IBM® Tivoli® Federated Identity Manager
Version 6.2.2.7

Guide de configuration

GC11-6781-02

IBM® Tivoli® Federated Identity Manager
Version 6.2.2.7

Guide de configuration

GC11-6781-02

Important
Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 863.

Troisième édition - juillet 2013

Réf. US : GC27-2719-02

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- http://www.fr.ibm.com (serveur IBM en France)
- http://www.can.ibm.com (serveur IBM au Canada)
- http://www.ibm.com (serveur IBM aux Etats-Unis)

Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex

Remarque : La présente édition s'applique à la version 6, édition 2, modification 2.7 d'IBM Tivoli Federated Identity Manager (numéro de produit 5724-L73) ainsi qu'à toutes les éditions et modifications ultérieures, sauf mentions contraires dans les nouvelles éditions.

Table des matières

Figures

Tableaux

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
(Pos1)		Home
Fin	Fin	End
(PgAr)		PgUp
(PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
(Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

A propos de ce document

IBM® Tivoli Federated Identity Manager version 6.2.2 implémente des solutions de connexion unique fédérée, de gestion de la sécurité des services Web et d'application des accès basées sur des normes ouvertes. IBM Tivoli Federated Identity Manager étend les solutions d'authentification et d'autorisation fournies par IBM Tivoli Access Manager afin de simplifier l'intégration de plusieurs solutions Web existantes.

Ce guide décrit comment configurer IBM Tivoli Federated Identity Manager.

Public ciblé

Ce guide s'adresse aux architectes de la sécurité des réseaux, aux administrateurs système, aux administrateurs de réseau et aux intégrateurs système. Les lecteurs doivent savoir gérer les aspects de la sécurité des réseaux, la technologie de chiffrement, les clés et les certificats. Ils doivent également avoir une bonne connaissance de la mise en oeuvre des règles d'authentification et d'autorisation dans un environnement réparti.

Le présent guide décrit une mise en oeuvre d'une solution de services Web qui prend en charge plusieurs normes des services Web. Les lecteurs doivent connaître les normes des services Web spécifiques, telles qu'elle sont extraites de la documentation diffusée par l'organisme de normalisation pour chacune des normes concernées.

Les lecteurs doivent avoir une bonne connaissance du développement et du déploiement des applications à utiliser dans un environnement de services Web. Cela inclut une expérience en déploiement d'applications dans un environnement IBM WebSphere Application Server.

Accès aux publications et à la terminologie

Cette section fournit :

- Une liste des publications dans la Bibliothèque IBM Tivoli Federated Identity Manager.
- Des liens vers «Publications en ligne», à la page xx.
- Un lien vers «Site Web de terminologie IBM », à la page xx.

Bibliothèque IBM Tivoli Federated Identity Manager

Les documents suivants sont disponibles dans la bibliothèque IBM Tivoli Federated Identity Manager :

- *IBM Tivoli Federated Identity Manager - Guide de démarrage rapide*
- *IBM Tivoli Federated Identity Manager - Guide d'installation*, GC11-6700-01
- *IBM Tivoli Federated Identity Manager - Guide de configuration*, GC11-6781-02
- *IBM Tivoli Federated Identity Manager Installing, configuring, and administering risk-based access*, SC27-4445-02
- *IBM Tivoli Federated Identity Manager - Configuration de la sécurité des services Web*, GC11-2633-03

- *IBM Tivoli Federated Identity Manager - Guide d'administration*, SC11-6426-02
- *IBM Tivoli Federated Identity Manager - Guide d'audit*, GC11-2634-04
- *IBM Tivoli Federated Identity Manager - Guide de traitement des incidents*, GC11-6782-01
- *IBM Tivoli Federated Identity Manager Error Message Reference*, GC32-2289-04

Publications en ligne

Au lancement du produit et lorsque les publications sont mises à jour, IBM met les documents à disposition aux emplacements suivants :

Centre de documentation IBM Tivoli Federated Identity Manager
Le site http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tspm.doc_7.1/welcome.html affiche la page d'accueil du centre de documentation de ce produit.

Page IBM Security Systems Documentation Central and Welcome
La page IBM Security Systems Documentation Central fournit une liste alphabétique de toutes les documentations produit d'IBM Security Systems ainsi que des liens vers le centre de documentation des produits pour les versions spécifiques à chaque produit.

La page Welcome to IBM Security Systems Information Centers propose une introduction, des liens et des informations générales sur les centres de documentation IBM Security Systems.

IBM Publications Center
Le site http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss offre des fonctions de recherche personnalisées vous aidant à trouver toutes les publications IBM dont vous avez besoin.

Site Web de terminologie IBM

Le site Web de terminologie IBM regroupe la terminologie des bibliothèques de logiciels en un seul emplacement. Vous pouvez y accéder à l'adresse http://www.ibm.com/software/globalization/terminology.

Accessibilité

Les fonctions d'accessibilité permettent à un utilisateur souffrant d'un handicap physique, comme une mobilité réduite ou une vision limitée, d'utiliser plus facilement les logiciels. Ce logiciel permet d'utiliser des technologies d'assistance pour entendre les commandes et naviguer dans l'interface. Il permet également d'utiliser le clavier à la place de la souris pour activer toutes les fonctions de l'interface graphique.

Pour plus d'informations, voir la rubrique "Accessibilité" dans le centre de documentation à l'adresse http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc_6.2.2/ic/ic-homepage.html.

Formation technique Tivoli

Pour obtenir des informations sur la formation technique Tivoli, consulte le site Web IBM Tivoli à l'adresse http://www.ibm.com/software/tivoli/education.

Informations de support

Si vous rencontrez un problème avec votre logiciel IBM, vous souhaitez le résoudre rapidement. IBM vous propose différentes manières d'obtenir l'aide dont vous avez besoin :

En ligne
Accédez au site du support logiciel IBM à l'adresse http://www.ibm.com/software/support/probsub.html et suivez les instructions.

IBM Support Assistant
L'assistant IBM Support Assistant (ISA) est un outil de support logiciel local et gratuit, qui vous permet de trouver des réponses aux questions et aux incidents liés à l'utilisation d'un logiciel IBM. L'assistant ISA fournit un accès rapide aux informations d'aide et aux outils de mise en service pour la détermination d'un problème. Pour installer le logiciel ISA, voir le document *IBM Tivoli Federated Identity Manager - Guide d'installation*. Voir également : http://www.ibm.com/software/support/isa.

Guide d'identification des problèmes
Pour plus d'informations sur la résolution des problèmes, voir *IBM Tivoli Federated Identity Manager - Guide d'identification des problèmes*.

Déclaration de pratiques de sécurité recommandées

La sécurité des systèmes informatiques implique la protection des systèmes et des informations par la prévention, la détection et la réponse aux accès non autorisés depuis l'intérieur ou l'extérieur de votre entreprise. Un accès non autorisé peut se traduire par la modification, la destruction, ou une utilisation inadéquate ou malveillante de vos systèmes, y compris l'utilisation de ces derniers pour attaquer d'autres systèmes. Aucun système ou produit informatique ne doit être considéré comme étant complètement sécurisé et aucun produit, service ou mesure de sécurité ne peut être entièrement efficace contre une utilisation ou un accès non autorisé. Les systèmes, les produits et les services IBM sont conçus pour s'intégrer à une approche de sécurité complète, qui implique nécessairement des procédures opérationnelles supplémentaires, et peuvent avoir besoin d'autres systèmes, produit ou services pour optimiser leur efficacité. IBM NE GARANTIT PAS QUE TOUS LES SYSTEMES, PRODUITS OU SERVICES SONT A L'ABRI DES CONDUITES MALVEILLANTES OU ILLICITES DE TIERS OU QU'ILS PROTEGERONT VOTRE ENTREPRISE CONTRE CELLES-CI.

Conventions utilisées dans ce document

Ce document utilise plusieurs conventions typographiques pour signaler des actions et des termes particuliers, ainsi que des commandes et des chemins d'accès spécifiques du système d'exploitation.

Conventions typographiques

Cette publication utilise les conventions typographiques suivantes :

Gras

- Commandes en minuscules ou majuscules-minuscules combinées difficiles à distinguer du texte environnant
- Composants d'interface (cases à cocher, boutons de fonction, boutons radio, sélecteurs rotatifs, zones, dossiers, icônes, listes déroulantes, options des listes déroulantes, listes à colonnes multiples, conteneurs, options de menu, noms de menus, onglets et pages de propriétés) et intitulés de rubrique (tels que **Astuce :** ou **Spécificité liée au système d'exploitation** :)
- Mots clés et paramètres du texte

Italique

- Citations (exemple : titres des publications, disquettes et CD)
- Mots définis dans le texte (exemple : une ligne spécialisée est appelée une *ligne point-à-point*)
- Emphase mise sur les mots et les lettres (mots en mots, exemple : "Utilisez le mot *que* pour commencer une clause restrictive."; lettres en lettres, exemple : "L'adresse LUN doit commencer par la lettre *L*.")
- Nouveaux termes du texte (sauf dans une liste de définitions) : une *vue* est un cadre dans un espace de travail contenant des données.
- Variables et valeurs à saisir : ... où *valeur1* repésente....

`Espacement simple`

- Exemples et exemples de code
- Noms de fichier, mots clés de programmation et autres éléments difficiles à distinguer du texte environnant
- Texte des messages et invites destinés à l'utilisateur
- Texte que l'utilisateur doit taper
- Valeurs des arguments ou des options de commande

Variables et chemins de système d'exploitation

Le présent document utilise la convention UNIX pour la spécification des variables d'environnement pour la notation des répertoires.

Lorsque vous utilisez la ligne de commande Windows, remplacez la **$***variable* par la **%** *variable***%** pour les variables d'environnement et remplacez chaque barre oblique (**/**) par une barre oblique inverse (\) dans les noms de chemin des répertoires. Les noms des variables d'environnement ne sont pas toujours les mêmes dans les environnements Windows et UNIX. Par exemple, %TEMP% dans les environnements Windows est équivalent à $TMPDIR dans les environnements UNIX.

Remarque : Si vous utilisez le shell bash sur un système Windows, vous pouvez vous servir des conventions UNIX.

Partie 1. Configuration et utilisation de l'outil Fédération - Premiers pas

Les rubriques de la section Configuration fournissent des instructions détaillées pour la configuration de la fonctionnalité Fédération - Premiers pas. La console de gestion est dotée d'assistants qui vous guident dans de nombreuses tâches de configuration.

Vous pouvez utiliser l'outil Fédération - Premiers pas pour créer une fédération SAML 2.0 sur la base d'un modèle et de votre rôle préféré.

Commencez par la rubrique Chapitre 1, «Personnalisation des modèles de fédération», à la page 3.

Chapitre 1. Personnalisation des modèles de fédération

L'outil Fédération - Premiers pas contient des modèles que vous pouvez utiliser pour créer des fédérations SAML 2.0. Les modèles de fédération sont des fichiers de réponses comportant des macros qui sont développées au cours de l'exécution. Vous pouvez modifier ces modèles afin de personnaliser certaines propriétés.

Pourquoi et quand exécuter cette tâche

Avant d'utiliser l'outil Fédération - Premiers pour créer une fédération, vous avez la possibilité de personnaliser les modèles. Les sections suivantes expliquent comment procéder.

- Personnalisation des modèles de fédération
- Utilisation d'un modèle de fédération personnalisé

Personnalisation d'un modèle de fédération

Pourquoi et quand exécuter cette tâche

Il existe deux moyens de personnaliser un modèle de fédération :

- Modification du modèle de fédération dans le répertoire `fedfirststeps`
- Modification du modèle de fédération dans un autre répertoire

Modification du modèle de fédération dans le répertoire fedfirststeps

Procédure

1. Accédez au `<dossier d'installation de FIM >/firststeps/fedfirststeps/templates` et recherchez le modèle que vous souhaitezpersonnaliser.
2. Modifiez le modèle à personnaliser à l'aide d'un éditeur de texte.
3. Modifiez les variables souhaitées. Vous pouvez indiquer une autre valeur pour le nom de l'entreprise, la règle de mappage, etc.
4. Cliquez sur **Sauvegarder**.

Modification du modèle de fédération dans un autre répertoire

Procédure

1. Accédez au `<dossier d'installation de FIM >/firststeps/fedfirststeps/templates` et recherchez le modèle que vous souhaitez personnaliser.
2. Si vous devez déplacer le dossier des modèles, copiez-le à partir du `<répertoire d'installation de FIM>/firststeps/fedfirststeps/templates`. Placez-le dans un autre répertoire et renommez-le.
3. Personnalisez le modèle à l'aide d'un éditeur de texte.
4. Modifiez les variables souhaitées. Vous pouvez indiquer une autre valeur pour le nom de l'entreprise, la règle de mappage, etc.
5. Cliquez sur **Sauvegarder**.

Utilisation d'un modèle de fédération personnalisé

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser le modèle de fédération personnalisé de deux façons dans l'outil Fédération - Premiers pas : soit vous modifiez le fichier fedfirststep.ini, soit vous employez l'interface de ligne de commande. Si, par exemple vos modèles personnalisés se trouvent dans le répertoire `c:\custom_tfim_fed_templates\`, procédez comme suit :

Procédure

- Modifiez le fichier fedfirststep.ini, puis exécutez l'outil Fédération - Premiers pas.
  1. Accédez au `répertoire d'installation FIM/tools/fedfirststeps/fedfirststeps.ini` et ouvrez le fichier fedfirststep.ini dans un éditeur de texte.
  2. Ajoutez **`-custom-template-dir`**.
  3. Cliquez sur **Sauvegarder**.
  4. Lancez l'outil Fédération - Premiers pas.
- Utilisez l'interface de ligne de commande
  1. Ouvrez l'interface de ligne de commande.
  2. Entrez :
     - **`fedfirststeps.exe -custom-template-dir`**
     - `c:\custom_tfim_fed_templates`

Chapitre 2. Outil Fédération - Premiers pas

Utilisez l'outil Fédération - Premiers pas pour créer une fédération SAML générique, configurer l'accès basé sur les risques ou ajouter des fournisseurs de services en tant que partenaires.

L'outil Fédération - Premiers pas comporte les limites suivantes :

- Son interface utilisateur ne prend pas en charge l'encapsulage de l'intitulé de la case à cocher. Le texte ne s'affiche pas sur la ligne suivante et est tronqué en raison d'un incident Standard Widget Toolkit connu.
- L'utilisation de l'outil Fédération - Premiers pas dans des environnements en cluster n'est pas prise en charge.

Lancement de l'outil Fédération - Premiers pas

Lancez l'outil Fédération - Premiers pas pour créer des fédérations.

Avant de commencer

- Pour les utilisateurs Linux et Solaris, assurez-vous que vous exécutez GIMP Toolkit (GTK) version 2.12.0 ou ultérieure.
- Pour les utilisateurs AIX, assurez-vous que vous disposez de la dernière version de `motif`.

Procédure

Sélectionnez l'une des options suivantes pour lancer l'outil Fédération - Premiers pas.

- Dans l'assistant d'installation, la case **Lancer la console Premiers pas** est cochée par défaut. Cliquez sur **Terminer**.

  Remarque : Cette option est uniquement disponible sur une nouvelle installation de Tivoli Federated Identity Manager.
- Dans le répertoire `fedfirststeps`, lancez `fedfirststeps.exe` ou `fedfirststeps`.
- Exécutez les commandes suivantes dans l'interface de ligne de commande :
  - Microsoft Windows

    ```
    $FIM_INSTALL_DIR\firststeps\fedfirststeps\fedfirststeps.exe
    ```
  - UNIX

    ```
    $FIM_INSTALL_DIR/firststeps/fedfirststeps/fedfirststeps
    ```

    Remarque : Si vous utilisez un système d'exploitation 64 bits et que l'outil Fédération - Premiers pas ne démarre pas, vérifiez que le contenu du fichier `$FIM_INSTALL_DIR/firststeps/fedfirststeps.ini` est bien le suivant :

    ```
    -vm
    $FIM_INSTALL_DIR/_uninst/_jvm/bin
    -nl
    fr_FR
    ```

Configuration côté fournisseur d'identité

Configurez les paramètres de fédération du fournisseur d'identité à l'aide de l'outil Fédération - Premiers pas.

Création d'une fédération SAML 2.0 générique avec un nouveau domaine ou un domaine existant

L'assistant Fédération - Premiers pas crée une fédération générique qui permet aux utilisateurs de Tivoli Federated Identity Manager d'accéder aux applications du fournisseur de services.

Avant de commencer

Vous devez fournir les informations suivantes pour pouvoir exécuter l'assistant :

- L'option de clé de signature. Voir Stockage et gestion des clés et certificats.
- Le nom du domaine. Voir Configuration de domaine.
- Le serveur point de contact. Voir Gestion des serveurs point de contact.

Procédure

1. Lancez l'outil Fédération - Premiers pas.
2. Sélectionnez **Assistant SAML 2.0**.
3. Cliquez sur **Démarrer**. L'outil analyse les paramètres de la configuration existante.
4. Indiquez les informations demandées par l'assistant.
5. Cliquez sur **Terminer** pour lancer le traitement de la tâche. Cliquez sur **Précédent** pour effectuer des modifications.
6. Facultatif : Activez le domaine local.
   a. Connectez-vous à la console Integrated Solutions Console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Domaines** Une invite concernant le domaine local détecté s'affiche.
   c. Cliquez sur **OK** pour activer le domaine local.
7. Facultatif : Vérifiez les détails de la fédération que vous avez créée.
   a. Connectez-vous à la console Integrated Solutions Console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**.
   c. Le panneau Fédérations présente une liste des fédérations configurées. Sélectionnez la nouvelle fédération créée.
   d. Cliquez sur **Propriétés**.
   e. Sélectionnez les propriétés à modifier. Les propriétés de la fédération sont décrites dans l'aide en ligne.
   f. Cliquez sur **OK** pour fermer le panneau Propriétés de la fédération.

Configuration de l'accès basé sur les risques à l'aide de l'outil Fédération - Premiers pas

Utilisez l'outil Fédération - Premiers pas d'IBM Tivoli Federated Identity Manager pour configurer et activer l'accès basé sur les risques. L'accès basé sur les risques est un composant d'IBM Tivoli Federated Identity Manager. Il permet de déterminer le choix des accès et de les appliquer en fonction de l'évaluation des risques ou du niveau de confiance d'une transaction.

Avant de commencer

Avant de lancer l'outil Fédération - Premiers pas, procédez comme suit :

1. Si vous disposez d'un environnement en cluster WebSphere Application Server, vous devez créer et configurer un contexte JNDI nommé `jdbc/rba` dans WebSphere Application Server et créer le schéma de base de données pour l'accès basé sur les risques. Voir Manually configuring the database.
2. Installez l'accès basé sur les risques. Voir Installing risk-based access.
3. Configurez le serveur point de contact WebSEAL pour IBM Tivoli Federated Identity Manager. Voir Configuration d'un serveur point de contact WebSEAL pour la fédération SAML.

Vous devez fournir les informations suivantes pour pouvoir exécuter l'assistant :

- URL du serveur point de contact d'IBM Tivoli Federated Identity Manager
- URI de la ressource sécurisée IBM Tivoli Access Manager que vous souhaitez protéger avec l'accès basé sur les risques
- Nom de l'instance WebSEAL

Procédure

1. Lancez l'outil Fédération - Premiers pas.
2. Sélectionnez **Risk-based Access Configuration Wizard**.
3. Cliquez sur **Démarrer**. L'outil analyse les paramètres de la configuration existante.
4. Indiquez les informations demandées par l'assistant.
5. Facultatif : Sur la page General Configuration Settings, sélectionnez **Configure Tivoli Access Manager**, si vous souhaitez configurer l'environnement IBM Tivoli Access Manager afin qu'il délègue les décisions d'autorisation à l'accès basé sur les risques pour vos ressources sécurisées.

   Remarque : Si vous sélectionnez cette option, vous devez vous assurer qu'IBM Tivoli Access Manager est installé et configuré localement sur le même système qu'IBM Tivoli Federated Identity Manager.
6. Indiquez l'URL du serveur point de contact d'IBM Tivoli Federated Identity Manager qui est utilisé pour la collecte d'attributs.

   `http://host_name/webseal-junction-name`

   Par exemple :

   `http://mywebsealhost.company.com/FIM`

   A l'issue du processus de configuration, la page Risk-based Access Configuration Summary indique si la configuration a échoué ou abouti.
   - Si la configuration a abouti, les étapes qui restent à exécuter pour terminer la configuration de l'accès basé sur les risques s'affichent sur la page Risk-based Access Configuration Summary. Suivez les instructions indiquées sur la page récapitulative pour terminer la configuration d'IBM Tivoli Access Manager et d'EAS (External Authorization Service) pour votre environnement.
   - Si la configuration a échoué, les messages d'erreur et du journal s'affichent sur la page Risk-based Access Configuration Summary. Utilisez les détails fournis dans les messages d'erreur et de journaux pour identifier l'échec du processus de configuration et la cause probable de l'échec. Corrigez les erreurs de configuration et exécutez de nouveau l'outil Fédération - Premiers pas pour configurer l'accès basé sur les risques.
7. Cliquez sur **Terminer**.

Que faire ensuite

Une fois que vous avez terminé toutes les étapes suivantes spécifiées sur la page récapitulative, vérifiez que l'accès basé sur les risques est correctement configuré sur votre système. Voir Verifying the configuration.

Ajout d'un fournisseur de services à l'aide de l'outil Fédération - Premiers pas

L'assistant Fédération - Premiers pas ajoute un fournisseur de services en tant que partenaire de sorte que les utilisateurs de Tivoli Federated Identity Manager puissent accéder aux applications du fournisseur de services. Les fournisseurs de services pris en charge sont Salesforce, Google Apps, Microsoft Office 365 et Workday.

Avant de commencer

Vous devez fournir les informations suivantes pour pouvoir exécuter l'assistant :

- Noms de partenaire et de domaine du fournisseur de services que vous souhaitez ajouter en tant que partenaire.
- Nom de la fédération
- Noms des utilisateurs fédérés
- Si vous disposez de plusieurs noms de domaine dans Google Apps, l'émetteur dans la requête SAML doit être défini sur `google.com/a/example.com` et non sur `google.com`. Cette option doit correspondre à l'option de connexion unique dans Google Apps.
- Option de clé de signature
- Serveur point de contact

Pourquoi et quand exécuter cette tâche

L'assistant Fédération - Premiers pas ajoute un fournisseur de services en tant que partenaire dans les cas suivants :

- Ajout à un domaine existant ou une fédération existante
- Ajout à un domaine existant et à une nouvelle fédération
- Ajout à un nouveau domaine et à une nouvelle fédération

Procédure

1. Lancez l'outil Fédération - Premiers pas.
2. Sélectionnez le plug-in spécifique au fournisseur de services que vous souhaitez utiliser.
3. Cliquez sur **Démarrer**. L'outil analyse les paramètres de la configuration existante.
4. Indiquez les informations demandées par l'assistant.
5. Cliquez sur **Terminer** pour lancer le traitement des tâches. Cliquez sur **Précédent** pour effectuer des modifications.
6. Facultatif : Vérifiez les détails de la fédération que vous avez créée.
   a. Connectez-vous à la console Integrated Solutions Console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**.
   c. Le panneau Fédérations présente une liste des fédérations configurées. Sélectionnez la nouvelle fédération créée.

d. Cliquez sur **Propriétés**.
e. Sélectionnez les propriétés à modifier. Les propriétés de la fédération sont décrites dans l'aide en ligne.
f. Cliquez sur **OK** pour fermer le panneau Propriétés de la fédération.

7. Facultatif : Vérifiez les détails du partenaire que vous avez créé.
   a. Connectez-vous à la console Integrated Solutions Console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la connexion unique fédérée** > **Partenaires**.
   c. Le panneau Partenaires affiche une liste des partenaires configurés. Sélectionnez le nouveau partenaire créé.
   d. Cliquez sur **Propriétés**.
   e. Sélectionnez les propriétés à modifier. Les propriétés du partenaire sont décrits dans l'aide en ligne.
   f. Cliquez sur **OK** pour fermer le panneau Propriétés du partenaire.
8. Facultatif : Activez le domaine local.
   a. Connectez-vous à la console Integrated Solutions Console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Domaines** Une invite concernant le domaine local détecté s'affiche.
   c. Cliquez sur **OK** pour activer le domaine local.

Concepts associés:

«Présentation de la stratégie UPN et immutableID pour Microsoft Office 365», à la page 12
Vous devez choisir une stratégie pour ImmutableID avant de configurer les paramètres de connexion unique pour Microsoft Office 365. Les utilisateurs Microsoft Office 365 sont identifiés par le nom principal d'utilisateur (UPN) et la valeur ImmutableID.

Configuration côté fournisseur de services

Configurez les paramètres de fédération du fournisseur de services à l'aide de l'outil Fédération - Premiers pas.

Remarque : Les instructions de configuration relatives à ces fournisseurs de services peuvent changer. Pour connaître les mises à jour, consultez la documentation du fournisseur de services.

Plug-in Premiers pas pour Google Apps

Utilisez le plug-in Premiers pas pour créer une fédération avec Google Apps.

1. Sélectionnez la configuration appropriée pour votre fournisseur d'identité. Pour plus d'informations, voir «Configuration côté fournisseur d'identité», à la page 5.
2. «Configuration des paramètres de connexion unique Google Apps»
3. «Ajout d'utilisateurs à Google Apps», à la page 10
4. «Test de la connexion unique à Google Apps», à la page 11

Configuration des paramètres de connexion unique Google Apps

Configurez les paramètres de connexion unique Google Apps pour activer l'authentification d'utilisateur.

Avant de commencer

La configuration nécessite que vous fournissiez un certificat qui est utilisé pour signer le message SAML dans Federated Identity Manager. Exportez votre certificat Federated Identity Manager en utilisant un format PEM (Privacy-Enhanced Message). Voir la rubrique relative à l'exportation d'un format dans le document *IBM Tivoli Federated Identity Manager Configuration Guide*.

Procédure

1. Accédez au site Web de votre fournisseur de services.
   a. Ouvrez un navigateur Web.
   b. Entrez l'URL fournie par Google pour accéder à votre compte. Par exemple, `https://www.google.com/a/example.com`.
2. Connectez-vous à l'aide de vos données d'identification.
3. Accédez à la page de configuration de la connexion unique.
   a. Cliquez sur **Outils avancés**.
   b. Sélectionnez **Configuration de la connexion unique**.
4. Sélectionnez le paramètre **Activer la connexion unique**.
5. Configurez les paramètres de connexion unique en indiquant les informations suivantes :

   URL de la page de connexion
   Entrez l'URL de noeud final de connexion Federated Identity Manager. Par exemple, `https://idp.example.com/FIM/sps/<federation name>/saml20/login`

   URL de la page de déconnexion
   Entrez l'URL permettant de rediriger les utilisateurs lorsqu'ils se déconnectent.

   URL de changement de mot de passe
   Indiquez l'URL permettent aux utilisateurs de changer leur mot de passe dans votre système.
6. Téléchargez le certificat de vérification que vous avez exporté au début de cette tâche dans la zone. Ce certificat doit contenir la clé publique de la paire de clés qui est utilisée pour signer les messages SAML dans Federated Identity Manager.
7. Sauvegardez vos paramètres.

Que faire ensuite

Testez la connexion unique sur Google Apps.

Ajout d'utilisateurs à Google Apps

Ajoutez des utilisateurs à Google Apps pour qu'ils puissent être authentifiés via la connexion unique.

Procédure

1. Connectez-vous à Google Apps.
2. Cliquez sur **Organisation & utilisateurs**.
3. Cliquez sur **Créer un nouvel utilisateur**.
4. Spécifiez les informations requises concernant votre utilisateur.
5. Suivez les instructions affichées à l'écran pour exécuter les étapes d'ajout d'un utilisateur.

Résultats

Les utilisateurs sont ajoutés dans Google Apps.

Test de la connexion unique à Google Apps

Testez l'authentification de connexion unique à Google Apps après avoir exécuté toutes les étapes de configuration de domaine et de fédération.

Avant de commencer

Vérifiez que vous avez ajouté des utilisateurs dans Google Apps et Federated Identity Manager. Pour plus d'informations sur l'ajout d'utilisateurs dans Google Apps, voir «Ajout d'utilisateurs à Google Apps», à la page 10. Pour plus d'informations sur l'ajout d'utilisateurs pour la connexion unique dans Federated Identity Manager, voir *IBM Tivoli Federated Identity Manager Configuration Guide*.

Pourquoi et quand exécuter cette tâche

Les étapes suivantes fournissent les instructions relatives au test de la connexion unique.

Procédure

Lancez la connexion unique.

- **Pour lancer la connexion unique depuis le fournisseur d'identité :**
  1. Accédez au noeud final initial de connexion du fournisseur d'identité.
  2. Indiquez `google.com` pour `PartnerId`.
  3. Indiquez n'importe quelle chaîne pour la cible. Par exemple, `https://idp.example.com/sps/<federationName>/saml20/logininitial?PartnerId=google.com&Target=<anystring>`. Vous êtes redirigé vers la page de connexion du fournisseur d'identité.

     Remarque : Si pour la cible, vous ne spécifiez pas de valeur ou vous indiquez une chaîne vide, Google affiche un message d'erreur indiquant que le paramètre de réponse requis **`RelayState`** est manquant.
  4. Connectez-vous à l'aide de l'ID utilisateur et du mot de passe de l'utilisateur que vous testez.
- **Pour lancer une connexion unique à partir de Google :**
  1. Accédez à l'URL de ressource protégée. Par exemple, `https://drive.google.com/a/example.com`. Vous êtes redirigé vers la page de connexion du fournisseur d'identité.
  2. Connectez-vous à l'aide de l'ID utilisateur et du mot de passe de l'utilisateur que vous testez.

Résultats

L'utilisateur peut accéder à la ressource protégée dans Google Apps.

Plug-in Premiers pas pour Microsoft Office 365

Utilisez le plug-in Premiers pas pour créer une fédération avec Microsoft Office 365. L'utilisation de Windows PowerShell for single sign-on afin d'ajouter des utilisateurs n'est pas prise en charge par IBM®.

1. «Présentation de la stratégie UPN et immutableID pour Microsoft Office 365»
   a. «Remplissage du service d'alias Federated Identity Manager», à la page 13
   b. «Envoi d'une demande à Tivoli Federated Identity Manager Security Token Service», à la page 15
2. Sélectionnez la configuration appropriée pour votre fournisseur d'identité. Pour plus d'informations, voir «Configuration côté fournisseur d'identité», à la page 5.
3. «Configuration des paramètres de connexion unique Microsoft Office 365», à la page 16
4. «Ajout d'utilisateurs à Microsoft Office 365», à la page 17
5. «Test de la connexion unique à Microsoft Office 365», à la page 17

Présentation de la stratégie UPN et immutableID pour Microsoft Office 365

Vous devez choisir une stratégie pour ImmutableID avant de configurer les paramètres de connexion unique pour Microsoft Office 365. Les utilisateurs Microsoft Office 365 sont identifiés par le nom principal d'utilisateur (UPN) et la valeur ImmutableID.

UPN L'UPN est le nom d'utilisateur de compte local qui est ajouté à `@domainname` pour un domaine enregistré que vous possédez. L'outil Fédération - Premiers pas génère automatiquement la règle de mappage qui mappe le nom d'utilisateur du compte local au format UPN.

ImmutableID
Identificateur unique non recyclé pour le compte. Il doit être transmis en tant qu'attribut dans l'assertion SAML au cours de la connexion unique à Microsoft Office 365.

Vous devez déterminer la valeur ImmutableID et son emplacement.

Utilisation de l'identificateur unique universel principal Tivoli Access Manager comme valeur ImmutableID
Cette technique s'applique uniquement si vous utilisez Tivoli Access Manager WebSEAL en tant que point de contact pour Federated Identity Manager. Elle s'applique lors de l'authentification d'utilisateur standard à partir du registre Tivoli Access Manager.

Tivoli Access Manager attribue un ID utilisateur universel principal à chaque compte utilisateur. Cet ID est utilisé lors de l'ajout d'utilisateurs et lors de la connexion unique avec WS-Federation dans Federated Identity Manager.

L'avantage lié à l'utilisation de cette technique réside dans le fait que lors de la connexion unique pendant la phase d'exécution, l'identificateur unique universel figure déjà dans les données d'identification Tivoli Access Manager. Il n'est pas nécessaire d'écrire une règle de mappage Federated Identity Manager pour le récupérer et l'insérer dans l'assertion SAML.

L'inconvénient de cette technique réside dans le fait qu'elle ne vous permet pas de déterminer si un utilisateur a été ajouté à Microsoft Office 365 avant de tenter la connexion unique.

L'ajout d'utilisateurs doit être effectué avant toute tentative de connexion unique. Sinon, la connexion unique échoue. Pour plus d'informations, voir «Ajout d'utilisateurs à Microsoft Office 365», à la page 17.

Le scénario ci-après illustre une opération `ldapsearch` pour un identificateur unique universel principal Tivoli Access Manager. Par exemple, `secUUID`. Le domaine Tivoli Access Manager est `default`. L'utilisateur est `jane`. Entrez la commande sur une seule ligne.

```
/opt/IBM/ldap/V6.1/bin/idsldapsearch -L -h localhost -p 389 -D
cn=root -w <your_ldap_pwd> -b "cn=users,secauthority=default"
-s one "(&(objectclass=secUser)(principalName=jane))"
```

Le code suivant est un exemple de résultat :

```
....
secDN: principalName=jane,cn=Users,secAuthority=Default
secUUID: a1a2a3a4-b1b2-c1c2-1111-000000000000
....
```

Utilisation du service d'alias Federated Identity Manager pour gérer ImmutableID

Cette technique requiert un service d'alias configuré pour Federated Identity Manager. Pour plus d'informations sur le service d'alias, voir le centre de documentation d'IBM Tivoli Federated Identity Manager.

Si Tivoli Access Manager n'est pas votre point de contact ou si vous ne souhaitez pas utiliser l'identificateur unique universel principal Tivoli Access Manager comme valeur ImmutableID Microsoft Office 365, stockez et gérez un autre identificateur unique universel.

Federated Identity Manager fournit un service d'alias générique. Il est utilisé dans les fédérations SAML 2.0 avec des identificateurs de nom persistants.L'interface de programmes avec le service d'alias Federated Identity Manager est prise en charge via les API de la classe **`IDMappingExtUtils`**. La classe peut être appelée à partir des règles de mappage Javascript et Java™ dans Federated Identity Manager Security Token Service.

Pour remplir le service d'alias, vous devez créer une chaîne d'accréditation STS et appeler ces API. Voir «Remplissage du service d'alias Federated Identity Manager».

Utilisation de la valeur codée en base64 de l'UPN

Avec cette technique, la valeur ImmutableID qui est envoyée à Microsoft Office 365 est la valeur codée en base64 de l'UPN.

Remarque : Utilisez cette méthode uniquement dans un environnement de test et non dans un environnement de production.

Remplissage du service d'alias Federated Identity Manager :

Pour utiliser le service d'alias Federated Identity Manager pour gérer la valeur ImmutableID, commencez par le remplir.

Procédure

1. Créez une chaîne STS avec la structure suivante :
   - **STSUU par défaut** (valider)
   - **Mappage par défaut** (mapper)
   - **STSUU par défaut** (émettre)
2. Définissez les adresses suivantes :
   - **Adresse Appliquer à** `http://appliesto/alias`
   - **Adresse de l'émetteur** `http://issuer/alias`

3. Utilisez la règle de mappage JavaScript suivante pour le module de mappage par défaut :

```
// BEGIN Javascript mapping rule
// mapping rule for performing simple store and fetch alias operations
importPackage(Packages.com.tivoli.am.fim.trustserver.sts);
importPackage(Packages.com.tivoli.am.fim.trustserver.sts.utilities);
importPackage(Packages.com.tivoli.am.fim.trustserver.sts.uuser);

// Figure out the "operation" being performed
var operation = stsuu.getAttributeValueByName("operation");

// store operation
if (operation != null && operation == "store") {
 // get username, federation ID, and alias value to store
 var username = stsuu.getPrincipalName();
 var federationid = stsuu.getAttributeValueByName("federationID");
 var alias = stsuu.getAttributeValueByName("alias");

 if (username != null && federationid != null && alias != null) {
  // store it, first removing any existing values
  var existingAliasValues =
    IDMappingExtUtils.lookupAliasesForUserAsStringArray(
      federationid, username);
  if (existingAliasValues != null) {
   for (var i = 0; i < existingAliasValues.length; i++) {
    IDMappingExtUtils.removeAliasForUser(
      federationid, username, existingAliasValues[i]);
   }
  }
  IDMappingExtUtils.addAliasForUser(federationid, username, alias);
 }
}

// fetch operation
if (operation != null && operation == "fetch") {
 // get username, federation ID
 var username = stsuu.getPrincipalName();
 var federationid = stsuu.getAttributeValueByName("federationID");

 if (username != null && federationid != null) {
  // fetch alias value(s) and put in STSUU
  var existingAliasValues =
    IDMappingExtUtils.lookupAliasesForUserAsStringArray(
      federationid, username);
  var attr = new Attribute("alisValues", "", existingAliasValues);
  stsuu.addAttribute(attr);
 }
}

// delete operation
if (operation != null && operation == "delete") {
 // get username, federation ID
 var username = stsuu.getPrincipalName();
 var federationid = stsuu.getAttributeValueByName("federationID");

 if (username != null && federationid != null) {
  // remove any existing values
  var existingAliasValues =
    IDMappingExtUtils.lookupAliasesForUserAsStringArray(
      federationid, username);
  if (existingAliasValues != null) {
   for (var i = 0; i < existingAliasValues.length; i++) {
    IDMappingExtUtils.removeAliasForUser(
      federationid, username, existingAliasValues[i]);
   }
```

```
  }
 }
}
// END Javascript mapping rule
```

4. Ajoutez un alias pour un utilisateur. Voir «Envoi d'une demande à Tivoli Federated Identity Manager Security Token Service».

Envoi d'une demande à Tivoli Federated Identity Manager Security Token Service :

L'utilitaire `curl` vous permet d'envoyer des demandes à Tivoli Federated Identity Manager Security Token Service pour mettre à disposition un alias utilisateur.

Procédure

1. Enregistrez le message `RequestSecurityToken` 1.2 suivant dans un fichier. Par exemple, `rst.xml`.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <wst:RequestSecurityToken xmlns:wst=
"http://schemas.xmlsoap.org/ws/2005/02/trust">
      <wst:RequestType xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust
">http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
      <wst:Issuer xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
        <wsa:Address xmlns:wsa=
"http://schemas.xmlsoap.org/ws/2004/08/addressing">
http://issuer/alias</wsa:Address>
      </wst:Issuer>
      <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference xmlns:wsa=
"http://schemas.xmlsoap.org/ws/2004/08/addressing">
          <wsa:Address>http://appliesto/alias</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:Base xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
<stsuuser:STSUniversalUser
        xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser">
        <stsuuser:Principal>
                <stsuuser:Attribute name="name"
                        type="urn:ibm:names:ITFIM:5.1:accessmanager">
                        <stsuuser:Value>jane</stsuuser:Value>
                </stsuuser:Attribute>
        </stsuuser:Principal>
        <stsuuser:AttributeList>
                <stsuuser:Attribute name="operation">
                        <stsuuser:Value>store</stsuuser:Value>
                </stsuuser:Attribute>
                <stsuuser:Attribute name="federationID">
                        <stsuuser:Value>urn:federation:
MicrosoftOnline</stsuuser:Value>
                </stsuuser:Attribute>
                <stsuuser:Attribute name="alias">
                        <stsuuser:Value>myalias</stsuuser:Value>
                </stsuuser:Attribute>
        </stsuuser:AttributeList>
</stsuuser:STSUniversalUser>
      </wst:Base>
    </wst:RequestSecurityToken>
  </soapenv:Body>
</soapenv:Envelope>
```

2. Remplacez vos propres valeurs pour `username` et `alias`, ce qui correspond à l'attribut `ImmutableID` pour chaque utilisateur que vous souhaitez mettre à disposition. Dans l'exemple, la valeur de `username` est `jane` et la valeur de `alias` est `myalias`.
3. L'utilitaire `curl` vous permet d'envoyer la demande au noeud final WS-Trust 1.2. Par exemple :

```
curl --header "soapaction: blah"
--header "Content-Type: text/xml" \
    --data-binary @rst.xml http://localhost:9080
/TrustServer/SecurityTokenService
```

Configuration des paramètres de connexion unique Microsoft Office 365

La configuration des paramètres de connexion unique Microsoft Office 365 implique l'installation de Windows PowerShell for Single sign-on, l'exécution de l'outil de ligne de commande et la confirmation de l'activation de votre domaine fédéré.

Procédure

1. Installez Windows PowerShell for single sign-on. Pour plus d'informations, voir Microsoft Online and search for *Windows PowerShell for single sign-on*.
2. Exécutez Windows PowerShell for single sign-on et créez un domaine fédéré.
   a. Connectez-vous au service en ligne MS en tapant la commande suivante :
      ```
      Connect-MsolService
      ```
   b. Créez un nouveau domaine fédéré en tapant la commande suivante :
      ```
      New-MsolDomain -authentication federated -DomainName example.com
      ```
   c. Obtenez le serveur de noms de domaine de vérification de domaine en tapant la commande suivante :
      ```
      Get-MsolDomainVerificationDns -DomainName example.com -Mode DnsTxtRecord
      ```
   d. Connectez-vous au registre de domaine où vous possédez le domaine.
   e. Ajoutez l'entrée `mx`.
   f. Confirmez le domaine et remplissez les propriétés de fédération.

      Dans l'exemple ci-après, le nom de fédération est `office365` et la jonction WebSEAL est `FIM`.

      ```
      Confirm-MsolDomain \
        -DomainName example.com \
        -FederationBrandName "Example, Inc" \
        -IssuerUri https://identityprovider.example.com/FIM/sps/office365/wsf \
        -LogOffUri https://profile.example.com/FIM/sps/office365/wsf \
        -PassiveLogOnUri https://profile.example.com/FIM/sps/office365/wsf  \
        -PreferredAuthenticationProtocol WsFed \
        -SigningCertificate MIICBz........Q==
      ```

      Remarque : Cette opération échoue si Microsoft Office 365 ne peut pas résoudre l'entrée `mx`. Attendez que la propagation DNS soit terminée.
3. Confirmez l'activation de votre domaine fédéré.
   a. Accédez à `https://portal.microsoftonline.com`.
   b. Connectez-vous à votre compte d'administration.
   c. Accédez à **Vue d'ensemble de l'administration** > **Gestion** > **Domaines**.
   d. Vérifiez que votre domaine figure dans la liste des domaines et que l'état est **Vérifié**.

Ajout d'utilisateurs à Microsoft Office 365

Ajoutez des utilisateurs à Microsoft Office 365 pour qu'ils puissent être authentifiés via la connexion unique. L'utilisation de Windows PowerShell for single sign-on afin d'ajouter des utilisateurs n'est pas prise en charge par IBM.

Procédure

1. Connectez-vous au service en ligne MS en tapant la commande suivante :

   ```
   Connect-MsolService
   ```

2. Ajoutez des utilisateurs à l'aide de la commande **New-MsolUser**. Entrez la commande sur une seule ligne.

   ```
   New-MsolUser -userprincipalname $upn \
                -immutableID $base64 \
       -lastname $sn \
       -firstname $gn \
       -Displayname $displayName \
       -BlockCredential $false \
       -LicenseAssignment $license \
       -usageLocation $usagelocation
   ```

   Remarque : Utilisez la commande **Get-MsolAccountSku** pour connaître les valeurs que vous pouvez définir pour **LicenseAssignment**. Pour plus d'informations, voir Windows PowerShell cmdlets for Microsoft Office 365.

Test de la connexion unique à Microsoft Office 365

Testez l'authentification de connexion unique à Microsoft Office 365 après avoir exécuté les étapes de configuration de domaine et de fédération pour vérifier que cela fonctionne correctement.

Avant de commencer

Vérifiez que vous avez ajouté des utilisateurs dans Microsoft Office 365 et Federated Identity Manager. Pour plus d'informations sur l'ajout d'utilisateurs dans Microsoft Office 365, voir «Ajout d'utilisateurs à Microsoft Office 365». Pour plus d'informations sur l'ajout d'utilisateurs pour la connexion unique dans Federated Identity Manager, voir *IBM Tivoli Federated Identity Manager Configuration Guide*.

Procédure

1. Assurez-vous qu'aucune session de navigation n'est active dans le fournisseur d'identité et Microsoft Office 365.
2. Accédez à `https://portal.microsoftonline.com/`.
3. Entrez l'ID utilisateur fourni sous le domaine fédéré dans la zone **ID utilisateur**. Par exemple, `john@example.com`. L'écran est mis à jour pour indiquer que vous n'avez pas besoin de spécifier un mot de passe.
4. Cliquez sur **Se connecter à <nom de domaine>**. Vous êtes redirigé vers la page de connexion de votre fournisseur d'identité.
5. Connectez-vous à l'aide de vos données d'identification.

Résultats

Vous êtes redirigé vers et connecté à Microsoft Office 365.

Plug-in Premiers pas pour Salesforce

Utilisez le plug-in Premiers pas pour créer une fédération avec Salesforce.

1. Sélectionnez la configuration appropriée pour votre fournisseur d'identité. Pour plus d'informations, voir Configuration côté fournisseur d'identité.
2. «Configuration des paramètres de connexion unique Salesforce»
3. «Test de la connexion unique à Salesforce», à la page 19

Configuration des paramètres de connexion unique Salesforce

Configurez les paramètres de connexion unique Salesforce pour activer l'authentification d'utilisateur.

Avant de commencer

La configuration nécessite que vous fournissiez un certificat qui est utilisé pour signer le message SAML dans Federated Identity Manager. Exportez votre certificat Federated Identity Manager en utilisant un format PEM (Privacy-Enhanced Message). Voir la rubrique relative à l'exportation d'un format dans le document *IBM Tivoli Federated Identity Manager Configuration Guide*.

Procédure

1. Accédez au site Web de votre fournisseur de services.
   a. Ouvrez un navigateur Web.
   b. Entrez l'URL fournie par Salesforce pour accéder à votre compte. Par exemple, `https://www.salesforce.com`.
2. Connectez-vous à l'aide de vos données d'identification.
3. Accédez à la page de configuration de la connexion unique.
   a. Cliquez sur votre nom de compte pour afficher le menu utilisateur.
   b. Sélectionnez **Configurer**
4. Configurez les paramètres de connexion unique.
   a. Sous **Configuration de l'administration**, sélectionnez **Contrôles de sécurité** > **Paramètres de connexion unique**.
   b. Cliquez sur **Editer**.
   c. Cochez la case **SAML activé**.
5. Fournissez les informations suivantes :

   SAML activé
   Vous devez activer cette option.

   Version SAML
   Le plug-in prend en charge SAML 1.1.

   Emetteur
   Cette option correspond à l'URL de noeud final Federated Identity Manager. Par exemple, `https://idp.example.com/sps/<federation name>/saml11`. Cette valeur doit correspondre à la valeur `Emetteur` dans l'assertion SAML. La valeur `Emetteur` provient de la valeur d'ID fournisseur qui est définie dans les paramètres de fédération SAML 1.1 du fournisseur d'identité.

   Certificat de fournisseur d'identité
   Cette option correspond au certificat Federated Identity Manager qui a été exporté au début de cette tâche. Il s'agit du certificat public de la clé permettant de signer les messages SAML. Il doit s'agir d'un fichier certificat PEM téléchargé contenant la clé publique qui correspond au certificat de signature au niveau du fournisseur d'identité Federated Identity Manager.

Type d'ID utilisateur SAML
Sélectionnez la première option si vous souhaitez utiliser les identités sur votre site Web de fournisseur d'identité. Sinon, sélectionnez **L'assertion contient l'ID fédération de l'objet utilisateur**. Vous devez entrer un ID fédération pour chaque utilisateur dans le menu **Gérer les utilisateurs**.

Emplacement d'ID utilisateur SAML
L'emplacement d'ID utilisateur figure dans l'élément `NameIdentifier` de l'instruction d'objet. Sélectionnez la seconde option si vous souhaitez utiliser des scénarios de mappage utilisateur avancés. Vous pouvez également sélectionner la seconde option si vous souhaitez que la valeur soit lue à partir d'un attribut nommé dans l'instruction `AttributeStatement` de l'assertion SAML.

6. Sauvegardez vos paramètres.

Test de la connexion unique à Salesforce

Testez l'authentification de connexion unique à Salesforce après avoir exécuté toutes les étapes de configuration de domaine et de fédération.

Avant de commencer

Vérifiez que vous avez ajouté des utilisateurs dans Salesforce et Federated Identity Manager. Pour plus d'informations sur l'ajout d'utilisateurs dans Salesforce, consultez la documentation Salesforce ou adressez-vous à votre administrateur système Salesforce. Pour plus d'informations sur l'ajout d'utilisateurs pour la connexion unique dans Federated Identity Manager, voir *IBM Tivoli Federated Identity Manager Configuration Guide*.

Procédure

1. Fermez les éventuelles sessions de navigation Identity Provider et Salesforce.
2. Accédez au noeud final initial de connexion du fournisseur d'identité. Vous êtes redirigé vers la page de connexion du fournisseur d'identité. Par exemple, `https://idp.example.com/FIM/sps/<fed name>/saml11/login?TARGET=https://saml.salesforce.com`
3. Entrez vos données d'identification.

Résultats

Vous êtes redirigé vers et connecté à Salesforce.

Plug-in Premiers pas pour Workday

Utilisez le plug-in Premiers pas pour créer une fédération avec Workday.

1. Sélectionnez la configuration appropriée pour votre fournisseur d'identité. Pour plus d'informations, voir Configuration côté fournisseur d'identité.
2. «Configuration des paramètres de connexion unique Workday»
3. «Test de la connexion unique à Workday», à la page 20

Configuration des paramètres de connexion unique Workday

Configurez la configuration de sécurité Workday pour activer la connexion unique.

Avant de commencer

La configuration nécessite que vous fournissiez un certificat pour signer le message SAML dans Federated Identity Manager. Exportez votre certificat Federated Identity Manager en utilisant un format PEM (Privacy-Enhanced Message). Voir la rubrique relative à l'exportation d'un format dans le document *IBM Tivoli Federated Identity Manager Configuration Guide*.

Procédure

1. Accédez au site Web de votre fournisseur de services.
   a. Ouvrez un navigateur Web.
   b. Entrez l'URL fournie par Workday pour accéder à votre compte. Par exemple, `https://www.myworkday.com/<your company>/login.flex`.
2. Connectez-vous à votre compte d'administration.
3. Accédez à la page de configuration de la connexion unique.
   a. Cliquez sur **Plan de travail** > **Administration de compte** > **Editer la configuration des clients - Sécurité**
4. Configurez les paramètres de connexion unique en indiquant les informations suivantes :
   a. Sous **Configuration SAML**, sélectionnez l'option **Activer l'authentification SAML**.
   b. Spécifiez les informations suivantes :

      ID fournisseur d'identité
      Entrez l'URL de noeud final de connexion Federated Identity Manager. Par exemple, `https://idp.example.com/FIM/sps/<federation name>/saml20/login`

      Clé publique x509
      Téléchargez le certificat que vous avez exporté au début de cette tâche dans la zone. Ce certificat doit contenir la clé publique de la paire de clés qui est utilisée pour signer les messages SAML dans Federated Identity Manager.
5. Sauvegardez vos paramètres.

Que faire ensuite

Testez la connexion unique sur Workday.

Test de la connexion unique à Workday

Testez l'authentification de connexion unique à Workday après avoir exécuté toutes les étapes de configuration de domaine et de fédération.

Avant de commencer

Vérifiez que vous avez ajouté des utilisateurs dans Workday et Federated Identity Manager. Pour plus d'informations sur l'ajout d'utilisateurs dans Workday, consultez la documentation Workday ou adressez-vous à votre administrateur système Workday. Pour plus d'informations sur l'ajout d'utilisateurs pour la connexion unique dans Federated Identity Manager, voir le document *IBM Tivoli Federated Identity Manager Configuration Guide*.

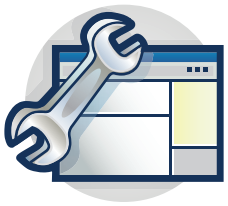
Procédure

1. Lancez la connexion unique en accédant au noeud final de connexion du fournisseur d'identité. Par exemple, `https://idp.example.com/FIM/sps/<fed name>/saml20/logininitial?PartnerId=http%3A%2F%2Fwww.workday.com`. Vous êtes redirigé vers la page de connexion du fournisseur d'identité.
2. Connectez-vous à l'aide de vos données d'identification.

Résultats

L'utilisateur peut accéder à la ressource protégée dans Workday.

Partie 2. Configuration d'un domaine



Les rubriques de la section Configuration vous guident pas à pas lors de la configuration d'un domaine. La console de gestion est dotée d'assistants qui vous guident dans de nombreuses tâches de configuration.

Tous les déploiements Tivoli Federated Identity Manager nécessitent le déploiement d'un domaine. Vous devez déployer un domaine avant de configurer d'autres fonctions telles que la fédération de connexion unique, la gestion de sécurité des services Web, les services de jeton ou User Self Care.

Démarrez par la rubrique :

- Chapitre 3, «Configuration de domaine», à la page 25

Chapitre 3. Configuration de domaine

Un domaine Tivoli Federated Identity Manager est un déploiement du composant d'exécution Tivoli Federated Identity Manager sur un serveur WebSphere unique ou sur un cluster WebSphere.

Il existe un domaine par cluster WebSphere. Un environnement comportant un serveur unique ne peut contenir qu'un seul domaine.

Chaque domaine est géré indépendamment. Vous pouvez utiliser l'installation de la console de gestion Tivoli Federated Identity Manager pour gérer plusieurs domaines. Vous ne pouvez gérer qu'un seul domaine à un moment donné. Le domaine à gérer est désigné par *domaine actif*.

Une fois que Tivoli Federated Identity Manager est installé, aucun domaine n'existe. Utilisez la console de gestion pour créer un domaine. Une fois Tivoli Federated Identity Manager installé, le service de gestion est déployé sur un serveur WebSphere (mode serveur unique) ou sur un gestionnaire de déploiement WebSphere (mode cluster WebSphere).

Connectez-vous au service de gestion et sélectionnez un serveur ou cluster WebSphere sur lequel déployer le composant d'exécution Tivoli Federated Identity Manager. Une fois que le module d'exécution est déployé et configuré, vous êtes prêt à configurer les fonctionnalités complémentaires telles que la connexion unique fédérée ou la gestion de la sécurité des services Web.

Dans un environnement WebSphere Network Deployment, le déploiement et la configuration du module d'exécution Tivoli Federated Identity Manager sur les membres du cluster constituent un processus automatisé. Il n'est pas nécessaire d'installer également les logiciels Tivoli Federated Identity Manager ou Tivoli Access Manager sur les ordinateurs WebSphere en cluster.

Le service de gestion Tivoli Federated Identity Manager utilise les services de déploiement d'application de WebSphere Deployment Manager pour déployer et configurer le module d'exécution sur les membres de cluster distribués.

La console de gestion offre un assistant qui vous guidera tout au long de la création du domaine. Les sections ci-après répertorient les propriétés que l'assistant vous invite à spécifier.

Propriétés des noeuds finals de service de gestion de domaine

Hôte Nom de domaine complet de l'**hôte** sur lequel WebSphere Application Server est en cours d'exécution. Par exemple :

```
idp.exemple.com
```

Port de connexion SOAP
Le port SOAP (autonome) par défaut WebSphere Application Server est 8880. Lors de la création d'un domaine pour un serveur WebSphere Application Server qui fait partie d'un cluster WebSphere, le numéro de port SOAP peut s'avérer différent. Par exemple, 8879. Si vous avez un doute sur le numéro de port SOAP correct, utilisez la console d'administration de WebSphere Application Server pour déterminer le port.

Propriétés de sécurité globale WebSphere

WebSphere Application Server est doté d'une option d'activation de sécurité globale. Lorsque la sécurité globale est activée, les propriétés de sécurité doivent être configurées pour le service de gestion Tivoli Federated Identity Manager. La sécurité globale est activée dans la plupart des déploiements.

Nom de l'utilisateur d'administration
Nom de l'administrateur WebSphere Application Server. Par exemple : `wsadmin`

Mot de passe d'administration
Mot de passe de l'administrateur WebSphere Application Server tel qu'il a été spécifié lors de l'installation de WebSphere.

Fichier de clés certifiées SSL
Fichier de clés utilisé par WebSphere Application Server.

Si vous avez installé Tivoli Federated Identity Manager sur un ordinateur hébergeant une installation WebSphere existante, le chemin d'accès par défaut sous Linux ou UNIX est le suivant :

```
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/etc/trust.p12
```

Sous Windows :

```
C:\Program Files\IBM\WebSphere\AppServer\
  profiles\AppSrv01\etc\trust.p12
```

Si vous avez installé l'instance WebSphere intégrée dans le cadre de l'installation de Tivoli Federated Identity Manager, le chemin d'accès par défaut sous Linux ou UNIX est le suivant :

```
/opt/IBM/FIM/ewas/profiles/
 itfimProfile/etc/trust.p12
```

Sous Windows :

```
C:\Program Files\IBM\FIM\ewas\
   profiles\AppSrv01\etc\trust.p12
```

Mot de passe du fichier de clés certifiées SSL
Mot de passe requis pour l'accès au fichier de clés certifiées SSL.

Le mot de passe par défaut pour la clé WebSphere est le suivant :

```
WebAS
```

Fichier de clés client SSL
Fichier de clés utilisé par WebSphere Application Server.

Ce fichier de clés est un élément de configuration optionnel. Certains déploiements WebSphere ne nécessitent aucun fichier de clés client SSL.

Mot de passe du fichier de clés client SSL
Mot de passe requis pour l'accès au fichier de clés client SSL. Cette zone doit être renseignée lorsque vous avez entré un fichier de clés client SSL.

Nom de serveur ou cluster WebSphere

Lors de la création d'un domaine, l'assistant de domaine vous demande le nom de serveur ou de cluster WebSphere.

Nom du serveur
Nom du serveur WebSphere Application Server sur lequel le service de gestion Tivoli Federated Identity Manager est configuré.

Le serveur correspond à un serveur unique et ne fait pas partie d'un cluster.

Le nom par défaut est créé automatiquement par l'assistant. A titre d'exemple, pour un hôte appelé `host1` :

```
WebSphere:cell=host1Node01Cell,node=host1Node01,server=server1
```

Nom de cluster
Nom du cluster WebSphere Application Server dans lequel le service de gestion Tivoli Federated Identity Manager est configuré.

Propriétés d'environnement Tivoli Access Manager

L'assistant vous invite à préciser si vous devez ou non effectuer la configuration dans un environnement Tivoli Access Manager. N'effectuez *pas* la configuration dans un environnement Tivoli Access Manager si vous utilisez un serveur point de contact autre que WebSEAL. A titre d'exemple, n'effectuez *pas* la configuration dans un environnement Tivoli Access Manager si vous utilisez WebSphere en tant que serveur point de contact.

L'invite affichée par l'assistant est la suivante :

Cet environnement utilise Tivoli Access Manager
Si vous désélectionnez cette case, vous n'aurez besoin de configurer aucune propriété pour Tivoli Access Manager.

Si vous cochez cette case, spécifiez les propriétés indiquées dans le tableau suivant.

Nom d'utilisateur de l'administrateur
Administrateur Tivoli Access Manager. L'ID par défaut est sec_master. Si vous avez choisi un autre ID administrateur lorsque vous avez installé Tivoli Access Manager, entrez l'ID administrateur dans la zone **Administrator Username**.

Mot de passe de l'administrateur
Mot de passe de l'administrateur Tivoli Access Manager.

Nom d'hôte du serveur de règles
Nom d'hôte complet de l'ordinateur qui exécute le serveur de règles de Tivoli Access Manager. Par exemple :

```
idp.exemple.com
```

Port Numéro de port permettant de communiquer avec le serveur de règles.

Ce nombre correspond au numéro de port que vous avez spécifié lorsque vous avez configuré Tivoli Access Manager. La valeur par défaut est `7135`.

Nom d'hôte du serveur d'autorisations
Nom d'hôte complet de l'ordinateur qui exécute le serveur d'autorisations de Tivoli Access Manager. Par exemple :

```
idp.exemple.com
```

Port Numéro de port permettant de communiquer avec le serveur d'autorisations.

Ce nombre correspond au numéro de port que vous avez spécifié lorsque vous avez configuré Tivoli Access Manager. La valeur par défaut est `7136`.

Domaine Tivoli Access Manager
Nom du domaine d'administration Tivoli Access Manager que vous avez indiqué lors de la configuration de ce dernier. La valeur par défaut est `Default`.

Formulaire de configuration de domaine

Complétez ce formulaire avant d'exécuter l'assistant pour créer et déployer le domaine et l'environnement d'exécution.

Les propriétés figurant dans ce formulaire sont décrites dans Chapitre 3, «Configuration de domaine», à la page 25.

Tableau 1. Propriétés de configuration du domaine

Propriété	Votre valeur
Hôte	
Port de connexion SOAP	
Nom de l'utilisateur d'administration	
Mot de passe d'administration	
Fichier de clés certifiées SSL	
Mot de passe du fichier de clés certifiées SSL	
Fichier de clés client SSL	
Mot de passe du fichier de clés client SSL	
Nom du cluster WebSphere ou Nom du serveur WebSphere	
Cet environnement utilise Tivoli Access Manager	Sélectionner ou Désélectionner

Lorsque votre environnement inclut Tivoli Access Manager (par exemple, lors de l'utilisation de WebSEAL en tant que serveur point de contact), vous devez également fournir un certain nombre de propriétés de configuration Tivoli Access Manager.

Tableau 2. Propriétés d'environnement Tivoli Access Manager

Propriété	Description
Nom d'utilisateur de l'administrateur	
Mot de passe de l'administrateur	
Nom d'hôte du serveur de règles	
Port	
Nom d'hôte du serveur d'autorisations	
Port	
Domaine Tivoli Access Manager	La valeur par défaut est `Default`.

Création et déploiement d'un nouveau domaine

Vous devez créer un domaine et déployer un module d'exécution pour chaque instance de Tivoli Federated Identity Manager.

Avant de commencer

Remarque : IBM a déprécié le client Tivoli Federated Identity Manager Security Token Service (STS) dans cette version.

Si vous utilisez WebSphere 6.X, vous pouvez continuer de vous servir du client Tivoli Federated Identity Manager Security STS tant que Tivoli Federated Identity Manager prend en charge WebSphere 6.X. Lorsque Tivoli Federated Identity Manager arrêtera son support pour WebSphere 6.X, vous devrez utiliser WebSphere Application Server version 7 Update 11 et version ultérieure. Voir API client WS-Trust et WS-Trust Clients pour plus d'informations.

Un assistant vous invite à fournir les propriétés de configuration nécessaires. Vous pouvez utiliser les propriétés indiquées sur le formulaire que vous avez préparé. Pour plus d'informations sur le formulaire, voir Chapitre 3, «Configuration de domaine», à la page 25

Pourquoi et quand exécuter cette tâche

Cette tâche est une condition prérequise pour la configuration de fonctionnalités complémentaires de Tivoli Federated Identity Manager telles que la gestion de la connexion unique fédérée ou de la sécurité des services Web. Elle constitue également une condition prérequise pour les déploiements qui utilisent l'échange de jetons via le service STS de Tivoli Federated Identity Manager.

Un exemple de scénario d'échange de jeton est celui du déploiement d'une délégation contrainte Kerberos Tivoli Federated Identity Manager avec des jonctions WebSEAL.

Procédure

1. Vérifiez que l'application WebSphere Application Server est en cours d'exécution.
2. Copiez tous les fichiers de clés de WebSphere du gestionnaire de déploiement dans tous les noeuds du cluster dans les cas suivants :
   - lorsque vous déployez un domaine dans un cluster WebSphere Application Server ;
   - lorsque la sécurité globale de WebSphere est activée.

   Sur chaque noeud, placez les clés dans un répertoire identique à celui du gestionnaire de déploiement. WebSphere 6.1 effectue automatiquement cette procédure. Néanmoins, assurez-vous que lorsque la console d'administration est utilisée à distance du service de gestion DMGR, le certificat serveur présenté par DMgr est accrédité par la console. Pour cela, il vous suffit de copier le fichier de clés sécurisées de DMgr dans le profil de la console.
3. Connectez-vous à la console WebSphere.
4. Cliquez sur **Tivoli Federated Identity Manager → Mise en route**.

   Le portlet Mise en route s'ouvre.

5. Cliquez sur **Gestion des domaines**. Le portlet Domaines s'ouvre.
6. Cliquez sur **Créer**. Le panneau de bienvenue de l'assistant Domaine s'affiche.
7. Cliquez sur **Suivant**. Le panneau Noeud final du service de gestion s'ouvre.
8. Entrez des valeurs pour les propriétés indiquées.
9. Cliquez sur **Suivant**. Le panneau Sécurité de WebSphere s'affiche.
10. Indiquez si la sécurité globale WebSphere est activée.
    - Si c'est le cas, entrez des valeurs pour les propriétés indiquées, puis cliquez sur **Suivant**.
    - Si ce n'est pas le cas, ne renseignez pas les zones des propriétés restantes. Cliquez sur **Suivant**.
11. Cliquez sur **Test de la connexion**. Lorsqu'elle s'établit, le message d'information suivant s'affiche :

    ```
    FBTCON317I Tivoli Federated Identity Manager s'est connecté correctement.
    ```
12. Cliquez sur **Suivant**. Le panneau Mappage de cible WebSphere s'affiche.
13. Sélectionnez ou entrez le nom de votre serveur ou cluster.
14. Une fois que vous avez terminé, cliquez sur **Suivant**.
    - Lorsque l'environnement WebSphere comprend un serveur unique, l'écran affiche un menu Nom du serveur comportant un nom par défaut.
    - Lorsque l'environnement WebSphere est composé d'un cluster, le panneau affiche le menu Nom de cluster. Ce menu répertorie les noms des clusters définis dans la cellule. Sélectionnez le nom du cluster à utiliser.

    Le panneau Sélection du domaine s'affiche. Un nom par défaut est fourni.
15. Acceptez-le ou entrez le nom du nouveau domaine. Le panneau Paramètres d'environnement Tivoli Access Manager s'ouvre.
16. Sélectionnez ou désélectionnez l'option **Cet environnement utilise Tivoli Access Manager**, selon les cas.
17. Cliquez sur **Suivant**. Lorsque vous sélectionnez cette option, indiquez des valeurs pour le reste des propriétés. Le panneau Récapitulatif s'affiche.
18. Vérifiez que les informations de domaine sont correctes.
19. Cliquez sur **Terminer**.

    La création du domaine est terminée et l'assistant de domaine se ferme. Le panneau Création du domaine terminée s'affiche.
20. Cochez les deux cases qu'il contient.
21. Cliquez sur **OK**.

    La création et le déploiement initiaux du service de gestion et de l'environnement d'exécution Tivoli Federated Identity Manager requièrent l'exécution des deux tâches suivantes :
    - **Faire de ce domaine le domaine de gestion actif**
    - **Ouvrir la fonction Gestion des noeuds d'exécution à la fin de l'opération**
22. Si vous déployez Tivoli Federated Identity Manager dans un cluster WebSphere, assurez-vous que l'agent de noeud WebSphere est en cours d'exécution sur tous les noeuds du cluster.

    Utilisez la console d'administration WebSphere pour vérifier le statut des agents de noeud.

    Les portlets Domaine en cours et Gestion des noeuds d'exécution s'affichent.
23. Dans le portlet Gestion des noeuds d'exécution, cliquez sur **Déployer l'environnement d'exécution**. Le message suivant s'affiche :

```
FBTCON355I - Une requête de déploiement de l'environnement d'exécution
Tivoli Federated Identity Manager est en cours.
```

Le lien suivant s'affiche :

```
Cliquez pour régénérer l'état de déploiement et vérifier l'exécution de l'opération.
```

L'opération de déploiement peut prendre plusieurs minutes. Durant cette période, vous pouvez cliquez sur le lien pour vérifier l'avancement de l'opération. Une fois le déploiement terminé, cliquez sur le lien pour revenir au message :

```
FBTCON132I Le module d'exécution a été déployé dans le domaine.
```

Le panneau Gestion des noeuds d'exécution s'affiche de nouveau. Une entrée correspondant au composant d'exécution est ajoutée dans le tableau **Noeuds d'exécution** pour chaque noeud du domaine. Le bouton **Configurer** est également activé.

24. Dans le tableau Noeuds d'exécution, cochez la case correspondant à votre noeud.
25. Cliquez sur **Configurer**.

    L'application d'exécution est configurée dans l'environnement.
26. Dans un environnement WebSphere en cluster, configurez chaque noeud du cluster en répétant l'étape précédente.
27. Une fois que tous les noeuds sont configurés, cliquez sur le bouton **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Runtime**.

    Le bouton se trouve dans le portlet Domaine en cours.
28. Poursuivez avec les instructions appropriées suivant votre déploiement :
    - Dans un environnement WebSphere en *clusters*, poursuivez avec la rubrique «Mappage du composant d'exécution vers un serveur Web».
    - Dans un environnement WebSphere *non configuré en clusters* (serveur autonome), la création et le déploiement du domaine s'arrêtent là. Poursuivez avec les instructions appropriées suivant votre scénario.

Que faire ensuite

Redémarrez WebSphere Application Server dans les cas suivants :

- S'il manque des informations dans le panneau Sécurité de WebSphere de l'assistant Domaine.
- Lors de la création d'un domaine Tivoli Federated Identity Manager ou d'une connexion à un domaine.

Si vous avez tenté de corriger les informations et que vous ne parvenez toujours pas à vous connecter à la console Tivoli Federated Identity Manager, redémarrez WebSphere Application Server.

Utilisez **Test de la connexion** dans le panneau pour vérifier la connexion entre la console Tivoli Federated Identity Manager et le service de gestion.

Mappage du composant d'exécution vers un serveur Web

Apprenez à mapper le composant d'exécution Tivoli Federated Identity Manager vers un serveur Web IBM HTTP sur les environnements cluster.

Pourquoi et quand exécuter cette tâche

Lors du déploiement du composant d'exécution Tivoli Federated Identity Manager, celui-ci est automatiquement mappé vers le serveur WebSphere Application Server par défaut. Dans les environnements groupés WebSphere, WebSphere Application Server est déployé dans une configuration dotée d'un serveur web, tel qu'IBM HTTP Web Server. Dans ce cas, un plug-in WebSphere a été installé et configuré pour IBM HTTP Web Server.

IBM HTTP Web Server exécute l'équilibrage de charge de travail entre les membres du cluster. Cela signifie que le composant d'exécution Tivoli Federated Identity Manager doit être mappé vers le serveur Web.

Procédure

1. Connectez-vous à la console d'administration WebSphere :
   `http://votre_nom_hôte:9060/admin`
2. Revenez à la page **Applications d'entreprise -> Environnement d'exécution ITFIM**. L'onglet Configuration s'affiche.
3. Dans la section Propriétés du module Web, sélectionnez le lien **Hôtes virtuels**. Une section intitulée Appliquer plusieurs mappages affiche un tableau contenant une ligne pour chaque module Web.
4. Cochez la case de chaque module Web. Assurez-vous que toutes les cases sont cochées.
5. Acceptez l'entrée par défaut de `default_host` dans la zone Hôte virtuel de chaque module Web. Une boîte de message vous invite à sauvegarder vos modifications.
6. Cliquez sur le lien **Sauvegarder**. Le panneau Sauvegarde s'affiche.
7. Cliquez sur **Sauvegarder**.
8. Revenez à la page **Applications d'entreprise -> ITFIM Runtime**. L'onglet Configuration s'affiche.
9. Dans la section Modules, sélectionnez le lien **Gérer les modules**. La page **Applications d'entreprise -> Environnement d'exécution ITFIM -> Gérer les modules** s'affiche. Dans la partie supérieure, le titre Gestion des modules apparaît.
10. Cochez la case correspondant à *chacun* des modules web. Pour Tivoli Federated Identity Manager, la liste des modules peut inclure les éléments suivants, de façon non limitative :
    - ITFIM-Runtime
    - ITFIM Security Token Service
    - ITFIM Information Service
    - TokenService
    - TrustServerWST13

    Remarque : IBM a déprécié le client Tivoli Federated Identity Manager Security Token Service (STS) dans cette version.

    Si vous utilisez WebSphere 6.X, vous pouvez continuer de vous servir du client Tivoli Federated Identity Manager Security STS tant que Tivoli Federated Identity Manager prend en charge WebSphere 6.X. Lorsque Tivoli Federated Identity Manager arrêtera son support pour WebSphere 6.X, vous

devrez utiliser WebSphere Application Server version 7 Update 11 et version ultérieure. Voir API client WS-Trust et WS-Trust Clients pour plus d'informations.

11. Une fenêtre déroulante affiche Clusters et serveurs. Sélectionnez les deux entrées suivantes :
    - L'entrée correspondant à votre cluster. Par exemple, `cluster=fimCluster`.
    - L'entrée correspondant à votre serveur Web. Par exemple, `server=webserver1`
12. Lorsque les deux éléments sont mis en évidence, cliquez sur **Valider**. Dans le tableau des modules, la définition de chaque serveur et cluster s'ajoute à l'entrée, dans la colonne Serveur, pour chacun des modules Web que vous avez sélectionnés.
13. Cliquez sur **OK** au bas de la page. Un message vous invite à sauvegarder vos modifications.
14. Cliquez sur le lien **Sauvegarder**. Le panneau **Applications d'entreprise → Sauvegarder** s'ouvre.
15. Cliquez sur **Sauvegarder**.
16. Pour terminer la configuration du composant d'exécution de Tivoli Federated Identity Manager dans un cluster WebSphere, passez à la section «Activation de la réplication dans un cluster WebSphere».

Activation de la réplication dans un cluster WebSphere

Apprenez à activer la réplication de cache dans un environnement cluster afin d'améliorer le module d'exécution Tivoli Federated Identity Manager.

Pourquoi et quand exécuter cette tâche

Remarque : Cette tâche de configuration s'applique aux environnements groupés WebSphere. Une fois le composant d'exécution Tivoli Federated Identity Manager dans un environnement WebSphere composé d'un seul serveur, ignorez cette procédure.

WebSphere prend en charge l'utilisation d'un *service de mémoire cache dynamique* pour le stockage des données d'application. Les objets données gérés par ce service peuvent être répartis dans des *instances de cache* pouvant être configurées individuellement. L'administrateur WebSphere peut configurer des paramètres, tels que la taille du cache, la persistance sur disque, etc. Chaque instance de cache peut faire partie d'un *domaine de réplication*. Dans un *domaine de réplication*, les données de la mémoire cache sont répliquées et accessibles à tous les serveurs participant au domaine de réplication.

Lorsque le composant d'exécution Tivoli Federated Identity Manager est déployé, certaines étapes de configuration WebSphere sont automatiquement exécutées :

- Un domaine de réplication est créé. Le nom du domaine de réplication est `FIM-nom_votre_cluster` ou `FIM-nom_votre_serveur`.
- Plusieurs instances de cache utilisant le domaine de réplication sont créées.

Une configuration supplémentaire est requise.

Le service de mémoire cache dynamique des serveurs d'applications du cluster doit désormais être configuré sous la forme d'un *client* du domaine de réplication.

Remarque : Les étapes de cette procédure doivent être exécutées pour chaque serveur du cluster.

Exécutez les étapes suivantes pour *chaque* serveur d'applications qui fait partie du cluster :

Procédure

1. Sur la console d'administration de WebSphere, sélectionnez **Serveurs -> Serveurs d'application ->** ***nom_de_votre_serveur***. Les propriétés du serveur sélectionné s'affichent.
2. Dans la section Paramètres du conteneur, développez **Services du conteneur**. Cliquez sur **Service de mémoire cache dynamique**.
3. Dans la section Propriétés générales de l'écran, accédez à la section des paramètres de cohérence. Sélectionnez **Activer la réplication de cache**. Vérifiez que la zone des paramètres de cohérence comporte les valeurs suivantes :
   - Domaine de réplication de groupe complet

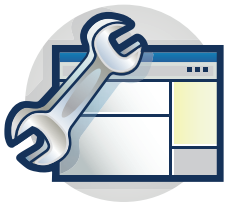
     Sélectionnez le nom du cluster où vous avez déployé le module d'exécution.
   - Type de réplication : **Insertion et extraction**
   - Fréquence d'insertion : **0**
4. Cliquez sur **OK**. Lorsque vous êtes invité à sauvegarder vos modifications, cliquez sur le lien **Sauvegarder**. Lorsque la page suivante s'affiche, cliquez sur **Sauvegarder**.
5. Sur la console d'administration de WebSphere, sélectionnez **Serveurs -> Serveurs d'application ->** ***nom_de_votre_serveur***.

   Remarque : Il se peut que les propriétés indiquées dans la présente section soient déjà définies.
6. Dans la section **Paramètres du conteneur** section, sélectionnez **Gestion de session**. L'onglet Configuration s'affiche.
7. Dans la section Propriétés supplémentaires, sélectionnez **Paramètres de l'environnement distribué**. L'écran Propriétés générales est régénéré.
8. Consultez la section **Paramètres de l'environnement distribué**.
   a. Sélectionnez **Réplication mémoire à mémoire**.
   b. Cliquez sur le lien hypertexte **Réplication mémoire à mémoire**.

   Le panneau Propriétés générales s'affiche.
9. Indiquez dans ce panneau les paramètres de réplication :
   a. Pour le domaine de réplication, sélectionnez le nom du cluster où vous avez déployé le module d'exécution Tivoli Federated Identity Manager.
   b. Paramétrez le mode de réplication sur **Client et serveur**.
10. Lorsque vous êtes invité à sauvegarder vos modifications, cliquez sur le lien **Sauvegarder**. Lorsque la page suivante s'affiche, cliquez sur **Sauvegarder**.
11. Dans le panneau Cluster de serveurs, cochez la case correspondant à votre cluster et cliquez sur **Démarrage en cascade**.

    Vous devez redémarrer le cluster pour que les modifications effectuées soient prises en compte.

Partie 3. Configuration d'une fédération de connexion unique



Les rubriques de la section Configuration vous guident pas à pas lors de la configuration d'une fédération à connexion unique. La console de gestion est dotée d'assistants qui vous guident dans de nombreuses tâches de configuration.

De nombreuses tâches de configuration sont communes à tous les types de fédération. Certaines tâches de configuration concernent uniquement des types de fédération spécifiques.

Complétez les tâches de configuration dans l'ordre suivant :

1. Passez en revue les tâches de configuration communes à tous les types de fédérations. Effectuez les tâches de configuration applicables à votre déploiement.

   Remarque : La plupart des types de fédération prennent en charge une grande variété de scénarios de déploiement. Les étapes effectives requises pour chaque tâche de configuration varient selon le scénario.

   a. Chapitre 5, «Rôles du fournisseur d'identité et du fournisseur de services», à la page 39
   b. Chapitre 6, «Utilisation des clés et certificats pour sécuriser les communications», à la page 41
   c. Chapitre 7, «Configuration de LTPA et de ses clés», à la page 49
   d. Chapitre 8, «Configuration de la sécurité des messages», à la page 51
   e. Chapitre 9, «Configuration de la sécurité du transport», à la page 73
   f. Chapitre 10, «Sélection d'un serveur point de contact», à la page 87
   g. Chapitre 11, «Configuration de WebSphere en tant que serveur point de contact», à la page 93
   h. Chapitre 12, «Configuration d'un plug-in de serveur Web», à la page 127
   i. Chapitre 13, «Configuration de la base de données de service d'alias», à la page 141
   j. Chapitre 14, «Planification du mappage des identités d'utilisateur», à la page 155
2. Suivez les instructions relatives à votre type de fédération :
   - Chapitre 15, «Fédérations SAML : présentation», à la page 179
   - Chapitre 21, «Planification d'une fédération Information Card», à la page 301
   - Chapitre 24, «Présentation de la planification sous OpenID», à la page 347
   - Chapitre 30, «Planification d'une fédération Liberty», à la page 495

- Chapitre 32, «Configuration d'une fédération de connexion unique WS-Federation», à la page 529

Chapitre 4. Présentation des tâches de configuration pour la connexion unique fédérée

Utilisez Tivoli Federated Identity Manager pour établir une fédération de connexions uniques à laquelle les utilisateurs peuvent se connecter une seule fois pour accéder à plusieurs applications Web via différents fournisseurs.

Une fédération est un groupe composé d'au moins deux partenaires commerciaux IBM qui souhaitent initier ou recevoir le transfert d'identités d'utilisateurs au sein de la fédération. L'intégrité de l'identité repose sur des relations d'accréditation entre les membres de la fédération, souvent codifiées par accord légal. Un utilisateur d'une entreprise qui a participé à une fédération par le biais d'une connexion unique fédérée peut en toute sécurité accéder aux ressources de leur partenaire commercial IBM fédéré. L'accès aux ressources est généralement effectuée à l'aide d'un navigateur Web.

Lorsque vous établissez la fédération au moyen de Tivoli Federated Identity Manager, vous bénéficiez des caractéristiques suivantes du produit :

- Normes ouvertes pour la connexion unique
- Intégration aux capacités de connexion unique d'IBM WebSphere Application Server 6.1, éliminant ainsi la nécessité de recourir à une authentification via des applications individuelles
- Prise en charge d'un nombre illimité de fédérations et possibilité de définir des configurations personnalisées unique pour chaque fédération.

  Vous pouvez par exemple assumer le rôle de fournisseur d'identité ou de fournisseur de services dans toutes les fédérations, avec une seule installation de Tivoli Federated Identity Manager.
- Support d'intégration pour les applications Web exécutées sur l'un des types de serveurs suivants :
  - WebSphere Application Server version 6.1 et ultérieure
  - Microsoft Internet Information Server (IIS)
  - IBM HTTP Server (IHS)
  - Serveur HTTP Apache version 2.0 ou 2.2
- Administration Web simplifiée

Le déploiement d'une fédération à connexion unique nécessite l'accomplissement d'une série de tâches. Certaines de ces tâches sont communes à tous les types de fédérations. D'autres tâches concernent plus particulièrement le protocole standard de la fédération (par exemple SAML 2.0).

Pour déployer une fédération à connexion unique, vous pouvez d'abord passer en revue les tâches communes, puis procéder à la configuration spécifique du protocole standard.

Remarque : Vous devez créer un domaine avant de déployer une fédération de connexion unique. Si vous n'avez pas encore déployé un domaine, suivez les instructions au Chapitre 3, «Configuration de domaine», à la page 25.

Les tâches décrites dans les rubriques suivantes sont communes à tous les types de fédérations. Parcourez chaque rubrique dans l'ordre suivant avant de configurer une fédération pour le protocole sélectionné.

1. Chapitre 5, «Rôles du fournisseur d'identité et du fournisseur de services», à la page 39
2. Chapitre 6, «Utilisation des clés et certificats pour sécuriser les communications», à la page 41
3. Chapitre 7, «Configuration de LTPA et de ses clés», à la page 49
4. Chapitre 8, «Configuration de la sécurité des messages», à la page 51
5. Chapitre 9, «Configuration de la sécurité du transport», à la page 73
6. Chapitre 10, «Sélection d'un serveur point de contact», à la page 87
7. Chapitre 11, «Configuration de WebSphere en tant que serveur point de contact», à la page 93
8. Chapitre 12, «Configuration d'un plug-in de serveur Web», à la page 127
9. Chapitre 13, «Configuration de la base de données de service d'alias», à la page 141
10. Chapitre 14, «Planification du mappage des identités d'utilisateur», à la page 155

Pour plus d'informations sur les concepts de fédération et sur l'assistance à la mise en oeuvre d'une solution de gestion d'identité fédérée, reportez-vous à la section Federation concepts du document *Enterprise Security Architecture Using IBM Tivoli Security Solutions* disponible à l'adresse http://www.redbooks.ibm.com/redbooks/pdfs/sg246014.pdf.

Chapitre 5. Rôles du fournisseur d'identité et du fournisseur de services

Au sein d'une fédération, chaque partenaire a un rôle. Il s'agit du rôle **Fournisseur d'identité** ou **Fournisseur de services**. Le fournisseur d'identité est un partenaire de fédération qui garantit l'identité des utilisateurs. Un fournisseur de services est un partenaire de fédération qui fournit des services à l'utilisateur.

- **Fournisseur d'identité**

  Le fournisseur d'identité authentifie un utilisateur et transmet un *jeton d'authentification* (c'est-à-dire les informations permettant de vérifier l'authenticité de l'utilisateur) au fournisseur de services.

  Le fournisseur d'identité effectue les deux types d'authentification suivants :

  – Authentification d'utilisateur directe. Par exemple, la validation d'un nom d'utilisateur et d'un mot de passe.

  – Authentification d'utilisateur indirecte. Par exemple, en validant une assertion concernant l'identité de l'utilisateur, telle que représentée par un autre fournisseur d'identité.

  Le fournisseur d'identité traite la gestion des identités utilisateur afin de dégager le fournisseur de services de cette responsabilité.

- **Fournisseur de services**

  En général, les fournisseurs de services n'authentifient pas les utilisateurs, mais demandent à un fournisseur d'identité de prendre les décisions d'authentification. Les fournisseurs de services comptent sur les fournisseurs d'identité pour vérifier l'identité d'un utilisateur, ainsi que certains attributs relatifs à l'utilisateur qui sont gérés par le fournisseur d'identité.

  Les fournisseurs de services peuvent également gérer un compte local pour l'utilisateur, ainsi que les attributs qui sont propres à leur service.

  Les fournisseurs de services peuvent gérer pour l'utilisateur un compte local qui peut être référencé par un identificateur de l'utilisateur.

  Certains protocoles de fédération utilisent une terminologie distincte pour faire référence au rôle du fournisseur de services :

  – **Partie de confiance**

    La spécification du protocole Information Card emploie le terme de partie de confiance pour désigner le rôle du fournisseur de services. Sélectionnez le rôle Fournisseur de services pour votre parti de confiance lorsque vous configurez la fédération Information Card dans l'assistant Tivoli Federated Identity Manager.

  – **Consommateur**

    La spécification du protocole OpenID emploie le terme de consommateur pour désigner le rôle du fournisseur de services. Sélectionnez le rôle Fournisseur de services pour votre consommateur lorsque vous configurez le protocole OpenID dans l'assistant Tivoli Federated Identity Manager.

Avant d'installer Tivoli Federated Identity Manager, vous devez savoir si vous serez le fournisseur d'identité ou le fournisseur de services dans chacune des fédérations à configurer. Vous devrez également comprendre les options du serveur point de contact correspondant à votre rôle.

Chapitre 6. Utilisation des clés et certificats pour sécuriser les communications

Dans un environnement de production standard, tous les messages et toutes les communications de ces messages entre les partenaires et les utilisateurs membres de la fédération sont sécurisés. En outre, vous devez sécuriser les communications entre les serveurs de votre environnement, par exemple, les communications entre votre serveur et votre registre d'utilisateurs.

Par exemple, les normes SAML stipulent que les partenaires doivent utiliser une infrastructure PKI (Public Key Infrastructure) et mettre en oeuvre le protocole SSL (Secure Sockets Layer) sur HTTP ou HTTPS pour établir une relation de confiance. Ainsi, l'intégrité et la confidentialité des messages pendant le transport sont garanties.

La mise en oeuvre de la sécurité est un sujet complexe et est fonction de la configuration de votre environnement et des règles de sécurité de votre organisation. Cette présentation explique les concepts généraux de la sécurisation des éléments dans un environnement Tivoli Federated Identity Manager. Si vous avez besoin d'une assistance concernant cette rubrique, consultez les exigences de sécurité contenues dans le document relatif aux spécifications du protocole, ou contactez un conseiller en sécurité informatique.

Sécurité de niveau message

Pour sécuriser le contenu des messages et des assertions, les normes SAML préconisent l'utilisation d'un chiffrement à clé publique. Grâce à cette méthode, les partenaires d'une fédération échangent des paires de clés publiques/privées, et les utilisent pour signer, chiffrer, valider et déchiffrer des messages, ainsi que les assertions dans les messages. Le processus de signature, de chiffrement et de validation des messages est requis dans la norme SAML ou selon les exigences de leur environnement.

Lorsque vous configurez une fédération dans Tivoli Federated Identity Manager, l'assistant de configuration de fédération vous présente, selon les cas, des *exigences* relatives à la signature, à la validation ou au chiffrement, ou des *options* de signature, validation ou chiffrement selon le protocole SAML, le profil ou les liaisons que vous avez sélectionnés.

Si vous avez, par exemple, choisi lors de la configuration de votre fédération d'indiquer qu'une signature était requise, l'assistant vous invite à spécifier une clé de signature. Si vos sélections entraînent la définition visant à signer ou à ne pas signer, l'assistant vous invite à effectuer une sélection.

Avant de pouvoir utiliser l'assistant de configuration de fédération, vous devez avoir créé les clés appropriées. Les informations du Chapitre 8, «Configuration de la sécurité des messages», à la page 51 vous permettent de planifier les clés requises dans votre environnement et contiennent des instructions relatives à leur création et à leur obtention.

Les sections qui suivent fournissent une description générale des clés en usage dans les fédérations SAML.

Signature

La signature des messages XML et des assertions SAML est effectuée par un partenaire unique, afin de protéger l'intégrité du message. La signature permet à la partie récipiendaire de savoir si le message a été modifié durant la transmission.

La signature est réalisée à l'aide d'une clé privée. Le partenaire qui reçoit le message XML ou l'assertion SAML signé(e) a besoin du certificat X.509 (clé publique) qui correspond à la clé privée du signataire du message. Par défaut, le certificat X.509 (clé publique) est inclus dans la signature sous forme de certification X.509 codée en base 64. Toutefois, vous avez la possibilité de spécifier les données de certificat à inclure avec vos signatures.

Validation

Les signatures contenues dans les messages et assertions peuvent être validées par le partenaire récipiendaire. La validation confirme que l'identité du signataire a été garantie. La validation s'effectue au moyen de la clé publique ou du partenaire ayant signé les messages ou les assertions.

Chiffrement et déchiffrement

Dans SAML 2.0, le chiffrement des messages peut avoir lieu en plus de leur signature. L'usage des paires de clés publique/privée lors du chiffrement et du déchiffrement diffère de la procédure appliquée lors de la signature et de la validation. La *clé publique du destinataire prévu* est utilisée pour le chiffrement. Pour qu'un partenaire puisse chiffrer un message, il doit être en possession de la clé publique du partenaire auquel le message est adressé.

Le partenaire récipiendaire du message chiffré doit utiliser sa clé *privée* pour déchiffrer le message. DansTivoli Federated Identity Manager, et lorsque SAML 2.0 est utilisé, les deux partenaires doivent obtenir leurs propres paires de clés publiques/privées afin de les utiliser pour le chiffrement. Ils doivent échanger leurs clés publiques de manière à ce que chaque partenaire puisse chiffrer les messages adressés à l'autre.

Sécurité de niveau transport

La sécurité de niveau message, telle que décrite à la section précédente, ne protège que le contenu du message. Pour permettre la protection du message lors de sa communication entre les partenaires, SAML recommande d'utiliser une connexion SSL (Secure Sockets Layer) avec authentification de serveur et, dans certains cas, une authentification réciproque.

Le protocole SSL permet d'établir l'authenticité, l'intégrité et la confidentialité entre les parties impliquées dans la transmission de données par le biais d'autres protocoles, tels que HTTP, sur un réseau.

Remarque : Le protocole SSL est une notion complexe. Cette présentation est une introduction qui permet de vous familiariser avec les concepts de base et la terminologie employés dans ce manuel.

authentification sur le serveur

Dans un environnement Tivoli Federated Identity Manager, le protocole SSL a pour rôle de protéger les noeuds finals en provenance et en direction desquels les

messages SAML sont envoyés et reçus. Dans le cadre de communications protégées par SSL entre des partenaires de fédération, l'un des partenaires agit en tant que *client* ou la partie qui demande des données. L'autre partenaire agit en tant que *serveur*, ou le récipiendaire de la requête et émetteur de la réponse à cette requête.

Dans une fédération SAML 1.x, une connexion unique est reçue au niveau du partenaire fournisseur d'identité. Lorsqu'une connexion SSL est établie entre les partenaires de la fédération, le partenaire fournisseur d'identité joue le rôle de *serveur*, tandis que le fournisseur de services agit en tant que *client*.

Dans une fédération SAML 2.0, une requête de connexion unique peut être reçue par l'un ou l'autre des partenaires. Chaque partenaire peut donc aussi bien jouer le rôle de serveur que de client.

Le protocole SSL peut être configuré sur le serveur uniquement (*authentification de serveur*, ou bien à la fois sur le serveur et le client (*authentification réciproque*). Les normes SAML nécessitent l'utilisation d'une authentification de serveur entre les partenaires, au minimum. L'ajout d'une authentification réciproque permet de bénéficier d'une sécurité renforcée.

Pour activer l'authentification du serveur, vous devez créer une paire de clés publique/privée et obtenir un certificat. Votre serveur utilise un certificat pour s'authentifier auprès du client. Ce certificat est également appelé *certificat serveur* ou encore, *certificat personnel*.

Bien que vous puissiez créer votre propre certificat serveur avec un logiciel qui prend en charge la création de certificats dans un environnement de production, vous pouvez obtenir un certificat serveur d'une partie tierce. Le certificat serveur est également appelé *autorité de certification* ou *CA* qui émet des certificats.

Avant de tenter d'établir une connexion SSL, le client auquel le serveur présente son certificat doit obtenir le certificat de la part de l'autorité de certification (CA) qui a émis le certificat serveur. Le client assure la gestion d'une liste d'émetteurs dignes de confiance à laquelle il ajoute le certificat de CA.

Le certificat serveur contient les informations suivantes :

- la clé publique du certificat serveur
- le numéro de série du certificat
- la période de validité du certificat
- le nom distinctif du serveur qui inclut le nom d'hôte associé au serveur
- le nom distinctif de l'émetteur
- la signature numérique de l'émetteur

Pour établir la connexion SSL, le serveur présente son certificat, qui doit être vérifié par le client. Par exemple, le client vérifie sa liste d'émetteurs de confiance ou autorités de certification afin de voir si l'émetteur du certificat du serveur est sécurisé. Il compare ensuite la signature numérique de l'émetteur contenue dans le certificat du serveur avec celle qui est contenue dans le certificat de CA.

Le serveur doit exporter son certificat de CA et le fournir à son client partenaire.

En résumé, l'authentification du serveur nécessite les clés et certificats suivants :

Tableau 3. Exigences liées au certificat d'authentification de serveur SSL

Certificat requis	Qui doit obtenir le certificat	Remarque
Certificat serveur et clé privée associée à ce certificat	Partenaire agissant en tant que serveur	Dans une fédération SAML 1.x, le fournisseur d'identité agit toujours en tant que serveur.
Certificat de CA lié à l'émetteur du certificat serveur	Partenaire agissant en tant que client	Dans une fédération SAML 1.x, le fournisseur de services agit toujours en tant que client.

Les instructions d'activation de SSL sont décrites à la rubrique «Activation de SSL sur WebSphere Application Server», à la page 74.

Authentification client

Un serveur peut être configuré en vue d'exiger une authentification de la part de ses clients pour confirmer leurs identités. Tivoli Federated Identity Manager accepte l'une des méthodes d'authentification client suivantes :

Authentification de base (par mot de passe)
Si l'authentification de base est configurée, le serveur demande au client de s'authentifier au moyen d'un nom d'utilisateur et d'un mot de passe. Aucun certificat n'est utilisé avec cette méthode.

Authentification par certificat client
Un *certificat client* est similaire à un certificat serveur. Pour obtenir un certificat client, le client en effectue généralement la demande auprès d'une autorité de certification. Avant d'établir une fédération, les partenaires s'accordent généralement sur le choix d'une autorité de certification pour la signature du certificat client. Le serveur doit s'assurer que cette autorité de certification se trouve dans la liste des émetteurs dignes de confiance.

Lorsque l'authentification par certificat client est configurée, le serveur demande l'authentification auprès du client. Celui-ci répond en envoyant au serveur son certificat client et sa signature numérique via un élément de données à génération aléatoire. Le certificat client inclut généralement les éléments suivants :

- clé publique du client
- numéro de série du certificat
- période de validité du certificat
- nom distinctif du client
- nom distinctif de l'émetteur
- signature numérique de l'émetteur

Le serveur vérifie les informations client. Le client exporte par exemple son certificat et le fournit au serveur partenaire. Ensuite, le serveur utilise la clé publique du client dans le certificat client pour effectuer les tâches suivantes :

- valider la signature numérique du client
- vérifier sa liste d'émetteurs digne de confiance ou d'autorités de certification
- voir si l'émetteur de certificat client est digne de confiance

- comparer la signature numérique de l'émetteur contenue dans le certificat client avec celle contenue dans le certificat de CA

Si l'authentification par certificat client est employée, les certificats suivants sont requis :

Tableau 4. Exigences liées au certificat d'authentification client SSL

Certificat requis	Qui doit obtenir le certificat	Remarque
Certificat client et clé privée associée	Partenaire agissant en tant que client	Dans une fédération SAML 1.x, le fournisseur de services agit toujours en tant que client.
Certificat de CA lié à l'émetteur du certificat client	Partenaire agissant en tant que serveur	Dans une fédération SAML 1.x, le fournisseur d'identité agit toujours en tant que serveur.

Les partenaires agissant en tant que *serveurs* doivent suivre les instructions relatives à la configuration d'une authentification client sur leurs serveurs ; voir «Configuration des exigences relatives à l'authentification client», à la page 78.

Les partenaires agissant en tant que *clients dont les partenaires exigent une authentification par certificat client* doivent suivre les instructions relatives à la configuration de leurs certificats client ; voir «Configuration des certificats client», à la page 83.

Stockage et gestion des clés et certificats

Les clés et certificats sont stockés dans des fichiers de clés et des fichiers de clés certifiées.

Fichiers de clés
Les clés privées et les certificats sont stockés dans des fichiers de clés.

Fichier de clés certifiées
Les clés privées et les certificats CA sont stockés dans des fichiers de clés certifiées. Un fichier de clés certifiées est un fichier de clés qui, par convention, contient uniquement des clés certifiées et des certificats.

Dans votre environnement Tivoli Federated Identity Manager, un certain nombre de clés et de certificats sont stockés dans les fichiers de clés et fichiers de clés certifiées de WebSphere Application Server, tandis que d'autres sont stockés dans ceux de Tivoli Federated Identity Manager. Une Une fonction Tivoli Federated Identity Manager appelée le *service de clés* gère ces fichiers de clés et fichiers de clés certifiées. L'emplacement dépend de l'objectif d'utilisation des clés et des certificats.

Les clés et certificats sont stockés un fichier de clés et un fichier de clés certifiées de WebSphere Application Server :

- Certificats serveur SSL et leurs clés privées (dans le fichier de clés WebSphere du serveur partenaire)
- Certificat de CA pour les clients présentant un certificat client (dans le fichier de clés certifiées WebSphere du serveur partenaire)

Les clés et certificats sont stockés un fichier de clés et un fichier de clés certifiées de Tivoli Federated Identity Manager :

- Certificats client SSL et leurs clés privées

Certificats utilisés pour l'authentification par certificat client. Les clés privées sont celles qui se trouvent dans le fichier de clés client

- Certificats de CA pour les serveurs sur lesquels l'authentification de serveur SSL est configurée

  S'applique aux certificats dans le fichier de clés certifiées du client

- Les clés de signature, de validation et de chiffrement sont également gérées dans les fichiers de clés et fichiers de clés certifiées de Tivoli Federated Identity Manager. Par exemple :

  Clés de signature
  Il s'agit des clés privées stockées dans les fichiers de clés.

  Clés de validation
  Il s'agit des clés publiques qui correspondent aux clés privées utilisées pour la signature. Ces clés sont stockées dans les fichiers de clés certifiées.

  Clés de chiffrement

  – La clé employée pour le chiffrement de données est une clé publique obtenue auprès de votre partenaire. Vous devez la stocker dans votre fichier de clés certifiées.

  – La clé servant à déchiffrer les données est une clé privée. Vous devez la stocker dans votre fichier de clés.

Par défaut, WebSphere Application Server et Tivoli Federated Identity Manager comportent des fichiers de clés, fichiers de clés certifiées, clés et certificats destinés aux environnements de test.

WebSphere Application Server
Durant la création de profils, WebSphere Application Server crée les éléments suivants :

- Fichier de clés key.p12
- Fichier de clés certifiées trust.p12
- Un certificat d'auto-signature par défaut dans le fichier de clés key.p12

Le mot de passe utilisé à la fois pour le fichier de clés et le fichier de clés certifiées est `WebAS`.

Tivoli Federated Identity Manager

Tivoli Federated Identity Manager fournit deux fichiers de clés Java par défaut, un certificat d'auto-signature et quelques certificats de CA :

- DefaultKeyStore.jks pour les clés de signature t de chiffrement (clés privées)
- DefaultTrustedKeyStore.jks pour les certificats de CA
- Un certificat d'auto-signature comportant l'alias `testkey`, qui peut être utilisé en tant que clé de signature en environnement de test, et est contenu dans le fichier de clé s
- Plusieurs certificats de CA contenus dans le fichier de clés certifiées

Le mot de passe par défaut des fichiers de clés certifiées est `testonly`.

Les clés et certificats par défaut sont fournies à des fins de test uniquement. Vous devez créer une nouvelle série de clés et de certificats ainsi que, le cas échéant, de nouveaux fichiers de clés lors de la configuration de Tivoli Federated Identity Manager.

Pour plus d'informations, voir «Création des magasins de clés, clés et certificats».

Création des magasins de clés, clés et certificats

Comme décrit dans les sections précédentes, la configuration de la sécurité au niveau des messages et du transport nécessite l'utilisation de paires de clés publiques et privées, ainsi que de certificats. Pour pouvoir utiliser les clés et certificats appropriés, vous devez appliquer un processus général lors de leur création, ainsi que lors de la création des fichiers de clés dans lesquels vous devez les stocker, si vous choisissez de ne pas utiliser les fichiers de clés par défaut.

La procédure générale de création des clés, certificats et de leurs magasins de clés est la suivante :

1. Créez le magasin de clés (sous forme de fichier de clés normales ou de fichier de clés certifiées), ou reprenez un magasin existant.
2. Créez la demande de certificat, qui permet de générer une paire de clés publique/privée et peut être adresse à une autorité de certification (CA). La demande de certificat contient la clé publique et les données relatives à vous-même (en tant que demandeur) spécifiées dans le certificat.
3. Envoyez la demande de certificat à l'autorité de certification. L'autorité de certification émet le certificat.
4. Recevez le certificat en provenance de l'autorité de certification et importez-le dans les magasins de clés appropriés.
5. Partagez les clés publiques des certificats avec votre partenaire selon les besoins.

En outre, vous devez également importer certaines clés et certains certificats dans vos fichiers de clés en provenance de votre partenaire.

WebSphere Application Server et Tivoli Federated Identity Manager comprennent tous deux des utilitaires permettant de créer des demandes de certificat et de recevoir celles-ci de la part de l'autorité de certification.

Les informations relatives à l'exécution de toutes les tâches de sécurité au niveau des messages et du transport, y compris la création de magasins de clés, clés et certificats à l'aide des utilitaires concernés, sont décrites dans les sections suivantes du présente manuel :

Instructions de sécurité de niveau message :
Chapitre 8, «Configuration de la sécurité des messages», à la page 51.

Instructions de sécurité de niveau transport :
Chapitre 9, «Configuration de la sécurité du transport», à la page 73

Critère de sélection de clé

Configurez l'ordre des certificats ou clés en utilisant les critères de sélection de clé d'exécution.

Par défaut, Tivoli Federated Identity Manager, version 6.1, crée une liste de certificats ou de clés partageant la même valeur SubjectDN et optimisée de la durée de vie la plus longue à la plus courte. Cette fonction de produit, appelée Auto Key Rollover (Substitution de clé automatique), possède les caractéristiques suivantes :

- Lors de la signature de documents, la fonction utilise une clé valide avec la durée de vie restante la plus courte (par exemple, le certificat X.509 le plus ancien ou la clé privée la plus ancienne).
- Lors de la validation, la fonction parcoure la liste des clés pour la valeur SubjectDN donnée jusqu'à réussite de la validation. Une validation échoue signifie que toutes les clés disponibles étaient incorrectes.

Vous pouvez utiliser la propriété d'exécution personnalisée `key.selection.criteria` pour configurer l'ordre des certificats ou des clés. Utilisez ces valeurs pour la propriété personnalisée :

only.alias
: Alias uniquement : la clé sélectionnée uniquement, sans substitution automatique. Si la clé est incorrecte, le logiciel indique un échec. Configurez la propriété pour pouvoir utiliser la valeur.

shortest.lifetime
: Durée de vie la plus courte : pour la signature, une clé valide avec la durée de vie disponible la plus courte. Pour la validation, la disponibilité de la durée de vie de clé fonctionne de la plus courte à la plus longue.

longest.lifetime
: Durée de vie la plus longue : pour la signature, une clé valide avec la durée de vie disponible la plus longue. Pour la validation, la disponibilité de la durée de vie de clé fonctionne de la plus longue à la plus courte.

Chapitre 7. Configuration de LTPA et de ses clés

Vous devez consulter les paramètres LTPA de votre serveur WebSphere Application Server après avoir installé Tivoli Federated Identity Manager. Vous pouvez utiliser la configuration LTPA par défaut ou l'adapter à votre environnement.

Pourquoi et quand exécuter cette tâche

La configuration LTPA par défaut est la suivante :

Groupe de jeux de clés
Les clés LTPA servent à chiffrer et à déchiffrer les données à envoyer d'un serveur à un autre. Elles sont stockées par jeux, qui sont à leur tour stockés par groupes. Le groupe de jeux de clés par défaut est NodeLTPAKeySetGroup.

Jeux de clés
Les jeux de clés par défaut sont NodeLTPAKeyPair et NodeLTPASecret.

Génération de clés
Par défaut, les clés LTPA sont générées automatiquement au premier démarrage du serveur après l'installation. Elles sont régénérées automatiquement toutes les 12 semaines le dimanche à 22 h 00 (horloge au format 24 heures).

Avertissement : Si votre configuration comporte un serveur d'applications cible distinct, les clés LTPA doivent résider sur le serveur point de contact WebSphere Application Server et sur le serveur d'applications cible. Un serveur d'applications cible peut être un WebSphere Application Server distinct, ou un serveur pris en charge par le plug-in du serveur Web Tivoli Federated Identity Manager.

Si vous générez des clés en mode automatique, conservez-les sur le serveur d'applications en les synchronisant avec les clés générées sur votre serveur point de contact WebSphere Application Server.

Pour plus d'informations sur l'exportation de clés à partir de WebSphere Application Server et leur importation sur votre serveur d'applications, consultez les rubriques «Exportation de la clé LTPA à partir du serveur point de contact», à la page 124 et «Importation de la clé LTPA dans WebSphere Application Server», à la page 132, or «Configuration de la clé LTPA sur le serveur Web», à la page 135 .

Délai d'attente du cache d'authentification
Cette valeur indique le délai de validité (en minutes) d'un jeton LTPA. Par défaut, ce délai est de 10 minutes.

Délai de validité du transfert de droits d'accès entre serveurs
Cette valeur indique le délai de validité des droits d'accès émanant d'un autre serveur. La valeur par défaut est 120 minutes.

Pour vérifier ou modifier ces paramètres, procédez comme suit :

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Administration, applications et infrastructure sécurisées**.

   Le panneau Administration, application et infrastructure sécurisées s'affiche.
3. Sur la gauche, cliquez sur **Mécanismes d'authentification et expiration**. L'onglet Configuration s'affiche. Utilisez-le pour vérifier ou modifier le groupe de jeux de clés défini, le délai d'attente du cache d'authentification et le délai de validité du transfert de droits d'accès entre serveurs.
4. Pour modifier les paramètres du groupe de jeux de clés et de génération de clés, cliquez sur **Groupe de jeux de clés**. Modifiez votre environnement, puis cliquez sur **Appliquer**.
5. Revenez à l'onglet Configuration.
6. Dans la section Expiration de l'authentification de l'onglet Configuration, vérifiez ou modifiez les valeurs des zones **Délai d'attente du cache d'authentification** et **Délai de validité du transfert de droits d'accès entre serveurs**.
7. Lorsque vous avez terminé, cliquez sur **Appliquer**.
8. Enregistrez les modifications dans le fichier de configuration principale.

Que faire ensuite

Poursuivez avec la configuration de votre environnement. Par exemple, poursuivre avec le Chapitre 8, «Configuration de la sécurité des messages», à la page 51.

Chapitre 8. Configuration de la sécurité des messages

Tivoli Federated Identity Manager utilise des certificats (paires de clés publiques et privées) pour sécuriser les messages.

Avant l'établissement d'une fédération, vous devez déterminer, en accord avec votre partenaire, les configurations de sécurité à utiliser au sein de votre fédération. Vous devez ensuite créer ou demander des certificats, ou encore les obtenir auprès de votre partenaire, selon les cas, puis les importer dans le service de clés Tivoli Federated Identity Manager.

Remarque : Les instructions relatives à la configuration des certificats compatibles avec SSL, tels que les certificats serveur, certificats client ou exigences d'authentification client, sont décrites au Chapitre 9, «Configuration de la sécurité du transport», à la page 73. Les rubriques de ce chapitre traitent uniquement de la sécurité au niveau des messages, à l'exception des rubriques relatives à la préparation de vos fichiers de clés.

Pour configurer la sécurité des messages dans votre environnement, procédez comme suit :

1. Préparez les fichiers de clés. Voir «Préparation des fichiers de clés».
2. Discutez avec votre partenaire des consignes de sécurité de message, puis établissez une liste des fichiers de clés et des certificats dont chacun de vous a besoin. Evaluez la possibilité d'utiliser les listes de contrôle mentionnées à la rubrique «Planification de la sécurité au niveau message», à la page 54.
3. Obtenez les certificats requis pour votre environnement. Voir «Obtention de vos clés et certificats», à la page 57.
4. Ajoutez les certificats aux fichiers de clés. Voir «Ajout de vos certificats à votre fichier de clés», à la page 60.
5. Demandez à votre partenaire les certificats dont vous avez besoin. Voir «Obtention d'un certificat de votre partenaire», à la page 63.
6. Fournissez à votre partenaire les certificats dont celui-ci est susceptible d'avoir besoin. Voir «Transmission de certificats au partenaire», à la page 66.
7. Si un des certificats utilisés correspond à un fichier PKCS#12, vous devez mettre à jour votre stratégie de cryptographie Java. Voir «Mise à jour de la règle de cryptographie», à la page 68.
8. Si vous configurez un environnement de production et que vous n'utilisez pas les fichiers de clés et les certificats par défaut, supprimez-les de sorte qu'ils ne soient pas utilisés par inadvertance. Voir «Suppression de fichiers de clés par défaut», à la page 69.

Préparation des fichiers de clés

Préparez les fichiers de clés dans le service de clés Tivoli Federated Identity Manager, quel que soit votre rôle au sein d'une fédération ou la norme SAML que vous utilisez. Les fichiers de clés ont pour fonction de conserver les clés et certificats utilisés pour sécuriser le contenu et le transport des messages.

Pourquoi et quand exécuter cette tâche

Le service de clés contient au minimum les deux fichiers de clés suivants :

Fichier des clés de signature et de chiffrement

Fichier de clés où sont enregistrées vos clés privées Les clés privées sont celles que vous utilisez pour la signature et le chiffrement. Elles sont également employées pour les certificats client si vous tenez le rôle de client dans une connexion SSL et que votre partenaire exige votre authentification par certificat.

Fichier de clés des certificats de CA (appelé magasin de clés, ou fichier de clés certifiées)

Ce fichier de clés est celui dans lequel vous stockez les clés publiques de votre partenaire, ainsi que les certificats de CA liés aux autorités de certification que vous accréditez. Les clés publiques permettent de valider les signatures ou de chiffrer les données envoyées à votre partenaire.

Pour préparer le fichier de clés et le fichier de clés certifiées pour votre environnement, plusieurs méthodes sont possibles :

- Utilisez le fichier de clés et le fichier de clés certifiées par défaut, et modifiez leurs mots de passe, afin que leurs mots de passe par défaut ne soient plus en usage. Pour ce faire, consultez la rubrique «Modification du mot de passe d'un fichier de clés».
- Créez un fichier de clés et un fichier de clés certifiées, comme décrit à la rubrique «Création d'un fichier de clés», à la page 53, puis importez-les dans le service de clés.

Vous pouvez créer autant de fichiers de clés et de fichiers de clés certifiées que nécessaire, afin de simplifier la répartition des clés en plusieurs catégories uniques pour vos fédérations.

Modification du mot de passe d'un fichier de clés

Vous pouvez modifier le mot de passe d'un fichier de clés ou d'un fichier de clés certifiées à l'aide de la console.

Pourquoi et quand exécuter cette tâche

Vous pouvez être amené à modifier le mot de passe d'un fichier de clés dans les cas suivants :

- Vous voulez utiliser le fichier de clés ou le fichier de clés certifiées par défaut dans un environnement de production.
- La sécurité du mot de passe du fichier de clés est compromise.
- Votre stratégie de sécurité vous impose de modifier périodiquement les mots de passe de fichier de clés.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.

   Le panneau Fichiers de clés s'ouvre.
3. Sélectionnez un fichier de clés dans le tableau Fichier de clés. Le bouton **Modifier le mot de passe** s'active.
4. Cliquez sur le bouton **Modifier le mot de passe**. Le panneau Modification du mot de passe du fichier de clés s'ouvre.
5. Entrez l'ancien mot de passe et le nouveau. L'ancien mot de passe par défaut du fichier de clés et du fichier de clés certifiées est `testonly`.
6. Cliquez sur **OK**. Le mot de passe est modifié.

7. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Que faire ensuite

Répétez ce processus pour chaque fichier de clés dont le mot de passe doit être modifié. Poursuivez ensuite par la création de nouveaux fichiers de clés ou la planification de la sécurité au niveau message.

Création d'un fichier de clés

Vous devez créer un fichier de clés si vous avez besoin de fichiers de clés supplémentaires ou que vous ne souhaitez pas faire usage des fichiers par défaut. Le service de clés de Tivoli Federated Identity Manager ne prend en charge que les fichiers de clés Java (.jks).

Pourquoi et quand exécuter cette tâche

Tivoli Federated Identity Manager ne fournit pas d'utilitaire de création de fichiers de clés. Toutefois, vous pouvez créer un fichier de clés à l'aide de n'importe quel utilitaire de génération de clés. Par exemple, faites appel à l'utilitaire **keytool** qui est inclus dans WebSphere Application Server pour créer un fichier de clés comme suit :

