Notes de version de la mise à jour 2 de OpenSSO d'Oracle®

Beta

Référence : 821–2837
Juillet 2010

ORACLE®

101013@24999

Table des matières

Préface

Les notes de version de la mise à jour 2 de OpenSSO 8.0 d'Oracle donnent des informations sur le téléchargement et l'installation du logiciel de la mise à jour 2 de OpenSSO. Ce document contient également des informations sur les modifications apportées au logiciel depuis la sortie de la mise à jour 1 de OpenSSO.

Public cible

Ces notes de version doivent être utilisées par les administrateurs d'entreprise et les développeurs qui ont déjà installé et déployé Oracle OpenSSO 8.0. Vous devez déjà être familiarisé avec les concepts et les procédures décrits dans la documentation du produit de base.

Manuels connexes

Ces notes de version complètent la documentation du produit de base OpenSSO 8.0 d'Oracle à l'URL suivante : http://docs.sun.com/app/docs/coll/1767.1.

Références à des sites Web tiers

Des adresses URL de sites tiers, qui renvoient à des informations complémentaires connexes, sont référencées dans ce document.

Remarque – Oracle décline toute responsabilité quant à la disponibilité des sites tiers mentionnés dans ce document. Oracle ne garantit pas le contenu, la publicité, les produits ni autres matériaux disponibles sur ces sites ou dans ces ressources, ou accessibles par leur intermédiaire, et ne saurait en être tenu pour responsable. Oracle ne pourra en aucun cas être tenu pour responsable, directement ou indirectement, de tous dommages ou pertes, réels ou invoqués, causés par ou liés à l'utilisation des contenus, biens ou services disponibles dans ou par l'intermédiaire de ces sites ou ressources.

Documentation, support et formation

Consultez les sites Web suivants pour obtenir d'autres ressources :

- Documentation (`http://docs.sun.com`)
- Support (`http://www.oracle.com/us/support/systems/index.html`)
- Formation (`http://education.oracle.com`) – Cliquez sur le lien Sun dans la partie gauche de la barre de navigation.

Vos commentaires sont les bienvenus

Oracle accueille volontiers vos commentaires et suggestions sur la qualité et l'utilité de sa documentation. Si vous trouvez des erreurs ou avez d'autres suggestions en vue d'améliorations, allez sur `http://docs.sun.com` et cliquez sur Evaluation. Indiquez le titre et la référence de la documentation, ainsi que le chapitre, la section et le numéro de page, si possible. Veuillez nous indiquer si vous souhaitez une réponse.

Oracle Technology Network (`http://www.oracle.com/technetwork/index.html`) offre toute une gamme de ressources liées aux logiciels Oracle :

- Discutez des problèmes techniques et des solutions sur les Forums de discussion (`http://forums.oracle.com`).
- Profitez de tutoriels pratiques étape par étape grâce à Oracle By Example (`http://www.oracle.com/technology/obe/start/index.html`).
- Téléchargez l'exemple de code sur (`http://www.oracle.com/technology/sample_code/index.html`).

Conventions typographiques

Le tableau suivant indique les conventions typographiques utilisées dans cet ouvrage.

TABLEAU P–1 Conventions typographiques

Caractère ou symbole	Signification	Exemple
`AaBbCc123`	Noms de commandes, fichiers et répertoires ; messages système.	Modifiez le fichier `.login`. Utilisez `ls -a` pour dresser la liste des fichiers. `nom_machine% Vous avez du courrier.`
`AaBbCc123`	Caractères saisis par l'utilisateur, par opposition aux messages système.	`nom_machine%` **`su`** `Password:`

TABLEAU P–1 Conventions typographiques *(Suite)*

Caractère ou symbole	Signification	Exemple
aabbcc123	Remplacez les variables de ligne de commande par des noms ou des valeurs réels.	Pour supprimer un fichier, tapez `rm` *nomfichier*.
AaBbCc123	Titres d'ouvrages, nouveaux mots ou termes, mots importants.	Lisez le chapitre 6 du *Guide de l'utilisateur*.
		Un *cache* est une copie qui est stockée localement.
		N'enregistrez *pas* le fichier.
		Remarque : certains éléments mis en évidence apparaissent en caractères gras.

Invites de shell dans les exemples de commande

Le tableau suivant présente les invites système et superutilisateur par défaut UNIX pour les shells inclus dans le système d'exploitation Solaris d'Oracle. Notez que l'invite système par défaut affichée dans nos exemples de commande peut varier en fonction de la version de Solaris d'Oracle que vous utilisez.

TABLEAU P–2 Invites de shell

Shell	Invite
Shell Bash, shell Korn et shell Bourne	`$`
Shell Bash, shell Korn et shell Bourne pour le superutilisateur	`#`
C shell	`nom_machine%`
C shell pour superutilisateur	`nom_machine#`

◆ ◆ ◆ CHAPITRE 1

A propos de la mise à jour 2 de OpenSSO 8.0

Ce chapitre se compose des rubriques suivantes :

Nouveautés de la mise à jour 2 de OpenSSO 8.0

La mise à jour 2 de OpenSSO 8.0 comprend des améliorations apportées au service de jeton de sécurité et du Fedlet de OpenSSO.

Améliorations du service de jeton de sécurité

Le service de jeton de sécurité comprend désormais les fonctions suivantes :

- Prise en charge de type de jeton pour générer un jeton de sécurité de fournisseur de services Web spécifique
- Prise en charge des liaisons asymétriques et de transport pour X509 et des jetons de sécurité par nom d'utilisateur comme demandeur.
- Application de liaisons SSL/de transport avec un jeton de sécurité par nom d'utilisateur lorsque OpenSSO STS est configuré avec un nom d'utilisateur sur SSL.
- Emission d'un jeton de sécurité détenteur de clé SAML pour le type de clé asymétrique avec clé d'utilisation comme clé publique du client de services Web et jeton de sécurité X509 de client de services Web.
- WSDL est mis à jour de façon dynamique en fonction de la configuration du jeton de sécurité.

- Prise en charge du chiffrement par la clé publique du fournisseur de services Web.
- Chiffrement du mot de passe du nom d'utilisateur statique avant son stockage dans le magasin de configuration.
- Prise en charge du jeton Nom d'utilisateur comme jeton de sécurité Au nom de via une requête WS-Trust.
- Prise en charge de l'émission de jetons SAML Bearer.
- Nouveau module d'authentification de sécurité des services Web, WSSAuth prend en charge la validation des mots de passe Digest.
- Le nouveau module d'authentification OAMAuth permet la connexion unique à l'aide d'Oracle Access Manager avec OpenSSO.

Pour en savoir plus, consultez le Chapitre 3, " Utilisation du service de jeton de sécurité".

Améliorations du Fedlet

Le Fedlet comprend désormais les nouvelles fonctions suivantes :

- Prise en charge du chiffrement dans le Fedlet .NET.
- Prise en charge de la connexion dans le Fedlet .NET.
- Le Fedlet .NET prend désormais en charge la déconnexion unique.
- Le Felet .NET offre au fournisseur de service une connexion unique lancée et le support des artefacts.
- Prise en charge de plusieurs fournisseurs d'identités et détection du fournisseur d'identités dans le Fedlet .NET.
- Offre d'informations sur la version dans les propriétés et les fichiers de configuration pour le Fedlet.
- Nouvelle implémentation de mot de passe SPI
- Prise en charge de la requête d'attributs
- Prise en charge de la déconnexion unique

Pour en savoir plus, consultez le Chapitre 4, "Utilisation du Fedlet de OpenSSO d'Oracle".

Configuration matérielle et logicielle requise pour la mise à jour 2 de OpenSSO 8.0

Consultez "Hardware and Software Requirements For OpenSSO Enterprise 8.0 Update 1" du *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*

Prise en charge des nouveaux conteneurs Web

La mise à jour 2 de OpenSSO 8.0 prend en charge les conteneurs Web décrits dans "Support for New Web Containers" du *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*, et le nouveau conteneur Web suivant :

- Serveur Oracle WebLogic 10g, version 3 (10.3)

Mise à jour 2 de OpenSSO 8.0 : problèmes et solutions

CR 6959610 : les exemples de la mise à jour 2 de OpenSSO 8.0 doivent être supprimés dans l'environnement de production.

Les exemples de la mise à jour 2 de OpenSSO 8.0 pourraient entraîner des problèmes de sécurité.

Solution. Si vous déployez la mise à jour 2 de OpenSSO 8.0 dans un environnement de production, supprimez les exemples pour prévenir tout problème de sécurité éventuel.

CR 6964648 : les nouvelles autorisations de sécurité Java sont requises pour le serveur WebLogic 10.3.3.

Si vous déployez la mise à jour 2 de OpenSSO 8.0 sur le serveur Oracle WebLogic 10.3.3 avec le gestionnaire de sécurité activé, une autorisation de sécurité Java supplémentaire sera nécessaire.

Solution. Ajoutez l'autorisation suivante au serveur WebLogic 10.3.3 , `fichier weblogic.policy` :

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

CR 6939443 : échec de l'authentification des certificats avec vérification LDAP ou OCSP sur le serveur WebLogic 10.3.x.

En raison d'un problème dans les versions précédentes du serveur Oracle WebLogic telles que 10.3.0 et 10.3.1, l'authentification des certificats avec vérification LDAP ou OSCP activée échoue.

Solution. Ce problème a été corrigé dans le serveur WebLogic 10.3.3. Pour utiliser l'authentification des certificats avec vérification LDAP ou OSCP, utilisez la mise à jour 2 de OpenSSO avec le serveur WebLogic 10.3.3.

CR 6967026 : le configurateur ne peut pas se connecter à l'instance de serveur d'annuaire LDAPS depuis GlassFish 2.1.x.

Si le serveur GlassFish Enterprise v2.1.1 ou v2.1.2 est déployé comme conteneur Web de la mise à jour 2 de OpenSSO 8.0, le configurateur ne peut pas se connecter à une instance de serveur d'annuaire LDAPS.

Solution. Pour utiliser un serveur d'annuaire LDAPS avec GlassFish comme conteneur Web, déployez le serveur GlassFish Enterprise v2.1.

CR 6948937 : l'activation de la mise à jour 2 de OpenSSO 8.0 dans la console d'administration du serveur WebLogic 10.3.3 entraîne des exceptions.

Si vous déployez la mise à jour 2 de OpenSSO 8.0 (`opensso.war`) dans la console d'administration du serveur WebLogic 10.3.3 et cliquez sur Démarrer pour permettre à la mise à jour 2 de OpenSSO 8.0 de commencer à recevoir des requêtes, des exceptions seront lancées dans la console où le domaine du serveur WebLogic a été démarré.

Note : après avoir démarré la mise à jour 2 de OpenSSO 8.0, il reste lancé et aucune exception n'est lancée à nouveau avant l'arrêt puis le redémarrage de la mise à jour 2 de OpenSSO 8.0.

Solution. Copiez le fichier `saaj-impl.jar` depuis le fichier `opensso-client-jdk15.war` de la mise à jour 2 de OpenSSO 8.0 dans le répertoire `endorsed` de configuration du serveur WebLogic 10.3.3 comme suit :

1. Arrêtez le domaine du serveur Oracle WebLogic 10.3.3.
2. Si nécessaire, décompressez le fichier `opensso.zip` de la mise à jour 2 de OpenSSO 8.0.
3. Créez un répertoire temporaire et décompressez le fichier *zip-root*`/opensso/samples/opensso-client.zip` dans ce répertoire, où *zip-root* correspond à l'emplacement où vous avez décompressé le fichier `opensso.zip`. Par exemple :

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

4. Créez un répertoire temporaire et extrayez le fichier `saaj-impl.jar` depuis `opensso-client-jdk15.war`. Par exemple :

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

5. Créez un nouveau répertoire nommé `endorsed` sous le répertoire `WEBLOGIC_JAVA_HOME/jre/lib` (si `endorsed` n'existe pas déjà), où `WEBLOGIC_JAVA_HOME` est le JDK que le serveur WebLogic doit utiliser selon sa configuration.
6. Copiez le fichier `saaj-impl.jar` dans le répertoire `WEBLOGIC_JAVA_HOME/jre/lib/endorsed`.
7. Démarrez le domaine du serveur WebLogic.

CR 6959373 : le conteneur Web doit redémarrer après avoir exécuté le script `updateschema`.

Après avoir exécuté le script `updateschema.sh` ou `updateschema.bat` , vous devez redémarrer le conteneur Web de la mise à jour 2 de OpenSSO 8.0.

CR 6961419 : l'exécution du script `updateschema.bat` nécessite un fichier de mots de passe.

Le script `updateschema.bat` exécute plusieurs commandes `ssoadm`. Par conséquent, avant d'exécuter `updateschema.bat` sous Windows, créez un fichier de mots de passe qui contient l'utilisateur du mot de passe en texte clair pour l'utilisateur `amadmin`. Le script `updateschema.bat` vous invite à indiquer le chemin d'accès au fichier de mots de passe. Avant la fin du script, le fichier de mots de passe est supprimé.

