-Skript zum Erstellen der Quelldateien. Im PFAD muss ein kompatibler Ant-Skript installiert und konfiguriert werden.

▼ So erstellen Sie die Quelldateien für Oracle Access Manager:

1. **Führen Sie folgenden Befehl aus:**

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

   Mit diesem Befehl werden Quelldateien erstellt und wird `fam_oam_integration.jar` im Verzeichnis `$openssozipdir/integrations/oracle/dist` integriert.

2. **Bündeln Sie das Authentifizierungsmodul in die OpenSSO WAR-Datei.**

   a. **Erstellen Sie ein temporäres Verzeichnis und dekomprimieren Sie die Datei `opensso.war`. Beispiel:**

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

      Ab jetzt wird `/export/tmp` als ein WAR-Staging-Bereich verwendet und wird mit dem Makro `$WAR_DIR` dargestellt.

   b. **Kopieren Sie `$openssozipdir/integrations/oracle/dist/fam_oam_integration.jar` nach `$WAR_DIR/WEB-INF/lib`.**

   c. **Kopieren Sie `$openssozipdir/integrations/oracle/config/OblixAuth.properties` nach `$WAR_DIR/WEB-INF/classes`.**

   d. **Kopieren Sie `$openssozipdir/integrations/oracle/config/OblixAuthModule.xml` nach `$WAR_DIR/config/auth/default` und in das Verzeichnis `$WAR_DIR/config/auth/default_en`.**

   e. **Dekomprimieren Sie `opensso.war` mithilfe von `jar cvf opensso.war` in `$WAR_DIR`.**

   Beispiel-TBD

(Optional) Erstellen Sie ein Authentifizierungsschema für OpenSSO in Oracle Access Manager.

Hinweis: Dieser Nutzungsfall tritt nicht häufig auf. Sie müssen dieses Schema nur erstellen, wenn es benötigt wird, z. B. im Nutzungsfall für einen SAML2-Dienstanbieter.

Zum Erstellen des Oblix-Authentifizierungsschemas müssen Sie die `makefile` anpassen. Da es sich um ein C-gestütztes Authentifizierungsmodul handelt, arbeitet es systemabhängig.

▼ So erstellen Sie ein Authentifizierungsschema für OpenSSO in Oracle Access Manager:

Bevor Sie beginnen Die Authentifizierungsschemadateien befinden sich im Verzeichnis `$openssozipdir/integrations/oracle/oamauth/solaris`.

1. **Laden Sie die OpenSSO C-SDK 2.2-Version herunter und konfigurieren Sie sie.**

   Die Datei `authn_api.c` enthält einen Verweis zur Datei `AMAgent.propeties`. Bearbeiten Sie die Datei entsprechend.

2. **Passen Sie `makefile` an Ihre Umgebung an.**

   Geben Sie beispielsweise die Kompilierposition `gcc` an. Bearbeiten Sie auch die `LDFLAGGS`, sodass sie auf das Verzeichnis `OpenSSO C-SDK` verweist.

3. **Führen Sie den Befehl `make` aus.**

   Der Befehl `make` muss eine `authn_api.so`-Datei ergeben.

Konfigurieren des einmaligen Anmeldens mit Oracle Access Manager und Oracle OpenSSO STS

▼ So konfigurieren Sie die einmalige Anmeldung mithilfe von Oracle Access Manager und Oracle OpenSSO 8.0 Update 2:

Vorbereitung: Sun Java System Web Server 7.x muss bereits installiert und konfiguriert sein. Anweisungen zur Web Server-Installation finden Sie unter Sun Java System Web Server Documentation Wiki .

1. **Installieren Sie OpenSSO auf Sun Java System Web Server 7.x.**

2. **Installieren Sie einen OpenSSO-Richtlinienagenten auf einem unterstützten Container und konfigurieren Sie den Agenten für den Betrieb mit OpenSSO.**

   Installationsanweisungen finden Sie in *Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for J2EE Agents* undr *Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for Web AgentsSun OpenSSO Enterprise Policy Agent 3.0 User's Guide for Web Agents*.