```
keytool -import -noprompt -trustcacerts -alias myca
-file myca.pem -keystore mykeys.jks -storepass passw0rd
```

Pour plus de détails sur l'utilitaire keytool, consultez le centre de documentation WebSphere Application Server 8.0 à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp.

Que faire ensuite

Vous devez importer le fichier de clés dans le service de clés Tivoli Federated Identity Manager. Pour plus détails, voir «Importation d'un fichier de clés».

Importation d'un fichier de clés

Si vous avez créé un fichier de clés, vous devez l'importer dans le service de clés de Tivoli Federated Identity Manager avant de pouvoir l'utiliser.

Pourquoi et quand exécuter cette tâche

Procédure

1. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
   Le panneau Fichiers de clés s'ouvre.
2. Cliquez sur **Importer**. L'assistant d'importation démarre et affiche le panneau Importer le fichier de clés.
3. Entrez un chemin d'accès complet dans la zone **Emplacement du fichier de clés**. Par exemple :
   ```
   /tmp/mykeys.jks
   ```
   Vous pouvez également cliquer sur **Parcourir** pour localiser le fichier de clés sur le système de fichiers.

   Remarque : Le fichier de clés à importer doit se trouver sur la même machine que le navigateur utilisé pour accéder à la console d'administration.

4. Entrez le **Mot de passe du fichier de clés**.

   Avertissement : Vous pouvez chiffrer les clés privées (personnelles) contenues dans un fichier de clés à l'aide d'un mot de passe. Le fichier de clés lui-même est protégé par mot de passe. Toutefois, le service de clés ne retient qu'un mot de passe pour un fichier de clés. En conséquence,la clé privée chiffrée et son fichier de clés doivent partager le même mot de passe.
5. Entrez le **Nom du fichier de clés**.
6. Indiquez le type.
   - **Clés de signature/chiffrement**
   - **Certificats de CA**

   Le type indique le type de clé ou de certificat que vous souhaitez stocker dans le fichier de clés. Si, par exemple, vous souhaitez utiliser ce fichier de clés pour stocker des certificats de votre partenaire, sélectionnez Certificats de CA. Pour utiliser le fichier de clés afin de stocker vos propres clés de signature, choisissez Clés de signature/chiffrement.

   Le type est indiqué à titre d'information uniquement et ne vous empêche pas d'ajouter d'autres types de clés dans le fichier de clés. Toutefois, l'usage de types cohérents (c'est-à-dire un type privé et un type public) permet de simplifier l'organisation et la localisation des clés.
7. Répétez ces étapes pour chaque fichier de clés à créer pour vos certificats et pour les certificats de votre partenaire.
8. Cliquez sur **Terminer**.

Que faire ensuite

Votre fichier de clés est prêt pour la réception des clés et des certificats. Répétez l'opération pour importer d'autres fichiers de clés, ou passez à l'étape «Planification de la sécurité au niveau message».

Planification de la sécurité au niveau message

Pour commencer le processus de configuration de la sécurité des messages pour votre environnement, vous devez déterminer vos besoins.

Discutez avec votre partenaire de vos environnements. Utilisez les tables de la liste de contrôle ci-dessous pendant la discussion. Pensez à noter vos consignes dans les tables des listes de contrôle.

Les options disponibles pour sécuriser le contenu des messages dépendent de la norme SAML et du profil que vous utilisez dans votre fédération. Les options de sécurité dépendent également parfois de votre rôle (fournisseur d'identité ou fournisseur de service) dans la fédération.

En général, vous pouvez signer des messages, des assertions et valider les signatures de votre partenaire. Dans une fédération SAML 2.0, tous les partenaires doivent également chiffrer les données qu'ils s'envoient les uns aux autres. Chaque partenaire doit ensuite déchiffrer les données afin de pouvoir les utiliser dans la fédération.

- Pour signer, utilisez votre clé privée provenant d'une paire de clés publique/privée.
- Pour valider, utilisez la clé publique de votre partenaire qui correspond à la clé privée que le partenaire utilise pour signer les données.

- Pour chiffrer, utilisez la clé publique de votre partenaire qui correspond à la clé privée que le partenaire utilise pour chiffrer les données. De la même manière, donnez votre clé publique à votre partenaire. Votre partenaire doit l'utiliser pour chiffrer des données pour vous, puis vous devez déchiffrer ces données à l'aide de votre clé privée correspondante.

Utilisez la liste de contrôle suivante pour déterminer les paires de clés publique/privées dont vous avez besoin et les clés que vous devez échanger avec votre partenaire.

Vos clés

Utilisez la *clé privée* d'une paire de clés publique/privée pour exécuter les actions répertoriées dans le tableau suivant. Vous pouvez utiliser la même clé pour toutes ces actions, ou une différente par action. Toutes les clés sont facultatives et accessibles à l'ensemble des normes et règles de fournisseurs SAML, *sauf mention contraire* dans la colonne Remarques.

Tableau 5. Vos clés

But de la clé	**Alias de la paire clé publique/clé privée**	**Fichier de stockage de la clé**	**Remarques**
Clé de signature des messages			Requise si vous êtes un fournisseur d'identité dans une fédération SAML 1.x. **Remarque :** Dans SAML 2.0, la même clé est utilisée pour la signature des messages et assertions.
Clé de signature des assertions			Requise pour les fournisseurs d'identité. **Remarque :** Dans SAML 2.0, la même clé est utilisée pour la signature des messages et assertions.
Clé de déchiffrement			Requise dans SAML 2.0. Non disponible dans les fédérations SAML 1.x.

Clés requises fournies par votre partenaire

Utilisez la *clé publique* de la paire de clés publique/privée de votre partenaire pour exécuter les actions répertoriées. La colonne Remarques indique si une clé est requise, ou si elle ne peut pas être utilisée à cause d'un rôle de fournisseur spécifique ou de la spécification SAML utilisée par la fédération.

Dans la plupart des cas, vous pouvez obtenir ces clés auprès de votre partenaire par le biais d'un fichier de métadonnées. Toutefois, si vous utilisez une fédération SAML 1.x, vous devez obtenir ces clés manuellement. Envisagez de partager ce tableau avec votre partenaire, afin de vous assurer que celui-ci a pris connaissance des clés qu'il doit vous fournir.

Tableau 6. Clés requises fournies par votre partenaire

But de la clé	Alias de clé publique	Fichier de clés certifiées pour le stockage	Remarques
Clé de validation pour les signatures de messages			Correspond à la clé de signature de votre partenaire. Requise si votre partenaire doit signer des messages.
Clé de validation pour les signatures d'assertion			Correspond à la clé de signature de votre partenaire. Requise si votre partenaire doit signer des assertions. Non disponible pour les fournisseurs d'identité SAML 1.x.
Clé de chiffrement			Correspond à la clé de déchiffrement de votre partenaire. Requise dans SAML 2.0. Non disponible dans les fédérations SAML 1.x.

Clés que vous devez fournir à votre partenaire

Fournissez la clé publique de votre paire de clés publique/privée à votre partenaire de sorte qu'il puisse exécuter les actions répertoriées. La colonne Remarques indique si une clé est requise, ou si elle ne peut être utilisée en raison d'un rôle de fournisseur spécifique, ou de la spécification SAML en vigueur dans la fédération.

Dans la plupart des cas, vous devez fournir ces clés en exportant les propriétés de votre fédération dans un fichier de métadonnées que votre partenaire doit importer dans sa configuration. Toutefois, si vous utilisez une fédération SAML 1.x, vous devez exporter ces clés à partir de votre fédération et les fournir manuellement à votre partenaire.

Tableau 7. Clés que vous devez fournir à votre partenaire

But de la clé	Alias de la paire clé publique/clé privée	Fichier de stockage de la clé	Remarques
Clé de validation pour les signatures de messages			Correspond à votre clé de signature. Requise si vous devez signer des messages.
Clé de validation pour les signatures d'assertion			Correspond à votre clé de signature. Requise si vous devez signer des assertions. Non disponible pour les fournisseurs d'identité SAML 1.x.
Clé de chiffrement			Correspond à votre clé de déchiffrement. Requise dans SAML 2.0. Non disponible dans les fédérations SAML 1.x.

Obtention de vos clés et certificats

Après avoir déterminé quelles clés et quels certificats vous devez utiliser pour la signature et le déchiffrement, vous devez obtenir ces éléments.

Pourquoi et quand exécuter cette tâche

Vous devez en principe obtenir les clés privées suivantes :

Clé de signature
Si vous devez signer des messages ou des assertions, vous devez posséder une paire de clés publique/privée et utiliser la clé privée pour la signature.

Clé de déchiffrement
Si vous utilisez SAML 2.0, votre partenaire doit chiffrer les données à votre intention. Vous devez posséder une biclé publique/privée pour pouvoir exécuter cette opération. Votre partenaire utilise votre clé publique pour chiffrer les données qu'il vous envoie et vous utilisez votre clé privée pour les déchiffrer.

La méthode utilisée pour obtenir ces clés varie selon que vous utilisez un environnement de test ou un environnement de production :

- Dans un environnement de test, vous pouvez utiliser la clé de test par défaut ou créer un certificat autosigné. Voir «utilisation de la clé par défaut en tant que clé de signature et de déchiffrement», à la page 58 ou «Création de certificats d'auto-signature», à la page 58.

- Dans un environnement de production, vous devez demander vos clés à une autorité de certification. Voir «Demande de certificats signés par l'autorité de certification», à la page 59.

Les types de certificat suivants peuvent être utilisés dans le service de clés Tivoli Federated Identity Manager. Lors de l'obtention de certificats, veillez à utiliser les types pris en charge suivants :

- PEM

  Format PEM (Privacy-Enhanced Message). Il s'agit des certificats publics au format PEM.

- PKCS#12

  Public Key Cryptography Standard 12 : norme d'échange d'informations personnelles.

  Avant d'utiliser des certificats PKCS#12, vous devez mettre à jour la règle de cryptographie. Voir «Mise à jour de la règle de cryptographie», à la page 68.

utilisation de la clé par défaut en tant que clé de signature et de déchiffrement

Vous pouvez, dans un environnement de test, utiliser la clé d'essai contenue dans le fichier DefaultKeyStore en tant que clé de signature et clé de déchiffrement.

Pourquoi et quand exécuter cette tâche

Assurez-vous que la clé d'essai se trouve dans le fichier de clés. Aucune autre préparation n'est nécessaire pour cette clé.

Création de certificats d'auto-signature

Dans un environnement de test, vous pouvez utiliser un certificat autosigné pour votre signature et comme clé de déchiffrement. Vous pouvez également utiliser un certificat d'authentification du client que vous présentez au serveur lors d'une communication SSL.

Pourquoi et quand exécuter cette tâche

Un certificat d'auto-signature est une paire de clés publique/privée générée de manière aléatoire et signée par sa propre clé privée. Vous pouvez créer un certificat autosigné à l'aide de l'utilitaire de Tivoli Federated Identity Manager. Vous pouvez également employer un autre utilitaire de création de clés. La procédure suivante décrit le maniement de l'utilitaire Tivoli Federated Identity Manager.

Remarque : Cette procédure n'est prise en charge que par WebSphere Application Server Version 6.1.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.

   Le panneau Fichiers de clés s'ouvre.
3. Sélectionnez un fichier de clés dans le tableau Fichier de clés. L'option **Afficher les clés** est activée.
4. Cliquez sur **Afficher les clés**. Le panneau Mot de passe s'affiche.
5. Entrez le mot de passe du fichier de clés, puis cliquez sur **OK**.

6. Cliquez sur **Créer un certificat d'auto-signature**. Le panneau Créer un certificat d'auto-signature s'ouvre.
7. Entrez la valeur appropriée dans chaque zone.
8. Cliquez sur **OK**. Une paire de clés publique/privée est ajoutée au fichier de clés.
9. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**

Que faire ensuite

Pour vérifier que le certificat a été créé, répétez les étapes 1 à 5, à la page 58.

Demande de certificats signés par l'autorité de certification

Dans un environnement de production, vous avez besoin de recevoir des certificats de signature, de déchiffrement et d'authentification du client émis et signés par une autorité de certification. Vous pouvez émettre une demande de signature de certificat à l'aide de la console.

Avant de commencer

Vérifiez que vous disposez d'un fichier de clés dans lequel vous pouvez stocker la demande de certificat et, par la suite, le certificat.

Pourquoi et quand exécuter cette tâche

Une demande de certificat signé (RSC) est un fichier électronique que vous envoyez par email, via FTP ou par tout autre moyen de communication exigé par l'autorité de certification, à une CA ,VeriSign, Thawte, etc., pour requérir un certificat signé par elle.

L'autorité de certification utilise les données contenues dans la demande de signature de certificat, puis génère le certificat et le signe à l'aide de sa propre clé privée.

Cette signature valide la fiabilité du certificat.

Une RSC contient les données suivantes :

- L'identité du demandeur (vous) sous forme d'un nom distinctif du sujet.
- Les extensions du certificat (le cas échéant).
- La clé publique du certificat.
- Les algorithmes à utiliser pour la signature et la clé.

Une fois la demande générée, un certificat d'auto-signature temporaire est créé dans le fichier de clés. Il sera remplacé par le certificat signé par la CA lorsque vous recevrez celui-ci.

Remarque : Cette procédure n'est prise en charge que par WebSphere Application Server Version 6.1.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
   Le panneau Fichiers de clés s'ouvre.

3. Sélectionnez un fichier de clés dans le tableau Fichier de clés. L'option **Afficher les clés** est activée.
4. Cliquez sur **Afficher les clés**. Le panneau Mot de passe s'affiche.
5. Entrez le mot de passe du fichier de clés.
6. Cliquez sur **OK**.
7. Cliquez sur **Créer une demande de certificat**. Le panneau Créer une demande de certifcat s'affiche.
8. Entrez la valeur appropriée dans chaque zone.
9. Cliquez sur **OK**. La fenêtre Requête de signature de certificat générée s'affiche.
10. Copiez et collez le texte de la demande dans un fichier texte ou cliquez sur **Exporter la requête de signature de certificat** pour le télécharger. Le fichier enregistré ou téléchargé peut désormais être envoyé à une CA.
11. Cliquez **Terminé** après avoir enregistré le fichier. Une paire de clés publique/privée est ajoutée au fichier de clés et un fichier contenant les données BASE64 codées est créé. Le certificat autosignétemporaire doit être remplacé par le certificat signé par l'autorité de certification.
12. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**

Que faire ensuite

Répétez cette procédure pour chaque certificat que vous souhaitez demander. Par exemple, vous pouvez demander un certificat distinct pour chaque activité ou utiliser un seul certificat pour toutes les activités. Les différentes activités sont la signature, le chiffrement et l'authentification client.

Une fois toutes vos demandes de signature de certificat créées, suivez les instructions de l'autorité de certification pour lui transmettre le fichier de la demande. Ensuite, poursuivez la réception du certificat émis par la CA en suivant la procédure décrite dans «Réception d'un certificat signé de la part d'une autorité de certification», à la page 62.

Ajout de vos certificats à votre fichier de clés

Avant d'établir votre fédération, vous devez ajouter les clés permettant de signer et de déchiffrer votre fichier de clés.

Pourquoi et quand exécuter cette tâche

La méthode permettant d'ajouter vos clés à votre fichier de clés dépend de la manière dont vous les avez obtenues :

Création d'un certificat d'auto-signature
Si vous employez l'utilitaire de Tivoli Federated Identity Manager pour créer un certificat autosigné, le certificat est automatiquement importé dans votre fichier de clés. Si vous avez créé un certificat autosigné, mais avez employé un utilitaire autre que celui fourni avec Tivoli Federated Identity Manager, importez votre certificat dans le fichier de clés comme indiqué dans «Importation d'un certificat», à la page 61.

Demande d'un certificat signé
Si vous avez généré une demande de signature de certificat et envoyé celle-ci à une autorité de certification, vous recevrez le certificat dans votre fichier de clés, comme décrit à la rubrique «Réception d'un certificat signé de la part d'une autorité de certification», à la page 62.

Importation d'un certificat

Importez un certificat si vous l'avez reçu à l'aide d'un utilitaire ou si vous l'avez obtenu manuellement d'une autorité de certification.

Pourquoi et quand exécuter cette tâche

Vous devez importer un certificat si vous l'avez reçu de l'une des manières suivantes :

- Vous avez créé un certificat autosigné à l'aide d'un utilitaire différent de celui qui est fourni avec Tivoli Federated Identity Manager.
- Vous avez obtenu manuellement un certificat auprès d'une CA

Vous devez également importer un certificat que vous avez reçu de la part de votre partenaire. Pour plus d'informations sur l'importation de certificats de partenaire, voir «Importation d'un certificat depuis votre partenaire», à la page 64.

Avertissement : Vous pouvez chiffrer les clés privées (personnelles) contenues dans un fichier de clés à l'aide d'un mot de passe. Le fichier de clés lui-même est protégé par mot de passe. Toutefois, le service de clés ne retient qu'un mot de passe pour un fichier de clés. En conséquence,la clé privée chiffrée et son fichier de clés doivent partager le même mot de passe.

Cette tâche permet d'importer les éléments suivants :

- Un certificat à partir d'un fichier PEM
- Une clé à partir d'un fichier PKCS#12.

  Remarque : Si vous devez utiliser un fichier PKCS#12, suivez également les instructions de la section «Mise à jour de la règle de cryptographie», à la page 68.

Vérifiez que votre clé ou votre certificat est prêt et disponible avant de poursuivre cette procédure.

Les clés importées sont activées par défaut.

Procédure

1. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
   Le panneau Fichiers de clés s'affiche et est activé.
2. Sélectionnez un fichier de clés dans le tableau Fichier de clés pour y stocker votre paire de clés publique/privée. Le bouton **Afficher les clés** s'affiche et est activé.

Avertissement : N'importez pas des clés privées (par exemple, des clés de signature ou de chiffrement) dans un fichier de clés **Certificat de CA**. Ce type de fichier de clés ne permet pas de stocker un mot de passe de clé (obligatoire pour les clés privées).

3. Cliquez sur **Afficher les clés**.
4. Entrez le mot de passe du fichier de clés lorsque vous y êtes invité.
5. Cliquez sur **OK**. Le panneau Clés s'ouvre. Il répertorie les éléments du fichier de clés sélectionné.
6. Cliquez sur **Importer**. L'assistant de clés démarre et affiche le panneau Bienvenue.
7. Cliquez sur **Suivant**. Le panneau Format de fichier de clés s'ouvre.

8. Sélectionnez le **format de fichier de clés** adapté au fichier à importer. Les formats sont les suivants :

   PEM)
   (Privacy-Enhanced Message) Certificat public

   PKCS#12
   Public Key Cryptography Standard 12 : norme d'échange d'informations personnelles

   JKS
   Fichier de clés Java
9. Cliquez ensuite sur **Suivant**. Le panneau **Télécharger le fichier de clés** s'affiche.
10. Indiquez le chemin d'accès à l'emplacement de la clé puis, si vous y êtes invité, entrez le mot de passe du ficher de clés. Cliquez ensuite sur **Suivant**.
11. Entrez un intitulé pour la clé et, à l'invite, sélectionnez la clé à importer.
12. Cliquez ensuite sur **Suivant**. Le panneau Récapitulatif s'affiche.
13. Cliquez sur **Terminer** pour quitter l'assistant.
14. Répétez ces étapes pour importer l'ensemble des clés et certificats que vous allez utiliser dans la fédération.

Que faire ensuite

Ajoutez ensuite les clés de votre partenaire dans votre fichier de clés certifiées. Voir «Obtention d'un certificat de votre partenaire», à la page 63.

Réception d'un certificat signé de la part d'une autorité de certification

Si vous envoyez à une autorité de certification (CA) une demande de certificat signé créée à l'aide de la console, vous pouvez recevoir le certificat de la CA dans votre fichier de clés.

Avant de commencer

Vérifiez que vous avez bien effectué la procédure de la rubrique «Demande de certificats signés par l'autorité de certification», à la page 59 et enregistré le certificat de l'autorité de certification à un emplacement accessible au service de clés.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
   Le panneau Fichiers de clés s'ouvre.
3. Sélectionnez le fichier dans lequel la RSC a été générée dans le tableau Fichiers de clés. L'option **Afficher les clés** est activée.
4. Cliquez sur **Afficher les clés**. Le panneau Mot de passe s'affiche.
5. Entrez le mot de passe du fichier de clés.
6. Cliquez sur **OK**.
7. Cliquez sur **Réception de certificat de CA**.
8. Sélectionnez l'emplacement du certificat que vous avez reçu de la CA.
9. Cliquez ensuite sur **OK**. Le certificat autosigné temporaire présent dans le fichier de clés est remplacé par le certificat signé que vous avez reçu.

10. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**

Que faire ensuite

Ajoutez ensuite les clés de votre partenaire dans votre fichier de clés certifiées. Voir «Obtention d'un certificat de votre partenaire».

Obtention d'un certificat de votre partenaire

En fonction des besoins de votre environnement, vous devez vous procurer des certificats auprès de votre partenaire.

Avant de commencer

Utilisez le formulaire «Planification de la sécurité au niveau message», à la page 54 pour déterminer les certificats que vous devez peut-être demander à votre partenaire. En règle générale, les clés publiques que vous devez obtenir auprès de votre partenaire sont les suivantes :

Clé de validation
Si votre partenaire signe les messages ou assertions et que vous devez valider ces signatures, vous devez être en possession de la clé publique correspondant à celle qui a été utilisée pour la signature.

Clé de chiffrement
Si vous devez procéder au chiffrement des données que vous envoyez à votre partenaire, vous devez obtenir une clé publique auprès de celui-ci. L'utilisation de cette clé publique est destinée au chiffrement des données, lesquelles seront déchiffrées par votre partenaire au moyen de sa clé privée correspondante.

Pourquoi et quand exécuter cette tâche

Dans une fédération SAML 2.0, vous recevez généralement les clés de validation et de chiffrement de votre partenaire sous la forme d'un fichier de métadonnées fourni par votre partenaire. Ce processus est expliqué plus en détails à la section «Importation de certificats à partir du fichier de métadonnées de votre partenaire», à la page 64. En plus de ces clés, d'autres informations de votre partenaire, telles que le nom de l'entreprise, sont incluses dans le fichier de métadonnées.

Lorsque vous créez votre partenaire dans la fédération, vous êtes invité à sauvegarder les clés du partenaire dans le fichier de clés approprié. Il convient de sauvegarder les clés du partenaire dans votre fichier de clés certifiées.

Vous pouvez recevoir les clés manuellement (par exemple, via FTP, par e-mail ou une autre méthode de transfert), puis les importer dans votre fichier de clés certifiées en suivant les instructions de la rubrique «Importation d'un certificat depuis votre partenaire», à la page 64 dans les situations suivantes :

- si vous avez déjà reçu les métadonnées de la part de votre partenaire, et que vous devez seulement recevoir un nouveau certificat de celui-ci ou
- si vous utilisez une fédération SAML 1.x.

Importation de certificats à partir du fichier de métadonnées de votre partenaire

Si un fichier de métadonnées relatif à la configuration de votre fédération vous est fourni par votre partenaire, il convient que les clés publiques de ce dernier soient incluses dans ce fichier.

Pourquoi et quand exécuter cette tâche

Selon la sécurité définie pour les messages et la spécification SAML que vous appliquez avec votre partenaire sur la fédération, il convient que le fichier de métadonnées contienne l'une ou plusieurs des clés publiques suivantes :

- Clé de validation des assertions signées, si vous êtes censé valider les assertions signées par votre partenaire
- Clé de validation des messages signés, si vous êtes censé valider les messages signées par votre partenaire
- Clé de chiffrement (dans une fédération SAML 2.0)

Voir «Planification de la sécurité au niveau message», à la page 54.

Si votre partenaire utilise Tivoli Federated Identity Manager, les clés publiques qui correspondent aux clés privées que le partenaire a définies dans sa configuration sont automatiquement ajoutées au fichier de métadonnées. Les clés publiques sont ajoutées au fichier de métadonnées lorsque le partenaire exporte sa configuration.

Si vous obtenez les clés de votre partenaire à partir d'un fichier de métadonnées, vous devez importer les métadonnées dans le cadre de l'établissement de votre fédération. Pour poursuivre, exécutez les tâches indiquées dans ce chapitre.

Importation d'un certificat depuis votre partenaire

Vous pouvez obtenir les clés publiques de votre partenaire via une connexion SSL, ou en important un fichier de métadonnées contenant les paramètres de la configuration du partenaire. Toutefois, si aucune de ces méthodes n'est disponible, vous pouvez obtenir les clés manuellement et les importer.

Avant de commencer

Assurez-vous d'avoir reçu une ou plusieurs clés publiques de la part de votre partenaire (par exemple, via FTP, par e-mail ou une autre méthode de transfert).

Pourquoi et quand exécuter cette tâche

Vous pouvez avoir besoin d'importer un certificat dans les circonstances suivantes :

- Un certificat d'auto-signature créé à l'aide d'un utilitaire autre que celui fourni avec Tivoli Federated Identity Manager
- Certificat obtenu manuellement auprès d'une autorité de certification

Il peut également être nécessaire d'importer un certificat que vous avez reçu de la part de votre partenaire. Pour plus d'informations sur l'importation de certificats de partenaire, voir «Importation d'un certificat depuis votre partenaire».

Avertissement : Vous pouvez chiffrer les clés privées (personnelles) contenues dans un fichier de clés à l'aide d'un mot de passe. Le fichier de clés lui-même est protégé par mot de passe. Toutefois, le service de clés ne retient qu'un mot de passe pour un fichier de clés. En conséquence,la clé privée chiffrée et son fichier de clés doivent partager le même mot de passe.

Cette tâche permet d'importer les éléments suivants :

- Un certificat à partir d'un fichier PEM
- Une clé à partir d'un fichier PKCS#12.

  Remarque : Si vous allez utiliser un fichier PKCS#12, veillez à suivre également les instructions de la section «Mise à jour de la règle de cryptographie», à la page 68.

Vérifiez que votre clé ou votre certificat est prêt et disponible avant de poursuivre cette procédure.

Les clés importées sont activées par défaut.

Procédure

1. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.

   Le panneau Fichiers de clés s'ouvre.
2. Sélectionnez un fichier de clés dans le tableau Fichier de clés pour y stocker votre paire de clés publique/privée. Le bouton **Afficher les clés** est activé.

Avertissement : N'importez pas des clés privées (par exemple, des clés de signature ou de chiffrement) dans un fichier de clés **Certificat de CA**. Ce type de fichier de clés ne permet pas de stocker un mot de passe de clé (obligatoire pour les clés privées).

3. Cliquez sur **Afficher les clés**.
4. Entrez le mot de passe du fichier de clés lorsque vous y êtes invité.
5. Cliquez sur **OK**. Le panneau Clés s'ouvre. Il répertorie les éléments du fichier de clés sélectionné.
6. Cliquez sur le bouton **Importer**. L'assistant de clés démarre et affiche le panneau Bienvenue.
7. Cliquez sur **Suivant**. Le panneau Format de fichier de clés s'ouvre.
8. Sélectionnez le **format de fichier de clés** adapté au fichier à importer. Les formats ci-dessous sont disponibles :

   PEM)
   (Privacy-Enhanced Message) Certificat public

   PKCS#12
   Public Key Cryptography Standard 12 : norme d'échange d'informations personnelles

   JKS
   Fichier de clés Java

   Le panneau **Télécharger le fichier de clés** s'affiche.
9. Cliquez sur **Suivant**.
10. Indiquez le chemin d'accès à l'emplacement de la clé puis, si vous y êtes invité, entrez le mot de passe du ficher de clés.
11. Cliquez sur **Suivant**.

12. Entrez un intitulé pour la clé et, à l'invite, sélectionnez la clé à importer.
13. Cliquez sur **Suivant**. Le panneau Récapitulatif s'affiche.
14. Cliquez sur **Terminer** pour quitter l'assistant.
15. Répétez ces étapes pour importer l'ensemble des clés et certificats que vous allez utiliser dans la fédération.

Que faire ensuite

Transmettez vos clés à votre partenaire. Voir «Transmission de certificats au partenaire».

Transmission de certificats au partenaire

En fonction des besoins de votre environnement, vous devez peut-être fournir une clé à votre partenaire.

Avant de commencer

Reportez-vous à la rubrique «Planification de la sécurité au niveau message», à la page 54 pour connaître les certificats que vous devrez peut-être fournir à votre partenaire. En règle générale, les clés publiques que vous devez obtenir sont les suivantes :

Clé de validation
Si vous devez signer des messages ou des assertions et que votre partenaire doit valider ces signatures, vous devez fournir la clé publique correspondant à celle qui a été utilisée pour la signature.

Clé de chiffrement
Pour que votre partenaire puisse chiffrer les données à votre intention, vous devez lui fournir une clé publique. Votre partenaire utilise cette clé publique pour le chiffrement des données que vous pourrez déchiffrer au moyen de votre clé privée correspondante.

Pourquoi et quand exécuter cette tâche

Placez de votre clé de validation et de chiffrement dans un fichier de métadonnées que vous créez et fournissez à votre partenaire. En plus de ces clés, d'autres informations de votre partenaire, telles que le nom de l'entreprise, sont incluses dans le fichier de métadonnées. La création de ce fichier a lieu ultérieurement au cours du processus de configuration. Pour plus d'informations, voir «Exportation de certificats dans un fichier de métadonnées».

Dans une fédération SAML 1.0, vous avez également la possibilité de fournir manuellement ces informations à votre partenaire. Pour plus d'informations, voir «Exportation d'un certificat», à la page 67. Vous pouvez également faire appel à la méthode manuelle si vous avez déjà fourni vos métadonnées à votre partenaire et que vous avez besoin de délivrer un certificat mis à jour par vos soins.

Exportation de certificats dans un fichier de métadonnées

Si vous fournissez à votre partenaire un fichier de métadonnées sur la configuration de votre fédération, vos clés publiques doivent être incluses dans ce fichier.

Pourquoi et quand exécuter cette tâche

Selon la sécurité définie pour les messages et la spécification SAML que vous appliquez avec votre partenaire sur la fédération, il convient que le fichier de métadonnées contienne l'une ou plusieurs des clés publiques suivantes :

- Clé que le partenaire utilise pour la validation des assertions signées, si vous signez les assertions
- Clé que le partenaire utilise pour la validation des messages signées, si vous signez les messages
- Clé que le partenaire utilise pour le chiffrement des messages à votre intention (dans une fédération SAML 2.0)

Voir «Planification de la sécurité au niveau message», à la page 54.

Si votre partenaire utilise Tivoli Federated Identity Manager, vous pouvez exporter votre configuration vers un fichier de métadonnées, y compris vos clés, afin que votre partenaire puisse importer le fichier.

Si vous choisissez cette méthode pour fournir les clés à votre partenaire, exportez les métadonnées dans le cadre de l'établissement de votre fédération. Pour poursuivre, exécutez les tâches indiquées dans ce chapitre.

Exportation d'un certificat

Exportez un certificat si vous n'êtes pas en mesure de fournir à votre partenaire un fichier de métadonnées contenant vos clés.

Procédure

1. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
   Le panneau Fichiers de clés s'ouvre.
2. Sélectionnez le fichier de clés approprié dans le tableau Fichier de clés. Vous êtes invité à entrer le mot de passe du fichier de clés.
3. Entrez le mot de passe.
4. Cliquez sur **OK**. Le bouton **Afficher les clés** est activé.
5. Cliquez sur **Afficher les clés**. Le panneau Clés s'ouvre. Il répertorie les éléments du fichier de clés sélectionné.
6. Sélectionnez les clés à exporter.
7. Cliquez sur le bouton **Exporter**. Le panneau Exporter la clé s'affiche.
8. Sélectionnez le format de la clé à exporter.

   (PEM)
   (Privacy-Enhanced Message) Certificat public

   PKCS#12
   Public Key Cryptography Standard 12 : norme d'échange d'informations personnelles
9. Vérifiez que la case **Inclure une clé privée** *n'est pas* cochée. Vous devez être la seule personne à détenir votre clé privée.
10. Cliquez sur **Télécharger la clé**.
11. Lorsque vous y êtes invité, entrez le nom de fichier de la clé exportée.
    Par exemple : `maclépublique.pem`
    (Facultatif) Cliquez sur **Parcourir** pour rechercher le fichier sur le système de fichiers.

12. Cliquez sur **Annuler** pour quitter.

Mise à jour de la règle de cryptographie

L'utilisation de la technologie de chiffrement est contrôlée par la législation des Etats-Unis. Les SDK IBM Java comprennent des fichiers de règles de juridiction strictes, mais limitées. Avant d'utiliser les fichiers PKCS#12 avec Tivoli Federated Identity Manager, vous devez vous procurer les fichiers de règles JCE (Java Cryptography Extension) de juridiction illimitée.

Pourquoi et quand exécuter cette tâche

Pour consulter les informations de sécurité relatives aux kits de développement de logiciels IBM Java, accédez à l'adresse URL suivante :

`http://www.ibm.com/developerworks/java/jdk/security/index.html`

Pour vous procurer les fichiers de règles de juridiction illimitée, procédez comme suit :

Procédure

1. Mettez à jour WebSphere à l'aide de fichiers de règles JCE (Java Cryptography Extension) non limitées. Accès : http://www.ibm.com/developerworks/java/jdk/security/index.html
2. Sélectionnez le lien vers le SDK qui correspond à votre environnement, par exemple, pour Java 1.5, le SDK est J2SE 5.0. Une page affichant l'en-tête Security Information (informations de sécurité) apparaît.
3. Sélectionnez le lien suivant : **IBM SDK Policy Files**.

   Remarque : Lorsque vous cliquez sur ce lien, vous êtes redirigé vers le fichier de règles contenu dans le kit SDK compatible avec votre version de Java. Il est à noter, toutefois, que le numéro de version du kit SDK n'est pas nécessairement le même que celui de la version de Java utilisée. Par exemple, pour Java 1.5, vous pouvez être redirigé vers le kit SDK 1.4.
4. Vous êtes invité à vous connecter à l'aide de votre ID utilisateur et mot de passe IBM. Si vous ne disposez pas d'un ID utilisateur et d'un mot de passe IBM, vous devez vous inscrire. Suivez le lien d'inscription figurant sur la page de connexion.
5. Connectez-vous.
6. A l'invite, sélectionnez le fichier .zip correspondant à la version de Java que vous utilisez.
7. Cliquez sur **Continuer** pour commencer le téléchargement.
8. Décomprimez le fichier .zip. Les fichiers JAR sont les suivants :
   - local_policy.jar
   - US_export_policy.jar
9. Placez les fichiers dans le répertoire suivant :

   rep_installation_composant_exec_Java/jre/lib/security

   Par exemple, il se peut que le composant d'exécution Java ait été installé dans le cadre de la version imbriquée de WebSphere Application Server. Dans ce cas, le répertoire peut être le suivant :

   /opt/IBM/FIM/*ewas*/java/jre/lib/security

Suppression de fichiers de clés par défaut

Des fichiers de clés et des certificats par défaut sont inclus dans Tivoli Federated Identity Manager. Si vous avez créé vos propres fichiers de clés, vous pouvez être amené à supprimer les fichiers de clés par défaut. Toutefois, cette tâche est facultative.

Procédure

1. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
   Le panneau Fichiers de clés s'ouvre.
2. Sélectionnez **DefaultKeyStore**.
3. Cliquez sur **Supprimer**. Un message vous invite à confirmer que vous voulez supprimer le fichier de clés indiqué.
4. Cliquez sur **OK** pour supprimer le magasin de clés.
5. Sélectionnez **DefaultTrustedKeyStore**.
6. Cliquez sur **Supprimer**. Un message vous invite à confirmer que vous voulez supprimer le fichier de clés indiqué.
7. Cliquez sur **OK** pour supprimer le magasin de clés.

Activation de la vérification du retrait de certificat

Vous pouvez recourir au gestionnaire d'accréditation IbmPKIX pour déterminer la validité des certificats serveur. Si vous activez cette fonction, le gestionnaire d'accréditation vérifie le certificat présenté par le serveur SSL lorsque le client SOAP établit une connexion SSL. Le gestionnaire d'accréditation vérifie ensuite les certificats utilisés pour la signature, la validation, le chiffrement et le déchiffrement des messages XML. S'il découvre que le certificat a été retiré, l'opération de fédération échoue.

Pourquoi et quand exécuter cette tâche

L'activation de la vérification du retrait de certificat nécessite les procédures suivantes :

- «Activation du contrôle de la révocation de certificat sous WebSphere».
- «Activation du gestionnaire d'accréditation IbmPKIX pour connexion SSL», à la page 71.
- «Activation du gestionnaire d'accréditation IbmPKIX pour la signature, la validation, le chiffrement et le déchiffrement des messages XML», à la page 72

Activation du contrôle de la révocation de certificat sous WebSphere

Vous devez activer certains paramètres dans WebSphere Application Server pour pouvoir configurer Tivoli Federated Identity Manager et procéder à la vérification de révocation de certificat.

Pourquoi et quand exécuter cette tâche

Activez les paramètres en fonction du type WebSphere Application Server que vous utilisez. Choisissez la procédure appropriée suivant votre installation :

Embedded WebSphere Application Server (version intégrée)
«Activation de CRC sur une instance intégrée de WebSphere Application Server», à la page 70,

Serveur WebSphere Application Server existant
«Activation de CRC sur une instance existante de WebSphere Application Server».

Activation de CRC sur une instance intégrée de WebSphere Application Server

Si vous utilisez une version intégrée de WebSphere Application Server, vous devez activer les paramètres requis pour le contrôle du retrait de certificats (CRC) avant de configurer le contrôle de retrait de certificats dans votre environnement Tivoli Federated Identity Manager.

Avant de commencer

Avertissement : Utilisez cette procédure uniquement si vous avez installé Tivoli Federated Identity Manager à l'aide de la version intégrée de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Pour activer les paramètres appropriés, procédez comme suit :

Procédure

1. Ouvrez une invite de commande.
2. Démarrez l'outil wsadmin de WebSphere Application Server. A partir de votre profil WebSphere, entrez la commande appropriée afin que votre système d'exploitation démarre l'outil :

   Windows
   ```
   wsadmin.bat
   ```

   AIX, Linux, HP-UX ou Solaris
   ```
   wsadmin.sh
   ```

   Remarque : Pour de plus amples informations sur les options pouvant être définies à l'aide de l'outil wsadmin, reportez-vous au centre d'information en ligne à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp.
3. A l'invite de commande, exécutez les commandes suivantes en remplaçant *server1* par le nom de votre serveur :
   ```
   set jvm [$AdminConfig getid
     /Server:server1/JavaProcessDef:/JavaVirtualMachine:/]
    $AdminConfig modify $jvm {{genericJvmArguments
     "-Dcom.ibm.jsse2.checkRevocation=true
      -Dcom.ibm.security.enableCRLDP=true"}}
    $AdminConfig save
   ```
4. Redémarrez WebSphere Application Server.

Que faire ensuite

Allez aux étapes de la section «Activation du gestionnaire d'accréditation IbmPKIX pour connexion SSL», à la page 71.

Activation de CRC sur une instance existante de WebSphere Application Server

Si vous avez installé Tivoli Federated Identity Manager sur une version existante de WebSphere Application Server, vous devez activer le gestionnaire

d'accréditation IbmPKIX avant de configurer le contrôle de retrait de certificat dans votre environnement Tivoli Federated Identity Manager.

Procédure

1. Connectez-vous à la console de WebSphere Application Server.
2. Cliquez sur **Serveurs** > **Serveurs d'application**.
3. Sélectionnez votre serveur.
4. Cliquez sur **Gestion des processus et Java** > **Définition des processus** > **Machine virtuelle Java**.
5. Sous Arguments JVM génériques, ajoutez le texte suivant :

   ```
   dcom.ibm.jsse2.checkRevocation=true
   Dcom.ibm.security.enableCRLDP=true
   ```

6. Redémarrez WebSphere Application Server.

Que faire ensuite

Allez aux étapes de la section «Activation du gestionnaire d'accréditation IbmPKIX pour connexion SSL».

Activation du gestionnaire d'accréditation IbmPKIX pour connexion SSL

Activez Tivoli Federated Identity Manager pour vérifier la révocation des certificats utilisés pour les connexions SSL, à l'aide du gestionnaire d'accréditation IbmPKIX.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution**.
3. Cliquez sur **Propriétés personnalisées de l'environnement d'exécution** dans le panneau Gestion des noeuds d'exécution. Le panneau Propriétés personnalisées de l'environnement d'exécution s'affiche.
4. Cliquez sur **Créer**. Un élément est ajouté à la liste des propriétés, avec le nom **nouvelle clé** et la valeur **nouvelle valeur**.
5. Cliquez de nouveau sur **Créer**. Un autre élément est ajouté à la liste des propriétés, avec le nom **nouvelle clé** et la valeur **nouvelle valeur**.
6. Sélectionnez une des propriétés de la marque de réservation.
7. Tapez `com.tivoli.am.fim.soap.client.jsse.provider` dans la zone **Nom**. N'insérez pas d'espace dans cette zone.
8. Tapez `JSSE2` dans la zone **Valeur**.
9. Sélectionnez la propriété suivante de la marque de réservation.
10. Tapez `com.tivoli.am.fim.soap.client.trust.provider` dans la zone **Nom**.
11. Tapez `IbmPKIX` dans la zone **Valeur**.
12. Cliquez sur **OK** pour appliquer les modifications effectuées et quitter le panneau.

Que faire ensuite

Utilisez le gestionnaire d'accréditation IbmPKIX pour vous assurer de la validité des certificats employés dans les opérations de sécurité XML.

Activation du gestionnaire d'accréditation IbmPKIX pour la signature, la validation, le chiffrement et le déchiffrement des messages XML

Vous pouvez configurer IBM Tivoli Federated Identity Manager pour utiliser le gestionnaire d'accréditation IbmPKIX. Ce gestionnaire garantit que les certificats utilisés pour des opérations de sécurité XML telles que la signature, le chiffrement, la validation et le déchiffrement sont conformes à leurs listes de révocation de certificats.

Procédure

1. Connectez-vous à la console Integrated Solutions Console.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution**.
3. Cliquez sur **Propriétés personnalisées de l'environnement d'exécution**. Le panneau Propriétés personnalisées de l'environnement d'exécution s'affiche.
4. Cliquez sur **Créer**. Un élément est ajouté à la liste des propriétés, avec le nom **nouvelle clé** et la valeur **nouvelle valeur**.
5. Sélectionnez la propriété de la marque de réservation créée.
6. Tapez `kessjksservice.revocation.enabled` dans la zone **Nom**. N'insérez pas d'espace dans cette zone.
7. Tapez `true` dans la zone **Valeur**.
8. Cliquez sur **OK** pour appliquer les modifications effectuées et quitter le panneau.

Chapitre 9. Configuration de la sécurité du transport

Pour permettre la protection du message lors de sa communication (ou de son transport) entre les partenaires, SAML requiert une connexion SSL (Secure Sockets Layer) avec authentification sur le serveur et, dans certains cas, une authentification réciproque.

Pourquoi et quand exécuter cette tâche

Vous pouvez assurer la sécurité d'un environnement Tivoli Federated Identity Manager en activant le protocole SSL sur l'instance de WebSphere Application Server sur laquelle le composant d'exécution et du service de gestion est installé. En outre, si vous avez le rôle de client dans une communication SSL où l'authentification réciproque repose sur un certificat client, vous devez également configurer ce certificat.

La procédure générale permettant d'activer l'authentification côté client et côté serveur comprend les tâches suivantes :

Procédure

1. «Activation de SSL sur WebSphere Application Server», à la page 74.

   Remarque : Si vous tenez le rôle de fournisseur de services dans une fédération SAML 1.x, vous serez le *client* dans une configuration SSL. Il n'est donc pas nécessaire de configurer le protocole *SSL serveur*. Reportez-vous à la procédure de configuration des certificats client dans la section «Configuration des certificats client», à la page 83.

   L'activation SSL sur un serveur comprend les tâches annexes suivantes :

   a. «Création d'une demande de certificat», à la page 74.
   b. «Réception d'un certificat signé émis par une autorité de certification», à la page 75.
   c. «Association d'un certificat à la configuration SSL», à la page 76.
   d. Vous pouvez, en option, effectuer les étapes de la section «Suppression du certificat par défaut», à la page 77.
   e. «Extraction d'un certificat en vue de le partager avec votre partenaire», à la page 78.
2. «Configuration des exigences relatives à l'authentification client», à la page 78. Les options liées aux exigences d'authentification sont les suivantes :
   - Aucune authentification
   - Authentification de base, au cours de laquelle un nom d'utilisateur et un mot de passe sont demandés
   - Authentification par certificat client
3. Si vous tenez le rôle de client dans la fédération et que votre partenaire exige votre authentification à l'aide d'un certificat client, vous devez également accomplir les étapes de la rubrique «Configuration des certificats client», à la page 83.

Activation de SSL sur WebSphere Application Server

Pour vous assurer que les messages sont sécurisés lorsqu'ils sont communiqués entre les partenaires de la fédération, activez SSL sur l'instance de WebSphere Application Server sur laquelle le composant d'exécution et du service de gestion est installé.

Avant de commencer

Remarque : Si vous tenez le rôle de fournisseur de services dans une fédération SAML 1.x, vous êtes toujours le client dans une configuration SSL. Il ne vous est donc pas nécessaire de configurer le protocole SSL sur votre serveur. Reportez-vous à la procédure de configuration des certificats client dans la section «Configuration des certificats client», à la page 83.

Création d'une demande de certificat

Pour assurer une communication SSL, les serveurs exigent un certificat personnel (également appelé certificat serveur) signé par une autorité de certification (CA). Vous devez au préalable créer une demande de certificat personnel pour obtenir un certificat signé par une CA.

Avant de commencer

Il doit exister un fichier de clés, qui contient la demande de certificat et qui contiendra, par la suite, le certificat. Vous pouvez utiliser le fichier de clés par défaut de WebSphere Application Server, NodeDefaultKeyStore, ou en créer un. Pour les instructions de création d'un fichier de clés, voir le centre de documentation de WebSphere Application Server 6.1 à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp.

Pourquoi et quand exécuter cette tâche

Exécutez les tâches ci-dessous sur la console. Pour plus d'informations, voir la rubrique du centre de documentation de WebSphere relative à la création d'une demande de certificat.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Certificat SSL et gestion des clés**.
3. Sous l'option **Articles liés** de droite, cliquez sur **Magasins de clés et certificats**.
4. Cliquez sur le nom du fichier de clés dans lequel le certificat sera enregistré, par exemple **NodeDefaultKeyStore**.
5. Cliquez sur **Demandes de certificats personnels (Personal certificate requests)** sous Propriétés supplémentaires (Additional Properties).
6. Cliquez sur **Nouveau**.
7. Dans la zone **Fichier de demande de certificat (File for certificate request)**, entrez le chemin d'accès complet de l'emplacement d'enregistrement de la demande de certificat ainsi que le nom du fichier. Le fichier comporte une extension .arm. Exemple : `c:\servercertreq.arm` (sur un serveur Windows).
8. Entrez un alias pour le certificat dans la zone **Intitulé de clé**. L'alias est le nom que vous donnez à votre demande de certificat afin de l'identifier dans le magasin de clés.

9. Entrez une valeur de nom usuel. Celui-ci est le nom de l'entité que le certificat représente. Le nom usuel correspond souvent au nom d'hôte DNS dans lequel réside le serveur.
10. Dans la zone **Unité organisationnelle**, entrez la partie du nom distinctif relative à l'unité organisationnelle.
11. Dans la zone **Localité**, entrez la partie du nom distinctif relative à la localité.
12. Dans la zone **Etat ou province**, entrez la partie du nom distinctif relative à l'Etat.
13. Dans la zone **Code postal**, entrez la partie du nom distinctif relative au code postal.
14. Dans la liste **Pays ou région**, sélectionnez la partie du nom distinctif correspondant aux deux lettres du code pays.
15. Cliquez sur **Appliquer**.

    Avertissement :
    Les outils du fichier de clés (notamment iKeyman et keyTool) ne peuvent pas recevoir les certificats signés générés par les demandes de certificat issues de WebSphere Application Server. De même, WebSphere Application Server n'accepte pas les certificats générés par les demandes de certificat issues d'autres utilitaires de fichiers de clés.
16. Cliquez sur **Sauvegarder**. La demande de certificat est créée à l'adresse de fichier spécifiée dans le fichier de clés. Elle fonctionne comme une marque de réservation temporaire du certificat signé jusqu'à ce que vous receviez manuellement le certificat dans le fichier de clés.
17. Envoyez le fichier .arm de la demande de certificat à une autorité de certification pour qu'elle la signe. Chaque autorité de certification a adopté une méthode privilégiée pour la réception des demandes. Utilisez celle qui est recommandée par l'autorité de certification à laquelle vous envoyez votre demande.
18. Gardez une copie de sauvegarde de votre fichier de clés jusqu'à réception du certificat demandé. Vous pouvez rechercher votre fichier de clés à l'aide du chemin d'accès figurant sur la console. Copiez-le dans un autre emplacement en guise de sauvegarde.

Que faire ensuite

Pour terminer la procédure d'obtention d'un certificat signé pour votre serveur, recevez le certificat de la CA, comme indiqué dans «Réception d'un certificat signé émis par une autorité de certification».

Réception d'un certificat signé émis par une autorité de certification

Lorsqu'une autorité de certification (CA) reçoit une demande, elle émet un nouveau certificat sous forme de marque de réservation temporaire d'un certificat émis par la CA. Le nouveau certificat est reçu dans un fichier de clés qui génère le certificat personnel signé par la CA, que WebSphere Application Server peut utiliser pour la sécurité SSL.

Avant de commencer

La demande de certificat doit être déjà créée et figurer dans un fichier de clés de WebSphere comme indiqué dans «Création d'une demande de certificat», à la page 74. De même, le certificat doit provenir de la CA et être placé sur votre ordinateur pour vous permettre de le recevoir dans le fichier de clés.

WebSphere Application Server ne peut recevoir que les certificats générés par une demande de certificat de WebSphere Application Server. Il ne peut pas recevoir les certificats demandés à l'aide d'autres outils du fichiers de clés, comme iKeyman ou keyTool.

Pourquoi et quand exécuter cette tâche

Exécutez les tâches ci-dessous sur la console. Pour plus d'informations, voir la rubrique du centre de documentation de WebSphere, http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp, concernant la réception d'un certificat émis par une autorité de certification.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Certificat SSL et gestion des clés** > **Gérer les configurations de sécurité du noeud final**.
3. Cliquez sur le nom de votre noeud dans l'arborescence **Entrant**.
4. Cliquez sur le bouton **Gérer les certificats**.
5. Cliquez sur **Recevoir un certificat émis pas une autorité de certification**.
6. Entrez le chemin d'accès complet et le nom du fichier du certificat que vous avez reçu de l'autorité de certification.
7. Sélectionnez le type de données par défaut dans la liste.
8. Cliquez sur **Valider**, puis sur **Enregistrer**. Le fichier de clés contient un nouveau certificat personnel émis par une CA. La configuration SSL peut désormais utiliser le nouveau certificat personnel signé par la CA.

Que faire ensuite

Associez le certificat à votre configuration SSL. Voir «Association d'un certificat à la configuration SSL».

Association d'un certificat à la configuration SSL

Après avoir ajouté un certificat signé à votre fichier de clés, vous devez lui associer les paramètres de configuration SSL de votre serveur.

Pourquoi et quand exécuter cette tâche

Lorsque vous installez WebSphere Application Server 6.1 et Tivoli Federated Identity Manager, deux configurations SSL sont créées sur WebSphere Application Server :

- NodeDefaultSSLSettings
- FIMSOAPEndpointSSLSettings

NodeDefaultSSLSettings est le paramètre de configuration SSL par défaut défini par WebSphere Application Server. Il est destiné à la stratégie SSL de votre serveur WebSphere. La configuration FIMSOAPEndpointSSLSettings est ajoutée par Tivoli Federated Identity Manager pour vous permettre d'avoir une stratégie SSL distincte, spécialement dédiée à la communication des messages SOAP avec votre partenaire de fédération.

Après l'installation, les deux configurations utilisent le certificat à signature automatique `par défaut` présent dans NodeDefaultKeystore.

Lorsque vous demandez et recevez un certificat personnel signé, les paramètres des deux configurations SSL sont définis sur aucun.

Vous devez spécifier manuellement le certificat personnel à utiliser dans chaque configuration SSL. Vous pouvez utiliser le même certificat pour les deux configurations. Si vous souhaitez utiliser un certificat différent, suivez les instructions relatives à la «Création d'une demande de certificat», à la page 74 et à la «Réception d'un certificat signé émis par une autorité de certification», à la page 75 pour créer et recevoir un certificat signé supplémentaire, puis répétez ces instructions.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Certificat SSL et gestion des clés**.
3. Sous **Articles liés**, sur la droite, cliquez sur **Configurations SSL**.
4. Cliquez sur le nom de la configuration SSL à utiliser. Par exemple, cliquez sur **NodeDefaultSSLSettings**.
5. Vérifiez que le fichier de clés dans lequel votre certificat est enregistré apparaît dans la zone **Nom de fichier de clés**.
6. Cliquez sur le bouton **Obtenir des alias de certificat** pour vérifier que tous les alias de certificat de votre fichier de clés sont affichés.
7. Dans la zone **Alias de certificat serveur par défaut (Default server certificate alias)**, sélectionnez votre certificat signé.
8. Cliquez sur **Appliquer**.
9. Cliquez sur **Sauvegarder** lorsque vous êtes invité à sauvegarder la configuration dans la configuration principale. La configuration SSL utilise le nouveau certificat.

Que faire ensuite

Répétez la procédure pour associer l'autre configuration SSL au certificat approprié. Poursuivez ensuite avec les instructions de suppression du certificat par défaut, indiquées à la rubrique «Suppression du certificat par défaut», afin d'empêcher tout risque d'utilisation par inadvertance.

Suppression du certificat par défaut

Une fois que vous avez reçu votre certificat personnel signé, supprimez la clé par défaut, afin d'empêcher toute utilisation de celle-ci par inadvertance.

Pourquoi et quand exécuter cette tâche

Avertissement : Avant d'accomplir cette procédure, assurez-vous qu'aucune configuration SSL n'utilise la clé par défaut. Suivez les instructions de la section «Association d'un certificat à la configuration SSL», à la page 76.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Certificat SSL et gestion des clés**.
3. Sous l'option **Articles liés**, cliquez sur**Magasins de clés et certificats**.
4. Cliquez sur **NodeDefaultKeyStore**.
5. Sous Propriétés supplémentaires, cliquez sur **Certificats personnels**.
6. Cochez la case en regard du certificat **default**.

7. Cliquez sur le bouton **Supprimer**.
8. Cliquez sur **Appliquer**, puis sur **Enregistrer**.

Que faire ensuite

Poursuivez avec les instructions de la rubrique «Extraction d'un certificat en vue de le partager avec votre partenaire» pour pouvoir transmettre les données à votre partenaire.

Extraction d'un certificat en vue de le partager avec votre partenaire

Après avoir ajouté un certificat d'autorité de certification signé à votre serveur, exportez une copie de ce certificat accompagnée de sa clé publique et transmettez-la à votre partenaire.

Avant de commencer

Il doit exister un fichier de clés et un certificat personnel.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Certificat SSL et gestion des clés** > **Gérer les configurations de sécurité du noeud final**.
3. Sélectionnez votre noeud dans l'arborescence **Sortant**.
4. Cliquez sur **Gérer les certificats**.
5. Sélectionnez le certificat signé de l'autorité de certification.
6. Cliquez sur **Extraire** dans l'angle supérieur droit.
7. Entrez le chemin d'accès complet de l'emplacement d'extraction du certificat. Incluez le nom de fichier du certificat dans le chemin d'accès. Le certificat de signataire est enregistré dans ce fichier. Sous Windows, par exemple, vous pouvez spécifier : `c:\certificates\local_cert.arm`
8. Sélectionnez le type de données par défaut dans la liste.
9. Cliquez sur **Appliquer**.
10. Cliquez sur **Sauvegarder**. La partie du certificat personnel relative au signataire est stockée dans le fichier .arm spécifié.

Que faire ensuite

Vous voilà prêt à fournir le fichier à votre partenaire, afin que celui-ci puisse ajouter votre certificat à son fichier de clés certifiées.

Remarque : Si votre partenaire utilise Tivoli Federated Identity Manager, il doit importer votre certificat dans son fichier de clés certifiées Tivoli Federated Identity Manager.

Pour achever la configuration SSL, exécutez les étapes de la rubrique «Configuration des exigences relatives à l'authentification client».

Configuration des exigences relatives à l'authentification client

Parmi les options de sécurisation des messages, vous pouvez exiger de votre partenaire qu'il s'authentifie sur votre serveur point de contact.

Pourquoi et quand exécuter cette tâche

Remarque : Dans une fédération SAML 1.x, seul le fournisseur d'identité agit en tant que serveur. Toutefois, seul le partenaire du fournisseur d'identité est tenu de configurer un paramètre d'authentification client.

Vous devez d'abord décider si vous allez demander l'authentification client.

- Si vous ne demandez pas l'authentification client, reportez-vous à la section «Configuration de l'accès sans aucune authentification».
- Si vous demandez l'authentification client, vous disposez des deux options suivantes :
  - Authentification de base. Voir «Configuration de l'accès à l'authentification de base», à la page 80.
  - Authentification par certificat client. Voir «Configuration de l'accès via l'authentification par certificat client», à la page 81.

Configuration de l'accès sans aucune authentification

Si vous ne demandez pas d'authentification client à votre partenaire, configurez les paramètres d'authentification SOAP de façon appropriée.

Pourquoi et quand exécuter cette tâche

Par défaut, après l'installation, les paramètres de sécurité de noeud final ont pour valeur **Autoriser les utilisateurs non authentifiés à accéder aux noeuds finals SOAP**.

Remarque : Ces instructions s'appliquent aux serveurs WebSphere autonomes. Pour les serveurs WebSphere Network Deployment inclus dans un cluster, voir «Configuration du serveur IHS pour le formulaire client», à la page 97.

Pour vous assurer que ce paramètre est sélectionné :

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Point de contact**.
3. Sélectionnez le serveur point de contact utilisé dans votre environnement.
4. Cliquez sur **Avancé**. Le panneau Paramètres de sécurité de noeud final SOAP s'ouvre.
5. Assurez-vous que le port SOAP est correct dans votre configuration et que l'option **Autoriser les utilisateurs non authentifiés à accéder aux noeuds finaux SOAP** est sélectionnée.
6. Cliquez sur **OK**.
7. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**

Que faire ensuite

Si vous configurez une fédération SAML 2.0, poursuivez avec les étapes permettant de configurer votre certificat client à la rubrique «Configuration des certificats client», à la page 83. Si vous configurez une fédération SAML 1.x, la procédure est terminée.

Configuration de l'accès à l'authentification de base

Si vous demandez l'authentification de base à votre partenaire, créez dans votre registre d'utilisateurs un utilisateur représentant votre fournisseur de services partenaire.

Avant de commencer

Avant de démarrer cette tâche, procédez comme suit :

- Choisissez si vous allez autoriser l'accès au noeud final à des utilisateurs authentifiés individuellement ou à des utilisateurs authentifiés faisant partie de groupes spécifiques.
- Veillez à connaître le nom d'utilisateur et le mot de passe que votre fournisseur de services doit utiliser.

Pourquoi et quand exécuter cette tâche

Pour configurer l'authentification de base, effectuez les étapes ci-dessous.

Procédure

1. Dans votre registre d'utilisateurs, créez un utilisateur dont le nom évoque votre partenaire de fournisseur de service. Par exemple, créez un utilisateur dont le nom est `soapclient`.

   Remarque : Reportez-vous aux instructions de création d'utilisateur correspondant au registre d'utilisateurs que vous avez configuré pour votre environnement.
2. L'étape suivante varie selon que vous autorisez des utilisateurs authentifiés individuels ou des utilisateurs authentifiés faisant partie de groupes spécifiques.
   - Si vous demandez l'authentification de base à des utilisateurs individuels, répétez l'étape 1 pour chaque utilisateur de fournisseur de services à configurer. Allez ensuite à l'étape 3.
   - Si vous demandez l'authentification de base à des utilisateurs faisant partie de groupes spécifiques, créez un groupe pour les utilisateurs, puis ajoutez, dans ce groupe, l'utilisateur que vous avez créé à l'étape 1. Par exemple, créez un groupe intitulé `soapgroup`, puis ajoutez l'utilisateur `soapclient` dans ce groupe.

     Remarque : Reportez-vous aux instructions de création de groupe correspondant au registre d'utilisateurs que vous avez configuré pour votre environnement.
3. Configurez les paramètres d'authentification SOAP sur la Tivoli Federated Identity Manager :

   Remarque : Ces instructions s'appliquent aux serveurs WebSphere autonomes. Pour les serveurs WebSphere Network Deployment inclus dans un cluster, voir «Configuration du serveur IHS pour le formulaire client», à la page 97.
   a. Connectez-vous à la console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
   c. Sélectionnez le serveur point de contact utilisé dans votre environnement.
   d. Cliquez sur **Avancé**. Le panneau Paramètres de sécurité de noeud final SOAP s'ouvre.

e. Vérifiez que vous avez correctement configuré le port SOAP, puis sélectionnez l'option correspondant à votre configuration :
   - Pour exiger l'authentification des utilisateurs individuels, sélectionnez **Autoriser les utilisateurs authentifiés à accéder aux noeuds finals SOAP**.
   - Pour exiger l'authentification des utilisateurs faisant partie de groupes spécifiques, sélectionnez, **Autoriser les utilisateurs du groupe spécifié à accéder aux noeuds finaux SOAP**, puis indiquez le nom du groupe dans la zone **Nom de groupe**.

f. Sélectionnez **Authentification de base**.

g. Cliquez sur **OK**.

h. Cliquez sur le bouton **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Que faire ensuite

Si vous configurez une fédération SAML 2.0, poursuivez avec les étapes permettant de configurer votre certificat client à la rubrique «Configuration des certificats client», à la page 83.

Si vous configurez une fédération SAML 1.x, la procédure est terminée.

Configuration de l'accès via l'authentification par certificat client

Si vous demandez l'authentification par certificat client à votre partenaire, vous devez effectuer les étapes tâches présentées dans cette rubrique.

Avant de commencer

1. Configurez la reconnaissance du certificat client par WebSphere Application Server.
2. Créez un utilisateur et éventuellement un groupe représentant le fournisseur de services partenaire.
3. Configurez la demande d'authentification sur Tivoli Federated Identity Manager.

Avant de démarrer cette tâche, procédez comme suit :

- Vérifiez que vous disposez du certificat de clé publique correspondant au certificat client qui permet à votre partenaire d'accéder à votre noeud final de résolution d'artefacts.
- Vérifiez que vous disposez de l'attribut de nom usuel du certificat qui permet à votre partenaire d'accéder à votre noeud final. Par exemple, si le nom distinctif (DN) du certificat est "/C=US/ST=TX/L=AUSTIN/O=SERVICEPROVIDER/CN=soapclient", alors le nom usuel (CN) est "soapclient".
- Indiquez si vous souhaitez autoriser l'accès au noeud final à des utilisateurs authentifiés individuellement ou à des utilisateurs authentifiés faisant partie de groupes spécifiques.

Procédure

1. Copiez le certificat de clé publique que votre partenaire présente pour s'authentifier à votre serveur WebSphere Application Server.

Remarque : Dans ces instructions, le certificat du partenaire est nommé `partnerca.pem`, et le répertoire dans lequel il a été copié `/tmp`.

2. Connectez-vous à la console.
3. Sélectionnez **Sécurité** > **Certificat SSL et gestion des clés**.
4. Sélectionnez **Magasins de clés et certificats**.
5. Sélectionnez **NodeDefaultTrustStore**.
6. Sélectionnez **Certificats de signataires**.
7. Sélectionnez **Ajouter**.
8. Entrez dans les zones les informations pertinentes relatives au certificat. Par exemple :
   - Alias : `CACert`
   - Nom de fichier : `/tmp/partnerca.pem`
   - Type de données : `Base64-encoded`
9. Cliquez sur **OK**.
10. WebSphere doit pouvoir mapper le certificat client présenté par votre partenaire à une identité utilisateur de votre registre d'utilisateurs à l'aide de l'attribut de nom usuel du certificat. Pour afficher cet attribut, cliquez sur le certificat dans la console, puis recherchez la zone **Emis pour**.
    a. Dans votre registre d'utilisateurs, créez un utilisateur dont le nom évoque votre partenaire de fournisseur de service. Par exemple, créez un utilisateur dont le nom est `soapclient`.

       Remarque : Reportez-vous aux instructions de création d'utilisateur correspondant au registre d'utilisateurs que vous avez configuré pour votre environnement.
    b. L'étape suivante varie selon que vous autorisez des utilisateurs authentifiés individuels ou des utilisateurs authentifiés faisant partie de groupes spécifiques.
       - Si vous demandez l'authentification par certificat client à des utilisateurs individuels, répétez l'étape 10a pour chaque utilisateur de fournisseur de services à configurer. Allez ensuite à l'étape 11.
       - Si vous demandez l'authentification par certificat client à des utilisateurs faisant partie de groupes spécifiques, créez un groupe pour les utilisateurs, puis ajoutez, dans ce groupe, l'utilisateur que vous avez créé à l'étape 10a. Par exemple, créez un groupe dont le nom est `soapgroup`, puis ajoutez-y l'utilisateur `soapclient`.

         Remarque : Reportez-vous aux instructions de création de groupe correspondant au registre d'utilisateurs que vous avez configuré pour votre environnement.
         Allez ensuite à l'étape 11.
11. Configurez les paramètres d'authentification SOAP sur la Tivoli Federated Identity Manager :

    Remarque : Ces instructions s'appliquent aux serveurs WebSphere autonomes. Pour les serveurs WebSphere Network Deployment inclus dans un cluster, voir «Configuration du serveur IHS pour le formulaire client», à la page 97.
    a. Connectez-vous à la console.
    b. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
    c. Sélectionnez le serveur point de contact utilisé dans votre environnement.

d. Cliquez sur **Avancé**. Le panneau Paramètres de sécurité de noeud final SOAP s'ouvre.
e. Vérifiez que vous avez correctement configuré le port SOAP, puis sélectionnez l'option correspondant à votre configuration :
   - Pour exiger l'authentification des utilisateurs individuels, sélectionnez **Autoriser les utilisateurs authentifiés à accéder aux noeuds finals SOAP**.
   - Pour exiger l'authentification des utilisateurs faisant partie de groupes spécifiques, sélectionnez, **Autoriser les utilisateurs du groupe spécifié à accéder aux noeuds finaux SOAP**, puis indiquez le nom du groupe dans la zone **Nom de groupe**.
f. Sélectionnez **Authentification par certificat client**.
g. Cliquez sur **OK**.
h. Cliquez sur le bouton **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Que faire ensuite

Si vous configurez une fédération SAML 2.0, poursuivez avec les étapes permettant de configurer votre certificat client à la rubrique «Configuration des certificats client». Si vous configurez une fédération SAML 1.x, la procédure est terminée.

Configuration des certificats client

Si l'authentification par certificat client est également requise par votre partenaire, vous devez créer et importer le certificat que vous présenterez lors de l'authentification. Exportez ensuite le certificat pour votre partenaire.

Réception certificat serveur de votre partenaire

Si l'authentification serveur est configurée chez votre partenaire, vous devez disposer de la clé publique de ce certificat serveur. Enregistrez-la dans un fichier de clés certifiées utilisé par votre service de clés Tivoli Federated Identity Manager.

Avant de commencer

Avant de poursuivre cette procédure, vérifiez qu'un fichier de clés certifiées est disponible pour stocker le certificat. voir la rubrique «Préparation des fichiers de clés», à la page 51.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
   Le panneau Fichiers de clés s'ouvre.
3. Sélectionnez le fichier de clés certifiées dans lequel vous voulez stocker le certificat dans le tableau des fichiers de clés. Le bouton **Afficher les clés** s'active.
4. Cliquez sur **Extraire le certificat de SSL**. Le panneau Mot de passe s'affiche.
5. Entrez le mot de passe du fichier de clés.
6. Cliquez sur **OK**.
7. Remplissez les zones prévues pour le nom d'hôte et le nom de port à partir desquels vous devez récupérer le certificat.

(Facultatif) Cliquez sur **Afficher les informations sur le signataire** pour visualiser le certificat avant de le récupérer.

8. Entrez dans la zone **Alias** le nom à attribuer au certificat.
9. Cliquez sur **OK**. Le certificat est ajouté au fichier de clés certifiées.

Que faire ensuite

Si vous avez le rôle de client dans une connexion SSL et que votre partenaire exige votre authentification à l'aide d'un certificat client, passez à l'étape «Obtention de votre certificat client».

Obtention de votre certificat client

Si vous avez le rôle de client dans une connexion SSL et que votre partenaire exige votre authentification à l'aide d'un certificat client, récupérez le certificat et configurez-le. Partagez-le ensuite avec votre partenaire.

Avant de commencer

Vérifiez qu'un fichier de clés certifiées est disponible pour stocker le certificat. voir la rubrique «Préparation des fichiers de clés», à la page 51.

Procédure

1. Demandez un certificat de paire de clés publique/privée auprès d'une autorité de certification (CA). Pour ce faire :
   a. Connectez-vous à la console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
      Le panneau Fichiers de clés s'ouvre.
   c. Sélectionnez un fichier de clés dans le tableau Fichier de clés. Le bouton **Afficher les clés** est activé.
   d. Cliquez sur **Afficher les clés**. Le panneau Mot de passe s'affiche.
   e. Entrez le mot de passe du fichier de clés, puis cliquez sur **OK**.
   f. Cliquez sur **Demande de certificat**. Le panneau Créer une demande de certificat s'affiche.
   g. Remplissez les zones du panneau.
   h. Cliquez ensuite sur **OK**. Une paire de clés publique/privée est ajoutée au fichier de clés et un fichier contenant les données BASE64 codées est créé. Le certificat autosigné temporaire est remplacé par le certificat signé par l'autorité de certification.

   Revenez à ces instructions lorsque la CA vous aura notifié que votre certificat signé est prêt.
2. Recevez le certificat signé émis par la CA. Pour ce faire :
   a. Connectez-vous à la console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.
      Le panneau Fichiers de clés s'ouvre.
   c. Sélectionnez le fichier dans lequel la RSC a été générée dans le tableau Fichiers de clés. Le bouton **Afficher les clés** est activé.
   d. Cliquez sur **Afficher les clés**. Le panneau Mot de passe s'affiche.
   e. Entrez le mot de passe du fichier de clés, puis cliquez sur **OK**.
   f. Cliquez sur **Réception de certificat de CA**.
   g. Sélectionnez l'emplacement du certificat que vous avez reçu de la CA.

h. Cliquez sur **OK**. Le certificat autosigné temporaire présent dans le fichier de clés est remplacé par le certificat signé que vous avez reçu.

3. Fournissez la clé publique de ce certificat à votre partenaire. Pour ce faire :
   a. Connectez-vous à la console.
   b. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.

      Le panneau Fichiers de clés s'ouvre.
   c. Sélectionnez le fichier de clés approprié dans le tableau Fichier de clés. Vous êtes invité à entrer le mot de passe du fichier de clés.
   d. Entrez le mot de passe.
   e. Cliquez sur **OK**. Le bouton **Afficher les clés** est activé.
   f. Cliquez sur **Afficher les clés**. Le panneau Clés s'ouvre. Il répertorie les éléments du fichier de clés sélectionné.
   g. Sélectionnez les clés à exporter.
   h. Cliquez sur le bouton **Exporter**. Le panneau Exporter la clé s'affiche.
   i. Sélectionnez le format de la clé à exporter.

      (PEM)
      (Privacy-Enhanced Message) Certificat public

      PKCS#12
      Public Key Cryptography Standard 12 : norme d'échange d'informations personnelles
   j. Vérifiez que la case **Inclure une clé privée** *n'est pas* cochée. Vous devez être la seule personne à détenir votre clé privée.
   k. Cliquez sur **Télécharger la clé**.
   l. Lorsque vous y êtes invité, entrez le nom de fichier de la clé exportée.

      Par exemple : `maclépublique.pem`

      (Facultatif) Cliquez sur **Parcourir** pour rechercher le fichier sur le système de fichiers.
   m. Cliquez sur **Annuler** pour quitter.

Que faire ensuite

Fournissez le certificat à votre partenaire. Ce dernier doit vérifier ce qui suit :

- Il a, dans son fichier de clés certifiées, le certificat de CA de l'autorité de certification qui a émis votre certificat.
- Le serveur peut accéder à la liste de révocation de certificat de la CA.

Chapitre 10. Sélection d'un serveur point de contact

Le serveur point de contact est un proxy ou une application qui interagit avec un utilisateur et gère à la fois l'authentification et les sessions. Dans un déploiement classique, le point de contact est situé au bord d'un réseau protégé et devant un pare-feu, comme dans une zone démilitarisée, par exemple.

Tivoli Federated Identity Manager n'est pas directement impliqué dans l'authentification d'utilisateurs ni dans la création d'une session d'application. Au lieu de cela, Tivoli Federated Identity Manager s'appuie sur un *serveur point de contact*.

Le serveur point de contact fournir des noeuds finals qui correspondent aux emplacements vers lesquels et à partir desquels les messages sont envoyés et reçus. Chaque noeud final possède une adresse URL, de sorte que les noeuds finals puissent être accessibles aux utilisateurs externes sous forme de sites Web sur Internet. Le point de contact reçoit les requêtes d'accès et fournit le service d'authentification.

Il représente le premier composant capable d'évaluer les données d'identification de l'utilisateur demandant l'accès au réseau protégé. En outre, il gère le cycle de vie des sessions de l'utilisateur, depuis leur création, l'accès et leur suppression (par exemple, en réponse aux services de fermeture de session).

Le type de serveur point de contact à utiliser est déterminé par les besoins en termes d'architecture de sécurité et de topologie de réseau. Tivoli Federated Identity Manager prend en charge les quatre options suivantes pour le serveur point de contact :

- IBM WebSphere Application Server
- Tivoli Access Manager WebSEAL
- WebSEAL, aucun ACLD
- Serveur point de contact générique
- Serveur point de contact personnalisé

WebSphere en tant que serveur point de contact

Si vous devez utiliser IBM WebSphere Application Server, vos options de configuration varient selon si vous êtes partenaire du fournisseur d'identité ou du fournisseur de services.

Options de fournisseur d'identité

Lorsqu'IBM WebSphere Application Server est utilisé en tant que serveur point de contact et que vous êtes le fournisseur d'identité dans une fédération, vous disposez des options suivantes en ce qui concerne le type d'authentification à utiliser :

- Authentification par formulaires, utilisant n'importe quel registre d'utilisateurs pris en charge
- Mécanisme SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) utilisant l'authentification TAI (Trust Association Interceptor) et le registre d'utilisateurs Microsoft Active Directory

Options de fournisseur de services

Lorsqu'IBM WebSphere Application Server est utilisé en tant que serveur point de contact et que vous êtes le fournisseur de services dans une fédération, la connexion unique est soumise à l'authentification STPA (Lightweight Third-Party Authentication).

Vous pouvez utiliser les options d'application d'hébergement suivantes dans une fédération qui est configurée dans Tivoli Federated Identity Manager :

- IBM WebSphere Application Server, soit le même serveur sur lequel Tivoli Federated Identity Manager est installé, soit un serveur séparé exécutant WebSphere Application Server version 5.1 ou 6.x
- Microsoft Internet Information Services version 6.0 avec plug-in du serveur Web Tivoli Federated Identity Manager installé
- IBM HTTP Server 6.1 avec plug-in du serveur Web Tivoli Federated Identity Manager installé
- Serveur Apache HTTP 2.0 ou 2.2 avec plug-in du serveur Web Tivoli Federated Identity Manager installé

Chacune de ces options est soumise à des exigences spécifiques. Pour plus d'informations sur ces configurations, voir «WebSphere en tant que point de contact pour les fournisseurs d'identité», à la page 99, et «Serveur point de contact WebSphere pour un fournisseur de services», à la page 117.

WebSEAL en tant que serveur point de contact

Pour répondre aux exigences fonctionnelles d'un serveur point de contact, Tivoli Federated Identity Manager peut tire parti des fonctions d'authentification et d'autorisation étendues de Tivoli Access Manager. Dans les environnements exploitant Tivoli Access Manager, un serveur WebSEAL fait généralement office de point de contact.

WebSEAL est généralement utilisé comme proxy inverse permettant de contrôler l'accès à des ressources protégées extensives via l'établissement et la gestion de jonctions WebSEAL. WebSEAL reçoit les requêtes d'accès et représente le premier composant capable d'évaluer les données d'identification de l'utilisateur demandant l'accès au réseau protégé. Il doit en outre gérer les sessions Web pour les utilisateurs.

L'assistant de création de fédération nécessite la spécification d'une URL pour les serveurs point de contact. L'assistant affiche une zone dans laquelle vous devez entrer l'URL donnant accès aux noeuds finals sur le serveur point de contact. L'URL doit contenir les éléments suivants :

- Un protocole de communication. Le protocole HTTPS ou HTTP être utilisé pour les communications entre le serveur point de contact et l'utilisateur. Utilisez HTTPS pour une sécurité optimale.

  Il est à noter que cette valeur doit correspondre à la façon dont vous avez configuré votre serveur point de contact (WebSEAL).

  Par exemple :

  ```
  https://
  ```

- L'adresse de domaine du serveur WebSEAL :

  Par exemple :