Documentation sur la mise à jour 2 de OpenSSO 8.0

En plus de ce document, une autre documentation sur OpenSSO 8.0 est disponible dans la collection suivante :

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

Problèmes détectés dans la documentation

La mise à jour 2 de OpenSSO 8.0 comprend les problèmes détectés dans la documentation suivants :

CR 6958580 : l'aide en ligne de la console indique les agents de détection non pris en charge.

L'aide en ligne de la console d'administration de la mise à jour 2 de OpenSSO 8.0 indique des agents de détection, même si ces agents ne sont pas pris en charge.

Solution. Aucune. Ignorez les informations sur les agents de détection dans l'aide en ligne.

CR 6967006 : l'aide en ligne de la console n'indique pas les modules d'authentification OAMAuth et WSSAuth.

L'aide en ligne de la console d'administration de la mise à jour 1 de OpenSSO 8.0 n'indique pas les modules d'authentification Oracle Access Manager (OAM) et Web Services Security (WSS).

Solution. Pour en savoir plus sur ces modules d'authentification, consultez le Chapitre 3, " Utilisation du service de jeton de sécurité"

CR 6953582 : la référence à l'API Java du Fedlet doit être publique.

La référence à l'API Java du Fedlet est disponible dans le cadre de la référence à l'API Java de la mise à jour 2 de Oracle OpenSSO 8.0, disponible dans la collection de documentation suivante : `http://docs.sun.com/coll/1767.1`.

Note : la mise à jour 2 de OpenSSO 8.0 ne prend pas en charge la méthode `getPolicyDecisionForFedlet`, même si cette méthode se trouve dans la référence à l'API Java.

CR 6953579 : le fichier LISEZMOI du Fedlet de OpenSSO doit indiquer une fonction de déconnexion unique.

Les fichiers LISEZMOI du Fedlet n'indiquent pas la fonction de déconnexion unique.

Solution. Pour la mise à jour 2 de Oracle OpenSSO 8.0, la fonction de déconnexion unique du Fedlet est expliquée au Chapitre 4, “Utilisation du Fedlet de OpenSSO d'Oracle”.

Informations et ressources complémentaires

Vous trouverez également des informations et ressources utiles aux emplacements suivants :

- “Notifications et annonces de désapprobation” à la page 17
- “Comment signaler des problèmes et apporter des commentaires” à la page 18
- “Fonctions d'accessibilité destinées aux personnes handicapées” à la page 18
- “Sites Web tiers associés” à la page 18

- Services client avancés d'Oracle pour les systèmes :

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

Notifications et annonces de désapprobation

- Les API SMS (Service Management Service) (pack `com.sun.identity.sm`) et le modèle SMS ne seront pas inclus dans une future version de OpenSSO.
- Le module d'authentification Unix et l'assistant d'authentification Unix (`amunixd`) ne seront pas inclus dans une future version de OpenSSO.
- Les notes de version de Sun Java System Access Manager 7.1 indiquaient que le pack `com.iplanet.am.sdk` d'Access Manager, plus connu sous le nom Access Manager SDK (AMSDK), et tous les modèles XML et API associés, ne seront pas inclus dans une future version de OpenSSO.

  Par conséquent, lorsque AMSDK est supprimé, l'option du mode hérité et le support seront également supprimés.

  Aucune option de migration n'est disponible pour l'instant et n'est prévue à l'avenir. Oracle Identity Manager offre des solutions de provisioning utilisateur que vous pouvez utilisez à la place de l'AMSDK. Pour en savoir plus sur Identity Manager, consultez

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

Comment signaler des problèmes et apporter des commentaires

Si vous avez des questions ou des préoccupations au sujet de la mise à jour 2 de OpenSSO 8.0 ou d'une version de patch ultérieure, contactez les Ressources du support sur http://sunsolve.sun.com/.

Ce site contient des liens renvoyant à la base de connaissance, au centre de support en ligne et à la page de suivi des produits (Product Tracker), ainsi qu'à des programmes de maintenance et des numéros de téléphone de support. Si vous demandez de l'aide au sujet d'un problème, veuillez indiquer les informations suivantes :

- la description du problème, notamment la situation dans laquelle il se produit et son impact sur le fonctionnement ;
- le type de machine, la version du système d'exploitation, le conteneur Web et sa version, la version de JDK et la version de OpenSSO, notamment tout patch ou autre logiciel qui pourrait concerner le problème ;
- Etapes à suivre pour atténuer le problème
- tous les journaux d'erreur ou vidages de la mémoire.

Fonctions d'accessibilité destinées aux personnes handicapées

Pour obtenir les fonctions d'accessibilité mises à disposition depuis la publication de ce média, consultez les évaluations produit de la section 508 sur demande afin d'identifier les versions les plus adaptées au déploiement de solutions accessibles.

Pour en savoir plus sur l'engagement d'Oracle à l'égard de l'accessibilité, consultez http://www.oracle.com/index.html.

Sites Web tiers associés

Des adresses URL de sites tiers, qui renvoient à des informations complémentaires connexes, sont référencées dans ce document.

Remarque – Oracle décline toute responsabilité quant à la disponibilité des sites tiers mentionnés dans ce document. Oracle ne garantit pas le contenu, la publicité, les produits ni autres matériaux disponibles sur ces sites ou dans ces ressources, ou accessibles par leur intermédiaire, et ne saurait en être tenu pour responsable. Oracle ne pourra en aucun cas être tenu pour responsable, directement ou indirectement, de tous dommages ou pertes, réels ou invoqués, causés par ou liés à l'utilisation des contenus, biens ou services disponibles dans ou par l'intermédiaire de ces sites ou ressources.

◆ ◆ ◆ CHAPITRE 2

Installation de la mise à jour 2 de OpenSSO 8.0

Ce chapitre se compose des rubriques suivantes :

Présentation de l'installation de la mise à jour 2 de OpenSSO 8.0

Mise à jour 2 de OpenSSO 8.0 est disponible sous forme de patch **TBS**.

Avant d'installer la mise à jour 2 de OpenSSO 8.0 (ou des patches ultérieurs), vérifiez les informations sur les nouvelles fonctions, les exigences matérielles et logicielles, ainsi que les problèmes et solutions dans ce document.

La mise à jour 2 de OpenSSO 8.0 comprend un fichier `opensso.war` que vous pouvez installer à l'aide des méthodes suivantes :

- **Appliquez un patch à un déploiement de OpenSSO 8.0 existant** : utilisez l'utilitaire `ssopatch` de la mise à jour 2 pour appliquer un patch à un déploiement de OpenSSO 8.0 existant, tel que décrit dans ce chapitre.

Note : Oracle prend en charge l'application de patch uniquement sur les versions de OpenSSO 8.0. Par exemple, l'application de patch sur OpenSSO 8.0 avec la mise à jour 2 de OpenSSO 8.0 est prise en charge.

- **Installez un nouveau déploiement de la mise à jour 2 de OpenSSO 8.0 :** installez et configurez le fichier `opensso.war` de la mise à jour 2 de OpenSSO 8.0, tel que décrit dans le *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.
- **Créez un nouveau fichier WAR spécialisé** : utilisez le script `createwar` pour créer un des nouveaux fichiers WAR suivants à partir du fichier `opensso.war` de la mise à jour 2 :
  - WAR de la console d'administration OpenSSO uniquement
  - WAR du serveur de l'IU d'authentification distribuée
  - WAR du serveur OpenSSO uniquement, sans la console d'administration
  - WAR du service de détection du fournisseur d'identités

    Pour en savoir plus, consultez le Chapitre 4, “Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File” du *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.
- **Appliquez un patch à un fichier WAR OpenSSO spécialisé existant** : utilisez l'utilitaire `ssopatch` de la mise à jour 2 pour appliquer un patch à un fichier WAR OpenSSO 8.0 spécialisé existant, tel que décrit dans leChapitre 23, “Patching OpenSSO Enterprise 8.0” du *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*

Remarque – Si vous exécutez Access Manager 7.1 ou Access Manager 7 2005Q4 et que vous voulez effectuer une mise à niveau vers la mise à jour 2, procédez aux étapes suivantes :

1. Mettez à niveau Access Manager 7.x vers OpenSSO 8.0, tel que décrit dans le *Sun OpenSSO Enterprise 8.0 Upgrade Guide*.
2. Appliquez le patch de la mise à jour 2, tel que décrit dans ce chapitre.

Patches de la mise à jour 2 de OpenSSO 8.0

Sun sort régulièrement des patches pour la mise à jour 2 de OpenSSO 8.0. Pour en savoir plus sur ces patches, consultez régulièrement cette section.

Préparation de votre opération de patch

▼ Pour préparer votre opération de patch pour OpenSSO 8.0

1. Lisez la "Présentation de l'utilitaire ssopatch " à la page 23.
2. Installez l'utilitaire de patch pour votre plate-forme, tel que décrit dans "Installation de l'utilitaire ssopatch" à la page 24.
3. Obtenez des informations sur votre fichier WAR existant, afin de déterminer s'il a été personnalisé ou modifié, tel que décrit dans "Comparaison d'un fichier WAR OpenSSO et de son fichier manifest interne" à la page 26.
4. Comparez votre fichier WAR existant et le fichier WAR de la mise à jour 2, pour retourner les fichiers personnalisés dans le WAR d'origine, les fichiers mis à jour dans le nouveau fichier WAR et les fichiers ajoutés ou supprimés entre les deux versions du WAR, tel que décrit dans "Comparaison de deux fichiers WAR OpenSSO" à la page 27.
5. Sauvegardez et archivez votre fichier WAR OpenSSO existant, tel que décrit dans "Sauvegarde d'un fichier WAR OpenSSO" à la page 25.
6. Appliquez un patch à votre fichier WAR OpenSSO, tel que décrit dans "Application de patch à un fichier WAR OpenSSO" à la page 28.
7. Exécutez le script `updateschema`, tel que décrit dans "Exécution du script updateschema" à la page 31.

   Note : si vous appliquez un patch à un fichier WAR spécialisé que vous avez généré à partir d'un `opensso.war`, tel qu'un serveur OpenSSO uniquement, une console d'administration uniquement, un serveur d'IU d'authentification distribuée ou un WAR de service de détection de fournisseur d'identités, consultez "Application de patch à un WAR OpenSSO spécialisé" à la page 31.

Présentation de l'utilitaire `ssopatch`

L'utilitaire `ssopatch` est un utilitaire de ligne de commande Java disponible sur les systèmes Solaris et Linux sous la forme `ssopatch` et sur Windows sous la forme `ssopatch.bat`.

Note : la syntaxe de `ssopatch` dans la mise à jour 2 de OpenSSO 8.0 a subi des modifications considérables depuis la sortie de OpenSSO 8.0. Pour connaître la nouvelle syntaxe, consultez "Exécution du script updateschema" à la page 31.

L'utilitaire de patch ssopatch effectue les fonctions suivantes :

- Comparer un fichier WAR OpenSSO à son fichier manifest d'origine, afin de déterminer s'il a été personnalisé ou modifié
- Comparer deux fichiers WAR OpenSSO, pour déterminer les différences entre les deux, notamment toute personnalisation apportée au fichier WAR d'origine et toute modification dans le nouveau fichier WAR
- Générer une zone de transit des fichiers devant générer un nouveau fichier WAR OpenSSO auquel on a appliqué un patch

Après avoir téléchargé et décompressé le fichier ZIP de la mise à jour 2 de OpenSSO 8.0 (opensso_80U2.zip), les utilitaires de patch et fichiers associés seront disponibles dans le fichier ssoPatchTools.zip, dans le répertoire *zip-root*/opensso/tools , où *zip-root* correspond à l'emplacement où vous avez décompressé opensso_80U2.zip.

L'utilitaire ssopatch utilise un fichier global pour déterminer le contenu d'un fichier WAR OpenSSO spécifique. Un fichier global est un fichier texte ASCII qui contient :

- une chaîne qui identifie la version spécifique du fichier WAR OpenSSO
- tous les fichiers individuels dans le fichier WAR OpenSSO, avec des informations sur la somme de contrôle pour chaque fichier

Le fichier global est généralement nommé OpenSSO.manifest et est stocké dans le répertoire META-INF du fichier WAR OpenSSO.

L'utilitaire ssopatch envoie ses résultats vers la sortie standard (stdout). Si vous préférez, vous pouvez capturer la sortie ssopatch en redirigeant la sortie vers un fichier. Si ssopatch se termine correctement, il retourne un code de sortie de zéro (0). En cas d'erreur, ssopatch retourne un code de sortie autre que zéro.