3. **Installieren und konfigurieren Sie Oracle Access Manager.**

   Siehe das *Oracle Access Manager-Installationshandbuch 10g (10.1.4.3)*

4. **Installieren und konfigurieren Sie Oracle Access Manager SDK mit Oracle Access Manager.**

   Siehe das *Oracle Access Manager-Installationshandbuch 10g (10.1.4.3)*

5. **Installieren Sie Oracle Webgate im gleichen Webcontainer, in dem auch der OpenSSO-Server installiert ist. (Sun Web Server 7.x)**

   Konfigurieren Sie OpenSSO, sodass es nur `deployURI/UI/*` der OpenSSO-Webanwendung schützt. Beispiel:`/opensso/UI/.../*`

   Oracle Access Manager-Richtlinien, Ressourcen und andere Konfigurationsangaben finden Sie im Oracle Access Manager-Administrationshandbuch. Heben Sie den Schutz jeder anderen URL in OpenSSO Enterprise auf. Dieser Vorgang ist für ein einmaliges Integrationsszenario einer einmaligen Anmeldung bestimmt, aber bewerten Sie die Richtlinien auf Basis einer vollständigen Integration und anderer Bereitstellungsabhängigkeiten.

6. **Konfigurieren Sie das Authentifizierungsmodul in OpenSSO.**

   a. **Greifen Sie auf die OpenSSO-Konsole zu.**

      Der Browser leitet zur Authentifizierung an Oracle Access Manager um. Nach der erfolgreichen Authentifizierung präsentiert OpenSSO eine Anmeldeseite. Melden Sie sich mit dem OpenSSO-Administrator-Benutzernamen und -Kennwort an.

   b. **Importieren Sie die Oracle-Authentifizierungsmoduldienst-XML-Datei in die OpenSSO-Konfiguration.**

      Der Authentifizierungsmoduldienst kann über das Befehlszeilen-Dienstprogramm `ssoadm` und die webgestützte ssoadm.jsp geladen werden.

   c. **Greifen Sie auf `http://host:port/opensso/ssoadm.jsp` zu.**

   d. **Wählen Sie die Option create-service aus.**

   e. **Kopieren Sie die XML-Datei aus `$openssozipdir/integrations/oracle/config/OblixAuthService.xml` , fügen Sie sie ein und klicken Sie auf Submit (Übermitteln).**

      Damit wird der Authentifizierungsmoduldienst in die OpenSSO-Konfiguration geladen.

f. Registrieren Sie das Authentifizierungsmodul im Authentifizierungs-Kerndienst.

Der Kerndienst enthält eine Liste mit Authentifizierern. Wählen Sie die Option `register-auth-module` in `http://host:port/opensso/ssoadm.jsp` . Geben Sie `com.sun.identity.authentication.oblix.OblixAuthModule` als den Authentifizierungsmodul-Klassennamen ein.

g. Verifizieren Sie, dass das Authentifizierungsmodul im Standardbereich registriert ist.

Greifen Sie über die URL http://host:port/opensso auf OpenSSO zu. Klicken Sie in der OpenSSO-Konsole auf den Standardbereich und anschließend auf die Registerkarte Authentication (Authentifizierung). Klicken Sie auf New (Neu), um ein neues Authentifizierungsmodul mit dem Namen OblixAuth zu erstellen.

h. Wählen Sie in der Registerkarte Authentication das OblixAuth-Authentifizierungsmodul aus.

Konfigurieren Sie das Verzeichnis Oblix SDK. Aktivieren Sie Enable Check Remote User Header Only und geben Sie den Remote-Kopfzeilennamen mit `OAM_REMOTE_USER` an. Dieser Parameter kann auf Basis der Bereitstellung konfiguriert werden.

7 (Optional) Aktivieren Sie die Option Ignore Profile (Profil ignorieren) im OpenSSO-Kernauthentifizierungsdienst.

Wechseln Sie in der OpenSSO-Konsole zu Configuration > Core > Realm Attributes > User Profile (Konfiguration > Kern > Bereichsattribute > Benutzerprofil). Wählen Sie Ignored (Ignoriert) aus und klicken Sie auf Save (Speichern).