  ```
  idp.exemple.com
  ```

- Lors de l'utilisation de WebSEAL, l'élément suivant est le nom de la jonction WebSEAL par laquelle passent les requêtes de services de connexion unique. N'importe quelle valeur est admise, mais elle doit correspondre au nom d'une jonction sur le serveur WebSEAL.

  Par exemple :

  `/FIM`
- Le dernier élément est la chaîne `/sps`. L'élément de l'URL est défini par Tivoli Federated Identity Manager et désigne un contexte WebSphere destiné aux services de connexion unique. La valeur de cette chaîne n'est pas modifiable.

Ces éléments sont combinés pour former une adresse URL. Par exemple :

`https://idp.exemple.com/FIM/sps`

Ultérieurement, lors de la configuration de fédération, l'URL est étendue lorsque vous sélectionnez un protocole de connexion unique (SAML, WS-Federation ou Liberty) et affectez des noeuds finals spécifiques pour des profils, par exemple pour la connexion et la déconnexion. Par conséquent, elle fait désormais partie de plusieurs chemins d'URL plus longs (noeuds finals) gérés comme objets Tivoli Access Manager protégés.

WebSEAL No ACLD en tant que serveur point de contact

Les déploiements Tivoli Access Manager incluent souvent un serveur de règles (pdmgrd) et un serveur d'autorisation (acld). Tivoli Access Manager requiert un serveur de règles déployé, mais ne requiert pas de serveur d'autorisation actif. Tivoli Federated Identity Manager requiert également un seul serveur de règles déployé. Le serveur point de contact WebSEAL ne dépend pas du serveur d'autorisation pour les services d'autorisation ou d'authentification.

Par défaut, l'instance de module IVCred par défaut du produit contacte le serveur d'autorisation Tivoli Access Manager (également appelé pdacld) pour émettre des données d'identification. Des données d'identification modèles sont alors créées à partir du nom d'utilisateur. Ces données incluent les groupes (et ID utilisateur universels) pour cet utilisateur tel que défini dans le registre utilisateur pour Tivoli Access Manager. Toutefois, lorsque vous sélectionnez WebSEAL No ACLD en point de contact, le produit n'utilise pas le serveur d'autorisation pour créer les données d'identification.

Pour configurer le profil de point de contact " WebSEAL No ACLD" :

1. Connectez-vous à la console.
2. Sélectionnez **Tivoli Federation Identity Manager** > **Configurer le service d'accréditation** > **Instances de module**.
3. Sélectionnez **Jeton IVCred par défaut** et cliquez sur **Propriétés**.
4. Décochez **Activer l'émission de droits d'accès Access Manager (IVCred) (nécessite la configuration de PDJRTE)**.
5. Cliquez sur **OK**.

Remarque : Si vous remplacez le point de contact par un serveur WebSEAL comprenant un serveur d'autorisation, sélectionnez **Activer l'émission de droits d'accès Access Manager (IVCred) (nécessite la configuration de PDJRTE)**.

Serveur point de contact générique

Le serveur point de contact générique est une implémentation de point de contact supplémentaire fournie par Tivoli Federated Identity Manager. Il s'agit d'une solution reposant sur des en-têtes HTTP qui offre aux administrateurs la possibilité de modifier leurs environnements de point de contact (par exemple, Apache) de manière à pouvoir définir et lire les en-têtes. ceci permet l'intégration à Tivoli Federated Identity Manager sans nécessiter la création d'un serveur point de contact personnalisé.

Le serveur point de contact générique fonctionne de manière très similaire au point de contact WebSEAL. La principale différence réside dans le fait que les noms des en-têtes sont utilisés pour définir les informations utilisateur.

Le serveur point de contact générique est inclus dans les profils de point de contact fournis avec Tivoli Federated Identity Manager. L'administrateur doit procéder à son activation en le sélectionnant sur la console et en la configurant comme étant active. L'administrateur peut, à l'aide de la console, modifier les noms des en-têtes utilisés par chaque rappel.

Serveur point de contact personnalisé

Un serveur point de contact personnalisé est constitué de plusieurs modules de rappel personnalisés d'ouverture qui définissent les paramètres d'ouverture de session, de fermeture de session, d'ID local et d'authentification. Un serveur point de contact personnalisé peut constituer le choix approprié pour votre environnement si vous souhaitez intégrer une application d'authentification ou de gestion d'accès Web existante à Tivoli Federated Identity Manager.

Un serveur point de contact personnalisé peut s'avérer utile dans les situations suivantes :

- Si vous disposez d'un cookie de connexion unique existant, qui est utilisé sur l'ensemble de l'entreprise, vous pouvez mettre en oeuvre un serveur point de contact personnalisé utilisant un rappel SignIn qui définit le cookie du domaine de connexion unique conformément à votre stratégie de connexion unique.
- Si vous disposez d'un logiciel de gestion d'accès Web qui expose une interface API personnalisée en vue de certifier l'identité d'un utilisateur dans l'environnement, ou d'extraire l'utilisateur actuel pour les besoins de la requête.

  Vous pouvez choisir l'une des implémentations de serveur point de contact :
  - Un serveur point de contact qui exécute un rappel d'identité local pour extraire l'utilisateur lié à la transaction.
  - Un serveur point de contact personnalisé qui utilise un rappel SignIn pour certifier l'identité d'un utilisateur dans l'environnement.
  - Un serveur point de contact qui exploite ces deux types de rappel.

La mise au point d'un serveur point de contact personnalisé nécessite une certaine expérience dans la programmation de modules de rappel, ainsi qu'une bonne connaissance des concepts de programmation de Tivoli Federated Identity Manager. Voir les liens des documents developerWorks dans le centre de documentation à l'adresse http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc_6.2.2/ic/ic-homepage.html.

Une fois le travail de développement terminé, intégrez la solution à votre environnement Tivoli Federated Identity Manager. Pour plus d'informations, reportez-vous au document *IBM Federated Identity Manager - Guide d'administration*.

Chapitre 11. Configuration de WebSphere en tant que serveur point de contact

Tivoli Federated Identity Manager peut être installé soit sur un serveur WebSphere intégré, soit dans un environnement WebSphere existant. Lors de l'installation du serveur intégré et lorsque WebSphere est utilisé en tant que serveur point de contact, le programme d'installation automatise une grande partie de la configuration. Lorsque vous effectuez l'installation dans un environnement WebSphere existant et que vous souhaitez utiliser WebSphere en tant que serveur point de contact, vous devez configurer manuellement les serveurs WebSphere et IHS afin qu'ils répondent aux spécifications de votre déploiement.

Lorsque WebSphere est configuré en tant que serveur point de contact, des services d'authentification sont disponibles. Les services d'authentification sont propres au rôle défini au sein de la fédération (fournisseur d'identité ou fournisseur de services).

Remarque : Pour WebSphere Application Server Version 6.0.2, WebSphere n'est pas pris en charge comme point de contact par Tivoli Federated Identity Manager.

Configurez un proxy HTTP sortant dans WebSphere Application Server en vue de son utilisation lorsque Tivoli Federated Identity Manager établit des connexions à d'autres serveurs HTTP.

Pour plus d'informations, consultez les rubriques suivantes :

- «Utilisation d'IBM HTTP Server avec WebSphere configuré en tant que point de contact»
- «WebSphere en tant que point de contact pour les fournisseurs d'identité», à la page 99
- «Serveur point de contact WebSphere pour un fournisseur de services», à la page 117
- «Configuration d'un serveur proxy HTTP sortant», à la page 98

Utilisation d'IBM HTTP Server avec WebSphere configuré en tant que point de contact

WebSphere Application Server Network Deployment (ND) peut être déployé soit en mode autonome, soit en tant que membre d'un cluster WebSphere. Dans un cas comme dans l'autre, un environnement de déploiement typique comprend une instance IBM HTTP Server (IHS) positionnée entre le serveur WebSphere et des connexions externes, telles que celles qui proviennent d'un pare-feu ou d'une zone démilitarisée (DMZ).

Le déploiement du serveur IHS repose généralement sur la configuration de connexions SSL (Secure Socket Layer) permettant de sécuriser à la fois les connexions externes et les liaisons établies en interne avec les serveurs WebSphere. Le succès du déploiement d'un environnement Tivoli Federated Identity Manager qui utilise WebSphere comme serveur point de contact nécessite l'activation de SSL sur le serveur IHS.

L'activation de SSL sur IHS nécessite la génération d'une base de données de clés SSL et d'une clé. Vous pouvez générer les clés nécessaires au moyen de l'utilitaire ikeyman. Si vous n'avez pas activé SSL sur le serveur IHS, vous devez accomplir cette tâche avant d'effectuer la configuration de Tivoli Federated Identity Manager.

Pour obtenir les instructions correspondantes, consultez le centre de documentation de votre instance IBM HTTP Server pour WebSphere Application Server : http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Consultez les rubriques relatives à la sécurité d'IBM HTTP Server, notamment :

- Maniement de la base de données de clés
- Sécurisation au moyen de communications SSL

Ajout d'un port SSL à un canal de retour SOAP

Les fédérations à connexion unique Tivoli Federated Identity Manager prennent en charge la configuration de l'authentification par certificat ou l'authentification de base entre les partenaires de la fédération. Lorsque l'environnement de déploiement inclut un serveur IHS, vous devez configurer un canal de retour SOAP afin de permettre la prise en charge de ces méthodes d'authentification.

Vous devez ajouter un hôte virtuel à la configuration IHS. Les paramètres de configuration sont généralement présents dans le fichier de configuration IHS standard. Par exemple, sous Linux ou UNIX:

```
/opt/IBM/HTTPServer/httpd.conf
```

Pour obtenir des instructions, consultez le centre de documentation de votre serveurIBM HTTP Server pour WebSphere Application Serverhttp://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Consultez les rubriques relatives à la sécurité d'IBM HTTP Server, notamment :

- Sécurisation au moyen de communications SSL

Mise à jour de la configuration de la fédération pour les connexions SOAP

Lorsque le serveur IBM HTTP Server est configuré pour écouter à la fois le port par défaut et le port de canal de retour SOAP, vous devez définir et configurer vos fédérations en utilisant ces ports pour les URL de la fédération.

Les URL de fédération doivent utiliser le port 443. Comme il s'agit du port HTTPS par défaut, il n'est pas nécessaire d'inclure le numéro de port réel dans la syntaxe de l'URL. Le port de retour SOAP est généralement le port 9444.

Du fait que la sécurité du canal de retour SOAP implique une connexion avec le serveur IHS, la procédure de configuration typique appliquée lors de la définition d'une fédération ne nécessite aucune authentification client sur le canal de retour SOAP.

Remarque : L'environnement WebSphere peut inclure la configuration de SSL entre IHS et les noeuds du cluster WebSphere. Si cette configuration est appropriée pour votre déploiement, reportez-vous à la documentation de WebSphere.

Confirmation des propriétés de sécurité de WebSphere Application Server

Si vous avez installé la version intégrée de WebSphere Application Server avec l'installation du composant de services d'exécution et de gestion, plusieurs de ses paramètres ont été configurés lors de l'installation. Si vous utilisez une version existante de WebSphere Application Server (par exemple, une version installée précédemment, ou la version installable séparément, vous devez configurer ces paramètres manuellement.

Avant de commencer

Les paramètres sont les suivants :

- La sécurité des applications et la sécurité d'administration sont activées.
- La connexion unique (cookie LTPA) est activée.

Utilisez les procédures ci-dessous pour confirmer que les paramètres de configuration sont corrects pour votre environnement Tivoli Federated Identity Manager.

Utilisez la console de gestion de WebSphere pour contrôler les paramètres de WebSphere.

Pourquoi et quand exécuter cette tâche

La sécurité des applications et la sécurité d'administration sont activées

Pour confirmer que la sécurité des applications et la sécurité d'administration sont activées, procédez comme suit :

1. Cliquez sur **Sécurité** > **Administration, applications et infrastructure sécurisées**.
2. Confirmez que la sécurité des applications et la sécurité d'administration sont activées.

La connexion unique est activée

Pour confirmer que la connexion unique est activée, procédez comme suit :

1. Cliquez sur **Sécurité** > **Administration, applications et infrastructure sécurisées**.
2. Développez Sécurité Web à droite pour afficher les options suivantes :
   - **Paramètres généraux**
   - **connexion unique**
   - **Relation de confiance**
3. Cliquez sur **connexion unique**.
4. Assurez-vous que l'option **Activé(e)** est sélectionnée.
5. Sélectionnez **Sécurité** > **Administration, applications et infrastructure sécurisées> Sécurité Web - Paramètres généraux**.
6. Sous l'onglet Configuration, dans la section Propriétés générales, cochez la case **Utiliser les données d'authentification disponibles lors de l'accès à un URI non protégé**.

Activation de codage multilingue sur WebSphere Application Server

Activez le codage multilingue en activant le codage client UTF-8 dans WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

La procédure pour activer le codage multilingue est identique à celle de la version intégrée de WebSphere Application Server et de la version de WebSphere Application Server existante.

Procédure

1. Ouvrez une invite de commande.
2. Démarrez l'outil 'wsadmin' de WebSphere Application Server. A partir de votre profil WebSphere, entrez la commande appropriée afin que votre système d'exploitation démarre l'outil :

   Windows

   ```
   wsadmin.bat
   ```

   AIX, Linux ou Solaris

   ```
   wsadmin.sh
   ```

   Remarque : Pour plus d'informations sur les options pouvant être définies à l'aide de l'outil wsadmin, reportez-vous au centre d'information en ligne de WebSphere Application Server à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp.
3. A l'invite de commande, exécutez les commandes suivantes pour activer le codage UTF-8 :
   a. Pour afficher les propriétés JVM en cours :

      ```
      $AdminTask showJVMProperties { -propertyName genericJvmArguments }
      ```

   b. Pour définir les propriétés JVM :

      ```
      $AdminTask setGenericJVMArguments { -genericJvmArguments
      "<current-jvm-properties>  -Dclient.encoding.override=UTF-8" }
      ```

   c. Pour sauvegarder les modifications de configuration :

      ```
      $AdminConfig save
      ```
4. Redémarrez WebSphere Application Server.

Mappage de rôles d'application avec des utilisateurs

Vous pouvez mapper différents rôles d'application avec les utilisateurs de IBM Tivoli Federated Identity Manager.

Avant de commencer

Lorsqu'IBM Tivoli Federated Identity Manager est déployé avec la version intégrée de WebSphere, l'installation d'IBM Tivoli Federated Identity Manager mappe automatiquement les rôles d'application avec des utilisateurs. Lorsqu'IBM Tivoli Federated Identity Manager est déployé avec une instance existante du serveur WebSphere, IBM Tivoli Federated Identity Manager, vous devez procéder à la création des mappages manuellement.

Vous pouvez définir les différents rôles d'après les besoins de votre déploiement en matière de sécurité.

Pourquoi et quand exécuter cette tâche

Utilisez la console d'administration WebSphere pour spécifier les mappages.

Procédure

1. Sélectionnez **Enterprise Applications > ITFIMRuntime > Rôle de sécurité pour le mappage utilisateur/groupe**.
2. Sélectionnez les mappages dans la table des rôles.

   Pour chaque rôle, sélectionnez soit **Tous les utilisateurs** ou **Tous les utilisateurs authentifiés**.

   Remarque : FIMAnyAuthenticated *ne doit pas* être mappé avec **Tous les utilisateurs**.

   Exemples de rôles :
   - TrustClientRole
   - FIMUnauthenticated
   - FIMSoapClient
   - FIMAnyAuthenticated
   - FIMAdministrator
   - TrustClientInternalRole
   - FIMNobody
3. Cliquez sur **OK** quand vous avez terminé.
4. Synchronisez tous les noeuds du cluster.

   Pour obtenir des instructions, consultez le centre de documentation de WebSphere : http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Reportez-vous à la rubrique *Mappage d'utilisateur avec des rôles*.

Résultats

Le composant d'exécution de Tivoli Federated Identity Manager fonctionne maintenant avec WebSphere en tant que serveur point de contact dans un environnement WebSphere Network Deployment (ND).

Configuration du serveur IHS pour le formulaire client

Lors de la configuration de partenaires pour une fédération de connexion unique, vous pouvez spécifier les méthodes prises en charge pour l'authentification client. L'assistant à interface graphique du partenaire de fédération vous invite à spécifier soit l'authentification par certificat SSL, soit l'authentification de base. Suivant votre choix, vous devez configurer IBM HTTP Server (IHS) en conséquence.

Suivez les instructions contenues dans la section suivante en fonction de votre méthode d'authentification.

Configuration de l'authentification par certificat pour IHS

Lorsque la configuration du partenaire de fédération implique la gestion d'un certificat client SSL sur une connexion SOAP, vous devez importer ce certificat en tant que certificat d'autorité de certification (CA) dans la base de données de clés utilisée par IHS sur SSL.

A titre d'exemple, voici une base de données de clés sur Linux ou UNIX :

```
/usr/IBM/HTTPServer/conf/httpkeys.kdb
```

Exécutez l'utilitaire **ikeyman** pour importer le certificat.

Pour obtenir les instructions correspondantes, consultez le centre de documentation de votre instance IBM HTTP Server pour WebSphere Application Server : http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Consultez les rubriques relatives à la sécurité d'IBM HTTP Server, notamment :

- Stockage d'un certificat pour l'autorité de certification

Configuration de l'authentification de base pour IHS

Lorsque la configuration du partenaire de fédération implique une authentification de base sur une connexion SOAP, vous devez activer l'authentification LDAP pour IHS.

Pour obtenir les instructions correspondantes, consultez le centre de documentation de votre instance IBM HTTP Server pour WebSphere Application Server : http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Consultez les rubriques relatives à la sécurité d'IBM HTTP Server, notamment :

- Authentification LDAP sur IBM HTTP Server

Configuration d'un serveur proxy HTTP sortant

Configurez un proxy HTTP sortant dans WebSphere Application Server en vue de son utilisation lorsque Tivoli Federated Identity Manager établit des connexions à d'autres serveurs HTTP.

Pourquoi et quand exécuter cette tâche

La configuration doit être définie sur les noeuds d'exécution de Tivoli Federated Identity Manager. Les paramètres de configuration de la machine virtuelle Java qui permettent les connexions HTTP sortantes via un proxy s'appliquent à toutes les connexions sortantes pour toutes les applications, y compris Tivoli Federated Identity Manager, qui s'exécutent dans WebSphere Application Server.

La procédure ci-après s'applique à WebSphere Application Server 7.0. Les étapes sont les mêmes que dans d'autres versions de WebSphere Application Server.

Procédure

1. Connectez-vous à la console **Integrated Solutions Console**.
2. Sélectionnez **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere**.
3. Cliquez sur le serveur approprié. Exemple : **server1**.
4. Sous **Infrastructure du serveur**, cliquez sur **Gestion des processus et Java** > **Définition des processus**.
5. Sous **Propriétés supplémentaires**, cliquez sur **Machine virtuelle Java**.
6. Cliquez sur l'onglet **Configuration**.
7. Sous **Propriétés supplémentaires**, cliquez sur **Propriétés personnalisées**.
8. Cliquez sur **Nouveau**.
9. Spécifiez le **nom** et la **valeur** appropriés en fonction de la configuration requise. La configuration requise diffère selon que vous utilisez le protocole HTTP ou le protocole HTTPS. Voir le tableau 1 pour plus de détails.

10. Cliquez sur **OK**.
11. Répétez les étapes 8 à 10 pour chaque configuration requise.
12. Cliquez sur **Sauvegarder directement dans la configuration principale**.
13. Redémarrez **WebSphere Application Server**.

Remarque : Répétez les étapes 2 à 13 pour chaque serveur. Cette configuration est conçue pour la machine virtuelle Java de WebSphere Application Server. Par conséquent, elle doit exister pour chaque noeud d'exécution Tivoli Federated Identity Manager et chaque instance WebSphere Application Server.

Tableau 8. Liste de tous les noms et de toutes les valeurs possibles pour un serveur proxy HTTP sortant

Nom	Exemples de valeur	Description
`http.proxyHost`	`http.proxy.ibm.com`	Nom d'hôte ou adresse IP du proxy HTTP
`http.proxyPort`	`3128`	Port du proxy HTTP
`http.proxyUser`	`admin`	Nom d'utilisateur indiqué pour l'authentification auprès du proxy pour les connexions HTTP
`http.proxyPassword`	`mot de passe`	Mot de passe utilisé pour l'authentification auprès du proxy pour les connexions HTTP
`https.proxyUser`	`admin`	Nom d'utilisateur indiqué pour l'authentification auprès du proxy pour les connexions HTTPS
`https.proxyPassword`	`mot de passe`	Mot de passe utilisé pour l'authentification auprès du proxy pour les connexions HTTPS
`https.proxyHost`	`https.proxy.ibm.com`	Nom d'hôte ou adresse IP du proxy HTTPS
`https.proxyPort`	`3128`	Port du proxy HTTPS
`http.nonProxyHosts`	`host1.ibm.com|internal.ibm.com`	Liste d'hôtes séparés par une barre verticale (`|`) qu'un proxy ne doit pas utiliser

Remarque : La propriété `http.nonProxyHosts` s'applique aux connexions HTTP et aux connexions HTTPS.

WebSphere en tant que point de contact pour les fournisseurs d'identité

Si vous êtes le fournisseur d'identité dans votre fédération et que vous utilisez IBM WebSphere Application Server en tant que serveur point de contact, vous disposez de deux options en ce qui concerne la méthode d'authentification. Le choix de la méthode d'authentification détermine la configuration requise dans votre environnement.

Choisissez une des options suivantes pour la méthode d'authentification sur votre WebSphere Application Server :

- Authentification par formulaires, à l'aide de tout registre d'utilisateurs pris en charge par WebSphere Application Server
- Authentification du bureau Windows via la prise en charge de SPNEGO TAI sur WebSphere Application Server 8.0 et l'utilisation de Microsoft Active Directory en tant que registre d'utilisateurs

Avertissement : Avant d'accomplir les tâches décrites dans le présent chapitre, assurez-vous que les paramètres sont corrects en appliquant la procédure «Confirmation des propriétés de sécurité de WebSphere Application Server», à la page 95.

authentification par formulaires

Dans cette configuration, le fournisseur d'identité utilise n'importe quel registre d'utilisateurs pris en charge par WebSphere Application Server doté de la fonction d'authentification par formulaires, afin d'authentifier les utilisateurs qui demandent la connexion unique. Tous les utilisateurs du fournisseur d'identité doivent exister dans le registre d'utilisateurs pris en charge. Lorsque des utilisateurs tentent de faire appel à la connexion unique pour accéder à une ressource (telle qu'une application Web), Tivoli Federated Identity Manager présente un formulaire de connexion. Ce dernier est fourni avec Tivoli Federated Identity Manager.

Un utilisateur non authentifié qui émet une demande de connexion unique sur une ressource de fournisseur de services est authentifié sur le registre d'utilisateurs configuré pour WebSphere Application Server.

Un exemple de cette configuration est présenté dans la figure 1, à la page 101.

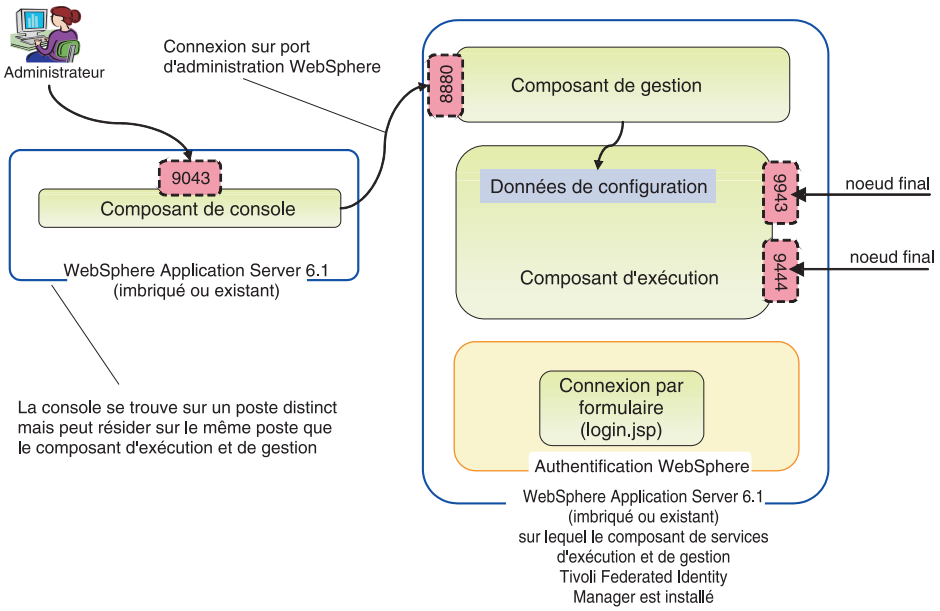


Figure 1. Exemple de WebSphere Application Server doté de la fonction d'authentification par formulaires

Remarques sur la configuration :

- L'instance de WebSphere Application Server peut être soit un déploiement existant de WebSphere (avec le niveau approprié de groupes de correctifs appliqués) soit la version intégrée de WebSphere Application Server version 8.0 distribuée avec Tivoli Federated Identity Manager.
- Un formulaire de connexion présenté par le serveur WebSphere Application Server sur lequelTivoli Federated Identity Manager est installé. Le formulaire de connexion est fourni.

Suivez les tâches indiquées à la section «Configuration de l'authentification par formulaires», à la page 103.

Authentification à partir du bureau Windows via le support SPNEGO TAI à l'aide de Microsoft Active Directory

Cette configuration utilise un support TAI (Trust Association Interceptor) WebSphere qui prend en charge une authentification en mode silencieux à l'aide du protocole SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism), qui est fourni avec WebSphere Application Server. Cette configuration permet à Tivoli Federated Identity Manager d'acquérir en toute sécurité l'identité du bureau de l'utilisateur, qui sert ensuite à créer l'assertion de la connexion unique fédérée.

Le fournisseur d'identité utilise Microsoft Active Directory en tant que registre d'utilisateurs, ainsi que l'authentification de domaine Microsoft Windows. Windows doit être configuré en tant que contrôleur de domaine. Tous les utilisateurs du fournisseur d'identité doivent exister dans le registre d'utilisateurs Active Directory. Pour établir une connexion unique à une application Web, les utilisateurs ont recours à leurs données d'identification sur le bureau Windows.

Un exemple de cette configuration est présenté dans la figure 2.

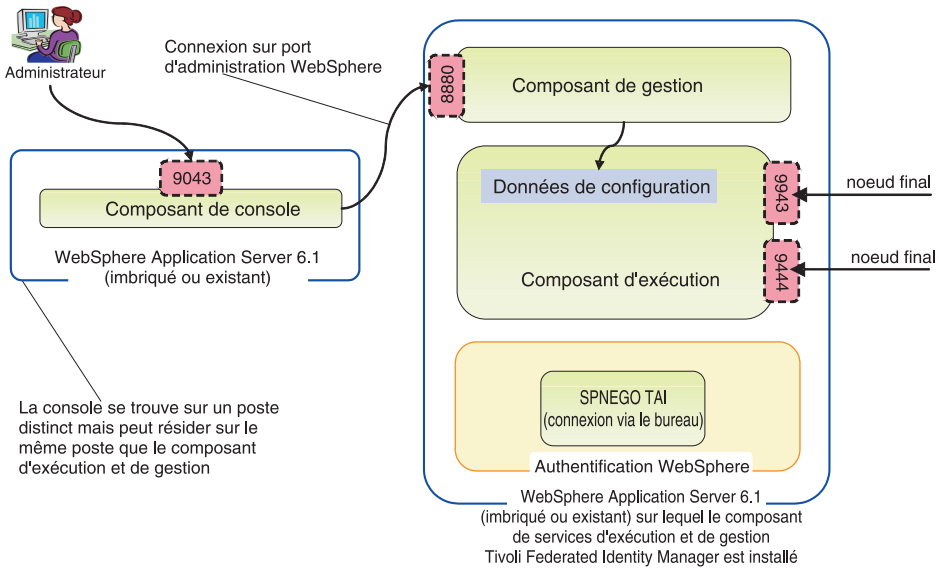


Figure 2. Exemple de WebSphere Application Server doté de l'authentification SPNEGO TAI

Remarques sur la configuration :

- L'instance de WebSphere Application Server peut être soit un déploiement existant de WebSphere (avec le niveau approprié de groupes de correctifs appliqués) soit la version intégrée de WebSphere Application Server version 8.0 distribuée avec Tivoli Federated Identity Manager.
- Microsoft Active Directory doit être utilisé en tant que registre d'utilisateurs. Utilisez une version prise en charge par Microsoft Windows Server 2003. Le registre d'utilisateurs doit inclure un utilisateur d'administration WebSphere et un utilisateur pour l'identité Kerberos. En outre, un fichier de clés doit être généré pour chaque utilisateur. Vous avez besoin des propriétés de connexion LDAP pour le serveur Active Directory avant de configurer Tivoli Federated Identity Manager.

  Le registre d'utilisateurs doit également être configuré avant IBM WebSphere Application Server.
- L'authentification SPNEGO est fournie avec WebSphere Application Server via le module d'extension TAI (Trust Association Interceptor). Elle utilise Kerberos pour effectuer l'authentification.
- Les utilisateurs se connectent au domaine Windows à l'aide de leur connexion bureau. Cette méthode de connexion peut également être appelée *connexion unique via le bureau*.
- Les navigateurs de vos utilisateurs doivent être configurés de sorte que l'authentification Windows intégrée soit activée.

Suivez les tâches indiquées à la section «Configuration de l'authentification SPNEGO», à la page 106.

Configuration de l'authentification par formulaires

Si vous utilisez WebSphere Application Server en tant que serveur point de contact doté de la fonction d'authentification par formulaire, vous devez effectuer plusieurs tâches de configuration.

Pourquoi et quand exécuter cette tâche

Les tâches sont les suivantes :

1. «Sélection et installation du registre d'utilisateurs»
2. «Configuration du registre d'utilisateurs», à la page 104
3. «Ajout d'utilisateurs de connexion unique», à la page 104
4. «Ajout d'utilisateurs d'administration», à la page 104
5. «Configuration du registre d'utilisateurs pour embeddedWebSphere», à la page 105
6. «Configuration d'une connexion SSL au registre d'utilisateurs», à la page 105
7. «Personnalisation du formulaire de connexion», à la page 106

Sélection et installation du registre d'utilisateurs

Un registre d'utilisateurs est requis dans votre environnement de fournisseur d'identité. Le registre d'utilisateurs sert de référentiel pour les informations relatives aux utilisateurs auxquels vous fournissez des fonctionnalités de connexion unique et aux fournisseurs de services avec lesquels vous partagez une fédération. Il peut également servir de référentiel pour les informations relatives aux utilisateurs d'administration de votre environnement ou vous pouvez choisir de conserver les utilisateurs d'administration dans un registre d'utilisateurs distinct.

Avant de commencer

Vous pouvez choisir un registre d'utilisateurs compatible avec votre serveur point de contact IBM WebSphere Application Server et avec la méthode d'authentification utilisée.

Si vous utilisez l'authentification par formulaires, vous pouvez choisir un registre d'utilisateurs à partir de nombreuses options. Reportez-vous au centre de documentation de WebSphere Application Server 8.0 à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Recherchez ensuite des informations sur le choix d'un registre d'utilisateurs, en sélectionnant **WebSphere Application Server (Distributed platforms and Windows)** > **Securing applications and their environment** > **Authenticating users** > **Selecting a registry or repository**.

Pourquoi et quand exécuter cette tâche

Si vous utilisez une installation existante de WebSphere Application Server, il se peut qu'un registre d'utilisateurs compatible soit déjà installé et configuré.

Si vous utilisez une nouvelle installation de la version imbriquée de WebSphere Application Server, vous disposez des options suivantes :

- Utiliser le domaine de référentiel d'utilisateurs basé sur un fichier, qui a été installé avec la version imbriquée de WebSphere Application Server. L'utilisateur d'administration a été configuré dans ce registre pendant l'installation. Les autres tâches requises pour l'ajout des utilisateurs de connexion unique sont indiquées ultérieurement dans ce chapitre.

- Utiliser un registre d'utilisateurs différent. Pour plus d'informations sur vos options de registre d'utilisateurs, consultez la documentation de WebSphere Application Server. Ensuite, installez et configurez le registre d'utilisateurs choisi, si vous n'utilisez pas un registre d'utilisateurs déjà existant. Configurez ensuite WebSphere pour qu'il utilise ce registre d'utilisateurs. Voir la rubrique «Configuration du registre d'utilisateurs pour embeddedWebSphere», à la page 105.

Configuration du registre d'utilisateurs

La configuration du registre d'utilisateurs est une étape importante de la configuration globale.

Avant de commencer

Vous devez d'abord avoir sélectionné et installé le registre d'utilisateurs requis, comme expliqué dans la rubrique «Sélection et installation du registre d'utilisateurs», à la page 103.

Pourquoi et quand exécuter cette tâche

Dans ce registre, créez des utilisateurs auxquels fournir des fonctionnalités de connexion unique. Vous pouvez également créer des utilisateurs pour les administrateurs de votre environnement ou choisir de conserver les utilisateurs d'administration dans un référentiel distinct.

Ajout d'utilisateurs de connexion unique :

Dans l'environnement du fournisseur d'identité, le registre d'utilisateurs sert à authentifier les utilisateurs qui ont recours à la connexion unique. Ajoutez ces utilisateurs dans votre registre d'utilisateurs, à l'aide de la documentation correspondante.

Ajout d'utilisateurs d'administration :

Si vous avez installé la version imbriquée de WebSphere Application Server, un domaine de référentiel d'utilisateurs basé sur un fichier, également appelé *référentiel fédéré*, a été configuré pour les utilisateurs d'administration de Tivoli Federated Identity Manager. Si vous préférez gérer des utilisateurs d'administration par l'intermédiaire du même registre d'utilisateurs que celui dans lequel vos utilisateurs de connexion unique sont configurés, vous devez les ajouter à ce registre d'utilisateurs.

Avant de commencer

L'utilisateur administratif que vous avez pendant l'installation a été créé dans le référentiel d'utilisateurs par défaut lors de l'installation de Tivoli Federated Identity Manager.

Procédure

1. Créez l'utilisateur à l'aide de la documentation associée à votre registre d'utilisateurs. Utilisez l'ID de nom et le mot de passe utilisés par l'administrateur lors de l'installation de Tivoli Federated Identity Manager.
2. Suivez les instructions de la section «Configuration du registre d'utilisateurs pour embeddedWebSphere», à la page 105.

Configuration du registre d'utilisateurs pour embeddedWebSphere

Si vous avez installé la version intégrée de WebSphere Application Server, cela signifie que le référentiel fédéré a été configuré en tant que registre d'utilisateurs. Si vous souhaitez utiliser un registre d'utilisateurs autre que le référentiel fédéré par défaut, modifiez les paramètres de WebSphere Application Server.

Procédure

1. Connectez-vous à la console.
2. Sélectionnez **Sécurité** > **Administration, application et infrastructure sécurisées**. L'onglet Configuration s'affiche.
3. Cliquez sur **Assistant de configuration des paramètres de sécurité** pour modifier le registre d'utilisateurs utilisé par le composant d'exécution WebSphere. Le panneau **Spécifier l'étendue de la protection** apparaît.
4. Vérifiez que la case **Activer la sécurité des applications** est cochée.
5. Cliquez sur **Suivant**. Le panneau **Sécuriser l'environnement de traitement des applications** apparaît.
6. Sélectionnez l'option correspondant au registre d'utilisateurs de votre choix :
   - **Référentiels fédérés**
   - **Registre LDAP autonome**
   - **Système d'exploitation local**
   - **Registre personnalisé autonome**
7. Cliquez sur **Suivant**. Le panneau **Configurer le référentiel d'utilisateurs** s'affiche.
8. Indiquez des valeurs pour chaque paramètre de configuration du registre. Pour obtenir une description des zones présentées, consultez l'aide en ligne.
9. Cliquez sur **Suivant** et quittez l'assistant.
10. Sauvegardez les modifications apportées à votre configuration.
11. Arrêtez WebSphere Application Server.
12. Redémarrez WebSphere Application Server. Vous devez utiliser le nom d'administrateur que vous avez choisi pour vous connecter et effectuer ces modifications.
13. Dans la console, sélectionnez **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Propriétés du domaine**.
14. Dans la section Sécurité WebSphere du panneau, mettez à jour les valeurs suivantes :

    Nom de l'utilisateur d'administration
    Remplacez l'entrée existante par le nom de compte administrateur LDAP entré à l'étape précédente. Par exemple, `ldapadmin`

    Mot de passe d'administration
    Entrez le mot de passe de l'administrateur LDAP.
15. Sauvegardez les modifications.
16. Arrêtez WebSphere Application Server.
17. Redémarrez WebSphere Application Server.

Configuration d'une connexion SSL au registre d'utilisateurs

Après avoir configuré votre registre d'utilisateurs, activez SSL pour protéger la connexion entre SSL et le serveur.

Pourquoi et quand exécuter cette tâche

Pour obtenir des instructions, consultez le centre de documentation de WebSphere Application Server 8.0 à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Pour plus d'informations sur la création de connexions SSL, sélectionnez **WebSphere Application Server (Distributed platforms and Windows)** > **Securing applications and their environment** > **Securing communications**.

Il peut être également nécessaire de consulter la documentation de votre registre d'utilisateurs.

Personnalisation du formulaire de connexion

Si vous utilisez une authentification sur la base de formulaires pour authentifier les utilisateur via une connexion unique, un formulaire de connexion et une page d'erreur liée à celui-ci sont affichés à votre intention.

Pourquoi et quand exécuter cette tâche

Le formulaire de connexion et la page d'erreur font partie des pages de réponse générées par Tivoli Federated Identity Manager. Vous pouvez personnaliser les pages afin de les rendre conformes aux besoins de votre environnement et de modifier leur aspect. Les identificateurs de ces pages sont les suivants :

proper/login/formlogin.html
: La page de connexion s'affiche côté client Web lorsque la connexion unique est déclenchée chez le fournisseur d'identité par un utilisateur non authentifié.

proper/login/formloginerror.html
: En cas d'échec d'authentification, la page d'erreur s'affiche.

Configuration de l'authentification SPNEGO

Si vous utilisez WebSphere Application Server comme serveur point de contact doté de la fonction d'authentification SPNEGO, vous devez effectuer plusieurs tâches de configuration.

Pourquoi et quand exécuter cette tâche

Réalisez ces tâches pour configurer l'authentification SPNEGO :

Procédure

1. Configuration de Microsoft Active Directory, notamment les tâches suivantes :
   a. Création d'un utilisateur Active Directory pour l'administrateur WebSphere.
   b. Création d'un utilisateur Active Directory contenant le nom principal de service (SPN) du serveur Tivoli Federated Identity Manager.
   c. Génération d'un fichier de clés Kerberos et attribution du nom SPN pour l'utilisateur Active Directory créé à l'étape 1b.
   d. Collecte des paramètres de configuration d'Active Directory
2. Configuration du domaine et des connexions utilisateurs Windows.
3. Configuration de WebSphere Application Server, notamment :
   a. Configuration de la sécurité d'administration, en utilisant Active Directory comme type de registre d'utilisateurs LDAP.
   b. (Facultatif) Configuration d'une connexion SSL sur Active Directory.

4. Activation de WebSphere SPNEGO et du support TAI (Trust Association Interceptor), à l'aide d'Integrated Solutions Console.

   (Facultatif) Personnalisez les attributs TAI, en fonction des besoins de votre environnement.
5. Instructions fournies à vos utilisateurs pour la configuration d'Internet Explorer, comme suit :
   a. Ajout du nom d'hôte sous forme d'hôte sécurisé dans la zone intranet.
   b. Activation de l'authentification intégrée de Windows.

Configuration d'Active Directory pour SPNEGO

Utilisez Microsoft Active Directory en tant que registre d'utilisateurs lorsque vous utilisez WebSphere Application Server avec l'authentification SPNEGO.

Avant de commencer

Vous devez effectuer plusieurs tâches de configuration dans Microsoft Active Directory :

- Créez un d'utilisateur pour l'administration de WebSphere.
- Créez un utilisateur contenant le nom principal de service (SPN) de votre serveur Tivoli Federated Identity Manager.
- Générez un fichier de clés Kerberos et attribuez le nom SPN à l'utilisateur Active Directory qui a été créé à cet effet.
- Collectez les paramètres de connexion d'Active Directory.

Microsoft Active Directory est un composant obligatoire dans un environnement de fournisseur d'identité dans lequel IBM WebSphere Application Server, doté de l'authentification SPNEGO, est utilisé en tant que serveur point de contact. Installez et configurez Microsoft Active Directory pour votre réseau avant de commencer cette tâche.

Pourquoi et quand exécuter cette tâche

Pour plus de détails sur l'exécution des étapes de cette procédure, reportez-vous à la documentation de Microsoft Active Directory.

Procédure

1. A l'aide de la console des configuration des utilisateurs et ordinateurs Active Directory, créez un utilisateur Active Directory pour l'administrateur WebSphere. Cet utilisateur correspond à un compte utilisateur classique dans Active Directory, sans aucun privilège de compte particulier. Employez un nom d'utilisateur reflétant le rôle de cet utilisateur. Utilisez par exemple `wasadmin`.
2. A l'aide de la console de configuration des utilisateurs et ordinateurs Active Directory, créez un utilisateur contenant le nom principal de service (SPN) de votre serveur Tivoli Federated Identity Manager. Le nom d'utilisateur de ce compte n'est pas important. Le nom de principal de service de cet utilisateur sera défini à l'aide de l'utilitaire **ktpass** dans une étape ultérieure. Attribuez à cet utilisateur un mot de passe correctement sécurisé et définissez ce dernier de sorte qu'il n'arrive jamais à expiration.

3. Exécutez la commande **ktpass** pour générer un fichier keytab pour l'utilisateur Kerberos de WebSphere. L'utilitaire ktpass est inclus dans le module d'outils de support de Microsoft Windows 2003 Server Support Tools package. Utilisez les paramètres suivants avec la commande :

Tableau 9. Paramètres à utiliser avec la commande Microsoft Windows **ktpass**

Paramètre	Exemple de valeur	Description
-out	`was1-krb5.keytab`	Nom de fichier dans lequel est stockée la clé secrète qui sera utilisée ultérieurement pour la validation de l'authentification Kerberos sur le serveur WebSphere. Ce fichier est chargé sur le serveur WebSphere lorsque vous activez SPNEGO. Voir «Activation et configuration de l'authentification SPNEGO», à la page 113.
-princ	`HTTP/ibm-fim611-1.fimtest. example.com@FIMTEST .EXAMPLE.COM`	Nom de principal de service Kerberos à utiliser pour la génération de la clé. Ce nom fait la distinction entre les majuscules et les minuscules et doit obligatoirement commencer par `HTTP/`. La portion située après `HTTP/` doit correspondre au nom de domaine DNS qualifié complet de l'adresse URL que les utilisateurs voient s'afficher dans leur navigateur lorsqu'ils accèdent au serveur WebSphere.
-pass	`*`	Mot de passe à définir pour le principal Kerberos. Lorsqu'une valeur `*` est indiquée, l'utilisateur est invité à entrer le mot de passe. Le mot de passe doit correspondre à l'utilisateur créé à l'étape 2, à la page 107.
-mapuser	`was-1`	Utilisateur Active Directory vers lequel le principal de service Kerberos est mappé. La valeur définie ici doit correspondre au nom d'utilisateur créé à l'étape 2, à la page 107.
-mapOp	`set`	Indique que le nom SPN doit remplacer toutes les valeurs existantes mappées pour cet utilisateur Active Directory.

L'exemple suivant illustre une exécution de la commande **ktpass**. Il explique également comment utiliser la commande **setspn** pour répertorier des noms de principal de service pour l'utilisateur was-1, à titre d'information et de vérification.

```
C:\Program Files\Support Tools>ktpass -out was1-krb5.keytab
 -princ HTTP/ibm-fim611-1.fimtest.example.com@FIMTEST.EXAMPLE.COM
 -pass * -mapuser was-1 -mapOp set

Targeting domain controller: ibm-fimtest-ad.fimtest.example.com

Successfully mapped HTTP/ibm-fim611-1.fimtest.example.com:

Type the password again to confirm:

Key created.

Output keytab to was1-krb5.keytab:

Keytab version:0x502

keysize 76 HTTP/ibm-fim-611-1.fimtest.example.com@FIMTEST.EXAMPLE.COM
 ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x3 (DES-CBC-MD5)
 keylength 8 (0x799b26bfe9ad3ba4)

Account was-1 has been set for DES-only encryption.

C:\Program Files\Support Tools>setspn -1 was-1

Registered ServicePrincipalNames for
CN=was-1,CN-Users,DC=fimtest,DC=ibm,DC=com:

   HTTP/ibm-fim611-1.fimtest.ibm.com
```

Figure 3. Exemple de commande ktpass

Le fichier keytab créé au cours de cette étape est chargé sur le serveur Tivoli Federated Identity Manager durant la configuration de WebSphere Application Server. Pour plus détails, voir «Configuration de WebSphere pour SPNEGO», à la page 110.

4. Collectez les informations de configuration de la connexion Active Directory permettant d'utiliser la configuration WebSphere Application Server en procédant comme suit :
   a. Recherchez les informations suivantes dans l'arborescence LDAP d'Active Directory.

      Nom d'hôte
      Nom d'hôte du serveur Active Directory.

      Port Numéro de port du serveur Active Directory.

      DN de base
      Nom distinctif (DN) de recherche de base pour les utilisateurs Active Directory.

      DN BIND
      Nom distinctif Active Directory d'un administrateur pour l'exécution de recherches LDAP. Cette valeur ne doit pas nécessairement correspondre au nom distinctif du compte d'administration de domaine, mais plutôt à celui d'un utilisateur valide d'Active Directory.

      Mot de passe Bind
      Mot de passe de l'utilisateur représenté par le DN BIND.

   b. Si une connexion SSL est requise par Active Directory, WebSphere doit être configuré avec le certificat de l'autorité de certification émettrice du contrôleur de domaine. Si le composant Windows Certificate Services a été

installé sur le contrôleur de domaine, il s'agit du certificat de CA lié à Certificate Services sur ce contrôleur de domaine.

Pour exporter le certificat de CA vers un fichier, procédez comme suit :

1) Ouvrez **Outils d'administration** > **Certification Authority**.
2) Cliquez avec le bouton droit sur le nom de l'autorité de certification de niveau supérieur.
3) Cliquez sur **Propriétés**.
4) Cliquez sur l'onglet **Général** puis sur **Afficher le certificat**.
5) Cliquez sur l'onglet **Details**, puis sur **Copy to File**.

   Le fichier est enregistré au format binaire DER. Utilisez ce fichier dans le cadre de la configuration de WebSphere, si l'authentification sur le serveur SSL est nécessaire pour établir le contact avec le serveur Active Directory via l'interface LDAP/SSL.

Configuration du domaine et des connexions utilisateurs Windows

Pour utiliser la connexion unique via le Bureau Windows, les connexions bureau des utilisateurs doivent être authentifiées sur le domaine Windows.

Pourquoi et quand exécuter cette tâche

L'utilisation de la connexion unique via le bureau Windows pour accéder au serveur Tivoli Federated Identity Manager nécessite que les utilisateurs se connectent à leur bureau en tant que membres d'un domaine Windows. En particulier, le domaine Windows doit prendre en charge l'authentification Kerberos auprès d'un service Microsoft Active Directory. Pour plus de détails sur la création de cet environnement, consultez la documentation Microsoft.

Cette configuration permet au fournisseur d'identité de prendre en charge les utilisateurs internes qui sont connectés à l'intranet de votre fournisseur d'identité, à l'aide d'une connexion bureau à un domaine Windows. Cependant, un fournisseur d'identité peut également être amené à prendre en charge des utilisateurs externes qui ne possèdent pas de connexion au domaine Windows. Ces utilisateurs externes doivent s'authentifier à l'aide d'un formulaire de connexion.

Par défaut, le support SPNEGO TAI dans Tivoli Federated Identity Manager affiche un formulaire de connexion si un utilisateur qui ne s'est pas authentifié via la connexion bureau tente une connexion unique. Par défaut, le formulaire de connexion est le modèle de formulaire de connexion fourni avec Tivoli Federated Identity Manager.

Vous pouvez personnaliser la présentation de ce formulaire, selon la procédure décrite dans la section «Personnalisation du formulaire de connexion», à la page 106. Si vous ne souhaitez pas afficher ce formulaire de connexion, vous pouvez modifier les attributs TAI selon la procédure décrite dans la section «Configuration des attributs TAI personnalisés», à la page 116.

Configuration de WebSphere pour SPNEGO

Avant de pouvoir utiliser WebSphere Application Server avec SPNEGO, vous devez configurer la sécurité des applications WebSphere en définissant Active Directory en tant que référentiel d'utilisateurs.

Pourquoi et quand exécuter cette tâche

Les étapes sont les suivantes :

- (Facultatif) Chargement du certificat racine de l'autorité de certification du serveur Active Directory afin d'activer le protocole SSL entre le serveur et Active Directory.
- Activation de la sécurité des applications WebSphere avec Active Directory en tant que registre d'utilisateurs.
- Configuration des caractéristiques du répertoire LDAP autonome de sorte que celui-ci pointe sur le serveur Active Directory.

Procédure

1. (Facultatif) Chargez le certificat racine de l'autorité de certification du serveur Active Directory. Cette étape n'est requise que si vous utilisez LDAP/SSL pour communiquer avec le serveur Active Directory.
2. Avant de poursuivre cette procédure, veillez à effectuer les étapes de la section «Configuration d'Active Directory pour SPNEGO», à la page 107, notamment l'étape 4b, à la page 109. Si vous n'utilisez pas le protocole, passez à l'étape suivante.
   a. Connectez-vous à la console.
   b. Cliquez sur **Certificat SSL et gestion des clés**.
   c. Dans le panneau Certificat SSL et gestion des clés, cliquez sur **Magasins de clés et certificats**.
   d. Dans le panneau Magasins de clés et certificats, cliquez sur **NodeDefaultTrustStore**.
   e. Dans le panneau NodeDefaultTrustStore, cliquez sur **Certificats de signataires**.
   f. Dans le panneau Certificats de signataires, cliquez sur **Ajouter** pour ajouter un nouveau signataire.
   g. Complétez les détails du certificat de signataire. Utilisez les valeurs suivantes :

Tableau 10. Caractéristiques du certificat de signataire dans l'environnement SPNEGO

Nom de zone	Valeur
Alias	Nom d'alias du certificat de CA issu d'Active Directory. Par exemple, vous pouvez utiliser le nom du contrôleur de domaine Active Directory.
Nom de fichier	Chemin et nom de fichier du certificat. Il est à noter que ce chemin d'accès, ainsi que le nom du fichier, sont définis sur le serveur où WebSphere Application Server est installé, et non sur le serveur qui exécute le navigateur. Cela signifie que le fichier doit être copié sur le serveur WebSphere avant la réalisation de cette étape.
Type de données	Format de fichier du certificat. Utilisez le même format que celui de l'étape 4b, à la page 109.

   Une fois que le chargement du certificat aboutit, ce dernier s'affiche dans la liste des certificats de signataire.
   h. Cliquez sur **OK**.
3. Activez la sécurité des applications WebSphere avec Active Directory en tant que registre d'utilisateurs.

Remarque : Pour effectuer cette étape, vous devez être en mesure d'accéder au serveur Active Directory à l'aide du port 389 (c'est-à-dire, sans utiliser SSL). Pendant cette étape, l'assistant de configuration de la sécurité effectue un test de connexion qui ne prend pas en charge SSL. Si ce test échoue, la configuration n'aboutit pas. Vous pouvez activer LDAP/SSL une fois que le test et la configuration ont été terminés.

a. Sur la console, sélectionnez **Sécurité** > **Administration, applications et infrastructure sécurisées**.
b. Cliquez sur le bouton **Assistant de Configuration des paramètres de sécurité** pour démarrer l'assistant de configuration des paramètres de sécurité.
c. Cliquez sur **Suivant**.
d. A l'étape 1 de l'assistant, assurez-vous que la case **Activer la sécurité des applications** est cochée.
e. Cliquez sur **Suivant**.
f. A l'étape 2 de l'assistant, sélectionnez **Répertoire LDAP autonome**.
g. Cliquez sur **Suivant**.
h. A l'étape 3 de l'assistant, entrez les paramètres ci-dessous.

Tableau 11. Paramètres du répertoire LDAP dans l'environnement SPNEGO

Nom de zone	Valeur
Nom de l'utilisateur administratif primaire	Utilisez le nom d'administrateur WebSphere créé dans Active Directory.
Type de serveur LDAP	Microsoft Active Directory
Hôte	Nom d'hôte du serveur Active Directory. Par exemple : `ibm-fimtest-ad.fimtest.example.com`
Port	Tant que vous n'exécutez pas l'étape de test de cet assistant de configuration, utilisez un port ne faisant pas appel à SSL. Par exemple, utilisez 389.
Nom distinctif (DN) de base	Nom distinctif (DN) de recherche de base pour les entrées utilisateur. Par exemple : `cn=users,dc=fimtest,dc=ibm,dc=com.`
Nom distinctif (DN) BIND	Nom distinctif d'un utilisateur Active Directory valide. Par exemple : `cn=administrator,cn=users,dc=fimtest,dc=ibm,dc=com.`
Mot de passe Bind	Mot de passe Active Directory de l'utilisateur représenté par le DN BIND.

i. Cliquez sur **Suivant**.
j. A l'étape 4 de l'assistant, la connexion au serveur Active Directory est testée.
k. Cliquez sur **Terminer** pour mettre fin à l'assistant.

4. Configurez les caractéristiques du répertoire LDAP autonome de sorte que celui-ci pointe sur le serveur Active Directory.
   a. Sur la console, sélectionnez **Sécurité** > **Administration, applications et infrastructure sécurisées**.
   b. Dans la liste des définitions de domaine disponibles, sélectionnez **Registre LDAP autonome**.
   c. Cliquez sur **Configurer**.
   d. Complétez les caractéristiques de votre configuration, y compris SSL et le port SSL si nécessaire.

e. Cliquez sur **OK** et enregistrez les modifications.

Activation et configuration de l'authentification SPNEGO

Avant de pouvoir utiliser WebSphere Application Server avec SPNEGO, vous devez activer l'authentification SPNEGO dans Tivoli Federated Identity Manager et configurer ses propriétés.

Pourquoi et quand exécuter cette tâche

Utilisez la console pour effectuer cette procédure qui comprend les étapes suivantes :

- Activation de SPNEGO à des fins d'utilisation avec Tivoli Federated Identity Manager.
- Configuration du client WebSphere Kerberos.
- Configuration du fichier de propriétés TAI.
- Configuration des paramètres de démarrage JVM.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Sélectionnez le profil du serveur point de contact que vous utilisez dans votre environnement.
4. Cliquez sur le bouton **Avancé**. Le panneau Paramètres de sécurité de noeud final SOAP s'ouvre.
5. Cliquez sur **Paramètres d'authentification SPNEGO**.
6. Cochez la case **Activer l'authentification SPNEGO**.
7. Entrez vos informations de configuration dans les zones du panneau. Pour obtenir la description de chaque zone, consultez l'aide en ligne.
8. Importez le fichier de clés Kerberos que vous avez créé à l'aide de l'option -out de l'utilitaire ktpass, en procédant comme suit :
   a. Cliquez sur le bouton **Importer le fichier de clés**.
   b. Dans la zone **Emplacement du fichier de clés**, entrez le chemin d'accès au fichier.

      (Facultatif) Utilisez le bouton **Parcourir** pour localiser le fichier.
   c. Cliquez sur **Terminer**.
9. Cliquez sur **OK**.

Configuration du support TAI (Trust Association Interceptor)

Si vous activez et configurez SPNEGO à l'aide de la console, le support TAI est automatiquement activé dans les paramètres de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

En règle générale, aucune autre configuration n'est nécessaire. Le fichier `tai.properties.template` contient des valeurs par défaut pour tous les instances TAI SPNEGO de WebSphere. Pour plus d'informations sur ces valeurs, voir «Attributs de configuration SPNEGO TAI», à la page 114.

Remarque : Pour apporter des modifications à ces valeurs par défaut, suivez les instructions figurant dans la section «Configuration des attributs TAI personnalisés», à la page 116.

Attributs de configuration SPNEGO TAI :

Les attributs de configuration personnalisée SPNEGO (Simple and Protected GSS-API Negotiation Mechanism) TAI (Trust Association Interceptor) contrôlent les différents aspects opérationnels de la méthode SPNEGO TAI. Ces attributs sont stockés dans le fichier `tai.properties.template`.

Contenu

Le fichier se trouve dans le répertoire par défaut suivant :

AIX, Linux ou Solaris

```
/opt/IBM/FIM/etc/tai.properties.template
```

Windows

```
C:\Program Files\IBM\FIM\etc\tai.properties.template
```

En règle générale, vous n'avez pas besoin de modifier ce fichier. Vous pouvez configurer la méthode TAI à l'aide de la console, selon la procédure décrite dans la section «Configuration du support TAI (Trust Association Interceptor)», à la page 113. Cependant, si vous devez apporter des modifications supplémentaires nécessitant la mise à jour du fichier `tai.properties.template`, utilisez les instructions figurant dans la section «Configuration des attributs TAI personnalisés», à la page 116.

Remarque : La version du fichier tai.properties.template installée en même temps que Tivoli Federated Identity Manager contient des attributs complémentaires non fournis avec WebSphere Application Server 8.0. Si votre environnement nécessite l'utilisation d'attributs qui ne sont pas décrits ici, consultez le centre de documentation WebSphere Application Server 8.0 pour obtenir la liste de tous les attributs disponibles en vue de la personnalisation de la configuration SPNEGO TAI. Le centre de documentation est accessible à l'adresse suivante : http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp.

La figure ci-dessous décrit le contenu du fichier `tai.properties.template`.

```
#########################################################
# Fichiers de propriétés modèles pour SPNEGO TAI
#
# Où les valeurs par défaut possibles ont été fournies.
#
#########################################################
#---------------------------------------------------------
# Nom d'hôte
#---------------------------------------------------------
com.ibm.ws.security.spnego.SPN1.hostName=@POCHOST@

#---------------------------------------------------------
# (Facultatif) SpnegoNotSupportedPage
#---------------------------------------------------------
com.ibm.ws.security.spnego.SPN1.spnegoNotSupportedPage=file:///@SPNEGOFAILED@

#---------------------------------------------------------
# (Facultatif) NTLMTokenReceivedPage
#---------------------------------------------------------
com.ibm.ws.security.spnego.SPN1.NTLMTokenReceivedPage=file:///@SPNEGOFAILED@

#---------------------------------------------------------
# (Facultatif) FilterClass
#---------------------------------------------------------
#com.ibm.ws.security.spnego.SPN1.filterClass=com.ibm.ws.spnego.HTTPHeaderFilter

#---------------------------------------------------------
# (Facultatif) Filter
#---------------------------------------------------------
com.ibm.ws.security.spnego.SPN1.filter=request-url%=/sps/wasauth

#---------------------------------------------------------
# (Facultatif) Credential Delegation
#---------------------------------------------------------
#com.ibm.ws.security.spnego.SPN1.enableCredDelegate

#---------------------------------------------------------
# (Facultatif) Credential Delegation
#---------------------------------------------------------
#com.ibm.ws.security.spnego.SPN1.trimUserName=
```

Figure 4. Fichier tai.properties.template

Macros

Les macros ci-dessous sont utilisées dans le fichier `tai.properties.template`.

Tableau 12. Macros utilisées dans le fichier tai.properties.template

Macro	Description	Valeur par défaut
`@POCHOST@`	Nom d'hôte complet du serveur point de contact. Ce nom d'hôte est utilisé dans l'adresse URL du serveur point de contact.	`poc.example.com`
`@SPNEGOFAILED@`	Chemin complet d'un fichier HTML qui est envoyé au navigateur lorsque la négociation de l'authentification SPNEGO n'aboutit pas. Ce fichier HTML redirige automatiquement le navigateur vers l'exemple de page de connexion fournie par Tivoli Federated Identity Manager.	*répertoire_installation*`/etc/spnego_failed.html` Ce paramètre ne peut pas être configuré à l'aide de la console. Le chemin correct est configuré lors de la configuration de SPNEGO.

Configuration des attributs TAI personnalisés :

Configurez les attributs TAI personnalisés pour répondre à vos exigences de déploiement.

Avant de commencer

Le support TAI est activé automatiquement lorsque vous activez SPNEGO à l'aide de la console, selon la procédure décrite dans la section «Activation et configuration de l'authentification SPNEGO», à la page 113. Cependant, pour personnaliser les attributs TAI, vous devez modifier le fichier tai.properties.template.

Pourquoi et quand exécuter cette tâche

Consultez le contenu du fichier tai.properties.template dans la section «Attributs de configuration SPNEGO TAI», à la page 114.

Procédure

1. Recherchez le fichier et faites-en une copie de sauvegarde. Le fichier se trouve dans le répertoire par défaut suivant :

   AIX, Linux ou Solaris
   ```
   /opt/IBM/FIM/etc/tai.properties.template
   ```

   Windows
   ```
   C:\Program Files\IBM\FIM\etc\tai.properties.template
   ```
2. Ouvrez le fichier dans un éditeur de texte.
3. Apportez les modifications appropriées pour votre environnement.
4. Enregistrez, puis fermez le fichier.

Configuration des navigateurs pour SPNEGO

Les utilisateurs doivent faire appel à la connexion unique via le bureau pour accéder au serveur Tivoli Federated Identity Manager une fois que l'authentification SPNEGO est configurée.

Avant de commencer

Conditions requises :

- Le navigateur de l'utilisateur reconnaît le serveur Tivoli Federated Identity Manager en tant que *site intranet*.
- Le navigateur de l'utilisateur est activé pour l'authentification intégrée de Windows.

Les instructions contenues dans cette procédure concernent Windows Internet Explorer 6 et versions ultérieures. Pour les autres types de navigateur, tels que Mozilla Firefox, consultez la documentation correspondante.

Pourquoi et quand exécuter cette tâche

En général, la configuration du navigateur pour SPNEGO implique les opérations suivantes :

- Ajoutez dans la liste des sites intranet locaux le nom d'hôte du poste WebSphere Application Server utilisé avec Tivoli Federated Identity Manager.

- Vérifiez que l'option d'authentification intégrée de Windows est sélectionnée dans les paramètres de sécurité avancés du navigateur.

Procédure

1. Ajoutez le nom d'hôte :
   a. Lancez Windows Internet Explorer.
   b. Cliquez sur **Outils** > **Options Internet**.
   c. Cliquez sur l'onglet **Sécurité**.
   d. Cliquez sur **Intranet local**.
   e. Cliquez sur le bouton **Sites**. Assurez-vous que la case **Inclure tous les sites locaux (intranet) non mentionnés dans d'autres zones** est cochée.
   f. Cliquez sur **Avancé**.
   g. Ajoutez les sites Web de WebSphere Application Server, tels qu'ils sont affichés dans le navigateur, à l'aide de http ou https, en fonction de vos besoins.

      Remarque : Ce nom d'hôte doit correspondre au nom principal configuré pour le fichier de clés.
      Par exemple :

      ```
      http://ibm-fim611-1.fimtest.example.com
      ```

      ```
      https://ibm-fim611-1.fimtest.example.com
      ```
2. Vérifiez que l'authentification intégrée de Windows est activée :
   a. Lancez Windows Internet Explorer.
   b. Cliquez sur **Outils** > **Options Internet**.
   c. Cliquez sur l'onglet **Avancé** et faites défiler la page jusqu'à la section Sécurité.
   d. Assurez-vous que la case **Activer l'authentification intégrée de Windows (nécessite un redémarrage)** est cochée.
   e. Enregistrez les modifications, puis redémarrez le navigateur, si nécessaire.

Serveur point de contact WebSphere pour un fournisseur de services

Il existe plusieurs options de configuration lorsque vous exercez le rôle du fournisseur de service dans votre fédération.

Si vous utilisez WebSphere Application Server en tant que serveur point de contact, vous pouvez utiliser n'importe lequel des types de serveurs suivants pour héberger les applications Web cible auxquelles vos utilisateurs de connexion unique accèdent généralement :

- IBM WebSphere Application Server 5.1 ou 6.0 ou version supérieure

  Dans la plupart des cas, les applications Web peuvent être hébergées sur une installation de WebSphere Application Server distincte du serveur sur lequel Tivoli Federated Identity Manager est installé.

  Vous pouvez utiliser le même serveur pourTivoli Federated Identity Manager et vos applications dans les cas suivants :

  – Si votre installation WebSphere Application Server est la version 6.1
  – Si votre installation WebSphere Application Server remplit les conditions d'hébergement de l'installation de Tivoli Federated Identity Manager et des applications Web
- Microsoft Internet Information Service 6.0

- IBM HTTP Server 6.1
- Apache HTTP Server 2.0 et 2.2

Si vous sélectionnez un serveur autre que WebSphere Application Server en tant qu'hôte pour vos applications, vous devez installer le plug-in du serveur Web Tivoli Federated Identity Manager sur votre serveur d'applications. La figure suivante montre un exemple d'environnement Tivoli Federated Identity Manager dans lequel les applications sont hébergées par un serveur Web séparé.

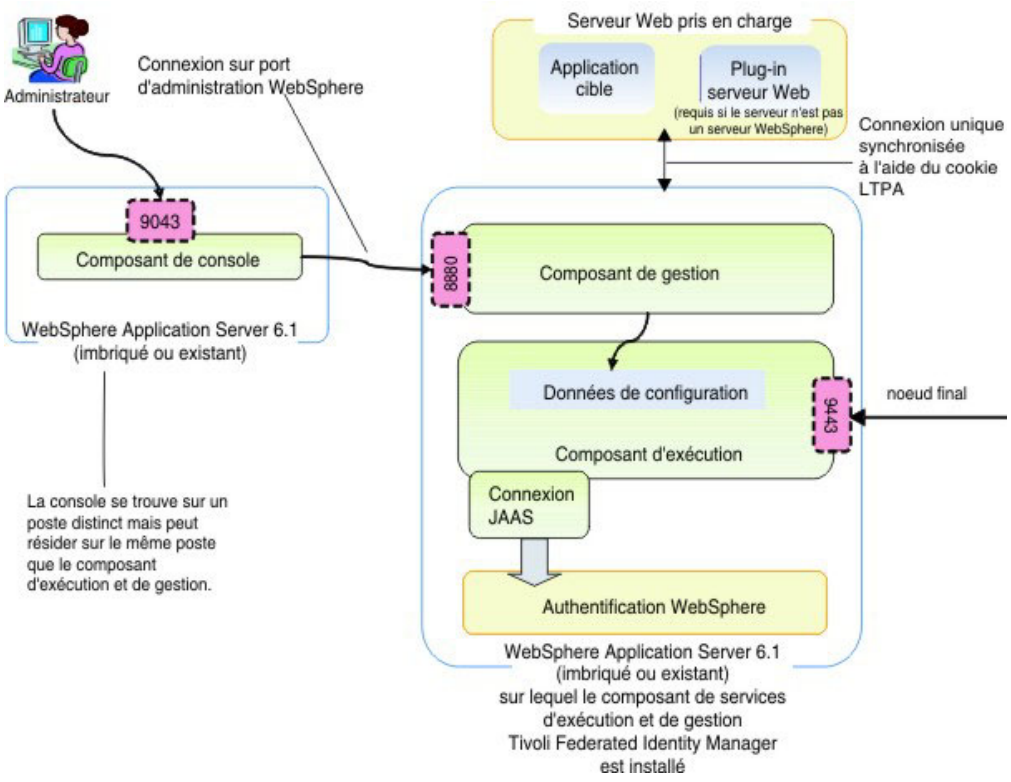


Figure 5. Exemple de Tivoli Federated Identity Manager avec un serveur d'applications Web

Dans la configuration illustrée, l'application cible est hébergée par un serveur distinct du serveur Tivoli Federated Identity Manager. L'utilisateur s'authentifie auprès du fournisseur d'identité, et les données d'identification sont transférées du fournisseur d'identité vers Tivoli Federated Identity Manager. Le fournisseur de services valide le jeton dans Tivoli Federated Identity Manager, et renvoie un cookie LTPA contenant l'identité de l'utilisateur. Il renvoie également tous les attributs contenus dans le jeton, ou ajoutés par les règles de mappage du fournisseur de service.

L'utilisateur est redirigé par le biais d'un protocole de connexion unique vers l'application cible sur laquelle le cookie LTPA est transféré du noeud Tivoli Federated Identity Manager vers celui du serveur Web. La clé LTPA doit être partagée entre ces noeuds pour que le cookie soit reconnu.

Si le serveur Web n'est pas un une instance de WebSphere Application Server, le plug-in du serveur Web Tivoli Federated Identity Manager doit être installé sur ce serveur. Le module d'extension extrait les données d'identité et les attributs du cookie LTPA, puis les fournit à l'application cible via un(e) ou plusieurs en-têtes HTTP ou variables de serveurs.

Exigences liées à l'environnement

Les applications cible peuvent être hébergées par l'un des serveurs suivants :

- WebSphere Application Server 5.1, 6.0 ou version supérieure
- Microsoft Internet Information Server 6.0
- IBM HTTP Server 6.1
- Apache HTTP Server 2.0 et 2.2

Avertissement : Si vous choisissez d'héberger les applications cible sur un serveur autre que WebSphere Application Server, vous devez installer le plug-in du serveur Web Tivoli Federated Identity Manager sur ce serveur.

- Les applications doivent être capables d'accepter l'identité des utilisateurs au moyen d'un en-tête HTTP ou d'une variable de serveur.
- Un registre d'utilisateurs est requis dans votre environnement à la fois pour votre serveur point de contact et votre serveur d'applications. Les utilisateurs auxquels vous allez fournir les fonctionnalités de connexion unique doivent exister dans les deux registres d'utilisateurs. Configurez un registre d'utilisateurs pour le serveur distinct qui héberge votre application cible.

  Les exemples de serveur distinct peuvent être, une autre instance de WebSphere Application Server, ou un serveur pris en charge doté d'un module d'extension tel qu'un serveur IHS, IIS ou Apache. Sélectionnez un registre d'utilisateurs exploitable par votre serveur point de contact et votre serveur Web afin de réduire le nombre de registres d'utilisateurs que vous devrez gérer dans votre environnement.

Exigences applicables aux modules d'extension

Vous devez vous assurer que votre environnement répond aux exigences suivantes :

- Les applications doivent être capables d'accepter l'identité des utilisateurs au moyen d'un en-tête HTTP ou d'une variable de serveur.
- Le nom d'utilisateur utilisé pour chaque connexion unique doit exister à la fois dans le registre d'utilisateurs de WebSphere Application Server et dans le registre d'utilisateurs du serveur Web. Il doit se trouver dans le même emplacement où Tivoli Federated Identity Manager est installé.
- Le serveur Tivoli Federated Identity Manager et le serveur Web doivent figurer dans le même domaine DNS et le cookie LTPA doit être configuré en tant que cookie de domaine.
- Le fichier de clés et le mot de passe LTPA doivent résider à la fois sur le serveur Tivoli Federated Identity Manager et sur le serveur Web sur lequel le module d'extension est installé.

Configuration de WebSphere

Pour configurer le serveur point de contact WebSphere Application Server, reportez-vous à la rubrique «Configuration d'un serveur point de contact WebSphere Application Server (fournisseur de services)».

Configuration d'un serveur point de contact WebSphere Application Server (fournisseur de services)

Si vous utilisez WebSphere Application Server comme serveur point de contact, vous devez effectuer plusieurs tâches de configuration.

Pourquoi et quand exécuter cette tâche

Avertissement : Avant d'accomplir les tâches décrites dans la présente section, assurez-vous que les paramètres sont corrects en appliquant la procédure «Confirmation des propriétés de sécurité de WebSphere Application Server», à la page 95.

Configuration du cookie LTPA

En général, la connexion unique fédérée n'est disponible que si les applications partagent un nom de domaine commun avec le noeud final du service d'assertion client du fournisseur de services. Pour vous assurer que vos applications partagent le nom de domaine correct, vous devez configurer le cookie LTPA sous forme de cookie e domaine, à l'aide du domaine de votre noeud final de service d'assertion client.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Administration, applications et infrastructure sécurisées** > **Sécurité Web**.
3. Cliquez sur **connexion unique**.
4. Pour restreindre le cookie LTPA aux sessions SSL, sélectionnez l'option **SSL requis**.
5. Renseignez la zone Nom de domaine. Faites précéder d'un point (.) le nom du domaine. La définition du nom de domaine permet de garantir que le LTPA est mis à la disposition de tous les serveurs Web membres du domaine spécifié.
6. Désélectionnez la case à cocher **Mode d'interopérabilité**. Le mode d'interopérabilité entraîne la création de deux cookies (un cookie LPTA version 1 et un cookie LTPA version 2) dans le navigateur. Le filtrageLe plug-in de serveur Web Tivoli Federated Identity Manager prend uniquement en charge les cookies LTPA version 2.
7. Cliquez sur **OK**, puis sur **Enregistrer**.

Que faire ensuite

Des informations sur la configuration correcte du domaine peuvent être consultées dans la rubrique relative à la mise en oeuvre d'une connexion unique en vue de minimiser les authentifications utilisateur sur le Web, dans le centre de documentation de WebSphere Application Server 8.0, à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp.

Définition des attributs du jeton LTPA

Par défaut, tous les attributs disponibles sont inclus dans le jeton LTPA. Si vous souhaitez restreindre les attributs à certains attributs spécifiques, vous devez modifier les paramètres de filtrage des attributs sur l'instance WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Remarque : Votre application cible doit être configurée de manière à utiliser les attributs inclus dans le jeton LTPA. Pour plus d'informations sur les rubriques relatives au développement, consultez les liens aux documents developerWorks sur la page de bienvenue du centre de documentation à l'adresse suivante : http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc_6.2.2/ic/ic-homepage.html.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Sécuriser l'administration, les Applications et les infrastructures**.
3. Développez la liste des rubriques **JAAS (Java Authentication and Authorization Service)**.
4. Cliquez sur **Connexions système**.
5. Dans le panneau des connexions système JAAS, sélectionnez **FIM_OUTBOUND**.
6. Dans la section Propriétés Supplémentaires du panneau FIM_OUTBOUND, cliquez sur **Modules de connexion JAAS**.
7. Sélectionnez le nom de classe du module de connexion pour le mappage d'attributs avec le point de contact WebSphere.

   ```
   com.tivoli.am.fim.fedmgr2.was.jaas.login.
            WASPocAttributesMapLoginModule
   ```

   La liste des propriétés de configuration s'affiche.

   Remarque : Si vous souhaitez supprimer tous les attributs, cochez la case située à côté de l'option **ssoAttributeNames**. Cliquez sur **Supprimer**. Sinon, pour modifier les attributs, exécutez les étapes restantes.
8. Cliquez sur **ssoAttributeNames** pour afficher les propriétés par défaut. Le paramètre **ssoAttributeNames** est configuré par défaut avec la valeur " * " dans la zone **Valeur**, ce qui signifie que tous les attributs doivent être inclus dans le jeton.
9. Si vous souhaitez modifier les attributs, supprimer la valeur " * " et entrez un nom d'attribut, tel que `AuthenticationMethod`, ou plusieurs noms d'attributs tels que `AuthenticationMethod,AuthenticationInstant`. Si vous spécifiez des attributs multiples, séparez-les par une virgule (,).

   Remarque : Les attributs que vous pouvez spécifier dépendent du niveau de personnalisation et de configuration de votre application cible.
10. Cliquez sur **OK**.

Sélection et installation d'un registre d'utilisateurs

Un registre d'utilisateurs est requis si vous utilisez WebSphere Application Server comme serveur point de contact. Le registre d'utilisateurs sert de référentiel pour les informations relatives aux utilisateurs auxquels vous fournissez des fonctionnalités de connexion unique. Il peut également servir de référentiel pour les informations relatives aux utilisateurs d'administration de votre environnement ou vous pouvez choisir de conserver les utilisateurs d'administration dans un registre d'utilisateurs distinct.

Avant de commencer

Dans la mesure où vous utilisez WebSphere Application Server en tant que serveur point de contact, vous pouvez choisir un registre d'utilisateurs à partir de nombreuses options. Reportez-vous au centre de documentation de WebSphere Application Server 8.0 à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Recherchez ensuite des informations sur le choix d'un registre d'utilisateurs, en sélectionnant **WebSphere Application Server (Distributed platforms and Windows)** > **Securing applications and their environment** > **Authenticating users** > **Selecting a registry or repository**.

- Si vous utilisez une installation existante de WebSphere Application Server, il se peut qu'un registre d'utilisateurs compatible soit déjà installé et configuré.
- Si vous utilisez une nouvelle installation de la version imbriquée de WebSphere Application Server, vous disposez des options suivantes :
  - Utiliser le domaine de référentiel d'utilisateurs basé sur un fichier, qui a été installé avec la version imbriquée de WebSphere Application Server. L'utilisateur d'administration a été configuré dans ce registre pendant l'installation. Les autres tâches requises pour l'ajout des utilisateurs de connexion unique sont indiquées ultérieurement dans ce chapitre.
  - Utiliser un registre d'utilisateurs différent. Pour plus d'informations sur vos options de registre d'utilisateurs, consultez la documentation de WebSphere Application Server. Ensuite, installez et configurez le registre d'utilisateurs choisi, si vous n'utilisez pas un registre d'utilisateurs déjà existant. Configurez ensuite WebSphere pour qu'il utilise ce registre d'utilisateurs. Voir «Configuration de WebSphere pour l'utilisation du registre d'utilisateurs», à la page 123.

Remarque : Si votre application cible est destinée à être hébergée sur un serveur séparé, tel qu'une autre instance de WebSphere Application Server ou un serveur pris en charge doté d'un module d'extension tel qu'un serveur IHS, IIS ou Apache, vous devez également configurer un registre d'utilisateurs pour ce serveur. Sélectionnez un registre d'utilisateurs exploitable par votre serveur point de contact et votre serveur d'applications cible afin de réduire le nombre de registres d'utilisateurs que vous devrez gérer dans votre environnement

Configuration du registre d'utilisateurs

La configuration du registre d'utilisateurs est une étape importante de la configuration globale.

Avant de commencer

Avant de procéder à cette tâche, vous devez sélectionner le registre d'utilisateurs à utiliser et l'installer selon la procédure décrite dans la section «Sélection et installation d'un registre d'utilisateurs», à la page 121.

Pourquoi et quand exécuter cette tâche

Dans ce registre, créez des utilisateurs auxquels fournir des fonctionnalités de connexion unique. Vous pouvez également créer des utilisateurs pour les administrateurs de votre environnement ou choisir de conserver les utilisateurs d'administration dans un référentiel distinct.

Ajout d'utilisateurs de connexion unique :

Dans l'environnement du fournisseur de services, le registre d'utilisateurs est utilisé lors de la création de l'identité locale qui est nécessaire pour que les utilisateurs accèdent à l'application cible. Ajoutez ces utilisateurs dans votre registre d'utilisateurs, à l'aide de la documentation correspondante.

Ajout d'utilisateurs d'administration :

Si vous avez installé la version imbriquée de WebSphere Application Server, un domaine de référentiel d'utilisateurs basé sur un fichier et désigné par *référentiel fédéré* a été configuré pour les utilisateurs d'administration de Tivoli Federated Identity Manager. Si vous préférez gérer les utilisateurs d'administration via le

même registre d'utilisateurs que celui dans lequel vos utilisateurs de connexion unique sont configurés, vous devez les ajouter dans ce registre d'utilisateurs.

Avant de commencer

Un utilisateur d'administration a été créé dans le référentiel d'utilisateurs par défaut pendant l'installation de Tivoli Federated Identity Manager.

Pourquoi et quand exécuter cette tâche

Pour ajouter cet utilisateur dans un registre d'utilisateurs différent, procédez comme suit :

Procédure

1. Créez l'utilisateur à l'aide de la documentation de votre registre d'utilisateurs.
2. Suivez les instructions de la section «Configuration de WebSphere pour l'utilisation du registre d'utilisateurs».

Configuration d'une connexion SSL au registre d'utilisateurs :

Après avoir configuré votre registre d'utilisateurs, activez SSL pour protéger la connexion entre SSL et le serveur.

Pourquoi et quand exécuter cette tâche

Pour obtenir des instructions, consultez le centre de documentation de WebSphere Application Server 8.0 à l'adresse http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp. Pour plus d'informations sur la création de connexions SSL, sélectionnez **WebSphere Application Server (Distributed platforms and Windows)** > **Securing applications and their environment** > **Securing communications**.

Il peut être également nécessaire de consulter la documentation de votre registre d'utilisateurs.

Configuration de WebSphere pour l'utilisation du registre d'utilisateurs

Si vous avez installé la version intégrée de WebSphere Application Server, cela signifie que le référentiel fédéré a été configuré en tant que registre d'utilisateurs. Si vous souhaitez utiliser un registre d'utilisateurs autre que le référentiel fédéré par défaut, modifiez les paramètres de WebSphere Application Server.

Avant de commencer

Avant de poursuivre cette tâche, consultez les informations de la section «Sélection et installation du registre d'utilisateurs», à la page 103. Vérifiez que vous avez sélectionné et installé l'option de registre d'utilisateurs appropriée pour votre environnement.

Pourquoi et quand exécuter cette tâche

Pour permettre à WebSphere d'utiliser votre registre d'utilisateurs, procédez comme suit :

Procédure

1. Connectez-vous à la console.
2. Sélectionnez **Sécurité** > **Administration, application et infrastructure sécurisées**. L'onglet Configuration s'affiche.
3. Cliquez sur **Assistant de configuration des paramètres de sécurité** pour modifier le registre d'utilisateurs utilisé par le composant d'exécution WebSphere.
4. Le panneau **Spécifier l'étendue de la protection** apparaît. Vérifiez que la case **Activer la sécurité des applications** est cochée. Cliquez sur **Suivant**.
5. Le panneau **Sécuriser l'environnement de traitement des applications** apparaît. Sélectionnez l'option correspondant au registre d'utilisateurs de votre choix :
   - **Référentiels fédérés**
   - **Registre LDAP autonome**
   - **Système d'exploitation local**
   - **Registre personnalisé autonome**
6. Cliquez sur **Suivant**. Le panneau **Configurer le référentiel d'utilisateurs** s'affiche.
7. Indiquez des valeurs pour chaque paramètre de configuration du registre. Pour obtenir une description des zones présentées, consultez l'aide en ligne.
8. Cliquez sur **Suivant** et quittez l'assistant. Sauvegardez les modifications apportées à votre configuration.
9. Arrêtez WebSphere Application Server.
10. Redémarrez WebSphere Application Server. Vous devez utiliser le nom d'administrateur que vous avez choisi pour vous connecter et effectuer ces modifications.
11. Dans la console, sélectionnez **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Propriétés du domaine**.
12. Dans la section Sécurité WebSphere du panneau, mettez à jour les valeurs suivantes :

    Nom de l'utilisateur d'administration
    Remplacez l'entrée existante par le nom de compte administrateur LDAP entré à l'étape précédente. Par exemple, `ldapadmin`

    Mot de passe d'administration
    Entrez le mot de passe de l'administrateur LDAP.
13. Sauvegardez les modifications.
14. Arrêtez WebSphere Application Server.
15. Redémarrez WebSphere Application Server.

Exportation de la clé LTPA à partir du serveur point de contact

Si vous utilisez votre serveur point de contact WebSphere Application Server avec une application cible hébergée par un autre système WebSphere Application Server ou par un serveur sur lequel un plug-in Tivoli Federated Identity Manager est installé, vous devez exporter votre clé LTPA pour la partager avec votre application cible.

Avant de commencer

Vérifiez que les serveurs d'exportation et d'importation de la clé ont les mêmes paramètres de date et d'heure. Si la date ou l'heure est différente, le serveur d'importation risque d'interpréter à tort la clé comme étant arrivée à expiration.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Sécurité** > **Administration, applications et infrastructure sécurisées** > **Mécanismes d'authentification et expiration**.
3. Dans les zones Mot de passe et Confirmer le mot de passe, entrez le mot de passe utilisé pour chiffrer la clé LTPA. Retenez-le afin de pouvoir l'utiliser plus tard pour importer la clé sur l'autre serveur.
4. Dans la zone **Nom complet du fichier de clés**, indiquez le chemin d'accès complet de l'emplacement dans lequel vous souhaitez enregistrer la clé LTPA exportée. Utilisez le nom de fichier de clés par défaut, `ltpa.keys`. Vous devez posséder un droit d'accès en écriture à ce fichier.
5. Cliquez sur **Exporter les clés** pour exporter la clé vers l'emplacement indiqué dans la zone **Nom complet du fichier de clés**.
6. Indiquez l'**ID de serveur interne** utilisé pour la communication interprocessus entre les serveurs. L'ID de serveur est protégé à l'aide d'un jeton LTPA lorsqu'il est envoyé par un système distant. Cet ID est désactivé par défaut.
7. Cliquez sur **OK**.

Que faire ensuite

Une fois la clé exportée, vous devez la partager avec votre application cible. Voir les instructions appropriées :

- Si vous utilisez un système WebSphere Application Server distinct, voir «Importation de la clé LTPA dans WebSphere Application Server», à la page 132.
- Si vous utilisez un serveur Apache, IHS ou IIS, voir «Configuration de la clé LTPA sur le serveur Web», à la page 135.

Chapitre 12. Configuration d'un plug-in de serveur Web

L'installation du plug-in de serveur Web est requise sur votre serveur Web *uniquement* s'il s'agit d'une instance prise en charge autre que WebSphere Application Server. La principale fonction du module d'extension consiste à extraire les informations d'identité des utilisateurs à parti du cookie LTPA contenu dans une requête Web, afin de mettre ces informations d'identification à la disposition de l'application cible hébergée par le serveur Web, au moyen soit d'en-têtes HTTP, soit de variables de serveur (si celles-ci sont prises en charge par le serveur Web).

Traitement des requêtes Web

Pour vous assurer que vous pouvez configurer correctement le plug-in du serveur Web et intégrer votre application avec le plug-in, il est utile de comprendre la manière dont les requêtes Web sont traitées par le plug-in.

Lorsqu'une requête d'application Web est reçue par le serveur, elle est transmise au module d'extension en vue du traitement, que celui-ci accomplit par les actions suivantes :

1. Extrait l'adresse URL de la requête Web.
2. Extrait de la requête le cookie du jeton LTPA, le cas échéant.
3. Vérifie sa configuration pour déterminer si la fonctionnalité du plug-in est activée. Si ce n'est pas le cas, le traitement prend fin. Si l'activation est effective, les actions suivantes sont exécutées :
   a. Vérifie si l'adresse URL contenue dans la requête correspond à l'une des adresses URL configurées dans le fichier de configuration du plug-in. Cette possibilité permet d'appliquer un traitement spécifique à des applications également spécifiques.
   b. Identifie la liste des en-têtes HTTP à supprimer de la requête. Le fichier de configuration du plug-in identifie les en-têtes HTTP à éliminer et empêche les attaques consistant pour un client à ajouter de "faux" en-têtes.
   c. Le cookie de jeton LTPA est examiné et l'une des actions suivantes est effectuée :
      - Si la requête ne contient pas de cookie de jeton LTPA valide, le module d'extension identifie la liste des cookies de session (s'ils existent) qui doivent être éliminés de la requête, d'après la configuration spécifiée dans le fichier de configuration du plug-in.

        Les cookies sont éliminés uniquement si le cookie de jeton LTPA est manquant, a expiré ou est codé de façon inappropriée. Les cookies de session sont présents uniquement à la suite d'une connexion unique fédérée, qui est indiquée par la présence d'un cookie de jeton LTPA.

        Un cookie de session sans cookie de jeton LTPA valide implique que le cookie de session n'est plus applicable. Le traitement prend fin.
      - Si la requête contient un cookie de jeton LPTA valide, ce cookie est décodé.

        En cas d'échec du décodage, ou si le jeton LTPA arrive à expiration, aucun traitement supplémentaire n'a lieu. La requête est transmise à l'application Web sans qu'aucun en-tête HTTP ne soit ajouté et la gestion de la condition est transférée à l'application.

Si le décodage du jeton LTPA aboutit, le traitement se poursuit et le module d'extension crée une liste des en-têtes HTTP à définir dans le requête. Celui-ci crée une liste à l'appui de la configuration spécifiée dans le fichier de configuration du plug-in et des valeurs d'attributs LTPA contenues dans le jeton. Pour plus d'informations sur l'attribut LTPA et le processus de mappage d'en-tête HTTP, consultez la rubrique «Mappage d'attribut LTPA avec un en-tête HTTP».

Remarque : Les jetons LTPA décodés sont sauvegardés dans une mémoire cache interne jusqu'à épuisement de leur délai d'expiration. Lorsqu'une requête est reçue, le module d'extension recherche la présence d'un jeton valide dans la mémoire cache. Si tel est le cas, celui-ci est réutilisé. Sinon, le jeton est décodé et ajouté en mémoire cache. La taille de la mémoire cache est limitée par le nombre d'entrées qu'elle comporte. Vous pouvez définir sa taille dans le fichier de configuration du plug-in.

d. Au cours de l'étape de traitement finale, le plug-in crée une liste des variables et valeurs du serveur, si celles-ci sont présentes et prises en charge par le serveur Web.

Remarque : L'usage des variables de serveur n'est pas pris en charge dans les environnements IIS.

4. La requête Web terminée est ensuite envoyée à l'application Web pour traitement.

Mappage d'attribut LTPA avec un en-tête HTTP

Pour mapper les informations de cookie LTPA sur un en-tête HTTP, le plug-in utilise un fichier de configuration spécial, itfimwebpi.xml, qui crée, puis modifie ou supprime les en-têtes HTTP dans la requête HTTP finale qui est envoyée à l'application cible.

La figure suivante illustre la façon dont le fichier de configuration est utilisé pour déterminer la requête HTTP finale. Il est à noter que cette figure est valable uniquement à titre d'exemple. Les attributs et en-têtes LTPA sont spécifiques à chaque application qui est exploitée dans un environnement.

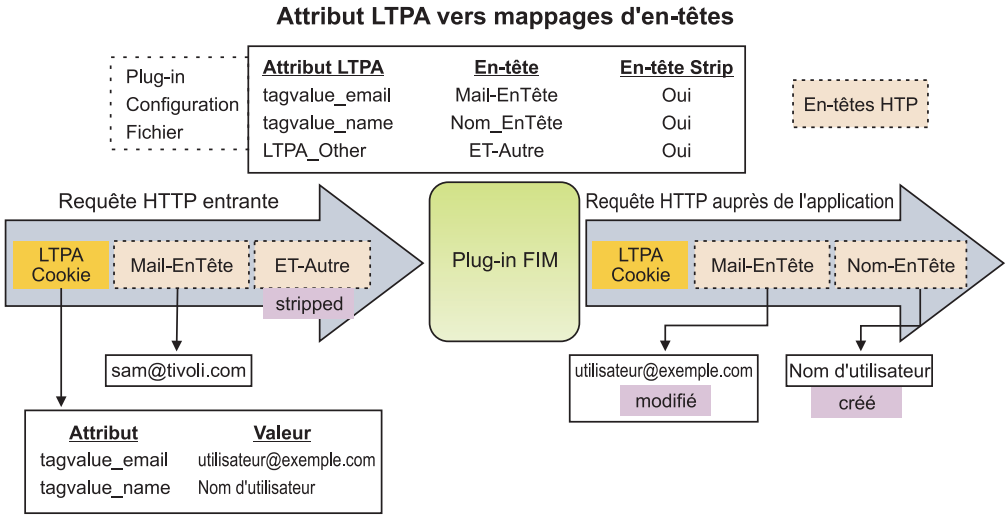


Figure 6. Exemple de mappage entre un attribut LPTA et un en-tête HTTP

1. La requête d'entrée HTTP de la figure précédente contient :
   - Le cookie LTPA créé par le fournisseur de services configuré dans Tivoli Federated Identity Manager
   - deux en-têtes HTTP : 'Header-mail' et 'Other'.
2. Les instructions du fichier de configuration de plug-in pour le mappage des attributs LTPA se présentent comme suit :
   - Attribut LTPA 'tagvalue_email' → Header-mail (éliminé si absent de LTPA)
   - Attribut LTPA 'tagvalue_name' → Header-Name (éliminé si absent de LTPA)
   - Attribut LTPA 'LTPA_Other' → Hdr-Other (éliminé si absent de LTPA)

   Pour tous les en-têtes, si l'attribut LTPA correspondant n'existe pas, tous les en-têtes comportant le nom configuré doivent être éliminés.

   A titre d'exemple, dans la figure, la valeur LTPA 'LTPA_Other' est absente, aussi l'en-tête d'entrée HTTP 'Hdr-Other' est-il éliminé. La valeur LTPA 'tagvalue_email' est présente, donc l'en-tête 'Header_mail' existant est modifié de manière à contenir la valeur issue du cookie LTPA : "user@example.com". La valeur LTPA 'tagvalue_name' est présente, donc l'en-tête 'Header_Name' est créé avec la valeur issue du cookie LTPA : "User_Name".

   Les en-têtes non listés dans le fichier de configuration restent inchangés. Si un cookie LTPA est absent, tous les en-têtes comportant la mention "strip=yes" sont supprimés.

Le plug-in peut également éliminer des cookies si le cookie LTPA n'est pas présenté et mapper des attributs LTPA aux variables de serveur. Toutefois, ces scénarios ne sont pas illustrés sur la figure.

Pour plus d'informations sur la configuration de votre environnement de fournisseur de services, ainsi que sur le fichier de configuration du plug-in, voir «Configuration des composants du fournisseur de services».

Configuration des composants du fournisseur de services

Si vous supposez que le rôle et les composants du fournisseur de service partenaire utilisent WebSphere Application Server comme serveur de point de contact, vous devez réaliser des tâches de configuration particulières avant de pouvoir créer une fédération. Des tâches de configuration spécifiques sont également requises sur le serveur destiné à héberger votre application cible.

Pourquoi et quand exécuter cette tâche

Pour configurer les composants du fournisseur de services, procédez comme suit :

1. Configurez le serveur d'applications destiné à héberger vos applications cible, comme décrit à la rubrique «Configuration de votre serveur Web».
2. Configurez votre application cible comme décrit à la rubrique «Configuration de l'application cible», à la page 139.

Configuration de votre serveur Web

Vous disposez de plusieurs options, suivant le type de serveurs que vous utilisez pour héberger les applications auxquelles les utilisateurs accèdent par le biais d'une connexion unique. Les applications que vous hébergez habituellement sont appelées *applications cible* car elles sont ciblées par la demande de connexion unique.

Pourquoi et quand exécuter cette tâche

Les options applicables aux serveurs de votre environnement Tivoli Federated Identity Manager incluent :

- IBM WebSphere Application Server 5.1, 6.0 ou version supérieure

  Remarque : Les serveurs décrits ici sont généralement dédiés à l'hébergement de l'application cible. Toutefois, il vous est également possible d'héberger votre application cible sur la même instance de WebSphere Application Server que celle sur laquelle vous avez installé le composant d'exécution deTivoli Federated Identity Manager. L'installation de votre composant d'exécution doit avoir été effectuée sur l'une des versions suivantes de WebSphere :

  – WebSphere Application Server version 6.1
  – Version intégrée de WebSphere Application Server 6.1, fournie avec Tivoli Federated Identity Manager
- Microsoft Internet Information Service 6.0
- IBM HTTP Server 6.1
- Apache HTTP Server 2.0 ou 2.2

Lors de la configuration de votre serveur, assurez-vous que l'environnement de Tivoli Federated Identity Manager et le serveur Web se trouvent dans le même domaine DNS, afin de permettre le transfert du cookie LTPA entre ces deux composants.

Pour configurer le serveur Web de sorte qu'il puisse être utilisé avec l'environnement Tivoli Federated Identity Manager, procédez comme suit :

Procédure

1. Sélectionnez et installez un registre d'utilisateurs pour le serveur, comme décrit à la rubrique «Sélection et installation d'un registre d'utilisateurs».
2. Configurez une connexion SSL au registre d'utilisateurs, comme décrit à la rubrique «Configuration du registre d'utilisateurs pour l'application cible», à la page 131.
3. «Configuration d'une connexion SSL au registre d'utilisateurs», à la page 131
4. Si votre application cible est hébergée par une instance de WebSphere Application Server distincte du serveur sur lequel Tivoli Federated Identity Manager est installé, exécutez la procédure indiquée à la rubrique «Configuration d'une instance séparée de WebSphere Application Server pour l'hébergement d'applications», à la page 132.
5. Si vous hébergez une application cible sur un serveur IIS, IHS ou Apache, exécutez la procédure indiquée à la rubrique «Configuration d'un serveur IIS, IHS ou Apache en vue d'héberger l'application», à la page 135.

Sélection et installation d'un registre d'utilisateurs

Un registre d'utilisateurs est requis dans votre environnement à la fois pour votre serveur point de contact et votre serveur d'applications. Les utilisateurs auxquels vous allez fournir les fonctionnalités de connexion unique doivent exister dans les deux registres d'utilisateurs.

Avant de commencer

Dans la plupart des cas, il est souhaitable que votre serveur d'applications exploite le même registre d'utilisateurs que celui que vous avez configuré pour votre

serveur point de contact. Si vous exploitez le même registre d'utilisateurs, assurez-vous qu'il est compatible à la fois avec le serveur point de contact et le serveur d'applications.

Toutefois, si vous exploitez un registre d'utilisateurs distinct, assurez-vous que celui-ci répond aux exigences applicables au serveur hébergeant votre application. Pour plus d'informations, reportez-vous à la documentation de votre serveur.

Si, par exemple, vous hébergez votre application sur WebSphere Application Server, consultez la bibliothèque de WebSphere Application Server et recherchez le centre de documentation de la version que vous utilisez : http://www.ibm.com/software/webservers/appserv/was/library/. Sur le centre de documentation approprié, recherchez les rubriques concernant la configuration d'un registre d'utilisateurs.

Configuration du registre d'utilisateurs pour l'application cible

La configuration du registre d'utilisateurs est une étape importante de la configuration globale.

Avant de commencer

Avant de procéder à cette tâche, vous devez sélectionner le registre d'utilisateurs à utiliser et l'installer selon la procédure décrite dans la section «Sélection et installation d'un registre d'utilisateurs», à la page 130.

Pourquoi et quand exécuter cette tâche

Si vous utilisez le même registre d'utilisateurs que celui que vous avez configuré sur le serveur point de contact, aucune configuration supplémentaire du registre n'est nécessaire. Toutefois, si vous utilisez un registre séparé, créez des utilisateurs auxquels fournir des fonctionnalités de connexion unique. Il peut s'agit ici des mêmes utilisateurs que ceux que vous avez définis dans le registre du serveur point de contact. Consultez la documentation de votre registre d'utilisateurs pour plus d'informations sur l'ajout d'utilisateurs.

Configuration d'une connexion SSL au registre d'utilisateurs

Après avoir configuré votre registre d'utilisateurs, activez SSL pour protéger la connexion entre SSL et le serveur.

Pourquoi et quand exécuter cette tâche

Si vous utilisez le même registre d'utilisateurs pour votre serveur point de contact que pour le serveur d'applications, il est possible que vous ayez déjà effectué cette tâche. Si vous utilisez un registre d'utilisateurs séparé, reportez-vous à la documentation relative à ce registre d'utilisateurs pour plus d'informations sur la configuration SSL.

Que faire ensuite

Une fois que vous avez configuré SSL, poursuivez avec les étapes appropriées pour le serveur sur lequel seront hébergées vos applications cible :

- «Configuration d'une instance séparée de WebSphere Application Server pour l'hébergement d'applications», à la page 132

- «Configuration d'un serveur IIS, IHS ou Apache en vue d'héberger l'application», à la page 135

Configuration d'une instance séparée de WebSphere Application Server pour l'hébergement d'applications

Dans un environnement Tivoli Federated Identity Manager, vous pouvez héberger vos applications cible sur la même instance de WebSphere Application Server que celle utilisée comme serveur point de contact ou sur une instance distincte de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Pour configurer une instance distincte de WebSphere Application Server afin qu'elle puisse héberger les applications hôtes auxquelles accèdent les utilisateurs par le biais d'une connexion unique, procédez comme suit :

Procédure

1. Importez la clé LTPA depuis votre serveur point de contact WebSphere Application Server, comme indiqué à la rubrique «Importation de la clé LTPA dans WebSphere Application Server».
2. Désactivez la génération automatique des clés LTPA, comme décrit à la rubrique «Désactivation de la génération automatique d'une clé LTPA», à la page 133.
3. Configurez WebSphere Application Server pour utiliser le registre d'utilisateurs, comme indiqué à la rubrique «Configuration de WebSphere pour l'utilisation du registre d'utilisateurs», à la page 133.

Importation de la clé LTPA dans WebSphere Application Server

Si votre application cible est hébergée sur un serveur WebSphere Application Server distinct de votre serveur point de contact WebSphere, vous devez importer la clé LTPA du serveur point de contact vers votre serveur d'applications cible.

Avant de commencer

Vérifiez que vous avez réalisé les tâches ci-dessous :

- Vérifiez que l'heure des serveurs est synchronisée.
- Copiez les clés LTPA de l'emplacement où vous les aviez exportées vers un emplacement de votre serveur d'applications cible.
- Procurez-vous le mot de passe des clés LTPA. Un mot de passe a été associé aux clés lors de leur exportation du serveur point de contact WebSphere.

Procédure

1. Connectez-vous à la console du *serveur d'applications cible*. Ne vous connectez pas à votre console Tivoli Federated Identity Manager pour exécuter cette procédure.
2. Sélectionnez **Sécurité** > **Administration, applications et infrastructure sécurisées** > **Mécanismes d'authentification et expiration**.
3. Dans les zones **Mot de passe** et **Confirmer**, entrez le mot de passe utilisé pour chiffrer les clés LTPA. Ce mot de passe doit correspondre à celui qui a été utilisé lors de l'exportation des clés.
4. Dans la zone **Nom complet du fichier de clés**, indiquez le chemin d'accès complet de l'emplacement des clés LTPA. Vous devez posséder un droit d'accès en écriture à ce fichier.

5. Cliquez sur **Importer les clés** pour importer les clés.
6. Cliquez sur **OK**.
7. Cliquez sur **Sauvegarder** pour importer les modifications dans la configuration principale.

Que faire ensuite

Désactivez la génération automatique des clés LTPA, comme décrit à la rubrique «Désactivation de la génération automatique d'une clé LTPA».

Désactivation de la génération automatique d'une clé LTPA

Par défaut, WebSphere Application Server génère automatiquement une clé LTPA. Toutefois, si vous utilisez un serveur WebSphere Application Server autre que votre serveur point de contact pour héberger votre application cible, vous devez utiliser la clé LTPA de votre serveur point de contact sur votre serveur d'applications. Vous devez donc désactiver la génération automatique de clé afin d'éviter tout conflit.

Avant de commencer

Pour exécuter cette tâche, vous devez connaître le nom du jeu de clés et la portée de la gestion dans laquelle ce jeu a été défini.

Procédure

1. Connectez-vous à la console du *serveur d'applications cible*. Ne vous connectez pas à votre console Tivoli Federated Identity Manager pour exécuter cette procédure.
2. Cliquez sur **Sécurité** > **Certificat SSL et gestion des clés** > **Gérer les configurations de sécurité des noeuds finals**.
3. Développez l'arborescence de la portée de gestion entrante et sortante qui contient le jeu de clés, puis cliquez sur le lien de la portée.
4. Sous l'option Articles liés, cliquez sur **Jeux de clés**.
5. Cliquez sur le jeu de clés que vous souhaitez désactiver.
6. Effacez la case à cocher **Générer les clés automatiquement**.
7. Cliquez sur **OK**.
8. Cliquez sur **Sauvegarder** pour importer les modifications dans la configuration principale.

Que faire ensuite

Poursuivez avec les étapes indiquées à la section «Configuration de WebSphere pour l'utilisation du registre d'utilisateurs».

Configuration de WebSphere pour l'utilisation du registre d'utilisateurs

Vérifiez que le serveur WebSphere Application Server que vous utilisez pour héberger votre application cible est configuré pour utiliser le registre d'utilisateurs sélectionné et installé.

Avant de commencer

Consultez les informations de la rubrique «Sélection et installation du registre d'utilisateurs», à la page 103. Vérifiez que vous avez sélectionné et installé l'option de registre d'utilisateurs appropriée pour votre environnement.

Pourquoi et quand exécuter cette tâche

Pour permettre à WebSphere d'utiliser votre registre d'utilisateurs, procédez comme suit :

Procédure

1. Connectez-vous à la console *de votre application cible*. Ne vous connectez pas à votre console Tivoli Federated Identity Manager pour exécuter cette procédure.
2. Sélectionnez **Sécurité** > **Administration, application et infrastructure sécurisées**. L'onglet Configuration s'affiche.
3. Cliquez sur **Assistant de configuration des paramètres de sécurité** pour modifier le registre d'utilisateurs utilisé par le composant d'exécution WebSphere. Le panneau **Spécifier l'étendue de la protection** apparaît.
4. Vérifiez que la case **Activer la sécurité des applications** est cochée.
5. Cliquez sur **Suivant**. Le panneau **Sécuriser l'environnement de traitement des applications** apparaît.
6. Sélectionnez l'option correspondant au registre d'utilisateurs de votre choix :
   - **Référentiels fédérés**
   - **Registre LDAP autonome**
   - **Système d'exploitation local**
   - **Registre personnalisé autonome**
7. Cliquez sur **Suivant**. Le panneau **Configurer le référentiel d'utilisateurs** s'affiche.
8. Indiquez des valeurs pour chaque paramètre de configuration du registre. Pour obtenir une description des zones présentées, consultez l'aide en ligne.
9. Cliquez sur **Suivant** et quittez l'assistant.
10. Sauvegardez les modifications apportées à votre configuration.
11. Arrêtez WebSphere Application Server.
12. Redémarrez WebSphere Application Server. Vous devez utiliser le nom d'administrateur que vous avez choisi pour vous connecter et effectuer ces modifications.
13. Dans la console, sélectionnez **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Propriétés du domaine**.
14. Dans la section Sécurité WebSphere du panneau, mettez à jour les valeurs suivantes :

    Nom de l'utilisateur d'administration
    Remplacez l'entrée existante par le nom de compte administrateur LDAP entré à l'étape précédente. Par exemple, `ldapadmin`

    Mot de passe d'administration
    Entrez le mot de passe de l'administrateur LDAP.
15. Sauvegardez les modifications.
16. Arrêtez WebSphere Application Server.
17. Redémarrez WebSphere Application Server.

Que faire ensuite

Une fois terminé, poursuivez en sélectionnant l'étape appropriée selon votre environnement:

- Si vous hébergez des applications sur un serveur IHS, IIS ou Apache, passez à l'étape «Configuration d'un serveur IIS, IHS ou Apache en vue d'héberger l'application».
- Si vous hébergez des applications uniquement sur votre serveur WebSphere, la configuration est terminée. Poursuivez vers l'étape de configuration des applications cible à la rubrique «Configuration de l'application cible», à la page 139.

Configuration d'un serveur IIS, IHS ou Apache en vue d'héberger l'application

Si vos applications cible sont destinées à être hébergées sur un serveur Microsoft Internet Information Services, une instance IBM HTTP Server ou une instance Apache HTTP Server, vous devez accomplir des tâches de configuration spécifiques.

Avant de commencer

Avant de poursuivre ces tâches, vous devez avoir procédé à l'installation du module d'extension (plug-in).

Assurez-vous que :

- Le plug-in est installé sur le serveur qui héberge l'application cible.
- Le serveur se trouve dans le même domaine que le serveur Tivoli Federated Identity Manager.

Puis, assurez-vous d'avoir exécuté les étapes de la section «Configuration de votre serveur Web», à la page 129, notamment :

- «Sélection et installation d'un registre d'utilisateurs», à la page 130
- «Configuration du registre d'utilisateurs pour l'application cible», à la page 131
- «Configuration d'une connexion SSL au registre d'utilisateurs», à la page 131

Pourquoi et quand exécuter cette tâche

Pour préparer votre environnement de plug-in, procédez comme suit :

Procédure

1. Copiez la clé LTPA sur votre serveur, en suivant la procédure décrite dans «Configuration de la clé LTPA sur le serveur Web».
2. Créez le fichier de configuration du plug-in, selon la procédure décrite dans la section «Création du fichier de configuration de plug-ins», à la page 136.
3. Copiez le fichier de configuration du plug-in sur le serveur, en suivant la procédure décrite dans «Copie de la configuration d'un plug-in sur le serveur», à la page 137.

Configuration de la clé LTPA sur le serveur Web

La clé LTPA utilisée par WebSphere Application Server sur votre serveur point de contact doit être partagée par le serveur sur lequel le plug-in est installé.

Avant de commencer

Vérifiez que vous avez réalisé les tâches ci-dessous :

- Installé le module d'extension sur le serveur Web.

- Complété la configuration de votre serveur point de contact comme décrit à la rubrique «Configuration d'un serveur point de contact WebSphere Application Server (fournisseur de services)», à la page 119.
- Exporté les clés LTPA depuis votre serveur point de contact, en suivant la procédure décrite dans «Exportation de la clé LTPA à partir du serveur point de contact», à la page 124.
- Vérifié la synchronisation de la date et de l'heure du serveur point de contact et du serveur sur lequel vous copiez la clé LTPA.

Procédure

1. Copiez la clé LTPA, normalement nommée `ltpa.keys`, à partir de l'emplacement dans lequel vous l'avez exportée.
2. Collez-la dans le répertoire webpi du serveur d'applications. Par exemple :

   Sur un serveur IHS ou Apache :
   ```
   /opt/IBM/FIM/webpi/etc
   ```

   Sur un serveur IIS :
   ```
   C:\Program Files\IBM\FIM\webpi\etc
   ```

Que faire ensuite

Passez à l'étape «Création du fichier de configuration de plug-ins».

Création du fichier de configuration de plug-ins

Après avoir installé le module d'extension et préparé votre environnement pour l'utilisation de celui-ci, vous devez le configurer au moyen des informations spécifiques relatives aux applications Web auxquelles les utilisateurs accéderont par le biais d'une connexion unique.

Avant de commencer

Pour exécuter cette tâche, vous devez disposer des informations suivantes :

- Le mot de passe utilisé pour chiffrer la clé en vue de son exportation.
- Le nom et l'adresse URL de chaque application cible hébergée par ce serveur.
- Les mappages corrects de l'en-tête HTTP et de l'attribut LTPA de votre environnement. Vous devez savoir quel attribut LTPA vous souhaitez mapper avec quel en-tête HTTP ou quelle variable de serveur. L'en-tête HTTP et les variables de serveur correspondent aux valeurs requises par l'application cible.
- La liste des cookies à supprimer si le cookie LTPA est absent ou non valide, ce qui indique généralement qu'il ne s'agit pas d'un utilisateur fédéré disposant d'une connexion unique.
- La liste des mappages entre les noms de variables de serveur et des noms d'attributs de jeton LTPA. Les variables de serveur peuvent être utilisées à la place des en-têtes HTTP pour présenter les attributs LTPA à l'application.

  Remarque : Le plug-in IIS ne prend pas en charge l'utilisation de variables de serveur.

Pour plus d'informations sur l'en-tête HTTP et les mappages d'attributs LTPA, ainsi que le mode de fonctionnement du plug-in dans l'environnement, voir Chapitre 12, «Configuration d'un plug-in de serveur Web», à la page 127.

Procédure

1. Connectez-vous à la console.

2. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Configuration du plug-in de serveur Web**. Le panneau Configuration de la connexion unique du plug-in de serveur Web s'ouvre.
3. Entrez les informations requises pour votre serveur dans les sections **Configuration de la connexion unique du plug-in de serveur Web** et **Configuration de la consignation du plug-in de serveur Web**. Pour obtenir une description de chaque zone, consultez l'aide en ligne.

   Remarque : Assurez-vous que le mot de passe LTPA spécifié dans la zone correspond à celui que vous avez créé lors de l'exportation du fichier ltpa.keys.
4. ne fois que vous avez renseigné toutes les zones, cliquez sur **Enregistrer**.
5. Dans **Configuration des applications du plug-in de serveur Web**, définissez une application pour la configuration de connexion unique en cliquant sur **Créer**. Le panneau Propriétés de l'application s'ouvre.
   a. Entrez les informations relatives à l'application à laquelle vos utilisateurs disposant d'une connexion unique peuvent accéder.
   b. Cliquez sur **Appliquer**.
   c. Cliquez sur **Mappages de l'en-tête HTTP à l'attribut LTPA**.
   d. Pour accepter les paramètres par défaut, cliquez sur **Appliquer**. Pour les modifier, cliquez sur **Créer**.
   e. Une fois les zones de ce panneau remplies, cliquez sur **Appliquer**.
   f. Cliquez sur **Cookies client à supprimer**.
   g. Pour accepter les paramètres par défaut, cliquez sur **Appliquer**. Pour les modifier, cliquez sur **Créer**.
   h. Une fois les zones de ce panneau remplies, cliquez sur **Appliquer**.
   i. Cliquez sur **Mappages des variables de serveur à l'attribut LTPA**.
   j. Pour accepter les paramètres par défaut, cliquez sur **Appliquer**. Pour les modifier, cliquez sur **Créer**.
   k. Une fois les zones de ce panneau remplies, exécutez l'une des actions ci-dessous :
      - Pour ajouter d'autres applications, cliquez sur **Appliquer**, puis répétez la procédure précédente pour chaque nouvelle application.
      - Si vous avez terminé l'ajout de l'application au serveur, cliquez sur **OK**.
6. Cliquez sur **Sauvegarder**.
7. Cliquez sur **Exporter le fichier de configuration du plug-in de serveur Web**. Ensuite, procédez comme suit :
   a. Cliquez sur **Enregistrer** dans la fenêtre en incrustation pour sauvegarder la configuration dans un fichier nommé `itfimwebpi.xml`.
   b. Sélectionnez le répertoire d'installation du plug-in de serveur Web. Par exemple, enregistrez le fichier `itfimwebpi.xml` sous `/opt/IBM/FIM/webpi/etc`.

Que faire ensuite

Passez à l'étape «Copie de la configuration d'un plug-in sur le serveur».

Copie de la configuration d'un plug-in sur le serveur

Après avoir créé le fichier de configuration d'un plug-in, vous devez copier cette configuration sur votre serveur Web.

Procédure

1. Recherchez le fichier de configuration que vous avez créé à l'aide de la procédure décrite dans «Création du fichier de configuration de plug-ins», à la page 136. Le fichier, qui porte le nom `itfimwebpi.xml`, est créé dans le répertoire que vous avez spécifié lors de l'exportation du fichier.
2. Copiez le fichier, puis collez-le dans le répertoire webpi de votre serveur Web :

   Sur un serveur IHS ou Apache :
   ```
   /opt/IBM/FIM/webpi/etc
   ```

   Sur un serveur IIS :
   ```
   C:\Program Files\IBM\FIM\webpi\etc
   ```
3. Redémarrez votre serveur Web pour que les modifications soient prises en compte.

Que faire ensuite

La configuration de votre serveur est terminée. Poursuivez vers l'étape de configuration des applications cible à la rubrique «Configuration de l'application cible», à la page 139.

Vérification de la configuration du module d'extension sur Apache ou IBM HTTP Server

Après avoir configuré le plug-in sur un serveur HTTP Apache ou une instance IBM HTTP Server, vous pouvez vérifier que la configuration a abouti.

Avant de commencer

Avant de poursuivre cette tâche, assurez-vous que vous avez effectué la tâche ci-après :

- «Configuration d'un serveur point de contact WebSphere Application Server (fournisseur de services)», à la page 119
- «Configuration d'un serveur IIS, IHS ou Apache en vue d'héberger l'application», à la page 135

Procédure

1. Sur le serveur, localisez le fichier httpd.conf. L'emplacement de ce fichier dépend de votre installation. Par exemple :

   ```
   /etc/httpd/conf/httpd.conf
   ```
2. Ouvrez le fichier dans un éditeur de texte et recherchez la ligne relative au module d'extension que vous utilisez :

   Apache HTTP Server 2.2 :

   ```
   LoadModule fimwebpi_module /opt/IBM/FIM/webpi/lib/libitfimwebpi-apache22.so
   ```

   Apache HTTP Server 2.0 ou IBM HTTP Server :

   ```
   LoadModule fimwebpi_module /opt/IBM/FIM/webpi/lib/libitfimwebpi-apache20.so
   ```

   Assurez-vous que le module webpi (`libitfimwebpi-apache22.so` ou `libitfimwebpi-apache20.so`) peut accéder en écriture au chemin d'accès du fichier journal définie dans le fichier de configuration de votre plug-in (`itfimwebpi.xml`).

Que faire ensuite

Poursuivez avec les étapes indiquées à la section «Configuration de l'application cible», à la page 139.

Configuration de l'application cible

En tant que fournisseur de services, votre rôle dans la fédération consiste à fournir un service, tel qu'une application Web, à l'utilisateur final.

Pourquoi et quand exécuter cette tâche

Dans le cadre de ce rôle, assurez-vous que l'application (appelée *application cible*) que vous fournissez aux utilisateurs est configurée de façon appropriée pour fonctionner dans l'environnement Tivoli Federated Identity Manager :

- L'application doit être capable d'accepter les informations sur l'identité des utilisateurs au moyen d'en-têtes HTTP ou de variables de serveur.
- L'environnement Tivoli Federated Identity Manager et l'application doivent se trouver dans le même domaine DNS.
- L'application doit être hébergée par un serveur Web pris en charge tel que :
  - Serveur Microsoft Internet Information Services (IIS) 6.0, avec module d'extension Tivoli Federated Identity Manager installé
  - IBM HTTP Server 6.1, avec plug-in Tivoli Federated Identity Manager installé
  - Serveur Apache HTTP 2.0 ou 2.2 avec plug-in Tivoli Federated Identity Manager installé
  - WebSphere Application Server version 5.1
  - WebSphere Application Server version 6.0 ou supérieure

    Remarque : Vous pouvez également définir, en tant qu'hôte des applications cibles, la même instance de WebSphere Application Server que celle sur laquelle vous avez installé le composant d'exécution de Tivoli Federated Identity Manager. Cette version de WebSphere Application Server peut être l'une des suivantes :

    - WebSphere Application Server version 6.1 avec fix pack 15
    - Version intégrée de WebSphere Application Server, fournie avec Tivoli Federated Identity Manager

Pour plus d'informations sur la configuration de votre application cible, reportez-vous à la documentation relative au serveur destiné à héberger l'application. Si, par exemple, vous hébergez l'application cible sur WebSphere Application Server, reportez-vous au centre de documentation relatif à la version de WebSphere Application Server que vous utilisez, en consultation la bibliothèque du site http://www.ibm.com/software/webservers/appserv/was/library/.

Configuration de la connexion pour votre application

Avant d'utiliser Tivoli Federated Identity Manager, vous avez certainement utilisé une méthode de connexion spécifique à votre application. Par exemple, il est possible que vous ayez fourni à vos utilisateurs une adresse URL qui leur a permis d'accéder à un formulaire de connexion ou que vous ayez demandé l'authentification client. Dans votre environnement Tivoli Federated Identity Manager, votre fournisseur d'identité partenaire est responsable de l'authentification des utilisateurs. Selon la configuration de votre fédération, il peut être nécessaire d'acheminer vos utilisateurs vers une nouvelle adresse URL (comme celle hébergée par votre fournisseur d'identité partenaire) ou de les réacheminer à partir de votre site vers la méthode de connexion appropriée utilisée par votre fournisseur d'identité partenaire.

Pourquoi et quand exécuter cette tâche

Discutez avec votre fournisseur d'identité partenaire de la configuration de connexion requise. Assurez-vous ensuite que votre environnement est configuré de sorte que les utilisateurs soient envoyés à l'emplacement de connexion approprié.

Instructions destinées aux utilisateurs pour l'activation des cookies

Les utilisateurs doivent activer des cookies dans leurs navigateurs lors de l'utilisation de la connexion unique à un fournisseur de services qui utilise WebSphere Application Server comme serveur point de contact.

Pourquoi et quand exécuter cette tâche

Conseillez aux utilisateurs de suivre les instructions relatives à l'activation des cookies pour leurs navigateurs.

Chapitre 13. Configuration de la base de données de service d'alias

SAML 2.0 prend en charge l'usage d'identificateurs (ou alias) pour la communication des identités d'utilisateurs entre les partenaires. Les alias ont pour rôle d'élever le niveau de confidentialité dont bénéficie un utilisateur lorsqu'il accède aux ressources d'un fournisseur de services. Lorsque des alias sont utilisés, un identificateur reconnu à la fois par le fournisseur d'identité et le fournisseur de services est envoyé à la place du nom de compte effectif de l'utilisateur. La création et l'enregistrement des alias a lieu durant l'opération d'association des comptes (fédération). Une fois l'association de compte effectuée, l'alias est inscrit dans tous les messages échangés entre les partenaires. Un alias différent est utilisé pour chaque partenaire. L'alias utilisé dans une direction, par exemple du fournisseur d'identité vers le fournisseur de services, peut être différent de celui qui est utilisé dans la direction opposée, du fournisseur de services vers le fournisseur d'identité.

Pourquoi et quand exécuter cette tâche

Remarque : L'utilisation des alias est optionnelle dans SAML 2.0.

Dans la configuration par défaut du service d'alias, les ID utilisés sont persistants.

Un service de Tivoli Federated Identity Manager, appelé *service d'alias*, génère de nouveaux alias, associe des alias avec des utilisateurs locaux et effectue des mappages bidirectionnels entre alias et utilisateurs.

La plupart des alias sont persistants et doivent être conservés pendant une durée prolongée. L'utilisation de certains types de base de données est donc nécessaire pour les stocker. Vous avez deux options pour le type de base de données à utiliser :

- Une base de données JDBC, telle que la base de données Derby de WebSphere Application Server
- Une base de données LDAP, telle qu'IBM Tivoli Directory Server (composant fourni séparément)

Les tâches de configuration de la base de données de service d'alias varient selon que vous avez installé la version imbriquée de WebSphere Application Server ou que vous utilisez une version existante de WebSphere Application Server avec votre installation du composant des services d'exécution et de gestion de Tivoli Federated Identity Manager.

Version intégrée de WebSphere

Les options pour la base de données sont les suivantes :

- **Base de données JDBC**

  Si vous avez installé la version imbriquée de WebSphere Application Server, une base de données JDBC, Cloudscape 10, également appelée Derby, est configurée sur WebSphere Application Server pour stocker les informations relatives à l'alias. Aucune configuration supplémentaire de la base de données n'est nécessaire.

- **Base de données LDAP**

  Vous avez la possibilité d'utiliser une base de données LDAP, telle que IBM Tivoli Directory Server, que vous avez achetée, installée et

configurée séparément de Tivoli Federated Identity Manager. Voir les informations de la rubrique «Configuration d'une base de données de service d'alias LDAP», à la page 144. Puis, pour pouvoir utiliser la base de données LDAP avec Tivoli Federated Identity Manager, vous devez modifier les paramètres du service d'alias comme décrit à la section «Modification des paramètres du service d'alias», à la page 144.

Version existante de WebSphere Application Server

Les options pour la base de données sont les suivantes :

- **base de données JDBC**

  Si vous avez installé Tivoli Federated Identity Manager sur une version existante de WebSphere Application Server et que vous souhaitez exploiter une base de données JDBC, vous devez créer et configurer manuellement la base de données en suivant une procédure similaire à celle décrite ci-dessous pour Cloudscape 10 (Derby). Voir la rubrique «Configuration d'une base de données d'alias JDBC». (comme indiqué précédemment, si vous avez installé la version intégrée de WebSphere Application Server, ces étapes ont déjà été accomplies automatiquement).

- **Base de données LDAP**

  Vous avez la possibilité d'utiliser une base de données LDAP, telle qu'IBM Tivoli Directory Server, que vous avez achetée, installée et configurée séparément de votre installation de Tivoli Federated Identity Manager. Voir les informations de la rubrique «Configuration d'une base de données de service d'alias LDAP», à la page 144. Puis, pour pouvoir utiliser la base de données LDAP avec Tivoli Federated Identity Manager, vous devez modifier les paramètres du service d'alias comme décrit à la section «Modification des paramètres du service d'alias», à la page 144.

Configuration d'une base de données d'alias JDBC

Si vous avez installé Tivoli Federated Identity Manager sur une version existante de WebSphere Application Server et que vous souhaitez utiliser une base de données JDBC, vous devez manuellement créer et configurer la base de données. Si vous avez installé la version intégrée de WebSphere Application Server, ces étapes ont déjà été exécutées automatiquement.

Pourquoi et quand exécuter cette tâche

Les instructions suivantes décrivent comment créer et utiliser la base de données JDBC Derby fournie avec WebSphere Application Server. La création de la base de données Derby s'effectue via un outil Apache appelé "ij". Celui-ci est mis en oeuvre avec la classe Java org.apache.derby.tools.ij.

Procédure

1. Créez la base de données FIMAliases et importez le schéma :
   a. Ouvrez une invite de commande et démarrez l'outil ij se trouvant dans le répertoire `/derby/bin/embedded` dans lequel vous avez installé WebSphere Application Server.

      Sous AIX, Linux ou Solaris

      Entrez *$was_home*`/derby/bin/embedded/ij.sh`.

      Sous Windows

      Entrez *$was_home*`/derby/bin/embedded/ij.bat`.

b. A partir de la ligne de commande ij, créez la base de données et le schéma en exécutant les commandes suivantes :