Installation de l'utilitaire ssopatch

Avant d'installer l'utilitaire ssopatch :

- Téléchargez et décompressez le fichier ZIP de la mise à jour 2 de OpenSSO 8.0 (opensso_80U2.zip).
- Paramétrez votre point de variable d'environnement JAVA_HOME à JDK 1.5 ou une version ultérieure.

Pour installer l'utilitaire ssopatch

1. Localisez le fichier ssoPatchTools.zip dans le répertoire *zip-root*/opensso/tools, où *zip-root* correspond à l'emplacement où vous avez décompressé opensso_80U2.zip .

2. Créez un nouveau répertoire pour décompresser le fichier `ssoPatchTools.zip`. Par exemple : `ssopatchtools`
3. Décompressez le fichier `ssoPatchTools.zip` dans le nouveau répertoire.
4. Si vous souhaitez exécuter l'utilitaire `ssopatch` à partir d'un répertoire autre que son répertoire actuel, sans fournir le chemin d'accès complet, ajoutez l'utilitaire à votre variable `PATH`.

Le tableau suivant décrit les fichiers du `ssoPatchTools.zip`.

Fichier ou répertoire	Description
`LISEZMOI`	Fichier LisezMoi qui décrit `ssopatch`
`/lib`	Fichiers JAR `ssopatch` requis
`/patch`	Scripts `updateschema` et `updateschema.bat` et fichiers XML associés
`/resources`	Fichiers de propriétés requis
`ssopatch` et `ssopatch.bat`	Utilitaires pour les systèmes Solaris, Linux et Windows

Sauvegarde d'un fichier WAR OpenSSO

Avant de commencer, sauvegardez votre fichier WAR OpenSSO existant et vos données de configuration :

- Copiez votre fichier WAR OpenSSO existant dans un emplacement sûr. Puis, si vous devez sauvegarder la mise à jour 2 pour une raison quelconque, vous pouvez redéployer votre copie de sauvegarde du fichier WAR.
- Sauvegardez vos données de configuration, tel que décrit dans le Chapitre 15, “Backing Up and Restoring Configuration Data” du *Sun OpenSSO Enterprise 8.0 Administration Guide*.

Exécution de l'utilitaire `ssopatch`

Pour exécuter l'utilitaire `ssopatch`, procédez comme suit :

```
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]
```

où les options sont :

- `-war-file|-o` indique un chemin d'accès à un fichier WAR (tel que `opensso.war`) déployé auparavant.
- `-manifest|-m` indique le chemin d'accès au fichier global que vous souhaitez créer. Le fichier global sera généré à partir du fichier WAR indiqué par `-war-file|-o` si cette option est offerte.
- `-war-file-compare|-c` indique un chemin d'accès à un fichier WAR à comparer avec le fichier WAR indiqué par `-war-file|-o` .
- `-staging|-s` indique un chemin d'accès à la zone de transit où les fichiers d'un WAR OpenSSO seront inscrits.
- `-locale|-l` indique la langue à utiliser. Si cette option n'est pas précisée, `ssopatch` utilise la langue du système par défaut.
- `-override|-r` écrase le contrôle des révisions pour les deux fichiers WAR. Le contrôle des révisions détermine les versions des fichiers WAR et continue uniquement si les versions sont compatibles. Cette option vous permet d'ignorer ce contrôle.

  La valeur par défaut est false (contrôle des révisions effectué).
- `-overwrite|-w` écrase les fichiers de la zone de transit existante. La valeur par défaut est false (les fichiers ne sont pas écrasés).

Comparaison d'un fichier WAR OpenSSO et de son fichier manifest interne

Utilisez cette procédure pour déterminer si un fichier WAR OpenSSO a été personnalisé ou modifié depuis son téléchargement.

L'utilitaire `ssopatch` génère un nouveau fichier manifest interne, puis le compare au fichier manifest stocké dans le fichier WAR OpenSSO d'origine dans le répertoire `META-INF`.

Pour comparer un fichier WAR OpenSSO et son fichier manifest interne

1. Exécutez `ssopatch` pour comparer le fichier WAR OpenSSO à son fichier manifest interne. Par exemple :

```
./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
```

Cet exemple montre les modifications apportées au fichier WAR d'origine :

- `images/login-origimage.jpg` est dans `opensso.war` mais pas dans le fichier manifest d'origine.
- `images/login-backimage.jpg` a été personnalisé dans `opensso.war` à partir du fichier manifest d'origine.
- Le fichier `WEB-INF/classes/amConfigurator.properties` a été personnalisé dans`opensso.war` à partir du fichier manifest d'origine.

Comparaison de deux fichiers WAR OpenSSO

Utilisez cette procédure pour comparer deux fichiers WAR, pour montrer les fichiers qui ont été :

- personnalisés dans un fichier WAR OpenSSO d'origine
- mis à jour dans un nouveau fichier WAR OpenSSO
- ajoutés ou supprimés entre les deux versions de WAR OpenSSO

Pour comparer deux fichiers WAR OpenSSO

1. Exécutez `ssopatch` pour comparer les deux fichiers WAR. Dans l'exemple, l'option `-override` est utilisée pour écraser le cotnrôle des révisions entre les deux fichiers WAR :

```
./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
```

Cet exemple montre les fichiers qui ont été mis à jour et personnalisés dans le nouveau fichier WAR.

Application de patch à un fichier WAR OpenSSO

Utilisez cette procédure pour créer une nouvelle zone de transit, où un fichier WAR d'origine est fusionné à un nouveau fichier WAR.

Cette opération permet de comparer les fichiers manifest pour chaque fichier WAR et indique :

- les fichiers personnalisés dans le fichier WAR d'origine
- les fichiers mis à jour dans un nouveau fichier WAR
- les fichiers ajoutés ou supprimés entre les deux versions de fichiers WAR

Le `ssopatch` copie alors les fichiers appropriés dans un répertoire de transit, où vous devez ajouter toute personnalisation avant de créer et de déployer le nouveau fichier WAR auquel on a appliqué un patch.

Pour créer une zone de transit pour appliquer un patch à un fichier WAR OpenSSO

1. Bien que le `ssopatch` ne modifie pas votre fichier `opensso.war` d'origine, il est recommandé de le sauvegarder, au cas où vous devriez supprimer le fichier `opensso.war` auquel un patch a été appliqué.
2. Exécutez `ssopatch` pour créer la zone de transit. Par exemple :

```
./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
```

Dans cet exemple, `/tmp/staging` correspond à la zone de transit où `ssopatch` copie les fichiers.

Mettez les fichiers à jour si nécessaire dans la zone de transit, à l'aide des résultats de l'étape précédente.

Utilisez le tableau suivant pour déterminer la mesure à prendre pour chaque fichier avant de générer un nouveau fichier WAR auquel o na appliqué un patch.

Résultats de `ssopatch`	Explication et action requise
`Fichier pas dans le WAR d'origine` *filename*	Le fichier indiqué n'existe pas dans le fichier WAR d'origine mais se trouve dans la dernière version du fichier WAR. **Action** : aucune
`Fichier mis à jour dans le nouveau WAR` *filename*	Le fichier indiqué existe dans les fichiers WAR d'origine et nouveau et a été mis à jour dans la dernière version du fichier WAR. Aucune personnalisation n'a été effectuée dans le fichier WAR d'origine. **Action** : aucune
`Fichier personnalisé` *filename*	Le fichier indiqué existe dans les deux fichiers WAR, a été personnalisé dans la version originale du fichier WAR, mais n'a pas été mis à jour dans la dernière version du fichier. **Action** : aucune
`Il se peut qu'une personnalisation manuelle soit nécessaire` *filename*	Le fichier existe dans les deux fichiers WAR, a été personnalisé dans la version originale du fichier WAR, et a été mis à jour dans la dernière version du fichier. **Action** : si vous souhaitez que vos personnalisations apparaissent dans le fichier, vous devez les ajouter manuellement au nouveau fichier mis à jour dans le répertoire de transit.
`Le fichier a été personnalisé dans la version originale, mais est introuvable dans le nouveau fichier WAR`	Le fichier existait dans le fichier WAR d'origine, mais n'est pas dans le nouveau. **Action** : aucune.

Etapes suivantes

1. Créez un nouveau fichier WAR OpenSSO à partir des fichiers de la zone de transit. Par exemple :

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

où `/patched/opensso.war` correspond au nouveau fichier WAR OpenSSO auquel un patch a été appliqué

2. Redéployez le fichier `/patched/opensso.war` sur le conteneur Web à l'aide de l'URI de déploiement d'origine. Par exemple, `/opensso`

Modifications de configuration de OpenSSO. Un nouveau fichier WAR OpenSSO peut comporter des modifications de configuration qui ne figuraient pas dans votre fichier WAR d'origine. Toute modification de configuration, le cas échéant, sera documentée séparément pour chaque patch. Consultez la documentation sur les patches ainsi que les *Sun OpenSSO Enterprise - 8.0 - notes de version* pour en savoir plus sur les modifications de configuration. (La chaîne de version du fichier global OpenSSO sera modifiée, même s'il n'y a aucune modification de configuration dans le nouveau fichier WAR.)

Si vous devez supprimer votre version avec patch, annulez le déploiement du fichier WAR avec patch, puis redéployez votre fichier WAR d'origine.

Création d'un fichier global WAR OpenSSO

Un fichier global OpenSSO est un fichier texte qui identifie tous les fichiers individuels d'un fichier WAR pour une version spécifique, avec des informations sur la somme de contrôle pour chaque fichier.

Utilisez cette procédure pour créer un fichier global que vous pouvez inclure dans un WAR OpenSSO spécialisé, comme un WAR de serveur OpenSSO uniquement, console d'administration uniquement, serveur d'IU d'authentification distribuée ou du service de détection de fournisseur d'identités

Pour créer un fichier global WAR OpenSSO

1. Exécutez `ssopatch` pour créer le fichier global OpenSSO. Par exemple :

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

   où `opensso.war` est un fichier WAR OpenSSO existant.

   L'utilitaire `ssopatch` crée un nouveau fichier global nommé `manifest` dans le répertoire `/tmp`.

2. Pour permettre au fichier WAR de se voir appliquer un patch, copiez ce nouveau fichier global dans le répertoire `META-INF` à l'intérieur du fichier `opensso.war` . Par exemple :

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

Application de patch à un WAR OpenSSO spécialisé

Si vous avez précédemment créé un WAR OpenSSO spécialisé, comme un WAR serveur OpenSSO uniquement, console d'administration uniquement, serveur IU d'authentification distribuée ou service de détection de fournisseur d'identités, vous pouvez y appliquer un patch à l'aide de l'utilitaire `ssopatch` .

Pour appliquer un patch à un WAR OpenSSO spécialisé

1. Créez un fichier global pour votre WAR OpenSSO spécialisé, tel que décrit dans "Création d'un fichier global WAR OpenSSO" à la page 30.

   Note : créez le fichier global basé sur le fichier `opensso.war` OpenSSO 8.0 d'origine, comme le fournit Sun, avant toute personnalisation. Si le fichier manifest est créé après les personnalisations, `ssopatch` peut utiliser les fichiers de la mise à jour 2, plutôt que vos personnalisations, vous devriez alors refaire vos personnalisations après l'application des patches.
2. Générez le fichier WAR OpenSSO spécialisé à partir du fichier `opensso.war` de la mise à jour 2 de OpenSSO 8.0, tel que décrit dans le Chapitre 4, "Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File" du *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.
3. Utilisez l'utilitaire `ssopatch` pour comparer vos ancien et nouveau fichiers WAR.
4. Générez une zone de transit pour le nouveau fichier WAR spécialisé, tel que décrit dans "Pour créer une zone de transit pour appliquer un patch à un fichier WAR OpenSSO " à la page 28.
5. Redéployez le nouveau fichier WAR spécialisé.

Exécution du script `updateschema`

Après avoir exécuté `ssopatch`, exécutez le `updateschema.sh` sur les systèmes Solaris ou Linux ou `updateschema.bat` sous Windows. Le script met à jour la version du serveur OpenSSO, ajoute de nouvelles propriétés serveur par défaut, ajoute de nouveaux schémas d'attributs requis pour les correctifs de bogues et les améliorations dans la mise à jour 2. Vous devez exécuter `updateschema` afin de mettre à jour la version du serveur.

Avant de commencer

- Le script `updateschema.sh` ou `updateschema.bat` nécessite la version de la mise à jour 2 (ou une version ultérieure) de l'utilitaire de ligne de commande `ssoadm` . Par conséquent, avant d'exécuter ce script, installez les outils d'administration de la mise à jour 2, tel que décrit dans le Chapitre 3, "Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools" du *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.

- Le script `updateschema.bat` exécute plusieurs commandes `ssoadm`. Par conséquent, avant d'exécuter `updateschema.bat` sous Windows, créez un fichier de mots de passe qui contient l'utilisateur du mot de passe en texte clair pour l'utilisateur `amadmin`. Le script `updateschema.bat` vous invite à indiquer le chemin d'accès au fichier de mots de passe. Avant la fin du script, le fichier de mots de passe est supprimé.