Mit dieser Konfiguration wird verhindert, dass OpenSSO nach der erfolgreichen Authentifizierung ein bestehendes Benutzerprofil sucht. Wenn das von OpenSSO und Oracle Access Manager verwendete Benutzer-Repository genau übereinstimmen, ist dieser Schritt nicht erforderlich. Wechseln Sie zu Configuration > Core > Realm Attributes > User Profile (Konfiguration > Kern > Bereichsattribute > Benutzerprofil). Wählen Sie Ignored (Ignoriert) aus und klicken Sie auf Save (Speichern).

8 Bearbeiten Sie den Webserver-Startskript, sodass er die gemeinsam genutzten Bibliotheken von Oracle Access Manager SDK enthält.

Aktualisieren Sie `LD_LIBRARY_PATH` im Skript `startserv`, sodass er die gemeinsam genutzten Bibliotheken aus `$ACCESSDKDIR/oblix/lib` enthält.

9 Starten Sie den Sun Web Server erneut, der sowohl OpenSSO als auch Oracle Webgate enthält.

10 Aktualisieren Sie die Anmelde-URL für Weg Agent-Werte wie `http://openssohost:openssoport/deployURI/UI/Login?module=OblixAuth`.

So testen Sie die einmalige Anmeldung:

Greifen Sie über die OpenSSO-geschützte Anwendung auf die geschützte Ressource zu. Der Browser muss sie zur Oracle Access Manager-Anmeldeseite weiterleiten, wenn sie nicht bereits authentifiziert sind. Nach der erfolgreichen Anmeldung erstellt es eine OpenSSO-Sitzung und leitet zuletzt an die vom Policy Agent geschützte Anwendungs-URL zurück. Sie können auf Basis der Richtlinie den Zugriff auf die geschützte Anwendung zulassen oder ablehnen.

(Optional) Installieren des Oblix-Authentifizierungsschema in Oracle Access Manager

Dieser Vorgang empfiehlt sich, wenn die Oracle Access Manager-Sitzung bei Validieren der OpenSSO-Sitzung generiert werden muss. Weitere Informationen zu relevanten Nutzungsfällen finden Sie in Kapitel 3, „Integrating Oracle Access Manager“ in *Sun OpenSSO Enterprise 8.0 Integration Guide*.

Die Oblix-Authentifizierungsschemata werden als C-Authentifizierungsmodule dargestellt. Dieses Authentifizierungsschema verwendet die OpenSSO C-SDK 2.2-Version zum Validieren der OpenSSO-Sitzung. Das OpenSSO-Authentifizierungsschema in Oblix verwendet eine Konfiguration für die OpenSSO-Konfiguration auf Client-Seite in `AMAgent.properties`. Diese Datei muss vor dem Konfigurieren des Authentifizierungsmoduls angepasst werden. In den Erstellanweisungen wird der Speicherort dieser Datei angegeben. Die kompilierten `authn_api.so`- und anderen C-SDK-Bibliotheken müssen in das Verzeichnis `$OAM_INSTALL_DIR/access/oblix/lib` kopiert werden, bevor das Authentifizierungsschema konfiguriert wird. Im *Sun OpenSSO 8.0-Integrationshandbuch* wird ein Beispielbildschirm dargestellt, in dem veranschaulicht wird, wie das Oracle-Authentifizierungsschema konfiguriert wird. Er dient nur als Referenz. Weitere Details finden Sie in der neuesten Oracle Access Manager-Dokumentation.

Integrieren von OpenSSO 8.0 Update 2 in Oracle Access Manager

Dieser Abschnitt enthält Anweisungen zum Implementieren einer einmaligen Anmeldung mit OpenSSO 8.0 Update 2 und Oracle Access Manager Version 10.1.4.0.1 und 11g. Diese Informationen ergänzen die Konzeptinformationen in Kapitel 3, „Integrating Oracle Access Manager“ in *Sun OpenSSO Enterprise 8.0 Integration Guide*. In diesem Fall wird eine einmalige Abmeldung auf OpenSSO-geschützten Anwendungen mittels einer Oracle Access Manager-Sitzung dargestellt. Das konfigurierte OpenSSO-Authentifizierungsmodul generiert eine OpenSSO-Sitzung auf Basis der Oracle Access Manager-Sitzung.