```
connect 'jdbc:derby:FIMAliases;create=true';
run '/opt/IBM/FIM/etc/Table.ddl';
quit;
```

Remarque : Le fichier `Table.ddl` se trouve dans le répertoire d'installation de Tivoli Federated Identity Manager. Si vous avez défini un répertoire d'installation différent, spécifiez-le lors de l'exécution de la commande. Sous Windows, le répertoire d'installation par défaut est `C:\Program Files\IBM\FIM`.

2. Vérifiez la base de données et le schéma :
   a. Ouvrez une invite de commande et accédez au répertoire *$was_home*/derby/FIMAliases.
   b. Vérifiez que le fichier SQL de sortie contient le schéma FIMAliases.

      Sous AIX, Linux ou Solaris
      Entrez *$was_home*`/derby/bin/embedded/dblook.sh -d jdbc:derby:FIMAliases -o FIMAliase.sql`.

      Sous Windows
      Entrez *$was_home*`/derby/bin/embedded/dblook.bat -d jdbc:derby:FIMAliases -o FIMAliases.sql`.

3. Créez le fournisseur JDBC Derby intégré et la source de données associée :
   a. Ouvrez la console WebSphere.
   b. Cliquez sur **Ressources** > **JDBC** > **Fournisseurs JDBC**.
   c. Définissez la porté sur Cell, par exemple *Cell=myhostNode01Cell*, dans la liste de paramètres **Portée**.
   d. Cliquez sur **Nouveau**.
   e. Complétez les zones requises comme suit :

      Type de base de données
      Sélectionnez **Derby**.

      Type de fournisseur
      Sélectionnez **Fournisseur JDBC Derby**.

      Type d'implémentation
      Sélectionnez **Source de données du pool de connexions**.

      Nom Spécifiez le nom du fournisseur JDBC du service d'alias pour Tivoli Federated Identity Manager.

      Spécifiez par exemple : ITFIM Alias Service JDBC Provider.

   f. Cliquez sur **Suivant**.
   g. Cliquez sur **Terminer**.
4. Créez une source de données pour ce fournisseur JDBC :
   a. Dans la console d'administration WebSphere, cliquez sur **Ressources** > **JDBC** > **Fournisseurs JDBC** > **ITFIM Alias Service JDBC Provider** > **Sources de données** > **Nouveau**.
   b. Complétez les zones requises comme suit :

      Nom de la source de base de données
      Entrez un nom identifiant la source de données, tel que `ITFIM Alias Service Datasource`.

Nom JNDI
Entrez `jdbc/IdServiceJdbc`.

Avertissement : Spécifiez ce nom exactement comme indiqué, afin que les mappages entre le service d'alias et la source de données s'établissent automatiquement.

c. Cliquez sur **Suivant**.
d. Indiquez un nom pour la base de données, tel que `FIMAliases`.
e. Cliquez sur **Suivant**.
f. Cliquez sur **Terminer**.

5. Pour vérifier la connexion à la base de données, sélectionnez la source de données que vous avez configurée et cliquez sur **Test de la connexion**.

Modification des paramètres du service d'alias

Vous pouvez modifier le paramètre de votre base de données des identificateurs dans la console ISC (Integrated Solutions Console).

Pourquoi et quand exécuter cette tâche

Pour modifier la configuration de la base de données de l'identificateur de noms, procédez comme suit :

Procédure

1. Connectez-vous à la console. Le portlet Paramètres du service d'alias s'ouvre.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Paramètres du service d'alias**.
3. Sélectionnez **Fournisseur et source de données JDBC**
   Utilisez cette option si vous envisagez de stocker les informations relatives à l'identificateur de nom dans une base de données JDBC.
4. Cliquez sur **Appliquer**.
5. Cliquez sur **OK**.

Configuration d'une base de données de service d'alias LDAP

Si vous installez Tivoli Federated Identity Manager avec la version intégrée de WebSphere, une base de données JDBC constitue l'option de base de données par défaut pour le service d'alias dans Tivoli Federated Identity Manager. Toutefois, vous pouvez la remplacer par une base de données LDAP.

Avant de commencer

Si vous installez Tivoli Federated Identity Manager avec un déploiement WebSphere existant, il se peut qu'une base de données LDAP soit déjà utilisée en tant que registre d'utilisateurs. Lors de l'utilisation de WebSEAL en tant que serveur point de contact, l'installation a lieu dans un environnement qui inclut Tivoli Access Manager. Le déploiement LDAP effectué le plus couramment avec Tivoli Access Manager est IBM Tivoli Directory Server.

Le service d'alias Tivoli Federated Identity Manager stocke les informations relatives aux alias dans un registre d'utilisateurs. Le service d'alias prend en charge les registres d'utilisateurs suivants :

- IBM Tivoli Directory Server

- Sun ONE

Remarque :

Vous pouvez développer votre propre service d'alias en vue de l'utiliser avec d'autres registres tels que Lotus Domino ou Microsoft Active Directory.

Le service d'alias nécessite un emplacement dans LDAP pour le stockage des informations requises et la fonction Tivoli Federated Identity Manager services d'exécution et de gestion a besoin d'un compte sur le serveur LDAP dans lequel rechercher les informations d'alias.

Si aucune base de données LDAP n'est encore installée sur votre système, vous devez installer un logiciel pour pouvoir utiliser le service d'alias.

Si vous avez besoin de LDAP, vous pouvez utiliser le produit IBM Tivoli Directory Server, qui peut être téléchargé sur le site http://www-306.ibm.com/software/tivoli/resource-center/security/code-directory-server.jsp.

Pourquoi et quand exécuter cette tâche

Si vous utilisez une base de données LDAP, les tâches de configuration suivantes sont requises :

- «Utilisation de tfimcfg pour configurer LDAP dans le service d'alias»

  Tivoli Federated Identity Manager est doté d'un utilitaire permettant d'automatiser ce processus lors de l'utilisation avec IBM Tivoli Directory Server ou Sun ONE Directory Server.Configuration du registre d'utilisateurs LDAP pour le service d'alias
- «Création d'un suffixe LDAP», à la page 149
- «Modification des paramètres du service d'alias», à la page 144

Utilisation de tfimcfg pour configurer LDAP dans le service d'alias

L'utilitaire tfimcfg vous permet d'automatiser la configuration LDAP pour le service d'alias.

Pourquoi et quand exécuter cette tâche

Ce guide d'installation vous explique comment exécuter l'utilitaire **tfimcfg** afin de configurer le service d'alias.

L'utilitaire tfimcfg fait appel à un fichier de données intitulé ldapconfig.properties pour sélectionner les actions à entreprendre. Vous pouvez modifier le comportement de l'utilitaire tfimcfg en éditant les valeurs contenues dans ce fichier via un éditeur de texte. Il vous est possible de spécifier si des ensembles de propriétés LDAP doivent ou non être définis. Pour chaque ensemble que vous créez, vous pouvez spécifier les valeurs des propriétés individuelles.

Pour permettre à l'utilitaire tfimcfg de configurer le protocole LDAP par voie de programme, vous devez lui fournir certaines informations relatives à LDAP, telles que le nom d'hôte et le numéro de port LDAP, ainsi que les informations relatives au compte de l'administrateur. Le fichier ldapconfig.properties contient des entrées

pour chacune de ces propriétés. Des valeurs par défaut sont fournies. Vous devez modifier ces valeurs afin de les adapter aux exigences de votre environnement de déploiement.

La procédure suivante permet de dresser la liste des propriétés pour lesquelles il convient de définir une valeur.

Procédure

1. Obtenez une copie du fichier ldapconfig.properties. Vous pouvez visualiser le contenu du fichier de l'une des manières suivantes :
   - En affichant la liste de fichiers par défaut dans Annexe A, «Référence de tfimcfg», à la page 827
   - En accédant au logiciel d'installation (CD ou répertoire d'installation) et en affichant le fichier par défaut :

     AIX, Solaris ou Linux

     ```
     /opt/IBM/FIM/tools/tamcfg/ldapconfig.properties
     ```

     Windows

     ```
     C:\Progra~1\IBM\FIM\tools\tamcfg\ldapconfig.properties
     ```
2. Indiquez si tfimcfg doit ajouter des suffixes au serveur LDAP, le cas échéant.

   Valeur par défaut :

   ```
   ldap.suffix.add=true
   ```

   L'utilitaire tfimcfg ajoute un certain nombre de suffixes, sur la base d'autres paramètres contenus dans le fichier ldapconfig.properties. Pour ignorer la création des suffixes, définissez la valeur 'false'.

   Les suffixes pouvant être créés sont les suivants :
   - Un suffixe de hiérarchie permettant de conserver les informations du service d'alias (alias d'identité des utilisateurs)

     Valeur par défaut : cn=itfim
   - Un suffixe utilisé par les serveurs Tivoli Access Manager

     Valeur par défaut : secAuthority=Default
   - Un suffixe de hiérarchie pour le stockage des informations relatives aux personnes et aux groupes

     Valeur par défaut : dc=com
3. Indiquez si l'utilitaire tfimcfg doit créer des conteneurs LDAP pour stocker les utilisateurs et groupes Tivoli Federated Identity Manager.

   Les utilisateurs et groupes Tivoli Federated Identity Manager sont les suivants :

   Valeur par défaut :

   ```
   ldap.suffix.user.configuration=true
   ldap.organization.configuration=true
   ```

   - Lorsque `ldap.suffix.user.configuration=true`, tfimcfg ajoute un suffixe LDAP `dc=com` et crée un objet associé. L'utilitaire définit également des propriétés complémentaires telles que spécifiées dans le fichier ldapconfig.properties. La liste des propriétés, ainsi que leurs valeurs par défaut, est la suivante :

     ```
     ldap.suffix.user.dn=dc=com
     ldap.suffix.user.name=com
     ldap.suffix.user.attributes=dc
     ldap.suffix.user.objectclasses=domain
     ```

- Lorsque `ldap.organization.configuration=true`, tfimcfg définit des propriétés complémentaires. Ces propriétés sont spécifiées dans le fichier ldapconfig.properties. La liste des propriétés, ainsi que leurs valeurs par défaut, est la suivante :

  ```
  ldap.user.container.dn=cn=users,dc=exemple,dc=com
  ldap.group.container.dn=cn=groups,dc=exemple,dc=com
  ldap.organization.dn=dc=exemple,dc=com
  ldap.organization.name=exemple
  ldap.organization.attributes=dc
  ldap.organization.objectclasses=domain
  ldap.user.objectclasses=person,organizationalPerson,inetOrgPerson
  ldap.group.objectclasses=groupOfUniqueNames
  ldap.user.shortname.attributes=cn,sn,uid
  ```

  Vous pouvez modifier les valeurs de ces conteneurs LDAP à l'aide d'un éditeur de texte.

4. Indique si tfimcfg crée un suffixe LDAP pour stocker les alias de connexion unique.

   Valeur par défaut :

   ```
   ldap.suffix.alias.configuration=true
   ```

   Si vous ne souhaitez pas que l'utilitaire spécifie un nouveau suffixe, définissez la valeur `false`.

   Lorsque cette propriété est définie sur `true`, tfimcfg utilise la valeur configurée dans la propriété suivante :

   ```
   ldap.suffix.alias.dn=cn=itfim
   ```

   Vous pouvez modifier la valeur du nom distinctif à l'aide d'un éditeur de texte. La valeur de cette propriété doit commencer par `cn=`.

5. Indiquez si tfimcfg doit créer le suffixe `secAuthority=Default` pour Tivoli Access Manager.

   Ce suffixe est utilisé par Tivoli Access Manager pour définir une hiérarchie LDAP exploitée par les serveurs Tivoli Access Manager. Ce suffixe est généralement créé par les scripts d'installation de Tivoli Access Manager. L'utilitaire tfimcfg ajoute le suffixe si celui-ci n'existe pas déjà.

   Valeur par défaut :

   ```
   ldap.suffix.tam.configuration=true
   ```

   - Si Tivoli Access Manager est déjà configuré, entrez `false` pour cette valeur.
   - Si Tivoli Access Manager n'utilise pas ce serveur LDAP, entrez `false` pour cette valeur.

   Remarque : Lorsque le suffixe secAuthority=Default existe déjà, le programme tfimcfg ignore la valeur de la propriété ldap.suffix.tam.configuration.

6. Indiquez si tfimcfg doit configurer LDAP pour le service d'alias Tivoli Federated Identity Manager.

   Valeur par défaut :

   ```
   ldap.fim.configuration=true
   ```

   Valeur par défaut : `true`.

   Lorsque la valeur 'true' est définie, tfimcfg configure les propriétés suivantes, comme spécifié dans le fichier ldapconfig.properties :

   - Nom distinctif, nom abrégé et mot de passe que le composant serveur Tivoli Federated Identity Manager (service d'exécution et de gestion) utilise pour se connecter au serveur LDAP. Valeurs par défaut :

     ```
     ldap.fim.server.bind.dn=uid=fimserver,cn=users,dc=exemple,dc=com
     ldap.fim.server.bind.shortname=fimserver
     ldap.fim.server.bind.password=passw0rd
     ```

- Nom distinctif et nom abrégé du groupe auquel appartient l'identité de l'utilisateur du serveur Tivoli Federated Identity Manager (fimserver). Valeurs par défaut :

  ```
  ldap.fim.admin.group.dn=cn=fimadmins,cn=groups,dc=exemple,dc=com
  ldap.fim.admin.group.shortname=fimadmins
  ```

  L'utilitaire tfimcfg ajoute ensuite l'utilisateur suivant :

  ```
  uid=fimserver,cn=users,dc=example,dc=com
  ```

  au groupe suivant :

  ```
  cn=fimadmins,cn=groups,dc=example,dc=com
  ```

7. Indiquez si tfimcfg connecte les listes de contrôle d'accès appropriées au serveur LDAP.

   Valeur par défaut :

   ```
   ldap.modify.acls=true
   ```

   Lorsque ce paramètre est défini sur `false`, vous devez connecter les listes de contrôle d'accès manuellement.

   Ces listes octroient l'accès en écriture et en lecture aux administrateurs Tivoli Federated Identity Manager créés par tfimcfg.

   A titre d'exemple, lorsque `ldap.modify.acls=true` et que tfimcfg est exécuté à l'aide des valeurs par défaut pour la création de suffixes, des listes de contrôle d'accès sont définies pour les suffixes suivants :

   - cn=itfim
   - secAuthority=Default
   - dc=com

   Remarque : L'outil tfimcfg connecte des listes de contrôle d'accès pour les serveurs IBM LDAP et Sun ONE. Pour les autres serveurs LDAP, vous devez connecter ces listes manuellement.

8. Indiquez des valeurs pour chacune des propriétés qui décrivent votre déploiement LDAP.

   Des valeurs par défaut sont fournies pour la plupart des propriétés. Modifiez les propriétés afin de les rendre conformes à votre déploiement. Lorsque la sécurité LDAP est activée, entrez le nom du fichier de clés Java contenant le certificat utilisé par SSL, puis le mot de passe utilisé par le service de gestion de Tivoli Federated Identity Manager.

Tableau 13. Propriétés LDAP à modifier pour tfimcfg

Propriété	Description	Votre valeur
ldap.hostname	Système qui héberge le serveur LDAP. La valeur par défaut est `localhost`.	
ldap.port	Numéro du port LDAP. La valeur par défaut est `389` pour la communication non SSL.	
ldap.admin.dn	Nom de l'administrateur LDAP. Valeur par défaut : `cn=root`	
ldap.admin.password	Mot de passe de l'administrateur LDAP.	
ldap.security.enabled	Valeur booléenne qui indique si la sécurité LDAP est activée. Cette valeur est désactivée par défaut.	
ldap.security.trusted.jks.filename	Nom du fichier de clés Java contenant le signataire du certificat SSL présenté par LDAP lors des communications sécurisées. Il n'existe pas d'entrée par défaut.	

Tableau 13. Propriétés LDAP à modifier pour tfimcfg (suite)

Propriété	Description	Votre valeur
ldap.fim.server.bind.password	Mot de passe pour les serveurs qui communiquent avec les serveurs LDAP. Remplacez les valeurs par défaut par les valeurs utilisées dans votre déploiement.	

9. Pour configurer le serveur LDAP, voir Annexe A, «Référence de tfimcfg», à la page 827.

Création d'un suffixe LDAP

Vous devez créer un suffixe LDAP tel que `cn=itfim` pour permettre au service d'alias d'accéder au registre d'utilisateurs LDAP.

Avant de commencer

Les instructions suivantes s'appliquent à IBM Tivoli Directory Server. Avant de passer aux étapes suivantes, assurez-vous d'avoir installé IBM Tivoli Directory Server et d'avoir procédé à la configuration initiale comme décrit dans la documentation.

Procédure

1. Arrêtez le serveur LDAP d'IBM.

   AIX, Linux ou Solaris :

   ```
   # ibmdirctl -D cn=root -w passw0rd stop
   ```

   Windows

   Utilisez l'icône Services.
2. Ajoutez le suffixe : `# idscfgsuf -s "cn=itfim"`.
3. Démarrez le serveur LDAP d'IBM.

   AIX, Linux et Solaris :

   ```
   # ibmdirctl -D cn=root -w passw0rd start
   ```

   Windows

   Utilisez l'icône Services.
4. Utilisez **ldapmodify** pour mettre à jour le fichier du schéma LDAP. Par exemple, sous Linux :

   ```
   ldapmodify -D cn=root -w passw0rd -f
     /opt/IBM/FIM/etc/itfim-secuser.ldif
   ```

Planification de la configuration des propriétés du service d'alias

Suivez ces instructions pour spécifier les propriétés du service d'alias permettant d'accéder à un ou plusieurs serveurs LDAP.

Pourquoi et quand exécuter cette tâche

Le service d'alias gère les alias en accédant à un registre d'utilisateurs LDAP. Un certain nombre d'informations relatives à l'environnement LDAP est nécessaire au service d'alias pour fonctionner dans celui-ci. La console de gestion est dotée d'une interface graphique que vous pouvez exploiter pour spécifier les propriétés

nécessaires. Les propriétés sont stockées dans un fichier de propriétés Tivoli Federated Identity Manager spécifique au domaine Tivoli Federated Identity Manager en cours d'utilisation.

La présente rubrique décrit les propriétés que vous avez besoin de spécifier. Elle fournit également une feuille de travail vous permettant de saisir les valeurs applicables à votre environnement. Dans de nombreux cas, vous aurez la possibilité de spécifier une valeur par défaut.

La valeur à définir pour certaines de ces propriétés correspond aux valeurs spécifiées antérieurement, lorsque vous avez planifié l'usage de l'utilitaire tfimcfg. Lors de cette étape, vous avez identité les valeurs à éditer dans le fichier ldapconfig.properties. Les tables présentées dans la séquence de tâches suivante identifient les zones de l'interface graphique qui contiennent des valeurs censées correspondre aux propriétés du fichier ldapconfig.properties.

Procédure

1. Déterminez la valeur de la propriété de recherche LDAP.

   La table suivante décrit le suffixe principal, la propriété de recherche LDAP configurable via l'interface graphique. Vous pouvez gérer la configuration en identifiant, à ce stade, la valeur appropriée pour votre environnement de déploiement.

Tableau 14. Propriétés de recherche LDAP

Propriété	Description	Votre valeur
Suffixe principal	Indique le suffixe racine dans lequel des paramètres du service d'alias sont écrits. Cette propriété ne peut comporter qu'une seule valeur (suffixe). La valeur de cette propriété correspond à la valeur de la propriété suivante dans ldapconfig.properties : `ldap.suffix.alias.dn` Par exemple : `cn=itfim`.	

2. Déterminez les valeurs des propriétés de l'environnement LDAP.

Tableau 15. Propriétés de l'environnement LDAP

Propriété	Description	Votre valeur
SSL activé	Case à cocher indiquant si les communications entre le service d'alias et les serveurs LDAP doivent être sécurisées grâce au protocole SSL (Secure Socket Layer). Lorsque les serveurs LDAP sont configurés pour utiliser SSL, le service d'alias doit également utiliser ce protocole lorsqu'il communique avec eux. La valeur de cette propriété correspond à la propriété suivante dans le fichier ldapconfig.properties : `ldap.security.enabled` Lors de l'utilisation de liaisons SSL, il convient de cocher la case **SSL activé** et de définir la propriété ldap.security.enabled sur la valeur `true`.	

Tableau 15. Propriétés de l'environnement LDAP (suite)

Propriété	Description	Votre valeur
Fichier de clés	Lorsque que la case **SSL activé** est cochée, vous devez sélectionner un fichier de clés dans le menu **Fichier de clés**. Il s'agit du nom du fichier de clés sécurisé contenant le certificat de CA du serveur LDAP. **Remarque :** Les certificats d'autorité de certification de tous les serveurs LDAP doivent se trouver dans le même fichier de clés. La valeur de cette propriété correspond à la propriété suivante dans le fichier ldapconfig.properties : `ldap.security.trusted.jks.filename`	

3. Déterminez les valeurs des propriétés du serveur LDAP

Tableau 16. Propriétés du serveur LDAP

Propriété	Description	Votre valeur
Nom d'hôte LDAP	La boîte **Hôtes LDAP** contient la liste des serveurs configurés, par ordre de préférence. Le service d'alias tente d'abord de contacter le serveur figurant au début de la liste. S'il ne parvient pas à établir le contact, il essaie avec le serveur suivant. La valeur de cette propriété inclut la propriété suivante dans le fichier ldapconfig.properties : `ldap.hostname` Le fichier ldapconfig.properties ne contient qu'une seule valeur pour cette propriété, mais vous pouvez en définir plusieurs pour le nom d'hôte LDAP.	
Port	Port sur lequel le serveur LDAP écoute. La valeur de cette propriété correspond à la propriété suivante dans le fichier ldapconfig.properties : `ldap.port` Port par défaut pour les communications autres que SSL : `389` Port par défaut pour les communications SSL : `636`	
Nom distinctif de la connexion	Nom distinctif utilisé par le service d'alias pour établir une liaison avec le serveur LDAP. La valeur de cette propriété correspond à la propriété suivante dans le fichier ldapconfig.properties : `ldap.fim.server.bind.dn` Le panneau graphique indique la valeur par défaut : cn=root. Toutefois, les droits d'accès de l'utilisateur root ne sont pas requis pour définir cette liaison. Vous pouvez spécifier le nom distinctif du service d'alias. Valeur par défaut : `uid=fimserver`	
Mot de passe BIND	Mot de passe du nom distinctif indiqué dans la zone **DN Bind**. La valeur de cette propriété correspond à la propriété suivante dans le fichier ldapconfig.properties : `ldap.fim.server.bind.password`	

Tableau 16. Propriétés du serveur LDAP (suite)

Propriété	Description	Votre valeur
Mode	Par défaut, le mode d'accès en lecture-écriture est configuré. Dans le cas de la configuration de serveurs LDAP multiples, il convient qu'une seule entité soit accessible en lecture-écriture. Dans le présent scénario, les autres serveurs LDAP sont généralement déployés pour les nécessités liées à la reprise en ligne et sont donc censés être dotés de copies du registre d'utilisateurs accessibles en lecture seule.	
Nombre minimal de connexions	Nombre de connexions (binds ou liaisons) minimal que le service d'alias peut établir avec le serveur LDAP. La plus petite valeur acceptée est zéro (0). La valeur maximale acceptée correspond à la valeur maximale prise en charge par le type de données. La valeur par défaut est 2. Utilisez-la, sauf si vous devez l'augmenter.	
Nombre maximal de connexions	Nombre de connexions (binds ou liaisons) maximal que le service d'alias peut établir avec le serveur LDAP. La valeur maximale acceptée correspond à la valeur maximale prise en charge par le type de données. La valeur par défaut est 10. Utilisez-la, sauf si vous avez besoin de l'augmenter.	

Modification des paramètres du service d'alias pour LDAP

Savoir modifier les paramètres de la base de données de l'identificateur de noms.

Procédure

1. Connectez-vous à la console. Le portlet Paramètres du service d'alias s'ouvre.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Paramètres du service d'alias**.
3. Sélectionnez **LDAP**.

   Spécifiez les propriétés à insérer dans le formulaire suivant la section «Planification de la configuration des propriétés du service d'alias», à la page 149
4. Si vous choisissez une communication SSL avec LDAP, sélectionnez le nom du fichier de clés sécurisé contenant le certificat de l'autorité de certification du serveur LDAP. Si vous n'avez pas encore déplacé l'autorité de certification LDAP vers le service de clés de Tivoli Federated Manager, vous pouvez extraire le certificat sur SSL en procédant comme suit :
   a. Dans la console, sélectionnez **Tivoli Federated Identity Manager** > **Service de clés**.
   b. Sélectionnez le fichier de clés certifiées dans lequel vous voulez stocker le certificat dans le tableau des fichiers de clés. Le bouton Afficher les clés est activé.
   c. Cliquez sur **Extraire le certificat de SSL**. Le panneau Mot de passe s'affiche.
   d. Entrez le mot de passe du fichier de clés.
   e. Cliquez sur **OK**.
   f. Remplissez les zones prévues pour le nom d'hôte et le nom de port à partir desquels vous devez récupérer le certificat.

      (Facultatif) Cliquez sur **Afficher les informations sur le signataire** pour visualiser le certificat avant de le récupérer.
   g. Entrez dans la zone **Alias** le nom à attribuer au certificat. Cliquez ensuite sur **OK**. Le certificat est ajouté au fichier de clés certifiées.

5. Cliquez sur **Appliquer**.
6. Cliquez sur **OK**.

Configuration d'une base de données de service d'alias Oracle

Vous pouvez configurer IBM Tivoli Federated Identity Manager pour qu'il utilise Oracle en tant que service d'alias.

Pourquoi et quand exécuter cette tâche

Ces instructions indiquent comment configurer IBM Tivoli Federated Identity Manager pour qu'il utilise Oracle en tant que service d'alias.

Procédure

1. Faites une sauvegarde du fichier itfim.ear. Utilisez les commandes suivantes :

   ```
   mkdir /tmp/work
   rm FIM_INSTALL_DIR/pkg/release/itfim.ear
   ```

2. Modifiez le fichier d'archive d'entreprise EAR pour qu'il prenne en compte Oracle pour le déploiement. Utilisez les commandes suivantes :

   ```
   mkdir /tmp/work
   rm FIM_INSTALL_DIR/pkg/release/itfim.ear
   WEBSPHERE_INSTALL_DIR/AppServer/bin/ejbdeploy.sh FIM_INSTALL_DIR/pkg/
   release/itfim-orig.ear /tmp/work FIM_INSTALL_DIR/pkg/release/itfim-
   oracle.ear -dbschema FIMAliasesSchema -dbname FIMAliases -dbvendor
   ORACLE_V10G -trace
   cp FIM_INSTALL_DIR/pkg/release/itfim-oracle.ear FIM_INSTALL_DIR/pkg/
   release/itfim.ear
   rm -rf /tmp/work
   ```

3. Un nouveau fichier `FIM_INSTALL_DIR/pkg/release/itfim.ear` est disponible pour que le déploiement soit compatible avec Oracle. Mettez à jour le fichier FIM_INSTALL_DIR/pkg/software.properties à l'aide d'un éditeur de texte afin de modifier la propriété `com.tivoli.am.fim.rte.software.serialId` en une valeur différent, par exemple, increment.
4. A l'aide de la console IBM Tivoli Federated Identity Manager, accédez à **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution**. Un message indiquant qu'une nouvelle exécution est disponible pour le déploiement s'affiche.
5. Utilisez la console pour déployer la phase d'exécution.
6. Redémarrez le processus WebSphere sur lequel la phase d'exécution est déployée.

Chapitre 14. Planification du mappage des identités d'utilisateur

Planifiez le mappage des identités utilisateur appropriées à votre déploiement.

Présentation des tâches :

1. Consultez cette série de rubrique sur le mappage des identités d'utilisateur
2. Vérifiez les fichiers de règles de mappage par défaut disponibles pour votre protocole. Déterminez s'ils peuvent vous être utiles tels quels ou après des modifications adaptées à votre déploiement.
3. Si les exigences liées à votre déploiement ne peuvent être satisfaites au moyen d'une règle de mappage, vous pouvez sélectionner l'une des options suivantes :
   - Utilisez le module de mappage de Tivoli Directory Integrator fourni avec Tivoli Federated Identity Manager.
   - Développez un module de mappage personnalisé.

L'une des fonctions principales du service d'accréditation Tivoli Federated Identity Manager consiste à transférer les informations sur l'identité des utilisateurs (données d'identification) entre les partenaires au sein d'une fédération à connexion unique. Ce transfert nécessite plusieurs modification du format des informations d'identité utilisateur pour passer des formats locaux de chaque partenaire et le format de jeton convenu pour échanger les données d'identification.

Le transfert des informations d'identité comprend un mappage d'identité où les informations utilisateur sont mappées depuis la structure fournie par un type de jeton ou des données d'identification, vers la structure requise par un autre type de jeton.

Pour effectuer cette étape de mappage, sélectionnez l'une des options suivantes :

- Créez une règle de mappage d'identité
- Déployez le module de mappage Tivoli Directory Integrator

  L'utilisation de ce module nécessite une bonne connaissance des fonctionnalités et de la configuration de Tivoli Directory Integrator. Reportez-vous à la documentation du produit pour Tivoli Directory Integrator.

  Tivoli Federated Identity Manager fournit une interface utilisateur permettant de définir certaines propriétés de configuration. Voir «Module de mappage d'identité Tivoli Directory Integrator», à la page 164.
- Développez un module de mappage personnalisé.

  La création de votre propre module conçu pour répondre sur mesure aux besoins des applications de votre déploiement correspond à une tâche de développement. Voir «Création d'un module de mappage personnalisé», à la page 176.

Si vous choisissez de créer une règle de mappage d'identité, utilisez le langage XSL (eXtensible Stylesheet Language) et sauvegardez le fichier XSL sur disque. Lorsque vous créez une fédération, l'assistant de fédération vous invite à indiquer le nom de votre fichier de règles de mappage. L'assistant importe ce fichier dans la configuration de la fédération.

Chaque fichier de règles de mappage est spécifique à un rôle et à une fédération en particulier. Par exemple, lorsque vous créez une fédération SAML pour un fournisseur d'identité, utilisez une règle de mappage différente de celle utilisée pour créer un fournisseur de service de fédération SAML. La règle de mappage d'identité d'une fédération Liberty est également différente de la règle de mappage d'une fédération SAML sur un fournisseur d'identité.

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Vous devez créer et enregistrer un fichier de règles de mappage avant de créer une fédération.

Remarque : Les règles de mappage d'identité indiquent les attributs associés aux données d'identification d'un utilisateur. Les utilisateurs peuvent accéder à plusieurs applications après s'être authentifiés. Par conséquent, vous devez vérifier que votre règle définit les attributs appropriés pour toutes les applications accessibles à l'utilisateur.

La console de gestion de Tivoli Federated Identity Manager fournit un assistant de fédération qui vous guide dans la configuration d'une fédération de connexion unique. Cet assistant contient un écran Mappage d'identité qui invite l'administrateur à fournir le nom d'un fichier de règles de mappage d'identité. L'assistant importe le fichier puis l'utilise lors de la génération de la configuration de la chaîne de modules d'accréditation qui est spécifique à la fédération.

L'administrateur doit créer le fichier de mappage d'identité avant d'utiliser l'assistant pour configurer la fédération. L'assistant s'attend à ce que l'administrateur crée un fichier XSL (eXtensible Stylesheet Language) décrivant les règles de mappage d'identité. Celles-ci servent à convertir les données échangées via la fédération entre les partenaires (fournisseur d'identité et fournisseur de services). Chaque règle de mappage d'identité doit fournir :

- la structure de données dont la génération est requise par le jeton de sécurité ;
- le contenu des données (attributs d'identité) qui est requis par les applications utilisant la fédération.

Pour écrire une règle de mappage d'identité, vous devez comprendre :

- le rôle du module de mappage d'identité ;
- l'expression des données d'identification d'utilisateur dans les fichiers XML ;
- l'utilisation du langage XSL pour la spécification des règles permettant de manipuler les données d'identification d'utilisateur.

Généralités sur le mappage d'identité

Lors de l'échange de jetons de sécurité avec des partenaires, il ne suffit pas de comprendre les différentes normes relatives aux jetons. Il est primordial de savoir quelles informations un partenaire particulier s'attend à trouver dans les jetons provenant de vos sites, et quelles informations vous vous attendez à recevoir de la part de vos partenaires. Vous pouvez utiliser le mappage d'identité Tivoli Federated Identity Manager et le service d'accréditation afin de personnaliser le format et le contenu des jetons entrants et sortants pour répondre aux besoins de chaque partenaire.

Dans une fédération de connexion unique, un fournisseur d'identité authentifie l'utilisateur. Cette authentification crée des données d'identification de l'utilisateur

dans l'environnement du fournisseur d'identité. Par exemple, il est possible que le fournisseur d'identité exige l'authentification des utilisateurs à l'aide d'un nom d'utilisateur et d'un mot de passe. Les informations utilisateur sont validées par rapport au registre d'utilisateurs du fournisseur d'identité. Ensuite, les droits d'accès locaux contenant les données d'appartenance à un groupe ainsi que des attributs facultatifs concernant l'utilisateur sont créés.

Dans le cas le plus général d'utilisation de SAML 2.0, le nom d'utilisateur n'est pas inclus dans l'assertion. L'utilisateur est, au lieu de cela, représenté par un alias.

Un fournisseur de services possède également des exigences spécifiques pour ses données d'identification de l'utilisateur, dont les utilisateurs doivent tenir compte pour accéder aux applications. Dans la plupart des cas, les données d'identification requises par le fournisseur de services diffèrent quant au format ou au contenu des données d'identification créées par le fournisseur d'identité. Il se peut, par exemple, que le fournisseur de services souhaite inclure un attribut spécifique, tel que le numéro de sécurité sociale de l'utilisateur, dans les données d'identification. Par conséquent, il est nécessaire de mapper les données d'identification entre le fournisseur d'identité et le fournisseur de services.

Dans Tivoli Federated Identity Manager, côté fournisseur d'identité, l'utilisateur authentifié en local (identité d'entrée) peut être mappé vers un utilisateur différent avant la création du jeton de connexion unique (identité de sortie). De même, côté fournisseur de services, l'identité reçue du jeton de connexion (identité d'entrée) peut être mappée vers une identité locale qui est nécessaire pour l'accès aux applications du fournisseur de services (identité de sortie). Le processus de mappage est présenté dans la figure 7.

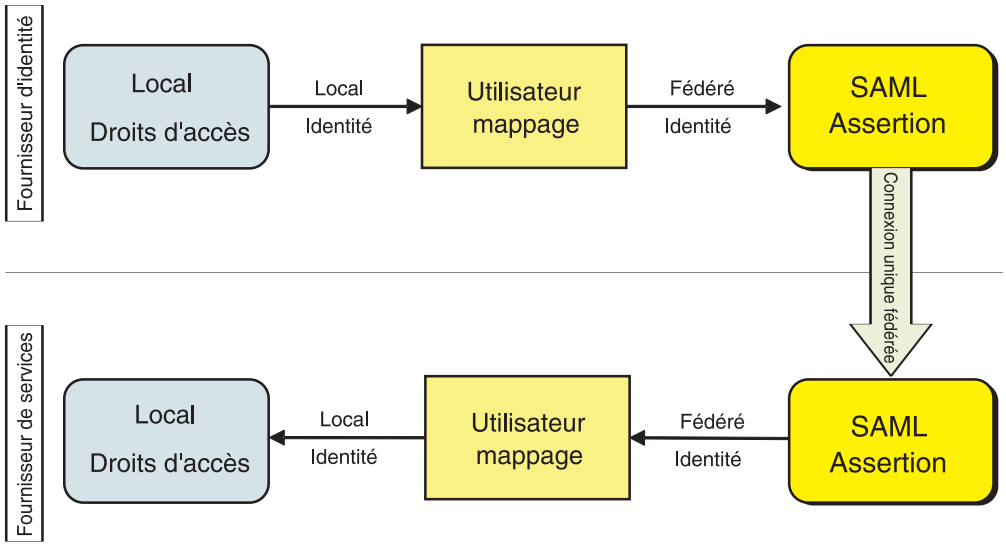


Figure 7. Exemple de mappage d'identité

Plusieurs méthodes sont possible pour générer l'identité sortante requise lors du mappage utilisateur. Par exemple, les informations codées en dur peuvent être ajoutées au jeton sortant. Ou un code Java peut être développé et utilisé pour acquérir les informations provenant de sources externes afin de les ajouter au jeton sortant. Cette souplesse est permise par les *règles de mappage d'identité*, qui sont définies de l'un ou l'autre des manières suivantes :

- Un fichier XSLT (eXtensible Stylesheet Language Transformation) est traité par le module de mappage d'identité de Tivoli Federated Identity Manager.

- Un module de mappage personnalisé créé à l'aide de Java.

Avant que vous ne décidiez quelle méthode employer, vous devez comprendre les facteurs suivants :

- le mode de représentation des identités d'utilisateurs dans Tivoli Federated Identity Manager ;
- le mode de traitement des jetons, ainsi que
- la manière dont les identités sont mappées entre les partenaires.

Document relatif à l'utilisateur universel STS (Security Token Service)

Pour garantir qu'un jeton entrant est correctement converti en un jeton sortant qui intègre le contenu et le format requis par le partenaire, Tivoli Federated Identity Manager crée un document intermédiaire dans un format XML générique, contenant les informations d'identité. Ce document s'appelle l'utilisateur universel STS (STS Universal User) ou STSUU. Le document STSUU comprend trois sections :

- Informations principales
- Informations de groupe
- Informations d'attribut

Pour créer le document STSUU, Tivoli Federated Identity Manager utilise un schéma XML qui spécifie la structure. Le schéma est défini dans le fichier stsuuser.xsd. L'exemple de code ci-dessous contient la totalité du contenu du schéma XML d'utilisateur universel STS (Secure Token Service).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:ibm:names:ITFIM:1.0:stsuuser"
xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser"
elementFormDefault="qualified">

 <xsd:element name="STSUniversalUser">
  <xsd:complexType>
   <xsd:sequence>
     <xsd:element name="Principal" type="stsuuser:PrincipalType"
             minOccurs="1" maxOccurs="1"/>
     <xsd:element name="GroupList" type="stsuuser:GroupListType"
             minOccurs="0" maxOccurs="1"/>
     <xsd:element name="AttributeList" type="stsuuser:AttributeListType"
             minOccurs="0" maxOccurs="1"/>
     <xsd:element name="RequestSecurityToken" type="stsuuser:RequestSecurityTokenType"
             minOccurs="0" maxOccurs="1"/>
   </xsd:sequence>
   <xsd:attribute name="version" type="xsd:string" use="required"/>
  </xsd:complexType>
 </xsd:element>

 <xsd:complexType name="PrincipalType">
  <xsd:sequence>
   <xsd:element name="Attribute" type="stsuuser:AttributeType"
         minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="RequestSecurityTokenType">
  <xsd:sequence>
   <xsd:element name="Attribute" type="stsuuser:AttributeType"
         minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="AttributeType">
                 <xsd:sequence>
                  <xsd:element name="Value" type="xsd:string"
                        minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                 <xsd:attribute name="name" type="xsd:string" use="required"/>
                 <xsd:attribute name="type" type="xsd:string" use="optional" />
                 <xsd:attribute name="nickname" type="xsd:string" use="optional" />
                  <xsd:attribute name="preferEncryption" type="xsd:boolean" use="optional" />
 </xsd:complexType>

 <xsd:complexType name="AttributeListType">
  <xsd:sequence>
   <xsd:element name="Attribute" type="stsuuser:AttributeType"
         minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="GroupListType">
  <xsd:sequence>
   <xsd:element name="Group" type="stsuuser:GroupType"
         minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="GroupType">
  <xsd:sequence>
   <xsd:element name="Attribute" type="stsuuser:AttributeType"
         minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required" />
  <xsd:attribute name="type" type="xsd:string" use="optional" />
 </xsd:complexType>

</xsd:schema>
```

Figure 8. Schéma du document STSUU

Bien que le schéma serve de base à tous les documents STSUU, es informations exactes contenues dans n'importe quel document STSUU sont fonction du type de jeton de sécurité qui a été utilisé en entrée. Les informations requises dans un document STSUU à la suite d'une transformation par le mappage d'identité sont fonction :

- du type de jeton à générer ;
- de la règle de mappage spécifique utilisée pour la conversion.

Lors du traitement d'un jeton dans le cadre d'une configuration de connexion unique typique, deux documents STSUU sont créés. L'un d'eux est le document STSUU d'entrée créé à partir du jeton d'entrée d'origine. L'autre est le STSUU de sortie créé une fois les règles de mappage d'identité appliquées. Pour plus d'informations, voir «Traitement des jetons».

Traitement des jetons

Dans une configuration typique de connexion unique, les jetons sont traités par le service d'accréditation Tivoli Federated Identity Manager, ainsi que trois types spécifiques de modules. Lorsque ces modules sont utilisés en combinaison, on par le *chaîne d'accréditation*. La figure 9, à la page 161 fournit un diagramme du traitement des jetons. A l'entrée de la chaîne d'accréditation se trouve le jeton de sécurité d'entrée. La création de ce jeton a lieu sur la base des droits d'accès locaux reçus au moment où l'utilisateur se connecte.

Le premier module de la chaîne d'accréditation convertit le jeton d'entrée en document STSUU. Tous les attributs contenus dans le jeton d'entrée sont disponibles dans le document STSUU. Le document STSUU est alors utilisé comme entrée du module de mappage d'identité. Ce module peut être le module de mappage de Tivoli Federated Identity Manager qui est utilisé avec un fichier XSLT. Ou bien il peut s'agir d'un module de mappage personnalisé que vous avez créé.

Un module de mappage donné peut être utilisé conjointement par de nombreux partenaires de la fédération, ou bien être associé à un partenaire spécifique. En sortie du module de mappage, un autre document STSUU est généré. Ce document STSUU "de sortie" est utilisé comme entrée du troisième module de traitement des jetons, qui convertit le STSUU de sortie en jeton de sortie. Le jeton de sortie est ensuite envoyé au partenaire.

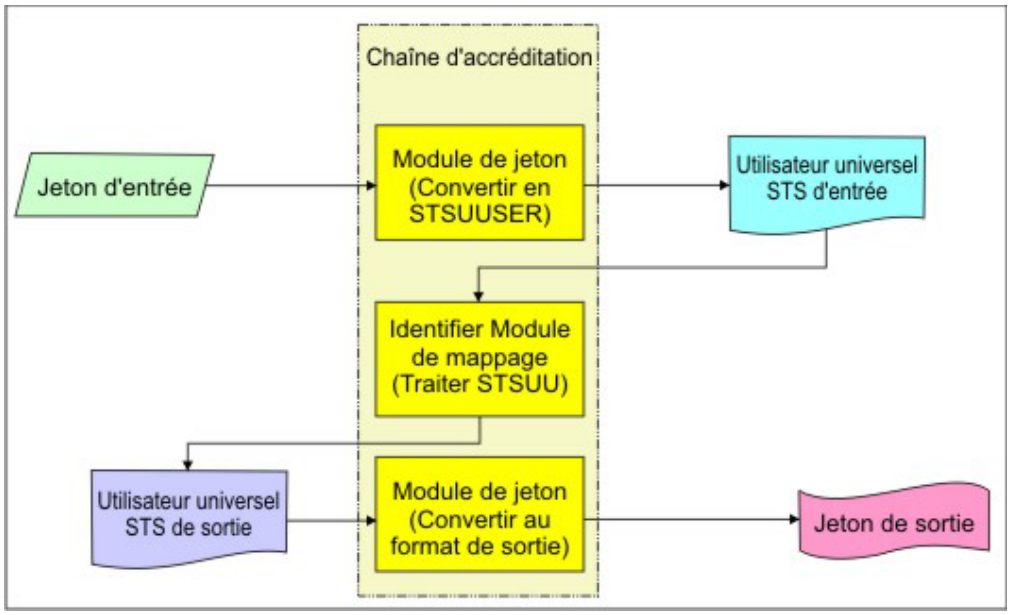


Figure 9. Traitement des jetons

Utilisation du langage XSL pour la création de fichiers de règles de mappage

Le module de mappage d'identité utilise l'API Java API d'analyse syntaxique XML (JAXP) pour transformer le document d'entrée. Cette transformation est effectuée en fonction de la feuille de style XSL que vous spécifiez dans un fichier XSL.

XSL est un langage qui permet de transformer et de formater des documents. XSL sert à définir des feuilles de style pour HTML et à formater des données XML, de manière à pouvoir les afficher dans un navigateur Web. Une partie de la norme XSL définit des transformations permettant de convertir des données d'un format à l'autre. Ce langage de transformation peut inclure des instructions conditionnels, des variables et des appels à des programmes Java.

Le service d'accréditation utilise XSL pour créer des règles de mappage. Les règles de mappage créées spécifient la procédure à suivre pour transformer un document d'utilisateur universel STS d'entrée en un document d'utilisateur universel STS de sortie. Le document d'utilisateur universel STS est utilisé en tant qu'entrée du module suivant dans la chaîne. Ce module est souvent utilisé pour générer un jeton de sortie, mais il peut également s'agir d'un autre module de mappage. L'analyseur syntaxique XSL traite les documents XSL en recherchant des modèles correspondants. Une fois qu'un modèle est détecté, le contenu de celui-ci est traité.

Voici les principales tâches qui sont exécutées dans les règles de mappage :

- Transfert des informations d'identité entre les éléments.
- Reformatage des informations d'identité existantes.
- Ajout de nouveaux éléments avec de nouvelles informations d'identité.
- Suppression des informations d'identité superflues.

Vous pouvez utiliser l'outil IBM Rational Application Developer pour exécuter un débogueur XSL à partir d'une ligne de commande. Utilisez le jeu d'outils du développeur pour tester votre code XSL sans exécuter le service d'accréditation.

Tivoli Federated Identity Manager fournit deux jeux de modèles de fichiers de mappage d'identité. Le premier jeu affiche le contenu minimal de chaque type de mappage, tandis que le second présente la fonctionnalité avancée qui est utilisées dans l'application de démonstration.

L'emplacement des fichiers de mappage de base est le suivant :
`/opt/IBM/FIM/examples/mapping_rules/`

Le tableau suivant répertorie les fichiers d'exemples de règles de mappage.

Tableau 17. Exemples de règles de mappage

Nom de fichier	Description du mappage
ip_liberty.xsl	Mappe une accréditation Tivoli Access Manager ou une identité d'utilisateur local à un jeton Liberty.
ip_saml_1x.xsl	Mappe une identité d'utilisateur local à un jeton SAML 1.0 ou SAML 1.1.
ip_saml_20.xsl	Utilise un jeton pour mapper une identité d'utilisateur local à un jeton SAML 2.0.
ip_saml_20_email_nameid.xsl	Utilise l'adresse de courrier électronique de l'utilisateur pour l'identité sans alias, pour mapper une identité d'utilisateur local à un jeton SAML 2.0.
ip_wsfederation.xsl	Mappe une accréditation Tivoli Access Manager ou une identité d'utilisateur local à un jeton SAML.
ip_infocard.xsl	Mappe un jeton entrant ou une identité d'utilisateur local à un jeton SAML 1.1. La principale fonction de cette règle est de compléter les attributs de réclamation requis par des valeurs.
ip_openid.xsl	Mappe un jeton IVCred ou une identité d'utilisateur local à un jeton STSUU (Security Token Service Universal User). La principale fonction de cette règle est de compléter les attributs requis (SREG et AX) et d'agir sur les règles PAPE demandées.
rp_infocard.xsl	Mappe un jeton SAML 1.1 ou une identité d'utilisateur local à un jeton IVCred.
sp_liberty.xsl	Mappe un jeton Liberty à une accréditation Tivoli Access Manager ou une identité d'utilisateur local.
sp_saml_20.xsl	Mappe un jeton SAML 2.0 à une identité d'utilisateur local.
sp_saml_1x.xsl	Mappe un jeton SAML 1.0 ou 1.1 à une identité d'utilisateur local.
sp_saml_1x_ext.xsl	Mappe un jeton 1.0 ou 1.1 avec une identité d'utilisateur local et vérifie l'acceptabilité de la méthode d'authentification. Ceci montre que le fournisseur de service peut exiger que l'authentification au fournisseur d'identité soit effectuée à un certain niveau. Selon cette règle de mappage, l'authentification par mot de passe n'est pas acceptée. Une erreur est générée si l'authentification du mot de passe a été utilisée.

Tableau 17. Exemples de règles de mappage (suite)

Nom de fichier	Description du mappage
sp_wsfederation.xsl	Mappe un jeton SAML à une accréditation Tivoli Access Manager ou à une identité d'utilisateur local.
sp_tagvalue.xsl	Mappe un jeton SAML à une accréditation Tivoli Access Manager IV Cred comportant des attributs de balise/valeur WebSEAL ou à une identité d'utilisateur local.
username_ivcred.xsl	Mappe un jeton de nom d'utilisateur à une accréditation Tivoli Access Manager ou à une identité d'utilisateur local.
sp_oauth_10.xsl	Prend en charge le flux OAuth 1.0.
sp_oauth_20.xsl	Prend en charge le flux OAuth 2.0.

Remarque : Pour plus d'informations sur les exemples de règles de mappage pour chaque protocole, consultez les instructions de configuration spécifiques au protocole fournies dans ce guide.

L'application de démonstration fournit des exemples de fichiers de règles de mappage d'identité XSL. Ces fichiers développent les règles de mappage minimales décrites dans la table précédente qui est personnalisée pour les comptes d'utilisateurs. Les scripts de configuration d'application de démonstration créent les comptes d'utilisateur.

L'emplacement des exemples de scripts de mappage de l'application de démonstration est le suivant :

```
/opt/IBM/FIM/examples/demo/demo_rules/
```

Remarque : Le noms de fichier sont identiques à ceux des règles de mappage minimales, mais les fichiers résident dans des répertoires différents.

Les exemples de fichier de mappage sont installés automatiquement durant l'installation.

Le tableau suivant répertorie les fichiers pour chaque type de fédération sur chaque type de fournisseur.

Tableau 18. Exemples de fichiers de règles de mappage de l'application de démonstration

Fournisseur	Type de fédération	Fichier de règles de mappage
Fournisseur d'identité	Liberty	`ip_liberty.xsl`
	SAML 1.0	`ip_saml_10.xsl`
	SAML 1.1	`ip_saml_11.xsl`
	SAML 2.0	`ip_saml_20.xsl`
	WS-Federation	`ip_wsfederation.xsl`
	Carte d'information	`ip_openid.xsl`
	OpenID	`ip_infocard.xsl`

Tableau 18. Exemples de fichiers de règles de mappage de l'application de démonstration (suite)

Fournisseur	Type de fédération	Fichier de règles de mappage
Fournisseur de services	Liberty	`sp_liberty.xsl`
	SAML 1.0 ou 1.1	`sp_saml_1x.xsl`
	SAML 2.0	`sp_saml_20.xsl`
	WS-Federation	`sp_wsfederation.xsl`
	Carte d'information	`rp_infocard.xsl`
	OpenID	`sp_openid.xsl`
	OAuth 1.0	`sp_oauth_10.xsl`
	OAuth 2.0	`sp_oauth_20.xsl`

Module de mappage d'identité Tivoli Directory Integrator

Ce module permet d'effectuer des opérations de mappage d'attributs et d'utilisateur génériques.

Une chaîne d'assemblage s'exécutant sur un serveur Tivoli Directory Integrator est appelée pour effectuer le mappage des données d'utilisateur et d'attribut dans un STSUniversalUser. Les données peuvent être résolues à partir de diverses sources de données prises en charge nativement par le serveur, y compris LDAP et les bases de données relationnelles. Le code personnalisé est également pris en charge via les connecteurs JavaScript.

Tivoli Federated Identity Manager fournit un exemple de fichiers de mappage pour Tivoli Directory Integrator. Le fichier se trouve dans le même répertoire que les autres fichiers d'exemple. Par exemple, sous Linux ou UNIX, le fichier se trouve à l'emplacement suivant

`/opt/IBM/FIM/examples/tdi_mappings/tdi_demo_mappings.xml`

Le déploiement de ce module nécessite les éléments suivants :

- Configuration des paramètres du module d'accréditation de Tivoli Directory Integrator
- Configuration du serveur Tivoli Directory Integrator
- Configuration de la communication SSL entre le serveur Tivoli Directory Integrator et le client, également appelé module d'accréditation

Suivez les instructions de configuration des rubriques :

- «Configuration du module d'accréditaion de Tivoli Directory Integrator»
- «Configuration du serveur Tivoli Directory Integrator», à la page 167
- «Configuration du protocole SSL pour le module d'accréditation de Tivoli Directory Integrator», à la page 168

Configuration du module d'accréditaion de Tivoli Directory Integrator

Vous devez fournir les propriétés de configuration pour le module de jeton de sécurité Tivoli Directory Integrator lors de la création d'une chaîne d'accréditation.

Les propriétés sont décrites dans la présente rubrique, qui contient également un formulaire que vous pouvez consulter lors de la configuration de votre module via la console d'administration.

Propriétés de configuration

Nom d'hôte du serveur
Nom d'hôte ou adresse IP de l'ordinateur sur lequel le serveur Tivoli Directory Integrator est exécuté. La valeur par défaut est localhost. Par exemple, `tdiserver.company.com`

Port du serveur
Numéro de port sur lequel le serveur Tivoli Directory Integrator est configuré pour l'exécution. La valeur par défaut est 1099.

Taille du pool de gestionnaires de chaîne d'assemblage
Nombre de gestionnaires de chaîne d'assemblage à prendre en charge pour cette chaîne d'accréditation. La valeur doit être un entier positif. La valeur par défaut est 10.

Nombre d'unités d'exécution en attente
Nombre maximal d'unités d'exécution en attente de gestionnaire de chaîne d'assemblage pour cette chaîne. La valeur indiquée doit être un entier. La valeur par défaut est 0.

Durée d'attente des unités d'exécution avant qu'un gestionnaire de chaîne d'assemblage devienne disponible
Déterminez la durée d'attente des unités d'exécution avant qu'un gestionnaire de chaîne d'assemblage devienne disponible. Sélectionnez l'une de ces options.

Attendre indéfiniment
N'imposez pas de limite pour le temps d'attente des unités d'exécution avant qu'un gestionnaire de chaîne d'assemblage devienne disponible. Il s'agit de l'option par défaut.

Ne pas attendre le gestionnaire de chaîne d'assemblage après la tentative initiale
Les unités d'exécution ne doivent pas attendre un gestionnaire de chaînes d'assemblage. Si un gestionnaire de chaîne d'assemblage n'est pas disponible immédiatement, le module Tivoli Directory Integrator indique que le délai a expiré.

Utiliser la valeur d'attente maximale suivante
Spécifiez une valeur de durée d'attente maximale.

Durée d'attente maximale (millisecondes)
Durée d'attente maximale d'un gestionnaire de chaîne d'assemblage par une unité d'exécution avant l'émission d'un message d'expiration du délai. Cette valeur est spécifiée en millisecondes et doit être un entier positif.

Reconnaissance des paramètres de configuration
Utilisez le port et le nom d'hôte du serveur fournis précédemment dans ce panneau pour établir la connexion au serveur Tivoli Directory Integrator. Lorsque vous êtes connecté, vous pouvez découvrir les configurations et les chaînes d'assemblage disponibles. Vous devez entrer le port et le nom d'hôte du serveur avant de sélectionner cette option. Une fois cette option sélectionnée, deux menus déroulants sont disponibles.

Sélectionner le fichier de configuration
Sélectionnez le fichier de configuration à utiliser dans la liste.

Sélectionner la chaîne d'assemblage
Sélectionnez la chaîne d'assemblage à utiliser dans la liste. Cette liste est issue du fichier de configuration que vous avez sélectionné dans la zone ci-dessus.

Entrer les paramètres de configuration manuellement
Entrez les paramètres de configuration manuellement en renseignant les zones suivantes :

Fichier de configuration
Nom de la solution ou nom du fichier de configuration à utiliser. Par exemple, `tdi_demo_mappings.xml`

Nom de la chaîne d'assemblage
Nom de la chaîne d'assemblage à utiliser. Par exemple, `assemblyLine1`

Sélectionnez le format d'identification pour les attributs de poste travaux
Sélectionnez le format d'identification pour les attributs de poste travaux. Sélectionnez l'une des options suivantes :

Nom d'attribut
Le poste travaux va utiliser le nom pour identifier ses attributs.

Nom d'attribut et type d'attribut
Le poste travaux va utiliser le nom et le type pour identifier ses attributs. Utilisez cette méthode si plusieurs attributs portant le même nom existent.

Tableau 19. Formulaire comportant les propriété de configuration de Tivoli Directory Integrator Module

Propriété	Votre valeur
Nom d'hôte du serveur	
Port du serveur	
Taille du pool de gestionnaires de chaîne d'assemblage	
Nombre d'unités d'exécution en attente	
Durée d'attente des unités d'exécution avant qu'un gestionnaire de chaîne d'assemblage devienne disponible	Le panneau de configuration propose 3 options : • Attendre indéfiniment • Ne pas attendre le gestionnaire de chaîne d'assemblage après la tentative initiale • Utiliser la valeur d'attente maximale suivante : Durée d'attente maximale (millisecondes)
Méthode de sélection des paramètres de la ligne d'assemblage	2 choix : • Reconnaissance des paramètres de configuration • Entrer les paramètres de configuration manuellement
Fichier de configuration	
Nom de la chaîne d'assemblage	
Format d'identification pour les attributs de poste travaux	2 choix : • Nom d'attribut • Nom d'attribut et type d'attribut

Configuration du serveur Tivoli Directory Integrator

Cette rubrique décrit la procédure minimale requise lors de la configuration d'une installation par défaut du serveur Tivoli Directory Integrator (TDI). Cette procédure s'applique aux versions 6.1.1, 7.0 et 7.1 du serveur TDI.

Pourquoi et quand exécuter cette tâche

La configuration du serveur Tivoli Directory Integrator permet d'exécuter les chaînes d'assemblage avec Tivoli Federated Identity Manager et le module STS de TDI. Le fichier `tdi_demo_mappings.xml` est un exemple de configuration.

Pour obtenir des instructions de configuration de TDI détaillées, consultez le centre de documentation de Tivoli Directory Integrator à l'adresse http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc_7.1/welcome.htm.

Le programme d'installation de TDI vous invite à sélectionner un répertoire de solutions parmi les suivants :

- Sous-répertoire TDI situé sous le répertoire de base (par défaut)
- Répertoire d'installation
- Sélectionner un répertoire à utiliser

La procédure présentée concerne l'utilisation de la version 7.1 du serveur TDI et du répertoire de solutions par défaut.

Procédure

1. Créez les fichiers de solutions.

   Après l'installation initiale, un sous-répertoire est créé sous le répertoire de base de l'utilisateur root : `/root/TDI`. Ce répertoire ne contient aucun fichier de solutions.

   Pour en ajouter, démarrez le serveur TDI sans aucun paramètre :

   ```
   # /opt/IBM/TDI/V7.1/ibmdisrv
   ```

   Vous pouvez également démarrer l'éditeur de configuration de TDI. Une fois le serveur démarré, les fichiers de solutions sont générés dans le répertoire `/root/TDI`, y compris le fichier `solutions.properties`.

2. Modifiez le fichier `solutions.properties`.

   `api.remote.on`
   Cette propriété vous permet d'utiliser l'interface API du serveur distant utilisée par le module STS TDI. Changez la valeur `false` en `true`.

   `api.remote.ssl.on`
   Ces instructions illustrent la configuration de TDI sans SSL. La configuration SSL est indiquée dans la section *Configuration du protocole SSL pour le module d'accréditation de Tivoli Directory Integrator*. Changez la valeur `true` en `false`.

   `api.remote.nonssl.hosts`
   Cette propriété est nécessaire lorsque le serveur TDI est exécuté sur un hôte autre que le composant d'exécution Tivoli Federated Identity Manager et lorsque le protocole SSL n'est pas utilisé. Spécifiez l'adresse IP de la machine exécutant le composant d'exécution (serveur d'accréditation).

3. Créez et remplissez le répertoire de configuration de TDI.

Le fichier `solutions.properties` contient un paramètre décrivant l'emplacement des fichiers de configuration de TDI que vous pouvez modifier via l'API du serveur. Cette propriété, ainsi que sa valeur par défaut, sont les suivantes :

```
api.config.folder=/opt/IBM/TDI/V7.1/configs
```

Vous pouvez sélectionner un autre répertoire. Assurez-vous toutefois qu'il existe bien et qu'il contient un fichier de configuration qui sera utilisé par le module STS du serveur TDI. Créez par exemple un répertoire et copiez le modèle de fichier de configuration de TDI dans ce répertoire, comme suit :

```
# mkdir /opt/IBM/TDI/V7.1/configs
# cp /opt/IBM/FIM/examples/tdi_mappings/tdi_demo_mappings.xml
    /opt/IBM/TDI/V7.1/configs
```

4. Démarrez le serveur TDI en mode démon.

   Entrez la commande suivante pour démarrer le serveur sans prise en charge SSL :

   ```
   # /opt/IBM/TDI/V7.1/ibmdisrv -d
   ```

   Le serveur doit être en cours d'exécution. Les informations de journal sont disponibles dans le fichier `/root/TDI/logs/ibmdi.log`.

Résultats

Le module STS du serveur TDI est maintenant prêt à charger et à exécuter les chaînes d'assemblage.

Configuration du protocole SSL pour le module d'accréditation de Tivoli Directory Integrator

Le module STS Tivoli Directory Integrator agit en tant que client par rapport au serveur Tivoli Directory Integrator. La configuration des communications SSL entre ces deux entités peut s'effectuer de nombreuses manières. La configuration du serveur et du client s'effectue séparément.

Pour plus d'informations sur la sécurité en général et la configuration SSL, voir le centre de documentation Tivoli Directory Integrator. Plusieurs scénarios d'authentification sur l'API serveur sont disponibles, mais le présent document décrit uniquement les liaisons SSL authentifiées par processus réciproque. Ce scénario correspond au modèle de déploiement pris en charge pour les installations sécurisées impliquant le module STS Tivoli Directory Integrator.

Configuration SSL pour le serveur Tivoli Directory Integrator

Apprenez à configurer une connexion SSL mutuellement authentifiée pour le serveur Tivoli Directory Integrator versions 6.1,1, 7.0 et 7.1.

Pourquoi et quand exécuter cette tâche

Cette rubrique propose les informations suivantes :

- Une des méthodes de configuration SSL.
- Un scénario possible lors de la configuration SSL pour le serveur Tivoli Directory Integrator.

Pour obtenir des instructions plus détaillées sur la configuration, consultez le centre de documentation de Tivoli Directory Integrator :

- Pour Tivoli Directory Integrator version 6.1.1

  Centre de documentation de Tivoli Directory Integrator version 6.1.1. Voir la rubrique *Secure Sockets Layer (SSL) Support* du document *Administration Guide*.

- Pour Tivoli Directory Integrator version 7.0

  Centre de documentation de Tivoli Directory Integrator version 7.0. Voir la rubrique *Secure Sockets Layer (SSL) Support* du document *Installation and Administration Guide*.
- Pour Tivoli Directory Integrator version 7.1

  Centre de documentation de Tivoli Directory Integrator version 7.1. Voir la rubrique *Secure Sockets Layer (SSL) Support* du document *Installation and Administration Guide*.

Le serveur Tivoli Directory Integrator a besoin de deux ensembles d'informations dans le cadre d'une configuration SSL à authentification mutuelle.

- Une clé privé et un certificat pour le serveur
- Le certificat public ou le signataire accrédité par le client

Pour permettre l'authentification mutuelle de la prise en charge SSL sur le serveur Tivoli Directory Integrator, procédez comme suit :

Procédure

1. Enregistrez la clé privée et le certificat dans un fichier de clés Java, tel que `server_identity.jks`.

   L'alias de certificat de la clé privée est appelé `tdi_server` dans ce fichier. Dans cet exemple, le fichier `.jks` est créé à l'aide de l'utilitaire IBM iKeyman et requiert un mot de passe de fichier de clés.

   L'utilitaire ne génère toutefois pas de mot de passe de clé distinct pour la clé privée individuelle. Cet autre mot de passe de clé est important lors de la création du fichier de dissimulation du serveur Tivoli Directory Integrator. Dans cet exemple, le mot de passe du fichier de clés `passw0rd`, mais le mot de passe de la clé privée n'existe pas.
2. Enregistrez le certificat public ou le signataire accrédité du serveur dans un fichier de clés Java, tel que `server_signer.jks`.

   Remarque : L'alias de certificat du signataire accrédité n'est pas important pour cette configuration. Le mot de passe utilisé pour le fichier de clés est important. Dans cet exemple, nous utilisons la valeur `passw0rd`.
   Le nom distinctif (DN) du certificat client est important. Dans cet exemple, le nom distinctif du certificat client est le suivant :

   ```
   CN=tdi_client, O=ibm, C=US
   ```
3. Modifiez `solution.properties` pour la prise en charge SSL du serveur Tivoli Directory Integrator :

   Remarque : Les propriétés dépendent de la version du serveur Tivoli Directory Integrator. N'ajoutez pas de propriété qui ne figure pas dans le fichier `solution.properties` de la version du serveur Tivoli Directory Integrator utilisée.

   com.ibm.di.server.keystore *(version 6.1.1) ou* **api.keystore** *(versions 7.0 et 7.1)*

   Indique le fichier de clés contenant la clé privée et le certificat du serveur Tivoli Directory Integrator.

   Le fichier `server_identity.jks` doit se trouver dans le répertoire de solutions (`/root/TDI` pour notre exemple). Changez la valeur par défaut `testserver.jks` en `server_identity.jks`.

com.ibm.di.server.key.alias *(version 6.1.1) ou* **api.key.alias***(versions 7.0 et 7.1)*
Indique dans le fichier de clés du serveur l'alias de certificat qui représente la clé privée du serveur Tivoli Directory Integrator.

Changez la valeur par défaut *default of server* en `tdi_server`.

{protect}-api.keystore.password
Indique le mot de passe du fichier de clés dans la propriété `api.keystore`.

{protect}-api.key.password
Indique le mot de passe de l'alias de clé dans la propriété `api.key.alias`.

com.ibm.di.server.encryption.keystore
Indique le chiffrement des données du fichier de clés qui héberge la clé utilisée par le serveur Tivoli Directory Integrator.

com.ibm.di.server.encryption.key.alias
Indique l'alias de clé du fichier de clés de chiffrement.

com.ibm.di.server.encryption.keystoretype
Indique le type de fichier de clés qui héberge la clé de chiffrement du serveur Tivoli Directory Integrator.

com.ibm.di.server.encryption.transformation
Indique le nom de transformation de cryptographie utilisé pour le chiffrement. Cette valeur peut être définir sur *RSA* (chiffrement de clé publique) ou sur une transformation de clé secrète.

api.truststore
Définit un fichier de clés spécifiant les certificats de signataires accédités et l'autorité d'accréditation pour les clients de l'API de serveur.

Le fichier `client_signer.jks` doit se trouver dans le répertoire de solutions (`/root/TDI` pour notre exemple). Changez la valeur par défaut `testserver.jks` en `client_signer.jks`.

{protect}-api.truststore.pass
Indique le mot de passe du fichier de clés dans la propriété `api.truststore`.

Ajoutez le préfixe `{protect}-` pour chiffrer automatiquement le mot de passe à la prochaine exécution du serveur. Changez la valeur par défaut `{encr}-key_string` sur `passw0rd`.

api.remote.ssl.on
Définissez cette propriété sur `true` pour activer SSL.

4. Créez le fichier de dissimulation du serveur Tivoli Directory Integrator.

   Le fichier de dissimulation du serveur Tivoli Directory Integrator est `idisrv.sth`, et se trouve dans le répertoire de solutions. Il peut contenir un ou deux mots de passe.

   Le premier mot de passe ouvre le fichier de clés contenant l'identité du serveur (`server_identity.jks`). Le second (facultatif) est pour la clé elle-même dans ce fichier de clés. Si un seul mot de passe en indiqué, le nouveau mot de passe est supposé être identique au premier.

   Lors de la création d'un certificat d'auto-signature dans un fichier de clés au moyen de l'utilitaire IBM iKeyman, le mot de passe du fichier de clés est spécifié manuellement lors de la création de celui-ci. Toutefois, il n'existe *aucun* mot de passe pour la clé privée.

Vous devez créer le fichier de dissimulation du serveur Tivoli Directory Integrator avec un mot de passe du fichier de clés. Définissez ensuite le mot de passe de la clé privée sur une valeur nulle (chaîne vide), comme suit :

```
# /opt/IBM/TDI/V6.1.1/bin/createstash.sh passw0rd ""
```

5. Mettez à jour le registre du serveur Tivoli Directory Integrator afin qu'il reconnaisse le nom distinctif du client en tant qu'administrateur.