Pour exécuter le script `updateschema`

1. Passez au répertoire *patch-tools*`/patch` , où *patch-tools* correspond à l'emplacement où vous avez décompressé `ssoPatchTools.zip`.
2. Exécutez `updateschema.sh` ou `updateschema.bat` . Par exemple, sur les systèmes Solaris :

   ```
   ./updateschema.sh
   ```
3. Lorsque les scripts vous y invitent, indiquez les informations suivantes :
   - Chemin d'accès complet vers l'utilitaire `ssoadm` (à l'exclusion de `ssoadm` lui-même). Par exemple : `/opt/ssotools/opensso/bin`
   - mot de passe `amadmin`

   Le script `updateschema.sh` ou `updateschema.bat` écrit des messages ou erreurs à la sortie standard.
4. Redémarrez le conteneur Web de la mise à jour 2 de OpenSSO 8.0.

Suppression de l'installation d'un patch

Si vous devez supprimez l'installation de votre patch, redéployez simplement le fichier `opensso.war` d'origine (ou le fichier WAR spécialisé).

◆ ◆ ◆ CHAPITRE 3

Utilisation du service de jeton de sécurité

En tant que service d'autorité de confiance, le service de jeton de sécurité OpenSSO émet et valide des jetons de sécurité. En tant que fournisseur de sécurité de services Web, le service de jeton de sécurité sécurise la communication entre le client du service Web et le service STS de OpenSSO lui-même. De nombreuses améliorations ont été apportées au service de jeton de sécurité depuis la mise à jour 2 de OpenSSO 8.0.

Ce chapitre se compose des rubriques suivantes :

Ajout d'un module d'authentification WSSAuth

Le module d'authentification de sécurité des services Web permet à OpenSSO de valider un nom d'utilisateur avec un mot de passe Digest reçu comme jeton d'authentification et contenu dans une requête de service de la part du client de services Web à un fournisseur de services Web.

▼ Pour ajouter une nouvelle instance de module d'authentification de sécurité des services Web

1. **Dans l'onglet Access Manager, cliquez sur le sous-onglet Authentification.**
2. **Dans la section Instances de module, cliquez sur Nouvelle.**

3 **Dans le champ Nom, saisissez un nom pour l'instance de module d'authentification WSSAuth.**

4 **Pour le Type, choisissez WSSAuth.**

5 **Configurez l'instance du module d'authentification WSSAuth.**

▼ Pour configurer une instance du module d'authentification WSSAuth

1 **Dans l'onglet Access Manager, cliquez sur le sous-onglet Authentification.**

2 **Dans la section Instances du module, cliquez sur le nom de l'instance du module d'authentification WSSAuth que vous souhaitez configurer.**

3 **Indiquez les valeurs des attributs de domaine de l'instance du module d'authentification WSSAuth.**

Le tableau suivant présente une liste et les descriptions des attributs que vous pouvez configurer.

Attribut de recherche utilisateur	A développer
Domaine utilisateur	A développer
Attribut de mot de passe utilisateur	A développer
Niveau d'authentification	A développer

Ajout d'un module d'authentification OAMAuth

Le module d'authentification d'Oracle permet à OpenSSO d'authentifier et de connecter un administrateur (connexion unique), qui s'est précédemment authentifié sur Oracle Access Manager, à OpenSSO. L'administrateur n'a pas à fournir de références pour OpenSSO.

▼ Pour ajouter une nouvelle instance du module d'authentification d'Oracle

1 **Dans l'onglet Access Manager, cliquez sur le sous-onglet Authentification.**

2 **Dans la section Instances de module, cliquez sur Nouvelle.**

3 **Dans le champ Nom, saisissez un nom pour l'instance de module d'authentification d'Oracle.**

4 **Pour le Type, choisissez OAMAuth.**

5 **Cliquez sur OK.**

6 **Configurez l'instance du module d'authentification OAMAuth.**

▼ Pour configurer une instance du module d'authentification d'Oracle

1 **Dans l'onglet Access Manager, cliquez sur le sous-onglet Authentification.**

2 **Dans la section Instances du module, cliquez sur le nom de l'instance du module d'authentification OAMAuth que vous souhaitez configurer.**

3 **Indiquez les valeurs des attributs de domaine de l'instance du module d'authentification d'Oracle.**

Le tableau suivant présente une liste et les descriptions des attributs que vous pouvez configurer.

Nom de l'en-tête de l'utilisateur distant	A développer
Valeurs d'en-tête autorisées	La liste des valeurs actuelles affiche A développer ■ Pour ajouter une valeur d'en-tête à la liste, dans le champ Nouvelle valeur, saisissez A développer, puis cliquez sur Ajouter. ■ Pour supprimer une entrée de la liste des valeurs actuelles, sélectionnez l'entrée puis cliquez sur Supprimer.
Niveau d'authentification	A développer

Génération de jetons de sécurité

Le service de jeton de sécurité (STS) de OpenSSO d'Oracle établit une relation de confiance entre un client de services Web et un fournisseur de services Web, puis agit en tant que courtier entre eux. Le service Web peut compter sur les jetons émis par une seule entité OpenSSO STS au lieu d'avoir à communiquer avec plusieurs clients. Ainsi, OpenSSO STS réduit considérablement les frais de gestion du point de confiance.

Les sections suivantes fournissent des instructions sur la détermination de vos besoins en jetons de sécurité, et sur la configuration du service de jeton de sécurité pour générer et valider des jetons afin de répondre à ces besoins.

Enregistrement d'un fournisseur de services Web à OpenSSO STS

Lorsque vous ajoutez un nouveau profil d'agent de sécurité de fournisseur de services Web, ce dernier est automatiquement enregistré sur OpenSSO STS. Consultez les sections suivantes pour en savoir plus :

Une fois que vous avez enregistré un fournisseur de services Web sur OpenSSO STS, vous pouvez configurer ce service de façon à ce qu'il génère des jetons de sécurité de client Web acceptables par le fournisseur de services Web.

Enregistrement d'un jeton de sécurité de client de services Web depuis OpenSSO STS

Avant de pouvoir configurer le service de jeton de sécurité, pour générer des jetons de sécurité de client Web, vous devez déterminer le type de jeton dont le fournisseur de services Web a besoin. OpenSSO STS prend en charge les jetons de sécurité du projet Liberty Alliance et les jetons du profil de sécurité de base d'interopérabilité avec les services Web.

Processus de génération de jetons de sécurité

Lorsque la sécurité est activée à l'aide de jetons du projet Liberty Alliance, le client HTTP, ou le navigateur, envoie une requête d'accès via le client de services Web au fournisseur de services Web. Un agent de sécurité des services Web redirige la requête vers le service d'authentification OpenSSO STS. Une fois le mécanisme de sécurité du projet Liberty Alliance en place, un agent de sécurité HTTP émet la redirection. Lorsque la sécurité WS-IBS est utilisée, un agent de sécurité SOAP émet la redirection.

Le service d'authentification STS de OpenSSO détermine le mécanisme de sécurité enregistré par le fournisseur de services Web, et récupère les jetons de sécurité appropriés. Suite à une authentification réussie, le client des services Web fournit un corps de message SOAP alors que l'agent de sécurité SOAP du côté du client des services Web insère l'en-tête de sécurité et un jeton. Le message est alors signé avant l'envoi de la requête au fournisseur de services Web.

L'agent de sécurité SOAP du côté du fournisseur de services Web vérifie la sécurité et le jeton de sécurité de la requête SOAP avant de transférer la requête au fournisseur de services Web lui-même. Le fournisseur de services Web la traite ensuite et renvoie une réponse, signée par

l'agent de sécurité SOAP, au client des services Web. L'agent de sécurité SOAP du côté du client des services Web vérifie alors la signature avant de transférer la réponse au client des services Web.

Le tableau suivant comporte une liste et de brèves descriptions des jetons pris en charge pour les transactions du projet Liberty Alliance.

TABLEAU 3–1 Jetons du demandeur : Projet Liberty Alliance

Jeton	Répond à ces besoins
X.509	■ Le service Web sécurisé utilise une infrastructure de clé publique (PKI) dans laquelle le client des services Web fournit une clé publique comme moyen d'identifier le demandeur, et de l'authentifier avec le fournisseur de services Web. ■ Le service Web sécurisé utilise une infrastructure de clé publique (PKI) dans laquelle le client des services Web fournit une clé publique comme moyen d'identifier le demandeur, et de l'authentifier avec le fournisseur de services Web.
Jeton Bearer	■ Le service Web sécurisé utilise la méthode de confirmation du jeton Bearer SAML (Security Assertion Markup Language). ■ Le client des services Web fournit une assertion SAML avec les informations sur la clé publique comme moyen d'authentifier le demandeur pour le fournisseur de services Web. ■ Une seconde signature associé l'assertion au message SOAP. ■ La liaison de la seconde signature utilise des règles définies par le projet Liberty Alliance.
Jeton SAML	■ Le service Web sécurité utilise la méthode de confirmation du détenteur de clé. ■ Le client des services Web ajoute une assertion SAML et une signature numérique à un en-tête SOAP. ■ Un certificat d'expéditeur ou une clé publique est également fourni avec la signature. ■ L'envoi est traité à l'aide de règles définies par le projet Liberty Alliance.

Le tableau suivant comporte une liste et de brèves descriptions des jetons pris en charge pour les transactions WS-IBS.

TABLEAU 3–2 Jetons du demandeur - WS-IBS

Jeton	Répond à ces besoins

TABLEAU 3–2 Jetons du demandeur - WS-IBS *(Suite)*

Nom d'utilisateur	■ Le service Web sécurisé nécessite un nom d'utilisateur, un mot de passe et, en option, une signature de la requête. ■ Le client du service Web fournit un jeton de nom d'utilisateur comme moyen d'identifier le demandeur. ■ Le client du service Web fournit un mot de passe, un secret partagé ou un équivalent de mot de passe pour authentifier l'identité pour le fournisseur de services Web.
X.509	Le service Web sécurisé utilise une infrastructure de clé publique (PKI) dans laquelle le client des services Web fournit une clé publique comme moyen d'identifier le demandeur et d'accomplir l'authentification avec le fournisseur de services Web.
Détenteur de clé SAML	■ Le service Web sécurité utilise la méthode de confirmation du détenteur de clé. ■ Le client des services Web fournit une assertion SAML avec les informations sur la clé publique comme moyen d'authentifier le demandeur pour le fournisseur de services Web. ■ Une seconde signature associé l'assertion à la charge utile SOAP.
Bons d'expéditeur SAML	■ Le service Web sécurité utilise la méthode de confirmation des bons d'expéditeur SAML. ■ Le client des services Web ajoute une assertion SAML et une signature numérique à un en-tête SOAP. Un certificat d'expéditeur ou une clé publique est également fourni avec la signature.

Utilisation de la matrice de génération de jetons de sécurité

Utilisez la matrice de génération de jetons de sécurité pour vous aider à configurer OpenSSO STS afin de générer un jeton de sécurité de client de services Web requis par le fournisseur de services Web. Tout d'abord, dans la dernière colonne intitulée Jeton de sortie OpenSSO STS, recherchez une description qui répond aux besoins en jetons du fournisseur de services Web. Puis utilisez les valeurs des paramètres de la même ligne lorsque vous configurez le service de jeton de sécurité. La « légende de la matrice de génération de jetons » fournit des informations sur les en-têtes de tableau et les options disponibles. Consultez la section 5.2.3, « Pour configurer le service de jeton de sécurité » pour obtenir des instructions détaillées sur la configuration. Pour obtenir des informations générales sur la sécurité des services Web et la terminologie associée, consultez :

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

La matrice de génération de jetons de sécurité résume les paramètres fréquemment utilisés du service de jeton de sécurité et les types de jetons de sécurité que OpenSSO STS génère en fonction de ces paramètres.