   Le serveur Tivoli Directory Integrator délivre une autorisation pour les requêtes d'API de serveur autorisées via un registre d'utilisateurs et les rôles assignés correspondants. Le registre par défaut est un fichier texte situé dans le répertoire :

   <répertoire_solutions>/serverapi/registry.txt

   Ajoutez le texte suivant au fichier `registry.txt` :

   ```
   [USER]
   [ID]:CN=tdi_client, O=ibm, C=US
   [ROLE]:admin
   [ENDUSER]
   ```

   Le texte doit comporter des valeurs identiques à celles de *issued to* et *issued by*. Pour vérifier ces valeurs, sélectionnez l'étiquette de `tdi_client.jks` dans *iKeyman* et cliquez sur **Afficher/modifier**.

   Pour une configuration plus approfondie du registre, consultez le centre de documentation de Tivoli Directory Integrator.

6. Démarrez le serveur Tivoli Directory Integrator et approuvez le message de démarrage suivant qui indique que le protocole SSL est en cours d'utilisation :

   ```
   # /opt/IBM/TDI/V6.1.1/ibmdisrv -d
   CTGDKD024I Remote API successfully started on port:1099,
   bound to:'SessionFactory'. SSL and Client Authentication
   are enabled.
   ```

Résultats

La configuration SSL côté serveur est terminée.

Configuration du protocole SSL pour le client Tivoli Directory Integrator

Il existe plusieurs manières de configurer le client Tivoli Directory Integrator pour établir une communication SSL à authentification mutuelle.

Pourquoi et quand exécuter cette tâche

Cette rubrique propose les informations suivantes :

- Une des méthodes de configuration SSL.
- Un scénario possible lors de la configuration SSL pour le client Tivoli Directory Integrator.

Pour obtenir des instructions plus détaillées sur la configuration, consultez le centre de documentation de Tivoli Directory Integrator :

- Pour Tivoli Directory Integrator version 6.1.1

  Centre de documentation de Tivoli Directory Integrator version 6.1.1. Voir la rubrique *Secure Sockets Layer (SSL) Support* du document *Administration Guide*.

- Pour Tivoli Directory Integrator version 7.0

  Centre de documentation de Tivoli Directory Integrator version 7.0. Voir la rubrique *Secure Sockets Layer (SSL) Support* du document *Installation and Administration Guide*.

- Pour Tivoli Directory Integrator version 7.1

  Centre de documentation de Tivoli Directory Integrator version 7.1. Voir la rubrique *Secure Sockets Layer (SSL) Support* du document *Installation and Administration Guide*.

Le module STS (Security Token Services) de Tivoli Directory Integrator fait office de client SSL, et peut fonctionner dans l'une ou l'autre des configurations suivantes :

- WebSphere Application Server JSSE

  Utilisation de la configuration de WebSphere Application Server pour la prise en charge SSL.

  Remarque : Vous devez utiliser cette option avec la version intégrée de WebSphere.
- Propriétés du système Java

  Spécification des propriétés du système Java pour déterminer le fichier de clés et le fichier de clés certifiées que doit utiliser l'API du serveur Tivoli Directory Integrator.

  Remarque : Cette option ne fonctionne pas avec la version intégrée de WebSphere.

Pour ces deux options, le client a besoin de deux ensembles d'informations dans le cadre d'une configuration SSL à authentification mutuelle :

- Une clé privée et une clé publique, nécessaires pour établir l'identité du client.
- Le certificat public ou le signataire accrédité par le serveur.

Pour configurer le protocole SSL côté client,procédez comme suit :

Procédure

1. Enregistrez la clé privée et le certificat dans un fichier de clés Java, par exemple `client_identity.jks`. L'alias de certificat de la clé privée est appelé `tdi_client` dans ce fichier.

   Remarque : Dans cet exemple, le fichier `.jks` est créé à l'aide de l'utilitaire IBM iKeyman et le mot de passe du fichier de clés est `passw0rd`. iKeyman n'attribue pas de second mot de passe à la clé. Pour démarrer la machine virtuelle Java, vous devez attribuer un mot de passe à la clé. Utilisez le même mot de passe que celui du fichier de clés.
2. Enregistrez le certificat public ou le signataire accrédité du serveur dans un fichier de clés Java, tel que `server_signer.jks`.

   Remarque : L'alias de certificat du signataire accrédité n'est pas important pour cette configuration, mais le mot de passe du fichier de clés est requis. Définissez le mot de passe sur `passw0rd`.
3. Modifiez le fichier de clés créé au moyen d'iKeyman à l'aide du paramètre d'outil de clé Java et attribuez un mot de passe à la clé, comme suit :

   ```
   # /opt/IBM/WebSphere/AppServer/java/bin/keytool -keypasswd
   -alias tdi_client -new passw0rd -keystore client_identity.jks
   -storepass passw0rd
   ```
4. Importez les deux fichiers de clés dans le référentiel de configuration WebSphere Tivoli Federated Identity Manager à l'aide du service de clés, pour y faire référence ultérieurement.

a. Dans la console d'administration, sélectionnez **Configuration du service de clés** > **Fichiers de clés** pour importerles fichiers `client_identity.jks` et `server_signer.jks`.
b. Sur le panneau de l'interface utilisateur de `client_identity.jks`, indiquez le nom de fichier de clés `tdi_client` et le type de clés de signature/ chiffrement.
c. Sur le panneau d'interface de `server_signer.jks`, indiquez le nom de fichier de clés `tdi_server` et un type de certificats d'autorité de certification.

Les fichiers du système de fichiers du référentiel de configuration de WebSphere se trouvent à l'emplacement suivant :

<racine_configuration>/itfim/*<domaine_fim>*/etc/jks/tdi_client.jks
<racine_configuration>/itfim/*<domaine_fim>*/etc/jks/tdi_server.jks

Cet exemple de configuration utilise la valeur `idp` pour `domaine_fim`.

5. Utilisez l'une des deux méthodes de configuration SSL côté client suivantes pour le module STS de Tivoli Directory Integrator :
   - «Configuration SSL côté client à l'aide de WebSphere JSSE»
   - «Configuration de SSL côté client à l'aide des propriétés système Java», à la page 175

   Remarque : Vous devez effectuer une seule des deux méthodes.

Configuration SSL côté client à l'aide de WebSphere JSSE

Utilisez WebSphere JSSE pour la configuration SSL côté client.

Pourquoi et quand exécuter cette tâche

Cette rubrique récapitule les informations décrites en détail aux emplacements suivants :

- Centre de documentation WebSphere Application Server.
- Rubrique developerWorks : *SSL, certificat et améliorations de la gestion des clés pour une sécurité renforcée dans WebSphere Application Server version 6.1*:

  http://www-128.ibm.com/developerworks/websphere/techjournal/0612_birk/0612_birk.html?ca=drs-

La configuration SSL du noeud final sortant dynamique ne peut pas être utilisée pour les raisons suivantes :

- Elle requiert l'utilisation de la classe WebSphere `JSSEHelper` par le client SSL pour définir des paramètres d'informations de connexion spécifiques.
- Tivoli Directory Integrator utilise uniquement les interfaces Java JSSE standard.

Par conséquent, vous devez modifier la configuration SSL du serveur qui exécute le composant d'exécution Tivoli Federated Identity Manager. Selon que vous exécutez le serveur d'applications en cluster ou en mode autonome, vous pouvez appliquer cette modification au niveau cellule ou au niveau noeud.

Cet exemple repose sur un serveur d'applications autonome et modifie le fichier de clés et le fichier de clés certifiées par défaut du noeud. Le fichier de clés par défaut du noeud est intitulé *NodeDefaultKeyStore* et le fichier de clés certifiées du noeud par défaut, *NodeDefaultTrustStore*.

Exécutez les tâches suivantes :

- Importez la clé privée et le certificat du client pour mettre à jour *NodeDefaultKeyStore*.
- Importez le certificat public du serveur pour mettre à jour *NodeDefaultTrustStore*.

Procédure

1. Utilisez la console d'administration WebSphere pour importer la clé privée et le certificat du client dans *NodeDefaultKeyStore*.
   a. Sélectionnez **Sécurité > Certificat SSL et gestion des clés > Magasins de clés et certificats > NodeDefaultKeyStore > Certificats personnels**.
   b. Cliquez sur **Importer** pour importer une autre clé et entrez les valeurs suivantes :

      Nom du fichier de clés
      `/opt/IBM/WebSphere/AppServer/profiles/idp/config/itfim/idp/etc/jks/tdi_client.jks`

      Type
      `JKS`

      Mot de passe du fichier de clés
      `passw0rd`

      Remarque : Cliquez sur **Obtenir les alias de fichier de clés**.

      Alias de certificat à importer
      `tdi_client`

      Alias de certificat importé
      `tdi_client`

   c. Une fois l'importation effectuée, la clé doit apparaître dans la colonne *Alias* sous la forme `tdi_client`. Sauvegardez la configuration WebSphere après le chargement de la clé.

   Pour pouvoir importer le certificat public du serveur dans le fichier *NodeDefaultTrustStore*, vous devez disposer du certificat du serveur dans un format de fichier simple, plutôt qu'au format JKS. Par exemple, au format PEM ASCII ou binaire DER. Utilisez IBM *iKeyman* ou *keytool* pour exporter le certificat public du serveur à partir du fichier :

   `<config_root>/itfim/<fim_domain>/etc/jks/tdi_server.jks`

   Vous pouvez par exemple exporter la clé publique via l'utilitaire iKeyman sous la forme d'un fichier PEM ASCII appelé :

   `/root/keys/tdi_server.arm`

2. Utilisez la console d'administration WebSphere pour importer le certificat public du serveur dans *NodeDefaultTrustStore*.
   a. Sélectionnez **Sécurité > Certificat SSL et gestion des clés > Magasins de clés et certificats > NodeDefaultTrustStore > Certificats de signature**.
   b. Cliquez sur **Ajouter** pour ajouter un certificat.

      Le panneau Ajout de certificat de signataire s'ouvre.

   c. Entrez les valeurs suivantes :

      Alias
      `tdi_server`

      Nom de fichier
      `/root/keys/tdi_server.arm`

Type de données
Données ASCII codées en base 64

d. Le certificat intitulé `tdi_server` doit maintenant figurer dans la liste des certificats. Sauvegardez la configuration WebSphere après avoir validé ce changement.

3. Cliquez sur **Extraire d'un port**.
4. Accédez à la section des propriétés générales, qui est composée des zones **Hôte**, **Port** et **Configuration SSL pour une connexion sortante**.
5. Redémarrez l'instance de WebSphere Application Server.

Résultats

Le client est configuré pour SSL.

Configuration de SSL côté client à l'aide des propriétés système Java

Utilisez les propriétés système Java puor sélectionner les fichiers de clés et certificats pour les communications SSL.

Pourquoi et quand exécuter cette tâche

Les propriétés système Java pour le protocole SSL côté client sont décrites dans les documents suivants :

- Pour Tivoli Directory Integrator version 6.1.1

  Centre de documentation de Tivoli Directory Integrator version 6.1.1. Voir la rubrique *Remote Server API* du document *Administration Guide*.
- Pour Tivoli Directory Integrator version 7.0

  Centre de documentation de Tivoli Directory Integrator version 7.0. Voir la rubrique *Server API Access Security* du document *Installation and Administration Guide*.
- Pour Tivoli Directory Integrator version 7.1

  Centre de documentation de Tivoli Directory Integrator version 7.1. Voir la rubrique *Server API Access Security* du document *Installation and Administration Guide*.

Remarque : La configuration du client à l'aide des propriétés système Java n'estp as disponible pour les installations WebSphere imbriquées.

Ces propriétés système Java peuvent servir à sélectionner les fichiers de clés et les certificats pour les communications SSL :

`api.client.ssl.custom.properties.on`
Indique à l'API du seveur Tivoli Directory Integrator d'utiliser des propriétés personnalisées pour la configuration du fichier de clés et du fichier de clés sécurisées plutôt que la configuration JSSE. Par exemple : `true`.

`api.client.keystore`
Spécifie le fichier de clés contenant le certificat client. Par exemple :
`${USER_INSTALL_ROOT}/config/itfim/idp/etc/jks/tdi_client.jks`

`api.client.keystore.pass`
Indique le mot de passe du fichier spécifié dans `api.client.keystore`. Par exemple, `passw0rd`.

api.client.key.pass
Indique le mot de passe de clé réelle dans `api.client.keystore`.

N'indiquez pas cette valeur, étant donné que l'utilitaire *keytool* permet de définir pour la clé un mot de passe identique à celui du fichier de clés.

api.truststore
Indique le fichier contenant le certificat public du serveur Tivoli Directory Integrator. Par exemple :

```
${USER_INSTALL_ROOT}/config/itfim/idp/etc/jks/tdi_server.jks
```

api.truststore.pass
Indique le mot de passe du fichier spécifié dans `api.truststore`. Par exemple, `passw0rd`.

Utilisez la console d'administration WebSphere pour mettre à jour les paramètres de démarrage de la machine virtuelle Java du serveur.

Procédure

1. Sélectionnez **Serveurs** > **Serveurs d'applications** > **server1** > **Gestion des processus et Java** > **Définition des processus** > **Machine virtuelle Java**.
2. Mettez à jour les propriétés :

   Arguments de la machine virtuelle Java génériques :
   -api.client.ssl.custom.properties
3. Redémarrez l'instance de WebSphere Application Server.

Résultats

Le client est configuré pour SSL.

Création d'un module de mappage personnalisé

La création d'un module de mappage personnalisé est une procédure de programmation intensive impliquant l'écriture d'une classe Java et son installation dans le répertoire des plug-ins de votre domaine.

Avant de commencer

Pour créer un module de mappage personnalisé, vous devez être familiarisé avec la structure des modules du service d'accréditation Tivoli Federated Identity Manager, ainsi que les procédures permettant de les créer et de les ajouter à votre environnement.

Pourquoi et quand exécuter cette tâche

Consultez des informations complémentaires sur les modules du service d'accréditation dans le document suivant :

- *Federated Identity Management and Web Services Security with IBM Tivoli Security Solutions* (SG24-6394-01). Ce guide est disponible au format PDF (Portable Document Format) à l'adresse http://www.redbooks.ibm.com/redbooks/pdfs/sg246394.pdf ou au format HTML (Hypertext Markup Language) à l'adresse http://www.redbooks.ibm.com/redbooks/SG246394/
- Un article developerWorks intitulé *Tivoli Federated Identity Manager: Implementing and deploying custom trust modules* est disponible à l'adresse http://www-128.ibm.com/developerworks/tivoli/library/t-sts-custom/

Ajout d'un module de mappage personnalisé

Pour procéder à l'ajout d'un module de mappage personnalisé que vous avez créé, vous devez d'abord le définir en tant que nouveau type de module dans l'environnement Tivoli Federated Identity Manager.

Avant de commencer

Vous devez écrire une classe Java pour un nouveau type de module et installer la classe dans le répertoire des plug-in de votre domaine. Vous pouvez ensuite utiliser les instructions ci-dessous pour créer une entrée de type de module dans la console.

Pourquoi et quand exécuter cette tâche

Cette tâche est nécessaire uniquement lorsque le module de transformation XSL fourni avec Tivoli Federated Identity Manager n'est pas conforme à la configuration de votre déploiement.

Procédure

1. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Gestion des noeuds d'exécution**. L'écran Gestion des noeuds d'exécution s'ouvre.
2. Cliquez sur le bouton **Publier les plug-in**.
3. Lorsque vous y êtes invité, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**. Le nouveau type de module s'affiche dans la liste Type de module.

Que faire ensuite

Poursuivez avec la tâche consistant à ajouter une instance du fichier de mappage dans «Ajout d'une instance de module de mappage personnalisé».

Ajout d'une instance de module de mappage personnalisé

Après avoir créé votre module de mappage et l'avoir ajouté en tant que type de module, vous devez créer une instance de ce type de module pour l'utiliser dans l'environnement Tivoli Federated Identity Manager.

Avant de commencer

Assurez-vous d'avoir configuré les tâches suivantes avant de passer à ces instructions :

- «Création d'un module de mappage personnalisé», à la page 176
- «Ajout d'un module de mappage personnalisé»

Pourquoi et quand exécuter cette tâche

La console est dotée d'un assistant qui vous guidera tout au long de l'ajout de l'instance de module.

Procédure

1. Sélectionnez **Tivoli Federated Identity Manager** > **Configuration du service d'accréditation** > **Instances de module**. L'écran Instances de module affiche des

instances de module créées par défaut. Il contient également toutes les instances de module que vous avez ajoutées.

2. Cliquez sur **Créer**. L'écran Type de jeton affiche les types de module que vous avez définis. La liste comprend les types de jeton par défaut, ainsi que tous ceux que vous avez définis à votre convenance.
3. Sélectionnez un type de jeton.
4. Cliquez sur **Suivant**. L'assistant Instances de module ouvre l'écran Nom d'instance du module.
5. Entrez les valeurs des propriétés demandées.
6. Cliquez sur **Terminer**. Pour obtenir une description de chaque zone, consultez l'aide en ligne.

Que faire ensuite

Le nouveau fichier de mappage est disponible dans la listes des modules que vous pouvez sélectionner lors de l'établissement d'une fédération.

Chapitre 15. Fédérations SAML : présentation

Le langage SAML (Security Assertion Markup Language) est un langage XML normalisé destiné à l'échange d'informations de connexion unique. Il s'appuie entre autres sur la technologie SOAP pour échanger des messages XML sur les réseaux informatiques. L'échange des messages XML s'effectue par le biais d'une série de requêtes et de réponses. Dans ce processus, l'un des partenaires de la fédération envoie un message de demande à l'autre partenaire de la fédération. Le partenaire récipiendaire envoie alors immédiatement un message de réponse au partenaire ayant émis la requête.

Tivoli Federated Identity Manager prend en charge les spécifications de sécurité OASIS suivantes lors de l'échange d'informations dans une fédération:

- SAML 1.0 et 1.1 (1.x)
- SAML 2.0

Les spécifications SAML incluent des descripteurs pour établir une fédération, initialiser et gérer la connexion unique. Les descripteurs suivants spécifient la structure, le contenu des messages, et le mode de communication des messages entre les partenaires et les utilisateurs.

Assertions
Jetons au format XML utilisés pour transférer les informations d'identité des utilisateurs, telles que les données d'authentification, attributs et autorisations d'utilisation contenues dans les messages.

Protocoles
Types de messages de requête et de réponse utilisés pour obtenir les données d'authentification et gérer les identités.

Liaisons
Méthode de communication utilisée pour le transport des messages.

Profils
Combinaisons des protocoles, assertions et liaisons utilisées conjointement pour créer une fédération et activer la connexion unique fédérée.

Lors de l'utilisation de Tivoli Federated Identity Manager, votre partenaire et vous devez effectuer les tâches suivantes :

- Utiliser la même spécification SAML (1.0, 1.1 ou 2.0).
- Trouver un accord sur les protocoles, liaisons et profils à utiliser.

Les rubriques suivantes décrivent brièvement la manière dont les spécifications SAML 1.x et SAML 2.0 sont utilisées dans Tivoli Federated Identity Manager. Toutefois, ces descriptions n'incluent pas de manière exhaustive les détails de ces spécifications. Pour plus de détails, voir les documents relatifs à la spécification OASIS à l'adresse http://www.oasis-open.org/specs/index.php.

SAML 1.x

Tivoli Federated Identity Manager prend en charge les spécifications SAML 1.0 et SAML 1.1. Ces spécifications sont communément référencées sur l'appellation SAML 1.x.

Si vous décidez, en accord avec partenaire, d'utiliser la norme SAML 1.x dans votre fédération, vous devez comprendre les principes du support SAML 1.x décrits à la rubrique Tivoli Federated Identity Manager.

Assertions

Les assertions créées par Tivoli Federation Identity Manager contiennent des instructions d'authentification permettant de confirmer que le principal (c'est-à-dire l'entité qui émet la demande d'accès) a été authentifié. Les assertions peuvent également comporter des attributs relatifs à l'utilisateur, que le fournisseur d'identité souhaite mettre à la disposition du fournisseur de services.

Les assertions sont généralement transmises au fournisseur de services par le fournisseur d'identité.

Les variables suivantes contrôlent le contenu des assertions créées par Tivoli Federated Identity Manager :

- La spécification (SAML 1.0 ou 1.1) que vous sélectionnez lorsque vous établissez une fédération.
- Les définitions utilisées dans la méthode de mappage d'identité TFIM que vous configurez.

Le mappage d'identité indique la manière dont les identités sont mappées entre les partenaires de la fédération.

La méthode de mappage d'identité Tivoli Federated Identity Manager peut être un module de mappage personnalisé ou un fichier de transformation XSL.

Protocole

Dans Tivoli Federated Identity Manager, SAML 1.x gère les demandes d'authentification en s'appuyant sur un protocole simple de requêtes/réponses.

Liaison

SAML 1.x peut utiliser le protocole HTTP de base (via des redirections du navigateur) ou le protocole SOAP pour le transport des messages. Le *profil* utilisé dans la fédération spécifie plus en détails la manière dont s'effectue la communications des messages.

Profils

Deux options sont spécifiées par SAML 1.x pour les profils :

Artefact du navigateur (Browser Artifact)
L'artefact du navigateur utilise des communications SOAP (également appelées canal de retour SOAP) pour échanger un artefact au cours de l'établissement et de l'utilisation d'une session sécurisée entre un fournisseur d'identité, un fournisseur de services et un client (navigateur).

POST du navigateur (Browser POST)
Le profil POST du navigateur utilise un formulaire qui renvoie l'action à lui-même (self-posting form) pour échanger un artefact au cours de l'établissement et de l'utilisation de la session sécurisée entre un fournisseur d'identité, un fournisseur de services et un client (navigateur).

Tivoli Federated Identity Manager prend en charge l'artefact de navigateur par défaut lorsque vous sélectionnez SAML 1.0 ou SAML 1.1 en tant que profil de fédération. Toutefois, vous pouvez utiliser le POST du navigateur dans votre fédération en l'appliquant à des partenaires choisis. A titre d'exemple, si vous être un fournisseur de services, vous pouvez spécifier que votre fournisseur d'identité partenaire doit utiliser le POST du navigateur, lors de la configuration de ce partenaire. Si vous êtes un fournisseur d'identité, vous pouvez activer l'extension IBM PROTOCOL lors de la configuration d'une fédération SAML 1.x.

L'adresse URL servant à déclencher la connexion unique varie selon que le fournisseur d'identité utilise ou non cette extension. Pour plus d'informations sur les adresses URL, voir «Adresse URL initiale SAML 1.x», à la page 843.

SAML 2.0

La spécification SAML 2.0 a permis d'introduire davantage de souplesse que les spécifications SAML 1.x antérieures.

Assertions

Les assertions créées par Tivoli Federated Identity Manager contiennent des instructions d'authentification. Ces instructions d'authentification confirment que le principal (à savoir l'entité qui émet la demande d'accès) a été authentifié. Les assertions peuvent également comporter des attributs relatifs à l'utilisateur, que le fournisseur d'identité souhaite mettre à la disposition du fournisseur de services.

Les assertions sont en général transmises au fournisseur de services par le fournisseur d'identité.

Le contenu des assertions créées est contrôlé par la spécification (SAML 2.0). Sélectionnez ces assertions lorsque vous établissez une fédération. Vous pouvez également sélectionner ces assertions en fonction des définitions utilisées dans la méthode de mappage d'identité Tivoli Federated Identity Manager que vous configurez.

La méthode de mappage d'identité peut être un module de mappage personnalisé ou un fichier de transformation XSL. Le mappage d'identité définit également la manière dont les identités sont mappées entre les partenaires de la fédération.

Protocoles

La spécification SAML 2.0 définit plusieurs protocoles de requête et de réponse, qui renvoient tous à l'action communiquée dans le message. Les protocoles SAML 2.0 pris en charge par Tivoli Federated Identity Manager sont les suivants :

- Requête d'authentification
- Single logout (SLO, Déconnexion unique)
- Résolution d'artefact
- Gestion des identificateurs de nom

Liaisons

Lors de l'utilisation de SAML 2.0 dans Tivoli Federated Identity Manager, vous avez plusieurs options de liaison. Ces options spécifient la manière dont les messages peuvent être transportés :

Réacheminement HTTP

La redirection HTTP du navigateur permet la transmission de messages de protocole SAML à l'intérieur de paramètres d'URL. Les demandeurs et les répondeurs SAML peuvent ainsi communiquer à l'aide d'un agent d'utilisateur HTTP employé comme intermédiaire.

Ceci peut être nécessaire si les entités de communication n'ont pas d'accès direct de communication. L'intermédiaire peut également être nécessaire si le répondeur requiert une interaction avec un agent d'utilisateur tel qu'un agent d'authentification.

La redirection HTTP est parfois appelée "redirection de navigateur" dans le cadre d'opérations de connexion unique. Ce profil est sélectionné par défaut.

HTTP POST

Le profil POST HTTP permet la transmission de messages de protocole SAML au format HTML codé en base64. Les demandeurs et les répondeurs SAML peuvent ainsi communiquer à l'aide d'un agent d'utilisateur HTTP employé comme intermédiaire.

L'agent peut être nécessaire si les entités de communication n'ont pas de chemin direct de communication. L'intermédiaire peut également être nécessaire si le répondeur requiert une interaction avec un agent d'utilisateur tel qu'un agent d'authentification.

HTTP POST est parfois appelé POST du navigateur, notamment quand il est utilisé dans les opérations de connexion unique. Ce profil utilise un formulaire qui renvoie l'action à lui-même (self-posting form) pour échanger un artefact au cours de l'établissement et de l'utilisation d'une session sécurisée entre un fournisseur d'identité, un fournisseur de services et un client (navigateur).

Artefact HTTP

L'artefact HTTP est une liaison dans laquelle une requête ou une réponse SAML (ou les deux) sont transmises par référence à un identificateur unique appelé artefact.

Une liaison séparée, telle qu'une liaison SOAP, est utilisée pour échanger l'artefact du message de protocole. Les demandeurs et les répondeurs SAML peuvent ainsi communiquer à l'aide d'un agent d'utilisateur HTTP employé comme intermédiaire.

Ce paramètre est utilisé lorsqu'il n'est pas recommandé d'exposer le contenu du message dans l'intermédiaire.

L'artefact HTTP est parfois appelé Artefact du navigateur, notamment quand il est utilisé dans les opérations de connexion unique. L'artefact HTTP utilise un canal de retour SOAP. Le canal de retour SOAP est utilisé pour échanger un artefact au cours de l'établissement et de l'utilisation d'une session sécurisée entre un fournisseur d'identité, un fournisseur de services et un client (navigateur).

SOAP

La liaison SOAP est une liaison qui utilise le protocole SOAP (Simple Object Access Protocol) pour les communications.

Le choix de la liaison dépend du profil que vous choisissez d'utiliser dans votre fédération.

Profils

Tivoli Federated Identity Manager prend en charge la configuration du profil de connexion unique de chaque partenaire pris séparément. Les profils prise en charge sont les suivants :

Connexion unique du navigateur Web

Le profil SSO du navigateur Web est constitué par le regroupement de l'artefact du navigateur et les profils POST du navigateur qui ont été introduits dans SAML 1.x.

Grâce à ce profil, un message de demande d'authentification est envoyé par un fournisseur de services à un fournisseur d'identité. Un message de réponse contenant une assertion SAML est envoyé au fournisseur de services depuis le fournisseur d'identité. D'autres messages sont envoyés liés à la résolution d'artefact, si cette liaison est utilisée.

Ce profil comprend des options concernant l'initialisation du flux de messages et le transport des messages :

Initialisation des messages

Le flux de messages peut être initialisé par le fournisseur de services ou le fournisseur d'identité.

Lorsque le fournisseur d'identité lance le flux de messages, un paramètre **RelayState** peut être transmis dans la réponse non sollicitée envoyée par le fournisseur d'identité au fournisseur de services. Ce paramètre contient la valeur codée dans l'URL de l'élément Cible fourni dans l'adresse URL initiale du service de connexion unique (fournisseur d'identité).

Liaisons

Dans un environnement Tivoli Federated Identity Manager, les liaisons suivantes peuvent être utilisées dans le profil SSO du navigateur Web :

- HTTP Redirect (disponible uniquement dans une configuration de fournisseur d'identité)
- HTTP POST
- Artefact HTTP

Le choix de la liaison dépend du type de messages envoyés. Par exemple, un message de requête d'authentification peut être envoyé depuis un fournisseur de services à un fournisseur d'identité. Le message de réponse peut être envoyé par un fournisseur d'identité à un fournisseur de services via une liaison HTTP POST ou un artefact HTTP. Il n'est pas obligatoire pour un couple de partenaires au sein d'une fédération d'utiliser la même liaison.

Options

Le profil SLO du navigateur Web fourni par Tivoli Federated Identity Manager propose également l'option suivante :

Proxy client amélioré Cette option de profil permet à un client ou proxy amélioré (ECP) de communiquer avec un fournisseur d'identité et un fournisseur de services pour le compte d'un utilisateur (client).

Par exemple, un utilisateur peut demander une ressource auprès d'un fournisseur de services. Le fournisseur de services

ne sait pas nécessairement à quel fournisseur d'identité il doit accéder pour authentifier l'utilisateur.

Grâce à l'option de profil ECP, le fournisseur de services peut contacter l'ECP, qui est en mesure de localiser le fournisseur d'identité approprié et d'accéder à celui-ci. Le profil d'ECP prend en charge les liaisons SOAP et les liaisons SOAP inverses (PAOS) lors du traitement des requêtes d'authentification.

Déconnexion unique (SLO)

Le profil SLO permet de mettre fin à toutes les sessions de connexion actives d'un utilisateur donné dans la fédération. Un utilisateur qui utilise la connexion unique sur une fédération établit des sessions avec plusieurs participants.

Les sessions sont gérées par une autorité de sessions, généralement un fournisseur d'identité. Lorsque l'utilisateur souhaite fermer toutes les sessions avec tous les participants de session, l'autorité de sessions peut utiliser le profil SLO pour arrêter de façon globale toutes les sessions actives.

Initialisation des messages

Le flux de messages peut être initialisé par le fournisseur de services ou le fournisseur d'identité.

Liaisons

Dans un environnement Tivoli Federated Identity Manager, les liaisons suivantes peuvent être utilisées dans le profil SLO :

- Réacheminement HTTP
- HTTP POST
- Artefact HTTP
- SOAP

Gestion des identificateurs de nom

Le profil Gestion des identificateurs de nom gère les identités utilisateur échangées entre fournisseurs d'identité et fournisseurs de services.

Ce profil permet aux fournisseurs d'identité d'informer les fournisseurs de services. Les fournisseurs de services sont informés de toute modification apportée au contenu ou au format de l'identité d'un utilisateur donné (principal).

Le profil permet aux fournisseurs de service de spécifier des *alias* uniques pour le principal. Les fournisseurs de service peuvent également envoyer ces alias au fournisseur d'identité à utiliser à la place du nom principal.

Le profil active également le fournisseur. Le profil informe son partenaire lorsqu'il décide de ne plus émettre ni d'accepter des messages utilisant l'identité du principal.

Pour gérer les alias, Tivoli Federated Identity Manager utilise une fonction appelée *service d'alias*. Le service d'alias enregistre et extrait les alias liés à une identité fédérée. Les alias peuvent être exploités de différentes manières :

Alias persistants

Lors de l'utilisation d'alias persistants, l'identité de l'utilisateur est fédérée par le fournisseur d'identité dans l'identité de l'utilisateur au niveau du fournisseur de service. Un identificateur de nom SAML persistant est utilisé. L'utilisateur reste indéfiniment membre

de la fédération, c'est-à-dire jusqu'à ce qu'une requête de suppression de la fédération soit émise.

Alias transitoires
Lorsque des alias transitoires sont utilisés, un identificateur temporaire est utilisé pour fédérer le fournisseur d'identité et le fournisseur de services. Un identificateur temporaire est utilisé seulement pour la durée de vie de la session de connexion unique de l'utilisateur.

Dans un environnement Tivoli Federated Identity Manager, le stockage et l'extraction d'alias s'effectuent en provenance et en direction des types de référentiel suivants :

- Une base de données LDAP.
- Une base de données relationnelle avec prise en charge JDBC.

Durant la configuration de Tivoli Federated Identity Manager, vous pouvez configurer votre environnement en vue d'exploiter l'un de ces types de référentiel.

Initialisation des messages
Le flux de messages peut être initialisé par le fournisseur de services ou le fournisseur d'identité.

Liaisons
Les liaisons suivantes peuvent être utilisées dans le profil Gestion d'identificateurs de nom :

- Réacheminement HTTP
- HTTP POST
- Artefact HTTP
- SOAP

Reconnaissance du fournisseur d'identité
Le profil Reconnaissance du fournisseur d'identité est employé par les fournisseurs de services pour reconnaître le fournisseur d'identité employé par un utilisateur (principal) lors d'une connexion unique du navigateur Web.

Certains déploiements possèdent plusieurs fournisseurs d'identité, et le fournisseur de services doit être en mesure de déterminer quel fournisseur d'identité est utilisé par un utilisateur principal.

Le profil de reconnaissance du fournisseur d'identité utilise un cookie. Le cookie est créé dans un domaine commun aux fournisseurs d'identité et fournisseurs de service dans un déploiement donné. Le cookie, appelé *cookie de domaine commun*, contient la liste des fournisseurs d'identité.

Lors de la configuration de votre fédération au moyen de la console Tivoli Federated Identity Manager, vous disposez des options de profil suivantes :

Basique : Connexion unique de navigateur Web, déconnexion unique
Ce paramètre active les profils et liaisons suivants :

- Connexion unique de navigateur Web avec liaisons HTTP POST et HTTP Artifact.
- Déconnexion SLO avec liaisons HTTP POST et HTTP Artifact.

Typique : Connexion unique de navigateur Web, déconnexion unique et identificateurs de nom
Ce paramètre active les profils et liaisons suivants :

- Connexion unique de navigateur Web avec liaisons HTTP POST et HTTP Artifact.
- Déconnexion SLO avec liaisons HTTP POST et HTTP Artifact.
- Client ou proxy évolué
- Gestion des identificateurs de nom avec liaisons HTTP POST et HTTP Artifact.

Activer tous les profils et toutes les liaisons
Ce paramètre active tous les profils et liaisons disponibles :

- Connexion unique de navigateur Web avec liaisons HTTP POST, HTTP Artifact et HTTP Redirect.

  Remarque : La liaison HTTP Redirect est disponible uniquement dans une configuration de fournisseur d'identité.
- Client ou proxy amélioré
- Déconnexion SLO avec liaisons HTTP Redirect, HTTP POST et HTTP Artifact.
- Gestion des identificateurs de nom avec liaisons HTTP Redirect, HTTP POST, HTTP Artifact et SOAP
- Reconnaissance du fournisseur d'identité

Manuel : Choisissez des profils et des liaisons individuels
L'ensemble des profils pris en charge et des liaisons disponibles est présenté de manière à ce que vous puissiez choisir les options souhaitées.

Liaison de compte

Dans SAML 2.0, la liaison de compte permet à un utilisateur de relier un compte de fournisseur d'identité à un fournisseur de service. La liaison survient lors de l'initiation de la connexion unique sur le fournisseur d'identité et le fournisseur de service. Dans les deux cas, la liaison de compte requiert l'authentification de l'utilisateur au niveau du fournisseur de service et du fournisseur d'identité.

Un administrateur peut activer cette fonction dans le panneau de configuration du partenaire. Si cette fonction est activée, l'utilisateur doit s'authentifier dans le fournisseur de service à la réception d'un alias persistant. L'alias ne peut pas avoir été lié précédemment à un compte dans le fournisseur de service pour que l'authentification fonctionne.

Une fois que l'utilisateur est authentifié, l'implémentation SAML 2.0 stocke l'alias au niveau du fournisseur de service et service d'alias et établit une liaison de compte.

Gestion d'un alias inconnu

SAML 2.0 prend en charge les alias pour communiquer les identités d'utilisateur entre les partenaires.

Un administrateur peut configurer les paramètres de partenaire SAML 2.0 de sorte à gérer un alias inconnu d'une des manières suivantes :

- La page d'authentification affiche une page d'erreur lorsque le fournisseur de services ne connaît pas l'alias reçu du fournisseur d'identité. Ce paramètre représente la valeur par défaut lorsque vous

  - Ne sélectionnez pas **Imposer l'authentification pour procéder à la liaison des comptes**.
  - Ne sélectionnez pas **Mapper les identificateurs de nom inconnu au nom d'utilisateur anonyme**.
- L'implémentation SAML 2.0 mappe l'identité de l'utilisateur sur le compte utilisateur par défaut. Un compte invité établit la session de connexion unique. Ce paramètre nécessite que vous
  - Ne sélectionnez pas **Imposer l'authentification pour procéder à la liaison des comptes**.
  - Sélectionnez **Mapper les identificateurs de nom inconnu au nom d'utilisateur anonyme**.
- L'utilisateur doit s'authentifier au niveau du fournisseur de service, ce qui active la liaison de comptes. Ce paramètre nécessite que vous
  - Sélectionniez **Imposez l'authentification pour procéder à la liaison des comptes**.
  - Ne sélectionnez pas **Mapper les identificateurs de nom inconnu au nom d'utilisateur anonyme**.

Chapitre 16. Noeuds finals SAML et adresses URL

Les communications échangées au sein d'une fédération sont établies entre des noeuds finals sur les serveurs des fournisseurs d'identité et fournisseurs de services partenaires.

Dans un environnement Tivoli Federated Identity Manager, les noeuds finals appartiennent à deux catégories :

- Noeuds finals définis par la spécification de la fédération (par exemple SAML 1.x ou SAML 2.0) et utilisés pour les communications inter-partenaires.
- Noeuds finals auxquels les utilisateurs peuvent accéder pour initier une activité de connexion unique.

Tous les noeuds finals sont accessibles via des adresses URL. La syntaxe des URL dépend spécifiquement du motif de l'accès, ainsi que l'auteur de cet accès (partenaire ou utilisateur final).

Adresses URL de communication entre les partenaires

Les adresses URL employées pour les communications inter-partenaires, par exemple lors de l'échange de requêtes, sont définies collectivement dans les fédérations SAML 1.x et SAML 2.0 par la notion d'*URL de noeud final*, ou individuellement par le nom du protocole, ainsi que la liaison ou le service dont elles dépendent. Les administrateurs responsables de l'installation, de la configuration et de la gestion de l'environnement Tivoli Federated Identity Manager, ainsi que des communications inter-partenaires au sein de cet environnement, peuvent visualiser des références à ces URL de noeud final, ce qui peut les aider à comprendre leur utilité. Voir «Noeuds finals et adresses URL SAML 1.x», à la page 190 ou «Noeuds finals SAML et adresses URL SAML 2.0», à la page 193.

Adresses URL pour l'accès utilisateur

Alors que les spécifications SAML définissent les noeuds finals pour les communications inter-partenaires, elles ne contiennent pas ou peu d'informations sur les noeuds finals ou les méthodes que l'utilisateur final doit employer pour initier des actions de connexion unique. Tivoli Federated Identity Manager prend en charge des URL spécifiques pour l'initiation d'actions de connexion unique par les utilisateurs finals.

Dans une fédération SAML 1.x, le processus de connexion unique est toujours initié au niveau du *service de transfert inter-sites*. La méthode selon laquelle la requête parvient à ce noeud final n'est pas précisée dans la spécification SAML. La syntaxe d'URL du service de transfert inter-sites dans un environnement Tivoli Federated Identity Manager est décrite à la rubrique «Adresse URL initiale SAML 1.x», à la page 843.

Dans une fédération SAML 2.0, les actions de connexion unique peuvent être initiées au niveau du site du fournisseur d'identité, ou celui du fournisseur de services. Les adresses URL auxquelles les utilisateurs peuvent se connecter pour déclencher une action de connexion unique dépendent spécifiquement de cette action (par exemple, l'établissement d'une connexion unique fédérée, d'une déconnexion unique ou d'une liaison de compte), ainsi que du site d'origine de

l'action (fournisseur d'identité ou fournisseur de services). Dans un environnement Tivoli Federated Identity Manager, les adresses URL pouvant servir au déclenchement d'actions de connexion unique sont référencées comme *adresse URL initiales de profil*. Les architectes et développeurs d'applications, qui conçoivent et implémentent l'interaction des utilisateurs avec le processus de connexion unique, doivent comprendre la fonction des adresses URL initiales de profil. Voir «Adresses URL initiales de profil SAML 2.0», à la page 845.

Noeuds finals et adresses URL SAML 1.x

Plusieurs noeuds finals sont configurés sur votre serveur point de contact afin de permettre une communications entre vous et votre partenaire. La configuration de ces noeuds finals a lieu en même temps que celle de votre fédération dans Tivoli Federated Identity Manager. Les noeuds finals sont accessibles via des URL et sont utilisés par les partenaires au sein de la fédération.

Si vous êtes responsable de l'installation, de la configuration ou de la gestion d'une fédération dans Tivoli Federated Identity Manager, il peut être utile de vous familiariser avec les noeuds finals SAML 1.x et les adresses URL.

Les noeuds finals suivants sont utilisés dans une fédération SAML 1.x.

Serveur point de contact
Noeud final du serveur point de contact sur lequel les communications ont lieu. La syntaxe de l'URL du serveur point de contact est la suivante :

`https://`*`nom_hôte:numéro_port`*

Où :

https HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est le 9443 sur WebSphere Application Server si le protocole SSL est activé, ou le 9080 dans le cas contraire.

Vous êtes invité à indiquer l'URL de votre serveur point de contact lors de la configuration de votre fédération. Une fois la configuration terminée, l'URL de votre serveur point de contact comporte le suffixe `/sps`, de sorte que la syntaxe de l'URL configurée pour le serveur point de contact est la suivante :

`https://`*`nom_hôte:numéro_port`*`/sps`

Le suffixe `/sps` indique que l'URL est définie pour les services de connexion unique.

Service de transfert inter-sites
Noeud final du serveur point de contact du fournisseur d'identité sur lequel démarre le processus de demande de connexion. Il s'agit de l'emplacement auquel sont envoyées les demandes de connexion unique. SAML n'indique pas comment les demandes arrivent sur ce noeud final.

Si vous êtes un fournisseur d'identité utilisant Tivoli Federated Identity Manager, la méthode utilisée est fonction de la procédure de connexion des utilisateurs et de l'emplacement dans lequel les utilisateurs ouvrent une

session. Par exemple, si les utilisateurs se connectent au site Web du fournisseur de services partenaire, votre fournisseur de services partenaire a besoin de l'URL de votre service de transfert inter-sites.

Il doit également configurer un certain type de réacheminement permettant aux utilisateurs d'accéder à partir de leur site à votre page de connexion.

L'adresse URL repose sur celle que vous avez spécifiée pour votre serveur point de contact. La syntaxe est la suivante :

`https://nom_hôte:numéro_port/sps/nom_fédération/samlxx/login`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Le numéro de port où les communications prennent effet sur le serveur.

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml*xx*
Version de SAML configurée pour la fédération. Les valeurs peuvent être les suivantes :

- saml (pour SAML 1.0)
- saml11 (pour SAML 1.1)

login

Désignation du type de noeud final qui utilise le port. Le type **login** est utilisé pour le service de transfert inter-sites dans les fédérations SAML 1.x.

Ce noeud final est utilisé uniquement dans les configurations de fournisseur d'identité et est défini automatiquement à votre place lors de la configuration de votre fédération.

Service de résolution des artefacts
Noeud final du serveur point de contact du fournisseur d'identité sur lequel des artefacts sont échangés pour les assertions. Ce noeud final est l'emplacement dans lequel les partenaires de la fédération communiquent. Il est parfois désigné par le terme de noeud final *Noeud final SOAP* du serveur point de contact du fournisseur d'identité.

Remarque : Il se peut aussi que vous connaissiez ce noeud final sous le nom de *service répondeur*.

L'adresse URL repose sur celle que vous avez spécifiée pour votre serveur point de contact. La syntaxe est la suivante :

`https://nom_hôte:numéro_port/sps/nom_fédération/samlxx/soap`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est 9444.

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml*xx*
Version de SAML configurée pour la fédération. Les valeurs peuvent être les suivantes :

- saml (pour SAML 1.0)
- saml11 (pour SAML 1.1)

soap

Désignation du type de noeud final qui utilise le port. Le type **soap** est utilisé pour le service de résolution des artefacts dans les fédérations SAML 1.x.

Ce noeud final est utilisé uniquement dans les configurations de fournisseur d'identité et est défini automatiquement à votre place lors de la configuration de votre fédération.

Service d'assertion client
Noeud final du serveur point de contact du fournisseur de services qui reçoit des assertions ou des artefacts. Ce noeud final est l'emplacement dans lequel les partenaires de la fédération communiquent. Ce noeud final est parfois désigné par le terme de noeud final *SOAP* sur le serveur point de contact du fournisseur de services.

Remarque : Si vous utilisez le profil Artefact du navigateur, vous connaissez peut-être ce noeud final sous le nom de *service client d'artefacts* ou *service de réception d'artefacts*.

L'adresse URL repose sur celle que vous avez spécifiée pour votre serveur point de contact. La syntaxe est la suivante :

`https://`*`nom_hôte:numéro_port`*`/sps/`*`nom_fédération`*`/saml`*`xx`*`/login`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est le 9443 sur WebSphere Application Server.

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml*xx*
Version de SAML configurée pour la fédération. Les valeurs peuvent être les suivantes :

- saml (pour SAML 1.0)
- saml11 (pour SAML 1.1)

login

Désignation du type de noeud final qui utilise le port. Le type **login** est utilisé pour le service d'assertion client.

Ce noeud final est utilisé uniquement dans les configurations de fournisseur de services des fédérations SAML 1.x et est défini automatiquement à votre place lors de la configuration de votre fédération.

Noeuds finals SAML et adresses URL SAML 2.0

Plusieurs noeuds finals sont configurés sur votre serveur point de contact afin de permettre l'échange de communications entre vous et votre partenaire. La configuration de ces noeuds finals a lieu en même temps que celle de votre fédération dans Tivoli Federated Identity Manager. Les noeuds finals sont accessibles via des URL et sont utilisés par les partenaires au sein de la fédération.

Si vous êtes responsable de l'installation, de la configuration ou de la gestion d'une fédération dans Tivoli Federated Identity Manager, il peut être utile de vous familiariser avec ces noeuds finals et adresses URL.

Les noeuds finals suivants sont utilisés dans une fédération SAML 2.0.

Serveur point de contact
Noeud final du serveur point de contact sur lequel les communications ont lieu. L'URL du serveur point de contact est également utilisée en tant qu'ID de fournisseur. La syntaxe de l'URL du serveur point de contact est la suivante :

`https://nom_hôte:numéro_port`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est le 9443 sur WebSphere Application Server si le protocole SSL est activé, ou le 9080 dans le cas contraire.

Vous êtes invité à indiquer l'URL de votre serveur point de contact lors de la configuration de votre fédération. Une fois la configuration terminée, l'URL de votre serveur point de contact comporte le suffixe `/sps`, de sorte que la syntaxe de l'URL configurée pour le serveur point de contact est la suivante :

`https://nom_hôte:numéro_port/sps`

Le suffixe `/sps` indique que l'URL est définie pour les services de connexion unique.

Service de résolution des artefacts (ou noeud final SOAP)
Noeud final du serveur point de contact du fournisseur d'identité ou de services sur lequel des artefacts sont échangés pour les messages SAML. Ce noeud final est l'emplacement dans lequel les partenaires de la fédération communiquent. Il est parfois appelé *noeud final SOAP*.

Remarque : Il se peut aussi que vous connaissiez ce noeud final sous le nom de *service répondeur*.

L'adresse URL repose sur celle que vous avez spécifiée pour le serveur point de contact. La syntaxe est la suivante :

`https://nom_hôte:numéro_port/sps/nom_fédération/saml20/soap`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est 9444.

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml20
Désignation du protocole SAML que vous choisissez d'utiliser dans votre fédération.

soap

Désignation du type de noeud final qui utilise le port. Le type **soap** est utilisé pour le service de résolution des artefacts dans les fédérations SAML 2.0.

Ce noeud est défini automatiquement à votre place lors de la configuration de votre fédération.

Service d'assertion client
Noeud final du serveur point de contact du fournisseur de services qui reçoit des assertions ou des artefacts. Ce noeud final est l'emplacement dans lequel les partenaires de la fédération communiquent.

L'adresse URL repose sur celle que vous avez spécifiée pour le serveur point de contact. La syntaxe est la suivante :

`https://nom_hôte:numéro_port/sps/nom_fédération/saml20/login`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est le 9443 sur WebSphere Application Server si le protocole SSL est activé, ou le 9080 dans le cas contraire.

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml20
Désignation du protocole SAML que vous choisissez d'utiliser dans votre fédération.

login

Désignation du type de noeud final qui utilise le port. Le type **login** est utilisé pour le service d'assertion client dans les fédérations SAML 2.0.

Ce noeud final est utilisé uniquement dans les configurations de fournisseur de services des fédérations SAML 2.0 et est défini automatiquement à votre place lors de la configuration de votre fédération.

Service de connexion unique
Noeud final du serveur point de contact du fournisseur d'identité qui reçoit les requêtes d'authentification.

L'adresse URL repose sur celle que vous avez spécifiée pour le serveur point de contact. La syntaxe est la suivante :

`https://nom_hôte:numéro_port/sps/nom_fédération/saml20/login`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est le 9443 sur WebSphere Application Server si le protocole SSL est activé, ou le 9080 dans le cas contraire.

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml20
Désignation du protocole SAML que vous choisissez d'utiliser dans votre fédération.

login

Désignation du type de noeud final qui utilise le port. Le type **login** est utilisé pour le service d'assertion client dans les fédérations SAML 2.0.

Ce noeud final est utilisé uniquement dans les configurations de fournisseur d'identité des fédérations SAML 2.0 et est défini automatiquement à votre place lors de la configuration de votre fédération.

Service SLO (Single Logout)
Noeud final du serveur point de contact du fournisseur d'identité ou de services qui reçoit les requêtes de déconnexion.

L'adresse URL repose sur celle que vous avez spécifiée pour le serveur point de contact. La syntaxe est la suivante :

`https://`*`nom_hôte:numéro_port`*`/sps/`*`nom_fédération`*`/saml20/slo`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port par défaut est le 9443 sur WebSphere Application Server si le protocole SSL est activé, ou le 9080 dans le cas contraire.

La valeur par défaut est attribuée à ce port, sauf si celui-ci est indisponible lors de l'installation de Tivoli Federated Identity Manager. Si le port par défaut est indisponible, le programme d'installation ajoute une valeur de 1 au numéro de port jusqu'à ce qu'il trouve un port disponible portant ce numéro.

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml20
Désignation du protocole SAML que vous choisissez d'utiliser dans votre fédération.

slo

Désignation du type de noeud final qui utilise le port. Le type **slo** est utilisé pour le service de déconnexion SLO dans les fédérations SAML 2.0.

Service de gestion des identificateurs de nom
Noeud final du serveur point de contact du fournisseur d'identité ou de services qui reçoit les messages liées à la gestion des noms. L'adresse URL repose sur celle que vous avez spécifiée pour le serveur point de contact, ainsi que la liaison utilisée.

La syntaxe des adresses de redirection HTTP, POST HTTP et d'artefact HTTP est la suivante :

`https://nom_hôte:numéro_port/sps/nom_fédération/saml20/mnids`

La syntaxe pour SOAP est la suivante :

`https://nom_hôte:numéro_port/sps/nom_fédération/saml20/soap`

Où :

https

HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom d'hôte
Nom d'hôte du serveur point de contact.

numéro_port
Numéro de port où les communications prennent effet sur le serveur. Le numéro de port dépend de la liaison utilisée. Les ports par défaut sont les suivants :

HTTP SOAP : 9444

POST HTTP, Artefact HTTP, Redirection HTTP : 9443

sps

Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml20
Désignation du protocole SAML que vous choisissez d'utiliser dans votre fédération.

mnids ou soap
Désignation du type de noeud final qui utilise le port. Le type **mnids** est employé pour le service de gestion des identificateurs de nom dans les fédérations SAML 2.0 utilisant les entités de redirection HTTP, POST HTTP ou artefact HTTP. Le type **soap** est utilisé lorsque le type de liaison SOAP est configuré.

URL du service Common Domain Cookie utilisée par le service de reconnaissance de fournisseur d'identité (Identity Provider Discovery service)
Par défaut, Tivoli Federated Identity Manager permet la mise en oeuvre d'un service de domaine commun qui permet à un fournisseur d'identité d'informer un fournisseur de services qu'un utilisateur spécifique est prêt à utiliser une fédération.

L'adresse URL par défaut est utilisée en interne pour indiquer si le service de cookies du domaine commun doit accéder en lecture ou en écriture ('get' ou 'set') aux valeurs, en rattachant le suffixe cdcwriter (fournisseur d'identité) ou cdcreader (fournisseur de services) à la fin de l'URL. La syntaxe par défaut de l'URL est :

`https://`*`nom_domaine_commun`*`/sps/`*`nom_fédération`*`/saml20/[cdcreader|cdcwriter}`

Où :

https
HTTPS peut être HTTP si le protocole SSL n'est pas activé sur le serveur.

nom_domaine_commun
Nom de domaine commun partagé.

sps
Racine de contexte de l'application de connexion unique sur WebSphere Application Server. Cette partie de l'URL ne peut pas être modifiée.

nom_fédération
Nom donné à la fédération lors de la configuration.

saml20
Désignation du protocole SAML que vous choisissez d'utiliser dans votre fédération.

cdcwriter *ou* **cdcreader**
Désignation du type d'action (read/get ou write/set) utilisé.

Remarque : Tivoli Federated Identity Manager prend également en charge l'utilisation d'un service de reconnaissance tierce partie ou personnalisé.

Chapitre 17. Exemples de règles de mappage d'identité pour les fédérations SAML

Les rubriques suivantes indiquent les règles de mappage d'identité fournies pour les fédérations SAML. Si vous avez décidé d'utiliser des règles de mappage d'identité pour votre fédération, vous pouvez consulter les règles XSLT.

Pour une présentation du mappage d'identité, y compris une description des options de mappage d'identité qui n'utilisent pas les fichiers de règles de mappage XSLT, voir Chapitre 14, «Planification du mappage des identités d'utilisateur», à la page 155

- «Mappage d'une identité d'utilisateur local vers un jeton SAML 1.x»
- «Mappage d'un jeton SAML 1.x vers une identité d'utilisateur local», à la page 200
- «Mappage d'une identité locale vers un jeton SAML 2.0 à l'aide d'un alias», à la page 201
- «Mappe un jeton SAML 2.0 avec une identité locale», à la page 203

Mappage d'une identité d'utilisateur local vers un jeton SAML 1.x

Ce scénario se produit lors de l'échange de messages entre des partenaires d'une fédération de connexion unique SAML 1.0 ou SAML 1.1.

Lorsqu'une demande d'utilisateur est reçue (par exemple, pour accéder à une ressource distante), Tivoli Federated Identity Manager prend contact avec le serveur point de contact (par exemple, WebSphere Application Server) et obtient une identité d'utilisateur local.

Le serveur Tivoli Federated Identity Manager place les informations relatives à l'identité d'utilisateur local dans un document XML qui est conforme au schéma STSUUSER (utilisateur universel STS (Security Token Service)). Le serveur consulte ensuite son entrée de configuration correspondant au partenaire de la fédération (par exemple, la destination qui héberge une ressource demandée). La configuration indique le type de jeton à créer. Dans ce cas, le type de jeton est SAML.

Le module de mappage d'identité modifie ensuite le document XML de sorte que ce dernier contienne les informations requises pour la génération d'un jeton SAML.

Tableau 20. Entrées STSUUSER servant à générer un jeton SAML

Elément STSUUSER	Informations de jeton SAML	Obligatoire/Facultatif
Principal Attr: Name	AuthenticationStatement/Subject/NameIdentifier	Obligatoire
Liste des attributs	Attributs personnalisés supplémentaires	Facultatif

Le module de mappage est responsable des deux tâches suivantes :

1. Mappage de l'élément Principal Attr Name vers une entrée Principal Name. Le type doit être valide pour SAML. Par exemple :
   ```
   urn:oasis:names:tc:SAML:1.0:assertion#emailAddress
   ```

La figure 10 présente une partie du fichier de règles de mappage par défaut, ip_saml_1x.xsl.

```
<!--
  Ce modèle remplace l'intégralité de l'élément Principal par un élément qui ne
  contient que le nom d'utilisateur iv.
 -->
 <xsl:template match="//stsuuser:Principal">
  <stsuuser:Principal>
   <stsuuser:Attribute name="name" type="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
    <stsuuser:Value>
     <xsl:value-of select="//stsuuser:Principal/stsuuser:Attribute[@name='name']
     [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>
  </stsuuser:Principal>
 </xsl:template>
```

Figure 10. Exemple de code XSL présentant le mappage d'une identité d'utilisateur local vers un nom de Principal pour un jeton SAML

Dans cet exemple, l'identité d'utilisateur local est désignée par le *nom d'utilisateur iv*.

```
<stsuuser:Value>
     <xsl:value-of select="//stsuuser:Principal/stsuuser:Attribute[@name='name']
     [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
```

2. Paramétrage de la méthode d'authentification sur le mécanisme `password`. Cette action est requise par la norme SAML.

   Voir la figure 11.

```
<xsl:template match="//stsuuser:AttributeList">
     <stsuuser:AttributeList>
         <!-- Tout d'abord l'attribut de la méthode d'authentification -->
         <stsuuser:Attribute name="AuthenticationMethod"
              type="urn:oasis:names:tc:SAML:1.0:assertion">
            <stsuuser:Value>urn:oasis:names:tc:SAML:1.0:am:password</stsuuser:Value>
         </stsuuser:Attribute>
     </stsuuser:AttributeList>
</xsl:template>
```

Figure 11. Exemple de code XSL présentant l'affectation d'une méthode d'authentification sous forme d'attribut pour un jeton SAML

Mappage d'un jeton SAML 1.x vers une identité d'utilisateur local

Le fournisseur de services reçoit un jeton SAML 1.0 ou SAML 1.1. Tivoli Federated Identity Manager convertit le contenu du jeton en un fichier XML conforme au schéma d'utilisateur universel STS (Security Token Service).

Tableau 21. Informations de jeton SAML converties en document d'utilisateur universel STS

Informations de jeton SAML	Elément STSUUSER
AuthenticationStatement/Subject/NameIdentifier	Principal Attr: Name

Tivoli Federated Identity Manager convertit cette information en identité d'utilisateur local.

- L'élément NameIdentifier sert à remplir l'attribut `name` de l'élément Principal.

La figure 12 présente l'affectation d'une valeur définie pour le nom Principal. Cet exemple de code est issu du fichier de mappage par défaut, sp_saml_1x.xsl.

```
<!--
  Ce modèle remplace le nom de principal par le nom local de l'utilisateur.
 -->
 <xsl:template match="//stsuuser:Principal/stsuuser:Attribute[@name='name']">
  <stsuuser:Attribute name="name" type="urn:ibm:names:ITFIM:5.1:accessmanager">
            <stsuuser:Value><xsl:value-of
select="//stsuuser:Principal/stsuuser:Attribute[@name='name']/stsuuser:Value" />
           </stsuuser:Value>
            </stsuuser:Attribute>
 </xsl:template>
```

Figure 12. Exemple de code XSL présentant l'affectation d'une valeur pour le nom Principal d'un jeton SAML

Autre exemple de fichier de mappage entre un jeton SAML 1.x et une identité locale : sp_saml_1x_ext.xsl. Ce fichier effectue le mappage comme décrit, mais ajoute une section chargée de vérifier que le fournisseur d'identité a utilisé un niveau d'authentification approprié. Dans cet exemple de fichier, une exception est générée su le fournisseur d'identité a utilisé une authentification par mot de passe.

```
<xsl:param name="message">Detected an unacceptable authentication method.
  A higher level of authentication is required.</xsl:param>
 <xsl:template match="//stsuuser:AttributeList">
 <xsl:variable name="result" select="//stsuuser:AttributeList/
  stsuuser:Attribute[@name='AuthenticationMethod']/stsuuser:Value"/>
 <xsl:if test="(contains($result,'password')) = 'true'">
  <xsl:value-of select="mapping-ext:throwSTSException($message)" />
 </xsl:if>
 </xsl:template>
```

Figure 13. Exemple de code XSL illustrant la vérification d'une valeur de AuthenticationMethod

Mappage d'une identité locale vers un jeton SAML 2.0 à l'aide d'un alias

Ce scénario se produit lors de l'échange de messages entre des partenaires dans une fédération de connexion unique SAML 2.0.

Lorsqu'une demande d'utilisateur est reçue, Tivoli Federated Identity Manager prend contact avec le serveur point de contact et obtient une identité d'utilisateur local. Par exemple, une demande peut être émise pour accéder à une ressource distante, et le serveur point de contact peut être un WebSphere Application Server. Le scénario décrit ici utilise l'exemple de fichier de mappage ip_saml_20.xsl contenant un alias pour l'identité.

Le serveur Tivoli Federated Identity Manager place les informations relatives à l'identité d'utilisateur local dans un document XML qui est conforme au schéma STSUUSER (utilisateur universel STS (Security Token Service)). Le serveur consulte ensuite son entrée de configuration correspondant au partenaire de la fédération (par exemple, la destination qui héberge une ressource demandée). La configuration indique le type de jeton à créer. Dans ce cas, le type de jeton est SAML.

Le module de mappage d'identité modifie ensuite le document XML de sorte que ce dernier contienne les informations requises pour la génération d'un jeton SAML 2.0.

Tableau 22. Entrées STSUUSER servant à générer un jeton SAML, à l'aide d'un alias

Elément STSUUSER	Informations de jeton SAML	Obligatoire/Facultatif
Attribut : AuthContextClassRef	Référence à la classe du contexte d'authentification. Cet élément est paramétré sur `password` par défaut, quelle que soit la méthode d'authentification définie dans les droits d'accès. Vous pouvez modifier la valeur de cet élément dans la règle de mappage.	Obligatoire
Attribut : AudienceRestriction	L'élément audience de la condition de restriction d'audience.	Facultatif
Liste des attributs	Attributs personnalisés supplémentaires	Facultatif

Le module de mappage est responsable des tâches suivantes :

1. Mappage de l'élément Principal Attr Name vers une entrée Principal Name. Lorsque le module de jeton génère le jeton, ce nom de Principal n'est pas utilisé directement. En revanche, la valeur de la zone **Name** est envoyée en entrée du service d'alias de Tivoli Federated Identity Manager. Le service d'alias obtient le nom d'alias, l'identificateur de nom, pour le principal et place l'alias renvoyé dans le module de jeton généré.

   Le type doit être valide pour SAML. Par exemple :

   `urn:oasis:names:tc:SAML:2.0:assertion`
2. Paramétrage de la méthode d'authentification sur le mécanisme `password`. Cette action est requise par la norme SAML.

   L'exemple de code suivant présente une partie du fichier de règles de mappage par défaut, ip_saml_20.xsl.

```
<!--
Remarque : aucun modèle principal n'est requis pour le fournisseur d'identité sous SAML 2.0 car les
identificateurs de noms sont fournis dans l'élément 'Subject' de l'assertion.
-->

 <xsl:template match="//stsuuser:AttributeList">
 <stsuuser:AttributeList>
                        <!-- First the authentcation context class ref. attribute -->
  <stsuuser:Attribute name="AuthnContextClassRef" type="urn:oasis:names:tc:SAML:2.0:assertion">
   <stsuuser:Value>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</stsuuser:Value>
   </stsuuser:Attribute>
  </stsuuser:AttributeList>
 </xsl:template>
```

Figure 14. Exemple de code XSL présentant le mappage d'une identité d'utilisateur local vers un jeton SAML, à l'aide d'un alias

3. Définition de l'élément audience de la condition de restriction d'audience à la valeur de l'élément STSUU `AudienceRestriction`. Si cet élément STSUU n'est pas présent, l'assistance est défini sur l'ID fournisseur du partenaire de fédération.
4. Remplissage de l'instruction d'attribut de la vérification à l'aide des attributs de l'élément AttributeList dans In-STSUU. Ces informations deviennent des informations personnalisées du jeton.

   Des attributs personnalisés peuvent être requis par les applications qui utilisent les informations à transmettre entre les partenaires d'une fédération.

Mappe un jeton SAML 2.0 avec une identité locale

Mappez un jeton SAML 2.0 avec une identité locale pour vous conformer au schéma utilisateur universel du service de jetons de sécurité.

Le fournisseur de services reçoit un jeton SAML 2.0. Puis, Tivoli Federated Identity Manager convertit le contenu du jeton en un document STSUU conforme au schéma d'utilisateur universel STS (Security Token Service).

Tableau 23. Informations de jeton SAML converties en document d'utilisateur universel STS

Informations de jeton SAML	**Elément STSUUSER**
AuthenticationStatement/Subject/NameIdentifier	Principal Attr: Name
Attributs personnalisés supplémentaires	AttributeList (facultatif)

Le module de jeton lit le jeton et extrait l'élément NameIdentifier. Le module de jeton transmet l'élément NameIdentifier, un alias, au service d'alias. Le service d'alias convertit l'alias reçu en identité locale. Le module de jeton place l'identité locale dans l'élément Principal du document STSUU.

- L'alias NameIdentifier renvoyé sert à remplir l'attribut `name` de l'élément Principal. Il s'agit de l'ID utilisateur local.

  L'exemple de code suivant présente l'affectation d'une valeur définie pour le nom Principal. Cet exemple de code est issu du fichier de mappage par défaut, sp_saml_20.xsl.

```
<!--
  Ce modèle remplace le nom de principal par le nom local de l'utilisateur.
 -->
 <xsl:template match="//stsuuser:Principal/stsuuser:Attribute[@name='name']">
  <stsuuser:Attribute name="name" type="urn:ibm:names:ITFIM:5.1:accessmanager">
   <stsuuser:Value>
    <xsl:value-of select="//stsuuser:Principal/stsuuser:Attribute[@name='name']/stsuuser:Value"/>
   </stsuuser:Value>
  </stsuuser:Attribute>
 </xsl:template>
```

Figure 15. Exemple de code XSL présentant l'affectation d'une valeur pour le nom Principal d'un jeton SAML 2.0.

- D'autres informations issues du jeton servent à remplir la zone Attributes de l'élément AttributeList.

  L'exemple de code suivant présente l'affectation facultative de valeurs supplémentaires aux attributs. Cet exemple de code est issu du fichier de mappage par défaut, sp_saml_20.xsl.

```
<xsl:variable name="department">
<xsl:value-of select="//stsuuser:AttributeList/stsuuser:Attribute[@name='Department']/stsuuser:Value"/>
 </xsl:variable>

<xsl:template match="//stsuuser:AttributeList">
 <stsuuser:AttributeList>
  <stsuuser:Attribute type="urn:ibm:names:ITFIM:5.1:accessmanager">
   <xsl:attribute name="Department">
    <stsuuser:Value>
  <xsl:value-of select="//stsuuser:AttributeList/stsuuser:Attribute[@name='Department']
  /stsuuser:Value"/>
    </stsuuser:Value>
   </stsuuser:Attribute>
 </stsuuser:AttributeList>
</xsl:template>
```

Figure 16. Exemple de code XSL présentant l'élément AttributeList pour un jeton SAML 2.0.

Chapitre 18. Requête d'attribut SAML 2.0

La fonction de requête d'attribut SAML 2.0 étend la fonction du protocole SAML 2.0. La fonction SAML 2.0 traditionnelle nécessite que le fournisseur d'identité envoie *tous* les attributs utilisateur requis au partenaire de fédération Les attributs sont inclus dans le cadre de l'assertion générée durant le flux de connexion unique.

La fonction de requête d'attribut SAML 2.0 supprime cette limitation. Les administrateurs pour les fournisseurs d'identité peuvent seulement inclure au flux de connexion unique les attributs utilisés par la plupart des applications cibles. Les applications peuvent utiliser un flux de requête d'attribut SAML 2.0 pour obtenir des conditions requises d'attribut ou des valeurs spécialisées.

La prise en charge de requête d'attribut fournit un ensemble d'attributs de base à l'établissement du contexte d'authentification initial. Vous pouvez demander des informations utilisateur selon nécessaire durant l'opération d'exécution de l'application. Différentes applications requièrent des informations utilisateur différentes. Par exemple, les applications nécessitant une autorisation à grains fins. requièrent des droits utilisateur spécifiques pour pendre les décisions d'autorisation.

La requête d'attribut prend en charge les modes suivants :

Mode direct
: L'application de la requête émet un appel direct au fournisseur d'identité pour obtenir les attributs requis.

Mode pour le compte
: L'application de requête contacte le fournisseur de service, qui transfère la requête d'attribut au fournisseur d'identité.

Mode direct

En mode direct, l'application de requête envoie une requête AttributeQuery au noeud final SOAP de fédération SAML 2.0 au niveau du fournisseur d'identité. Le protocole délégué SOAP termine les actions de protocole nécessaires et émet une assertion SAML. La fonction de requête d'attribut SAML utilise le module STS de requête d'attribut pour émettre l'assertion.

le mode direct nécessite que l'application (demandeur d'attribut) soit connue du fournisseur d'identité. Pour qu'une application soit connue au niveau du fournisseur d'identité, utilisez la commande de l'interface de ligne de commande `manageItfimPartner` pour importer les métadonnées du demandeur.

La flux de connexion unique pour le mode direct est :

1. L'utilisateur nécessite l'accès à une ressource ou application et lance le flux de connexion unique fédérée.
2. Le fournisseur d'identité authentifie l'utilisateur et émet une assertion SAML avec un sous-ensemble d'attributs requis par la plupart des applications ou ressources.
3. L'application ou la ressource détermine si d'autres attributs sont requis. Si c'est le cas, l'application émet une requête `AttributeQuery` au fournisseur d'identité.
4. Le fournisseur d'identité renvoie une assertion SAML avec les attributs requis.

5. L'application ou la ressource obtient les attributs renvoyés par le fournisseur d'identité dans le message de réponse SAML de requête d'attribut.

Mode pour le compte

Le mode on behalf (pour le compte) exige que les applications envoient des demandes de requêtes au fournisseur de service, qui les transmet alors au fournisseur d'identité. Le fournisseur d'identité fournit les attributs requis. Le mode on behalf (pour le compte) prend en charge deux différents types de requêtes :

- Requêtes SAML 2.0 *<AttributeQuery>*

  L'application doit envoyer des messages AttributeQuery au noeud final SOAP de fournisseur de service. Si un message de requête AttributeQuery est utilisé, le fournisseur de service renvoie un message de réponse SAML avec l'assertion correspondante.
- Messages de jeton de sécurité de requête WS-Trust.

  Pour ce protocole, l'application doit envoyer des messages WS-Trust au noeud final de service de confiance. Si l'application de requête envoie un message WS-Trust, le message de réponse est un jeton d'utilisateur universel.

  Remarque : Si votre application est un client WS-Trust, vous pouvez utiliser cette option plutôt qu'utiliser le protocole SAML.

Le mode on behalf (pour le compte) limite la configuration requise au niveau du fournisseur d'identité pour un grand nombre d'applications de fournisseur de service dans les attributs utilisateur de requête. Avec ce mode, le fournisseur de service représente la seule entité connue au niveau du fournisseur d'identité.

Le flux de connexion unique pour le mode on behalf (pour le compte) est le suivant :

1. L'utilisateur nécessite l'accès à une ressource ou application au niveau du fournisseur de service et lance le flux de connexion unique fédérée.
2. Le fournisseur d'identité authentifie l'utilisateur et émet une assertion SAML avec un sous-ensemble d'attributs requis par la plupart des applications ou ressources.
3. Le fournisseur de service sélectionne les attributs à rendre disponibles pour la ressource ou l'application. Le fournisseur de service crée alors la session authentifiée pour l'utilisateur.
4. L'application ou la ressource détermine si d'autres attributs sont requis. Si c'est le cas, l'application émet une valeur `AttributeQuery` ou `RequestSecurityToken` WS-Trust pour les obtenir. L'application envoie la requête au fournisseur de service. Le fournisseur de service transmet la requête au fournisseur d'identité.
5. Le fournisseur d'identité renvoie une assertion SAML avec les attributs requis.
6. L'application ou la ressource obtient les attributs renvoyés par le fournisseur d'identité dans le message de réponse SAML de requête d'attribut. Si une requête WS-Trust est émise, les attributs sont renvoyés à l'application client à l'aide d'un jeton d'utilisateur universel. Si la requête est une requête AttributeQuery SAML, les attributs sont renvoyés dans une réponse SAMLResponse générée par le fournisseur de service.

Partenaire de demande de requête d'attribut

La fonction de requête d'attribut définit un nouveau type de rôle. Les partenaires d'application d'une fédération SAML 2.0 peuvent maintenant agir avec un rôle de *demandeur de requête d'attribut*. Ce rôle est différent du rôle de partenaire de fournisseur de service ou d'identité.

Un demandeur de requête d'attribut représente une entité qui effectue les appels de requête `<AttributeQuery>` basés sur SOAP pour obtenir les attributs utilisateur.

Si vous comptez configurer un *partenaire de demandeur de requête d'attribut*, vous devez générer un fichier de métadonnées tel que spécifié dans la spécification SAML 2.0. Tivoli Federated Identity Manager utilise ce fichier de métadonnées pour créer le partenaire de requête d'attribut. Vous devez utiliser la commande `manageItfimPartner` pour créer le partenaire. Cette commande utilise un fichier de réponse, qui contient un paramètre qui spécifie l'emplacement du fichier de métadonnées.

Développement d'un module STS de requête d'attribut

La fonction de requête d'attribut utilise un module de jeton STS appelé *module de requête d'attribut*. Vous devez configurer le module pour la chaîne d'accréditation pour la fédération SAML 2.0.

Avant de configurer la requête d'attribut, vous devez :

1. Déterminer l'attribut souhaité par votre ressource ou application pour faire la demande à partir du fournisseur d'identité.
2. Développer un script ou module demandant les attributs. Cette requête peut être effectuée par un fichier XSLT ou JavaScript, une chaîne d'assemblage Tivoli Directory Integrator, ou un module de mappage STS personnalisé.

Limitation liée à la migration d'une version précédente de Tivoli Federated Identity Manager

Tivoli Federated Identity Manager prend en charge la migration des fédérations SAML 2.0 depuis une version précédente vers la version actuelle. La fonction de requête d'attribut n'était pas disponible dans les versions précédentes. Sans la fonction de requête d'attribut, la requête d'attribut n'est pas automatiquement activée dans la nouvelle version lors de la migration des fédérations SAML 2.0 depuis la version précédente.

Pour activer la requête d'attribut pour la fédération, procédez comme suit après la migration de la fédération :

- Cochez la case située sur la page de propriétés de la fédération pour activer la requête d'attribut.
- Utilisez l'interface utilisateur graphique de la page de propriétés de fédération pour configurer un module de requête d'attribut.
- Utilisez l'assistant Ajouter un partenaire pour ajouter tous les partenaires qui existaient précédemment pour la fédération.

Configuration de requête d'attribut

Vous pouvez configurer les fédérations et partenaires SAML 2.0 pour prendre en charge la fonction de requête d'attribut.

Les étapes de configuration de requête d'attribut varient selon le scénario de déploiement. Le déploiement inclut la création d'une fédération et l'addition d'un partenaire à la fédération.

Lorsque vous configurez les fédérations, les partenaires de fournisseur d'identité et de service, vous pouvez utiliser une interface utilisateur graphique qui vous demande des paramètres de requête d'attribut. La section fournit des descriptions détaillées de ces paramètres.

Certains paramètres de requête d'attribut utilisent des valeurs existantes pour les fédérations SAML 2.0. Pour ces paramètres, aucune configuration supplémentaire n'est demandée pour la requête d'attribut.

Par exemple, les fournisseurs signent ou valident des assertions selon les paramètres de configuration établis pour la fédération ou le partenaire SAML 2.0. Ce dernier signe ou valide les assertions de requête d'attribut tel que requis par le partenaire de fédération. Vous n'avez pas à spécifier d'autres paramètres pour appliquer la signature ou validation.

Si vous installez SAML 2.0 avec des profils standard ou tous les profils, la signature et la validation sont automatiquement activées. Si vous sélectionnez une installation manuelle des profils, l'assistant vous invite à spécifier s'il faut signer et valider les messages. L'assistant nécessite ces paramètres, que la fonction de requête d'attribut soit activée ou pas.

Pour configurer votre fédération et partenaire en mode direct, procédez comme suit :

- «Création d'une fédération en droit d'attribut»
- «Création d'un partenaire de demande de requête d'attribut», à la page 213

Pour configurer votre fédération et partenaire en mode on behalf (pour le compte), procédez comme suit :

- «Création d'une fédération en droit d'attribut»
- «Création d'un partenaire de fournisseur d'identité ou de service pour une fédération d'autorité d'attribut», à la page 210
- «Création d'un partenaire de demande de requête d'attribut», à la page 213

Création d'une fédération en droit d'attribut

Vous pouvez utiliser la console d'administration ou l'interface de ligne de commande pour créer une fédération SAML 2.0 en droit d'autorité.

Choisissez l'une des méthodes suivantes :

- «Utilisation de la console d'administration pour créer une fédération en autorité d'attribut»
- «Utilisation de l'interface de ligne de commande pour créer une fédération en droit d'attribut», à la page 209

Utilisation de la console d'administration pour créer une fédération en autorité d'attribut

Vous pouvez utiliser la console d'administration pour créer une fédération SAML 2.0 en autorité d'attribut.

Pourquoi et quand exécuter cette tâche

La configuration pour la requête d'attribut utilise le même assistant que celui de toutes les fédérations SAML. Lorsque vous utilisez l'assistant, vous activez la requête d'attribut et êtes invité à fournir des paramètres de configuration.

Les paramètres pour la requête d'attribut sont décrits dans les feuilles de travail pour la configuration de fédération. Voir la rubrique concernant votre type de partenaire :

- «Formulaire de fournisseur d'identité SAML 2.0», à la page 228.
- «Formulaire de fournisseur de services SAML 2.0», à la page 222.

Remarque : Combinez les informations de la procédure suivante avec les instructions de configuration pas à pas des fédérations SAML 2.0 dans le Chapitre 19, «Etablissement d'une fédération SAML», à la page 217.

Procédure

1. Dans le panneau Profils de l'assistant, sélectionnez **Tous** ou **Manuel**.
2. Dans le panneau Détails du profil, accédez à Requête d'attribut et sélectionnez **Activé**. Si vous cochez cette case, de nouveaux panneaux s'affichent.
3. Dans le panneau Assertions de SAML, indiquez la durée de validité d'une assertion avant sa date d'émission. Indiquez également la durée de validité de l'assertion après émission.

   Remarque : Lorsque vous utilisez une fédération de *fournisseur de service* pour une requête d'attribut, la fédération doit émettre des assertions. Cette condition signifie que lorsque vous activez une requête d'attribut pour une fédération de fournisseur de services, le panneau Assertions de SAML s'affiche et vous devez indiquer des valeurs. Lorsque vous configurez une fédération de fournisseur de service sans requête d'attribut, vous n'avez pas à définir de valeurs pour les assertions SAML.

   Le panneau Assertions de SAML s'affiche pour la création de fédération de *fournisseur d'identité* que la requête d'attribut soit sélectionné ou non. Dans ce type de fédération, les assertions SAML sont émises pour plusieurs fins.
4. Dans le panneau Sélection de module d'attribut, sélectionnez un des choix suivants :
   - Transformation XSLT ou JavaScript
   - Module Tivoli Directory Integrator
   - Module de mappage personnalisé.

   Basez votre sélection sur la méthode identifiée pour votre déploiement lorsque vous avez planifié la configuration.

Utilisation de l'interface de ligne de commande pour créer une fédération en droit d'attribut

Vous pouvez utiliser l'interface de ligne de commande pour créer une fédération SAML 2.0 en droit d'attribut.

Pourquoi et quand exécuter cette tâche

Lors de l'utilisation de l'interface de ligne de commande pour créer une fédération SAML 2, vous devez d'abord créer et renseigner un fichier de réponses de

fédération SAML 2. Pour établir la fédération SAML 2 en droit d'attribut, vous devez définir les valeurs dans le fichier de réponses pour les paramètres suivants :

- `AttributeQueryMappingRule`
- `AttributeQueryMappingRuleFileName`
- `AttributeAuthorityEnabled`
- `SignAttributeQueryRequest`
- `SignAttributeQueryResponse`

Pour toute description des paramètres nécessaires, voir la rubrique«Paramètres de fichier de réponses de fédération de requête d'attribut SAML 2.0», à la page 214.

Pour plus d'informations sur l'utilisation de l'interface de ligne de commande pour créer une fédération SAML 2 et un fichier de réponses, voir le document *IBM Tivoli Federated Identity ManagerGuide d'administration*.

Procédure

1. Créez un fichier de réponses SAML 2.

   Par exemple, pour créer un fichier de réponses SAML 2 basé sur une fédération existante :

   ```
   $AdminTask manageItfimFederation {-operation createResponseFile
   -fimDomainName domain1 -federationName idpsaml2
   -fileId c:\temp\saml2idp.rsp}
   ```

2. Editez le fichier de réponses SAML 2 pour définir les paramètres de requête d'attribut.

   Dans l'exemple, le fichier de réponses est `c:\temp\saml2idp.rsp`

3. Créez la fédération SAML 2 en droit d'attribut.

   Pour créer une fédération de fournisseur d'identité ou de fournisseur de service activé pour la requête d'attribut, utilisez la syntaxe standard. Aucune autre option n'est à spécifier.

   Par exemple, si le fichier de réponses est `c:\temp\saml2idp.rsp`:

   ```
   $AdminTask manageItfimFederation { -operation create -fimDomainName domain1
   -fileId c:\temp\saml2idp.rsp}
   ```

Création d'un partenaire de fournisseur d'identité ou de service pour une fédération d'autorité d'attribut

Vous pouvez créer un partenaire de fournisseur d'identité ou de service pour une fédération SAML 2.0 configurée en autorité d'attribut.

Lorsqu'une fédération a été configurée sur une autorité d'attribut, vous pouvez ajouter des partenaires des types suivants :

- Partenaire de fournisseur de service

  Ajoutez un partenaire de fournisseur de service à une fédération de fournisseur d'identité. Vous pouvez configurer ce partenaire pour échanger des requêtes-réponses de demandes d'attributs avec le fournisseur de fédération.

- Partenaire de fournisseur d'identité

  Ajoutez un partenaire de fournisseur d'identité dans une fédération de fournisseurs de service. Vous pouvez configurer ce partenaire pour échanger des requêtes-réponses de demandes d'attributs avec le fournisseur de fédération.

- Partenaire de demande de requête d'attribut

Ce type de partenaire représente un cas spécial à utiliser lorsque la ressource ou l'application de requête ne dispose pas de Tivoli Federated Identity Manager.

Remarque : Les instructions de cette rubrique ne concernant pas les partenaires de requête de demande d'attribut. Voir «Création d'un partenaire de demande de requête d'attribut», à la page 213.

Pour ajouter un partenaire de fournisseur d'identité ou de service, voir :

- «Utilisation de la console d'administration pour créer un partenaire de fournisseur d'identité ou fournisseur de service»
- «Utilisation de l'interface de ligne de commande pour créer un partenaire de fournisseur d'identité ou fournisseur de service», à la page 212

Utilisation de la console d'administration pour créer un partenaire de fournisseur d'identité ou fournisseur de service

Vous pouvez utiliser la console d'administration pour créer un partenaire de fournisseur d'identité ou fournisseur de service.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'assistant Ajouter un partenaire pour ajouter un partenaire de fournisseur de service ou d'identité à une fédération. Cet assistant est également utilisé pour ajouter des partenaires SAML 2.0 sans requête d'attribut.

Lorsque vous utilisez l'assistant pour ajouter un partenaire à une fédération, le programme de configuration détermine si la fédération est configurée en tant qu'autorité de requête d'attribut. Si la fédération est une autorité de requête d'attribut, d'autres panneaux vous invitent à entrer davantage d'informations.

Les panneaux de configuration sont légèrement différents pour les partenaires de fournisseur d'identité ou de service. Voir la table suivante.

Panneau de configuration	**Type de partenaire**	**Description**
Assertions SAML	Partenaire de fournisseur d'identité pour une fédération de fournisseur de service *uniquement*	Le panneau de configuration d'assertions SAML vous permet de spécifier les attributs à inclure à l'assertion. La valeur par défaut est d'inclure tous les attributs. Vous pouvez utiliser ce paramètre pour spécifier un ensemble de base d'attributs. Le panneau d'assertions SAML vous permet également de spécifier les attributs à chiffrer et les algorithmes de chiffrage à utiliser.

Panneau de configuration	Type de partenaire	Description
Sélection de module d'attribut	Partenaire de fournisseur d'identité pour une fédération de fournisseur de service et partenaire de fournisseur de service pour une fédération de partenaire d'identité	Dans le panneau Sélection de module d'attribut, vous devez sélectionner un des éléments suivants : • Transformation XSLT ou JavaScript • Module Tivoli Directory Integrator • Module de mappage personnalisé. Basez votre sélection sur la méthode identifiée pour votre déploiement lorsque vous avez planifié la configuration.

Les paramètres pour la configuration de partenaire pour la requête d'attribut sont décrits dans la feuille de travail pour la configuration de partenaire SAML 2.0. Voir la rubrique pour votre type de partenaire :

- «Formulaire de fournisseur d'identité partenaire SAML 2.0», à la page 261
- «Formulaire de fournisseur de services partenaire SAML 2.0», à la page 253

L'assistant d'interface utilisateur graphique pour l'ajout de partenaires SAML 2.0 inclut les panneaux pour la configuration de requête d'attribut. Pour configurer le partenaire de fournisseur d'identité ou de service, voir les instructions SAML 2.0 : «Ajout à votre partenaire», à la page 271

Utilisation de l'interface de ligne de commande pour créer un partenaire de fournisseur d'identité ou fournisseur de service

Vous pouvez créer un partenaire de fournisseur d'identité ou de service pour une fédération SAML 2.0 configurée en autorité d'attribut.

Pourquoi et quand exécuter cette tâche

Lors de l'utilisation de l'interface de ligne de commande pour créer un partenaire, vous devez d'abord créer et renseigner un fichier de réponses de la fédération SAML 2. Pour configurer les partenaires afin d'utilisateur la fonction de requête d'attribut, vous devez définir les valeurs pour les paramètres suivants dans le fichier de réponses :

- `AttributeQueryMappingRule`
- `AttributeQueryMappingRuleFileName`
- `ValidateAttributeQueryRequest`
- `ValidateAttributeQueryResponse`

Pour plus d'informations sur l'utilisation de l'interface de ligne de commande pour créer un partenaire SAML 2 et un fichier de réponses de partenaire, voir le document *IBM Tivoli Federated Identity ManagerGuide d'administration*.

Procédure

1. Créez un fichier de réponses de partenaire SAML 2.

Par exemple, pour créer un fichier de réponses de partenaire SAML 2 basé sur un partenaire existant :

```
$AdminTask manageItfimPartner {-operation createResponseFile
-fimDomainName domain1  -federationName fed1
-partnerName idppartner -fileId c:\temp\saml2idp.rsp }
```

2. Editez le fichier de réponses de partenaire SAML 2 pour définir les paramètres de requête d'attribut.

   Dans l'exemple, le fichier de réponses est `c:\temp\saml2idp.rsp`

   Pour la description des paramètres de fichier de réponses de requête d'attribut, voir la rubrique «Paramètres de fichier de réponses de partenaire de requête d'attribut SAML 2.0», à la page 215

3. Pour créer un partenaire de fournisseur d'identité configuré pour une requête d'attribut, utilisez la syntaxe standard.

   Vous pouvez éventuellement spécifier le rôle de partenaire dans la ligne de commande. Vous n'avez pas à spécifier le rôle partenaire. Lorsque le rôle n'est pas spécifié, le programme définit automatiquement le rôle du partenaire basé sur le rôle de la fédération.

   Par exemple, si le fichier de réponses est `c:\temp\saml2idp.rsp` :

   ```
   $AdminTask manageItfimPartner { -operation create -fimDomainName domain1
   -federationName idpsaml2 -partnerName idpartner
   -fileId c:\temp\saml2idp.rsp
   -signingKeystorePwd testonly -encryptionKeystorePwd testonly }
   ```

   Si vous souhaitez spécifier le rôle du partenaire dans la ligne de commande, ajoutez l'option `-partnerRole`, et indiquez `sp` ou `idp`. Par exemple, pour spécifier un partenaire de fournisseur de service :

   ```
   $AdminTask manageItfimPartner { -operation create -fimDomainName domain1
   -federationName idpsaml2 -partnerName idpartner
   -partnerRole sp
   -fileId c:\temp\saml2sp.rsp
   -signingKeystorePwd testonly -encryptionKeystorePwd testonly }
   ```

Création d'un partenaire de demande de requête d'attribut

Utilisez l'interface de ligne de commande pour créer un partenaire de demande de requête d'attribut.

Pourquoi et quand exécuter cette tâche

Vous devez utiliser l'interface de ligne de commande pour ajouter un partenaire de demande de requête d'attribut à une fédération. L'interface utilisateur graphique d'administration ne fournit pas d'assistant pour cette tâche.

Utilisez la commande `manageItfimPartner` pour créer le partenaire. Cette commande prend en charge un paramètre de rôle de partenaire `qr` qui indique qu'un partenaire de demande de requête va être créé.

Procédure

1. Créez un fichier de réponses de partenaire SAML 2.

   Par exemple, pour créer un fichier de réponses de partenaire de demande de requête d'attribut SAML 2 basé sur un partenaire existant :

   ```
   $AdminTask manageItfimPartner { -operation createResponseFile
   -fimDomainName fimipdomain  -federationName saml20ip
   -partnerRole qr -fileId /downloads/qr.out }
   ```

2. Editez le fichier de réponses de sorte à afficher l'emplacement du fichier de métadonnées à partir du partenaire de demande de requête d'attribut. Ce nom

de fichier est un paramètre du fichier de réponses. Vous devez également ajouter des informations spécifiques au partenaire.

Pour plus d'informations sur l'utilisation de l'interface de ligne de commande pour créer un partenaire SAML 2 et un fichier de réponses de partenaire, voir le document *IBM Tivoli Federated Identity ManagerGuide d'administration*.

3. Créez un partenaire de demande de requête d'attribut :

```
$AdminTask manageItfimPartner { -operation create
-fimDomainName fimipdomain
-federationName saml20ip -partnerName samlqr
-partnerRole qr -fileId /downloads/qr.out
-signingKeystorePwd testonly
-encryptionKeystorePwd testonly}
```

Paramètres de fichier de réponses de fédération de requête d'attribut SAML 2.0

Le fichier de réponses de fédération SAML 2.0 contient des paramètres utilisés par la requête d'attribut.

Tableau 24. Paramètres de requête d'attribut pour le fichier de réponse de fédération

Paramètre	Valeur	Description
AttributeQueryMappingRule	*contenu du fichier de règle de mappage*	Contient le contenu de la règle de mappage (XSL) nécessaire au formatage correct de la règle, afin que cette dernière puisse être incluse dans un fichier de réponses XML. Utilisez cette propriété pour spécifier une règle de mappage sans utiliser de fichier sur le système de fichiers. Utilisez également cette propriété si vous modifiez une fédération. Si vous souhaitez éditer la règle XSLT en fichier normal, indiquez-le dans le fichier de réponses à l'aide de la propriété **AttributeQueryMappingRuleFileName**. Cette règle est utilisée pour les opérations de requête d'attribut.
AttributeQueryMappingRuleFileName	*chemin et nom de fichier*	Indique le nom du chemin d'accès vers un fichier XSLT utilisé en règle de mappage. S'il est défini, il a la priorité sur la priorité **AttributeQueryMappingRule**. Cette règle est utilisée pour les opérations de requête d'attribut.
AttributeAuthorityEnabled	*true ou false*	Indique si la fonction de requête d'attribut est configurée dans la fédération. La valeur `true` active la requête d'attribut. La valeur `false` désactive la requête d'attribut. Par défaut : false
SignAttributeQueryResponse	*true ou false*	Indique si les réponses de requête d'attribut sont signées.
SignAttributeQueryRequest	*true ou false*	Indique si les requêtes d'analyse d'attributs sont signées.

Paramètres de fichier de réponses de partenaire de requête d'attribut SAML 2.0

Le fichier de réponses de partenaire SAML 2.0 contient des paramètres utilisés par la requête d'attribut.

Tableau 25. Paramètres de requête d'attribut pour le fichier de réponse de partenaire

Paramètre	Valeur	Description
AttributeQueryMappingRule	*contenu du fichier de règle de mappage*	Contient le contenu de la règle de mappage (XSL) nécessaire au formatage correct de la règle, afin que cette dernière puisse être incluse dans un fichier de réponses XML. Utilisez cette propriété si vous souhaitez spécifier une règle de mappage sans utiliser de fichier sur le système de fichiers. Utilisez également cette propriété si vous modifiez une fédération. Si vous souhaitez éditer la règle XSLT en fichier normal, indiquez-le dans le fichier de réponses à l'aide de la propriété **AttributeQueryMappingRuleFileName**. Cette règle est utilisée pour les opérations de requête d'attribut.
AttributeQueryMappingRuleFileName	*chemin et nom de fichier*	Indique le nom du chemin d'accès vers un fichier XSLT utilisé en règle de mappage. S'il est défini, il a la priorité sur la priorité **AttributeQueryMappingRule**. Cette règle est utilisée pour les opérations de requête d'attribut.
ValidateAttributeQueryResponse	*true ou false*	Indique que la validation des signatures de partenaire a lieu sur les réponses de requête d'attribut reçues. Une erreur se produit si le message n'est pas signé.
ValidateAttributeQueryRequest	*true ou false*	Indique qu'une demande d'analyse d'attributs reçue de la signature du partenaire est validée. Une erreur se produit si le message n'est pas signé.

Chapitre 19. Etablissement d'une fédération SAML

Etablissez une fédération SAML pour effectuer la configuration de votre fédération.

Pour configurer votre fédération, procédez comme suit :

1. «Rassemblement des informations relatives à la configuration de votre fédération».
2. «Création de votre rôle dans la fédération», à la page 234.
3. «Délivrance d'instructions à votre partenaire», à la page 237.
4. «Obtention des données de configuration de fédération de la part de votre partenaire», à la page 239.
5. «Ajout à votre partenaire», à la page 271.
6. «Transmission des propriétés de la fédération au partenaire», à la page 273.

Rassemblement des informations relatives à la configuration de votre fédération

L'assistant Fédération vous invite à indiquer les informations utilisées dans votre fédération. Avant de démarrer l'assistant, préparez le processus de configuration en réunissant les informations correspondantes dans le formulaire approprié.

Pourquoi et quand exécuter cette tâche

Sélectionnez un formulaire en fonction de la norme SAML que vous souhaitez utiliser dans la fédération, ainsi que le rôle que vous tenez dans cette dernière.

- «Formulaire de fournisseur de services IDP SAML 1.x»
- «Formulaire de fournisseur d'identité SAML 1.x», à la page 219
- «Formulaire de fournisseur de services SAML 2.0», à la page 222
- «Formulaire de fournisseur d'identité SAML 2.0», à la page 228

Formulaire de fournisseur de services IDP SAML 1.x

Si vous remplissez le rôle de fournisseur de services dans la fédération et utilisez SAML 1.0 ou SAML 1.1, enregistrez vos informations de configuration dans les tableaux suivants.

Tableau 26. Informations générales pour le fournisseur de services dans la fédération SAML 1.x

Informations générales	Description	Votre valeur
Nom de la fédération	Nom unique que vous attribuez à la fédération.	
Rôle	Rôle que vous remplissez dans la fédération. (Dans les présentes instructions, vous êtes le fournisseur de services.)	Fournisseur de services

Tableau 27. Informations de contact pour le fournisseur de services dans la fédération SAML 1.x

Personne à contacter	Description	Votre valeur
Nom de l'entreprise, adresse URL et nom de contact de l'entreprise.	Nom de votre société et autres informations facultatives sur le contact associé à votre rôle dans la fédération.	

Tableau 28. Protocole de fédération pour le fournisseur de services dans la fédération SAML 1.x

Protocole de fédération	Description	Votre valeur
Protocole	Protocole SAML que vous et votre partenaire utilisez dans la fédération.	Vous pouvez choisir une des options suivantes : • SAML 1.0 • SAML 1.1

Tableau 29. Informations du serveur point de contact pour le fournisseur de services dans la fédération SAML 1.x

Serveur point de contact	Description	Votre valeur
URL du serveur point de contact	Adresse URL donnant accès aux noeuds finals sur le serveur point de contact.	

Tableau 30. Informations de signature pour le fournisseur de services dans la fédération SAML 1.x

Signatures	Description	Votre valeur
Signer les requêtes de résolution d'artefact	Case à cocher indiquant que vous signerez les messages de requête. Valeur par défaut : Aucune signature. La case n'est pas cochée.	Vous pouvez choisir une des options suivantes : • Signer les messages de requête (cochez la case). • Ne pas signer les messages de requête (ne pas cocher la case).
Sélectionner la clé de signature • Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés • Clé privée utilisée pour la signature des messages de requête	Si vous cochez cette case, vous devez indiquer la clé de signature à utiliser pour signer les requêtes. **Remarque :** Avant d'effectuer cette tâche, veillez à créer la clé et à l'importer dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager. Pour plus d'informations, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	Nom de fichiers de clés : Mot de passe du fichier de clés : Nom d'alias de clé :

Tableau 31. Informations de mappage d'identité pour le fournisseur de services dans la fédération SAML 1.x

Mappage d'identité	Description	Votre valeur
Options de mappage d'identité Vous pouvez choisir une des options suivantes : • Fichier de transformation XSL (XSLT) contenant les règles de mappage • Module de mappage personnalisé	Type de mappage d'identité utilisé. Vous devez savoir si vous devez utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous souhaitez utiliser cette option, vous devez créer votre module de mappage et l'ajouter à l'environnement en tant que module type et module d'instance *avant* de pouvoir l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	L'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété les tables, poursuivez avec les instructions de la rubrique «Création de votre rôle dans la fédération», à la page 234.

Formulaire de fournisseur d'identité SAML 1.x

Si vous assumez le rôle de fournisseur d'identité dans la fédération et utilisez SAML 1.0 ou SAML 1.1, enregistrez vos informations de configuration dans les tableaux suivants.

Tableau 32. Informations générales pour le fournisseur d'identité dans la fédération SAML 1.x

Informations générales	Description	Votre valeur
Nom de la fédération	Nom unique que vous attribuez à la fédération.	
Rôle	Rôle que vous remplissez dans la fédération. (Dans les présentes instructions, vous êtes le fournisseur d'identité.)	Fournisseur d'identité

Tableau 33. Informations de contact pour le fournisseur d'identité dans la fédération SAML 1.x

Personne à contacter	Description	Vos valeurs
Nom de l'entreprise, adresse URL et nom de contact de l'entreprise.	Nom de l'entreprise et, le cas échéant, autres informations relatives au contact associé à la fédération.	Nom de l'entreprise :

Tableau 34. Informations sur le protocole de fédération pour le fournisseur d'identité dans la fédération SAML 1.x

Protocole de fédération	Description	Votre valeur
Protocole	Protocole SAML que vous et votre partenaire utilisez dans la fédération.	Vous pouvez choisir une des options suivantes : • SAML 1.0 • SAML 1.1

Tableau 35. Serveur point de contact pour le fournisseur de services dans la fédération SAML 1.x

Serveur point de contact	Description	Votre valeur
URL du serveur point de contact	Adresse URL donnant accès aux noeuds finals sur le serveur point de contact.	

Tableau 36. Informations de signature pour le fournisseur d'identité dans la fédération SAML 1.x

Signatures	Description	Votre valeur
Options de signature : • **Les messages SAML pour le profil POST du navigateur sont signés** (obligatoire) • **Signer les messages SAML du profil d'artefact** (facultatif)	• Lorsque le POST du navigateur est utilisé en tant que profil, les messages SAML doivent être signés. Cette option est donc présélectionnée et ne peut pas être désélectionnée. • Il est possible de signer également les messages SAML lorsque l'artefact du navigateur est utilisé.	Vous pouvez choisir une des options suivantes : • Signer les messages d'artefact du navigateur (cochez la case). • Ne pas signer les messages d'artefact du navigateur (ne pas cocher la case).
Sélectionner la clé de signature • Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés • Clé privée utilisée pour la signature	Etant donné que les messages POST du navigateur doivent être signés, vous êtes tenu de fournir une clé de signature. Si vous choisissez également de signer les messages lorsque l'artefact du navigateur est utilisé, cette même clé est utilisée pour les signer. **Remarque :** Avant d'effectuer cette tâche, veillez à créer la clé et à l'importer dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager. Pour plus d'informations, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	Nom de fichiers de clés : Mot de passe du fichier de clés : Nom d'alias de clé :

Tableau 37. Informations sur les paramètres de message SAML pour le fournisseur d'identité dans la fédération SAML 1.x

Paramètres des messages SAML	Description	Votre valeur
URL du service de résolution des artefacts	Adresse URL de votre noeud final de résolution d'artefacts. **Remarque :** La valeur de cette zone est renseignée automatiquement à l'aide de l'URL du serveur point de contact que vous avez définie précédemment.	
Durée de mise en cache d'artefact (en secondes)	Durée de mise en cache d'artefact en secondes. Valeur par défaut : 30 secondes.	
Autoriser l'extension IBM Protocol	Vous devez indiquer si vous autorisez l'utilisation de l'extension IBM PROTOCOL. Cette extension permet de spécifier un paramètre de chaîne de requête qui indique si l'artefact du navigateur ou le POST du navigateur est utilisé. Pour plus d'informations, voir «SAML 1.x», à la page 179.	Vous pouvez choisir une des options suivantes : • Autoriser l'extension IBM Protocol (cochez la case). • Ne pas autoriser l'extension Protocol (ne pas cocher la case).

Tableau 38. Informations sur les paramètres de jetons pour le fournisseur d'identité dans la fédération SAML 1.x

Configuration des paramètres de jeton	Description	Votre valeur
Durée de validité (en secondes) d'une assertion avant sa date d'émission	Durée en secondes pendant laquelle une assertion est considérée valide avant sa date de création. Valeur par défaut : 60	
Durée de validité de l'assertion après émission	Durée en secondes pendant laquelle une assertion est considérée valide après sa date de création. Valeur par défaut : 60	

Tableau 39. Informations de mappage d'identité pour le fournisseur d'identité dans la fédération SAML 1.x

Mappage d'identité	Description	Votre valeur
Options de mappage d'identité Vous pouvez choisir une des options suivantes : • Fichier de transformation XSL (XSLT) contenant les règles de mappage • Module de mappage personnalisé	Type de mappage d'identité utilisé. Vous devez savoir si vous devez utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous souhaitez utiliser cette option, vous devez créer votre module de mappage et l'ajouter à l'environnement en tant que module type et module d'instance *avant* de pouvoir l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	L'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété les tables, poursuivez avec les instructions de la rubrique «Création de votre rôle dans la fédération», à la page 234.

Formulaire de fournisseur de services SAML 2.0

Si vous assumez le rôle du fournisseur de services dans la fédération et utilisez SAML 2.0 ou SAML 1.1, enregistrez vos informations de configuration dans les tableaux suivants.

Tableau 40. Informations générales pour le fournisseur de services dans la fédération SAML 2.0

Informations générales	Description	Votre valeur
Nom de la fédération	Nom unique que vous attribuez à la fédération.	
Rôle	Rôle que vous remplissez dans la fédération. (Dans les présentes instructions, vous êtes le fournisseur de services.)	Fournisseur de services

Tableau 41. Informations de contact pour le fournisseur de services dans la fédération SAML 2.0

Personne à contacter	Description	Votre valeur
Nom de l'entreprise, adresse URL et nom de contact de l'entreprise.	Nom de votre entreprise ainsi que, le cas échéant, d'autres informations relatives au contact associé avec votre rôle dans la fédération.	

Tableau 42. Protocole de fédération pour le fournisseur de services dans la fédération SAML 2.0

Protocole de fédération	Description	Votre valeur
Protocole	Protocole SAML que vous et votre partenaire utilisez dans la fédération.	SAML 2.0

Tableau 43. Informations du serveur point de contact pour le fournisseur de services dans la fédération SAML 2.0

Serveur point de contact	Description	Votre valeur
URL du serveur point de contact	Adresse URL donnant accès aux noeuds finals sur le serveur point de contact.	

Tableau 44. Sélection de profil et informations de configuration pour le fournisseur de services dans la fédération SAML 2.0

Sélection de profil	Description	Votre valeur
Options de profil SAML 2.0 : Choisissez l'une des options de profil suivantes :	Profil de votre fédération. Pour plus d'informations sur les profils, voir «SAML 2.0», à la page 181.	Vous pouvez choisir une des options suivantes : • Basique • Typique • Tous • Manuel
Basique : Connexion unique de navigateur Web, déconnexion unique	Ce paramètre active les profils suivants avec toutes les liaisons prises en charge : • Liaison SSO de navigateur Web • Déconnexion unique (SLO)	(aucune valeur supplémentaire requise)
Typique : Connexion unique de navigateur Web, déconnexion unique et Gestion des identificateurs de nom	Ce paramètre active les profils suivants avec toutes les liaisons prises en charge : • Liaison SSO de navigateur Web • Déconnexion unique (SLO) • Client ou proxy évolué • Gestion des identificateurs de nom	(aucune valeur supplémentaire requise)

Tableau 44. Sélection de profil et informations de configuration pour le fournisseur de services dans la fédération SAML 2.0 (suite)

Sélection de profil	Description	Votre valeur
Activer tous les profils et toutes les liaisons	Si vous sélectionnez l'option **Activer tous les profils et toutes les liaisons**, vous devez être prêt à fournir les informations suivantes dans les panneaux successifs : **Paramètres de reconnaissance du fournisseur d'identité :** • Nom de domaine commun • URL du service de cookie de domaine commun **Panneau Proxy client amélioré :** • En-têtes HTTP	**Paramètres de reconnaissance du fournisseur d'identité** • Nom de domaine commun : • URL du service de cookie de domaine commun : **Proxy client amélioré** En-têtes HTTP :
Manuel : Choisissez des profils et des liaisons individuels	Si vous choisissez l'option **Manuel**, vous devez être prêt à sélectionner des profils individuels et des liaisons prises en charge.	Profils et liaisons :

Tableau 45. Informations de signature pour le fournisseur de services dans la fédération SAML 2.0

Signatures	Description	Votre valeur
Demander une signature sur les messages et assertions entrants	Case à cocher indiquant que votre partenaire utilise sa clé privée pour signer le message et l'assertion. Par défaut, la case est cochée.	Vous pouvez choisir une des options suivantes : • Le partenaire signe (la case est cochée). • Le partenaire ne signe pas (la case n'est pas cochée).
Sélectionner les messages et assertions sortants qui nécessitent une signature	Boutons indiquant quels messages sortants doivent être signés par vos soins. Lorsque le paramètre par défaut est sélectionné, l'ensemble des messages et assertions SAML sortants classiques (à l'exception des objets ArtifactResponse et AuthnResponse) doivent être signés.	Vous pouvez choisir une des options suivantes : • Les ensembles de messages SAML sortants classiques sont signés. • Toutes les assertions et tous les messages SAML sortants sont signés. • Aucune assertion ni aucun message SAML sortant ne sont signés.

Tableau 45. Informations de signature pour le fournisseur de services dans la fédération SAML 2.0 (suite)

Signatures	Description	Votre valeur
Sélectionner la clé de signature • Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés • Clé privée utilisée pour la signature des messages	Si vous signez des messages et des assertions, vous devez fournir la clé de signature que vous utilisez pour les signer. **Remarque :** Avant d'effectuer cette tâche, veillez à créer la clé et à l'importer dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager. Pour plus d'informations, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	Nom de fichiers de clés : Mot de passe du fichier de clés : Nom d'alias de clé :

Tableau 46. Informations de chiffrement pour le fournisseur de services dans la fédération SAML 2.0

Chiffrement	Description	Votre valeur
Clé de chiffrement: • Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés • Biclé publique/privée utilisée pour les données reçues en provenance de votre partenaire.	Biclé publique/privée utilisée pour le chiffrement. Votre partenaire utilise la clé publique pour chiffrer les données qu'il vous envoie. Vous utiliserez la clé privée pour déchiffrer les données qui vous sont envoyées par votre partenaire. Vous devez indiquer la paire de clés à utiliser. **Remarque :** Avant d'effectuer cette tâche, veillez à créer la clé et à l'importer dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.	Nom de fichiers de clés : Mot de passe du fichier de clés : Nom d'alias de clé :

Tableau 47. Paramètres des messages SAML pour le fournisseur de services dans la fédération SAML 2.0

Paramètres des messages	Description	Votre valeur
Options des messages : • Durée de vie du message (en secondes) • Durée de vie de l'artefact (en secondes) • Délai d'expiration de session	Intervalle de temps, spécifié en secondes, pendant lequel les messages, artefacts et sessions sont valides. Les valeurs par défaut sont les suivantes : • Durée de vie des messages : 300 • Durée de vue des artefacts : 120 • Délai d'expiration de session : 7200	Durée de vie du message (en secondes) : Durée de vie de l'artefact (en secondes) : Délai d'expiration de session :
Options de connexion unique • Le fournisseur d'identité est autorisé avec l'utilisateur • La connexion unique est passive • Forcer le fournisseur d'identité à authentifier l'utilisateur	Spécifie la manière dont le fournisseur d'identité interagit avec les utilisateurs.	Vous pouvez choisir une des options suivantes : • Le fournisseur d'identité est autorisé avec l'utilisateur • La connexion unique est passive • Forcer le fournisseur d'identité à authentifier l'utilisateur
Noeud final SOAP	URL du point d'extrémité SOAP. Valeur par défaut : la valeur contenue dans cette zone dépend de l'URL du serveur point de contact que vous avez fournie précédemment. **Remarque :** Si la liaison SOAP n'est pas utilisée dans le profil que vous avez sélectionné, cette zone n'est pas affichée.	

Tableau 48. Informations de requête d'attribut pour le fournisseur de service

Requête d'attribut	Description	Votre valeur
Activé	Indique si le fournisseur est autorisé à agir en tant qu'autorité d'attribut. Si cette option est sélectionnée, le profil de requête d'attribut est activé.	
Durée d'assertion avant sa date d'émission.	Durée en secondes pendant laquelle une assertion est considérée valide avant sa date de création. Valeur par défaut : 60	

Tableau 48. Informations de requête d'attribut pour le fournisseur de service (suite)

Requête d'attribut	Description	Votre valeur
Durée de validité de l'assertion après émission	Durée en secondes pendant laquelle une assertion est considérée valide après sa date de création. Valeur par défaut : 60	

Tableau 49. Informations de mappage de requête d'attribut pour le fournisseur de services dans la fédération SAML 2.0

Mappage de requête d'attribut	Description	Votre valeur
Options de mappage de requête d'attribut Vous pouvez choisir une des options suivantes : • Fichier de transformation XSL ou JavaScript contenant les règles de mappage • Module de mappage Tivoli Directory Integrator • Module de mappage personnalisé	Type de mappage de requête d'attribut utilisé. Vous devez sélectionner un fichier XSLT, un module de mappage Tivoli Directory Integrator ou un module de mappage personnalisé. Si vous utilisez un fichier XSLT, ce dernier doit avoir été créé avant la configuration de la fédération. Le module de mappage Tivoli Directory Integrator est un module STS. Le mappage personnalisé est une option avancée. Si vous utilisez cette option, vous devez créer et ajouter un nouveau type et instance de module *avant* de pouvoir l'utiliser dans votre configuration.	L'une des valeurs suivantes : • Chemin de fichier XSLT • Module de mappage Tivoli Directory Integrator • Nom de l'instance de module de mappage personnalisée

Tableau 50. Informations de mappage d'identité pour le fournisseur de services dans la fédération SAML 2.0

Mappage d'identité	Description	Votre valeur
Options de mappage d'identité Vous pouvez choisir une des options suivantes : • Fichier de transformation XSL contenant les règles de mappage • Module de mappage personnalisé	Type de mappage d'identité utilisé. Vous devez savoir si vous devez utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous souhaitez utiliser cette option, vous devez créer votre module de mappage et l'ajouter à l'environnement en tant que module type et module d'instance *avant* de pouvoir l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	L'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété les tables, poursuivez avec les instructions de la rubrique «Création de votre rôle dans la fédération», à la page 234.

Formulaire de fournisseur d'identité SAML 2.0

Si vous devez être le fournisseur d'identité d'une fédération SAML 2.0, enregistrez vos informations de configuration dans les tables suivantes.

Tableau 51. Informations générales pour le fournisseur d'identité dans la fédération SAML 2.0

Informations générales	Description	Votre valeur
Nom de la fédération	Nom unique que vous attribuez à la fédération.	
Rôle	Rôle que vous fournissez dans la fédération. (Dans les présentes instructions, vous êtes le fournisseur d'identité.)	Fournisseur d'identité

Tableau 52. Informations de contact pour le fournisseur d'identité dans la fédération SAML 2.0

Personne à contacter	Description	Votre valeur
Nom de l'entreprise, adresse URL et nom de contact de l'entreprise.	Nom de votre société et autres informations facultatives sur le contact associé à votre rôle dans la fédération.	

Tableau 53. Protocole de fédération pour le fournisseur d'identité dans la fédération SAML 2.0

Protocole de fédération	Description	Votre valeur
Protocole	Protocole SAML que vous et votre partenaire utilisez dans la fédération.	SAML 2.0

Tableau 54. Informations du serveur point de contact pour le fournisseur d'identité dans la fédération SAML 2.0

Serveur point de contact	Description	Votre valeur
URL du serveur point de contact	Adresse URL donnant accès aux noeuds finals sur le serveur point de contact.	

Tableau 55. Sélection de profil et informations de configuration pour le fournisseur d'identité dans la fédération SAML 2.0

Sélection de profil	Description	Votre valeur
Options de profil SAML 2.0 : Choisissez l'une des options de profil suivantes :	Profil de votre fédération. Pour plus d'informations sur les profils, voir «SAML 2.0», à la page 181.	Vous pouvez choisir une des options suivantes : • Basique • Typique • Tous • Manuel
Basique : Connexion unique de navigateur Web, déconnexion unique	Ce paramètre active les profils suivants avec toutes les liaisons prises en charge : • Liaison SSO de navigateur Web • Déconnexion unique (SLO)	(aucune valeur supplémentaire requise)
Typique : Connexion unique de navigateur Web, déconnexion unique et Gestion des identificateurs de nom	Ce paramètre active les profils suivants avec toutes les liaisons prises en charge : • Liaison SSO de navigateur Web • Déconnexion unique (SLO) • Client ou proxy évolué • Gestion des identificateurs de nom	(aucune valeur supplémentaire requise)

Tableau 55. Sélection de profil et informations de configuration pour le fournisseur d'identité dans la fédération SAML 2.0 (suite)

Sélection de profil	Description	Votre valeur
Activer tous les profils et toutes les liaisons	Si vous sélectionnez l'option **Activer tous les profils et toutes les liaisons**, vous devez être prêt à fournir les informations suivantes dans les panneaux successifs : **Panneau Reconnaissance du fournisseur d'identité** • Nom de domaine commun • URL du service de cookie de domaine commun • Durée de vie du cookie de domaine commun, en secondes. Valeur par défaut : 1 **Panneau Proxy client amélioré** • En-têtes HTTP	**Panneau Reconnaissance du fournisseur d'identité** • Nom de domaine commun • URL du service de cookie de domaine commun • Durée de vie du cookie de domaine commun, en secondes. Valeur par défaut : 1 **Panneau Proxy client amélioré** • En-têtes HTTP
Manuel : Choisissez des profils et des liaisons individuels	Si vous choisissez l'option **Manuel**, vous devez être prêt à sélectionner des profils individuels et des liaisons prises en charge.	Profils et liaisons :

Tableau 56. Informations de signature pour le fournisseur d'identité dans la fédération SAML 2.0

Signatures	Description	Votre valeur
Demander une signature sur les messages et assertions entrants	Case à cocher indiquant que votre partenaire utilise sa clé privée pour signer le message et l'assertion. Par défaut, la case est cochée.	Vous pouvez choisir une des options suivantes : • Le partenaire signe (la case est cochée). • Le partenaire ne signe pas (la case n'est pas cochée).
Sélectionner les messages et assertions sortants qui nécessitent une signature	Boutons indiquant les messages sortants que vous signez. Lorsque le paramètre par défaut est sélectionné, l'ensemble des messages et assertions SAML sortants classiques (à l'exception des objets ArtifactResponse et AuthnResponse) doivent être signés.	Vous pouvez choisir une des options suivantes : • Les ensembles de messages SAML sortants classiques sont signés. • Toutes les assertions et tous les messages SAML sortants sont signés. • Aucune assertion ni aucun message SAML sortant ne sont signés.

Tableau 56. Informations de signature pour le fournisseur d'identité dans la fédération SAML 2.0 (suite)

Signatures	Description	Votre valeur
Sélectionner la clé de signature • Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés • Clé privée utilisée pour la signature des messages	Si vous signez des messages et des assertions, vous devez fournir la clé de signature que vous utilisez pour les signer. **Remarque :** Avant d'effectuer cette tâche, veillez à créer la clé et à l'importer dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.	Nom de fichiers de clés : Mot de passe du fichier de clés : Nom d'alias de clé :

Tableau 57. Informations de chiffrement pour le fournisseur d'identité dans la fédération SAML 2.0

Chiffrement	Description	Votre valeur
Clé de chiffrement: • Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés • Biclé publique/privée utilisée pour les données reçues en provenance de votre partenaire.	Biclé publique/privée utilisée pour le chiffrement. Votre partenaire utilise la clé publique pour chiffrer les données qu'il vous envoie. Vous utilisez la clé privée pour déchiffrer les données que votre partenaire vous envoie. Vous devez indiquer la paire de clés à utiliser. **Remarque :** Avant d'effectuer cette tâche, veillez à créer la clé et à l'importer dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager. Pour plus d'informations, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	Nom de fichiers de clés : Mot de passe du fichier de clés : Nom d'alias de clé :

Tableau 58. Paramètres des messages SAML pour le fournisseur d'identité dans la fédération SAML 2.0

Paramètres des messages	Description	Votre valeur
Options des messages : • Durée de vie du message (en secondes) • Durée de vie de l'artefact (en secondes) • Délai d'expiration de session	Intervalle de temps, spécifié en secondes, pendant lequel les messages, artefacts et sessions sont valides. Les valeurs par défaut sont les suivantes : • Durée de vie des messages : 300 • Durée de vue des artefacts : 120 • Délai d'expiration de session : 7200	Durée de vie du message (en secondes) : Durée de vie de l'artefact (en secondes) : Délai d'expiration de session :
Nécessite un accord pour fédérer	Si vous cochez cette case, vous devez présenter une page à l'utilisateur afin de vérifier que celui-ci a émis une requête de fédération. Valeur par défaut : un accord de fédération est requis.	Vous pouvez choisir une des options suivantes : • Nécessite un accord pour fédérer (la case est cochée). • Ne pas demander d'accord pour fédérer (la case n'est pas cochée).
Noeud final SOAP	URL du point d'extrémité SOAP. Valeur par défaut : la valeur contenue dans cette zone dépend de l'URL du serveur point de contact que vous avez fournie précédemment. **Remarque :** Si aucune liaison SOAP n'est utilisée dans le profil que vous avez sélectionné, l'affichage de cette zone n'a pas lieu.	

Tableau 59. Informations relatives aux paramètres de jeton pour le fournisseur d'identité dans la fédération SAML 2.0

Configuration des paramètres de jeton	Description	Votre valeur
Durée de validité (en secondes) d'une assertion avant sa date d'émission	Durée en secondes pendant laquelle une assertion est considérée comme valide avant sa date de création. Valeur par défaut : 60	
Durée de validité de l'assertion après émission	Durée en secondes pendant laquelle une assertion est considérée comme valide après sa date de création. Valeur par défaut : 60	

Tableau 60. Informations de requête d'attribut pour le fournisseur d'identité

Requête d'attribut	Description	Votre valeur
Activé	Indique si le fournisseur est autorisé à agir en tant qu'autorité d'attribut. Si cette option est sélectionnée, le profil de requête d'attribut est activé.	

Tableau 61. Informations de mappage requête d'attribut pour le fournisseur d'identité

Mappage de requête d'attribut	Description	Votre valeur
Options de mappage de requête d'attribut Vous pouvez choisir une des options suivantes : • Fichier de transformation XSL ou JavaScript contenant les règles de mappage • Module de mappage Tivoli Directory Integrator • Module de mappage personnalisé	Type de mappage de requête d'attribut utilisé. Vous devez sélectionner un fichier XSLT, un module de mappage Tivoli Directory Integrator ou un module de mappage personnalisé. Si vous utilisez un fichier XSLT, ce dernier doit avoir été créé avant la configuration de la fédération. Le module de mappage Tivoli Directory Integrator est un module STS. Le mappage personnalisé est une option avancée. Si vous utilisez cette option, vous devez créer et ajouter un nouveau type et instance de module *avant* de pouvoir l'utiliser dans votre configuration.	L'une des valeurs suivantes : • Chemin de fichier XSLT • Module de mappage Tivoli Directory Integrator • Nom de l'instance de module de mappage personnalisée

Tableau 62. Informations de mappage d'identité pour le fournisseur d'identité dans la fédération SAML 2.0

Mappage d'identité	Description	Votre valeur
Options de mappage d'identité Vous pouvez choisir une des options suivantes : • Fichier de transformation XSL contenant les règles de mappage • Module de mappage personnalisé	Type de mappage d'identité utilisé. Vous devez savoir si vous devez utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous souhaitez utiliser cette option, vous devez créer votre module de mappage et l'ajouter à l'environnement en tant que module type et module d'instance *avant* de pouvoir l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	L'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété les tables, poursuivez avec les instructions de la rubrique «Création de votre rôle dans la fédération».

Création de votre rôle dans la fédération

Utilisez la console pour créer une fédération. Pour commencer, l'assistant Fédération vous invite à renseigner les informations relatives à votre rôle dans la fédération. Pour obtenir la description des zones que l'assistant vous invite à renseigner, consultez l'aide en ligne.

Avant de commencer

Avant de commencer cette procédure, complétez la formulaire standard SAML approprié, en indiquant le rôle que vous tenez dans la fédération :

- «Formulaire de fournisseur de services IDP SAML 1.x», à la page 217
- «Formulaire de fournisseur d'identité SAML 1.x», à la page 219
- «Formulaire de fournisseur de services SAML 2.0», à la page 222
- «Formulaire de fournisseur d'identité SAML 2.0», à la page 228

Pourquoi et quand exécuter cette tâche

Pendant la configuration, il se peut que vous soyez invité à redémarrer WebSphere Application Server. Avant de poursuivre la tâche, assurez-vous que le serveur a redémarré entièrement.

Procédure

1. Connectez-vous à la console.

2. Sélectionnez **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Le portlet Fédérations affiche plusieurs boutons d'action.
3. Cliquez sur **Créer**. L'assistant de fédération démarre. Le panneau Informations générales s'ouvre.
4. A l'aide de votre formulaire, complétez les panneaux affichés par l'assistant Fédération. Utilisez le formulaire complété comme guide pour renseigner les zones affichées. Si vous avez besoin de revenir à un panneau précédent, cliquez sur **Précédent**. Si vous souhaitez mettre fin à la configuration, cliquez sur **Annuler**. Sinon, cliquez sur **Suivant** après avoir complété chaque panneau. Lorsque vous avez complété tous les écrans de configuration, le panneau Récapitulatif s'affiche.
5. Vérifiez que les paramètres de configuration sont corrects.
6. Cliquez sur **Terminer**. Le portlet Création de fédération terminée s'affiche.
7. Vous pouvez ajouter votre partenaire maintenant ou ultérieurement. Choisissez-en un :
   - Cliquez sur **Ajouter un partenaire** pour lancer l'assistant Partenaire et ajouter la configuration de votre partenaire en suivant les étapes décrites à la section :
     a. «Obtention des données de configuration de fédération de la part de votre partenaire», à la page 239, et complétez le formulaire approprié en fonction du rôle de votre partenaire dans la fédération.
     b. «Ajout à votre partenaire», à la page 271.
   - Pour ajouter votre partenaire ultérieurement, cliquez sur **Terminé**. Vous serez redirigé vers l'écran des fédérations.

Configuration d'un serveur point de contact WebSEAL pour la fédération SAML

Si vous envisagez d'utiliser WebSEAL en tant que serveur point de contact, vous devez le configurer pour la fédération SAML.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Ces instructions ont pour hypothèse que le profil du point de contact WebSEAL est activé.

Pourquoi et quand exécuter cette tâche

L'assistant de fédération comporte un bouton qui vous permet de récupérer l'outil de configuration Tivoli Federated Identity Manager. Vous devez obtenir cet outil, puis l'exécuter. Pour configurer WebSEAL en tant que serveur point de contact, procédez comme suit.

Procédure

1. Une fois la fédération créée, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour recharger vos modifications.

   Remarque : La console de gestion vous offre la possibilité d'ajouter immédiatement un partenaire, mais pour cette configuration initiale de la fédération, vous devez d'abord exécuter d'autres tâches.
2. Cliquez sur **Terminé** pour revenir au panneau Fédérations.
3. Cliquez sur **Télécharger l'outil de configuration Tivoli Access Manager**.
4. Enregistrez l'outil de configuration sur le système de fichiers de l'ordinateur hébergeant le serveur WebSEAL.
5. Démarrez l'outil de configuration depuis une ligne de commande. La syntaxe est la suivante :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf
   ```

   Remarque : Si la norme FIPS (Federal Information Processing Standards) est activée pour votre environnement, une fabrique de connexions sécurisées doit être indiquée. Par exemple :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf -sslfactory TLS
   ```

   Vous aurez besoin de l'ID (par défaut : sec_master) et du mot de passe de l'utilisateur d'administration Tivoli Access Manager. L'utilitaire configure les noeuds finals sur le serveur WebSEAL, crée une jonction WebSEAL, relie les listes de contrôle d'accès adaptées et active les méthodes d'authentification adéquates.

Exemple

Par exemple, lorsque vous avez mis le fichier tfimcfg.jar dans le répertoire `/tmp` et que le nom de l'instance WebSEAL est `default`, la commande est la suivante :

```
java -jar /tmp/tfimcfg.jar -action tamconfig -cfgfile webseald-default
```

Pour plus d'informations, voir Annexe A, «Référence de tfimcfg», à la page 827.

Configuration de WebSphere en tant que serveur point de contact

Tivoli Federated Identity Manager est configuré par défaut pour utiliser Tivoli Access Manager WebSEAL en tant que serveur point de contact. Pour configurer WebSphere en tant que serveur point de contact, vous devez procéder à une modification de la configuration.

Procédure

1. Connectez-vous à la console d'administration.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Sélectionnez **WebSphere**
4. Cliquez sur **Activer**.

Résultats

Le serveur WebSphere est désormais configuré en tant que point de contact.

Délivrance d'instructions à votre partenaire

Lorsque vous travaillez en collaboration avec des partenaires en vue d'établir une fédération, vous devez leur fournir des informations et en recueillir également de leur part.

Selon le rôle que vous exercez au sein de vos fédérations, vous pouvez être amené à envisager la délivrance de conseils ou d'une assistance à votre partenaire, en outre des informations de configuration que vous leur fournissez. L'expérience de votre partenaire peut vous aider à déterminer le meilleur moyen de lui porter assistance.

Les partenaires ayant une expérience des connexions uniques sauront probablement besoin d'une aide limitée, par exemple via une assistance téléphonique ou par courriel. En revanche, les partenaires inexpérimentés en matière de connexion unique peuvent avoir besoin d'une réelle orientation, telle qu'un tutoriel ou une description écrite.

Le moment propice pour fournir ces instructions est laissé à votre discrétion. Il peut être souhaitable de les délivrer en même temps que vous sollicitez des informations de la part de votre partenaire. Vous pouvez également choisir de partager des informations de présentation dès les premières étapes de votre relation fédérée, avant que les procédures de configuration ne soient mises en place.

Le plan indiqué ci-après a pour but de vous aider à élaborer une instruction documentée à l'attention de votre partenaire. Ce plan suppose que vous êtes le partenaire responsable de la délivrance des instructions. En revanche, si vous êtes celui qui a besoin d'aide, envisagez de transmettre les instructions à votre partenaire, ou de faire évoluer le plan au profit d'un questionnaire que vous pourrez utiliser pour solliciter des informations de sa part.

Plan du guide d'intégration

1. Introduction

a. Décrivez la connexion unique et expliquez en quoi consiste l'usage de Tivoli Federated Identity Manager.

b. Définissez les termes tels que fédération, fournisseur d'identité, fournisseur de services ainsi que, le cas échéant, protocole, profil et liaison.

c. Identifiez le rôle que vous exercez, ainsi que celui de votre partenaire dans la fédération.

d. Décrivez le mode d'interaction entre les utilisateurs finaux de votre site, ainsi que celui de votre partenaire.

Par exemple : identifiez le service auquel les utilisateurs finals tentent d'accéder. Evaluez la possibilité d'inclure un graphique qui identifie le flux d'activités établi parmi les participants, tels que l'utilisateur final qui se connecte, le fournisseur d'identité qui authentifie l'utilisateur, le fournisseur de services qui délivre l'accès et l'utilisateur final qui accède au service.

2. Spécifications techniques

a. Définissez les exigences ou options relatives aux protocoles, aux liaisons et aux profils.

Vous avez peut-être besoin, par exemple, que votre partenaire utilise le protocole SAML 1.1 avec un artefact de navigateur. Ou bien, vous avez besoin que votre partenaire utilise le protocole 1.1, mais en laissant au partenaire le choix du type de profil.

b. Expliquez les exigences ou options relatives aux assertions.

Vous pouvez par exemple souhaiter que le partenaire inclue des zones spécifiques dans l'assertion, telles qu'une clé de mappage de groupe d'utilisateurs avec un identificateur individuel. Vous pouvez également expliquer que la spécification de certaines options d'assertion est nécessaire, par exemple pour la durée de vie de l'assertion ou de l'artefact (en cas d'utilisation de l'artefact de navigateur), ou encore les informations de signature.

c. Présentez les limitations quant aux types de périphériques que vous pouvez utiliser dans le cadre d'une fonction de connexion unique.

Il se peut, par exemple, que la fédération prenne uniquement en charge l'interaction des utilisateurs finals avec le navigateur Web.

d. Décrivez les exigences relatives à l'audit et à la consignation. Pour plus d'informations, voir la rubrique *IBM Tivoli Federated Identity Manager - Guide d'audit*.

e. Expliquez de quelle manière les utilisateurs sont amenés à voir s'afficher des messages d'événement lors des interactions avec la fédération.

A titre d'exemple, si vous êtes un fournisseur de services, vous pouvez proposer à votre partenaire des options de personnalisation, relatives aux circonstances dans lesquelles les utilisateurs se déconnectent ou reçoivent des messages de dépassement de délai d'attente, ou d'autres événements en provenance du système. Si vous êtes un fournisseur d'identité, vous pouvez proposer à votre partenaire des options de personnalisation, relatives aux circonstances dans lesquelles les utilisateurs se connectent.

f. Convenez, avec votre partenaire, du mode de synchronisation entre les horloges système.

III. Sécurité

a. Définissez les exigences relatives à SSL.

b. Demandez des informations de certificat (telles que le nom de l'autorité de certification ayant émis le certificat de votre partenaire, ou une copie du certificat de votre partenaire).

c. Expliquez les exigences ou options relatives à la signature.

IV. Echange de données

Etablissez le mode d'échange des données dans la fédération, y compris les clés.

Dans les fédérations SAML 1.x, l'échange de données peut s'effectuer manuellement ou par le biais d'un fichier de métadonnées. Dans SAML 2.0, l'usage d'un fichier de métadonnées est obligatoire.

Si une méthode manuelle est utilisée, répertoriez les informations dont vous avez besoin de votre partenaire. Pour cela, basez-vous sur les formulaires contenus dans les rubriques «Obtention des données de configuration de fédération de la part de votre partenaire», à la page 239 et «Transmission des propriétés de la fédération au partenaire», à la page 273.

V. Test

Décrivez votre aptitude à effectuer les tests de la fédération et mentionnez les exigences éventuelles auxquelles votre partenaire doit se conformer avant d'utiliser la fédération dans un environnement de production. Evaluez la possibilité d'inclure les URL nécessaires à votre partenaire aux fins de test.

A titre d'exemple, si vous êtes le fournisseur de services d'une fédération SAML 1.x, il se peut que vous deviez fournir à votre partenaire une URL cible et une URL d'assertion de client.

VI. Production

Décrivez les conditions à remplir pour que la fédération soit prête pour la production. Vous pouvez inclure des URL de production ou expliquer comment vous les produirez ultérieurement.

VII. Support

Expliquez la manière dont la prise en charge au niveau utilisateur ou au niveau administrateur est gérée dans la fédération.

VIII. Formulaire de partenaire

Il se peut qu'à différents stades au cours des sections précédentes, vous ayez demandé des informations à votre partenaire, ou que vous lui ayez expliqué les raisons de cette demande d'informations.

A la fin de votre document, envisagez d'ajouter un formulaire sur lequel votre partenaire aura la possibilité d'inscrire les informations demandées. La feuille de travail peut contenir des zones, telles que :

- Adresses URL de noeuds finals aux fins de test
- Adresses URL de noeuds finals pour la production
- Personne à contacter
- Informations relatives aux certificats SSL (nom de l'autorité de certification, etc.)
- Informations de signature (nature des informations signées ou validées, etc.)
- Méthode d'échange de données (manuelle ou métadonnées). Si une méthode manuelle est utilisée, il se peut que vous deviez ajouter votre formulaire d'autres zones relatives aux informations demandées.

Obtention des données de configuration de fédération de la part de votre partenaire

Vous devez obtenir des informations de configuration de la part de votre partenaire avant d'ajouter ce partenaire à une fédération.

Le partenaire peut exporter la configuration de la fédération dans un fichier de métadonnées ou, si ce partenaire utilise SAML 1.x, il peut vous la communiquer manuellement (la configuration manuelle par les partenaires n'est pas prise en charge dans les fédérations SAML 2.0).

Pour rassembler plus facilement les informations appropriées en provenance de votre partenaire, complétez la formulaire relatif à la norme SAML que vous envisagez d'utiliser dans la fédération, ainsi que pour le rôle attribué à votre partenaire dans cette fédération :

- *Si vous êtes le fournisseur d'identité*, ajoutez un partenaire fournisseur de services. Utilisez le formulaire destiné au partenaire fournisseur de services en fonction de la norme SAML que vous utilisez dans votre fédération :
  - «Formulaire pour fournisseur de services partenaire SAML 1.x», à la page 240

– «Formulaire de fournisseur de services partenaire SAML 2.0», à la page 253

- *Si vous êtes le fournisseur de services*, ajoutez un partenaire fournisseur d'identité. Utilisez le formulaire destiné au partenaire fournisseur d'identité en fonction de la norme SAML que vous utilisez dans votre fédération :
  – «Formulaire pour fournisseur d'identité partenaire SAML 1.x», à la page 246
  – «Formulaire de fournisseur d'identité partenaire SAML 2.0», à la page 261

Lorsque vous avez rassemblé les informations de configuration de votre partenaire, vous pouvez alors utiliser l'assistant de partenaire, dans la console, pour ajouter les propriétés de fédération de votre partenaire. Voir «Ajout à votre partenaire», à la page 271.

Formulaire pour fournisseur de services partenaire SAML 1.x

Vous devez ajouter un fournisseur de services partenaire à votre fédération si vous êtes un fournisseur d'identité qui utilise SAML 1.x. Certaines informations peuvent vous être fournies dans un fichier de métadonnées ou l'ensemble des informations peuvent vous être fournies manuellement.

Utilisez le formulaire suivant pour rassembler les informations nécessaires en provenance de votre partenaire. Modifiez ce formulaire de sorte qu'il reflète les informations spécifiques que votre partenaire doit vous fournir. Vous devez également demander à votre partenaire de compléter ce formulaire modifié.

Tableau 63. Options relatives aux métadonnées pour l'ajout d'un fournisseur de services partenaire dans une fédération SAML 1.x

Options de métadonnées	Description	Vos valeurs
Spécifiez manuellement les paramètres SAML **Importez le fichier de métadonnées**	Indique la manière dont vous devez entrer les données relatives au partenaire. Vous pouvez recevoir un fichier de métadonnées de votre partenaire ou saisir manuellement les informations de votre partenaire. Si vous optez pour l'importation d'un fichier de métadonnées, vous devez connaître son nom et son emplacement.	choisissez l'une des options suivantes : • Spécifiez manuellement les paramètres SAML • Importez le fichier de métadonnées et spécifiez le nom et le chemin d'accès du fichier :

Tableau 64. Informations de contact pour le fournisseur de services partenaire dans la fédération SAML 1.x

Personne à contacter	Description	Votre valeur
Remarque : Ce panneau s'ouvre uniquement si vous saisissez manuellement les informations de partenaire.		
Nom de l'entreprise, adresse URL et personne à contacter	Nom de l'entreprise et, le cas échéant, autres informations relatives au contact associé à la fédération.	Nom de l'entreprise :

Tableau 65. Paramètres des messages SAML pour le fournisseur de services partenaire dans une fédération SAML 1.x

Paramètres des messages SAML	Description	Votre valeur
Remarque : Ce panneau s'ouvre uniquement si vous saisissez manuellement les informations de partenaire.		
ID fournisseur	Adresse URL du serveur point de contact du fournisseur de services, également utilisée en tant qu'ID de fournisseur.	ID fournisseur :
URL du service d'assertion client	Adresse URL du noeud final du service d'assertion client sur le site du fournisseur de services.	URL du service d'assertion client :
Le partenaire utilise le profil POST du navigateur pour la connexion unique	Case à cocher indiquant que le fournisseur de services partenaire utilise le POST du navigateur.	Vous pouvez choisir une des options suivantes : • Le partenaire utilise le profil POST du navigateur (cochez la case). • Le partenaire n'utilise pas le profil POST du navigateur (ne pas cocher la case).

Tableau 66. Informations relatives à la validation de signature pour le fournisseur de services partenaire dans une fédération SAML 1.x

Signatures	Description	Votre valeur
Algorithme de signature pour la signature de messages SAML	Spécifie l'algorithme de signature à utiliser pour la transaction. La clé sélectionnée utilisée pour signer les messages SAML doit correspondre à l'option choisie dans le menu déroulant pour éviter l'échec de la signature. Sélectionnez l'algorithme de signature parmi les options suivantes. • RSA-SHA1 • DSA-SHA1 • RSA-SHA256	
Valider les signatures sur les requêtes d'artefact	Vous pouvez valider les signatures de message SAML lorsque l'artefact du navigateur est utilisé. Pour utiliser cette option, cochez la case Validation de signatures.	Vous pouvez choisir une des options suivantes : • Valider les signatures pour l'artefact (cochez la case). • Ne pas valider les signatures pour l'artefact (ne pas cocher la case).

Tableau 66. Informations relatives à la validation de signature pour le fournisseur de services partenaire dans une fédération SAML 1.x (suite)

Signatures	Description	Votre valeur
Sélectionner le fichier de clés certifiées ou la clé de validation • Fichier de clés dans Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés certifiées • Clé publique utilisée pour effectuer la validation	Si vous choisissez de valider les messages lorsque l'artefact du navigateur est utilisé, vous devez indiquer une clé pour cette validation. Il doit s'agir de la clé publique correspondant à la clé privée que votre partenaire utilise pour signer les messages. **Remarque :** Si vous importez les données de votre partenaire, la clé est fournie dans le fichier de métadonnées. Vous êtes invité à choisir un fichier de clés pour la clé. Veillez à créer le fichier de clés avant d'effectuer cette tâche. Si vous saisissez manuellement les données de votre partenaire, assurez-vous que vous avez obtenu la clé. Importez ensuite la clé dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager avant d'effectuer cette tâche. Pour plus d'informations, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	**Méthode utilisant les métadonnées :** • Nom du fichier de clés certifiées : • Mot de passe du fichier de clés certifiées : • Libellé de la clé : **Méthode manuelle :** • Nom du fichier de clés certifiées : • Mot de passe du fichier de clés certifiées : • nom d'alias de clé :

Tableau 67. Informations relatives aux paramètres jetons de sécurité pour le fournisseur de services partenaire dans une fédération SAML 1.x

Configuration du jeton de sécurité	Description	Votre valeur
Signer les assertions SAML	Vous avez la possibilité de signer les assertions SAML.	Vous pouvez choisir une des options suivantes : • Activer les signatures SAML (cochez la case). • Ne pas activer les signatures (ne pas cocher la case).

Tableau 67. Informations relatives aux paramètres jetons de sécurité pour le fournisseur de services partenaire dans une fédération SAML 1.x (suite)

Configuration du jeton de sécurité	Description	Votre valeur
Sélectionner la clé de signature • Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés • Clé privée utilisée pour la signature de l'assertion.	Si vous choisissez de signer les assertions, vous devez sélectionner un fichier de clés et une clé. **Remarque :** Créez le fichier de clés et la clé avant d'effectuer cette tâche. Pour plus d'informations, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	• Nom de fichiers de clés : • Mot de passe du fichier de clés : • nom d'alias de clé :
Inclure les données de certificat X509	Si vous optez pour la signature de l'assertion SAML, indiquez si vous souhaitez que les données de certificat codées en base 64 soient incluses dans votre signature. L'action par défaut consiste à inclure les données de certificat X.509 (**Oui**). Vous pouvez également choisir d'exclure les données du certificat X.509 (**Non**).	
Inclure les détails de l'émetteur de sujet X509	Si vous optez pour la signature de l'assertion SAML, indiquez si vous souhaitez que le nom de l'émetteur et le numéro de série du certificat soient inclus dans votre signature. L'action par défaut consiste à exclure (**Non**) les détails relatifs à l'émetteur de sujet X.509. Vous pouvez également choisir d'inclure les détails de l'émetteur de sujet X.509 (**Oui**).	

Tableau 67. Informations relatives aux paramètres jetons de sécurité pour le fournisseur de services partenaire dans une fédération SAML 1.x (suite)

Configuration du jeton de sécurité	Description	Votre valeur
Inclure le nom de sujet X509	Si vous optez pour la signature de l'assertion SAML, indiquez si vous souhaitez que le nom de sujet soit inclus dans votre signature. L'action par défaut consiste à exclure le nom de sujet X.509 (**Non**). Vous pouvez également choisir d'inclure le nom de sujet X.509 (**Oui**).	
Inclure l'identificateur de clé de sujet X509	Si vous optez pour la signature de l'assertion SAML, indiquez si vous souhaitez que l'identificateur de clé de sujet soit inclus dans votre signature. L'action par défaut consiste à exclure l'identificateur de clé de sujet (**Non**). Vous pouvez également choisir d'inclure l'identificateur de clé de sujet X.509 (**Oui**).	
Inclure la clé publique	Si vous optez pour la signature de l'assertion SAML, indiquez si vous souhaitez que la clé publique soit incluse dans votre signature. L'action par défaut consiste à exclure la clé publique (**Non**). Vous pouvez également choisir d'inclure la clé publique (**Oui**).	
Inclure l'élément InclusiveNamespaces	Si vous optez pour la signature de l'assertion SAML, vous pouvez sélectionner cette option afin d'inclure l'élément InclusiveNamespaces à la canonicalisation de l'assertion lors de la création de la signature. Par défaut, cette option est désélectionnée.	

Tableau 67. Informations relatives aux paramètres jetons de sécurité pour le fournisseur de services partenaire dans une fédération SAML 1.x (suite)

Configuration du jeton de sécurité	Description	Votre valeur
Algorithme de signature pour la signature d'assertions SAML	Spécifie l'algorithme de signature à utiliser pour la transaction. La clé sélectionnée utilisée pour signer les assertions SAML doit correspondre à l'option choisie dans le menu déroulant pour éviter l'échec de la signature. Sélectionnez l'algorithme de signature parmi les options suivantes. • RSA-SHA1 • DSA-SHA1 • RSA-SHA256	
Inclure les types d'attributs suivants	Cochez la case pour indiquer les types d'attributs à inclure dans l'assertion. L'astérisque (*), qui est le paramètre par défaut, indique que tous les types d'attribut spécifiés dans le fichier de mappage d'identité ou par le module de mappage personnalisé sont inclus dans l'assertion. Pour spécifier un ou plusieurs types d'attributs individuellement, tapez chaque type d'attribut dans la case. Par exemple, si vous souhaitez inclure uniquement des attributs du type urn:oasis:names:tc:SAML:2.0:assertion, entrez cette valeur dans la zone. Utilisez && pour séparer plusieurs types d'attributs.	

Tableau 68. Informations de mappage d'identité pour le fournisseur de services partenaire dans la fédération SAML 1.x

Mappage d'identité	Description	Votre valeur
Options de mappage d'identité Vous pouvez choisir une des options suivantes : • Module de mappage personnalisé • Fichier de transformation XSL contenant les règles de mappage • Ne sélectionnez aucune option si vous souhaitez appliquer l'option de mappage d'identité actuellement définie pour la fédération.	Type de mappage d'identité à utiliser avec votre partenaire. Vous pouvez ne sélectionner aucune de ces options si vous souhaitez que ce partenaire choisisse l'option de mappage d'identité déjà configurée pour la fédération. Sinon, vous pouvez choisir une option de mappage spécifique et l'appliquer à ce partenaire. Pour choisir une option de mappage, vous devez savoir si vous devez utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous envisagez d'utiliser cette option, votre module de mappage doit d'abord être créé et ajouté à l'environnement en tant que type et instance de module. Vous pouvez ensuite l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	Ne sélectionnez aucune option pour utiliser la configuration de mappage existante. Sinon, utilisez l'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété ce formulaire, poursuivez avec les étapes de la rubrique «Ajout à votre partenaire», à la page 271.

Formulaire pour fournisseur d'identité partenaire SAML 1.x

Vous devez ajouter un fournisseur d'identité partenaire à votre fédération si vous êtes un fournisseur de services qui utilise SAML 1.x. Certaines informations peuvent vous être fournies dans un fichier de métadonnées ou l'ensemble des informations peuvent vous être fournies manuellement.

Utilisez le formulaire suivant pour rassembler les informations nécessaires en provenance de votre partenaire. Modifiez ce formulaire afin qu'il reflète les informations spécifiques que votre partenaire doit vous fournir. Vous devez également demander à votre partenaire de compléter le formulaire modifié.

Tableau 69. Options relatives aux métadonnées pour l'ajout d'un fournisseur d'identité partenaire dans une fédération SAML 1.x

Options de métadonnées	Description	Vos valeurs
Spécifiez manuellement les paramètres SAML **Importez le fichier de métadonnées**	Indique le mode de saisie des informations relatives au partenaire. Vous pouvez recevoir un fichier de métadonnées de votre partenaire ou saisir manuellement les données de votre partenaire. Si vous choisissez d'importer un fichier de métadonnées, vous devez connaître le nom du fichier et l'emplacement des métadonnées.	choisissez l'une des options suivantes : • Spécifiez manuellement les paramètres SAML • Importez le fichier de métadonnées et spécifiez le nom et le chemin d'accès du fichier :

Tableau 70. Informations de contact pour le fournisseur d'identité partenaire dans une fédération SAML 1.x

Personne à contacter	Description	Votre valeur
Remarque : Ce panneau s'ouvre uniquement si vous entrez manuellement les informations relatives au partenaire.		
Nom de l'entreprise, adresse URL et personne à contacter	Nom de l'entreprise et, le cas échéant, autres informations relatives au contact associé à la fédération.	Nom de l'entreprise :

Tableau 71. Paramètres des messages SAML pour le fournisseur d'identité partenaire dans une fédération SAML 1.x

Paramètres des messages SAML	Description	Votre valeur
Remarque : Ce panneau s'ouvre uniquement si vous entrez manuellement les informations relatives au partenaire.		
ID fournisseur	Adresse URL du serveur point de contact du fournisseur de services, également utilisée en tant qu'ID de fournisseur.	ID fournisseur :
ID source • Générer l'ID source automatiquement • Indiquez une valeur explicite pour l'ID source	Vous avez la possibilité de générer un ID source pour le partenaire, ou d'en indiquer un.	ID source :
Noeuds finals • URL du service de transfert inter-sites • URL du service de résolution des artefacts	Adresses URL renvoyant aux noeuds finals du service de transfert inter-sites et du service de résolution d'artefacts.	URL du service de transfert inter-sites : URL du service de résolution des artefacts :

Tableau 72. Informations relatives à la validation de signature pour le fournisseur d'identité partenaire dans une fédération SAML 1.x

Validation des signatures	Description	Votre valeur
Algorithme de signature pour la signature des demandes de résolution d'artefact SAML	Indique l'algorithme de signature à utiliser pour la transaction. La clé sélectionnée utilisée pour la signature des demandes de résolution d'artefact SAML doit correspondre à l'option choisie dans le menu déroulant pour éviter l'échec de la signature. **Remarque :** Cette option ne s'affiche pas si vous n'avez pas choisi de signer la demande de résolution d'artefact pour la fédération. Sélectionnez l'algorithme de signature parmi les options suivantes. • RSA-SHA1 • DSA-SHA1 • RSA-SHA256	
Les messages SAML pour le profil POST du navigateur sont signés et doivent être validés (obligatoire) **Valider les signatures sur les messages SAML pour le profil d'artefact** (facultatif)	• Lorsque le POST du navigateur est utilisé en tant que profil, les messages SAML doivent être signés et validés. Cette option est donc présélectionnée et ne peut pas être supprimée. • Il est également possible de valider les signatures de message SAML lorsque l'artefact du navigateur est utilisé.	Vous pouvez choisir une des options suivantes : • Valider les signatures pour l'artefact (cochez la case). • Ne pas valider les signatures pour l'artefact (ne pas cocher la case).

Tableau 72. Informations relatives à la validation de signature pour le fournisseur d'identité partenaire dans une fédération SAML 1.x (suite)

Validation des signatures	Description	Votre valeur
Sélectionner le fichier de clés certifiées ou la clé de validation • Fichier de clés dans Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés certifiées • Clé publique à utiliser pour valider la signature de votre partenaire	Etant donné que les messages POST du navigateur doivent être signés et validés, vous êtes tenu de spécifier une clé permettant de valider la signature. Si vous choisissez de valider également des messages lors de l'utilisation de l'artefact du navigateur, la même clé est utilisée pour les valider. La clé publique que vous utilisez correspond à la clé privée que votre partenaire utilise pour signer les messages. **Remarque :** Si vous importez les données de votre partenaire, la clé est fournie dans le fichier de métadonnées. Il vous est demandé de choisir un fichier de clés pour cette clé. Assurez-vous que vous avez créé le fichier de clés avant d'effectuer cette tâche. Si vous saisissez manuellement les données de votre partenaire, assurez-vous que vous avez obtenu la clé auprès de votre partenaire. Importez ensuite la clé dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager avant d'effectuer cette tâche. Pour plus d'informations, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	**Méthode utilisant les métadonnées :** • Nom du fichier de clés certifiées : • Mot de passe du fichier de clés certifiées : • Libellé de la clé : **Méthode manuelle :** • Nom du fichier de clés certifiées : • Mot de passe du fichier de clés certifiées : • nom d'alias de clé :

Tableau 73. Validation de certificat serveur pour votre fournisseur d'identité partenaire dans une fédération SAML 1.x

Validation de certificat serveur pour SOAP	Description	Votre valeur
Sélectionner le certificat de validation du serveur	Clé publique du certificat qui s'affiche lors de la communication SSL avec votre partenaire. Vous et votre partenaire devez convenir ensemble du certificat à utiliser. Vous devez avoir déjà obtenu le certificat, ainsi que le fichier de clés de celui-ci. Voir «Réception certificat serveur de votre partenaire», à la page 83.	Nom du fichier de clés certifiées : Mot de passe du fichier de clés certifiées : Nom du certificat :

Tableau 74. Authentification du client SOAP pour votre fournisseur d'identité partenaire dans une fédération SAML 1.x

Authentification client pour SOAP	Description	Votre valeur
Informations d'authentification client L'une des options suivantes : • **Authentification de base** – Username – Mot de passe • **Authentification par certificat client** – Certificat que vous devez présenter au serveur du fournisseur d'identité. Certificat que vous et votre fournisseur d'identité partenaire avez convenu de présenter. – Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée – Mot de passe du fichier de clés	Si l'authentification réciproque est exigée par votre partenaire, vous devez en connaître le type. S'il s'agit d'une authentification standard, vous avez besoin d'un nom d'utilisateur et d'un mot de passe. S'il s'agit d'une authentification par certificat client, vous avez besoin du certificat que vous et votre partenaire avez convenu d'utiliser. **Remarque :** Si vous avez besoin d'un certificat, assurez-vous que vous avez convenu avec votre partenaire de l'emplacement où vous devez l'obtenir. Importez-le ensuite dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager avant d'effectuer cette tâche. Voir «Obtention de votre certificat client», à la page 84.	Vous pouvez choisir une des options suivantes : • Informations d'authentification de base : – Nom d'utilisateur : – Mot de passe : • Informations relatives à l'authentification par certificat client – Nom de fichiers de clés : – Mot de passe du fichier de clés : – Alias de clé :

Tableau 75. Informations relatives aux paramètres jetons de sécurité pour le fournisseur d'identité partenaire dans une fédération SAML 1.x

Configuration du jeton de sécurité	Description	Votre valeur
Activer la validation des signatures	Si votre partenaire signe les assertions, vous pouvez choisir de valider ces signatures. Dans certains cas, votre partenaire exige que vous validiez ces signatures.	Vous pouvez choisir une des options suivantes : • Activer les signatures de validation (cochez la case). • Ne pas valider les signatures (ne pas cocher la case).
Sélectionner la clé de validation	Spécifiez le type de validation de signature à utiliser.	Vous pouvez choisir une des options suivantes : • Utiliser l'élément KeyInfo de la signature XML pour rechercher le certificat X.509 pour la validation de signature • Utiliser l'alias du fichier de clés pour trouver une clé publique pour la validation des signatures (action par défaut). • Spécifier l'expression DN du sujet pour les certificats X.509 autorisés.
Sélectionner la clé et le fichier de clés certifiées • Fichier de clés dans Tivoli Federated Identity Manager, dans lequel la clé est stockée • Mot de passe du fichier de clés certifiées • Clé publique à utiliser pour valider la signature	Si vous choisissez de valider les signatures des assertions, ou que cette validation est exigée par votre partenaire, vous devez sélectionner un fichier de clés et une clé. **Remarque :** La clé publique que vous devez utiliser correspond à la clé privée que votre partenaire utilise pour signer les assertions. Obtenez cette clé et créez le fichier de clés avant d'effectuer cette tâche. (Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	• Nom du fichier de clés certifiées : • Mot de passe du fichier de clés certifiées : • nom d'alias de clé :

Tableau 75. Informations relatives aux paramètres jetons de sécurité pour le fournisseur d'identité partenaire dans une fédération SAML 1.x (suite)

Configuration du jeton de sécurité	Description	Votre valeur
Créer plusieurs instructions d'attribut dans Universal User.	Sélectionnez cette option pour conserver plusieurs instructions d'attribut dans les groupes dans lesquels elles ont été reçues. Cette option peut se révéler nécessaire si vos règles de mappage d'identité personnalisées sont écrites de manière à s'appliquer à un ou plusieurs groupes d'instructions d'attribut spécifiques. Si cette case n'est pas activée, plusieurs instructions d'attribut sont organisées dans un seul groupe (AttributeList) dans le document STSUniversalUser. Par défaut, cette option est désélectionnée, ce qui convient à la plupart des configurations.	

Tableau 76. Informations de mappage d'identité pour le fournisseur d'identité partenaire dans la fédération SAML 1.x

Mappage d'identité	Description	Votre valeur
Options de mappage d'identité Vous pouvez choisir une des options suivantes : • Module de mappage personnalisé • Fichier de transformation XSL contenant les règles de mappage • Ne sélectionnez aucune option si vous souhaitez appliquer l'option de mappage d'identité actuellement définie.	Type de mappage d'identité à utiliser avec ce partenaire. Vous pouvez ne sélectionner aucune de ces options si vous souhaitez que ce partenaire choisisse l'option de mappage d'identité déjà configurée pour la fédération. Sinon, vous pouvez choisir une option de mappage spécifique et l'appliquer à ce partenaire. Pour choisir une option de mappage, vous devez choisir d'utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous envisagez d'utiliser cette option, votre module de mappage doit d'abord être créé et ajouté à l'environnement en tant que type et instance de module. Vous pouvez ensuite l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	Ne sélectionnez aucune option pour utiliser la configuration de mappage existante. Sinon, sélectionnez l'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété ce formulaire, poursuivez avec les étapes de la rubrique «Ajout à votre partenaire», à la page 271.

Formulaire de fournisseur de services partenaire SAML 2.0

Si vous utilisez SAML 2.0 dans votre rôle en tant que fournisseur d'identité, vous devez ajouter un fournisseur de services partenaire à votre fédération.

Utilisez le formulaire suivant pour rassembler les informations nécessaires en provenance de votre partenaire. Modifiez ce formulaire afin qu'il reflète les informations spécifiques que votre partenaire doit vous fournir et demandez à votre partenaire de compléter ce formulaire modifié.

Tableau 77. Fédération à laquelle vous ajoutez un fournisseur de services partenaire dans une fédération SAML 2.0

Sélection de fédération	Description	Votre valeur
Nom de la fédération	Nom de la fédération à laquelle vous ajoutez le partenaire.	

Tableau 78. Fichier de métadonnées délivré par votre fournisseur de services partenaire dans une fédération SAML 2.0

Importer des métadonnées	Description	Votre valeur
Fichier de métadonnées	Nom et chemin d'accès du fichier que vous avez obtenu de votre partenaire et qui contient les informations de configuration de votre partenaire.	

Tableau 79. Validation de signature de votre fournisseur de services partenaire dans une fédération SAML 2.0

Validation des signatures	Description	Votre valeur
Sélectionner les assertions et les messages SAML entrants qui nécessitent une signature	Ces options indiquent les messages entrants signés par votre partenaire. Lorsque le paramètre par défaut est sélectionné, l'ensemble des messages et assertions SAML entrants classiques (à l'exception des objets ArtifactResponse et AuthnResponse) doivent être signés.	L'une des options suivantes : • Les ensembles d'assertions et de messages SAML entrants classiques sont signés. • Toutes les assertions et tous les messages SAML entrants sont signés. • Aucune assertion ni aucun message SAML entrant ne sont signés.
Fichier de clés	Le fichier de clés certifiées dans lequel vous stockez la clé que votre partenaire a fournie pour valider sa signature dans les messages lors de la signature des messages et des assertions. Le fichier de clés doit avoir été déjà créé pour cette clé. Pour plus détails, voir «Préparation des fichiers de clés», à la page 51.	Nom du fichier de clés certifiées : Mot de passe du fichier de clés certifiées : Libellé de la clé :

Tableau 80. Fichier de clés destiné au stockage de la clé de chiffrement délivrée par votre fournisseur de services partenaire dans une fédération SAML 2.0

Chiffrement	Description	Votre valeur
Fichier de clés	Fichier de clés certifiées dans lequel vous stockez la clé pour chiffrer les messages envoyés à votre partenaire. Cette option s'affiche, car votre partenaire vous a fourni dans ses métadonnées une clé publique à utiliser pour le chiffrement. Le fichier de clés doit avoir été déjà créé pour cette clé. Pour plus détails, voir «Préparation des fichiers de clés», à la page 51.	Nom du fichier de clés certifiées : Mot de passe du fichier de clés certifiées : Libellé de la clé :

Tableau 81. Validation du certificat serveur pour votre fournisseur de services partenaire dans une fédération SAML 2.0

Authentification SSL serveur pour la résolution d'artefact	Description	Votre valeur
Sélectionner le certificat de validation du serveur	Clé publique du certificat affichée durant les communications SSL avec votre partenaire. Vous et votre partenaire devez convenir du certificat à utiliser. Vous devez avoir déjà obtenu le certificat et l'avoir ajouté dans le fichier de clés certifiées. Pour plus détails, voir «Réception certificat serveur de votre partenaire», à la page 83.	Nom du fichier de clés certifiées : Mot de passe du fichier de clés certifiées : Nom du certificat :

Tableau 82. Authentification client pour votre fournisseur de services partenaire dans une fédération SAML 2.0

Authentification SSL client pour la résolution d'artefact	Description	Votre valeur
Informations d'authentification client L'une des options suivantes : • **Authentification de base** – Username – Mot de passe • **Authentification par certificat client** – Certificat à présenter au serveur du fournisseur d'identité. Ce certificat est le certificat que vous et votre fournisseur d'identité avez convenu de présenter. – Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée – Mot de passe du fichier de clés	Si votre partenaire exige une authentification mutuelle, vous devez connaître le type à utiliser. S'il s'agit d'une authentification standard, vous avez besoin d'un nom d'utilisateur et d'un mot de passe. S'il s'agit d'une authentification par certificat client, vous avez besoin du certificat que vous et votre partenaire avez convenu d'utiliser. **Remarque :** Si vous avez besoin d'un certificat, assurez-vous que vous avez convenu avec votre partenaire de son emplacement d'origine. Récupérez-le puis importez-le dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager avant d'effectuer cette tâche. Pour plus détails, voir «Obtention de votre certificat client», à la page 84.	L'une des options suivantes : • Informations d'authentification de base : – Nom d'utilisateur : – Mot de passe : • Informations relatives à l'authentification par certificat client – Nom de fichiers de clés : – Mot de passe du fichier de clés : – Alias de clé :

Tableau 83. Paramètres de votre fournisseur de services partenaire dans une fédération SAML 2.0

Paramètres des partenaires	Description	Votre valeur
Délai d'expiration de session (secondes)	Durée en secondes pendant laquelle une session reste valide, jusqu'à ce qu'il n'y ait plus aucune activité. Valeur par défaut : 3600 secondes.	Délai d'expiration de session :
Durée de vie de la requête de fermeture de session (secondes)	Spécifie la durée maximum, en secondes, pendant laquelle la requête de déconnexion reste valide. La valeur par défaut est 120 secondes.	Délai de déconnexion :

Tableau 83. Paramètres de votre fournisseur de services partenaire dans une fédération SAML 2.0 (suite)

Paramètres des partenaires	Description	Votre valeur
Algorithme de signature : L'une des options suivantes : • DSA-SHA1 OU • RSA-SHA1 • RSA-SHA256	**Remarque :** Les options **Algorithme de signature** sont disponibles uniquement dans Tivoli Federated Identity Manager, version 6.2.2 ou ultérieure. Type d'algorithme de signature utilisé pour générer les signatures XML pour votre partenaire. La liste des options affiche tous les algorithmes possibles en fonction du type de clé que vous avez choisi pour votre fédération. Seuls les algorithmes pris en charge par l'environnement d'exécution sont affichés. Si vous utilisez une clé DSA, DSA-SHA1 est sélectionné par défaut. Si vous utilisez une clé RSA, RSA-SHA1 est sélectionné par défaut. **Remarque :** Le système affiche l'option RSA-SHA256 en fonction de la version du groupe de correctifs de WebSphere Application Server qui s'exécute sur votre système.	Choisissez l'une des options suivantes : • DSA-SHA1 • RSA-SHA1 • RSA-SHA256

Tableau 83. Paramètres de votre fournisseur de services partenaire dans une fédération SAML 2.0 (suite)

Paramètres des partenaires	Description	Votre valeur
Paramètres du service d'alias • Inclure l'ID de fédération lors de l'exécution des opération de service d'alias	**Remarque :** L'option **Paramètres du service d'alias** est disponible uniquement dans Tivoli Federated Identity Manager, version 6.2.2 ou ultérieure. Indique si la clé pour l'indexation dans le service d'alias combine l'ID de fédération avec l'ID de fournisseur partenaire lors de l'exécution des opérations de service d'alias. Cette fonction est utile dans les scénarios où plusieurs fédérations qui utilisent des identificateurs de nom persistants importent les même métadonnées de partenaire. **Remarque :** Dans les versions précédentes de Tivoli Federated Identity Manager (antérieures à la version 6.2.2), les alias étaient stockés en fonction d'une clé utilisant uniquement l'ID du fournisseur partenaire. Utilisez la procédure de migration pour transférer les alias d'un format à l'autre par partenaire. Pour plus d'informations sur la procédure de migration, voir la section **Migrating SAML 2.0 alias service entries** du manuel *IBM Tivoli Federated Identity Manager - Guide d'installation*.	Cochez ou décochez la case.

Tableau 84. Paramètres d'assertion SAML pour votre fournisseur de services partenaire dans une fédération SAML 2.0

Paramètres d'assertion SAML	Description	Votre valeur
Inclure les types d'attributs suivants	Cochez la case pour indiquer les types d'attributs à inclure dans l'assertion. L'astérisque (*) est le paramètre par défaut. Il indique que tous les types d'attribut spécifiés dans le fichier de mappage d'identité ou par le module de mappage personnalisé sont inclus dans l'assertion. Pour spécifier un ou plusieurs types d'attributs individuellement, tapez chaque type d'attribut dans la case. Par exemple, si vous souhaitez inclure uniquement des attributs conformes au type suivant, entrez cette valeur dans la case : `urn:oasis:names:tc:SAML:2.0:assertion` Utilisez && pour séparer plusieurs types d'attributs.	
Options de chiffrement : • Chiffrer les identificateurs de nom • Chiffrer les assertions • Chiffrer tous les attributs de vérification	Ces cases à cocher indiquent les parties de l'assertion à chiffrer. Si vous n'effectuez aucune sélection et laissez les cases vides, aucune portion des assertions n'est chiffrée dans vos messages.	Laissez les cases vides ou sélectionnez l'une ou plusieurs des options suivantes : • Chiffrer les identificateurs de nom • Chiffrer les assertions • Chiffrer tous les attributs de vérification
Algorithme de chiffrement : • AES-128 • AES-256 • AES-192 • Triple DES	Type d'algorithme de chiffrement appliqué aux données de chiffrement destinées à votre partenaire. Si vous ne sélectionnez pas d'algorithme, Triple DES est utilisé.	Choisissez l'une des options suivantes si vous avez choisi une option de chiffrement : • AES-128 • AES-256 • AES-192 • Triple DES

Tableau 85. Informations de mappage requête d'attribut pour votre partenaire de fournisseur de service

Mappage de requête d'attribut	Description	Votre valeur
Options de mappage de requête d'attribut L'une des options suivantes : • Fichier de transformation XSL ou JavaScript contenant les règles de mappage • Module de mappage Tivoli Directory Integrator • Module de mappage personnalisé	Type de mappage de requête d'attribut utilisé. Vous devez sélectionner un fichier XSLT, un module de mappage Tivoli Directory Integrator ou un module de mappage personnalisé. Si vous utilisez un fichier XSLT, vous créez ce dernier avant de configurer la fédération. Le module de mappage Tivoli Directory Integrator est un module STS. Le mappage personnalisé est une option avancée. Si vous utilisez cette option, vous devez créer et ajouter un nouveau type et instance de module *avant* de pouvoir l'utiliser dans votre configuration.	L'une des valeurs suivantes : • Chemin de fichier XSLT • Module de mappage Tivoli Directory Integrator • Nom de l'instance de module de mappage personnalisée

Tableau 86. Options de mappage d'identité pour votre fournisseur de services partenaire dans une fédération SAML 2.0

Options de mappage d'identité	Description	Votre valeur
Options de mappage d'identité Vous pouvez choisir une des options suivantes : • Fichier de transformation XSL contenant les règles de mappage • Module de mappage personnalisé • Ne sélectionnez aucune option si vous souhaitez appliquer l'option de mappage d'identité actuellement définie.	Type de mappage d'identité à utiliser avec ce partenaire. Vous pouvez ne sélectionner aucune de ces options si vous souhaitez que ce partenaire choisisse l'option de mappage d'identité déjà configurée pour la fédération. Sinon, vous pouvez choisir une option de mappage spécifique et l'appliquer à ce partenaire. Pour choisir une option de mappage, vous devez savoir si vous devez utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous envisagez d'utiliser cette option, créez et ajoutez à l'environnement votre module de mappage en tant que type et instance de module. Vous devez effectuer cette étape *avant* de pouvoir l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	Ne sélectionnez aucune option pour utiliser la configuration de mappage existante. L'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété ce formulaire, poursuivez avec les étapes de la rubrique «Ajout à votre partenaire», à la page 271.

Formulaire de fournisseur d'identité partenaire SAML 2.0

Si vous utilisez SAML 2.0 dans votre rôle en tant que fournisseur de services, vous devez ajouter un fournisseur d'identité partenaire à votre fédération.

Utilisez le formulaire suivant pour rassembler les informations nécessaires en provenance de votre partenaire. Modifiez ce formulaire afin qu'il reflète les informations spécifiques que votre partenaire doit vous fournir et demandez à votre partenaire de compléter ce formulaire modifié.

Tableau 87. Fédération à laquelle vous ajoutez un fournisseur d'identité partenaire dans une fédération SAML 2.0

Sélection de fédération	Description	Votre valeur
Nom de la fédération	Nom de la fédération à laquelle vous ajoutez le partenaire.	

Tableau 88. Fichier de métadonnées délivré par votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Importer des métadonnées	Description	Votre valeur
Fichier de métadonnées	Nom et chemin du fichier que vous avez obtenu auprès de votre partenaire qui dispose de ses informations de configuration.	

Tableau 89. Validation de signature de votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Validation des signatures	Description	Votre valeur
Sélectionner les assertions et les messages SAML entrants qui nécessitent une signature	Ces options spécifient les messages entrants signés par votre partenaire. Lorsque le paramètre par défaut est sélectionné, l'ensemble des messages et assertions SAML entrants classiques (à l'exception des objets ArtifactResponse et AuthnResponse) doivent être signés.	L'une des options suivantes : • Un ensemble type de messages SAML entrants et d'assertions est signé. • Toutes les assertions et tous les messages SAML entrants sont signés. • Aucune assertion ni aucun message SAML entrant ne sont signés.
Fichier de clés	Fichier de clés certifiées dans lequel vous stockez la clé de votre partenaire pour valider sa signature lors de la signature des messages et des assertions. Le fichier de clés doit avoir été déjà créé pour cette clé. Pour plus détails, voir «Préparation des fichiers de clés», à la page 51.	Nom du fichier de clés certifiées : Mot de passe du fichier de clés certifiées : Libellé de la clé :

Tableau 90. Fichier de clés destiné au stockage de la clé de chiffrement délivrée par votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Chiffrement	Description	Votre valeur
Fichier de clés	Fichier de clés certifiées dans lequel vous stockez la clé permettant de chiffrer les messages envoyés à votre partenaire. Le système affiche cette option, car votre partenaire a fourni dans ses métadonnées une clé publique à utiliser pour le chiffrement. Vous devez avoir déjà obtenu le certificat et l'avoir importé dans le fichier de clés de celui-ci. Pour plus détails, voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.	Nom du fichier de clés certifiées : Mot de passe du fichier de clés certifiées : Libellé de la clé :

Tableau 91. Validation du certificat serveur pour votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Authentification SSL serveur pour la résolution d'artefact	Description	Votre valeur
Sélectionner le certificat de validation du serveur	Clé publique pour le certificat qui s'affiche durant la communication SSL avec votre partenaire. Vous et votre partenaire devez convenir du certificat à utiliser. Vous devez avoir déjà obtenu le certificat et l'avoir ajouté dans le fichier de clés certifiées. Pour plus détails, voir «Réception certificat serveur de votre partenaire», à la page 83.	Nom du fichier de clés certifiées : Mot de passe du fichier de clés certifiées : Nom du certificat :

Tableau 92. Authentification client pour votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Authentification SSL client pour la résolution d'artefact	Description	Votre valeur
Informations d'authentification client L'une des options suivantes : • **Authentification de base** – Username – Mot de passe • **Authentification par certificat client** – Certificat à présenter au serveur du fournisseur d'identité. Ce certificat est celui que vous et votre fournisseur d'identité partenaire avez convenu de présenter. – Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée – Mot de passe du fichier de clés	Si votre partenaire exige une authentification mutuelle, vous savez connaître le type à utiliser. S'il s'agit d'une authentification standard, vous avez besoin d'un nom d'utilisateur et d'un mot de passe. S'il s'agit d'une authentification par certificat client, vous avez besoin du certificat que vous et votre partenaire avez convenu d'utiliser. **Remarque :** Si vous avez besoin d'un certificat, assurez-vous que vous avez convenu avec votre partenaire de son emplacement d'origine. Récupérez-le et importez-le ensuite dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager avant d'effectuer cette tâche. Pour plus détails, voir «Obtention de votre certificat client», à la page 84.	L'une des options suivantes : • Informations d'authentification de base : – Nom d'utilisateur : – Mot de passe : • Informations relatives à l'authentification par certificat client – Nom de fichiers de clés : – Mot de passe du fichier de clés : – Alias de clé :

Tableau 93. Paramètres de votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Paramètres des partenaires	Description	Votre valeur
URL cible de post-authentification par défaut	Emplacement vers lequel est redirigé l'utilisateur lorsque le fournisseur de services ne fournit pas d'URL cible lors de la demande initiale. Cette URL doit être valide, mais ne doit pas nécessairement être active.	
Imposer l'authentification pour obtenir l'association de compte	Indique si un utilisateur est obligé de s'authentifier auprès du fournisseur de services pour effectuer l'association de compte. Cet événement se produit si une réponse SAML est reçue avec un alias inconnu dans le fournisseur de services.	

Tableau 93. Paramètres de votre fournisseur d'identité partenaire dans une fédération SAML 2.0 (suite)

Paramètres des partenaires	Description	Votre valeur
Algorithme de signature : L'une des options suivantes : • DSA-SHA1 OU • RSA-SHA1 • RSA-SHA256	**Remarque :** Les options **Algorithme de signature** sont disponibles uniquement dans Tivoli Federated Identity Manager, version 6.2.2 ou ultérieure. Type d'algorithme de signature utilisé pour générer les signatures XML pour votre partenaire. La liste des options affiche tous les algorithmes possibles en fonction du type de clé que vous avez choisi pour votre fédération. Seuls les algorithmes pris en charge par l'environnement d'exécution sont affichés. Si vous utilisez une clé DSA, DSA-SHA1 est sélectionné par défaut. Si vous utilisez une clé RSA, RSA-SHA1 est sélectionné par défaut. **Remarque :** Le système affiche l'option RSA-SHA256 en fonction de la version du groupe de correctifs WebSphere Application Server qui s'exécute sur votre système.	Choisissez l'une des options suivantes : • DSA-SHA1 • RSA-SHA1 • RSA-SHA256

Tableau 93. Paramètres de votre fournisseur d'identité partenaire dans une fédération SAML 2.0 (suite)

Paramètres des partenaires	Description	Votre valeur
Paramètres du service d'alias • Inclure l'ID de fédération lors de l'exécution des opération de service d'alias	**Remarque :** L'option **Paramètres du service d'alias** est disponible uniquement dans Tivoli Federated Identity Manager, version 6.2.2 ou ultérieure. Indique si la clé pour l'indexation dans le service d'alias combine l'ID de fédération avec l'ID de fournisseur partenaire lors de l'exécution des opérations de service d'alias. Cette fonction est utile dans les scénarios où plusieurs fédérations qui utilisent des identificateurs de nom persistants importent les même métadonnées de partenaire. **Remarque :** Dans les versions de Tivoli Federated Identity Manager antérieurs à la version 6.2.2, les alias étaient stockés en fonction d'une clé utilisant uniquement l'ID fournisseur partenaire. Utilisez la procédure de migration pour transférer les alias d'un format à l'autre par partenaire. Pour plus d'informations sur la procédure de migration, voir la rubrique "Migration des entrées de service d'alias SAML 2.0" dans le manuel *IBM Tivoli Federated Identity Manager - Guide d'installation*.	Cochez ou décochez la case.

Tableau 94. Paramètres d'assertion SAML pour votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Paramètres d'assertion SAML	Description	Votre valeur
Nom d'utilisateur à employer pour les utilisateurs anonymes	Utilisez cet identificateur de nom pour accéder à un service via une identité anonyme. Le nom d'utilisateur que vous indiquez ici sera reconnu par le fournisseur de services comme identificateur de nom à utilisation unique d'un registre d'utilisateurs local. Cette fonction permet aux utilisateurs d'accéder à une ressource du fournisseur de services sans établir une identité fédérée. Elle est utile dans les cas où le fournisseur n'a pas besoin de connaître l'identité du compte utilisateur mais uniquement de savoir que le fournisseur d'identité a authentifié l'utilisateur (et s'en porte garant).	
Mapper les identificateurs de nom inconnu au nom d'utilisateur anonyme.	Indique que le fournisseur de services peut mapper un alias d'identificateur de nom inconnu persistant à un compte d'utilisateur anonyme. Cette option est désactivée par défaut.	

Tableau 94. Paramètres d'assertion SAML pour votre fournisseur d'identité partenaire dans une fédération SAML 2.0 (suite)

Paramètres d'assertion SAML	Description	Votre valeur
Créer plusieurs instructions d'attribut dans l'utilisateur universel	Sélectionnez cette option pour conserver plusieurs instructions d'attribut dans les groupes dans lesquels elles ont été reçues. Cette option peut se révéler nécessaire si vos règles de mappage d'identité personnalisées sont écrites de manière à s'appliquer à un ou plusieurs groupes d'instructions d'attribut spécifiques. Si cette case n'est pas activée, plusieurs instructions d'attribut sont organisées dans un seul groupe (AttributeList) dans le document STSUniversalUser, ainsi que dans l'assertion. Par défaut, cette option est désélectionnée, ce qui convient à la plupart des configurations.	
Options de chiffrement : • Chiffrer les identificateurs de nom	Cette case à cocher indique si vous souhaitez chiffrer les identificateurs de nom dans les assertions ou non.	Cochez ou décochez la case.

Tableau 95. Informations de requête d'attribut pour le partenaire de fournisseur d'identité

Requête d'attribut	Description	Votre valeur
Inclure les types d'attributs suivants	Cochez la case pour indiquer les types d'attributs à inclure dans l'assertion. L'astérisque (*) est le paramètre par défaut. Il indique que tous les types d'attribut spécifiés dans le fichier de mappage d'identité ou par le module de mappage personnalisé sont inclus dans l'assertion. Pour spécifier un ou plusieurs types d'attributs individuellement, tapez chaque type d'attribut dans la case. Par exemple, si vous souhaitez inclure uniquement des attributs conformes au type suivant, entrez cette valeur dans la case : `urn:oasis:names:tc:SAML:2.0:assertion` Utilisez && pour séparer plusieurs types d'attributs.	
Options de chiffrement : • Chiffrer les identificateurs de nom • Chiffrer les assertions • Chiffrer tous les attributs de vérification	Ces cases à cocher indiquent les parties de l'assertion à chiffrer. Si vous n'effectuez aucune sélection et laissez les cases vides, aucune portion des assertions n'est chiffrée dans vos messages. L'option de **chiffrage de tous les attributs d'assertion** indique que tous les attributs de l'assertion sont chiffrés. Lorsque cette option n'est pas sélectionnée (définie sur false), vous pouvez gérer le chiffrement d'attributs spécifiques par le biais d'une règle de mappage de jeton SAML XSLT.	Laissez en blanc ou choisissez l'une des options suivantes : • Chiffrer les identificateurs de nom • Chiffrer les assertions • Chiffrer tous les attributs de vérification

Tableau 95. Informations de requête d'attribut pour le partenaire de fournisseur d'identité (suite)

Requête d'attribut	Description	Votre valeur
Algorithme de chiffrement : • AES-128 • AES-256 • AES-192 • Triple DES	Type d'algorithme de chiffrement appliqué aux données de chiffrement destinées à votre partenaire. Si vous ne sélectionnez pas d'algorithme, Triple DES est utilisé.	Choisissez l'un des chiffrements suivants si vous avez choisi une option de chiffrement : • AES-128 • AES-256 • AES-192 • Triple DES

Tableau 96. Informations de mappage requête d'attribut pour le partenaire de fournisseur d'identité

Mappage de requête d'attribut	Description	Votre valeur
Options de mappage de requête d'attribut L'une des options suivantes : • Fichier de transformation XSL ou JavaScript contenant les règles de mappage • Module de mappage Tivoli Directory Integrator • Module de mappage personnalisé	Type de mappage de requête d'attribut utilisé. Vous devez sélectionner un fichier XSLT, un module de mappage Tivoli Directory Integrator ou un module de mappage personnalisé. Si vous utilisez un fichier XSLT, vous créez ce dernier avant de configurer la fédération. Le module de mappage Tivoli Directory Integrator est un module STS. Le mappage personnalisé est une option avancée. Si vous utilisez cette option, vous devez créer et ajouter un nouveau type et instance de module *avant* de pouvoir l'utiliser dans votre configuration.	L'une des valeurs suivantes : • Fichier XSLT (chemin et nom) • Module de mappage Tivoli Directory Integrator • Nom de l'instance de module de mappage personnalisée

Tableau 97. Options de mappage d'identité pour votre fournisseur d'identité partenaire dans une fédération SAML 2.0

Options de mappage d'identité	Description	Votre valeur
Options de mappage d'identité L'une des options suivantes : • Fichier de transformation XSL contenant les règles de mappage • Module de mappage personnalisé • Ne sélectionnez aucune option si vous souhaitez appliquer l'option de mappage d'identité actuellement définie.	Type de mappage d'identité à utiliser avec ce partenaire. Vous pouvez ne sélectionner aucune de ces options si vous souhaitez que ce partenaire choisisse l'option de mappage d'identité déjà configurée pour la fédération. Sinon, vous pouvez choisir une option de mappage spécifique et l'appliquer à ce partenaire. Pour choisir une option de mappage, vous devez savoir si vous devez utiliser un fichier XSLT pour le mappage d'identité ou un module de mappage personnalisé. Le mappage personnalisé est une option avancée. Si vous envisagez d'utiliser l'option de mappage personnalisé, créez et ajoutez le module de mappage en tant que type et instance de module. Vous pouvez ensuite l'utiliser dans votre configuration. Si vous choisissez d'utiliser un fichier XSLT, vous devez préparer le fichier pour la fédération.	Ne sélectionnez aucune option pour utiliser la configuration de mappage existante. L'une des valeurs suivantes : • Fichier XSLT (chemin et nom): • Nom de l'instance de module de mappage personnalisée :

Une fois que vous avez complété ce formulaire, poursuivez avec les étapes de la rubrique «Ajout à votre partenaire».

Ajout à votre partenaire

Après avoir configuré votre rôle dans la fédération et rassemblé les informations relatives à votre partenaire, vous devez procéder à l'ajout de ce dernier.

Avant de commencer

avant de début cette procédure, complétez le formulaire d'informations relatives au partenaire approprié.

- Si vous êtes le fournisseur de services, ajoutez un partenaire fournisseur d'identité. Utilisez le formulaire destiné au partenaire fournisseur de services en fonction de la norme SAML que vous utilisez dans votre fédération :
  - «Formulaire pour fournisseur de services partenaire SAML 1.x», à la page 240

– «Formulaire de fournisseur de services partenaire SAML 2.0», à la page 253

- Si vous êtes le fournisseur de services, ajoutez un partenaire fournisseur d'identité. Utilisez le formulaire destiné au partenaire fournisseur d'identité en fonction de la norme SAML que vous utilisez dans votre fédération :
  – «Formulaire pour fournisseur d'identité partenaire SAML 1.x», à la page 246
  – «Formulaire de fournisseur d'identité partenaire SAML 2.0», à la page 261

Pourquoi et quand exécuter cette tâche

Après avoir complété le fournisseur relatif au partenaire approprié, utilisez l'assistant Partenaire de la console pour ajouter la partenaire. Pour obtenir la description des zones que l'assistant vous invite à renseigner, consultez le formulaire et l'aide en ligne.

Remarque : Pendant la configuration, il se peut que vous soyez invité à redémarrer WebSphere Application Server. Avant de poursuivre la tâche, assurez-vous que le serveur a redémarré entièrement.

Procédure

1. Assurez-vous d'avoir rassemblé les informations relatives à votre partenaire, telles que décrites dans les formulaires.

   Si, par exemple, vous utilisez un fichier de métadonnées fourni par votre partenaire, copiez-le à un emplacement facilement accessible sur votre ordinateur.
2. Connectez-vous à la console.
3. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Le panneau de fédération s'ouvre.
4. Sélectionnez la fédération à laquelle vous ajouterez le partenaire.
5. Cliquez sur **Ajouter un partenaire**. Selon la norme SAML que vous utilisez dans la fédération, l'un des panneaux suivants s'ouvre :

   Options de métadonnées
   Ce panneau s'ouvre si vous ajoutez un partenaire à une fédération SAML 1.x. Dans ce panneau, cliquez sur l'une des options suivantes :
   - **Importer les métadonnées**
   - **Spécifier manuellement les données SAML**

   Importer les métadonnées
   Ce panneau s'ouvre si vous ajoutez un partenaire à une fédération SAML 2.0.
6. Utilisez votre formulaire complété comme guide pour renseigner les zones qui s'affichent dans chaque panneau.

   Si vous avez besoin de revenir à un panneau précédent, cliquez sur **Précédent**. Si vous souhaitez mettre fin à la configuration, cliquez sur **Annuler**. Sinon, cliquez sur **Suivant** après avoir complété chaque panneau.
7. Vérifiez que les paramètres sont corrects.
8. Cliquez sur **Terminer**. L'écran Ajout de partenaire terminé s'ouvre. Le partenaire a été ajouté à la fédération, mais il est désactivé par défaut par mesure de sécurité. Vous devez activer le partenaire.
9. Cliquez sur **Activer le partenaire** pour activer ce partenaire.

Que faire ensuite

Si vous n'avez pas encire fourni vos informations de configuration à votre partenaire, vous pouvez le faire maintenant en suivant les instructions de la rubrique «Transmission des propriétés de la fédération au partenaire».

Transmission des propriétés de la fédération au partenaire

Si votre partenaire souhaite vous ajouter en tant que partenaire dans la configuration de sa fédération, vous devez lui fournir les informations nécessaires.

La procédure à suivre varie selon que vous pouvez fournir un fichier de métadonnées ou que vous devez saisir manuellement les informations.

- **Méthode utilisant un fichier de métadonnées**

  Si votre partenaire dispose d'un moyen pour importer vos données, vous pouvez appliquer la méthode du fichier de métadonnées, quel que soit le type de fédération SAML (1.x ou 2.0) configuré.

  1. A l'aide de la console, générez un fichier de métadonnées contenant les informations nécessaires pour la configuration de la fédération, ainsi qu'une clé permettant de valider les signature de message de réponse si vous exigez la validation des signatures. Pour plus d'informations, voir «Exportation des propriétés d'une fédération».
  2. Il se peut que vous deviez également fournir à votre partenaire les clés et certificats appropriés, suivant votre rôle et la norme SAML utilisés dans la fédération. Voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.

- **Méthode manuelle**

  Si vous avez configuré une fédération SAML 1.x, vous pouvez collecter manuellement la configuration nécessaire au lieu d'exporter les propriétés dans un fichier.

  Remarque : L'utilisation d'un fichier de métadonnées permet d'éliminer les risques d'erreurs susceptibles de se produire lors de l'entrée manuelle des données.

  Pour collecter manuellement des informations, procédez comme suit.

  1. Utilisez le panneau Propriétés de la fédération dans la console pour obtenir les propriétés. Pour afficher le panneau Propriétés de la fédération, voir «Affichage des propriétés d'une fédération», à la page 274.

     Utilisez le contenu du panneau Propriétés de la fédération pour accéder aux propriétés de votre fédération.
  2. Il se peut que vous deviez également fournir à votre partenaire les clés et certificats de votre fédération. Voir Chapitre 8, «Configuration de la sécurité des messages», à la page 51.

Exportation des propriétés d'une fédération

Lorsque vous souhaitez rejoindre la fédération d'un partenaire, vous devez fournir les propriétés de configuration de la fédération. Vous pouvez exporter vos propriétés de fédération vers un fichier et les partager avec votre partenaire.

Procédure

1. Connectez-vous à la console.

2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Le panneau Fédérations s'affiche.
3. Sélectionnez une fédération dans le tableau.
4. Cliquez sur **Exporter**. Le navigateur affiche un message vous invitant à sauvegarder le fichier contenant les données exportées.
5. Cliquez sur **OK**. La fenêtre de téléchargement du navigateur vous invite à entrer un emplacement de sauvegarde du fichier.
6. Sélectionnez un répertoire et un fichier de métadonnées. La syntaxe des fichiers de métadonnées se présente comme suit :

   *nomdelafédération_nomdel'entreprise*_métadonnées.xml

   Par exemple, pour une fédération dénommée fédération1 et une entreprise nommée ABC, le nom du fichier de métadonnées sera libellé comme suit :

   fédération1_ABC_métadonnées.xml

   Placez ce fichier dans un endroit facilement accessible. Fournissez ce fichier à votre partenaire lorsque ce dernier souhaite importer les informations de configuration de cette fédération.
7. Cliquez sur **Sauvegarder**.

Affichage des propriétés d'une fédération

Utilisez l'option Propriétés de la fédération pour afficher les caractéristiques d'une fédération existante ou pour modifier une fédération existante. Cette tâche peut être utile si vous devez collecter manuellement les propriétés de votre fédération afin de les partager avec votre partenaire.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Le panneau Fédérations affiche la liste des fédérations configurées.
3. Sélectionnez une fédération.
4. Cliquez sur **Propriétés** pour afficher les propriétés d'une fédération existante.
5. Sélectionnez les propriétés à modifier. Pour la description des propriétés d'une fédération, voir l'aide en ligne.
6. Une fois que vous avez terminé d'afficher ou de modifier les propriétés, cliquez sur **OK** pour fermer la fenêtre des propriétés de fédération.

Synchronisation des horloges système dans la fédération

Du fait que les jetons de sécurité possèdent des délais d'expiration, votre horloge système doit être synchronisée sur celle de votre partenaire.

Pourquoi et quand exécuter cette tâche

Vous devez vous assurer que, dans votre environnement, l'horloge du système sur lequel le composant d'exécution et les services de gestion de Tivoli Federated Identity Manager sont installés est synchronisée avec celle de votre partenaire.

Reportez-vous aux instructions contenues dans la documentation de votre système d'exploitation pour plus d'informations sur la synchronisation des horloges système. Envisagez d'utiliser le protocole de synchronisation d'horloge NTP (source d'horodatage null).

Chapitre 20. Configuration d'une fédération SAML à l'aide de l'interface de ligne de commande

Pour configurer une fédération SAML à l'aide de l'interface de ligne de commande, vous devez configurer la fédération du fournisseur d'identité, configurer la fédération du fournisseur de service et fournir les propriétés de configuration de fédération à votre partenaire.

Les rubriques suivantes détaillent les configurations des versions et des rôles SAML spécifiques utilisant l'interface de ligne de commande

- «Configuration d'une fédération de fournisseurs d'identités SAML 1.x à l'aide de l'interface de ligne de commande»
- «Configuration d'une fédération de fournisseurs de services SAML 1.x à l'aide de l'interface de ligne de commande», à la page 279
- «Importation d'un fournisseur de services SAML 1.x dans la fédération de fournisseurs d'identités SAML», à la page 281
- «Importation d'un fournisseur d'identités SAML 1.x dans la fédération de fournisseurs de services SAML», à la page 285
- «Configuration d'une fédération de fournisseurs d'identités SAML 2.0 à l'aide de l'interface de ligne de commande», à la page 288
- «Configuration d'une fédération de fournisseurs de services SAML 2.0 à l'aide de l'interface de ligne de commande», à la page 292
- «Importation d'un fournisseur de services SAML 2.0 dans la fédération de fournisseurs d'identités SAML», à la page 295
- «Importation d'un fournisseur d'identités SAML 2.0 dans la fédération de fournisseurs de services SAML», à la page 297

Configuration d'une fédération de fournisseurs d'identités SAML 1.x à l'aide de l'interface de ligne de commande

Utilisez les commandes de l'interface de ligne de commande pour configurer une fédération de fournisseurs d'identités SAML en créant un fichier de réponses et une fédération de fournisseurs d'identités.

Pourquoi et quand exécuter cette tâche

Cette tâche nécessite l'utilisation de la commande **manageItfimFederation**. La commande **manageItfimFederation** requiert des paramètres spécifiques pour pouvoir exécuter des opérations sur une fédération. Pour plus d'informations, voir *IBM Tivoli Federated Identity ManagerGuide d'administration*.