TABLEAU 3–3 Matrice de génération de jetons de sécurité

Ligne	Liaison de sécurité au niveau des messages	Jeton de client de services Web	Type de clé	Pour le jeton	Clé d'utilisation	Jeton de sortie de OpenSSO STS
1	Asymétrique	X509	Bearer	Oui	Aucun	Bearer SAML, pas de clé de vérification
2	Asymétrique	Nom d'utilisateur	Bearer	Oui	Aucun	Bearer SAML, pas de clé de vérification
3	Asymétrique	X509	Bearer	Aucun	Aucun	Bearer SAML, pas de clé de vérification
4	Asymétrique	Nom d'utilisateur	Bearer	Aucun	Aucun	Bearer SAML, pas de clé de vérification
5	Asymétrique	X509	Symétrique	Oui	Aucun	Détenteur de clé SAML, clé de vérification symétrique
6	Asymétrique	Nom d'utilisateur	Symétrique	Oui	Aucun	Détenteur de clé SAML, clé de vérification symétrique
7	Asymétrique	X509	Symétrique	Aucun	Aucun	Détenteur de clé SAML, symétrique
8	Asymétrique	Nom d'utilisateur	Symétrique	Aucun	Aucun	Détenteur de clé SAML, clé de vérification symétrique
9	Asymétrique	X509	Asymétrique	Aucun	Clé publique du client de services Web	Détenteur de clé SAML, clé de vérification asymétrique

TABLEAU 3–3 Matrice de génération de jetons de sécurité *(Suite)*

10	Asymétrique	X509	Propriétaire Oracle pour les bons d'expéditeur SAML	Oui	Aucun	Bons d'expéditeur SAML, pas de clé de vérification
11	Asymétrique	Nom d'utilisateur	Propriétaire Oracle pour les bons d'expéditeur SAML	Oui	Aucun	Bons d'expéditeur SAML, pas de clé de vérification
12	Asymétrique	X509	Propriétaire Oracle pour les bons d'expéditeur SAML	Aucun	Aucun	ERREUR
13	Asymétrique	Nom d'utilisateur	Propriétaire Oracle pour les bons d'expéditeur SAML	Aucun	Aucun	ERREUR
14	Transport	Nom d'utilisateur	Bearer	Oui	Aucun	Bearer SAML, pas de clé de vérification
15	Transport	Nom d'utilisateur	Bearer	Aucun	Aucun	Bearer SAML, pas de clé de vérification
16	Transport	Nom d'utilisateur	Symétrique	Oui	Aucun	Détenteur de clé SAML, symétrique
17	Transport	Nom d'utilisateur	Symétrique	Aucun	Aucun	Détenteur de clé SAML, clé de vérification symétrique
18	Transport	Nom d'utilisateur	Propriétaire Oracle pour les bons d'expéditeur SAML	Oui	Aucun	Bons d'expéditeur SAML, pas de clé de vérification

TABLEAU 3–3 Matrice de génération de jetons de sécurité *(Suite)*

19	Transport	Nom d'utilisateur	Propriétaire Oracle pour les bons d'expéditeur SAML	Aucun	Aucun	ERREUR
20	Asymétrique	Nom d'utilisateur	Asymétrique	Aucun	Clé publique du client de services Web	ERREUR
21	Transport	Nom d'utilisateur	Asymétrique	Aucun	Clé publique du client de services Web	ERREUR
22	Asymétrique	X509	Asymétrique	Oui	Aucun	ERREUR
23	Asymétrique	Nom d'utilisateur	Asymétrique	Oui	Aucun	ERREUR
24	Transport	Nom d'utilisateur	Asymétrique	Oui	Aucun	ERREUR
25	Asymétrique	X509	Asymétrique	Aucun	Aucun	Détenteur de clé SAML, clé de vérification asymétrique
26	Asymétrique	X509	Aucun	Aucun	Aucun	Détenteur de clé SAML, clé de vérification asymétrique
27	Asymétrique	Nom d'utilisateur	Aucun	Aucun	Aucun	Détenteur de clé SAML, clé de vérification symétrique
28	Transport	Nom d'utilisateur	Aucun	Aucun	Aucun	Détenteur de clé SAML, clé de vérification symétrique

Service de jeton de sécurité : problèmes et solutions

A développer

Configuration : problèmes et solutions

A développer

Erratum dans la documentation

A développer

◆ ◆ ◆ CHAPITRE 4

Utilisation du Fedlet de OpenSSO d'Oracle

Cette section donne les informations suivantes sur le Fedlet de OpenSSO d'Oracle :

A propos du Fedlet de OpenSSO d'Oracle

Le Fedlet de OpenSSO d'Oracle est une implémentation du fournisseur de services léger (SP) qui peut être déployée avec une application du fournisseur de services Java ou .NET, permettant à l'application de communiquer avec un fournisseur d'identités (IDP) comme la mise à jour 2 de OpenSSO d'Oracle à l'aide du protocole SAMLv2. Le Fedlet est disponible en deux versions, selon la plate-forme que vous utilisez :

- Le Fedlet Java est le premier sorti pour OpenSSO 8.0. Pour en savoir plus, consultez le Chapitre 5, "Using the OpenSSO Enterprise Fedlet to Enable Identity Federation" du *Sun OpenSSO Enterprise 8.0 Deployment Planning Guide*.
- Le Fedlet .NET est sorti dans la mise à jour 1 de OpenSSO 8.0. Pour en savoir plus, consultez le Chapitre 10, "Using the ASP.NET Fedlet with OpenSSO Enterprise 8.0 Update 1" du *Sun OpenSSO Enterprise 8.0 Update 1 Release Notes*.

Dans la mise à jour 2 de OpenSSO 8.0 d'Oracle, le Fedlet est acessible comme suit :

- Après avoir décompressé le fichier ZIP de la mise à jour 2 de OpenSSO 8.0, le Fedlet Java et le Fedlet .NET sont disponibles dans le fichier suivant :

  zip-root`/opensso/fedlet/fedlet-unconfigured.zip`, où *zip-root* correspond à l'emplacement où vous avez décompressé le fichier ZIP de la mise à jour 2 de OpenSSO 8.0 d'Oracle.

- Après avoir installé la mise à jour 2 de OpenSSO 8.0 d'Oracle, vous pouvez créer le Fedlet Java dans la console d'administration de OpenSSO 8.0 à l'aide du flux de travail Créer un Fedlet sous Tâches courantes.

Exigences pour le Fedlet OpenSSO d'Oracle

Le Fedlet a besoin de :

- le conteneur Web pris en charge par la mise à jour 2 de OpenSSO 8.0 d'Oracle, si vous prévoyez de déployer le fichier `fedlet.war`, ou une application du fournisseur de services Java intégrée au Fedlet. Consultez la "Configuration matérielle et logicielle requise pour la mise à jour 2 de OpenSSO 8.0 " à la page 12.
- Microsoft Internet Information Server (IIS) 7.0 et versions ultérieures, si vous prévoyez de déployer le Fedlet .NET
- JDK 1.6.x et versions ultérieures

Configuration du Fedlet de OpenSSO d'Oracle

Cette section décrit la première configuration du Fedlet avec une application de fournisseur de services :

- "Pour configurer le Fedlet Java" à la page 44
- "Pour configurer le Fedlet .NET" à la page 46

Après avoir terminé la première configuration du Fedlet, passez à n'importe quelle autre configuration que vous souhaitez effectuer. Plusieurs points à prendre en compte :

- Si vous modifiez le fichier du Fedlet `sp.xml`, vous devez réimporter ce fichier dans votre fournisseur d'identités.
- Si vous procédez à d'autres modifications de configuration du Fedlet du côté du fournisseur de services, transmettez les informations à l'administrateur du fournisseur d'identités, de façon à ce que les modifications de configuration requises puissent être apportées du côté du fournisseur d'identités.

▼ Pour configurer le Fedlet Java

1 **Du côté du fournisseur d'identités, générez les métadonnées XML pour le fournisseur d'identités et enregistrez ces métadonnées dans un fichier nommé `idp.xml`.**

Pour la mise à jour 2 de OpenSSO 8.0 d'Oracle, utilisez `exportmetadata.jsp`. Par exemple :

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

2 **Du côté du fournisseur de services, décompressez le fichier ZIP Fedlet (si nécessaire).**

3 Créez le répertoire de base du Fedlet, qui correspond au répertoire où le Fedlet lit ses métadonnées, son cercle de confiance et les fichiers de propriétés de configuration.

L'emplacement par défaut est le sous-répertoire du Fedlet sous le répertoire personnel de l'utilisateur qui exécute le conteneur Web du Fedlet (indiqué par la propriété JVM `user.home`). Par exemple, si ce répertoire personnel est `/home/webservd`, le répertoire de base du Fedlet est :

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

Pour modifier le répertoire de base par défaut du Fedlet, paramétrez la valeur de la propriété d'exécution de JVM `com.sun.identity.fedlet.home` à l'emplacement désiré. Par exemple :

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

Le Fedlet lit alors ses métadonnées, son cercle de confiance et les fichiers de configuration à partir du répertoire `/export/fedlet/conf`.

4 Copiez les fichiers suivants depuis le répertoire `java/conf` du Fedlet Java dans le répertoire de base du Fedlet :

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

5 Dans le répertoire de base du Fedlet, renommez les fichiers que vous avez copiés et supprimez `-template` de chaque nom.

6 Dans les fichiers que vous avez copiés et renommés dans le répertoire de base du Fedlet, remplacez les balises tel qu'indiqué dans le tableau suivant :

Balise	Remplacer par
`FEDLET_COT`	Nom du cercle de confiance (COT) dont le fournisseur d'identités distant et l'application du fournisseur de services du Fedlet Java sont membres.
`FEDLET_ENTITY_ID`	Identifiant (nom) de l'application du fournisseur de services du Fedlet Java. Par exemple : `fedletsp`
`FEDLET_PROTOCOL`	Protocole du conteneur Web pour l'application du fournisseur de services du Fedlet Java (par exemple, `fedlet.war`). Par exemple : `https`
`FEDLET_HOST`	Nom d'hôte du conteneur Web pour l'application du fournisseur de services du Fedlet Java (par exemple, `fedlet.war`). Par exemple : `fedlet-host.example.com`
`FEDLET_PORT`	Numéro de port du conteneur Web pour l'application du fournisseur de services du Fedlet Java (par exemple, `fedlet.war`). Par exemple : `80`

Balise	Remplacer par
`FEDLET_DEPLOY_URI`	URL de l'application du fournisseur de services du Fedlet Java. Par exemple : `http://fedletsp.example.com/myFedletApp`
`IDP_ENTITY_ID`	Identifiant (nom) du fournisseur d'identités distant. Par exemple : `openssoidp`

Note : si le fournisseur de services du Fedlet ou l'identifiant de l'entité du fournisseur d'identités contient un signe de pourcentage (%) ou une virgule (,), vous devez supprimer le caractère avant de le remplacer dans le fichier `fedlet.cot`. Par exemple, modifiez "%" en "%25" et "," en "%2C".

7. **Copiez le fichier `FedletConfiguration.properties` depuis le répertoire `java/conf` du Fedlet Java dans le répertoire de base du Fedlet.**

8. **Copiez le fichier XML de métadonnées standard du fournisseur d'identités (de l'étape 1) dans le répertoire de base du Fedlet. Ce fichier doit être nommé `idp.xml`.**

9. **Importez le fichier de métadonnées XML du Fedlet Java (`sp.xml`) dans le fournisseur d'identités.**

   Pour la mise à jour 2 de OpenSSO d'Oracle, utilisez le flux de travail Enregistrer le fournisseur de services distant sous Tâches courantes dans la console d'administration OpenSSO 8.0 pour importer les métadonnées du fournisseur de services du Fedlet Java et pour ajouter le fournisseur de services dans un cercle de confiance.

Étapes suivantes Selon vos besoins, continuez et procédez à n'importe quelle configuration supplémentaire pour le Fedlet Java.

▼ Pour configurer le Fedlet .NET

1. **Du côté du fournisseur d'identités, générez les métadonnées XML pour le fournisseur d'identités et enregistrez ces métadonnées dans un fichier nommé `idp.xml`.**

   Pour la mise à jour 2 de OpenSSO 8.0 d'Oracle, utilisez `exportmetadata.jsp`. Par exemple :

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

2. **Du côté du fournisseur de services, décompressez le fichier ZIP Fedlet (si nécessaire).**

3. **Copiez les fichiers suivants depuis le dossier `asp.net/conf` du Fedlet .NET dans le dossier `App_Data` de votre application :**

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

4. **Dans le dossier `App_Data`, renommez les fichiers que vous avez copiés et supprimez `-template` de chaque nom.**

5. **Dans les fichiers que vous avez copiés et renommés dans le dossier `App_Data`, remplacez les balises tel qu'indiqué dans le tableau suivant :**

Balise	Remplacer par
`FEDLET_COT`	Nom du cercle de confiance (COT) dont le fournisseur d'identités distant et l'application du fournisseur de services du Fedlet .NET sont membres.
`FEDLET_ENTITY_ID`	Identifiant (nom) de l'application du fournisseur de services du Fedlet .NET. Par exemple : `fedletsp`
`FEDLET_DEPLOY_URI`	URL de l'application du fournisseur de services du Fedlet .NET. Par exemple : `http://fedletsp.example.com/myFedletApp`
`IDP_ENTITY_ID`	Identifiant (nom) du fournisseur d'identités distant. Par exemple : `openssoidp`

6. **Copiez le fichier XML de métadonnées standard du fournisseur d'identités (de l'étape 1) dans le dossier `App_Data` de votre application. Ce fichier doit être nommé `idp.xml`.**

7. **Copiez les fichiers `Fedlet.dll` et `Fedlet.dll.config` du dossier `asp.net/bin` du Fedlet .NET dans le dossier `corbeille` de l'application.**

8. **Importez le fichier de métadonnées XML du Fedlet .NET (`sp.xml`) dans le fournisseur d'identités.**

   Pour la mise à jour 2 de OpenSSO d'Oracle, utilisez le flux de travail Enregistrer le fournisseur de services distant sous Tâches courantes dans la console d'administration OpenSSO 8.0 pour importer les métadonnées du fournisseur de services du Fedlet .NET et pour ajouter le fournisseur de services dans un cercle de confiance.