Procédure

1. Créez un fichier de réponses en exécutant la commande suivante dans la console WebSphere **wsadmin** :

```
wsadmin>$AdminTask manageItfimFederation { -operation createResponseFile
-fimDomainName fimipdomain -role ip -protocol SAML1_1 -fileId
/downloads/saml11_ip_properties.xml }
```

Remarque : Modifiez le type de protocole SAML selon la version SAML que vous souhaitez utiliser. Utilisez l'un des paramètres suivants pour le type de protocole :

- SAML1_1
- SAML1_0

2. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 98. Paramètres du fichier de réponses pour le fournisseur d'identités dans la fédération SAML 1.x

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Nom de la fédération	Nom unique de la fédération (obligatoire)	Tout nom Par exemple, `saml11ip`	**FedName**
Nom de la société	Nom de la société associée à la fédération (obligatoire)	Tout nom Par exemple, `IDP Company Name`	**CompanyName**
Adresse URL de la société	Adresse URL du site Web de la société associée à la fédération (obligatoire)	Adresse URL du site Web de votre société	**CompanyUrl**
ID fournisseur	ID du fournisseur du protocole SAML utilisé par la fédération Identificateur unique permettant au fournisseur de se faire reconnaître par le fournisseur de services. (obligatoire)	Cette valeur est constituée du protocole et du nom d'hôte de l'adresse URL du fournisseur d'identités (facultatif). Elle peut inclure un numéro de port. Par exemple, pour une fédération nommée `saml_fed: https://idp.example.com/FIM/sps/saml_fed/saml`, définissez toutes les propriétés de la colonne suivante sur la même valeur.	**ProviderID** **SAML11AssertionIssuerName**
URL du serveur point de contact	Adresse URL donnant accès aux noeuds finals sur le serveur point de contact. (obligatoire)	Adresse URL Par exemple, `https://www.idpexample.com/FIM/sps`	**BaseUrl**
Les messages SAML pour le profil POST du navigateur sont signés.	Lorsque le POST du navigateur est utilisé en tant que profil, les messages SAML doivent être signés. Cette option est donc présélectionnée et ne peut pas être désélectionnée. (obligatoire)	Signer les messages d'artefact du navigateur (A définir sur True.)	**SignArtifactResponse**

Tableau 98. Paramètres du fichier de réponses pour le fournisseur d'identités dans la fédération SAML 1.x (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande

Sélectionner la clé de signature
 	
Fichier de clés du service de clés Tivoli Federated Identity Manager dans lequel la clé est stockée.
	Entrer une clé de signature car les messages POST du navigateur doivent être signés.
	
	Si vous choisissez également de signer les messages lorsque vous utilisez l'artefact du navigateur, cette même clé est utilisée pour les signer. (obligatoire)
	
	Remarque : Avant d'exécuter cette tâche, créez la clé et importez-la dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.
		Nom de fichiers de clés :
		
		nom d'alias de clé :
		
		Ces données sont fournies sous la forme
		`"Keystore Name" _"Alias Name"`
		
		Par exemple :
		`DefaultKeyStore_ clé d'essai`
			`SigningKeyId`

URL de service de connexion unique	Adresse URL de votre noeud final de connexion unique. Ce paramètre est également appelé adresse URL du service de transfert intersite ou adresse URL à laquelle le fournisseur de services envoie les demandes 'authentification. (obligatoire)	Indiquez l'adresse URL du service de résolution d'assertion. Par exemple, pour une fédération intitulée `saml_fed:` `https://idp.example.com/ FIM/sps/saml_fed/saml11/ login`	**`SignonEndpoint`**
URL du service de résolution des artefacts	Adresse URL de votre noeud final de résolution d'artefacts. (obligatoire)	Indiquez l'adresse URL du service de résolution d'assertion. Par exemple, pour une fédération intitulée `saml_fed:` `https://idp.example.com/ FIM/sps/saml_fed/saml11/ soap`	**`ArtifactResolutionServiceEndpoint`**
Durée de mise en cache d'artefact (en secondes)	Durée de mise en cache d'artefact en secondes. Valeur par défaut : 30 secondes.	Utilisez la valeur par défaut.	**`ArtifactCacheLifetime`**

Tableau 98. Paramètres du fichier de réponses pour le fournisseur d'identités dans la fédération SAML 1.x (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Autoriser l'extension IBM Protocol	Indiquez si l'extension de protocole IBM doit être utilisée. Cette extension contient un paramètre de chaîne de requête qui indique si l'artefact du navigateur ou le POST du navigateur doit être utilisé.(obligatoire)	Ne pas autoriser l'extension Protocol A définir sur False.	**AllowIBMProtocolExtension**
Durée de validité d'une assertion avant sa date d'émission	Durée en secondes pendant laquelle une assertion est considérée valide avant sa date de création. Valeur par défaut : 60	Utilisez la valeur par défaut.	**SAML11AssertionValidBefore**
Durée de validité de l'assertion après émission	Durée en secondes pendant laquelle une assertion est considérée valide après sa date de création. Valeur par défaut : 60	Utilisez la valeur par défaut.	**SAML11AssertionValidAfter**
Options de mappage d'identité • Fichier de transformation XSL (XSLT) contenant les règles de mappage	Type de mappage d'identité à utiliser. Utilisez un fichier XSLT pour le mappage d'identité et préparez le fichier pour la fédération.(obligatoire)	Fichier XSLT correspondant au rôle IP pour les fédérations SAML 1.1 : `/opt/IBM/FIM/examples/mapping_rules/ip_saml_1x.xsl`	**MappingRuleFileName**

3. Entrez la commande suivante dans une invite de commande pour créer la fédération de fournisseur d'identités :

   ```
   wsadmin>$AdminTask manageItfimFederation { -operation create
   -fimDomainName fimipdomain -fileId
   /downloads/saml11_ip_properties.xml }
   ```

   Le message de confirmation suivant s'affiche :

   ```
   FBTADM001I Command completed successfully
   ```

Que faire ensuite

Poursuivez avec Configuration d'une fédération de fournisseurs de services SAML 1.x à l'aide de l'interface de ligne de commande.

Configuration d'une fédération de fournisseurs de services SAML 1.x à l'aide de l'interface de ligne de commande

Utilisez les commandes de l'interface de ligne de commande pour configurer une fédération de fournisseurs de services SAML 1.x en créant un fichier de réponses et une fédération de fournisseurs de services.

Pourquoi et quand exécuter cette tâche

Cette tâche nécessite l'utilisation de la commande **manageItfimFederation**. La commande **manageItfimFederation** requiert des paramètres spécifiques pour pouvoir exécuter des opérations sur une fédération. Pour plus d'informations, voir *IBM Tivoli Federated Identity ManagerGuide d'administration*.

Procédure

1. Créez un fichier de réponses en exécutant la commande suivante dans la console WebSphere **wsadmin** :

   ```
   wsadmin>$AdminTask manageItfimFederation { -operation createResponseFile
   -fimDomainName fimspdomain -protocol SAML1_1 -role sp -fileId
   /downloads/saml11_sp_properties.xml }
   ```

   Le message de confirmation suivant s'affiche :

   ```
   FBTADM001I Command completed successfully
   ```

   Remarque : Modifiez le type de protocole SAML selon la version SAML que vous souhaitez utiliser. Utilisez l'un des paramètres suivants pour le type de protocole :
   - SAML1_1
   - SAML1_0
2. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 99. Paramètres du fichier de réponses pour le fournisseur de services dans la fédération SAML

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Nom de la fédération	Nom unique de la fédération (obligatoire)	Tout nom Par exemple, `saml11ip`	**FedName**
Nom de la société	Nom de la société associée à la fédération (obligatoire)	Tout nom Par exemple, `SP Company Name`	**CompanyName**
Adresse URL de la société	Adresse URL du site Web de la société associée à la fédération (obligatoire)	Adresse URL du site Web de votre société	**CompanyUrl**
Protocole	Protocole SAML que vous et votre partenaire utilisez dans la fédération. (obligatoire)	L'une des valeurs suivantes : • `SAML1_1` • `SAML1_0`	**Protocole**

Tableau 99. Paramètres du fichier de réponses pour le fournisseur de services dans la fédération SAML (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
ProviderId	ID du fournisseur du protocole SAML utilisé par la fédération Identificateur unique permettant au fournisseur de se faire reconnaître par le fournisseur de services. (obligatoire)	Cette valeur est constituée du protocole et du nom d'hôte de l'URL du fournisseur de services. (facultatif) Elle peut inclure un numéro de port. Par exemple, pour une fédération intitulée `saml_fed: https://sp.example.com/FIM/sps/saml_fed/saml`, définissez toutes les propriétés de la colonne suivante sur la même valeur.	**`ProviderId`**
URL du serveur point de contact	Adresse URL donnant accès aux noeuds finals sur le serveur point de contact. (obligatoire)	Adresse URL Par exemple, `https:/sp.example.com/FIM/sps`	**`BaseUrl`**
Signer les requêtes de résolution d'artefact	La requête d'artefact SAML doit être signée.	Signer les messages de requête (A définir sur True.)	**`SignArtifactRequest`**
Sélectionner la clé de signature Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée.	Si vous avez également sélectionné la signature de la requête d'artefact, entrez une clé de signature. (obligatoire) **Remarque :** Avant d'exécuter cette tâche, créez la clé et importez-la dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.	Nom de fichiers de clés : nom d'alias de clé : Ces données sont fournies sous la forme `"Keystore Name"_"Alias Name"`. Par exemple, `DefaultKeyStore_ testkey`	**`SigningKeyId`**
URL de service de connexion unique	Adresse URL de votre noeud final de connexion unique. Ce paramètre est également appelé adresse URL du service de transfert intersite ou adresse URL à laquelle le fournisseur de services envoie les demandes d'authentification. (obligatoire)	Indiquez l'adresse URL du service de résolution d'assertion. Par exemple : `https://sp.example.com/FIM/sps/saml_fed/saml11/login`	**`SignonEndpoint`**

Tableau 99. Paramètres du fichier de réponses pour le fournisseur de services dans la fédération SAML (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Options de mappage d'identité • Fichier de transformation XSL (XSLT) contenant les règles de mappage	Type de mappage d'identité à utiliser. Utilisez un fichier XSLT pour le mappage d'identité et préparez le fichier pour la fédération. (obligatoire)	Fichier XSLT correspondant au rôle IP pour les fédérations SAML 1.1 : `/opt/IBM/FIM/examples/mapping_rules/sp_saml_1x.xsl`	**`MappingRuleFileName`**

3. Entrez la commande suivante dans une invite de commande pour créer la fédération de fournisseurs de services :

```
wsadmin>$AdminTask manageItfimFederation { -operation create
-fimDomainName fimspdomain -fileId
/downloads/saml11_sp_properties.xml }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

Que faire ensuite

Poursuivez avec Importation d'un fournisseur de services SAML 1.X dans la fédération de fournisseurs d'identités SAML.

Importation d'un fournisseur de services SAML 1.x dans la fédération de fournisseurs d'identités SAML

Pour ajouter un fournisseur de services à la fédération de fournisseurs d'identités, vous devez importer les propriétés de configuration du fournisseur de services.

Procédure

1. Entrez la commande suivante dans une invite de commande pour exporter les métadonnées du fournisseur de services et obtenir des informations sur l'environnement :

```
wsadmin>$AdminTask manageItfimFederation { -operation export
-fimDomainName fimspdomain -federationName saml11sp -fileId
/downloads/saml11_sp_metadata.xml }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

2. Créez un fichier de réponses de fournisseur de services en exécutant la commande suivante dans la console WebSphere **`wsadmin`** :

```
wsadmin>$AdminTask manageItfimPartner { -operation createResponseFile
-fimDomainName fimipdomain -federationName saml11ip -partnerRole sp -fileId
/downloads/saml11_sp_partner_properties.xml }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

3. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 100. Paramètres du fichier de réponses pour le partenaire du fournisseur de services dans la fédération SAML 1.x

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Importez le fichier de métadonnées	Pour pouvoir importer un fichier de métadonnées, vous devez connaître le nom du fichier et son emplacement. (obligatoire)	Nom complet spécifié du fichier de métadonnées. Par exemple : `/downloads/saml11_sp_metadata.xml`	**`metadataFileName`**
Valider les signatures sur les requêtes d'artefact	Validez les signatures de message SAML lorsque vous utilisez l'artefact du navigateur. (facultatif)	Valider les signatures pour l'artefact (A définir sur True.)	**`ValidateArtifactRequest`**

Tableau 100. Paramètres du fichier de réponses pour le partenaire du fournisseur de services dans la fédération SAML 1.x (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Identificateur de clé de validation	Si vous choisissez de valider les messages lorsque l'artefact du navigateur est utilisé, vous devez indiquer une clé pour cette validation. Il doit s'agir de la clé publique correspondant à la clé privée que votre partenaire utilise pour signer les messages. (obligatoire) **Remarque :** Si vous importez des données de partenaire, la clé est fournie dans le fichier de métadonnées. Avant d'importer les données, créez un fichier de clés puis spécifiez-le pour la clé. Avant d'entrer des données de partenaire manuellement, demandez la clé au partenaire et importez-la dans le fichier de clés approprié dans le service de clés Tivoli Federated Identity Manager.	**Méthode utilisant les métadonnées :** • Nom du fichier de clés certifiées : • Libellé de la clé :	**`ValidateKeyIdentifier`**
Signer les assertions SAML	Signez les assertions SAML. (facultatif)	Activer les signatures SAML (A définir sur True.)	**`com.tivoli.am.fim.sts.saml.1.1.assertion.sign`**

Tableau 100. Paramètres du fichier de réponses pour le partenaire du fournisseur de services dans la fédération SAML 1.x (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande

Sélectionner la clé de signature
• Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée
	Si vous choisissez de signer les assertions, vous devez sélectionner un fichier de clés et une clé. (obligatoire)
	
	Remarque : Créez le fichier de clés et la clé avant d'effectuer cette tâche.
		• Nom de fichiers de clés :
		• nom d'alias de clé :
		
		Ces données sont fournies sous la forme
		`"Keystore Name"_`
		`"Alias Name"`
		
		Par exemple :
		`DefaultKeyStore_`
		`clé d'essai`
		
		Définissez les deux propriétés sur la même valeur.
			SigningKeyId,
			SAML11SigningKeyIdentifier

Inclure les types d'attributs suivants	Cochez la case pour indiquer les types d'attributs à inclure dans l'assertion. Le paramètre par défaut (marqué d'un astérisque) indique que tous les types d'attributs spécifiés dans le fichier de mappage d'identité sont inclus à l'assertion. (obligatoire)	Utiliser la valeur par défaut (*)	**SAML11ExtendedAttributeTypes**
Le partenaire utilise le profil POST du navigateur pour la connexion unique	Valeur booléenne indiquant que le partenaire du fournisseur de services utilise le POST du navigateur. (obligatoire)	Le partenaire utilise le POST du navigateur (A définir sur True.)	**PartnerUsesBrowserPost**

4. Entrez la commande suivante dans une invite de commande pour ajouter le nouveau partenaire du fournisseur de services à la fédération de fournisseurs d'identités :

```
wsadmin>$AdminTask manageItfimPartner { -operation create
-fimDomainName fimipdomain
-federationName saml11ip -partnerName saml11sp -fileId
/downloads/saml11_sp_partner_properties.xml
-signingKeystorePwd testonly}
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

Que faire ensuite

Poursuivez avec Importation d'un fournisseur d'identités SAML 1.x dans la fédération de fournisseurs de services SAML.

Importation d'un fournisseur d'identités SAML 1.x dans la fédération de fournisseurs de services SAML

Pour ajouter un fournisseur d'identités à la fédération de fournisseurs de services, vous devez importer les propriétés de configuration du fournisseur d'identités.

Procédure

1. Entrez la commande suivante dans une invite de commande pour exporter les métadonnées du fournisseur d'identités et obtenir des informations sur l'environnement :

   ```
   wsadmin>$AdminTask manageItfimFederation { -operation export
   -fimDomainName fimipdomain -federationName saml11ip -fileId
   /downloads/saml11_ip_metadata.xml }
   ```

   Le message de confirmation suivant s'affiche :

   ```
   FBTADM001I Command completed successfully
   ```

2. Créez un fichier de réponses du fournisseur d'identités en exécutant la commande suivante dans la console WebSphere **wsadmin** :

   ```
   wsadmin>$AdminTask manageItfimPartner { -operation createResponseFile
   -fimDomainName fimspdomain -federationName saml11sp -partnerRole ip -fileId
   /downloads/saml11_ip_partner_properties.xml }
   ```

   Le message de confirmation suivant s'affiche :

   ```
   FBTADM001I Command completed successfully
   ```

3. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 101. Paramètres du fichier de réponses pour le partenaire du fournisseur d'identités dans la fédération SAML 1.x

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Importez le fichier de métadonnées	Pour pouvoir importer un fichier de métadonnées, vous devez connaître le nom du fichier et son emplacement. (obligatoire)	Nom complet spécifié du fichier de métadonnées. Par exemple : `/downloads/saml11_ip_metadata.xml`	**metadataFileName**
Valider les signatures des demandes de résolution d'artefact	Il est possible de valider les signatures de message SAML lorsque l'artefact du navigateur est utilisé.	Valider les signatures pour l'artefact (A définir sur True.)	**ValidateArtifactResponse**

Tableau 101. Paramètres du fichier de réponses pour le partenaire du fournisseur d'identités dans la fédération SAML 1.x (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Identificateur de clé de validation	Etant donné que les messages POST du navigateur doivent être signés et validés, vous devez indiquer une clé pour valider la signature. Si vous choisissez également de valider des messages lors de l'utilisation d'un artefact de navigateur, utilisez la même clé de validation pour les valider. La clé publique que vous utilisez correspond à la clé privée que votre partenaire utilise pour signer les messages. **Remarque :** Si vous importez des données de partenaire, la clé est fournie dans le fichier de métadonnées. Avant d'importer les données, créez un fichier de clés puis spécifiez-le pour la clé. Avant d'entrer des données de partenaire manuellement, demandez la clé au partenaire et importez-la dans le fichier de clés approprié dans le service de clés Tivoli Federated Identity Manager.	**Méthode utilisant les métadonnées :** • Nom du fichier de clés certifiées : • Libellé de la clé :	**`ValidateKeyIdentifier`**
Validation du certificat serveur	Activer la validation du certificat serveur	A définir sur True.	**`UseSoapServerCertAuth`**
Sélectionner le certificat de validation du serveur	Clé publique du certificat affiché durant les communications SSL avec votre partenaire. Déterminez le certificat que vous et votre partenaire utilisez. Vous devez déjà disposer du certificat et du fichier de clés de ce dernier.	• Mot de passe du fichier de clés certifiées : • Nom du certificat : 30 accès	**`rtAuthKeyId`**

Tableau 101. Paramètres du fichier de réponses pour le partenaire du fournisseur d'identités dans la fédération SAML 1.x (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Informations d'authentification client L'une des options suivantes : • Authentification de base – Username – Mot de passe • Authentification par certificat client – Certificat à présenter au serveur du fournisseur d'identités. Le certificat indiqué est celui que vous et votre partenaire de fournisseur d'identités avez déterminé. – Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée – Mot de passe du fichier de clés	Si votre partenaire exige une authentification mutuelle, déterminez le type à utiliser. • Pour une authentification standard, indiquez un nom d'utilisateur et un mot de passe. • Pour une authentification par certificat client, indiquez le certificat que vous et votre partenaire avec convenu d'utiliser. **Remarque :** Avant d'exécuter cette tâche, assurez-vous que vous et votre partenaire avez convenu de l'emplacement d'obtention du certificat et que vous avez importé ce dernier dans le fichier de clés du service de clés Tivoli Federated Identity Manager.	Désactiver l'authentification de client en définissant les propriétés sur False dans la colonne suivante	**UseClientBasicAuth** **UseSoapClientCertAuth**
Valider la signature des assertions SAML	Validez la signature des assertions SAML. (facultatif)	Activez la validation de la signature SAML. (A définir sur True.)	**com.tivoli.am.fim .sts.saml.1.1 .assertion.verify .signatures**
Sélectionner la clé de validation pour la signature de l'assertion	Indiquez la clé de validation de la signature des assertions à utiliser.	Utiliser l'alias du fichier de clés pour trouver une clé publique pour la validation des signatures (Par défaut)	**SAML11ValidationKey**

Tableau 101. Paramètres du fichier de réponses pour le partenaire du fournisseur d'identités dans la fédération SAML 1.x (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Créer plusieurs instructions d'attribut dans Universal User.	Sélectionnez cette option pour conserver plusieurs instructions d'attribut dans les groupes dans lesquels elles ont été reçues. Cette option peut se révéler nécessaire si vos règles de mappage d'identité personnalisées sont écrites de manière à s'appliquer à un ou plusieurs groupes d'instructions d'attribut spécifiques. Si cette option n'est pas sélectionnée, plusieurs instructions d'attribut sont organisées dans un seul groupe (**AttributeList**) dans le document **STSUniversalUser**.	Définissez la valeur sur False.	**SAML11Create** **MultipleUniversal** **UserAttributes**

4. Entrez la commande suivante dans une invite de commande pour ajouter le nouveau partenaire du fournisseur de services à la fédération de fournisseurs d'identités :

```
wsadmin>$AdminTask manageItfimPartner { -operation create -fimDomainName
fimspdomain -federationName saml11sp -partnerName saml11ip -fileId
/downloads/saml11_ip_partner_properties.xml -signingKeystorePwd testonly}
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

Configuration d'une fédération de fournisseurs d'identités SAML 2.0 à l'aide de l'interface de ligne de commande

Utilisez les commandes de l'interface de ligne de commande pour configurer une fédération de fournisseurs d'identités SAML en créant un fichier de réponses et une fédération de fournisseurs d'identités.

Pourquoi et quand exécuter cette tâche

Cette tâche nécessite l'utilisation de la commande **manageItfimFederation**. La commande **manageItfimFederation** requiert des paramètres spécifiques pour pouvoir exécuter des opérations sur une fédération. Pour plus d'informations, voir *IBM Tivoli Federated Identity ManagerGuide d'administration*.

Procédure

1. Créez un fichier de réponses en exécutant la commande suivante dans la console WebSphere **wsadmin** :

```
wsadmin>$AdminTask manageItfimFederation { -operation createResponseFile
-fimDomainName fimipdomain -role ip -protocol SAML2_0 -fileId
/downloads/saml20_ip_properties.xml }
```

2. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 102. Paramètres du fichier de réponses pour le fournisseur d'identités dans la fédération SAML 2.0

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Nom de la fédération	Nom unique de la fédération (obligatoire)	Tout nom Par exemple, `saml20ip`	**FedName**
Nom de la société	Nom de la société associée à la fédération (obligatoire)	Tout nom Par exemple, `IDP Company Name`	**CompanyName**
Adresse URL de la société	Adresse URL du site Web de la société associée à la fédération (obligatoire)	Adresse URL du site Web de votre société	**CompanyUrl**
Serveur point de contact	Adresse URL du serveur point de contact avec le nom de la fédération et le nom du protocole, par exemple /saml20. (obligatoire)	Adresse URL Par exemple, pour une fédération intitulée `saml_fed: https://idp.example.com/FIM/sps/saml_fed/saml20`	**BaseUrl**
ID fournisseur	Adresse URL ou URN qui identifie uniquement le fournisseur. Par défaut, Tivoli Federated Identity Manager utilise l'adresse URL du serveur point de contact avec le nom de la fédération et le nom du protocole, par exemple /saml20.	Adresse URL Par exemple, pour une fédération intitulée saml_fed:`https://idp.example.com/FIM/sps/saml_fed/saml20`)	**ProviderId**

Sélectionner la clé de signature
 	
Fichier de clés du service de clés Tivoli Federated Identity Manager dans lequel la clé est stockée.
	Entrez une clé de signature pour le fournisseur d'identités.
	
	Le protocole exige que la réponse SAML contenant l'assertion soit signée lors de l'utilisation de la liaison POST HTTP.
	
	Si vous choisissez également de signer tous les autres messages, la clé de signature indiquée est utilisée pour les signer. (obligatoire)
		Nom de fichiers de clés :
		
		nom d'alias de clé :
		
		Ces données sont fournies sous la forme
		`"Keystore Name" _"Alias Name"`
		
		Par exemple :
		`DefaultKeyStore_ clé d'essai`
			SigningKeyIdentifier

Tableau 102. Paramètres du fichier de réponses pour le fournisseur d'identités dans la fédération SAML 2.0 (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande

Sélectionner la clé de chiffrement
 	
Fichier de clés du service de clés Tivoli Federated Identity Manager dans lequel la clé est stockée.
	Biclé publique/privée utilisée pour le chiffrement.
	
	Votre partenaire utilise la clé publique pour chiffrer les données qu'il vous envoie.
	
	Utilisez la clé privée pour déchiffrer les données que votre partenaire vous envoie.
	
	Vous devez indiquer la paire de clés à utiliser.
	
	Remarque : Avant d'exécuter cette tâche, créez la clé et importez-la dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.(obligatoire)
		Nom de fichiers de clés :
		
		nom d'alias de clé :
		
		Ces données sont fournies sous la forme
		`"Keystore Name"`
		`_"Alias Name"`
		
		Par exemple :
		`DefaultKeyStore_`
		`clé d'essai`
			EncryptionKeyIdentifier

Fonction de connexion unique SSO	SAML 2.0 prend en charge la connexion unique à l'aide de différents profils. Utilisez ce paramètre pour les activer en conséquence.	True ou false. Par défaut : False. Vous devez activer au moins une propriété. Par exemple, définissez `SsoPostEnabled` sur True.	**SsoPostEnabled** **SsoArtifactEnabled** **SsoRedirectEnabled**
Déconnexion unique	Pour activer la déconnexion unique, définissez au moins une propriété sur True pour pouvoir choisir la liaison et le fournisseur pouvant être utilisés pour lancer la déconnexion unique.	True ou false. Par défaut : False. Vous devez activer au moins une propriété pour pouvoir activer le profil de déconnexion unique pour la fédération. Par exemple, définissez `SloIPPostEnabled` sur True.	**SloIPArtifactEnabled** **SloIPPostEnabled** **SloIPRedirectEnabled** **SloIPSOAPEnabled** **SloSPArtifactEnabled** **SloSPPostEnabled** **SloSPRedirectEnabled** **SloSPSOAPEnabled**

Tableau 102. Paramètres du fichier de réponses pour le fournisseur d'identités dans la fédération SAML 2.0 (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
URL du service de résolution des artefacts	Artifact Resolution Service est un noeud final SOAP du serveur de point de contact du fournisseur d'identités dans lequel des artefacts sont échangés pour des messages SAML. Par défaut, Tivoli Federated Identity Manager configure un seul noeud final SOAP pour Artifact Resolution Service. Vous pouvez éventuellement définir des noeuds finals SOAP supplémentaires.	Indiquez l'adresse URL du service de résolution d'assertion, l'index de l'URL. Définissez-la sur True si vous utilisez le noeud final par défaut. Sinon, définissez-la sur False. Par exemple, `https://idp.example.com/FIM/sps/saml_fed/saml20/soap;0;true`	**ArtifactResolutionServiceList**
Durée de mise en cache d'artefact (en secondes)	Durée de mise en cache d'artefact en secondes. Valeur par défaut : 120 secondes.	Utilisez la valeur par défaut.	**ArtifactLifetime**
Durée de validité d'une assertion avant sa date d'émission	Durée en secondes pendant laquelle une assertion est considérée valide avant sa date de création. Valeur par défaut : 60	Utilisez la valeur par défaut.	**AssertionValidBefore**
Durée de validité de l'assertion après émission	Durée en secondes pendant laquelle une assertion est considérée valide après sa date de création. Valeur par défaut : 60	Utilisez la valeur par défaut.	**AssertionValidAfter**
Options de mappage d'identité Fichier de transformation XSL (XSLT) contenant les règles de mappage	Type de mappage d'identité à utiliser. Utilisez un fichier XSLT pour le mappage d'identité et préparez le fichier pour la fédération. (obligatoire)	Fichier XSLT correspondant au rôle IP pour les fédérations SAML 2.0 : `/opt/IBM/FIM/examples/mapping_rules/ip_saml_20_email_nameid.xsl`	**MappingRuleFileName**

3. Entrez la commande suivante dans une invite de commande pour créer la fédération de fournisseur d'identités :

```
wsadmin>$AdminTask manageItfimFederation { -operation create
-fimDomainName fimipdomain -fileId
/downloads/saml20_ip_properties.xml }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

Que faire ensuite

Poursuivez avec Configuration d'une fédération de fournisseurs de services SAML 2.0 à l'aide de l'interface de ligne de commande.

Configuration d'une fédération de fournisseurs de services SAML 2.0 à l'aide de l'interface de ligne de commande

Utilisez les commandes de l'interface de ligne de commande pour configurer une fédération de fournisseurs de services SAML 2.0 en créant un fichier de réponses et une fédération de fournisseurs de services.

Pourquoi et quand exécuter cette tâche

Cette tâche nécessite l'utilisation de la commande **manageItfimFederation**. La commande **manageItfimFederation** requiert des paramètres spécifiques pour pouvoir exécuter des opérations sur une fédération. Pour plus d'informations, voir *IBM Tivoli Federated Identity ManagerGuide d'administration*.

Procédure

1. Créez un fichier de réponses en exécutant la commande suivante dans la console WebSphere **wsadmin** :

   ```
   wsadmin>$AdminTask manageItfimFederation { -operation createResponseFile
   -fimDomainName fimspdomain -protocol SAML2_0 -role sp -fileId
   /downloads/saml20_sp_properties.xml }
   ```

   Le message de confirmation suivant s'affiche :

   ```
   FBTADM001I Command completed successfully
   ```

2. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 103. Paramètres du fichier de réponses pour le fournisseur de services dans la fédération SAML 2.0

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Nom de la fédération	Nom unique de la fédération (obligatoire)	Tout nom Par exemple, `saml20sp`	**FedName**
Nom de la société	Nom de la société associée à la fédération (obligatoire)	Tout nom Par exemple, `SP Company Name`	**CompanyName**
Adresse URL de la société	Adresse URL du site Web de la société associée à la fédération (obligatoire)	Adresse URL du site Web de votre société	**CompanyUrl**
ID fournisseur	Adresse URL ou URN qui identifie uniquement le fournisseur. Par défaut Tivoli Federated Identity Manager utilise l'adresse URL du serveur point de contact avec le nom de la fédération et le nom du protocole, par exemple /saml20.	URL Par exemple, pour une fédération intitulée saml_fed: `https://sp.example.com/FIM/sps/saml_fed/saml20`	**ProviderId**

Tableau 103. Paramètres du fichier de réponses pour le fournisseur de services dans la fédération SAML 2.0 (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
URL du serveur point de contact	Adresse URL du serveur point de contact avec le nom de la fédération et le nom du protocole, par exemple /saml20. (obligatoire)	Adresse URL Par exemple, pour une fédération intitulée `saml_fed:` `https://` `sp.example.com/FIM/` `sps/saml_fed/saml20`	**`BaseUrl`**

Sélectionner la clé de signature
 	
Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée.
	Entrez une clé de signature pour le fournisseur de services. Si vous choisissez également de signer tous les autres messages, la clé de signature indiquée est utilisée pour les signer. (obligatoire)
	
	Remarque : Avant d'exécuter cette tâche, créez la clé et importez-la dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.
		Nom de fichiers de clés :
		
		nom d'alias de clé :
		
		Ces données sont fournies sous la forme
		`"Keystore Name"` `_"Alias Name"`
		
		Par exemple :
		`DefaultKeyStore_` `clé d'essai`
			`SigningKeyIdentifier`

Connexion unique	URL à laquelle le fournisseur de services envoie les demandes d'authentification.	True ou false. Par défaut : False. Vous devez activer au moins une propriété. Par exemple, définissez `SsoPostEnabled` sur True.	**`SsoPostEnabled`** **`SsoArtifactEnabled`** **`SsoRedirectEnabled`**

Sélectionner la clé de chiffrement
 	
Fichier de clés du service de clés Tivoli Federated Identity Manager, dans lequel la clé est stockée.
	Biclé publique/privée utilisée pour le chiffrement. Votre partenaire utilise la clé publique pour chiffrer les données qu'il vous envoie.
	
	Utilisez la clé privée pour déchiffrer les données que votre partenaire vous envoie.
	
	Vous devez indiquer la paire de clés à utiliser.
	
	Remarque : Avant d'exécuter cette tâche, créez la clé et importez-la dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.(obligatoire)
		Nom de fichiers de clés :
		
		nom d'alias de clé :
		
		Ces données sont fournies sous la forme
		`"Keystore Name"` `_"Alias Name"`
		
		Par exemple :
		`DefaultKeyStore_` `clé d'essai`
			`EncryptionKeyIdentifier`

Tableau 103. Paramètres du fichier de réponses pour le fournisseur de services dans la fédération SAML 2.0 (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Profil de déconnexion unique	URL qu'utilise le partenaire pour accéder au profil Déconnexion unique. Pour activer la déconnexion unique, définissez au moins une propriété sur True. Vous pouvez ensuite choisir la liaison et le fournisseur pouvant être utilisés pour lancer la déconnexion unique.	True ou false. Par défaut : False. Vous devez activer au moins une propriété pour pouvoir activer le profil de déconnexion unique pour la fédération. Par exemple, définissez `SloSPPostEnabled` sur True.	**SloIPArtifactEnabled** **SloIPPostEnabled** **SloIPRedirectEnabled** **SloIPSOAPEnabled** **SloSPArtifactEnabled** **SloSPPostEnabled** **SloSPRedirectEnabled** **SloSPSOAPEnabled**
Liste de services de résolution des artefacts	Artifact Resolution Service est un noeud final SOAP du serveur de point de contact du fournisseur de services dans lequel des artefacts sont échangés pour des messages SAML. Par défaut, Tivoli Federated Identity Manager configure un seul noeud final SOAP pour Artifact Resolution Service. Vous pouvez éventuellement définir des noeuds finals SOAP supplémentaires.	Indiquez l'adresse URL du service de résolution d'assertion, l'index de l'URL, puis définissez-la sur True si le noeud final est utilisé par défaut. Sinon, définissez-la sur False. Par exemple, `https://sp.example.com/FIM/sps/saml_fed/saml20/soap;0;true`	**ArtifactResolutionServiceList**
Options de mappage d'identité Fichier de transformation XSL (XSLT) contenant les règles de mappage	Type de mappage d'identité à utiliser. Utilisez un fichier XSLT pour le mappage d'identité et préparez le fichier pour la fédération. (obligatoire)	Fichier XSLT correspondant au rôle SP pour les fédérations SAML 2.0 : `/opt/IBM/FIM/examples/mapping_rules/sp_saml_20.xsl`	**MappingRuleFileName**

3. Entrez la commande suivante dans une invite de commande pour créer la fédération de fournisseurs de services :

```
wsadmin>$AdminTask manageItfimFederation { -operation create
-fimDomainName fimspdomain -fileId
/downloads/saml20_sp_properties.xml }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

Que faire ensuite

Poursuivez avec Importation d'un fournisseur de services SAML 2.0 dans la fédération de fournisseurs d'identités SAML.

Importation d'un fournisseur de services SAML 2.0 dans la fédération de fournisseurs d'identités SAML

Pour ajouter un fournisseur de services à la fédération de fournisseurs d'identités, vous devez importer les propriétés de configuration du fournisseur de services.

Procédure

1. Entrez la commande suivante dans une invite de commande pour exporter les métadonnées du fournisseur de services et obtenir des informations sur l'environnement :

   ```
   wsadmin>$AdminTask manageItfimFederation { -operation export
   -fimDomainName fimspdomain -federationName saml20sp -fileId
   /downloads/saml20_sp_metadata.xml }
   ```

   Le message de confirmation suivant s'affiche :

   ```
   FBTADM001I Command completed successfully
   ```

2. Créez un fichier de réponses de fournisseur de services en exécutant la commande suivante dans la console WebSphere **wsadmin** :

   ```
   wsadmin>$AdminTask manageItfimPartner { -operation createResponseFile
   -fimDomainName fimipdomain -federationName saml2ip -partnerRole sp -fileId
   /downloads/saml20_sp_partner_properties.xml }
   ```

   Le message de confirmation suivant s'affiche :

   ```
   FBTADM001I Command completed successfully
   ```

3. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 104. Paramètres du fichier de réponses pour le partenaire du fournisseur de services dans la fédération SAML 2.0

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Importez le fichier de métadonnées	Pour pouvoir importer un fichier de métadonnées, vous devez connaître le nom du fichier et son emplacement. (obligatoire)	Nom complet spécifié du fichier de métadonnées. Par exemple : `/downloads/saml20_ sp_metadata.xml`	**metadataFileName**
Options de validation de signature	Les métadonnées du partenaire contiennent la clé à utiliser pour la validation des signatures. Indiquez le magasin de clés et le nom d'alias où Tivoli Federated Identity Manager stocke la clé incluse aux métadonnées.	`DefaultTrustedKeyStore`	**signatureKeystoreName**

Tableau 104. Paramètres du fichier de réponses pour le partenaire du fournisseur de services dans la fédération SAML 2.0 (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
	Nom d'alias sous lequel la clé est stockée dans le fichier de clés spécifié.	`spsignkey`	**signatureKeyAlias**
Options de chiffrement	Les métadonnées du partenaire contiennent la clé à utiliser pour le chiffrement. Indiquez le nom et le nom d'alias du fichier de clés dans lequel Tivoli Federated Identity Manager stocke la clé incluse aux métadonnées.	`DefaultTrustedKeyStore`	**encryptionKeystore**
	Nom d'alias sous lequel la clé est stockée dans le fichier de clés spécifié.	`spenckey`	**encryptionKeyAlias**
Types d'attributs d'assertion	Indiquez les types d'attributs à ajouter à l'assertion générée par le fournisseur d'identités.	`*` (Par défaut)	**AssertionAttributeTypes**
Validation du certificat SSL du serveur partenaire	Le fournisseur d'identités établit une connexion directe avec le fournisseur de services pour certaines liaisons SAML. Indiquez la clé à utiliser pour valider le certificat SSL du serveur. **Remarque :** Avant d'exécuter cette tâche, importez la clé dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.	`DefaultTrustedKeystore_spsslcert`	**ServerCertKeyId**

4. Entrez la commande suivante dans une invite de commande pour ajouter le nouveau partenaire du fournisseur de services à la fédération de fournisseurs d'identités :

```
wsadmin>$AdminTask manageItfimPartner { -operation create
-fimDomainName fimipdomain
-federationName saml20ip -partnerName saml20sp -fileId
/downloads/saml20_sp_partner_properties.xml
-signingKeystorePwd testonly}
-encryptionKeystorePwd testonly }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

Que faire ensuite

Poursuivez avec Importation d'un fournisseur d'identités SAML 2.0 dans la fédération de fournisseurs de services SAML.

Importation d'un fournisseur d'identités SAML 2.0 dans la fédération de fournisseurs de services SAML

Pour ajouter un fournisseur d'identités à la fédération de fournisseurs de services, vous devez importer les propriétés de configuration du fournisseur d'identités.

Procédure

1. Entrez la commande suivante dans une invite de commande pour exporter les métadonnées du fournisseur d'identités et obtenir des informations sur l'environnement :

```
wsadmin>$AdminTask manageItfimFederation { -operation export
-fimDomainName fimipdomain -federationName saml20ip -fileId
/downloads/saml20_ip_metadata.xml }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

2. Créez un fichier de réponses du fournisseur d'identités en exécutant la commande suivante dans la console WebSphere **wsadmin** :

```
wsadmin>$AdminTask manageItfimPartner { -operation createResponseFile
-fimDomainName fimspdomain -federationName saml20sp -partnerRole ip -fileId
/downloads/saml20_ip_partner_properties.xml }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

3. Editez le fichier de réponses pour modifier les valeurs suivantes :

Tableau 105. Paramètres du fichier de réponses pour le partenaire du fournisseur d'identités dans la fédération SAML 2.0

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Importez le fichier de métadonnées	Pour pouvoir importer un fichier de métadonnées, vous devez connaître le nom du fichier et son emplacement. (obligatoire)	Nom complet spécifié du fichier de métadonnées. Par exemple : `/downloads/saml20_ip_metadata.xml`	**metadataFileName**
Options de validation de signature	Les métadonnées du partenaire contiennent la clé à utiliser pour la validation des signatures. Indiquez le magasin de clés et le nom d'alias où Tivoli Federated Identity Manager stocke la clé incluse aux métadonnées.	`DefaultTrustedKeyStore`	**signatureKeystoreName**
	Nom d'alias sous lequel la clé est stockée dans le fichier de clés spécifié.	`ipsignkey`	**signatureKeyAlias**
Options de chiffrement	Les métadonnées du partenaire contiennent la clé à utiliser pour le chiffrement. Indiquez le nom et le nom d'alias du fichier de clés dans lequel Tivoli Federated Identity Manager stocke la clé incluse aux métadonnées.	`DefaultTrustedKeyStore`	**encryptionKeystore**
	Nom d'alias sous lequel la clé est stockée dans le fichier de clés spécifié.	`ipenckey`	**encryptionKeyAlias**

Tableau 105. Paramètres du fichier de réponses pour le partenaire du fournisseur d'identités dans la fédération SAML 2.0 (suite)

Elément de configuration	Description	Votre valeur	Propriétés ou noms de l'interface de ligne de commande
Validation du certificat SSL du serveur partenaire	Le fournisseur d'identités établit une connexion directe avec le fournisseur de services pour certaines liaisons SAML. Indiquez la clé à utiliser pour valider le certificat SSL du serveur. **Remarque :** Avant d'exécuter cette tâche, importez la clé dans le fichier de clés approprié du service de clés Tivoli Federated Identity Manager.	`DefaultTrustedKeystore_ipsslcert`	**`ServerCertKeyId`**
Adresse URL cible par défaut	Adresse URL cible par défaut à laquelle le navigateur est envoyé lorsqu'une connexion unique aboutit sur le fournisseur de services. Cet emplacement est uniquement utilisé si aucune adresse URL cible n'a été spécifiée dans la requête.	`https://saml20clisp:444/FIM/fimivt/protected/ivtlanding.jsp`	**`DefaultPostAuthTargetURL`**
Utilisateur par défaut pour connexion unique	Nom d'utilisateur par défaut pour la connexion unique côté fournisseur de services.	`guest`	**`AnonymousUserUserName`**

4. Entrez la commande suivante dans une invite de commande pour ajouter le nouveau partenaire du fournisseur de services à la fédération de fournisseurs d'identités :

```
wsadmin>$AdminTask manageItfimPartner { -operation create -fimDomainName
fimspdomain -federationName saml20sp -partnerName saml20ip -fileId
/downloads/saml20_ip_partner_properties.xml -signingKeystorePwd testonly}
-encryptionKeystorePwd testonly }
```

Le message de confirmation suivant s'affiche :

```
FBTADM001I Command completed successfully
```

Chapitre 21. Planification d'une fédération Information Card

Ce guide de planification passe en revue la norme d'implémentation sous Tivoli Federated Identity Manager de la norme Information Card et explique comment planifier le processus de configuration. Ce guide ne constitue pas une présentation exhaustive de la norme Information Card.

Vous pouvez utiliser le système Information Card afin de gérer vos identités numériques à partir de divers fournisseurs d'identité. Ensuite, vous pouvez utiliser ces identités numériques pour accéder à divers services qui les acceptent.

Il convient que les administrateurs non familiarisés avec cette norme consultent la documentation relative à Information Card sur le site Web de Microsoft.

La prise en charge de Tivoli Federated Identity Manager pour Information Card inclut le déploiement de Tivoli Federated Identity Manager suivant les deux rôles fournis par Information Card : fournisseur d'identité gérée (Managed Identity Provider) et partie de confiance (Relying Party).

Le flux de protocole établi lorsque l'utilisateur fournit une carte d'information pour s'authentifier sur un site Web, s'apparente à un flux de connexion basée sur des formulaires. Ce flux requiert toutefois des étapes supplémentaires.

1. L'utilisateur dirige le navigateur sur une page Web protégée nécessitant une authentification.
2. Le site redirige le navigateur vers une page de connexion. Dans un navigateur compatible avec Information Card, la page de connexion contient une balise HTML qui permet à l'utilisateur de choisir une carte d'information afin de s'authentifier sur le site. Lorsque l'utilisateur sélectionne la balise, le navigateur démarre un *sélecteur d'identité*.

   Remarque : Un *sélecteur d'identité* est un module d'extension de navigateur qui permet à celui-ci d'utiliser le protocole Information Card. Les modules d'extension portent parfois le nom d'*agents d'identité*.
3. Le code de prise en charge du navigateur concernant les cartes d'information démarre le sélecteur d'identité. Le navigateur le transmet ensuite aux valeurs de paramètres fournies par la balise HTML d'Information Card obtenue à partir du site Web à l'étape 2.

   L'utilisateur sélectionne ensuite une carte d'information, qui représente une identité numérique utilisable pour s'authentifier sur le site.
4. Le sélecteur d'identité envoie la carte d'information au fournisseur d'identité Tivoli Federated Identity Manager. Le fournisseur d'identité utilise le service STS de Tivoli Federated Identity Manager pour traiter le message WS-Trust et les données de WS-Metadata Exchange. Il génère ensuite un jeton contenant les données d'identification de l'utilisateur. Le fournisseur d'identité renvoie le jeton au navigateur.

   Remarque : IBM a déprécié le client Tivoli Federated Identity Manager Security Token Service (STS) dans cette version.

   Si vous utilisez WebSphere 6.X, vous pouvez continuer de vous servir du client Tivoli Federated Identity Manager Security STS tant que Tivoli Federated Identity Manager prend en charge WebSphere 6.X. Lorsque Tivoli Federated

Identity Manager arrêtera son support pour WebSphere 6.X, vous devrez utiliser WebSphere Application Server version 7 Update 11 et version ultérieure. Voir API client WS-Trust et WS-Trust Clients pour plus d'informations.

5. Le navigateur transmet les données d'identification de l'utilisateur au site Web qui protège les ressources demandées. Le site valide ces droits d'accès, puis redirige le navigateur sur la page demandée à l'origine.

Dans le flux de protocole, la partie de confiance et le fournisseur d'identité ne communiquent pas directement ensemble. Par défaut, aucune partie n'a connaissance de l'existence de l'autre partie. La partie de confiance ne sait pas quel fournisseur d'identité a été sélectionnée par l'utilisateur tant que le jeton n'a pas été reçu à l'étape 5. A ce moment-là, la partie de confiance peut prendre connaissance de l'identité en examinant le continue de la zone de l'émetteur (Issuer) dans le jeton.

Vous pouvez utiliser Information Card pour inviter le fournisseur d'identité à demander l'identification depuis la partie de confiance. Toutefois, cela n'est pas une obligation, et n'est généralement pas recommandé.

Présentation du fournisseur d'identité Information Card

Lorsque Tivoli Federated Identity Manager fait office de fournisseur d'identité, il prend en charge l'émission des cartes gérées et émet des jetons de sécurité pour les cartes gérées.

Le fournisseur d'identité prend en charge les éléments suivants :

- Emission de cartes gérées

  L'émission de cartes gérées a lieu lorsqu'un utilisateur s'authentifie auprès d'un fournisseur d'identité Tivoli Federated Identity Manager et accède à une adresse URL de téléchargement de carte. L'adresse URL envoie à l'utilisateur un modèle de formulaire HTML lui demandant d'indiquer les informations requises pour émettre la carte. Une fois que l'utilisateur a fourni les informations requises, Tivoli Federated Identity Manager émet la carte et l'envoie vers le navigateur de l'utilisateur. Celui-ci peut alors la sauvegarder pour un usage ultérieur.
- Extraction des jetons de sécurité pour les cartes gérées

  Cette prise en charge est permise par le service de jeton de sécurité (STS). Ce composant prend en charge d eux types de messages SOAP issus d'un sélecteur d'identité Information Card. Les messages SOAP sont requis pour permettre à un sélecteur d'identité d'obtenir un jeton de sécurité destiné à la carte d'information gérée de l'utilisateur.

  Remarque : Un *sélecteur d'identité* est un module d'extension de navigateur. On l'appelle parfois *agent d'identité*.

  Seuls les jetons de sécurité SAML 1.1 sont pris en charge.

La prise en charge concerne les caractéristiques suivantes :

- Emission de cartes gérées
- Noeuds finals pour l'échange de métadonnées et traitement des messages WS-Trust
- Prise en charge des réclamations Information Card
- Fédération unique destinée à contenir les noeuds finals du fournisseur d'identité
- Une chaîne d'accréditation sécuritaire destinée à convertir les informations d'identité des utilisateurs en jetons SAML 1.1

Remarque : Les fédérations Information Card ne permettent pas de maintenir les paramètres de configuration sous forme de métadonnées. Il n'existe aucune métadonnées à exporter ou importer entre les fournisseurs d'identité et les parties de confiance pour les déploiements Information Card.

Emission de cartes gérées

Tivoli Federated Identity Manager permet de prendre en charge l'émission de cartes gérées par des fournisseurs d'identité, ainsi que l'extraction de jetons de sécurité à partir de cartes gérées émises par d'autres autorités.

Tivoli Federated Identity Manager fournit un noeud final sécurisé permettant de télécharger une carte gérée. Lorsqu'un utilisateur accède au noeud final via un navigateur, un fichier modèle HTML est chargé, puis renvoyé à l'utilisateur. L'utilisateur est invité à fournir les informations requises pour permettre l'émission de la carte gérée.

Les informations obligatoires sont les suivantes :

- Nom d'utilisateur

  Le nom d'utilisateur est une valeur arbitraire attribuée à la carte par l'utilisateur.
- Ensemble des réclamations prises en charge par la carte.

  Une réclamation est un identificateur URI (Uniform Resource Indidentificator) qui représente des noms d'attributs qualifiés. Tivoli Federated Identity Manager utilise la liste des réclamations pour déterminer la nature des informations qui doivent être placées dans le jeton de sécurité généré au moment de l'exécution, lorsque la carte gérée est traitée. Les exemples des informations placées dans le jeton de sécurité sont chaque réclamation ainsi que la valeur correspondante.
- Lorsque la fédération utilise une méthode d'authentification appelée *droits d'accès auto-émis* ou *assertion SAML auto-signée*, une partie de la demande consiste à inviter l'utilisateur à fournir un jeton généré par une carte auto-émise.

  Lorsque la fédération utilise une méthode d'authentification appelée *jeton de nom d'utilisateur*, l'utilisateur n'est pas tenu de spécifier ce paramètre.

Tivoli Federated Identity Manager fournit deux modèles de pages HTML.

- Lorsque la méthode d'authentification repose sur un jeton de nom d'utilisateur, le modèle utilisé est getcard_ut.html.
- Lorsque la méthode d'authentification repose sur des droit d'accès émis de façon autonome, le modèle utilisé est getcard_sss.html.

Les administrateurs peuvent modifier les fichiers modèles HTML afin que ceux-ci répondent au mieux aux exigences du déploiement local.

Les fichiers modèles getcard_* contiennent les macros suivantes, qui sont remplacées par des valeurs spécifiques à la demande émise par l'utilisateur.

@FORMACTION@
: Cette macro est remplacée par l'action URL du formulaire requis vers laquelle le formulaire HTML est envoyé.

@USERNAME@
: Cette macro est remplacée par le nom d'utilisateur tel que spécifié soit par le nom de connexion de l'utilisateur Tivoli Access Manager, soit par un utilisateur WebSphere authentifié. Le nom d'utilisateur Tivoli Access

Manager est employé lorsque WebSEAL est le serveur point de contact. Le nom d'utilisateur WebSphere est employé lorsque WebSphere est le serveur point de contact.

Cette valeur peut être utilisée pour renseigner préalablement le paramètre du nom de carte dans le modèle.

Lorsque l'utilisateur renvoie le formulaire par requête POST à Tivoli Federated Identity Manager, les informations sont placées dans les macros contenues dans une fichier modèle XML appelé infocard_template.xml. Ce fichier modèle représente la carte gérée renvoyée à l'utilisateur via le navigateur.

Dans la plupart des déploiements, les administrateurs système n'ont pas besoin de modifier les macros dans infocard_template.xml. Toutefois, le fichier contient un certain nombre de macros qui peuvent être modifiées si nécessaire.

Remarque : Pour afficher la liste des macros, voir «Macros de remplacement dans le fichier XML infocard_template», à la page 333.

La prise en charge d'Information Card par Tivoli Federated Identity Manager s'applique uniquement au type de jeton SAML 1.1. Deux représentations existent pour le type de jeton SAML 1.1 :

SAML 1.1
urn:oasis:names:tc:SAML:1.0:assertion

SAML 1.1
http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1

La plupart des cartes d'information gérées prennent en charge les deux représentations. L'utilisateur ne sélectionne pas le type de jeton. La macro@SUPPORTED_TOKENS@ contenue dans le fichier infocard_template.xml est définie dans les deux représentations SAML ci-dessus.

Tivoli Federated Identity Manager prend en charge deux méthodes utilisées par le sélecteur d'identité pour authentifier l'utilisateur auprès du noeud final du fournisseur d'identité (service STS). Chaque méthode prend en charge un modèle de remplacement distinct pour la macro @USERCRED@ dans le modèle de carte d'information (infocard_template.xml).

Le type d'authentification est spécifié par l'administrateur lors de la configuration de la fédération. Les valeurs de configuration du paramètre `authenticationMethod` correspondent aux fichiers modèles comme suit :

UsernameToken
Renvoie au fichier modèle **infocard_usercred_usernametoken.xml**

Le fichier modèle contient une macro de remplacement :

@USERNAME@
Cette macro est remplacée par le nom d'utilisateur. Le nom d'utilisateur est indiqué soit par le nom de connexion de l'utilisateur Tivoli Access Manager soit par un utilisateur WebSphere authentifié. Le nom d'utilisateur Tivoli Access Manager est employé lorsque WebSEAL est le serveur point de contact. Le nom d'utilisateur WebSphere est employé lorsque WebSphere est le serveur point de contact.

SelfSignedSAML

Renvoie au fichier modèle **infocard_usercred_selfsignedsaml.xml**

Le fichier modèle contient une macro de remplacement :

@PPID@

La macro est remplacée par l'identificateur PPID de la carte auto-émise envoyée avec le formulaire getcard_sss.html. Ce processus se produit lorsque la fédération utilise la méthode d'authentification SelfSignedSAML.

Tivoli Federated Identity Manager stocke cette valeur sous forme d'alias pour l'utilisateur en cours dans le service d'alias de Tivoli Federated Identity Manager. L'alias sert à mapper en retour la carte auto-émise avec l'utilisateur Tivoli Federated Identity Manager.

Ce processus se produit lorsque la carte auto-émise est utilisée au moment de l'exécution pour générer une assertion SAML et authentifier le service de jeton de sécurité du fournisseur d'identité.

Fédérations de fournisseurs d'identité

La configuration des fédérations liées à Information Card diffère de celle des fédérations destinées aux autres protocoles de connexion unique tels que SAML 2.0, Liberty, WS-Federation ou OpenID. La principale différence réside dans le fait que le fournisseur d'identité Information Card n'a pas besoin de connaître le destinataire du jeton de sécurité. Le service STS du fournisseur d'identité interagit uniquement avec le sélecteur d'identité. Cela permet d'éliminer la nécessité de configurer des propriétés qui contiennent des informations relatives aux partenaires.

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Le concept de configuration partenaire n'existe dans les configurations Information Card que dans le cadre de la configuration des modules de jetons utilisés par le service d'accréditation.

Les propriétés clés qui définissent une fédération pour un fournisseur d'identité sont les suivantes :

ProtocolID

Tivoli Federated Identity Manager utilise un ID de protocole en tant qu'identificateur unique. La fédération Information Card comporte la syntaxe protocolId suivante :

`https://<nom_hôte:port>/FIM/sps/<nom_fédération>/infocard`

Par exemple :

`https://www.exampleidentitydemo.com/FIM/sps/csip/infocard`

Noeud final permettant d'obtenir une carte gérée

Noeud final destiné au traitement de l'interaction HTML avec un utilisateur authentifié, afin de permettre la génération et le téléchargement d'une carte gérée.

L'adresse URL du noeud final repose sur le paramètre ProtocolID. Par exemple :

`https://www.exampleidentitydemo.com/FIM/sps/csip/infocard/getcard.crd`

Le composant Tivoli Federated Identity Manager (délégué du service de protocole de connexion unique) du noeud final exécute les tâches suivantes :

1. Invite l'utilisateur à indiquer les informations requises pour générer une carte d'information. Les données de la carte d'information concernent le nom de la carte et les réclamations prises en charge. Lorsque la méthode d'authentification repose sur des droit d'accès émis de façon autonome, une carte d'information personnelle est générée.
2. Lorsque le mécanisme d'authentification repose sur une carte émise de façon autonome, SelfSignedSAML, le délégué crée et stocke un alias dans le service concerné. Une correspondance entre l'alias et la carte personnelle présentée par l'utilisateur durant ce processus est établie dans le compte utilisateur de la personne actuellement authentifiée dans la session de navigateur.
3. Génère la carte gérée à partir d'un modèle XML dont les divers composants sont alimentés dynamiquement. Le délégué signe la carte en utilisant la clé privée du certificat SSL associé au serveur point de contact. Ensuite, le délégué renvoie la carte au navigateur.

Noeud final d'échange de métadonnées

Le sélecteur d'identité utilise un noeud final au moment de l'exécution pour échanger des métadonnées. Cette utilisation détermine la connexion RST et les exigences de formatage des messages du service de jeton de sécurité du fournisseur d'identité.

L'adresse URL du noeud final de métadonnées repose sur le paramètre ProtocolID. Par exemple :

```
https://www.exampleidentitydemo.com/FIM/sps/csip/infocard/mex
```

Le noeud final d'échange de métadonnées comporte un fichier modèle XML appelé metadata_template.xml. Ce fichier contient des macros disponibles en vue d'un remplacement.

Remarque : Les administrateurs peuvent utiliser les macros par défaut. Il n'est pas nécessaire de modifier les macros pour pouvoir utiliser le fichier modèle.

Les macros de remplacement du fichier metadata_template.xml sont les suivantes :

@IPSTS@
Adresse URL du noeud final de service STS pour le fournisseur d'identité dans la fédération.

@IPPOLICY@
Cette valeur contient des informations WS-Policy. Les informations dépendent du type de jeton d'authentification utilisé pour l'authentification auprès du service STS du fournisseur d'identité. Les informations WS-Policy information sont lues à partir d'un fichier modèle.

@IPCERTIFICATE@
Certificat SSL public codé sur base 64 du serveur point de contact.

Chacune des méthodes d'authentification prend en charge un modèle de remplacement distinct pour la macro @IPPOLICY@ dans le modèle d'échange de métadonnées.

Les fichiers modèles de chaque méthode d'authentification sont les suivants :

Authentification avec UsernameToken
metadata_policy_usernametoken.xml

Authentification avec SelfSignedSAML
metadata_policy_selfsignedsaml.xml

Les fichiers metadata_policy_usernametoken.xml et metadata_policy_selfsignedsaml.xml ne comportent aucune macro de remplacement. Les fichiers modèles sont constitués de différents ensembles de règles adaptés à chaque méthode. Les administrateurs Information Card n'ont pas besoin de modifier ces fichiers.

Noeud final pour la réception des messages WS-Trust
Le service STS du fournisseur d'identité dispose d'un noeud final qui reçoit les messages WS-Trust en provenance du sélecteur d'identité. Le module du fournisseur d'identité Information Card traite la demande entrante, modifie le service d'accréditation Tivoli Federated Identity Manager et communique avec le service d'accréditation pour obtenir le jeton.

Réclamations Information Card

Information Card utilise des informations appelées *réclamations* (claims) pour définir les attributs susceptibles d'être nécessaires lors de la satisfaction d'une requête d'utilisateur. Une carte d'information contient les indicateurs URI (Uniform Resource Indicators) relatifs à l'ensemble des réclamations prises en charge par leur émetteur.

Un sélecteur d'identité peut utiliser les informations des réclamations afin de déterminer si une carte d'identité peut être utile à la connexion auprès d'une partie de confiance spécifique. A titre d'exemple, lorsqu'une partie de confiance exige la réclamation relative à une adresse électronique et que le fournisseur d'identité associé à une carte gérée ne prend pas en charge cette réclamation, le fournisseur d'identité ne proposera pas la carte gérée en tant qu'option d'ouverture de session auprès de la partie de confiance concernée.

Le fournisseur de carte gérée de Tivoli Federated Identity Provider n'oppose aucune restriction quant à l'ensemble des réclamations qui peuvent être spécifiées dans les cartes. Les modèles (getcard_ut.html et getcard_sss.html) contiennent la série complète des réclamations standard prises en charge. Les administrateurs peuvent ajouter la prise en charge de réclamations supplémentaires en modifiant les modèles.

L'agent d'identité Information Card envoie une requête WS-Trust au module Tivoli Federated Identity Manager (délégué) portant sur le service du protocole de connexion unique. La requête WS-Trust contient un élément de réclamation (wst:Claims) dans lequel se trouve l'ensemble des réclamations émises.

La figure 17, à la page 308 illustre quelques exemples de réclamations.

```
<wst:Claims>
 <wsid:ClaimType
  Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
  xmlns:wsid="http://schemas.xmlsoap.org/ws/2005/05/identity" />
 <wsid:ClaimType
  Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
  xmlns:wsid="http://schemas.xmlsoap.org/ws/2005/05/identity" />
 <wsid:ClaimType
  Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"
  xmlns:wsid="http://schemas.xmlsoap.org/ws/2005/05/identity" />
 <wsid:ClaimType
  Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
        privatepersonalidentifier"
  xmlns:wsid="http://schemas.xmlsoap.org/ws/2005/05/identity" />
</wst:Claims>
```

Pour afficher la série standard des réclamations prises en charge sous Microsoft, voir : http://schemas.xmlsoap.org/ws/2005/05/identity/claims.xsd.

Figure 17. Exemples de réclamations provenant d'un agent d'identité Information Card

Pages d'erreurs Information Card

Les identificateurs de page suivants sont fournis :

/infocard/error_get_card.html
Renvoie à la page suivante :
`/infocard/error_get_card.html`

Permet d'afficher une erreur au format HTML lorsqu'un utilisateur tente de télécharger une carte.

/infocard/error_get_metadata.html
Renvoie à la page suivante :
`/infocard/error_get_metadata.html`

Permet d'afficher une erreur au format HTML lorsqu'un utilisateur sélectionnant une identité tente de télécharger des métadonnées via la requête HTTP GET (au lieu de SOAP sur HTTP/POST).

Présentation de la partie de confiance Information Card

Le rôle de la partie de confiance est similaire à celui d'un *fournisseur de services* en ce sens qu'elle est prise en charge par Tivoli Federated Identity Manager pour d'autres protocoles de connexion unique. La partie de confiance se compose d'un service de connexion implémenté dans un composant de service de connexion unique (délégué), ainsi qu'une chaîne WS-Trust.

L'implémentation Tivoli Federated Identity Manager prend en charge les activités suivantes :

- Réception des jetons d'assertion SAML 1.x
- L'utilisation de la connexion à la fois avec les cartes auto-émises et les cartes gérées émises par les autres fournisseurs d'identité.

Dans le modèle Information Card, la clé publique SSL (Secure Socket Layer) sert à chiffrer le jeton qui est envoyé à la partie de confiance sur les noeuds finals. La clé

SSL est la clé de la session SSL établie entre le navigateur et le site présentant la page Web (conformément à la spécification des balises OBJECT imbriquées). Ceci implique que Tivoli Federated Identity Manager a besoin d'accéder aux clés SSL utilisées par le serveur point de contact.

L'administrateur est tenu de configurer l'accès à ces clés durant la configuration de Tivoli Federated Identity Manager Information Card.

Les sites Web doivent utiliser des certificats X509v3 comportant des logotypes (également appelés certificats de validation étendue) à la place des certificats serveur SSL lors d'une identification d'entreprise.

Le terme de **partie de confiance** dans le contexte d'Information Card désigne un rôle similaire à celui de **fournisseur de services** dans d'autres protocoles de connexion unique pris en charge par Tivoli Federated Identity Manager.

En tant que partie de confiance, Tivoli Federated Identity Manager prend à la fois en charge les fournisseurs d'identité gérés et auto-émis.

La configuration de Tivoli Federated Identity Manager permet aux administrateurs de configurer la prise en charge d'un ou deux types de fournisseurs.

Avant de procéder à la configuration de Tivoli Federated Identity Manager, l'administrateur de la partie de confiance peut obtenir des clés publiques auprès du fournisseur d'identité, afin de les utiliser lors de la validation des signatures numériques sur les assertions reçues en provenance dudit fournisseur.

L'implémentation de Tivoli Federated Identity Manager inclut la prise en charge suivante :

- Accès de l'utilisateur à la partie de confiance
- Réclamations Information Card
- Fédérations pour le traitement de requêtes
- Echange de jetons

Accès de l'utilisateur à une partie de confiance

Lorsqu'un utilisateur tente d'accéder à une ressource protégée sur un site Web sans s'être préalablement vu octroyer des droits d'accès, un serveur Web *point de contact* invite généralement l'utilisateur à établir ses droits d'accès en renseignant une page de connexion. L'utilisation d'une carte d'information dans ce scénario dépend des conditions préalables suivantes :

- L'utilisateur doit utiliser un navigateur configuré pour Information Card. Les navigateurs compatibles avec les cartes d'information sont équipés d'un *sélecteur d'identité* installé sous forme de module d'extension.
- La page de connexion émise par le point de contact qui protège les ressources sur le site Web doivent comporter des balises OBJECT spécifiques. Les balises OBJECT contenues dans la page déclenchent l'interaction de Information Card avec le navigateur.
- L'adresse URL à laquelle accède le navigateur doit utiliser le protocole HTTPS.

```
<form method="post" action="/FIM/sps/infocard-fed/infocard/login">
   ...
   <input type="hidden" name="TARGET" value="/TheResource"/>
   <object type="application/x-informationCard" name="xmlToken">
      <param name="requiredClaims"
             value="http://schemas.xmlsoap.org/ws/2005/05/identity/
                    claims/privatepersonalidentifier" />
   </object>
   <input type="submit" value="Login"/>
   ...
</form>
```

Figure 18. Exemple de format de connexion utilisé par la partie de confiance

La figure 18 illustre des exemples d'éléments XML contenus dans le format de connexion requis. Le format de connexion requiert plusieurs paramètres importants :

Action de la méthode de formulaire

La valeur du paramètre `action` doit correspondre à l'adresse URL du noeud final de la fédération Information Card. Le navigateur compatible avec Information Card est redirigé vers ce noeud final afin de traiter le jeton de sécurité émis par le fournisseur d'identité.

Remarque : L'administrateur spécifie ce noeud final lors de la configuration de Tivoli Federated Identity Manager Information Card.

Nom masqué du type d'entrée

Le formulaire de connexion doit normalement comporter un élément `masqué` qui contient :

- Le paramètre `name` défini sur `TARGET`
- Le paramètre `value` défini selon l'adresse URL vers laquelle le navigateur est redirigé lorsque le processus de connexion aboutit.

Il existe une manière alternative de spécifier l'adresse URL vers laquelle le navigateur doit être redirigé. La cible peut être spécifiée au moyen d'un paramètre de requête au format chaîne appliqué à la valeur du paramètre `action`. Par exemple, en utilisant les valeurs de la figure 18 :

```
action=''FIM/sps/infocard-fed/infocard/login?TARGET=/theResource''
```

Lorsque WebSEAL tient le rôle de serveur point de contact, la macro `%URL%` prise en charge par WebSEAL peut être utilisée pour spécifier l'adresse URL cible.

Nom du type d'objet

La valeur du paramètre `name` contenu dans l'élément `OBJECT` doit être définie sur `xmlToken`.

Le navigateur envoie cette valeur à la partie de confiance. L'implémentation de Tivoli Federated Identity Manager pour la partie de confiance Information Card utilise ce paramètre pour accéder au jeton de sécurité.

Tivoli Federated Identity Manager défini en tant que partie de confiance prend en charge les types de jetons SAML suivants :

- Identificateur URI pris en charge par tous les types de fournisseurs :

  ```
  urn:oasis:names:tc:SAML:1.0:assertion
  ```

- Identificateur URI pris en charge uniquement par les fournisseurs d'identité autonomes :

  `http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1`

Il est possible de spécifier l'un ou plusieurs de ces types d'URI dans le paramètre `tokenType` de la balise `OBJECT`.

Fédérations de parties de confiance

Tivoli Federated Identity Manager établit et utilise les fédérations Information Card de manière similaire, mais pas totalement identique aux fédérations employées pour d'autres protocoles de connexion unique. Les différences sont :

- L'interaction entre parties de confiance fait partie du processus d'authentification utilisé par le serveur point de contact (par exemple WebSEAL) pour octroyer l'accès aux ressources protégées.

  Pour les autres protocoles de connexion unique, le serveur point de contact présente une page de connexion lors de l'accès à une ressource protégée, puis authentifie l'utilisateur et produit les droits d'accès destinés à l'utilisateur. Pour Information Card, Tivoli Federated Identity Manager agit en tant que partie de confiance pour exécuter le processus d'authentification et produire les droits d'accès destinés à l'utilisateur.

  La partie de confiance est avertie qu'une connexion utilisateur est en cours au moment où un jeton de sécurité (assertion) est reçu au niveau du noeud final de messages. La partie de confiance doit alors décider d'accepter ou rejeter le jeton de sécurité.
- Contrairement à un fournisseur de services dans le cas des autres protocoles de connexion unique, la partie de confiance n'envoie aucun message au fournisseur d'identité. Les messages sont envoyés par le *Sélecteur d'identité*, sans que la partie de confiance ne soit connue.
- Dans une fédération Information Card, les fournisseurs d'identité constituent une série l'entités fédérées de manière souple dont le site Web accepte les jetons d'assertion.
- Information Card prend en charge le fournisseur d'identité à émission autonome.

Information Card nécessite la création d'une fédération pour représenter la partie de confiance autonome (*self*). Le terme *self* ne doit pas être confondu avec l'autonomie d'émission du fournisseur d'identité. Le terme sert à distinguer l'origine (ou émetteur) de la fédération de tous les partenaires qui y sont ajoutés par la suite. Les propriétés de l'entité autonome sont les suivantes :

- Le noeud final de connexion
- Paramètres indiquant les types de jetons acceptés
- L'alias du fichier de clés issu du serveur point de contact destiné à être utilisé dans les connexions SSL (Secure Socket Layer).
- Une règle de mappage par défaut. La règle de mappage peut être remplacée par une configuration de partenaire.

La fédération Information Card utilise la convention de dénomination standard Tivoli Federated Identity Manager pour le paramètre **protocolID**. La syntaxe est la suivante :

`https://<nom_hôte:port>/FIM/sps/<nom_fédération>/infocard`

Si par exemple l'hôte défini pour les noeuds finals de la fédération est `rp.example.com`, que l'écoute s'effectue sur le port `443` et que le nom de la fédération est `MyInfoCard-rp`, l'ID de protocole est le suivant:

```
https://rp.example.com:443/FIM/sps/MyInfoCard-rp/infocard
```

Les fédérations de partenaires sont nécessaires pour représenter les fournisseurs d'identité. Il ne peut exister qu'un seul partenaire émettant des jetons de façon autonome. Le nombre de partenaires fournisseurs d'identité gérés est illimité. Il est également possible d'ajouter **n'importe quel** partenaire fournisseur d'identité. Ce partenaire peut être utilisé pour l'accès au compte invité.

Partenaire à émission automatique

Tivoli Federated Identity Manager configure un partenaire en définissant le paramètre protocolId comme suit :

```
http://schemas.xmlsoap.org/ws/2005/05/identity/issuer/self
```

Ce partenaire est utilisé pour traiter les cartes auto-émises.

Partenaire de fournisseur d'identité géré désigné

Un partenaire fournisseur géré doit posséder un élément `Issuer URI`. La zone `Issuer` du mappage de la chaîne d'accréditation est définie sur la valeur `protocolID`. Lors de la signature des assertions provenant de ce fournisseur, un alias de clé publique doit être configuré pour le partenaire.

L'administrateur doit importer la clé publique dans un fichier de clés Tivoli Federated Identity Manager avant de configurer la fédération. Il convient d'importe la clé via le service de clés de Tivoli Federated Identity Manager.

Tout fournisseur d'identité partenaire

La configuration de `tout` partenaire permet d'effectuer une configuration générique. Les assertions provenant de ces fournisseurs doivent utiliser *une seule* des valeurs suivantes pour `<saml:SubjectConfirmationMethod>` :

```
urn:oasis:names:tc:SAML:1.0:cm:bearer
urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
```

Lorsque l'assertion est signée, elle doit inclure un élément `<ds:KeyInfo>` inclus dans la signature, contenant une clé publique permettant la validation des signatures.

Remarque : Il convient d'utiliser cette configuration uniquement pour l'accès utilisateur invité. Dans une telle configuration, tous les utilisateurs sont rattachés à un compte invité.

Activation du site Web pour Information Card

L'implémentation Tivoli Federated Identity Manager du profil Information Card interagit avec l'implémentation Microsoft CardSpace™ version 1.0. Ces deux implémentations reposent sur le profil Information Card version 1.0. Cette version est prise en charge sous Microsoft Internet Explorer version 7.

Les navigateurs prenant en charge Information Card doivent :

- reconnaître les balises HTML ou XHTML spéciales pour démarrer le sélecteur d'identité,
- transmettre les paramètres codés au sélecteur d'identité de la plateforme, et
- renvoyer le jeton résultant à partir du type d'authentification sélectionné par l'utilisateur pour le choix d'une identité numérique.

Les sites Web employant l'authentification Information Card doivent prendre en charge deux ensembles de fonctionnalités :

- L'ajout de balises HTML ou XHTML à leur page de connexion afin de pouvoir demander une ouverture de session Information Card
- Le code permettant à l'utilisateur de se connecter au site au moyen des droits d'accès qu'il fournit dans l'opération HTTP POST

En réponse à la connexion Information Card, le site Web se comporte généralement comme suit :

- Inscription du même cookie de navigateur côté client que celui qui serait utilisé si l'ouverture de session avait lieu à partir d'une authentification par nom d'utilisateur et mot de passe (ou suivant d'autres mécanismes)
- Emission des mêmes redirections du navigateur

Modifications des pages de connexion

Les extensions HTML telles que la balise OBJECT sont utilisées pour signaler au navigateur le moment où le sélecteur d'identité doit être démarré. Néanmoins, toutes les extensions HTML ne sont pas prises en charge par l'ensemble des navigateurs.

En outre, certaines extensions HTML couramment prises en charge sont désactivées dans les configurations de navigateur hautement sécurisées. A titre d'exemple, la balise OBJECT est désactivée par les paramètres de sécurité de certains navigateurs, y compris Internet Explorer.

Une alternative à l'utilisation d'extensions HTML consiste à employer une syntaxe XHTML qui n'est pas désactivée lors de la modification des paramètres de sécurité du navigateur. Toutefois, certains navigateurs n'offrent pas une prise en charge complète du format XHTML.

Pour fournir une solution répondant à tous les types de scénarios, deux formats d'extension HTML existent. Les navigateurs peuvent prendre en charge l'un ou l'autre de ces formats d'extension, voire les deux.

Syntaxe OBJECT

La figure 19, à la page 314 montre un exemple de page utilisant la syntaxe OBJECT pour demander que l'utilisateur se connecte à l'aide de données Information Card.

```
<html>
<head>
<title>Welcome to Fabrikam</title>
</head>
<body>
<img src='fabrikam.jpg'/>
<form name="ctl00" id="ctl00" method="post"
action="https://www.fabrikam.com/InfoCard-Browser/Main.aspx">
<center>
<img src='infocard.bmp' onClick='ctl00.submit()'/>
<input type="submit" name="InfoCardSignin" value="Log in"
id="InfoCardSignin" />
</center>
<OBJECT type="application/x-informationCard" name="xmlToken">
<PARAM Name="tokenType"
Value="urn:oasis:names:tc:SAML:1.0:assertion">
<PARAM Name="issuer" Value=
"http://schemas.xmlsoap.org/ws/2005/05/identity/issuer/self">
<PARAM Name="requiredClaims" Value=
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname">
</OBJECT>
</form>
</body>
</html>
```

Figure 19. Exemple de syntaxe OBJECT

Veuillez noter le type OBJECT `application/x-informationCard`. Lorsque l'utilisateur sélectionne une carte, le jeton de sécurité résultant est inclus dans la réponse (POST) sous forme de valeur `xmlToken`. Les paramètres OBJECT d'Information Card sont utilisés pour encoder les informations WSSecurityPolicy requises dans le code HTML.

Dans cet exemple, la partie de confiance demande un jeton SAML 1.0 auprès d'un fournisseur d'identité délivré à titre autonome, en fournissant les réclamations requises `emailaddress`, `givenname` et `surname`.

Remarque : Vous pouvez omettre l'émetteur afin d'indiquer que *n'importe quel* émetteur Information Card disponible dans le navigateur de l'utilisateur est acceptable.

Syntaxe XHTML

La syntaxe XHTML est la suivante :

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:ic>
<head>
<title>Welcome to Fabrikam</title>
</head>
<body>
<img src='fabrikam.jpg'/>
<form name="ctl00" id="ctl00" method="post"
action="https://www.fabrikam.com/InfoCard-Browser/Main.aspx">
<ic:informationCard name='xmlToken'
style='behavior:url(#default#informationCard)'
issuer="http://schemas.xmlsoap.org/ws/2005/05/identity/issuer/self"
tokenType="urn:oasis:names:tc:SAML:1.0:assertion">
<ic:add claimType=
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
optional="false" />
<ic:add claimType=
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"
optional="false" />
<ic:add claimType=
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
optional="false" />
</ic:informationCard>
<center>
<input type="submit" name="InfoCardSignin" value="Log in"
id="InfoCardSignin" />
</center>
</form>
</body>
</html>
```

Figure 20. Exemple de syntaxe XHTML InfoCard

Paramètres d'appel du sélecteur d'identité

Les paramètres contenus dans les objets Information Card OBJECT et XHTML servent à encoder les informations dans le code HTML. Dans les cas où un sélecteur d'identité est employé dans le contexte de services Web, ces informations sont fournies en tant qu'informations WS-SecurityPolicy via WSMetadataExchange.

La liste suivante indique les paramètres pris en charge par la norme Information Card pour l'appel du sélecteur d'identité.

Remarque : Tous les paramètres sont facultatifs. Aucun d'eux n'est obligatoire.

issuer Ce paramètre détermine l'adresse URL du service STS (Security Token Service) à partir duquel a lieu l'obtention d'un jeton. Lorsque ce paramètre est omis, aucun service STS spécifique n'est requis. La valeur spécialehttp://schemas.xmlsoap.org/ws/2005/05/identity/issuer/self indique que le jeton provient d'un fournisseur d'identité autonome.

Remarque : Ce paramètre n'est pas pris en charge par Tivoli Federated Identity Manager.

issuerPolicy
Ce paramètre spécifie l'adresse URL d'un noeud final à partir duquel l'entité WS-SecurityPolicy peut être extraite via WS-MetadataExchange. S'il est omis, la valeur prise en compte est <issuer>/mex. Ce noeud final doit utiliser HTTPS.

Remarque : Ce paramètre n'est pas pris en charge par Tivoli Federated Identity Manager.

tokenType
Ce paramètre spécifie le type du jeton demandé auprès du service STS, sous forme d'URI. Vous pouvez omettre le paramètre dans les cas suivants :

- lorsque le service STS et le point de contact du site Web sont préalablement convenus du type de jeton à fournir, ou
- si le site Web est prêt à accepter *n'importe quel* type de jeton.

requiredClaims
Ce paramètre spécifie les types de réclamations qui doivent être fournis par l'identité. Si ce paramètre est omis, aucune réclamation n'est exigée. La valeur de `requiredClaims` est une liste d'URI délimitée par des espaces dont chaque entrée spécifie un type de réclamation requis.

optionalClaims
Ce paramètre spécifie les types de réclamations facultatives qui peuvent être fournis par l'identité. Si ce paramètre est omis, il n'existe aucune réclamation optionnelle. La valeur de `optionalClaims` est une liste d'URI délimitée par des espaces dont chaque entrée spécifie un type de réclamation pouvant être soumis à titre facultatif.

privacyURL
Ce paramètre détermine l'adresse URL des règles de confidentialité lisibles par l'utilisateur et applicables au site, le cas échéant.

privacyVersion
Ce paramètre spécifie la version des règles de confidentialité. Il doit s'agir d'une valeur supérieure à 0 si un paramètre `privacyUrl` est spécifié. En cas de modification de cette valeur, l'utilisateur en est averti et est autorisé à consulter les modifications apportées aux règles de confidentialité.

Exemple de page de connexion WebSEAL

La figure suivante est un exemple de page WebSEAL login.html modifiée, dans laquelle les balises OBJECT sont indiquées en **gras**.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<!-- Copyright (C) 2000 Tivoli Systems, Inc. -->
<!-- Copyright (C) 1999 IBM Corporation -->
<!-- Copyright (C) 1998 Dascom, Inc. -->
<!-- All Rights Reserved. -->
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>Connexion Access Manager pour l'e-business</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000">
<B>Connexion Access Manager pour l'e-business (www-default---2)</B>
<BR>
%ERROR%
<BR><BR>
<!--- DO NOT TRANSLATE OR MODIFY any part of the hidden parameter(s) --->

<!---
  The following block of code provides users with a warning message
  if they do not have cookies configured on their browsers.
  If this environment does not use cookies to maintain login sessions,
  simply remove or comment out the block below.
--->

<!--- BEGIN Cookie check block --->
<!---
<! ..... edited from this example for brevity ....
<!--- END Cookie check block --->

<BR>
   <form name="ctl00" id="ctl00" method="post"
        action="https://example.com:443/FIM/sps/infocard/login">
      <center>
          <input type="submit" name="InfoCardSignin" value="Log in"
          id="InfoCardSignin" />
      </center>
      <OBJECT type="application/x-informationCard" id="oCard" name="xmlToken">
        <PARAM Name="tokenType" Value="urn:oasis:names:tc:SAML:1.0:assertion">
        <PARAM Name="issuer" Value=
            "http://schemas.xmlsoap.org/ws/2005/05/identity/issuer/self">
        <PARAM Name="requiredClaims" Value=
"http://schemas.xmlsoap.org/ws/2005/05/identity/
   claims/privatepersonalidentifier">
        <PARAM Name="optionalClaims" Value=
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
">      </OBJECT>
    </form>
   </BODY>
</HTML>
```

Figure 21. Exemple de page de connexion WebSEAL avec des balises OBJECT

Configuration requise pour Information Card

Vous devez configurer la configuration requise pour Information Card avant de pouvoir créer une fédération.

Exigences relatives à WebSphere version 6.1

Tivoli Federated Identity Manager prend en charge Information Card sur WebSphere Application Server 6.1. Information Card n'est pas pris en charge sous WebSphere 6.0. Information Card repose sur l'algorithme de chiffrement **rsa-oaep-mgf1p** pour la protection des clés. Cet algorithme est pris en charge par WebSphere 6.1, mais n'est pas disponible sous WebSphere 6.0.

Tivoli Federated Identity Manager nécessite l'application d'un groupe de correctifs pour WebSphere Application Server 6.1. Consultez les exigences relatives à la configuration logicielle et matérielle requise dans le centre de documentation de Tivoli Federated Identity Manager, en fonction du niveau de groupe de correctifs requis.

Mise à jour des règles de cryptographie pour Information Card

Les algorithmes de chiffrement employés par Information Card nécessitent un puissant support de bibliothèques cryptographiques. Ceci implique la nécessité de remplacer les fichiers de règles de sécurité Java par défaut, local_policy.jar et US_export_policy.jar.

Pourquoi et quand exécuter cette tâche

L'utilisation de la technologie de chiffrement est contrôlée par la législation des Etats-Unis. Les SDK IBM Java comprennent des fichiers de règles de juridiction strictes, mais limitées. Pour pouvoir déployer Information Card avec Tivoli Federated Identity Manager, vous devez vous procurer les fichiers de règles JCE (Java Cryptography Extension) de juridiction illimitée.

Pour consulter les informations de sécurité relatives aux kits de développement de logiciels IBM Java, accédez à l'adresse URL suivante :

`http://www.ibm.com/developerworks/java/jdk/security/index.html`

Procédure

1. Mettez à jour WebSphere à l'aide de fichiers de règles JCE (Java Cryptography Extension) non limitées. Accès : http://www.ibm.com/developerworks/java/jdk/security/index.html.
2. Sélectionnez le lien vers le SDK qui correspond à votre environnement, par exemple, pour Java 1.5, le SDK est J2SE 5.0. Une page contenant l'en-tête Security Information (Informations de sécurité) apparaît.
3. Sélectionnez le lien suivant : **IBM SDK Policy Files**.

   Remarque : Lorsque vous cliquez sur ce lien, vous êtes redirigé vers le fichier de règles contenu dans le kit SDK compatible avec votre version de Java. Il est à noter, toutefois, que le numéro de version du kit SDK n'est pas nécessairement le même que celui de la version de Java utilisée. Par exemple, pour Java 1.5, vous pouvez être redirigé vers le kit SDK 1.4.
4. Vous serez invité à vous connecter à l'aide de votre ID utilisateur IBM et de votre mot de passe. Si vous ne disposez pas d'un ID utilisateur et d'un mot de passe IBM, vous devez vous inscrire. Suivez le lien d'inscription figurant sur la page de connexion.
5. Connectez-vous.
6. A l'invite, sélectionnez le fichier .zip correspondant à la version de Java que vous utilisez. Cliquez ensuite sur **Continuer** pour démarrer le téléchargement.
7. Décomprimez le fichier .zip. Les fichiers JAR sont les suivants :

- local_policy.jar
- US_export_policy.jar

8. Placez les fichiers dans le répertoire suivant :
   rep_installation_composant_exec_Java/jre/lib/security

   Par exemple, il se peut que le composant d'exécution Java ait été installé dans le cadre de la version imbriquée de WebSphere Application Server. Dans ce cas, le répertoire peut être le suivant :
   /opt/IBM/FIM/*ewas*/java/jre/lib/security

Exigences liées à Information Card pour le service d'alias

Le service d'alias doit être configuré si des cartes gérées appuyées par une authentification sur la base de droits d'accès émis de façon autonome doivent être utilisées.

Clé de déchiffrement provenant d'un serveur point de contact

La configuration de la carte d'information nécessite la spécification d'une clé destinée au décryptage des messages au sein de la fédération. Le déchiffrement est un processus obligatoire.

Cela signifie qu'un alias de la clé de déchiffrement doit être ajouté dans le fichier de clés Tivoli Federated Identity Manager. La clé doit être la clé privée utilisée par le serveur point de contact. L'importation de cette clé doit être effectuée à l'aide du service de clés de Tivoli Federated Identity Manager.

Cela signifie qu'un alias de la clé de déchiffrement doit être ajouté dans le fichier de clés Tivoli Federated Identity Manager. La clé peut provenir de n'importe quel site Web qui présente au navigateur compatible Information la page HTML comportant les balises OBJECT requises. Le site peut être le serveur point de contact, mais cela n'est pas une condition obligatoire. L'adresse URL générée dans la page balisée doit *obligatoirement* utiliser le protocole SSL.

Par exemple :
`https://pointofcontact.example.com/FIM`

La clé SSL utilisée pour l'adresse URL doit être importée dans un fichier de clés Tivoli Federated Identity Manager pour la partie de confiance.

Remarque : En cas de modification de la clé SSL ou de sa mise à jour pour les besoins du site Web ou du point de contact, l'administrateur doit également mettre à jour le fichier de clés Tivoli Federated Identity Manager en vue de prendre en compte la nouvelle clé SSL. Cette opération peut également impliquer des modifications de la configuration afin de mettre à jour l'alias du fichier de clés.

Exigences de synchronisation temporelle pour Information Card

Le succès d'un déploiement Information Card dépend de la synchronisation temporelle entre les systèmes.

Les exigences suivantes doivent être satisfaites pour parvenir à un déploiement réussi :

- Lorsque la méthode d'authentification UsernameToken est configurée pour une fédération, l'heure doit être synchronisée entre le fournisseur d'identité et les systèmes des parties de confiance.
- Lors de l'utilisation de la méthode d'authentification reposant sur des droits d'accès auto-générés pour l'obtention d'une carte gérée, le système du navigateur (c'est-à-dire celui qui héberge le navigateur contenant les fonctionnalités Information Card) doit également être synchronisé temporellement.
- Lorsqu'une carte auto-générée est utilisée pour la connexion auprès de la partie de confiance, le système du navigateur (c'est-à-dire celui qui héberge le navigateur contenant les fonctionnalités Information Card) doit également être synchronisé temporellement.

La synchronisation temporelle requise peut être définie via la propriété **clock skew** pour chaque fédération Information Card. Pour modifier cette propriété à partir du panneau de propriétés du partenaire de fédération, vous pouvez utiliser la console d'administration de Tivoli Federated Identity Manager.

Mappage d'identité pour Information Card

La prise en charge de Tivoli Federated Identity Manager pour les fournisseurs d'identité Information Card s'appuie sur une chaîne d'accréditation qui contient des modules destinés à accomplir des actions standard de validation, de mappage et d'envoi.

Fournisseur d'identité

L'opération de validation est appliquée à un jeton d'authentification envoyé par le sélecteur d'identité afin de représenter l'utilisateur. Le jeton est soit un nom d'utilisateur, soit une assertion SAML. L'assertion SAML est employée lors de l'authentification sur la base de droits d'accès émis de façon autonome.

Le module de mappage peut correspondre à l'un des éléments suivants :

- Module de mappage XSLT
- Module Tivoli Directory Integrator
- Module de mappage Java développé sur mesure

Tivoli Directory Integrator est généralement employé comme module de mappage associé à Information Card. Dans les déploiements Information Card, l'un des objectifs principaux de la chaîne d'accréditation est de permettre l'identification des valeurs d e réclamation et leur définition dans les données STSUU. Les valeurs des réclamations peuvent provenir de sources de données externes, telles qu'un registre LDAP.

Le module Tivoli Directory Integrator peut, par exemple, convertir les entrées LDAP d'un utilisateur en valeurs de réclamations correspondantes, telles que définies dans le schéma spécifié par Microsoft.

Les modules Tivoli Directory Integrator peuvent également servir à combiner aisément des valeurs de réclamation issues de différentes sources. Certaines valeurs de réclamation peuvent par exemple provenir d'un registre LDAP, tandis que d'autres proviennent de sources, telles que des bases de données, code Java ou JavaScript, ou encore d'autres services Web.

La sortie du module de mappage est utilisée pour produire un jeton SAML 1.1 en mode *émission*.

Partie de confiance

La chaîne d'accréditation applicable pour une fédération de partie de confiance comprend :

- Un module de jeton SAML 1.1 en mode validation.
- Le module de mappage par défaut.
- Le module de jeton IVCred en mode émission

L'assistant de fédération invite l'administrateur à spécifier les règles de mappage d'identité à l'aide de XSLT, suivant le déploiement. Les règles de mappage utilisent les attributs des assertions ou les informations contenues dans les réclamations afin de déterminer l'identité d'utilisation.

Les modules de jeton SAML créent les attributs STSUniversalUser destinés à chaque attribut de l'assertion SAML. Le nom, l'espace de nom et la valeur de chaque attribut SAML servent à définir le nom, le type et la valeur de STSUniversalUser/Attribute.

Formulaire de configuration du fournisseur d'identité

Tivoli Federated Identity Manager comprend un assistant qui vous guidera tout au long de la configuration des fédérations Information Card. L'assistant vous invite à renseigner les propriétés nécessaires pour votre déploiement.

Ce formulaire décrit les invites. Ce formulaire vous permet de planifier vos propriétés et vous pouvez vous y référer lors de l'exécution de l'assistant.

Nom de la fédération
Chaîne arbitraire choisie pour dénommer cette fédération. Par exemple :
`infocard-idp`

Rôle de la fédération
Sélectionnez un fournisseur d'identité.

Nom de la société
L'assistant demande de spécifier les informations de contact. La zone Nom de la société est requise. Ce nom peut correspondre à n'importe quelle chaîne de caractères. Les autres zones sont facultatives.

Protocole de la fédération
Sélectionnez Information Card.

Serveur point de contact
Serveur qui agit en tant que point de contact initial pour les requêtes entrantes. Par exemple :
`https://pointofcontact.example.com/FIM`

Remarque : Pour le support d'Information Card, le serveur point de contact doit utiliser le protocole SSL (Secure Socket Layer). L'adresse URL spécifiée doit être du type `https://`.

Identificateur de clé de noeud final SSL

L'assistant de configuration vous invite à indiquer la clé à utiliser pour les opérations de déchiffrement dans la fédération. Il doit s'agir de la clé utilisée par le serveur point de contact pour les connexions SSL.

L'assistant vous demande cette clé dans le panneau **Paramètres de configuration Infocard**. Pour spécifier la clé, sélectionnez le fichier de clés, puis la clé.

Remarque : Vous devez importer cette clé du serveur point de contact vers le fichier de clés Tivoli Federated Identity Manager avant de configurer la fédération.

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé.

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Clé à sélectionner
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Option d'authentification

Vous devez choisir l'une des options d'authentification suivantes :

- Authentification avec une carte auto-émise
- Authentification avec un nom d'utilisateur et un mot de passe

L'option par défaut est l'authentification avec une carte auto-émise.

Le choix de l'option d'authentification détermine la valeur par défaut de la propriété **Fichier modèle de téléchargement de carte**.

Fichier modèle de téléchargement de carte
Il s'agit d'un fichier modèle HTML qui vous invite à saisir les paramètres requis pour émettre une carte d'information gérée. Des valeurs par défaut sont fournies par l'assistant de configuration. Vous pouvez appliquer les valeurs par défaut, sauf si vous avez modifié et renommé les fichiers modèle.

- Lorsque l'authentification avec une carte auto-émise est sélectionnée, la valeur par défaut est :

  `/infocard/getcard_sss.html`
- Lorsque l'authentification avec un nom d'utilisateur et un mot de passe est sélectionnée, la valeur par défaut est :

  `/infocard/getcard_ut.html`

Fichier modèle de carte d'information
Il s'agit d'un fichier modèle HTML contenant la carte d'information à vous renvoyer. fichier par défaut :

`/infocard/infocard_template.xml`

Fichier image de carte d'information
Fichier image à utiliser pour la carte d'information. Celui-ci doit se trouver dans le répertoire de l'environnement local en cours. La valeur par défaut est identique pour les deux options d'authentification. fichier par défaut :

`/infocard/fim_infocard.gif`

Expiration de la carte
cette propriété spécifie le nombre de jours de validité de la carte d'information à partir de la date d'émission. La valeur par défaut est identique pour les deux options d'authentification. Valeur par défaut :

365

Options de mappage d'identité

Vous devez sélectionner l'une des options suivantes :

- Utiliser XSL pour le mappage d'identité

  Sélectionnez cette option pour utiliser une règle de mappage XSLT. vous devez indiquer le nom d'un fichier contenant les règles de mappage d'identité. Tivoli Federated Identity Manager fournit un exemple de fichier de règles de mappage d'identité destiné aux fédérations de fournisseurs d'identité Information Card :

  `/répertoire_installation/examples/ip_infocard.xsl`

- Utiliser Tivoli Directory Integrator pour le mappage

  Sélectionnez cette option lorsque vous avez préalablement configuré une chaîne d'assemblage Tivoli Directory Integrator pour le mappage d'identité requis par votre fédération Information Card.

- Utiliser une instance de modèle de mappage personnalisé

  Sélectionnez cette option lorsque vous avez écrit et déployé un module de service d'accréditation personnalisé pour le mappage d'identité requis par votre fédération Information Card.

Tableau 106. Formulaire pour les propriétés d'une fédération de fournisseurs d'identité

Propriété	**Votre valeur**
Nom de la fédération	
Rôle	Fournisseur d'identité
Nom de l'entreprise	
Protocole de fédération	Carte d'information
Serveur point de contact	
Identificateur de clé de noeud final SSL : fichier de clés	
Identificateur de clé de noeud final SSL : mot de passe de fichier de clés	
Identificateur de clé de noeud final SSL : clé à sélectionner	
Option d'authentification	
Fichier modèle de téléchargement de carte	
Fichier modèle de carte d'information	
Fichier image de carte d'information	Valeur par défaut : /infocard/fim_infocard.gif
Expiration de la carte	Valeur par défaut : 365 jours
Options de mappage d'identité	Sélectionnez l'une des options suivantes : • Utiliser XSL pour le mappage d'identité • Utiliser Tivoli Directory Integrator pour le mappage • Utiliser une instance de modèle de mappage personnalisé

Tableau 106. Formulaire pour les propriétés d'une fédération de fournisseurs d'identité (suite)

Propriété	Votre valeur
Fichier de règles de mappage d'identité	Si vous utilisez XSL pour le mappage d'identité, spécifiez le nom de fichier de règle de mappage suivant :
Module de mappage personnalisé	Si vous utilisez un module de mappage personnalisé, notez le nom du module :

Formulaire de configuration de la partie de confiance

Tivoli Federated Identity Manager comprend un assistant qui vous guidera tout au long de la configuration des fédérations Information Card. L'assistant vous invite à renseigner les propriétés nécessaires pour votre déploiement.

Ce formulaire décrit les invites. Ce formulaire vous permet de planifier vos propriétés et vous pouvez vous y référer lors de l'exécution de l'assistant.