Étapes suivantes Selon vos besoins, continuez et procédez à n'importe quelle configuration supplémentaire pour le Fedlet .NET.

Nouvelles fonctions du Fedlet dans la mise à jour 2 de OpenSSO 8.0

La mise à jour 2 de OpenSSO d'Oracle comprend les nouvelles fonctions suivantes pour le Fedlet :

Informations sur la version du Fedlet (CR 6941387)

Le Fedlet de OpenSSO d'Oracle comprend des informations sur la version. Après avoir extrait les fichiers dans le pack du Fedlet (fichier ZIP), déterminez la version du Fedlet en consultant l'un des fichiers suivants :

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

Chiffrement et déchiffrement de mots de passe du Fedlet Java (CR 6930477)

Le Fedlet Java fournit `fedletEncode.jsp` dans le fichier `fedlet.war` pour chiffrer les mots de passe `storepass` et `keypass`. Par défaut, une clé de chiffrement différente est générée pour chaque Fedlet. Pour modifier cette clé de chiffrement, paramétrez la propriété `am.encryption.pwd` dans le fichier `FederationConfig.properties` du Fedlet.

Prise en charge de la signature et du chiffrement par le Fedlet Java

Le Fedlet Java prend en charge la vérification des signatures XML et le déchiffrement des éléments chiffrés `assertion` et `NameID` ainsi que leurs attributs correspondants.

▼ Pour configurer le Fedlet Java pour la signature et le chiffrement

1 Créez un fichier keystore nommé `keystore.jks` à l'aide de l'utilitaire `keytool`.

2. **Ajoutez la clé privée (et le certificat public le cas échéant) utilisée pour la signature et la clé privée (et le certificat public le cas échéant) utilisée pour le chiffrement dans le fichier `keystore.jks`.**

3. **Créez un fichier `.storepass`.**

4. **Ajoutez le mot de passe au fichier `.storepass`. Pour chiffrer le mot de passe, utilisez `fedletEncode.jsp`.**

5. **Créez un fichier `.keypass`.**

6. **Ajoutez le mot de passe au fichier `.keypass`. Pour chiffrer le mot de passe, utilisez `fedletEncode.jsp`.**

7. **Si vous utilisez des mots de passe en texte clair, commentez la ligne suivante dans le fichier `FederationConfig.properties` :**

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

8. **Paramétrez le chemin d'accès complet pour les attributs suivants dans le fichier `FederationConfig.properties` , où *path* correspond au chemin d'accès complet vers le fichier respectif :**

   ```
   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. **Utilisez `keytool` pour exporter le certificat de signature. Par exemple :**

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

   L'outil vous invite à saisir le mot de passe utilisé pour accéder à `keystore.jks` , puis génère le certificat.

10. **Si vous avez besoin d'un certificat de chiffrement, utilisez `keytool` pour l'exporter, tel qu'indiqué dans l'étape précédente. (Ou utilisez le même certificat pour la signature et le chiffrement.)**

11. **Créez un bloc XML `KeyDescriptor` et ajoutez-y le certificat de chiffrement. Par exemple, notez la balise `use="signing"` de l'élément `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 Créez un autre bloc XML `KeyDescriptor` et ajoutez-y le certificat de chiffrement. Par exemple, notez la balise `use="encryption"` de l'élément `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 Dans le fichier `sp.xml` du Fedlet Java, ajoutez les blocs XML avec les certificats de signature et de chiffrement sous l'élément `SPSSODescriptor` . Pour obtenir un échantillon de l'élément `SPSSODescriptor`, consultez l'Exemple 4–1.

L'attribut `AuthnRequestsSigned` est paramétré à `true`, configurant ainsi le Fedlet Java pour qu'il signe toutes les requêtes d'authentification.

14 Dans le fichier `sp-extended.xml` du Fedlet Java, paramétrez les valeurs pour les documents suivants :

- `signingCertAlias` contient l'alias du certificat de signature XML dans le keystore.
- `encryptionCertAlias` contient l'alias du certificat de chiffrement XML dans le keystore.

15 Pour appliquer ce que le fournisseur de services du Fedlet Java chiffre, paramétrez les attributs suivants dans le fichier `sp-extended.xml` à `true`:

- `wantAssertionEncrypted`
- `wantNameIDEncrypted`

- `wantAttributeEncrypted`

16 Pour appliquer ce que le fournisseur de services du Fedlet Java signe et veut voir signé, paramétrez les attributs suivants à `true`:

- `wantAuthnRequestsSigned` dans le fichier `idp.xml` indique au Fedlet quoi signer.
- `AuthnRequestsSigned` et `WantAssertionsSigned` dans le fichier `sp.xml` indique au fournisseur d'identités ce que le Fedlet prévoit de signer.
- `wantArtifactResponseSigned` dans le fichier `sp-extended.xml` indique au Fedlet quoi signer.
- `wantPOSTResponseSigned` dans le fichier `sp-extended.xml`
- `wantLogoutRequestSigned` dans le fichier `sp-extended.xml`
- `wantLogoutResponseSigned` dans le fichier `sp-extended.xml`

Si le fournisseur d'identités a besoin d'une signature pour des messages particuliers, paramétrez les attributs respectifs à `true` dans le fichier `idp-extended.xml` . Par exemple, `wantLogoutRequestSigned` et `wantLogoutResponseSigned`.

Remarque – Si vous paramétrez des attributs dans le fichier `sp-extended.xml`, transmettez ces informations à l'administrateur du fournisseur d'identités, de façon à ce que les modifications nécessaires de configuration puissent être apportées dans le fournisseur d'identités.

---

17 Redémarrez le conteneur Web du Fedlet Java.

18 Importez le fichier `sp.xml` du Fedlet Java dans le fournisseur d'identités.

Exemple 4–1 Exemple d'élément `SPSSODescriptor` du Fedlet Java

```
<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
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDBnzANBgkqhkiG9w0B
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>
```

Prise en charge des requêtes d'attributs du Fedlet Java (CR 6930476)

Le Fedlet Java prend en charge la requête d'attribut SAMLv2 pour demander au fournisseur d'identités tel que la mise à jour 2 de OpenSSO 8.0 d'Oracle des valeurs d'attributs d'identités spécifiques. Vous pouvez configurer le Fedlet de façon à ce qu'il signe la requête et la chiffre. La signature est obligatoire pour émettre une requête de Fedlet, mais le chiffrement est facultatif.

▼ Pour configurer le Fedlet Java pour une requête d'attribut

1. **Activez la signature XML pour signer la requête d'attribut, tel que décrit dans "Prise en charge de la signature et du chiffrement par le Fedlet Java" à la page 48.**

2. **Ajoutez le certificat généré à l'étape précédente à l'élément `RoleDescriptor` dans le fichier `sp.xml` du Fedlet. Dans l'exemple suivant, il existe deux balises `KeyDescriptor` dans lesquelles**

vous collez le certificat. L'une sert à la signature, et l'autre au chiffrement. Si vous n'activez pas le chiffrement, la balise `KeyDescriptor use="encryption"` n'est pas nécessaire.

```
<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 **Dans le fichier `sp-extended.xml` du Fedlet Java, indiquez la valeur de l'attribut `signingCertAlias` et, s'il est configuré, de l'attribut `encryptionCertAlias`.**

Si vous prévoyez de configurer le fournisseur d'identités de façon à ce qu'il chiffre l'assertion, chiffrez également l'élément `NameID`. Ainsi, la valeur de l'attribut `wantNameIDEncrypted` doit être paramétrée à `true`. Ajoutez le code XML à l'élément `AttributeQueryConfig`. Par exemple :

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

Dans cet exemple, `test` est l'alias de l'exemple de clé.

4 **Importez le fichier de métadonnées du Fedlet Java (`sp.xml`) dans le fournisseur d'identités.**

Procédez également aux étapes de configuration supplémentaires dans le fournisseur d'identités pour prendre en charge la requête d'attribut pour le Fedlet.

Chiffrement et déchiffrement des requêtes et réponses du Fedlet .NET (CR 6939005)

Le Fedlet .NET peut chiffrer les requêtes XML sortantes et déchiffrer les réponses entrantes pour les éléments NameID, Attribute et Assertion.

▼ Pour configurer le Fedlet .NET pour le chiffrement et le déchiffrement des requêtes et réponses

1. **Importez votre certificat X.509 dans le dossier Personnel dans le compte Ordinateur local à l'aide de la capture de certificats pour la console de gestion Microsoft. Pour utiliser cette capture, consultez l'article suivant de Microsoft :**

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

2. **Donnez un nom convivial à ce certificat en consultant la boîte de dialogue Propriétés et en saisissant une valeur. (Enregistrez cette valeur pour l'étape 4.)**

3. **Paramétrez les autorisations appropriées pour permettre l'accès en lecture au certificat pour le compte utilisateur utilisé par le serveur d'information Internet (IIS) tel que décrit dans l'article de Microsoft. Par exemple :**

   a. **Dans la capture des certificats, naviguez jusqu'à Action, Toutes les tâches, puis vers Gérer les clés privées.**

   b. **Indiquez Permettre les autorisations de lecture pour le compte utilisateur exécutant IIS (en général, le SERVICE RESEAU).**

4. **Dans le fichier de métadonnées étendues du Fedlet .NET (`sp-extended.xml`), indiquez le nom convivial indiqué à l'étape 2 comme valeur de l'attribut `encryptionCertAlias`. Par exemple :**

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

5. **Dans le fichier de métadonnées du fournisseur de services du Fedlet .NET (`sp.xml`), ajoutez le `KeyDescriptor` pour la clé de chiffrement.**

   Utilisez la capture des certificats pour la console de gestion Microsoft utilisée précédemment pour exporter la clé publique de votre certificat en codage Base64 pour l'inclure dans le bloc XML `KeyDescriptor`. Ce `KeyDescriptor` doit être le premier élément enfant au sein du `SPSSODescriptor`. Par exemple :

   ```
   <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 Redémarrez le pool d'applications associé à votre application .NET.

Étapes suivantes Pour tester cette configuration, utilisez l'exemple d'application. Paramétrez également les attributs suivants pour qu'ils chiffrent les requêtes et déchiffrent les réponses avec le fournisseur d'identités, avec les modifications appropriées apportées aux métadonnées configurées :

- Assertion : paramétrez l'attribut `wantAssertionEncrypted` dans le fichier de métadonnées `sp-extended.xml` à `true` pour que le Fedlet .NET déchiffre l'élément `EncryptedAssertion` dans les réponses entrantes depuis le fournisseur d'identités.
- Attribut : paramétrez l'attribut `wantAttributeEncrypted` dans le `fichier`de métadonnées `sp-extended.xml` à true pour que le Fedlet .NET déchiffre l'élément `EncryptedAttribute` dans les réponses entrantes du fournisseur d'identités.
- NameID : paramétrez l'attribut `wantNameIDEncrypted` dans le fichier de métadonnées `idp-extended.xml` à `true` pour que le Fedlet .NET chiffre l'élément `NameID` dans les requêtes sortantes. Paramétrez ce même attribut dans `sp-extended.xml` pour que le Fedlet .NET déchiffre l'élément `EncryptedID` dans les réponses entrantes depuis le fournisseur d'identités.

Signature des requêtes et réponses du Fedlet .NET (CR 6928530)

Le Fedlet .NET prend en charge la signature des requêtes XML sortantes telles que les requêtes `Authn` et les requêtes de déconnexion.

▼ Pour configurer le Fedlet .NET pour signer les requêtes et les réponses :

1. **Importez votre certificat X.509 dans le dossier Personnel dans le compte Ordinateur local à l'aide de la capture de certificats pour la console de gestion Microsoft. Pour utiliser cette capture, consultez l'article suivant de Microsoft :**

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

2. **Donnez un nom convivial à ce certificat en consultant la boîte de dialogue Propriétés et en saisissant une valeur. (Enregistrez cette valeur pour l'étape 4.)**

3. **Paramétrez les autorisations appropriées pour permettre l'accès en lecture au certificat pour le compte utilisateur utilisé par le serveur d'information Internet (IIS) tel que décrit dans l'article de Microsoft. Par exemple :**

   a. **Dans la capture des certificats, naviguez jusqu'à Action, Toutes les tâches, puis vers Gérer les clés privées.**

   b. **Indiquez Permettre les autorisations de lecture pour le compte utilisateur exécutant IIS (en général, le SERVICE RESEAU).**

4. **Dans le fichier de métadonnées étendues du Fedlet .NET (`sp-extended.xml`), indiquez le nom convivial indiqué à l'étape 2 comme valeur de l'attribut `signingCertAlias`. Par exemple :**

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

5. **Dans le fichier de métadonnées du fournisseur de services du Fedlet .NET (`sp.xml`), ajoutez le `KeyDescriptor` pour la clé de signature.**

   Utilisez la capture des certificats pour la console de gestion Microsoft utilisée précédemment pour exporter la clé publique de votre certificat en codage Base64 pour l'inclure dans le bloc XML `KeyDescriptor`. Ce `KeyDescriptor` doit être le premier élément enfant au sein du `SPSSODescriptor`. Par exemple :

   ```
   <KeyDescriptor use="signing">
               <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                  <ds:X509Data>
                     <ds:X509Certificate>
   MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
   bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
   ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
   CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
   BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDBnzANBgkqhkiG9w0B
   AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
   RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
   Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
   QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
   cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
   /FfwWigmrW0Y0Q==
                     </ds:X509Certificate>
   ```

```
                </ds:X509Data>
              </ds:KeyInfo>
</KeyDescriptor>
```

6 Redémarrez le pool d'applications associé à votre application .NET.

Déconnexion unique du Fedlet .NET (CR 6928528 et CR 6930472)

Le Fedlet .NET prend en charge à la fois la déconnexion unique initiée par le fournisseur d'identités et par le fournisseur de services. Pour implémenter la déconnexion unique, l'exemple d'application du Fedlet .NET comprend les fichiers `logout.aspx` et `spinitiatedslo.aspx` dans le dossier `asp.net/SampleApp`. Pour voir comment la fonction de déconnexion unique du Fedlet fonctionne, déployez l'exemple d'application du Fedlet .NET.

▼ Pour configurer une application de fournisseur de services du Fedlet .NET pour la déconnexion unique :

1 Si vous n'avez pas configuré le Fedlet .NET, suivez les étapes du fichier `Lisezmoi`.

2 Copiez les fichiers `logout.aspx` et `spinitiatedslo.aspx` dans le contenu public de votre application .NET.

3 Apportez les modifications suivantes aux fichiers de configuration pour votre application :

- Dans le fichier `sp.xml`, veillez à ce que le chemin d'accès au fichier `logout.aspx` pointe vers l'emplacement correct du fichier pour votre application.
- Dans le fichier `idp.xml` (ou lors de la configuration du fournisseur d'identités), veillez à ce que le chemin d'accès au fichier `spinitiatedslo.aspx` pointe vers l'emplacement correct du fichier pour votre application.

4 Si vous souhaitez que la requête de déconnexion et la réponse de déconnexion soient signées, paramétrez les attributs suivants à `true` dans les fichiers `sp-extended.xml` et `idp-extended.xml` :

- `wantLogoutRequestSigned`
- `wantLogoutResponseSigned`

5 Importez le fichier de métadonnées du fournisseur de services du Fedlet (`sp.xml`) dans le fournisseur d'identités.

Informez également l'administrateur du fournisseur d'identités que vous avez configuré la déconnexion unique pour le fournisseur de services du Fedlet, pour que toute modification supplémentaire nécessaire puisse être apportée à la configuration du fournisseur d'identités.

Connexion unique initiée par le fournisseur de services du Fedlet .NET (CR 6928525)

Le Fedlet .NET prend en charge la connexion unique (SSO) initiée par le fournisseur de services SAMLv2. De plus, la prise en charge des artefacts est nécessaire pour permettre au Fedlet .NET de recevoir un artefact, puis de le résoudre via SOAP avec le service de résolution des artefacts du fournisseur d'identités.

L'exemple d'application du Fedlet .NET indique comment configurer la connexion unique. Une fois que les artefacts nécessaires sont installés sur votre application, un URI spécifique est nécessaire pour recevoir le POST HTTP contenant la réponse SAMLv2 après authentification réussie par le fournisseur d'identités. L'exemple de code suivant indique comment récupérer ces informations dans une application .NET :

EXEMPLE 4–2 Exemple de code pour récupérer la `AuthnResponse` dans une application du Fedlet .NET

```
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)
}
```

Si votre application reçoit la réponse SAMLv2, l'objet `authnResponse` sera renseigné avec les informations sur l'assertion. L'exemple d'application indique comment récupérer les attributs et les informations sur le sujet à partir de cet objet.

Prise en charge par le Fedlet .NET de plusieurs fournisseurs d'identités et du service de détection (CR 6928524)

Le Fedlet .NET prend en charge plusieurs fournisseurs d'identités et le service de détection de fournisseur d'identités.

Dans certains déploiements, vous voudrez peut-être configurer le Fedlet .NET avec plusieurs fournisseurs d'identités tels que la mise à jour 2 de OpenSSO 8.0 d'Oracle. Effectuez la tâche suivante pour chaque fournisseur d'identités supplémentaire que vous voulez ajouter.

▼ Pour configurer le Fedlet .NET pour plusieurs fournisseurs d'identités

1. **Obtenez le fichier de métadonnées XML auprès du fournisseur d'identités supplémentaire.**

2. **Nommez le fichier de métadonnées du fournisseur d'identités supplémentaire `idp`*n*`.xml`, où *n* correspond au fournisseur d'identités que vous ajoutez. Par exemple, nommez le fichier du deuxième fournisseur d'identités `idp2.xml`, le troisième `idp3.xml`, etc. Cette procédure utilise `idp2.xml` comme nom de fichier.**

3. **Copiez le fichier `idp2.xml` de l'étape 2 dans le dossier `App_Data` de votre application.**

4. **Ajoutez ce nouveau fournisseur d'identités au cercle de confiance du Fedlet .NET.**

   Pour ajouter le nouveau fournisseur d'identités à un cercle de confiance existant :

   Dans le fichier `fedlet.cot` du dossier `App_Data` de votre application, ajoutez l'identifiant de l'entité du nouveau fournisseur d'identités (indiqué par l'attribut `entityID` dans le fichier de métadonnées `idp2.xml`) à la valeur de l'attribut `sun-fm-trusted-providers`, en utilisant une virgule (,) comme séparateur.

   Pour ajouter le nouveau fournisseur d'identités à un nouveau cercle de confiance :

   a. **Créez un nouveau fichier nommé `fedlet2.cot` dans le dossier `App_Data` de votre application. Utilisez le `fedlet.cot` existant comme modèle, mais modifiez la valeur de l'attribut `cot-name` pour lui donner le nom du nouveau cercle de confiance (par exemple, `cot2`). Indiquez à la fois le nouvel identifiant du fournisseur d'identités et l'identifiant de l'entité du Fedlet comme valeur de l'attribut `sun-fm-trusted-providers`, en séparant les deux identifiants par une virgule (,).**

   b. **Dans le fichier `sp-extended.xml`, ajoutez le nom du nouveau cercle de confiance à valeur de l'attribut `cotlist`. Par exemple, pour un cercle de confiance nommé `cot2` :**

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

5. **Dans le dossier `App_Data` de votre application, créez un nouveau fichier `idp2-extended.xml` comme métadonnées étendues pour le nouveau fournisseur d'identités. Utilisez le fichier `idp-extended.xml` existant comme modèle, mais modifiez le `entityID` pour indiquer l'identifiant d'entité du nouveau fournisseur d'identités. Modifiez la valeur de l'attribut `cotlist` pour indiquer le nom du cercle de confiance, si un nouveau cercle de confiance est créé pour le fournisseur d'identités. Veillez à ce que le fournisseur d'identités supplémentaire soit une identité distante.**

6. **Redémarrez le pool d'applications associé à votre application du Fedlet .NET.**

7 **Le fichier XML de métadonnées du Fedlet (`sp.xml`) doit être importé dans le fournisseur d'identités supplémentaire et ajouté au même cercle de confiance comme entité de fournisseur d'identités. Importez le fichier `sp.xml` dans le fournisseur d'identités, ou donnez le fichier à l'administrateur de votre fournisseur d'identités pour qu'il l'importe.**

Prise en charge par le Fedlet .NET du service de détection du fournisseur d'identités (CR 6928524)

Dans ce scénario, le Fedlet .NET est configuré avec plusieurs fournisseurs d'identités dans un cercle de confiance et vous voulez configurer le Fedlet pour qu'il utilise le service de détection du fournisseur d'identités afin de déterminer le fournisseur favori.

Le service de détection doit être configuré pour les fournisseurs d'identités que vous utilisez avec le Fedlet .NET. Pour en savoir plus sur la configuration du service de détection du fournisseur d'identités dans la mise à jour 2 de OpenSSO 8.0 d'Oracle, consultez la collection de documentation suivante : `http://docs.sun.com/coll/1767.1`.

▼ Pour configureer le Fedlet .NET pour qu'il utilise le service de détection du fournisseur d'identités :

1 **Dans le fichier `fedlet.cot` du Fedlet .NET, paramétrez la propriété `sun-fm-saml2-readerservice-url` sur l'URL du service lecture de SAMLv2. Par exemple :**

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

2 **Redémarrez le pool d'applications associé à votre application du Fedlet .NET.**

Problèmes généraux et solutions pour le Fedlet de OpenSSO d'Oracle

A développer

Erratum dans la documentation

La référence à l'API Java du Fedlet est disponible dans la référence à l'API Java de la mise à jour 2 de OpenSSO 8.0 d'Oracle dans la collection de documentation suivante : `http://docs.sun.com/coll/1767.1`

Remarque – La méthode `getPolicyDecisionForFedlet` n'est pas prise en charge dans la version de la mise à jour 2 de OpenSSO 8.0.

◆ ◆ ◆ CHAPITRE 5

Intégration de la mise à jour 2 de OpenSSO 8.0 à Oracle Access Manager

Ce chapitre donne des instructions sur l'implémentation de la connexion unique à l'aide de la mise à jour 2 de OpenSSO 8.0 et d'Oracle Access Manager 10g ou 11g. Ces informations complètent les informations conceptuelles contenues dans le Chapitre 3, "Integrating Oracle Access Manager" du *Sun OpenSSO Enterprise 8.0 Integration Guide*. Ce cas d'utilisation offre une expérience de connexion unique aux applications protégées par OpenSSO en honorant une session Oracle Access Manager. Le module d'authentification OpenSSO configuré génère une session OpenSSO basée sur la session Oracle Access Manager.

Présentation des étapes de l'intégration

1. "Avant de commencer" à la page 63
2. Décompression des bits d'intégration
3. Création de fichiers source pour Oracle Access Manager dans OpenSSO
4. "(facultatif) Créer un plan d'authentification pour OpenSSO dans Oracle Access Manager" à la page 67
5. "Configuration d'une connexion unique à l'aide d'Oracle Access Manager et d'Oracle OpenSSO STS" à la page 68
6. "Pour tester la connexion unique" à la page 70
7. "(facultatif) Installation du plan d'authentification Oblix dans Oracle Access Manager" à la page 70

Avant de commencer

Veillez à avoir accès aux composants suivants avant de tenter d'installer la mise à jour 2 de OpenSSO 8.0 pour l'intégrer à Oracle Access Manager :

opensso.zip	Ce fichier zip contient le fichier opensso.war, le code source d'intégration, les fichiers de

	configuration et autres outils nécessaires pour l'installation et la configuration de la mise à jour 2 de OpenSSO 8.0.
Agent OpenSSO	L'Agent OpenSSO est utilisé lorsqu'une application protégée par OpenSSO peut réellement utiliser la session d'authentification établie par Oracle Access Manager.
Oracle Access Manager 10g ou 11g	Téléchargez Oracle Access Manager depuis le site Web d'Oracle. Consultez la page Téléchargements logiciels Oracle Fusion Middleware 11gR1.
Oracle Web Gate 10g ou 11g	Téléchargez Oracle Webgate pour disposer d'un conteneur pris en charge par OpenSSO et Oracle Webgate à la fois. Pour l'instant, Sun Web Server 7.x est le seul conteneur pris en charge par les deux produits. Consultez la page Téléchargements logiciels Oracle Fusion Middleware 11gR1
Oracle Access Manager SDK 10g ou 11g	Téléchargez Oracle Access Manager. Le SDK est nécessaire pour compiler et créer des modules d'authentification OpenSSO pour l'intégration d'Oracle Access Manager. Consultez la page Téléchargements logiciels Oracle Fusion Middleware 11gR1
OpenSSO C-SDK 2.2	(facultatif) Le C-SDK OpenSSO est nécessaire pour créer un module d'authentification dans Oracle Access Manager lui-même afin de générer une session OAM. Ceci peut ne pas être un cas d'utilisation courant du point de vue de OpenSSO. Voir "Where is the C SDK?" du *Sun OpenSSO Enterprise 8.0 C API Reference for Application and Web Policy Agent Developers*

Décompression des bits d'intégration

Le répertoire `opensso/integrations/oracle` contient la source et des configurations pour compiler et créer des modules d'authentification et autres plugins. Consultez le Chapitre 3, "Integrating Oracle Access Manager" du *Sun OpenSSO Enterprise 8.0 Integration Guide* pour connaître les options des cas d'utilisation et les informations associées. Le tableau suivant résume les fichiers sous le répertoire `opensso/integrations/oracle` et les descriptions de chaque fichier.

`LISEZMOI.html`	C'est le fichier que vous êtes en train de lire.
`build.xml`	Un fichier de création ant pour créer un module d'authentification personnalisé pour Oracle Access Manager dans OpenSSO
`config`	Fichiers de configuration nécessaires pour créer un module d'authentification pour Oracle Access Manager dans OpenSSO. ■ `OblixAuthService.xml` Fichier de service d'authentification pour le module d'authentification d'Oracle Access Manager ■ `OblixAuthModule.xml` Rappels du module d'authentification pour Oracle Access Manager. C'est un fichier vide par défaut, mais il doit être présent aux fins de configuration. ■ `OblixAuth.properties` Fichier de propriétés qui stocke les clés d'internationalisation pour l'authentification
`lib`	Ce répertoire est vide par défaut. Ce répertoire `lib` doit contenir les bibliothèques suivantes pour compiler les bibliothèques source. ■ `jobaccess.jar` Copiez ce fichier depuis le SDK Oracle Access Manager. ■ `openfedlib.jar`, `amserver.jar` et `opensso-sharedlib.jar` Copiez ces fichiers depuis opensso.war ■ `servlet.jar or javaee.jar` Copiez le répertoire `lib` GlassFish. Dans l'idéal, tout fichier JAR qui dispose de catégories EE Java standard telles que `javax.servlet.http.Cookie` est correct.
`source`	Répertoire contenant les fichiers source suivants : ■ `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`

  Cette catégorie est un adaptateur d'extension SAML2 pour les fournisseurs de services SAML. Cette catégorie effectue l'authentification à distance à Oracle Access Manager à l'aide du service de session OpenSSO.

oamauth (facultatif) — Ce répertoire contient les fichiers source pour le plan d'authentification Oblix pour OpenSSO. C'est un module d'authentification basé sur C qui exploite le C-SDK OpenSSo pour la validation.

- oam/solaris/authn_api.c

  Ce fichier implémente le plan d'authentification personnalisé Oblix pour OpenSSO.

- oam/solaris/include/*.h

  Tous les fichiers d'en-têtes nécessaires pour compiler un plan d'authentification.

- oam/solaris/AMAgent.properties

  Exemple de fichier de configuration de l'Agent OpenSSO. Il est nécessaire pour que le plan d'authentification puisse valider la session OpenSSO.

Création de fichiers source pour Oracle Access Manager dans OpenSSO

Utilisez le script ant pour créer les fichiers source. Un script ant compatible doit être installé et configuré dans le PATH.

▼ Pour créer les fichiers source pour Oracle Access Manager

1 Exécutez la commande suivante :

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

Cette commande crée des fichiers source et génère `fam_oam_integration.jar` dans le répertoire `$openssozipdir/integrations/oracle/dist`.

2. **Intégrez le module d'authentification au fichier WAR OpenSSO.**

   a. **Créez un répertoire temporaire et décompressez le fichier `opensso.war`. Exemple :**

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

   A partir de là, `/export/tmp` est utilisé comme zone de transit WAR et est représenté avec une marque `$WAR_DIR`.

   b. **Copiez `$openssozipdir/integrations/oracle/dist/fam_oam_integration.jar` dans `$WAR_DIR/WEB-INF/lib`.**

   c. **Copiez `$openssozipdir/integrations/oracle/config/OblixAuth.properties` dans `$WAR_DIR/WEB-INF/classes`.**

   d. **Copiez `$openssozipdir/integrations/oracle/config/OblixAuthModule.xml` dans `$WAR_DIR/config/auth/default`, ainsi que dans le répertoire `$WAR_DIR/config/auth/default_en`.**

   e. **Recompressez `opensso.war` à l'aide de `jar cvf opensso.war` à partir de `$WAR_DIR`.**

   Exemple non disponible

(facultatif) Créer un plan d'authentification pour OpenSSO dans Oracle Access Manager

Note : il ne s'agit pas d'un cas d'utilisation courant. Vous n'avez pas à le créer à moins d'y être obligé, comme dans un cas d'utilisation de fournisseur de services SAML2.

Pour créer le plan d'authentification Oblix, vous devez personnaliser le `makefile` . Comme il s'agit d'un module d'authentification basé sur C, il dépend également du système d'exploitation.

▼ Pour créer un plan d'authentification pour OpenSSO dans Oracle Access Manager

Avant de commencer

Les fichiers du plan d'authentification se trouvent sous le répertoire `$openssozipdir/integrations/oracle/oamauth/solaris` .

1. **Téléchargez et configurez la version C-SDK 2.2 de OpenSSO.**

   Le fichier `authn_api.c` contient une référence au fichier `AMAgent.properties`. Modifiez le fichier en conséquence.

2. **Personnalisez `makefile` selon votre environnement.**

   Par exemple, indiquez l'emplacement de compilation de `gcc`. Modifiez également le `LDFLAGS` pour qu'il pointe vers votre répertoire `lib` du C-SDK OpenSSO.

3. **Exécutez la commande `make`.**

   La commande `make` doit donner un fichier `authn_api.so`.

Configuration d'une connexion unique à l'aide d'Oracle Access Manager et d'Oracle OpenSSO STS

▼ Pour configurer une connexion unique à l'aide d'Oracle Access Manager et de la mise à jour 2 de OpenSSO 8.0 d'Oracle

Avant de commencer : Le serveur Web 7.x de Sun Java System doit déjà être installé et configuré. Consultez Wiki de documentation du serveur Web Sun Java System pour obtenir des instructions sur l'installation du serveur Web.

1. **Installez OpenSSO sur le serveur Web Sun Java System 7.x.**

2. **Installez un agent de règles OpenSSO sur un conteneur pris en charge et configurez l'agent pour qu'il fonctionne avec OpenSSO.**

   Consultez le *Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for J2EE Agents* ou le *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* pour obtenir des instructions sur l'installation.

3. **Installez et configurez Oracle Access Manager.**

   Consultez la page *Guide d'installation d'Oracle Access Manager 10g (10.1.4.3)*

4. **Installez et configurez Oracle Access Manager SDK avec Oracle Access Manager.**

   Consultez la page *Guide d'installation d'Oracle Access Manager 10g (10.1.4.3)*

5 Installez Oracle Webgate sur le même conteneur Web sur lequel le serveur OpenSSO est installé. (Serveur Web Sun 7.x)

Configurez OpenSSO de façon à ce qu'il protège uniquement `deployURI/UI/*` de l'application Web OpenSSO. Exemple :`/opensso/UI/.../*`

Pour connaître les règles, ressources et autres détails de la configuration d'Oracle Access Manager, consultez le guide d'administration d'Oracle Access Manager. Annulez la protection de toute autre URL dans OpenSSO Enterprise. Ceci correspond à un simple scénario d'intégration de connexion unique, mais évalue les règles basées sur une intégration totale et d'autres dépendances de déploiement.

6 Configurez le module d'authentification dans OpenSSO.

a. Accédez à la console OpenSSO.

Le navigateur redirige vers Oracle Access Manager pour l'authentification. Après une authentification réussie, OpenSSO présente une page de connexion. Connectez-vous à l'aide du nom d'utilisateur et du mot de passe administrateur de OpenSSO.

b. Importez le fichier XML du service du module d'authentification Oracle dans la configuration OpenSSO.

Le service du module d'authentification peut être chargé depuis l'utilitaire de ligne de commande `ssoadm`, ainsi que ssoadm.jsp basé sur le navigateur.

c. Accédez à `http://host:port/opensso/ssoadm.jsp`.

d. Choisissez l'option Créer un service.

e. Copiez et collez le fichier XML à partir de `$openssozipdir/integrations/oracle/config/OblixAuthService.xml` et cliquez sur Soumettre.

Ceci charge le service du module d'authentification dans la configuration de OpenSSO.

f. Enregistrez le module d'authentification dans le service de base d'authentification.

Le service de base contient une liste d'authentificateurs. Choisissez l'option `register-auth-module` dans `http://host:port/opensso/ssoadm.jsp` . Saisissez `com.sun.identity.authentication.oblix.OblixAuthModule` comme nom de classe de module d'authentification.

g. Vérifiez que le module d'authentification est enregistré dans le domaine par défaut.

Accédez à OpenSSO à l'aide de l'URL http://host:port/opensso. Dans la console OpenSSO, cliquez sur le domaine par défaut, puis cliquez sur l'onglet Authentification. Cliquez sur Nouveau pour créer un nouveau module d'authentification nommé OblixAuth.

h. Sur l'onglet Authentification, sélectionnez le module d'authentification OblixAuth.

Configurez le répertoire Oblix SDK. Activez Vérifier l'en-tête utilisateur distant uniquement, et nommez le nom de l'en-tête distant `OAM_REMOTE_USER` . Ce paramètre est configurable selon le déploiement.

7 (facultatif) Activez l'option Ignorer le profil dans le service d'authentification de base OpenSSO.

Dans la console OpenSSO, allez à Configuration > Base > Attributs du domaine > Profil utilisateur. Choisissez Ignoré, puis cliquez sur Enregistrer.

Cette configuration empêche OpenSSO de rechercher un profil utilisateur existant après une authentification réussie. Cependant, si le référentiel utilisateur utilisé par OpenSSO et Oracle Access Manager est exactement le même, cette étape n'est pas nécessaire. Allez à Console d'administration -> Configuration -> Base -> Attributs du domaine -> Profil utilisateur. Choisissez Ignoré, puis cliquez sur Enregistrer.

8 Modifiez le script de démarrage du serveur Web pour inclure les bibliothèques partagées du SDK d'Oracle Access Manager.

Mettez à jour `LD_LIBRARY_PATH` dans le script `startserv` pour inclure les bibliothèques partagées depuis `$ACCESSDKDIR/oblix/lib`.

9 Redémarrez le serveur Web Sun qui contient OpenSSO et Oracle Webgate.

10 Mettez à jour l'URL de connexion pour la valeur Agent Web :
`http://openssohost:openssoport/deployURI/UI/Login?module=OblixAuth .`

Pour tester la connexion unique

Accédez à la ressource protégée depuis l'application protégée par OpenSSO. Le navigateur doit vous rediriger vers la page de connexion à Oracle Access Manager si vous n'êtes pas déjà authentifié. Après une connexion réussie, il crée une session OpenSSO, et vous redirige finalement vers l'URL de l'application protégée par l'Agent de règles. En fonction de la règle, l'accès à l'application protégée vous est accordé ou refusé.

(facultatif) Installation du plan d'authentification Oblix dans Oracle Access Manager

Ceci est utile lorsque la session Oracle Access Manager doit être générée sur validation de la session OpenSSO. Consultez le Chapitre 3, “Integrating Oracle Access Manager” du *Sun OpenSSO Enterprise 8.0 Integration Guide*pour obtenir des informations sur les cas d'utilisation pertinents.

Les plans d'authentification Oblix sont exposés comme modules d'authentification C, et ce plan utilise la version 2.2 du C-SDK OpenSSO pour valider la session OpenSSO. Le plan d'authentification OpenSSO dans Oblix utilise une configuration pour la configuration côté client de OpenSSO dans `AMAgent.properties`. Ce fichier doit être personnalisé avant de configurer le module d'authentification. Les instructions de création indiquent l'emplacement de ce fichier. Le fichier compilé `authn_api.so` et autres bibliothèques C-SDK doivent être copiées dans le répertoire `$OAM_INSTALL_DIR/access/oblix/lib` avant de configurer le plan d'authentification. Le *Guide d'intégation de Sun OpenSSO 8.0* montre un exemple de capture d'écran illustrant comment configurer le plan d'authentification d'Oracle ; ceci doit être utilisé à titre de référence uniquement. Pour en savoir plus, consultez la dernière documentation sur Oracle Access Manager.

Intégration de la mise à jour 2 de OpenSSO 8.0 à Oracle Access Manager

Cette section donne des instructions sur l'implémentation de la connexion unique à l'aide de la mise à jour 2 de OpenSSO 8.0 et des versions 10.1.4.0.1 et 11g d'Oracle Access Manager. Ces informations complètent les informations conceptuelles contenues dans le Chapitre 3, “Integrating Oracle Access Manager” du *Sun OpenSSO Enterprise 8.0 Integration Guide*. Ce cas d'utilisation offre une expérience de connexion unique aux applications protégées par OpenSSO en honorant une session Oracle Access Manager. Le module d'authentification OpenSSO configuré génère une session OpenSSO basée sur la session Oracle Access Manager.