Nom de la fédération
Chaîne arbitraire choisie pour dénommer cette fédération. Par exemple, `infocard-rp`.

Rôle de la fédération
Sélectionnez un fournisseur de services. Cette valeur est requise par la partie de confiance.

Nom de la société
L'assistant demande de spécifier les informations de contact. La zone Nom de la société est requise. Ce nom peut correspondre à n'importe quelle chaîne de caractères. Les autres zones sont facultatives.

Protocole de la fédération
Sélectionnez Information Card.

Serveur point de contact
Serveur qui agit en tant que point de contact initial pour les requêtes entrantes. Par exemple :

```
https://pointofcontact.example.com/FIM
```

Remarque : Pour le support d'Information Card, le serveur point de contact doit utiliser le protocole SSL (Secure Socket Layer). L'adresse URL spécifiée doit être du type `https://`.

Déchiffrement

L'assistant de configuration vous invite à indiquer la clé à utiliser pour les opérations de déchiffrement dans la fédération. Il doit s'agir de la clé utilisée par le serveur point de contact pour les connexions SSL.

L'assistant vous demande cette clé dans le panneau **Déchiffrement**. Pour spécifier la clé, sélectionnez le fichier de clés, puis la clé.

Remarque : Vous devez importer cette clé du serveur point de contact vers le fichier de clés Tivoli Federated Identity Manager avant de configurer la fédération.

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé.

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Clé à sélectionner
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Partenaire standard
L'assistant vous invite à sélectionner une option :

- Ajouter un partenaire pouvant gérer n'importe quel fournisseur d'identité

  Cette option est l'option par défaut.

  La sélection de cette option entraîne l'ajout automatique d'un partenaire. Cette configuration de partenaire peut accepter tout fournisseur d'identité pour les cartes d'information, y compris un fournisseur à émission automatique.
- Ajouter un partenaire pouvant gérer le fournisseur d'identité à émission automatique

  La sélection de cette option entraîne l'ajout automatique d'un partenaire. Ce partenaire Tivoli Federated Identity Manager accepte uniquement les cartes personnelles émises par le fournisseur à émission automatique intégré au navigateur.
- Ne pas ajouter de partenaire standard

  La sélection de cette option entraîne l'absence d'ajout de partenaires. L'administrateur doit ajouter explicitement des partenaires via l'assistant Ajout de partenaire de la console Tivoli Federated Identity Manager.

Options de mappage d'identité
Vous devez sélectionner l'une des options suivantes :

- Utiliser XSL pour le mappage d'identité

  Sélectionnez cette option pour utiliser une règle de mappage XSLT. vous devez indiquer le nom d'un fichier contenant les règles de mappage d'identité. Tivoli Federated Identity Manager fournit un exemple de fichier de règles de mappage d'identité destiné aux fédérations de fournisseurs d'identité Information Card :

  répertoire_installation/examples/rp_infocard.xsl
- Utiliser Tivoli Directory Integrator pour le mappage

  Sélectionnez cette option lorsque vous avez préalablement configuré une chaîne d'assemblage Tivoli Directory Integrator pour le mappage d'identité requis par votre fédération Information Card.
- Utiliser une instance de modèle de mappage personnalisé

  Sélectionnez cette option lorsque vous avez écrit et déployé un module de service d'accréditation personnalisé pour le mappage d'identité requis par votre fédération Information Card.

Tableau 107. Formulaire pour les propriétés d'une fédération de parties de confiance

Propriété	Votre valeur
Nom de la fédération	
Rôle	Fournisseur de services
Nom de l'entreprise	
Protocole de fédération	Carte d'information
Serveur point de contact	
Déchiffrement : fichier de clés	
Déchiffrement : mot de passe du fichier de clés	
Déchiffrement : clé à sélectionner	
Partenaire standard	Option par défaut : **Ajouter un partenaire pouvant gérer n'importe quel fournisseur d'identité.** Votre option :
Options de mappage d'identité	Sélectionnez l'une des options suivantes : • Utiliser XSL pour le mappage d'identité • Utiliser Tivoli Directory Integrator pour le mappage • Utiliser une instance de modèle de mappage personnalisé
Fichier de règles de mappage d'identité	Si vous utilisez XSL pour le mappage d'identité, spécifiez le nom de fichier de règle de mappage suivant :
Module de mappage personnalisé	Si vous utilisez un module de mappage personnalisé, notez le nom du module :

Formulaire de partenaire géré

Lors de la création d'une fédération pour un fournisseur d'identité, un partenaire est créé automatiquement.

Après avoir créé une fédération pour une partie de confiance, vous pouvez choisir l'une des nombreuses options de configuration d'un partenaire. Lorsque vous choisissez de ne pas ajouter de partenaire standard, vous pouvez créer un partenaire pour la fédération ultérieurement. Lorsque vous procédez de la sorte, vous devez fournir certaines valeurs de configuration.

La console de Tivoli Federated Identity Manager offre un assistant qui vous guidera tout au long de cette procédure.

Nom de la société du fournisseur d'identité
Informations de contact.

Emetteur de jeton de sécurité
Cette valeur sert à définir le paramètre protocolID et l'adresse URL du

noeud final dans le fichier etc/feds.xml, ainsi que la zone Emetteur dans la configuration du mappage de chaîne STS. Par exemple :

```
https://example.com
```

Décalage d'horloge maximum autorisé entre les hôtes (en secondes)
Il s'agit de la valeur maximale d'écart d'horloge autorisée entre l'hôte de la partie de confiance et celui du fournisseur d'identité. La valeur de décalage d'horloge est utilisée durant la validation de la période de validité de l'assertion.

La valeur par défaut est de 60 secondes.

Valider des signatures sur les jetons de la carte d'information
Vous pouvez cocher cette case pour indiquer que les jetons de sécurité entrants doivent être signés. Lorsque vous sélectionnez cette option, vous devez, à l'aide des propriétés de configuration complémentaires, spécifier la clé publique qui sert à valider la signature numérique.

Type de clé de validation de signature
Vous devez sélectionner l'une des options suivantes :

- Clé publique fournie par le KeyInfo dans la signature du jeton de la carte d'information

  Vous pouvez choisir cette option si vous ne souhaitez pas distribuer ni mettre à jour la clé publique et que vous avez seulement besoin de vous assurer que l'intégrité du jeton est maintenue.
- Clé publique d'un fichier de clés

  Cette clé publique doit avoir été préalablement obtenue auprès du fournisseur de l'identité gérée, puis importée dans un fichier de clés Tivoli Federated Identity Manager à l'aide des services de clés.

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Clé à sélectionner
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Tableau 108. Formulaire des propriétés de configuration du partenaire géré

Propriété	Votre valeur
Nom de la société du fournisseur d'identité	
Emetteur de jeton de sécurité	
Décalage d'horloge maximum autorisé entre les hôtes (en secondes)	
Valider des signatures sur les jetons de la carte d'information	

Tableau 108. Formulaire des propriétés de configuration du partenaire géré (suite)

Propriété	**Votre valeur**
Type de clé de validation de signature	En cas de validation de signatures, sélectionnez une option : • Clé publique fournie par le KeyInfo dans la signature du jeton de la carte d'information • Clé publique d'un fichier de clés
Fichier de clés	*Lors de l'utilisation d'une* ***clé publique provenant d'un fichier de clés*** :
Mot de passe du fichier de clés	
Clé à sélectionner	

Chapitre 22. Planification d'une fédération Information Card

Vérification des dépendances liées à Information Card

Vérifiez que les exigences de création d'une fédération Information Card sont respectées.

Avant de commencer

Avant d'utiliser l'assistant de création de fédération, assurez-vous que les conditions relatives aux dépendances Information Card sont satisfaites.

Procédure

1. Vérifiez que vous effectuez l'installation sur WebSphere Application Server 6.1. Les versions antérieures ne sont pas prises en charge. Voir «Exigences relatives à WebSphere version 6.1», à la page 318.
2. Vérifiez que vous possédez les bibliothèques de chiffrement correctes. Voir «Mise à jour des règles de cryptographie pour Information Card», à la page 318.
3. Déterminez su vous devez configurer le service d'alias. Voir «Exigences liées à Information Card pour le service d'alias», à la page 319.
4. Assurez-vous que vous avez importé la clé de chiffrement pour le serveur point de contact Cette clé doit être importée dans le service de clés de Tivoli Federated Identity Manager.

Planification d'une fédération Infocard

Pour configurer une fédération à connexion unique Infocard, vous devez créer la fédération, y ajouter votre partenaire, puis fournir à celui-ci les informations de configurations issues de votre nouvelle fédération.

Avant de commencer

Assurez-vous que vous avez préparé les informations de configuration avant de créer la fédération au moyen de l'assistant. Les activités de planification sont décrites dans une série de rubriques du présent manuel. Voir Chapitre 4, «Présentation des tâches de configuration pour la connexion unique fédérée», à la page 37.

Pourquoi et quand exécuter cette tâche

Pour utiliser l'assistant de fédération afin de créer et configurer une fédération Infocard, procédez comme suit :

Procédure

1. Connectez-vous à la Integrated Solutions Console.
2. Cliquez sur **Tivoli Federated Identity Manager → Configurer la connexion unique fédérée → Fédérations**. Les portlets Domaine en cours et Fédérations s'ouvrent. Le portlet Fédérations affiche plusieurs boutons d'action.
3. Cliquez sur **Créer**. L'assistant de fédération démarre. L'assistant affiche une série de panneaux de configuration.

4. Utilisez votre formulaire complété afin d'indiquer des valeurs dans chaque panneau. Entrez les valeurs demandées. Pour obtenir des informations sur des zones spécifiques, affichez l'aide en ligne.
   a. La première série de panneaux vous demande d'indiquer les paramètres relatifs au nom, au rôle et serveur point de contact de la fédération.
   b. Puis, le panneau de configuration Infocard vous invite à indiquer les valeurs requises pour un fournisseur d'identité ou une partie de confiance Infocard.
   c. La dernière série de panneaux vous invite à indiquer les paramètres de configuration du mappage d'identité.

   Lorsque vous avez terminé d'entrer les paramètres de configuration, le panneau Récapitulatif s'affiche.
5. Cliquez sur **Suivant** pour passer au panneau suivant. Si vous avez besoin de revenir en arrière pour ajuster un paramètre de configuration, cliquez sur **Précédent**.
6. Vérifiez que les paramètres de configuration sont corrects.
7. Cliquez sur **Terminer**. Le portlet Création de fédération terminée s'affiche.

Configuration de WebSEAL en tant que serveur point de contact pour une fédération Information Card

Si vous envisagez d'utiliser WebSEAL en tant que serveur point de contact, vous devez le configurer pour la fédération Information Card.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Ces instructions ont pour hypothèse que le profil du point de contact WebSEAL est activé.

Pourquoi et quand exécuter cette tâche

Le portlet Création de fédération terminée comporte un bouton qui vous permet d'obtenir un Tivoli Federated Identity Manager utilitaire de configuration. Vous devez obtenir cet outil, puis l'exécuter. Pour configurer WebSEAL en tant que serveur point de contact, procédez comme suit :

Procédure

1. Une fois la fédération créée, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour recharger vos modifications.

   Remarque : La console de gestion vous offre la possibilité d'ajouter immédiatement un partenaire, mais pour cette configuration initiale de la fédération, vous devez d'abord exécuter d'autres tâches.
2. Cliquez sur **Terminé** pour revenir au panneau Fédérations.
3. Cliquez sur **Télécharger l'outil de configuration Tivoli Access Manager**.

4. Enregistrez l'outil de configuration sur le système de fichiers de l'ordinateur hébergeant le serveur WebSEAL.
5. Démarrez l'outil de configuration depuis une ligne de commande. La syntaxe est la suivante :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf
   ```

   Remarque : Si la norme FIPS (Federal Information Processing Standards) est activée pour votre environnement, une fabrique de connexions sécurisées doit être indiquée. Par exemple :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf -sslfactory TLS
   ```

   Vous aurez besoin de l'ID (par défaut : sec_master) et du mot de passe de l'utilisateur d'administration Tivoli Access Manager. L'utilitaire configure les noeuds finals sur le serveur WebSEAL, crée une jonction WebSEAL, relie les listes de contrôle d'accès adaptées et active les méthodes d'authentification adéquates.

Exemple

Par exemple, lorsque vous avez mis le fichier tfimcfg.jar dans le répertoire `/tmp` et que le nom de l'instance WebSEAL est `default`, la commande est la suivante :

```
java -jar /tmp/tfimcfg.jar -action tamconfig -cfgfile webseald-default
```

Pour plus d'informations, voir Annexe A, «Référence de tfimcfg», à la page 827.

Configuration de WebSphere en tant que serveur point de contact

Tivoli Federated Identity Manager est configuré par défaut pour utiliser Tivoli Access Manager WebSEAL en tant que serveur point de contact. Pour configurer WebSphere en tant que serveur point de contact, vous devez procéder à une modification de la configuration.

Procédure

1. Connectez-vous à la console d'administration.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Sélectionnez **WebSphere**.
4. Cliquez sur **Activer**.

Résultats

Le serveur WebSphere est désormais configuré en tant que point de contact.

Spécification d'un index de personnes

Un index de personnes est une collection comprenant plusieurs séries d'attributs, accessible à un utilisateur ou à un fournisseur d'identité. L'utilisateur peut spécifier les attributs qui décrivent une personne. Un utilisateur peut par exemple avoir des données professionnelles telles que son adresse électronique et son numéro de téléphone professionnels, et des données personnels relatives à son domicile, comme son adresse électronique et son numéro de téléphone à titre privé. Ces données personnelles peuvent porter respectivement les noms *work* et *home*.

Lorsqu'un utilisateur télécharge une carte gérée, il peut souhaiter associer celle-ci à des données personnelles particulières. Cela permet au fournisseur d'identité de déterminer quel ensemble d'attributs personnels doit être utilisé pour alimenter le jeton lorsqu'un jeton de connexion unique est requis pour la carte.

L'utilisation des données personnelles est permise grâce à un paramètre optionnel de zone de formulaire appelé `userdata`. Ce paramètre peut être inclus dans les pages des modèles getcard_ut.html et getcard_sss.html.

Bien que n'étant pas incluse dans les fichiers modèles fournis, cette zone d'entrée est néanmoins prise en charge.

Lorsque ce paramètre est spécifié, une macro de remplacement appelée @USERDATA@ peut être alimentée dans le fichier infocard_template.html. La macro @USERDATA@ est utilisée dans la section CardId du fichier infocard_template.html.

Le fichier infocard_template.html par défaut contient le modèle de macro suivant pour la section CardId :

```
<InformationCardReference>
  <CardId>@IPSTS@/@UUID@</CardId>
  <CardVersion>1</CardVersion>
</InformationCardReference>
```

Lorsque les administrateurs souhaitent utiliser @USERDATA@, il leur est suggéré d'utiliser le modèle de macro suivant :

```
<InformationCardReference>
  <CardId>@IPSTS@/@UUID@/@USERDATA@</CardId>
  <CardVersion>1</CardVersion>
</InformationCardReference>
```

Les informations CardId font partie des données RST envoyées par le sélecteur d'identité à Tivoli Federated Identity Manager lors de la demande d'un jeton de connexion unique. Les informations sur les règles de mappage permettent de lire et différencier les données personnelles à utiliser lorsque vous disposez des informations CardId.

Si ce modèle est appliqué, un utilisateur est invité à spécifier son index de données personnelles lors du téléchargement d'une carte gérée, et la carte est liée à un individu donné. Un utilisateur a la possibilité de télécharger différentes cartes pour chacun des ensembles de données personnelles qu'il a spécifiés auprès du fournisseur d'identité. La règle de mappage contenue dans la chaîne d'accréditation STS peut lire les données CardID, ainsi que l'index des données personnelles. Elle peut également spécifier, dans le jeton d'identité d'exécution, les attributs provenant des données personnelles valides.

Chapitre 23. Références pour Information Card

Macros de remplacement dans le fichier XML infocard_template

Fournit une liste des macros de remplacement et leurs utilisations dans le fichier XML infocard_template.

Les macros de remplacement pour le fichier infocard_template.xml sont les suivantes :

@IPSTS@
Adresse URL désignant le noeud final du fournisseur d'identité pour la fédération.

@IPMEX@
Adresse URL désignant le noeud final d'échange des métadonnées du fournisseur d'identité pour la carte gérée. Il est à noter que l'adresse URL est spécifique au type d'authentification utilisé.

@UUID@
Cette macro est remplacée par un UUID (identifiant d'utilisateur) généré de façon aléatoire. cette valeur permet de garantir que l'identité de la carte est unique.

@USERDATA@
cette macro n'est pas incluse dans le fichier par défaut. Vous pouvez l'ajouter au conteneur CardId lorsque vous souhaitez spécifier des attributs. Cette macro est utile lorsque des utilisateurs membres de votre déploiement ont de multiples identités. Les utilisateurs peuvent fournir des attributs destinés à identifier la personne à prendre en compte.

@CARDNAME@
Nom de la carte tel que spécifié par l'utilisateur dans l'envoi de réponse via le formulaire getcard_ut.html ou getcard_sss.html.

@CARDIMAGE@
Fichier image MIME (Multi-purpose Internet Email Extension) encodé qui est proposé à l'utilisateur par le sélecteur d'identité. Il existe un fichier image pour chaque fédération.

@ISSUETIME@
Heure d'émission de la carte. Le calcul de l'heure a lieu au moment de l'exécution.

@EXPIRETIME@
Heure d'expiration de la carte. L'heure est calculée en additionnant l'heure d'émission et la valeur de *lifetime* de la carte.

@IPCERTIFICATE@
Certificat public encodé en base 64 qui est configuré pour la fédération. Il convient qu'il s'agisse également du certificat public du noeud final SSL pour le serveur point de contact.

@USERCRED@
Bloc de métadonnées relatives aux droits d'accès utilisés par le sélecteur d'identité pour authentifier l'utilisateur auprès du noeud final du fournisseur d'identité (service STS). Les métadonnées proviennent d'un autre fichier modèle, suivant le type d'authentification en vigueur.

La prise en charge d'Information Card par Tivoli Federated Identity Manager s'applique aux deux formes d'authentification :

- Jeton de nom d'utilisateur

  Les métadonnées correspondant aux données d'identification de l'utilisateur sont chargées à partir du fichier modèle inforcard_usercred_usernametoken.xml.

- Droits d'accès auto-émis

  Les métadonnées correspondant aux données d'identification de l'utilisateur sont chargées à partir du fichier modèle infocard_usercred_selfsignedsaml.xml.

@SUPPORTED_TOKENS@
La prise en charge d'Information Card par Tivoli Federated Identity Manager s'applique uniquement au type de jeton SAML 1.1. Deux représentations par défaut sont disponibles.

@SUPPORTED_CLAIMS@
Ensemble des réclamations prises en charge par cette carte. Ces valeurs sont héritées du formulaire envoyé par l'utilisateur dans le fichier getcard_*.html. Les valeurs doivent être présentées selon le format XML imposé par les spécifications Information Card.

Réclamations Information Card

Fournit une liste des types de réclamation, en mentionnant l'URI et la description de chacun d'eux.

Les types de réclamations sont récapitulés ici pour des raisons pratiques, mais il convient que les utilisateurs consultent la liste officielle dans le schéma référencé.

Remarque : La prise en charge d'Information Card dans Tivoli Federated Identity Manager ne se limite pas à cette série de réclamations.

Prénom
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname

Nom ou prénom préférentiel d'un individu. La spécification RFC 2256 indique que l'attribut `givenName` sert à contenir la partie du nom d'un individu qui ne correspond ni à son nom de famille, ni à son deuxième prénom.

Nom de famille
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname

Nom de famille d'un individu. La spécification RFC 2256 utilise `sn` et indique qu'il s'agit de l'attribut de nom X.500 contenant le nom de famille d'un individu.

Adresse électronique
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress

Adresse préférentielle indiquée pour la zone d'adresse électronique `To:` servant à l'envoi à destination du sujet, généralement spécifiée sous la forme `<utilisateur>@<domaine>`.

Le terme `mail` est utilisé par inetOrgPerson dans la spécification RFC1274, qui indique que ce type d'attribut spécifie un attribut d'adresse de messagerie électronique selon la syntaxe définie dans la spécification RFC 822.

Adresse
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/streetaddress

Composant concernant l'adresse postale dans les informations d'adresse du sujet.

La spécification RFC 2256 utilise le terme `street` et indique que cet attribut contient l'adresse physique de l'objet auquel l'entrée correspond, telle qu'une adresse de livraison. Son contenu est arbitraire, mais se présente généralement sous la forme d'une boîte postale ou d'un numéro d'appartement ou de maison suivi d'un nom de rue. Exemple : 303 rue des Jonquilles.

Nom de localité ou ville
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/locality

Composant concernant la localité dans les informations d'adresse du sujet. La spécification RFC 2256 utilise le terme `l` et indique que cet attribut contient le nom de la localité, par exemple une ville, un pays ou une autre région géographique. Exemple : Marseille.

Etat ou province
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/stateorprovince

Abréviation désignant le nom de l'état ou de la province dans les informations d'adresse du sujet. La spécification RFC 2256 utilise le terme `st` et indique que cet attribut contient le nom l'état ou de la province. Il convient d'harmoniser les valeurs à l'échelle nationale et, si des abréviations connues existent comme c'est le cas pour les abréviations à deux lettres est Etats américains, il convient de les privilégier par rapport aux noms entiers.

L'abréviation `TX`, par exemple, est utilisée pour désigner l'Etat du Texas.

Code postal
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode

Composant concernant le code postal dans les informations d'adresse du sujet. La spécification X.500 (2001) emploie le terme `postalCode` et indique que le type d'attribut du code postal spécifie le code postal de l'objet nommé. Si cette valeur d'attribut est présente, elle fait partie de l'adresse postale de objet, c'est-à-dire le code postal aux Etats-Unis ou dans d'autres pays.

Pays http://schemas.xmlsoap.org/ws/2005/05/identity/claims/country

Pays de résidence d'un sujet. La spécification RFC 2256 utilise le terme `c` et indique que cet attribut contient le code de pays à deux chiffres ISO 3166.

Numéro de téléphone
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/homephone

Numéro de téléphone principal ou secondaire d'un individu. Le terme `homePhone` est employé pour inetOrgPerson dans la spécification RFC 1274, qui indique que ce type d'attribut sert à spécifier le numéro de téléphone d'un domicile rattaché à une personne.

Il convient que les valeurs d'attribut respectent le format en vigueur pour les numéros téléphoniques internationaux. Exemple : +99 99 999 9999.

Numéro de téléphone secondaire ou de travail
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/otherphone

Numéro de téléphone secondaire ou du lieu de travail d'un individu. La spécification X.500 (2001) emploie le terme `telephoneNumber` et indique que ce type d'attribut spécifie un numéro téléphonique professionnel ou scolaire associé à une personne.

Il convient que les valeurs d'attribut respectent le format en vigueur pour les numéros téléphoniques internationaux. Exemple : +99 99 999 9999.

Numéro de téléphone mobile
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/mobilephone

Numéro de téléphone mobile d'un individu. Le terme `mobile` est employé pour inetOrgPerson dans la spécification RFC 1274, qui indique que ce type d'attribut sert à spécifier le numéro de téléphone mobile rattaché à une personne.

Il convient que les valeurs d'attribut respectent le format en vigueur pour les numéros téléphoniques internationaux. Exemple : +99 99 999 9999.

Date de naissance
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth

Date de naissance d'un sujet, suivant le format autorisé par le type de données `xs:date`.

Sexe http://schemas.xmlsoap.org/ws/2005/05/identity/claims/gender

Sexe d'un individu. La valeur doit être l'une des chaînes suivantes :

0 Non spécifié

1 Homme

2 Femme

L'utilisation de ces valeurs rend les paramètres indépendants de la langue.

Identificateur personnel privé
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier

Identificateur de type PPID (Private Personal Identifier) qui identifie le sujet auprès d'une partie de confiance. Le terme **privé** signifie ici que l'identificateur du sujet est spécifique par rapport à une partie de confiance donnée et qu'il n'est par conséquent connu que de celle-ci (d'où le terme *privé*). Le PPID d'un individu auprès d'une partie de confiance unique ne peut être corrélé à celui du même individu auprès d'une autre partie de confiance.

Page Web
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/webpage

Page Web d'un individu exprimée sous forme d'adresse URL.

Propriétés de fédération pour les fournisseurs d'identité

Lorsque vous créez une fédération Information Card pour un fournisseur d'identité, l'assistant de configuration assigne automatiquement les valeurs par défaut à certaines propriétés.

Vous ne pouvez pas modifier les propriétés de fédération des fournisseurs d'identité pendant la configuration initiale. Vous pouvez toutefois modifier ces propriétés une fois la configuration initiale terminée.

Identification de la fédération

Nom de la fédération
Chaîne arbitraire choisie pour dénommer cette fédération.

Par exemple, pour un fournisseur d'identité géré :

```
infocard-idp
```

Nom de la société
L'assistant demande de spécifier les informations de contact. Il s'agit de la seule zone obligatoire. La valeur peut correspondre à n'importe quelle chaîne de caractères.

Propriétés de connexion unique

ID fournisseur
Identificateur unique permettant au fournisseur de se faire reconnaître par le fournisseur de services. Cette valeur est constituée du protocole et du nom d'hôte de l'URL du fournisseur d'identité. Elle peut également contenir un numéro de port. Par exemple, pour une fédération nommée infocard_fed :

```
https://idp.example.com/sps/infocard_fed/infocard
```

Noeud final de téléchargement de la carte
Noeud final permettant de créer et de télécharger une carte gérée. L'extension du fichier doit être .crd. La valeur par défaut est la suivante :

ID_fournisseur/getCard.crd

Noeud final d'échange de métadonnées
Noeud final utilisé par les sélecteurs d'identité pour demander les métadonnées relatives au service de jeton de sécurité (STS) du fournisseur d'identité. La valeur par défaut est la suivante :

ID_fournisseur/mex

Noeud final du service de jeton de sécurité
Noeud final utilisé par les sélecteurs d'identité pour demander les jetons de sécurité dans le cadre d'une authentification de carte d'information. La valeur par défaut est la suivante :

ID_fournisseur/sts

Noeud final de gestion d'alias
Noeud final utilisé pour gérer l'association ou le lien entre une carte auto-émise et le compte Tivoli Federated Identity Manager de l'utilisateur. Le lien est établi lorsque l'utilisateur télécharge une carte gérée à l'aide du mécanisme d'authentification **carte auto-émise**. Ce noeud final peut être utilisé pour vérifier et supprimer ce lien. La valeur par défaut est la suivante :

ID_fournisseur/alias

Cette propriété n'est pas utilisée si vous avez sélectionné l'option d'authentification pour le nom d'utilisateur et le mot de passe.

Option d'authentification

Vous pouvez remplacer l'option d'authentification par l'une des options suivantes :

- Authentification avec une carte auto-émise
- Authentification avec un nom d'utilisateur et un mot de passe

Fichier modèle de téléchargement de carte
Cette propriété est un fichier modèle HTML qui invite l'utilisateur à saisir les paramètres d'entrée requis pour émettre une carte d'information gérée.

- Lorsque l'authentification avec une carte auto-émise est sélectionnée, la valeur par défaut est :

  `/infocard/getcard_sss.html`
- Lorsque l'authentification avec un nom d'utilisateur et un mot de passe est sélectionnée, la valeur par défaut est :

  `/infocard/getcard_ut.html`

Fichier modèle de carte d'information
Cette propriété est un fichier modèle HTML contenant la carte d'information qui vous est renvoyée. fichier par défaut :

`/infocard/infocard_template.xml`

La valeur par défaut est identique pour les deux options d'authentification.

Fichier image de carte d'information
Cette propriété est le fichier image à utiliser pour la carte d'information. Celui-ci doit se trouver dans le répertoire de l'environnement local en cours. La valeur par défaut est identique pour les deux options d'authentification. fichier par défaut :

`/infocard/fim_infocard.gif`

La valeur par défaut est identique pour les deux options d'authentification.

Modèle de carte de métadonnées
Nom du fichier à utiliser en tant que modèle pour les métadonnées ou la carte d'information. Le fichier par défaut est :

`/infocard/metadata_template.xml.`

Règles des métadonnées de droits d'accès SAML d'auto-signature
Nom du fichier de règles à utiliser pour les métadonnées de droits d'accès SAML d'auto-signature. Le fichier par défaut est :

`/infocard/metadata_policy_selfsignedsaml.xml`

Cette zone s'affiche si vous avez sélectionné **Authentification avec une carte auto-émise**.

Règles des métadonnées de droits d'accès du nom d'utilisateur
Nom du fichier de règles à utiliser pour les métadonnées de droits d'accès du nom d'utilisateur. Le fichier par défaut est :

`/infocard/metadata_policy_usernametoken.xml`

Cette zone s'affiche si vous avez sélectionné **Authentification avec un nom d'utilisateur et un mot de passe**.

Expiration de la carte
cette propriété spécifie le nombre de jours de validité de la carte d'information à partir de la date d'émission. La valeur par défaut est identique pour les deux options d'authentification. Valeur par défaut :

`365`

Identificateur de clé de noeud final SSL

Remarque : Il s'agit de la clé que vous devez importer depuis le serveur point de contact dans le fichier de clés Tivoli Federated Identity Manager avant de configurer les fédérations.

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé.

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Liste des clés
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Identificateur de la clé de signature Information Card

Paire de clés publique/privée utilisée par les cartes d'information récemment émises.

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé.

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Liste des clés
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Propriétés du module de jeton

Lors de la configuration initiale de la fédération Information Card, la chaîne d'accréditation est générée et configurée automatiquement. La chaîne d'accréditation contient les modules d'accréditation qui nécessitent une configuration. Les propriétés de cette section peuvent être modifiées.

Activer l'utilisation unique des assertions
Utiliser l'assertion une seule fois et ne pas la mettre en mémoire cache pour l'utiliser ultérieurement. Cette propriété est activée par défaut.

Elle est utilisée uniquement avec l'authentification par carte auto-émise.

Sauter la validation du mot de passe
Ne pas effectuer la validation du mot de passe pour le jeton Username. L'option par défaut est désélectionnée, ce qui signifie que la validation du mot de passe a lieu.

Cette propriété est utilisée uniquement avec l'authentification par nom d'utilisateur et mot de passe.

Durée de validité (en secondes) d'une assertion avant sa date d'émission
Valeur par défaut : 60 secondes Aucune valeur minimale ou maximale n'est appliquée.

Durée de validité (en secondes) de l'assertion après émission.
Valeur par défaut : 60 secondes Aucune valeur minimale ou maximale n'est appliquée.

Propriétés de mappage d'identité

Les propriétés de mappage d'identité sont identiques pour tous les protocoles pris en charge parTivoli Federated Identity Manager.

Instance du module de mappage d'identité
Cette valeur correspond à votre choix lors de la configuration initiale.

Modifier l'instance du module de mappage d'identité
Appelle le panneau Options de mappage d'identité. Le panneau Options de mappage d'identité permet de sélectionner une transformation XSL, Tivoli Directory Integrator, ou une instance de module de mappage.

Modification des propriétés en cours
Appelle un autre panneau dans lequel vous pouvez modifier les propriétés :

- Si la fédération utilise une transformation XSL, sélectionnez cette option pour ouvrir le panneau Règle de mappage d'identité. Vous pouvez modifier ou supprimer la règle de mappage d'identité dans ce panneau.
- Si la fédération utilise un module de mappage personnalisé, sélectionnez cette option pour ouvrir un panneau dans lequel vous pouvez afficher ou modifier les propriétés d'instance de mappage personnalisé.

Propriétés de fédération pour les parties de confiance

Fournit une liste de valeurs et leurs descriptions pour les propriétés de fédération d'une partie de confiance.

Identification de la fédération

Nom de la fédération
Chaîne arbitraire choisie pour dénommer cette fédération.

Par exemple, pour un fournisseur de confiance :

```
infocard-rp
```

Nom de la société
L'assistant demande de spécifier les informations de contact. Il s'agit de la seule zone obligatoire. Ce nom peut correspondre à n'importe quelle chaîne de caractères.

Propriétés de connexion unique

ID fournisseur
Identificateur unique permettant au fournisseur de se faire reconnaître par le fournisseur de services. Cette valeur est constituée du protocole et du nom d'hôte de l'URL du fournisseur d'identité. Elle peut également contenir un numéro de port. Par exemple, pour une fédération nommée infocard_fed :

```
https://rp.example.com/sps/infocard_fed/infocard
```

URL d'authentification
URL à laquelle l'utilisateur envoie les demandes d'authentification. Cette valeur ne peut pas être modifiée sur le panneau Propriétés. Par exemple, pour une fédération nommée infocard_fed, l'URL d'authentification serait :

```
https://idp.example.com/sps/infocard_fed/infocard/login
```

Propriétés de la clé de déchiffrement

Clé à utiliser pour déchiffrer les jetons entrants. Il est à noter que celle-ci doit être identique à la clé utilisée pour SSL par le point de contact (par exemple, WebSEAL).

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé.

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Liste des clés
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Propriétés de mappage d'identité

Les propriétés de mappage d'identité sont identiques pour tous les protocoles pris en charge parTivoli Federated Identity Manager.

Instance du module de mappage d'identité
Cette valeur correspond à votre choix lors de la configuration initiale.

Modifier l'instance du module de mappage d'identité
Appelle le panneau Options de mappage d'identité. Le panneau Options de mappage d'identité permet de sélectionner une transformation XSL, Tivoli Directory Integrator, ou une instance de module de mappage.

Modification des propriétés en cours
Appelle un autre panneau permettant de modifier des propriétés :

- Si la fédération utilise une transformation XSL, ce bouton appelle le panneau Règle de mappage d'identité. Ce panneau permet de modifier ou de supprimer la règle de mappage d'identité.
- Si la fédération utilise un module de mappage personnalisé, ce bouton appelle un panneau permettant d'afficher ou de modifier les propriétés d'instance de mappage personnalisé.

Caractéristiques des partenaires des fournisseurs d'identité dans les fédérations de parties de confiance

Fournit une liste de valeurs de propriété et leurs descriptions pour les partenaires de fournisseur d'identité dans les fédérations de parties de confiance.

Identification de la fédération

Nom du membre de la fédération
Fédération à laquelle ce partenaire a été ajouté. Vous ne pouvez pas modifier cette propriété.

Exemple : le fournisseur d'identité est dorénavant un partenaire de la fédération de fournisseurs de confiance :

```
infocard-rp
```

Rôle du partenaire
Fournisseur d'identité. Vous ne pouvez pas modifier cette propriété.

Etat La fenêtre des propriétés du partenaire permet de savoir si un partenaire est activé ou désactivé. Les partenaires doivent être activés pour pouvoir participer à une fédération.

- Si l'état d'un partenaire est désactivé, cliquez sur Activer pour activer ce partenaire.
- Si l'état du partenaire est activé, cliquez sur Désactiver pour le désactiver.

Nom de la société du fournisseur d'identité
Nom de la société du partenaire. Ce nom peut correspondre à n'importe quelle chaîne de caractères. Vous pouvez utiliser des espaces. Cette zone est obligatoire.

Adresse URL de la société
URL de la société du partenaire. Cette zone est facultative. Par exemple :

```
http://www.example.com
```

Personne à contacter
Informations de contact optionnelles pour l'administrateur. Vous pouvez également utiliser la zone Autres informations si nécessaire.

Propriétés des jetons

Emetteur de jeton de sécurité
Spécifiez l'URI (Uniform Resource Identifier) de l'émetteur unique du fournisseur d'identité. Cette valeur doit être utilisée dans l'élément saml:Issuer de saml:Assertion. Voici un exemple :

```
https://example.com
```

Vous pouvez entrer une astérisque (*) pour indiquer qu'un fournisseur d'identité est acceptable.

Décalage d'horloge maximum autorisé entre les hôtes (en secondes)
Spécifiez une valeur entière qui indique le décalage d'horloge maximum autorisé, en secondes, entre l'hôte de la partie de confiance et l'hôte du fournisseur d'identité. Vous devez spécifier une valeur minimale de zéro secondes dans cette zone. La valeur par défaut est 60. Cette zone s'affiche uniquement si la fédération utilise l'option Authentification avec une carte auto-émise.

Propriétés de la clé de validation de signature

Valider des signatures sur les jetons Infocard
Lorsque cette option est sélectionnée, cela indique que vous devez signer les jetons de la carte d'information puis indiquer le type de clé publique à utiliser pour valider la signature numérique. Décochez cette case pour désactiver la validation de signature. Cette case est cochée par défaut.

Type de clé de validation de signature

- **Clé publique fournie par le KeyInfo dans la signature du jeton de la carte d'information**

  Choisissez d'utiliser la clé publique fournie par le KeyInfo dans la signature du jeton de la carte d'information. Il s'agit de la sélection par défaut.
- **Clé publique d'un fichier de clés**

Sélectionnez une clé publique dans un fichier de clés. Lorsque vous sélectionnez cette option, vous devez sélectionner le fichier de clés et la clé.

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé.

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Liste des clés
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Propriétés de mappage d'identité

Les propriétés de mappage d'identité sont identiques pour tous les protocoles pris en charge parTivoli Federated Identity Manager.

Instance du module de mappage d'identité
Cette valeur correspond à votre choix lors de la configuration initiale.

Modifier l'instance du module de mappage d'identité
Appelle le panneau Options de mappage d'identité. Le panneau Options de mappage d'identité permet de sélectionner une transformation XSL, Tivoli Directory Integrator, ou une instance de module de mappage.

Modification des propriétés en cours
Appelle un autre panneau permettant de modifier des propriétés :

- Si la fédération utilise une transformation XSL, ce bouton appelle le panneau Règle de mappage d'identité. Ce panneau permet de modifier ou de supprimer la règle de mappage d'identité.
- Si la fédération utilise un module de mappage personnalisé, ce bouton appelle un panneau permettant d'afficher ou de modifier les propriétés d'instance de mappage personnalisé.

Caractéristiques des partenaires de confiance pour les fédérations de fournisseurs d'identité

Fournit une liste de valeurs de propriété et les descriptions associées pour les partenaires de confiance des fédérations de fournisseurs d'identité.

Identification de la fédération

Nom du membre de la fédération
Fédération à laquelle ce partenaire a été ajouté. Vous ne pouvez pas modifier cette propriété.

Exemple : la partie de confiance est dorénavant un partenaire de la fédération de fournisseurs d'identité :

```
infocard-idp
```

Rôle du partenaire
Fournisseur de services (partie de confiance). Vous ne pouvez pas modifier cette propriété.

Etat La fenêtre des propriétés du partenaire permet de savoir si un partenaire

est activé ou désactivé. Les partenaires doivent être activés pour pouvoir participer à une fédération. Cette propriété n'est pas modifiable, car elle s'applique à toutes les parties de confiance.

Nom de la société du fournisseur de services
Cette valeur indique que cette configuration partenaire est appliquée à tous les partenaires.

Par exemple, pour une fédération de fournisseurs d'identité dénommée infocard-idp, la valeur par défaut est :

```
toutes les parties de confiance de infocard-idp
```

Adresse URL de la société
URL de la société du partenaire. Cette zone est facultative. Par exemple :

```
http://www.example.com
```

Personne à contacter
Informations de contact optionnelles pour l'administrateur. Vous pouvez également utiliser la zone Autres informations si nécessaire.

Paramètres de configuration globale Infocard

Décalage d'horloge maximum autorisé entre les hôtes (en secondes)
Spécifiez une valeur entière qui indique le décalage d'horloge maximum autorisé, en secondes, entre l'hôte de la partie de confiance et l'hôte du fournisseur d'identité. Vous devez spécifier une valeur minimale de zéro secondes dans cette zone. La valeur par défaut est 60. Cette zone s'affiche uniquement si la fédération utilise l'option Authentification avec une carte auto-émise.

Sélectionner la clé de signature des assertions
Indiquez la clé à utiliser pour la signature des assertions SAML.

Fichier de clés
Fichier de clés Tivoli Federated Identity Manager contenant la clé.

Par exemple : Tivoli Federated Identity Manager fournit un fichier de clés appelé DefaultKeystore.

Mot de passe du fichier de clés
Mot de passe requis pour accéder au fichier de clés spécifié.

Liste des clés
L'assistant présente la liste des alias de clés (ou noms) stockés dans le fichier de clés. Vous devez sélectionner la clé à utiliser.

Propriétés des jetons

Inclure les types d'attribut suivants
Indiquez les types d'attributs à inclure dans l'assertion. L'astérisque (*), qui est le paramètre par défaut, indique que tous les types d'attributs spécifiés dans le fichier de mappage personnalisé ou par le module de mappage personnalisé seront inclus dans l'assertion. Pour spécifier un ou plusieurs types d'attributs individuellement, tapez chaque type d'attribut dans la case. Utilisez && pour séparer plusieurs types d'attributs.

Inclure l'élément InclusiveNamespaces à la canonicalisation de l'assertion lors de la création de la signature
Sélectionnez cette option pour inclure l'élément InclusiveNamespaces à la canonicalisation de l'assertion lors de la création de la signature. Par défaut, cette option est désélectionnée.

Inclure les données de certificat X509 dans l'élément KeyInfo de la signature
Sélectionnez cette option pour utiliser les données de certificat X509 dans l'élément KeyInfo de la signature. Par défaut, cette option est sélectionnée.

Inclure les données de clé publique dans l'élément KeyInfo de la signature
Sélectionnez cette option pour utiliser les données de clé publique (clé publique RSA/DSA X509) dans l'élément KeyInfo de la signature. Par défaut, cette option est sélectionnée. L'élément KeyInfo contient des informations sur la clé nécessaires pour valider la signature.

Propriétés de mappage d'identité

Les propriétés de mappage d'identité sont identiques pour tous les protocoles pris en charge parTivoli Federated Identity Manager.

Instance du module de mappage d'identité
Cette valeur correspond à votre choix lors de la configuration initiale.

Modifier l'instance du module de mappage d'identité
Appelle le panneau Options de mappage d'identité. Le panneau Options de mappage d'identité permet de sélectionner une transformation XSL, Tivoli Directory Integrator, ou une instance de module de mappage.

Modification des propriétés en cours
Appelle un autre panneau permettant de modifier des propriétés :

- Si la fédération utilise une transformation XSL, ce bouton appelle le panneau Règle de mappage d'identité. Ce panneau permet de modifier ou de supprimer la règle de mappage d'identité.
- Si la fédération utilise un module de mappage personnalisé, ce bouton appelle un panneau permettant d'afficher ou de modifier les propriétés d'instance de mappage personnalisé.

Chapitre 24. Présentation de la planification sous OpenID

Tivoli Federated Identity Manager prend en charge les connexions uniques via le protocole OpenID.

Cette présentation décrit l'implémentation sous Tivoli Federated Identity Manager d'OpenID. Les informations de la présentation permettent à un administrateur de déployer et configurer des fédérations de connexion unique.

Les spécifications OpenID font référence à une partie nommée *fournisseur OpenID ou fournisseur d'identité*, qui a pour rôle de certifier qu'un utilisateur possède une adresse URL d'identité particulière. Une partie de confiance ou un *consommateur* reçoit ces informations de la part du fournisseur d'identité. Sous Tivoli Federated Identity Manager, le terme de *fournisseur d'identité* correspond directement à la notion de *fournisseur OpenID ou fournisseur d'identité* d'OpenID. Le *consommateur* OpenID correspond bien au concept de fournisseur de service de Tivoli Federated Identity Manager.

La prise en charge par Tivoli Federated Identity Manager de l'authentification OpenID permet de définir tous les modes pour les messages OpenID :

associate
Mode permettant d'établir un secret partagé avec le consommateur.

checkid_immediate
Mode permettant d'exécuter un contrôle non bloquant si un utilisateur détient l'URL de l'identificateur réclamé.

checkid_setup
Mode permettant d'exécuter un contrôle si un utilisateur détient l'URL de l'identificateur réclamé. Le contrôle peut, le cas échéant, inclure une interaction avec l'utilisateur.

check_authentication
Mode permettant de déterminer la validité d'une signature de message. Ce mode est généralement sélectionné pour les consommateurs muets ou dépourvus d'état.

Remarque : Pour obtenir une description complète des spécifications OpenID, consultez le site Web d'OpenID :

http://www.openid.net

Prise en charge d'OpenID 1.1 et 2.0

OpenID 1.1 et OpenID 2.0 sont tous les deux pris en charge.

Adresses URL d'ID OpenID

Une adresse URL d'identité OpenID est une identité numérique conçue pour authentifier les utilisateurs et leur accorder l'accès aux services.

URL d'identité avec un point de contact WebSEAL

Prenons les exemples suivants de valeurs pour créer une URL d'identité avec un point de contact WebSEAL :

- Une fédération de fournisseur d'identité appelée `openidfedip`
- Un serveur Tivoli Federated Identity Manager dont le serveur point de contact est WebSEAL et portant le nom d'hôte `webseal.example.com`.
- Une identité d'utilisateur (dans le cas présent, un utilisateur Tivoli Access Manager) intitulée `john`.

L'URL d'identité OpenID peut être n'importe quelle adresse URL répondant aux exigences suivantes :

- Pouvoir être résolue sur votre site Web. Dans notre exemple :
  - Lancez l'adresse `http(s)://webseal.example.com`
  - Ou bien, si vous utiliser des entrées de DNS génériques et un certificat de site pour `*.example.com`, il peut s'agir d'une valeur du type `http(s)://john.example.com`
- Il doit contenir un identificateur unique à l'utilisateur. En général, cet identificateur est votre identité d'utilisateur au niveau du fournisseur d'identité. Toutefois, il peut s'agir d'un alias généré pour des raisons de confidentialité.
- Elle doit correspondre à une expression régulière que vous avez configurée pour votre fédération de fournisseur d'identité OpenID.
- Le point final de fournisseur d'identité OpenID doit être reconnaissable à l'aide de la reconnaissance Yadis ou HTML à partir de votre URL d'identité tel que décrit dans les spécifications OpenID.

URL d'identité avec un point de contact WebSphere

Prenons les exemples de valeurs suivants pour créer l'URL d'identité avec un point de contact WebSphere :

- Une fédération de fournisseur d'identité appelée `openidfedip`
- Un serveur Tivoli Federated Identity Manager sur lequel le serveur de point de contact est WebSphere, avec le nom d'hôte `poc.example.com`
- Une identité d'utilisateur intitulée `john`

Le exigences qui s'appliquent à l'adresse URL sont les mêmes que dans le premier exemple. La figure 22 illustre un exemple de code lorsqu'un serveur de point de contact WebSphere est déployé et que la reconnaissance HTML est utilisée.

```
<html>
<head>
<link rel="openid.server"
href="https://poc.example.com/sps/openidfedip/openid/sso">
<link rel="openid2.provider"
href="https://poc.example.com/sps/openidfedip/openid/sso">
</head>
...
</html>
```

Figure 22. Exemple de code pour le renvoi d'un pointeur vers votre serveur OpenID à partir de votre URL d'identité à l'aide de la reconnaissance HTML

Remarque : Vous pouvez également utiliser la reconnaissance Yadis pour renvoyer un pointeur à votre serveur OpenID à partir de votre URL d'identité.

Exemple d'URL d'identité

Lorsque vous configurez une fédération pour OpenID, définissez une expression régulière pour les URL d'identité. Un moyen simple de vous assurer qu'un lien au noeud final de votre serveur OpenID est renvoyé par votre URL d'identité consiste à vérifier les points suivants :

1. Assurez-vous que la page d'URL d'identité est une page non protégée.
2. Intégrez le lien au serveur OpenID dans le formulaire de connexion du serveur point de contact.

   Le serveur point de contact est généralement une instance WebSEAL ou WebSphere.

La restriction imposée par cette méthode est qu'une seule fédération de fournisseur d'identité OpenID peut résider sur l'ordinateur. Cette restriction ne donne théoriquement lieu à aucun problème et correspond à un déploiement typique d'OpenID.

Exemples :

- Par exemple, lorsque l'expression régulière configurée est :

```
http://webseal.example.com/@ID@
```

un exemple d'URL d'identité est le suivant :

```
http://webseal.example.com/john
```

Cette méthode de configuration simple ne nécessite aucune interaction avec l'utilisateur pour pouvoir établir une URL d'identité. Tivoli Federated Identity Manager détermine si un utilisateur possède cette URL d'identité en :

1. Remplaçant la macro `@ID@` dans l'expression régulière configurée par le nom d'utilisateur de Tivoli Federated Identity Manager, et en
2. Vérifiant qu'une correspondance exacte existe avec l'URL d'identité réclamée par l'utilisateur dans la demande de connexion unique.

- Un autre exemple est celui dans lequel le déploiement :
  - Utilise un certificat de site avec `CN=*.example.com`
  - Utilise une entrée DNS générique correspondant à `*.example.com` pour un site protégé par WebSEAL.
  - Permet à l'utilisateur de détenir au choix des URL OpenID `http` ou `https`.

  Pour cet exemple, les URL d'identité suivantes sont valides :
  - `john.example.com`
  - `http://john.example.com`
  - `https://john.example.com`

  Remarque :
  - Lorsque le protocole n'est pas spécifié, comme dans le premier exemple, le protocole utilisé est HTTP.

    L'expression régulière configurée pour cette fédération inclut dans ce cas un nom d'hôte générique et une prise en charge multi-protocole. Par exemple :

    ```
    http[s]?://@ID@.example.com
    ```

    La macro `@ID@` est mappée à un nom d'utilisateur.

– Dans certains environnements d'application, l'utilisation d'une barre oblique de fin peut être souhaitée dans les modèles d'URL d'identité :

  ```
  webseal.example.com/john/
  ```

  Certaines applications ajoutent une barre oblique de fin (/) lors de la normalisation de l'entrée utilisateur. Une non concordance se produit lorsqu'une barre oblique de fin est ajoutée par l'application mais pas spécifiée pour l'URL d'identité. L'accès est refusé.

  Pour ces environnements, assurez-vous que l'expression régulière configurée inclut la barre oblique de fin. Par exemple :

  ```
  http://webseal.example.com/@ID@/
  ```

Générateur d'identificateur personnel privé

Dans certains scénarios d'authentification, vous pouvez conserver la confidentialité de l'utilisateur en masquant son identité à la partie de confiance. En outre, vous pouvez vouloir le même utilisateur pour vous connecter à deux parties de confiance différentes à l'aide d'identificateurs demandés.

Le générateur d'identificateur personnel privé (PPID) crée l'identificateur. Les identités d'utilisateur de la partie de confiance n'entrent ainsi pas en collision car ils utilisent des identificateurs demandés différents.

Ce type de scénario d'authentification est appelé *identité dirigée*. L'identité dirigée exige que l'utilisateur lance la connexion au niveau de la partie de confiance à l'aide de l'identificateur de fournisseur d'identité. Par exemple `https://example.ibm.com`

Selon la configuration, le fournisseur OpenID génère un identificateur pour l'utilisateur d'une partie de confiance spécifique. Un générateur d'identificateur personnel privé (PPID) crée l'identificateur. Le fournisseur OpenID génère un identificateur distinct pour chaque partie de confiance à laquelle le même utilisateur s'authentifie.

La création d'identificateurs demandés différents empêche le partage des informations entre les parties de confiance. Cette fonction protège également efficacement l'identité de l'utilisateur.

L'utilisation de la fonction PPIG implique que le fournisseur OpenID informe son serveur d'informations de noeud final à l'aide du document Extensible Resource Descriptor Sequence (XRDS). Le document XRDS est requis.

Un utilisateur peut uniquement se connecter à une partie de confiance à l'aide d'un identificateur de fournisseur d'identité inclus au document XRDS. Le document XRDS représente le seul moyen, pour la partie de confiance, de distinguer l'identificateur demandé d'un utilisateur et l'identificateur du fournisseur d'identité.

Un plug-in fournit un générateur PPIG dans l'implémentation de fournisseur d'identité. Le plug-in fournit plusieurs implémentations de générateur standard. Un administrateur peut utiliser le plug-in pour écrire et intégrer un IDGenerator personnalisé. Cette fonction détermine comment générer l'identité d'un identificateur demandé pour un utilisateur particulier au niveau d'une partie de confiance donnée.

Lorsqu'un identificateur de fournisseur d'identité est utilisé au niveau de la partie de confiance pour lancer l'authentification, le fournisseur d'identité est responsable de la génération de l'identificateur demandé pour l'utilisateur. Tivoli Federated Identity Manager génère un identificateur demandé à l'aide d'un simple modèle d'adresse URL configuré. L'adresse URL doit contenir la macro `@ID@`. La valeur de `@ID@` est générée par le générateur PPID.

Par exemple, la configuration par défaut est la suivante :

```
https://myidp.com/@ID@
```

Les valeurs IDGenerators suivantes peuvent être utilisées pour remplacer la macro `@ID@` de l'URL d'identité :

- Générateur d'ID de nom d'utilisateur
- Générateur d'ID de hachage
- Générateur d'ID de service d'alias

Générateur d'ID de nom d'utilisateur

Lorsque le générateur d'ID de nom d'utilisateur est utilisé, un nom d'utilisateur est renvoyé en portion `@ID@` de l'expression pour les URL d'identité.

Par exemple, lorsque l'expression d'URL d'identité est la suivante :

```
http://webseal.example.com/@ID@
```

un exemple d'URL d'identité est le suivant :

```
http://webseal.example.com/john
```

Ce paramètre représente le comportement par défaut de Tivoli Federated Identity Manager.

Générateur d'ID de hachage

Le générateur d'ID de hachage remplace la valeur `@ID@` par une valeur de hachage sha256. Cette valeur de hachage représente une combinaison de l'ID de fédération en cours, du nom d'utilisateur et de la racine de la partie de confiance.

L'avantage du hachage est que le nom d'utilisateur n'est pas exposé sur chaque site visité pour la connexion unique OpenID. La valeur de hachage est rapide à générer, sans recherche externe. ce masquage du nom de compte permet de protéger l'utilisateur du détournement de son identité par des pirates informatiques. L'exposition du nom de compte fournit un point de départ pour les attaques malignes ou pour le verrouillage d'un utilisateur sur un compte.

Par exemple, lorsque l'expression régulière configurée est :

```
http://webseal.example.com/@ID@
```

un exemple d'URL d'identité est le suivant :

```
http://webseal.example.com/
3d0f1d5e9a3a617771608b390b5c7fc1601a3839f161060cbad8e93b98f034c2
```

Générateur d'ID de service d'alias

L'implémentation de service d'alias affecte automatiquement un UUID généré de manière aléatoire pour la valeur `@ID@`.

Lors de la première utilisation, un UUID est généré et stocké dans le service d'alias. La clé de recherche pour l'UUID est basée sur le nom d'utilisateur, l'ID de fédération en cours et la racine de la partie de confiance. Lors des utilisations suivantes, le même UUID est extrait du service d'alias. Cette méthode permet de s'assurer qu'un identificateur cohérent est utilisé pour l'utilisateur au niveau de cette partie de confiance spécifique.

Comme dans le cas du mode de hachage, le nom d'utilisateur n'est pas exposé sur chaque site utilisé pour la connexion unique OpenID.

Par exemple, lorsque l'expression régulière configurée est :

```
http://webseal.example.com/@ID@
```

Voici un exemple d'URL d'identité : `http://webseal.example.com/c84911b2-0124-14f0-991a-a5a8f0e6f99d`

Evitez de réutiliser les identités d'utilisateur pour définir des URL d'identité

Les URL d'identité OpenID ne doivent jamais être réutilisés. Une fois l'adresse URL attribuée à un utilisateur individuel, il convient de ne jamais la réaffecter à un autre utilisateur. Cette règle est importante, car pour tout site Web de consommation sur lequel l'utilisateur d'origine s'est authentifié, il est possible qu'un compte associé à l'adresse URL soit encore existant.

Les exigences concernant la non réutilisation des URL d'identité OpenID doivent être mises en oeuvres par l'environnement de déploiement. Tivoli Federated Identity Manager ne peut pas effectuer de vérification de la réutilisation. L'application des accès des noms d'utilisateurs doit suivre un processus permettant de s'assurer que chaque adresse URL est affectée une seule fois.

Fédérations de fournisseurs d'identité

Les fédérations de fournisseur d'identité OpenID partagent des similarités avec d'autres fédérations de connexion unique prises en charge par Tivoli Federated Identity Manager. Toutefois, les concepts de *fédération* et *partenaires* s'appliquent différemment.

Une différence majeure est qu'un fournisseur d'identité OpenID n'a pas besoin de connaître d'avance la partie consommatrice. La négociation de secret partagé fait partie du protocole et aucune pré-configuration des clés ou des partenaires n'est nécessaire.

Dans OpenID, l'utilisateur est impliqué dans sa décision de faire confiance à des partenaires consommateurs particuliers. Cette décision est prise en examinant l'adresse URL *trust_root* de la page de consentement d'authentification. Ceci implique le concept de fournisseurs de services partenaires.

Remarque : Dans OpenID 2.0, la racine *trust_root* porte le nom de domaine (*realm*).

La configuration de la fédération comprend certaines propriétés de configuration des partenaires, mais celles-ci sont utilisées par les modules de jeton destinés au service de jetons de sécurité (STS).

La dénomination de la fédération OpenID est conforme à la norme Tivoli Federated Identity Manager relative à l'identificateur unique ou ID de protocole `protocolID`. La syntaxe est la suivante :

`https://<nom_hôte:port>/FIM/sps/<nom_fédération>/openid`

Par exemple :

`https://www.example.com/FIM/sps/openidfedip/openid`

Noeud final de connexion unique

Le noeud final de connexion unique correspond à l'URL du serveur OpenID. Il prend en charge les requêtes émises par le consommateur et le navigateur lorsqu'elles sont redirigées par le consommateur. Cette adresse URL nécessite un accès non authentifié, afin que les requêtes émises par les clients consommateurs anonymes puissent être générées selon les modes de message suivants :

- associate
- checkid_immediate
- check_authentication

Lorsqu'une requête checkid_setup est reçue et que l'utilisateur n'a encore jamais reçu accrédité le consommateur, cette adresse URL fournit également l'invite de consentement d'authentification, *consent-to-authenticate*.

Ce noeud final renvoie les résultats de l'authentification aux sites de consommation.

Exemple de noeud final :

`https://webseald.example.com/FIM/sps/openidfedip/openid/sso`

Noeud final d'authentification

Lorsqu'un utilisateur ne s'est pas encore connecté à un fournisseur d'identité, le noeud final de connexion unique redirige le navigateur vers le noeud final d'authentification. L'utilisateur est alors authentifié. Ce noeud final est requis durant les opérations de type 'checkid_setup', lorsque l'utilisateur ne s'est pas encore authentifié auprès du fournisseur d'identité et que la connexion unique est initiée à partir d'un consommateur.

En cas de succès de l'authentification, le noeud final redirige généralement l'utilisateur vers le noeud final de connexion unique pour la permettre la suite du traitement. Cette redirection est commandée par un paramètre de requête au format chaîne. La syntaxe est la suivante :

```
<protocolID>/authn?return=<url>
```

Voici un exemple de chaîne continue unique :

```
https://webseald.example.com/FIM/sps/openidfedip/openid/authn?return=
      https://webseald.example.com/FIM/sps/openidfedip/sso
```

Noeud final de gestion de site

Lorsque le fournisseur d'identité reçoit un message checkid_setup, il demande à l'utilisateur la permission (ou le consentement) de fournir les informations d'authentification et d'attribut relatives à l'utilisateur.

Le fournisseur d'identité utilise un modèle de page et un cookie de navigateur pour cet utilisateur de sorte à mémoriser les préférences utilisateur. Le fournisseur d'identité doit être en mesure d'extraire les préférences sauvegardées pour pouvoir répondre aux messages en mode checkid_setup.

Tivoli Federated Identity Manager sauvegarde les préférences utilisateur au moyen d'un point d'extension de gestionnaire de sites dignes de confiance. Le point d'extension utilise une interface connectable qui permet aux administrateurs de remplace l'implémentation d'extension par défaut par une implémentation personnalisée qui prend par exemple en charge un modèle de stockage côté serveur. Un autre objectif de ce point d'extension est le fait qu'une implémentation personnalisée peut être utilisée pour consentir automatiquement toutes les décisions de confiance dans des environnements d'authentification fermés.

Le fournisseur d'identité utilise ce gestionnaire de sites dignes de confiance pour gérer les sites de consommation sécurisés et non sécurisés. Lorsque le gestionnaire de site demande à l'utilisateur de donner son consentement d'authentification, l'utilisateur peut spécifier des règles applicables au consommateur spécifié, en procédant comme suit :

- Toujours autoriser (Always Allow)
- Autoriser une fois (Allow Once)
- Refuser une fois (Deny Once)
- Toujours refuser (Always Deny)

L'utilisateur peut, par la suite, utiliser ce gestionnaire de site pour accéder aux préférences sauvegardées et les modifier. Les utilisateurs peuvent, en option, supprimer définitivement de la liste un site sécurité ou non sécurisé. Lors d'une telle opération, l'utilisateur est amené à donner son consentement d'authentification dès sa tentative de connexion unique suivante sur ce site consommateur.

Le gestionnaire de site sécurisé mémorise également les attributs facultatifs éventuels demandés par un fournisseur de services (si l'utilisateur ait autorisé le partage de ces attributs).

Le noeud final exécute les tâches suivantes :

1. Utilise un modèle HTML pour inviter l'utilisateur à indiquer la liste définitive de ses sites dignes et non dignes de confiance
2. Permet à l'utilisateur de supprimer des sites dans la liste permanente.

La syntaxe de l'URL du noeud final est la suivante :

```
<protocolID>/sites
```

Par exemple :

```
https://webseald.example.com/FIM/sps/openidfedip/openid/sites
```

Chaînes d'accréditation de fournisseur d'identité

Dans le modèle OpenID, le consommateur peut exiger que des attributs spécifiques soient fournis pour chaque identité d'utilisateur. Tivoli Federated Identity Manager utilise une chaîne du service d'accréditation spécifiée sur le fournisseur d'identité pour obtenir les attributs et les placer dans un jeton XML simple.

Lorsqu'un utilisateur contacte le fournisseur d'identité en présentant une URL d'identité OpenID, le fournisseur d'identité vérifie l'identification de l'utilisateur. Lorsque le fournisseur d'identité fonctionne en mode checkid_immediate ou checkid_setup, le service d'accréditation est démarré afin d'intercepter et fournir les données d'attributs. En outre, le service de confiance est utilisé pour confirmer que toutes les exigences d'authentification PAPE (Provider Authentication Policy Extension) sont satisfaites.

Le fournisseur d'identité utilise une chaîne de modules de services d'accréditation principalement dans le but de permettre l'extraction des valeurs d'attributs obligatoires et facultatives. Le service d'accréditation s'appuie sur le flux de modules standard suivant :

1. Validate (Validation)

   L'opération de validation est exécutée sur un jeton IVCred généré à partir des droits d'authentification de l'utilisateur.
2. Mappage

   Le module de mappage peut appartenir à n'importe lequel des types pris en charge. Lorsque les données d'attributs de l'utilisateur peuvent être extraites du jeton d'entrée IVCred, la définition d'une règle de mappage XSLT constitue souvent une bonne option. Un module de mappage de Tivoli Directory Integrator, ou un module de mappage Java personnalisé, est utile lorsque les données d'attributs doivent être obtenues depuis une source externe.
3. Issue (Emission)

   L'opération d'émission génère un jeton STSUU (Security Token Service Universal User). Le jeton fournit l'ensemble d'attributs requis et facultatifs au service de protocole de connexion unique, avec les informations d'authentification PAPE validées. Cette opération permet au service de générer une réponse de connexion OpenID ou de demander à nouveau une authentification si cela est nécessaire pour satisfaire les autres règles PAPE demandées.

Pour que le module de mappage puisse fournir les attributs obligatoires et facultatifs, il doit connaître la liste de ces attributs. La liste des attributs requis et facultatifs est envoyée au service d'accréditation sous forme de réclamations. Les informations PAPE demandées sont également disponibles à la règle de mappage en informations de réclamations.

La liste des réclamations peut également contenir des données de préférence utilisateur. Un index des identités peut par exemple être inclus dans le formulaire de consentement d'authentification. L'index est extrait via le point d'extension de gestion des consommateurs accrédités, puis inclus dans les réclamations.

```
<fimopenid:OpenIDClaims
    xmlns:fimopenid="urn:ibm:names:ITFIM:openid"
    xmlns:fimpape="urn:ibm:names:ITFIM:openid:PAPE"
    xmlns:fimqs="urn:ibm:names:ITFIM:queryservice"
    ClaimedId="http://specs.openid.net/auth/2.0/identifier_select"
    DiscoveredIdentifier="https://www.myidp.ibm.com/FIM/op/
85da8845-0127-1a04-9a9f-dca9f50a9649"
    IdentityURL="http://specs.openid.net/auth/2.0/identifier_select"
    IsOPIdentifierLogin="true"
    IsRPReturnToValidated="false"
    OPLocalId="http://specs.openid.net/auth/2.0/
identifier_select"
    OpenIDServerURL="https://www.myidp.ibm.com/
FIM/sps/openididp/openid/sso"
    PolicyURL="http://www.ibm.com"
    ReauthCount="0"
    ReturnTo="https://www.myrp.ibm.com/sps/myrp/openid/
loginreturn?nonce=uuid85d96a6f-0127-1f6e-bafb-c3b7deb3ed5d"
    TrustRoot="https://www.myrp.ibm.com/"
    Userdata=""
    Version="http://specs.openid.net/auth/2.0">
  <fimopenid:PrincipalName>shane</fimopenid:PrincipalName>
  <fimqs:RequestedAttributes>
    <fimqs:Attribute name="openid.sreg.email" optional="false" />
    <fimqs:Attribute name="openid.sreg.nickname" optional="true" />
    <fimqs:Attribute name="openid.sreg.fullname" optional="true" />
  </fimqs:RequestedAttributes>
  <fimpape:OpenIDPAPEClaims>
    <fimpape:Attribute name="openid.pape.preferred_auth_levels">
      <fimpape:Value>urn:ibm:names:ITFIM:5.1:accessmanager</fimpape:Value>
    </fimpape:Attribute>
    <fimpape:Attribute name="openid.pape.preferred_auth_policies">
      <fimpape:Value>http://schemas.xmlsoap.org/ws/2005/05/
identity/claims/privatepersonalidentifier</fimpape:Value>
      <fimpape:Value>http://www.idmanagement.gov/schema/2009/05/
icam/openid-trust-level1.pdf</fimpape:Value>
    </fimpape:Attribute>
  </fimpape:OpenIDPAPEClaims>
</fimopenid:OpenIDClaims>
```

Figure 23. Exemples de réclamations durant l'appel du service d'accréditation par le fournisseur d'identité

La figure 23 illustre un exemple de réclamations transmises au service d'accréditation. Notez les réclamations optionnelles contenues dans la liste `RequestedAttributes` :

- `openid.sreg.email`
- `openid.sreg.nickname`
- `openid.sreg.fullname`

Lorsque des attributs sont requis, la valeur facultative est définie sur `false`.

Dans l'exemple, vous pouvez remarquer que la propriété `userdata` est une chaîne vide. Cette chaîne vide indique qu'aucune donnée optionnelle n'a été définie durant la phase de consentement pour l'authentification. Ces données (ainsi que leur consultation dans le module de mappage de la chaîne) sont celles dans lesquelles l'extraction d'attributs spécifiques à une personne peut être accomplie pour un utilisateur.

Gestion de grandes quantités de données d'attributs d'utilisateur

L'authentification OpenID fonctionne sur la base d'adresses URL redirigées. Lorsqu'une réponse à une authentification de la part du fournisseur d'identité doit contenir plus de 2 Ko de données de registre d'utilisateurs, Tivoli Federated Identity Manager commute automatiquement les données en messages POST. Ce comportement est conforme à la spécification OpenID 2.0, qui permet d'appliquer l'envoi automatique de transactions POST aux messages indirects.

Remarque : La commutation automatique en messages POST n'est pas prise en charge dans les déploiements OpenID 1.1.

Reconnaissance du correspondant associé

A l'aide de la reconnaissance RP, les fournisseurs OpenID peuvent détecter et vérifier les adresses `return_to` des domaines prenant en charge OpenID.

La reconnaissance de partie de confiance s'effectue lorsqu'un fournisseur OpenID reçoit une demande de connexion unique sollicitée. La partie de confiance effectue alors le processus de reconnaissance sur l'URL spécifiée dans le paramètre `openid.realm` du message de connexion. Les parties de confiance peuvent publier leur URL `return_to` dans XRDS.

Un administrateur peut configurer le panneau de propriétés du fournisseur d'identité pour appliquer la reconnaissance de partie de confiance réussie. La macro figurant dans la page consent.html permet au fournisseur d'identité d'indiquer si la reconnaissance de partie de confiance n'a pas encore été réalisée. Pour plus d'informations sur la page d'accord d'authentification, voir .

Cette spécification permet aux fournisseurs OpenID de vérifier les demandes d'authentification et de s'assurer que les réponses sont redirigées vers des noeuds finaux `return_to` valides.

Si la reconnaissance ne peut pas vérifier l'URL `return_to` sur le domaine de la partie de confiance, Tivoli Federated Identity Manager affiche une erreur ou un avertissement selon la configuration.

L'attribut de réclamation IsRPReturnToValidated indique à la règle de mappage si la validation d'URL `return_to` s'est produite. Tivoli Federated Identity Manager ajoute cet attribut dans l'élément OpenIDClaims transmis au service de jeton de sécurité. Il permet à une règle de mappage de détecter les échecs de reconnaissance de la partie de confiance et de procéder à l'action appropriée. La valeur de cet attribut de réclamation peut être `true` ou `false`.

Modes d'authentification

OpenID prend en charge deux modes d'authentification : checkid_immediate et checkid_setup

Le mode d'authentification checkid_immediate s'emploie généralement dans les environnements comprenant de nombreux clients, dans lesquels un objet fenêtre intelligent exécute les tâches suivantes :

- Déterminer si l'utilisateur d'un navigateur détient une adresse URL OpenID particulière réclamée
- Eviter l'interaction entre le navigateur et l'utilisateur

Pour qu'un consommateur puisse adresser une requête checkid_immediate à un fournisseur d'identité, ajoutez ce paramètre d'entrée dans le formulaire de connexion :

```
<input type=''hidden'' name=''openid.mode'' value=''checkid_immediate''>
```

Le noeud de situé à l'URL du protocole de connexion unique Tivoli Federated Identity Manager initie la connexion à partir du consommateur.

- Lorsque la réponse fournie par le fournisseur d'identité est une assertion valide selon laquelle l'utilisateur est détenteur de l'URL d'identité, Tivoli Federated Identity Manager procède à un échange de jetons avec le service STS, puis établit la connexion avec le serveur point de contact. Ce comportement est le même que pour checkid_setup.
- Lorsque la réponse renvoyée par le fournisseur d'identité est une assertion ayant échoué, un modèle de page HTML est chargé à partir de la fabrique de pages. La macro de remplacement contenue dans la page est complétée avec la valeur du paramètre `open.user_setup_url` renvoyée par le fournisseur d'identité.

Le mode d'authentification checkid_setup permet au fournisseur d'identité d'interagir avec l'utilisateur, afin d'émettre une demande d'authentification ou d'auto-enregistrement avant que le résultat ne soit renvoyé au consommateur. Lorsqu'aucun mode d'authentification n'est spécifié dans le formulaire de connexion, le mode par défaut est checkid_setup.

Etant donné que checkid_setup est le mode par défaut, il n'est pas nécessaire de spécifier le mode dans le formulaire de connexion. Toutefois, le consommateur peut demander ce mode de manière spécifique. Voir le code suivant pour demander ce mode :

```
<input type=''hidden'' name=''openid.mode'' value=''checkid_setup''>
```

Le support de Tivoli Federated Identity Manager pour checkid_setup est un flux de connexion unique fédérée comportant une redirection vers le fournisseur d'identité pour l'authentification, ainsi qu'une interaction avec l'utilisateur pour l'approbation de la connexion. Le flux d'authentification a pour résultat le renvoi au consommateur des attributs de réponse signés. Lorsqu'une signature numérique est validée, les attributs sont générés dans un jeton STSUU (Security Token Service Universal User) et envoyés au service d'accréditation pour un échange contre une données d'identification IVCred. Les droits d'accès sont ensuite utilisés pour la connexion.

Fédérations de consommateurs

Le *consommateur* OpenID de Tivoli Federated Identity Manager joue un rôle similaire à celui que joue un *fournisseur de services* dans d'autres protocoles de connexion unique.

Le consommateur OpenID utilise une fédération Tivoli Federated Identity Manager qui présente un certain nombre de similitudes, mais aussi différences significatives avec les fédérations liées aux autres protocoles de connexion unique.

Il n'est notamment pas nécessaire d'associer directement les partenaires du fournisseur d'identité à des fédérations de consommateurs OpenID. L'échange de clés et l'association avec des fournisseurs d'identité particuliers sont contrôlés par l'URL d'identification OpenID, telle que déterminée au moment de l'exécution. Aucun ajout ni aucune configuration des partenaires n'a lieu pour une fédération de fournisseur de services OpenID.

L'entité de la fédération Tivoli Federated Identity Manager destinée au consommateur contient les éléments suivants :

- Un noeud final de connexion
- Un noeud final de renvoi de connexion
- Une adresse URL racine authentifiée (connue sous forme de *domaine* (realm) dans OpenID 2.0)
- Des paramètres indiquant le type de module de mappage dans la chaîne d'accréditation
- Les paramètres de configuration éventuellement associés
- Les règles d'agent d'utilisateur qui contrôlent la plage d'adresse IP autorisée, les réseaux et/ou les modèles de noms d'hôte destinés aux URL d'identification OpenID, ainsi qu'aux noeuds finals de serveurs OpenID.

La syntaxe de l'ID de protocole pour la fédération OpenID est la suivante :

`https://<nom_hôte:port>/FIM/sps/<nom_fédération>/openid`

Par exemple :

`https://webseald.example.com/FIM/sps/openidfedsp/openid`

Le noeud final de connexion

Le client Tivoli Federated Identity Manager prend en charge une adresse URL de renvoi de connexion. L'URL de connexion reçoit la requête POST du formulaire de connexion initial et lance le programme checkid_setup ou checkid_immediate.

Conformément au paramètre de fédération protocolID dans l'exemple précédent, le noeud final est :

`https://webseald.example.com/FIM/sps/openidfedsp/openid/login`

Remarque : Voici un exemple de noeud final pour les déploiements effectués avec WebSphere en tant que serveur point de contact :

`https://poc.example.com/sps/openidfedsp/openid/login`

Le délégué de la connexion unique situé sur le noeud final exécute les tâches suivantes :

1. Détermine le nom de connexion entrant à partir de l'adresse URL d'identité OpenID, ainsi que les paramètres d'extension.
2. Déterminez la forme canonique de l'adresse URL d'identification conformément à la spécification d'authentification OpenID applicable. La reconnaissance HTML et Yadis sont prises en charge.
3. Effectuez l'extraction de l'adresse URL d'identification finale, y compris les délégués, de l'utilisateur. Déterminez le serveur OpenID associé à l'utilisateur.
4. Si aucune association avec un fournisseur d'identité n'existe, créez-en une.
5. Adressez une requête checkid_setup ou checkid_immediate au serveur OpenID et redirigez le navigateur vers le fournisseur d'identité.

Noeud final de renvoi de connexion

Tivoli Federated Identity Manager prend en charge une adresse URL de renvoi de connexion. Le navigateur est redirigé vers cette adresse par le fournisseur d'identité une fois le processus de connexion unique terminé. Ce noeud final est transmis en tant que paramètre openid.return_to lors de la requête de connexion unique.

Ce noeud final est par exemple :

```
https://webseald.example.com/FIM/sps/openidfedsp/openid/loginreturn
```

Le délégué de connexion unique situé sur ce noeud final traite les réponses aux requêtes checkid_setup et checkid_immediate. Le délégué traite les réponses, ainsi que toute demande check_authentication ou invalidation de gestionnaire d'association susceptible d'être générée dans le résultat.

- Lorsqu'une réponse est renvoyée avec une signature validée avec succès, le service d'accréditation utilise les paramètres contenus dans la réponse. Les paramètres sont utilisés pour créer un jeton STSUU (Security Token Service Universal User). Le délégué utilise le service d'accréditation pour échanger le jeton STSUU avec une accréditation IVCred. Les droits d'accès sont ensuite utilisés pour l'authentification de Tivoli Federated Identity Manager.
- Lors du renvoi d'une réponse sans succès, une page d'erreur s'affiche.

Racine d'accréditation ou adresse URL de domaine

Le consommateur Tivoli Federated Identity Manager fournit également une *racine d'accréditation* ou une adresse URL de *domaine*. Cette adresse URL sert de base affichée pour l'utilisateur au niveau du fournisseur d'identité.

Tivoli Federated Identity Manager lit l'adresse URL de la racine d'accréditation à partir des propriétés de configuration. Cette propriété est initialement générée par les entrées de l'administrateur, afin de combiner les valeurs suivantes :

- Protocole

  Par exemple, `https`.
- Nom d'hôte

  Nom d'hôte du serveur point de contact
- Port

  Facultatif. Spécifié uniquement lorsqu'il ne s'agit pas du port standard.
- Barre oblique (/)

Par exemple :

```
https://webseald.example.com/
```

Connexion OpenID

Le consommateur Tivoli Federated Identity Manager présente un formulaire de connexion afin de demander l'URL OpenID de l'utilisateur. Ce formulaire peut comporter au choix des méthodes POST ou GET adressées au noeud final de connexion du consommateur Tivoli Federated Identity Manager. Les paramètres inclus peuvent comporter plusieurs adresses URL si nécessaire.

Tivoli Federated Identity Manager prend en charge :

- Les spécifications d'authentification OpenID 1.1
- Les spécifications d'authentification OpenID 2.0
- L'extension OpenID Simple Registration Extension 1.0
- L'extension OpenID Simple Registration Extension 1.1
- L'extension OpenID Attribute Exchange
- L'extension PAPE (Provider Authentication Policy Extension) 1.0

Remarque : La méthode de connexion utilisée par le consommateur Tivoli Federated Identity Manager consumer est la même que lors de l'accès au fournisseur d'identité Tivoli Federated Identity Manager ou à un autre fournisseur d'identité.

Considérons par exemple le scénario de déploiement suivant :

- WebSEAL est le point de contact d'un hôte appelé `www.example.com`
- Une fédération de consommateurs OpenID porte le nom `openidfedsp`

La figure 24 montre un exemple de formulaire de connexion valable pour cet exemple.

```
<html>
  <form method="post"
  action="https://www.example.com/FIM/sps/openidfedsp/openid/login">
    <img src="login-bg.gif" /> 
    <input type="text" name="openid_identifier" /> 
    <input type="submit" value="Login"/>
  </form>
</html>
```

Figure 24. Formulaire de connexion OpenID simple

Le fournisseur de services Tivoli Federated Identity Manager procède comme suit :

1. Lit le paramètre `openid_identifier`
2. Exécute le flux d'authentification spécifié pour l'authentification OpenID 2.0
3. Exécute une connexion EAI (External Authentication Interface) sur WebSEAL

Après une réponse checkid_immediate ou checkid_setup positive, le consommateur Tivoli Federated Identity Manager appelle le service d'accréditation afin de traiter les éventuelles manipulations sur les attributs obligatoires ou l'identité de l'utilisateur.

Au cours du processus de connexion, le consommateur peut demander des attributs auprès du fournisseur d'identité en spécifiant des paramètres additionnels dans le formulaire de connexion. Ces paramètres doivent correspondre aux noms des paramètres décrits dans l'extension OpenID ERS (Simple Registration Extension) 1.0. Vous pouvez également utiliser d'autres spécifications prises en charge, telles que Simple Registration Extension 1.1, Attribute Exchange 1.0 and Private Personal Identifier Generator 1.0.

A titre d'exemple, la figure 25, à la page 362 illustre un formulaire de connexion qui accomplit les exigences suivantes à l'aide de Simple Registration Extension :

- Demande d'adresse électronique au fournisseur d'identité
- Demande de la date de naissance au fournisseur d'identité
- Eventuellement, demande du nom complet de l'utilisateur
- Attribution d'une adresse URL renvoyant à une page descriptive des règles de confidentialité

```
<html>
  <form method="post"
  action="https://www.example.com/FIM/sps/openidfedsp/openid/login">
    <input type="hidden" name="openid.sreg.required"
     value="email,dob" />
    <input type="hidden" name="openid.sreg.optional"
     value="fullname" />
    <input type="hidden" name="openid.sreg.policy_url"
value="http://www.example.com/privacy_policy.html" />
    <img src="login-bg.gif" /> 
    <input type="text" name="openid_identifier" /> 
    <input type="submit" value="Login"/>
  </form>
</html>
```

Figure 25. Formulaire de connexion OpenID comportant les paramètres d'extension de registre

Lorsque ces paramètres sont présents dans la demande de connexion, Tivoli Federated Identity Manager les envoie au fournisseur d'identité. Cette action est effectuée lors des requêtes checkid_immediate et checkid_setup.

Les paramètres ne doivent pas nécessairement être masqués, ni organisés sous forme de liste séparée par des virgules.

Les paramètres peuvent être constitués d'attributs à valeurs multiples. L'usage des attributs à valeurs multiples permet au serveur de présenter à l'utilisateur des boutons d'option, de s zones de liste ou d'autres objets fenêtre à valeur multiples dans le code HTML. Tivoli Federated Identity Manager traite chaque valeur sous forme de liste séparée par des virgules. Les valeurs multiples constituées d'une seule entrée par valeur sont autorisées.

Vous pouvez mettre en oeuvre une connexion avec redirection automatique vers une adresse URL spécifiée. Lorsque WebSEAL est le serveur point de contact, les règles de traitement de l'authentification EAI s'appliquent. Vous pouvez inclure un paramètre `TARGET` optionnel dans le formulaire de connexion afin de rediriger l'utilisateur à la suite d'une authentification ayant abouti.

Modèles de pages

Le client Tivoli Federated Identity Manager utilise plusieurs modèles de pages HTML lors du traitement des demandes et erreurs d'authentification :

- Lorsque le consommateur traite une requête checkid_immediate et que le fournisseur d'identité ne parvient pas à déterminer la validité de l'adresse URL OpenID fournie par l'utilisateur, un fichier modèle est renvoyé par le consommateur.

  Voir «Modèle de page renvoyé pour checkid_immediate», à la page 405.
- Le consommateur utilise un fichier modèle pour les besoins de prise en charge du transport POST des messages indirects volumineux. Le consommateur Tivoli Federated Identity Manager prend en charge le transport POST des messages indirects volumineux. La prise en charge utilise un fichier modèle.

  Voir «Modèle de page pour l'envoi indirect de requêtes OpenID 2.0», à la page 404.
- Lorsque le traitement d'une requête checkid_immediate ou checkid_setup produit une erreur, le consommateur renvoie celle-ci au moyen d'un fichier modèle.

Voir «Modèle de page renvoyé pour les erreurs du serveur», à la page 406.

- Lorsqu'une erreur se produit sur le consommateur et qu'elle entraîne un blocage du traitement, le consommateur renvoie une erreur au moyen d'un fichier modèle.

  Voir «Modèle de page pour les erreurs liées à OpenID», à la page 402.

Chaînes d'accréditation du consommateur

Au cours du processus de connexion, Tivoli Federated Identity Manager gère le mappage des attributs et des identités. Lorsqu'une réponse checkid_immediate ou checkid_setup renvoyée par le fournisseur d'identité signale une assertion ayant abouti, Tivoli Federated Identity Manager génère tous les attributs et données de réponse PAPE renvoyés par le fournisseur d'identité dans un jeton STSUU (Security Token Service Universal User) puis utilise le service d'accréditation pour échanger ce jeton avec une accréditation IVCred.

La chaîne d'accréditation est constituée des éléments suivants :

- Un jeton STSUU en mode validation

  Le jeton contient l'URL d'identité OpenID ainsi que tous les paramètres d'extension dont les attributs utilisateur. Le jeton STSUU est généré par le délégué chargé du renvoi de la connexion OpenID une fois que la signature a été vérifiée en réponse à une connexion lancée par le fournisseur d'identité.

- Un module de mappage

  Le consommateur peut utiliser le module de mappage pour effectuer le mappage nécessaire des identités et des attributs.

  Le type de module de mappage à appliquer pour la fédération de consommateur est défini lors de la configuration de celle-ci. Les types de module de mappage standard sont pris en charge :

  – Règles de mappage XSLT ou Javascript
  – Module de mappage de Tivoli Directory Integrator
  – Modules de mappage personnalisés.

  Dans de nombreux cas, l'utilisation des règles de mappage de script est suffisante, puisqu'elles ne nécessitent généralement l'extraction d'aucun attribut externe.

  La distribution du produit Tivoli Federated Identity Manager inclut plusieurs exemples de règles de mappage de script et un exemple de chaîne d'assemblage Tivoli Directory Integrator (module de mappage).

- Une accréditation IVCred en mode émission

Liaisons avec les comptes

L'un des scénarios importants pour le consommateur du point de vue d'OpenID consiste à exécuter une liaison de compte. Si, par exemple, un utilisateur s'est authentifié directement sur un site Web faisant également office de consommateur OpenID, ce site Web peut autoriser l'utilisateur à lier son compte à une identification OpenID. En effectuant une connexion OpenID alors que l'utilisateur est déjà connecté au site Web, le site de consommation peut associer cet identificateur OpenID avec le compte actuellement connecté.

Pour permettre la prise en charge de ce scénario, Tivoli Federated Identity Manager envoie des réclamations dans un appel WS-Trust adressé au service STS. L'appel inclut le nom de l'utilisateur actuellement connecté lorsqu'une session d'authentification existe.

La figure 26 illustre un exemple de format pour les réclamations adressées au service d'accréditation. Ce format est disponible pour mapper des implémenteurs de modules via le jeton STSUU.

```
<fimopenid:OpenIDClaims
xmlns:fimopenid="urn:ibm:names:ITFIM:openid"
xmlns:fimpape="urn:ibm:names:ITFIM:openid:PAPE"
ClaimedId="https://www.myidp.ibm.com/FIM/op/
85da8845-0127-1a04-9a9f-dca9f50a9649"
IdentityURL="https://www.myidp.ibm.com/FIM/op"
IsOPIdentifierLogin="true"
IsRPReturnToValidated="false"
NormalizedIdentityURL="https://www.myidp.ibm.com/FIM/op/
85da8845-0127-1a04-9a9f-dca9f50a9649"
OPLocalId="https://www.myidp.ibm.com/FIM/op/
85da8845-0127-1a04-9a9f-dca9f50a9649"
OpenIDServerURL="https://www.myidp.ibm.com/FIM/
sps/openididp/openid/sso"
ReauthCount="0"
ReturnTo="https://www.myrp.ibm.com/sps/myrp/openid/
loginreturn?nonce=uuid85e85b2a-0127-1286-99d4-d5e72a774a5f"
Signed="openid.op_endpoint,openid.return_to,
openid.response_nonce,openid.assoc_handle,
openid.claimed_id,openid.identity,
openid.sreg.dob,openid.sreg.gender,
openid.sreg.email,openid.sreg.language,
openid.sreg.timezone,openid.sreg.fullname,
openid.sreg.postcode,openid.sreg.country,
openid.sreg.nickname,openid.ns.sreg,
openid.ns.pape,openid.pape.auth_time,
openid.pape.auth_policies,openid.pape.auth_level.ns1,
openid.pape.auth_level.ns.ns1"
Target="https://www.myrp.ibm.com/fimivt/protected/ivtlanding.jsp"
Version="http://specs.openid.net/auth/2.0">
<fimpape:OpenIDPAPEClaims>
<fimpape:Attribute name="satisfied_auth_age">
<fimpape:Value>true</fimpape:Value>
</fimpape:Attribute>
<fimpape:Attribute name="openid.pape.preferred_auth_levels">
<fimpape:Value>urn:ibm:names:ITFIM:5.1:accessmanager
</fimpape:Value>
</fimpape:Attribute>
<fimpape:Attribute name="satisfied_auth_policies">
<fimpape:Value>true</fimpape:Value>
</fimpape:Attribute>
<fimpape:Attribute name="openid.pape.preferred_auth_policies">
<fimpape:Value>http://www.idmanagement.gov/schema/
2009/05/icam/openid-trust-level1.pdf</fimpape:Value>
<fimpape:Value>http://schemas.xmlsoap.org/ws/2005
/05/identity/claims/privatepersonalidentifier</fimpape:Value>
</fimpape:Attribute>
</fimpape:OpenIDPAPEClaims>
</fimopenid:OpenIDClaims>
```

Figure 26. Réclamations OpenID lors d'un appel consommateur WS-Trust

L'attribut PrincipalName, lorsqu'il est présent dans les réclamations, contient le nom d'utilisateur Tivoli Federated Identity Manager qui désigne l'utilisateur actuellement authentifié. Ceci permet aux chaînes d'accréditation d'associer automatiquement, par le biais de règles de mappage, un OpenID particulier à un compte existant.

L'exemple contient d'autres réclamations sous forme de paramètres issus de la réponse à une connexion unique à partir du serveur OpenID. L'attribut relatif à l'URL d'identité correspond à celle qui est fournie par l'utilisateur dans le formulaire de connexion. L'attribut NormalizedIdentityURL représente la forme canonique de l'URL d'identité qui résulte de la normalisation accomplie dans le cadre du procédure de reconnaissance.

Le jeton STSUU envoyé au service d'accréditation avec la requête contient des attributs relatifs à chacun des composants répertoriés dans le jeu d'attributs de `Signed`, ainsi que certains autres paramètres portant sur des chaînes de requête.

La figure 27, à la page 366 illustre le jeton STSUU généré à partir de l'appel WS-Trust illustré à la figure 26, à la page 364.

```
<?xml version="1.0" encoding="UTF-8"?>
<stsuuser:STSUniversalUser xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser">
<stsuuser:Principal><stsuuser:Attribute name="name"><stsuuser:Value>https://www.myidp.ibm.com/FIM/
op/85da8845-0127-1a04-9a9f-dca9f50a9649</stsuuser:Value></stsuuser:Attribute>
</stsuuser:Principal><stsuuser:AttributeList><stsuuser:Attribute name="openid.identity">
<stsuuser:Value>https://www.myidp.ibm.com/FIM/op/85da8845-0127-1a04-9a9f-dca9f50a9649</stsuuser:Value>
</stsuuser:Attribute></stsuuser:AttributeList><stsuuser:RequestSecurityToken />
<stsuuser:ContextAttributes><stsuuser:Attribute name="openid.op_endpoint">
<stsuuser:Value>https://www.myidp.ibm.com/FIM/sps/openididp/openid/sso</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.sreg.email">
<stsuuser:Value>jsmith@ibm.com</stsuuser:Value></stsuuser:Attribute>
<stsuuser:Attribute name="openid.sig"><stsuuser:Value>NuKNV1ypZC16d3og6HbvjbCedPVjhRbWAWZ9Gq6g1DU=
</stsuuser:Value></stsuuser:Attribute>
<stsuuser:Attribute name="openid.pape.auth_level.ns1"><stsuuser:Value>1</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.claimed_id">
<stsuuser:Value>https://www.myidp.ibm.com/FIM/op/85da8845-0127-1a04-9a9f-dca9f50a9649</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.ns">
<stsuuser:Value>http://specs.openid.net/auth/2.0</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.sreg.language">
<stsuuser:Value>en</stsuuser:Value></stsuuser:Attribute>
<stsuuser:Attribute name="openid.sreg.fullname"><stsuuser:Value>John Smith</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="nonce">
<stsuuser:Value>uuid85e85b2a-0127-1286-99d4-d5e72a774a5f</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.pape.auth_time">
<stsuuser:Value>2010-03-22T12:47:41Z</stsuuser:Value> </stsuuser:Attribute>
<stsuuser:Attribute name="openid.return_to"><stsuuser:Value>https://www.myrp.ibm.com/sps/myrp/
openid/loginreturn?nonce=uuid85e85b2a-0127-1286-99d4-d5e72a774a5f</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.signed"><stsuuser:Value>
op_endpoint,return_to,response_nonce,assoc_handle,claimed_id,identity,sreg.dob,
sreg.gender,sreg.email,sreg.language,sreg.timezone,sreg.fullname,
sreg.postcode,sreg.country,sreg.nickname,ns.sreg,ns.pape,pape.auth_time,
pape.auth_policies,pape.auth_level.ns1,pape.auth_level.ns.ns1</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.sreg.nickname">
<stsuuser:Value>Smithy</stsuuser:Value></stsuuser:Attribute>
<stsuuser:Attribute name="openid.identity">
<stsuuser:Value>https://www.myidp.ibm.com/FIM/
op/85da8845-0127-1a04-9a9f-dca9f50a9649</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.ns.sreg">
<stsuuser:Value>http://openid.net/extensions/sreg/1.1</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.pape.auth_level.ns.ns1">
<stsuuser:Value>urn:ibm:names:ITFIM:5.1:accessmanager</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.sreg.dob">
<stsuuser:Value>1980-12-25</stsuuser:Value></stsuuser:Attribute>
<stsuuser:Attribute name="openid.sreg.postcode"><stsuuser:Value>99999</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.assoc_handle">
<stsuuser:Value>uuid85ca8353-0127-1776-9b7b-c75a4586c507</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.sreg.country">
<stsuuser:Value>AU</stsuuser:Value></stsuuser:Attribute>
<stsuuser:Attribute name="openid.pape.auth_policies">
<stsuuser:Value>http://schemas.xmlsoap.org/ws/2005/05/
identity/claims/privatepersonalidentifier http://www.idmanagement.gov/schema/2009/05/icam/
openid-trust-level1.pdf</stsuuser:Value></stsuuser:Attribute>
<stsuuser:Attribute name="openid.mode"><stsuuser:Value>id_res</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.sreg.timezone">
<stsuuser:Value>Australia/Brisbane</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.ns.pape">
<stsuuser:Value>http://specs.openid.net/extensions/pape/1.0</stsuuser:Value>
</stsuuser:Attribute><stsuuser:Attribute name="openid.sreg.gender">
<stsuuser:Value>M</stsuuser:Value> </stsuuser:Attribute>
<stsuuser:Attribute name="openid.response_nonce">
<stsuuser:Value>2010-03-22T12:48:08Zuuid85ea7849-0127-1515-b2b5-e9223d6c6970
</stsuuser:Value></stsuuser:Attribute></stsuuser:ContextAttributes>
<stsuuser:AdditionalAttributeStatement />
</stsuuser:STSUniversalUser>
```

Figure 27. Exemple de jeton STSUU lors d'une requête de service d'accréditation sur le consommateur OpenID

Règles relatives à l'agent d'utilisateur

Il est possible de configurer l'agent d'utilisateur en vue de restreindre la liste des emplacements auxquels il a tenté d'accéder. Cette configuration est réalisée pour dissuader les utilisateurs mal intentionnés de se connecter aux ressources internes par l'intermédiaire de l'agent d'utilisateur.

Le consommateur fait appel à un *agent d'utilisateur* (client HTTP) pour établir directement la connexion avec les adresses URL d'identité OpenID, ainsi que les

adresses URL de serveurs OpenID auxquels elles font référence. Le fournisseur utilise également le même type de configuration de règle d'agent utilisateur pour les opérations de reconnaissance de partie de confiance.

Les restrictions sont gérées au moyen de la configuration de règles de connexion statique et d'un module d'autorisation de point final dynamique personnalisable. Le module de point final dynamique peut être utilisé pour limiter davantage l'accès aux points finaux lors de l'exécution.

Chaque fédération Tivoli Federated Identity Manager possède une configuration de règles globales. La configuration par défaut définit le comportement par défaut lorsque l'hôte ne figure pas explicitement dans les listes des accès autorisés ou refusés. Ce paramètre permet, selon les cas, d'autoriser ou de refuser l'accès aux adresses URL en tant que comportement par défaut. En plus des règles globales, les administrateurs peuvent créer des plug-ins d'accès de point finaux dynamiques personnalisés. Ces plug-ins peuvent vérifier une liste en ligne de points finaux dignes de confiance ou non. Les administrateurs peuvent ajouter les plug-ins personnalisés à la liste de modules d'autorisation d'accès de pointa finaux dynamiques.

Règle de connexion statique

Avec une règle de connexion statique, un administrateur peut répertorier les hôtes autorisés et refusés. Selon le comportement par défaut sélectionné, l'administrateur peut également spécifier une liste d'hôtes dans la liste autorisée ou refusée.

Lorsque le comportement par défaut sélectionné consiste à **refuser** l'accès, seuls les hôtes contenus dans les listes **autorisées** sont accessibles via l'agent d'utilisateur. Ce paramètre est restrictif. Chaque URL d'identité OpenID et chaque serveur dont vous souhaitez autoriser l'accès doit figurer dans les listes d'autorisation. Lorsque le comportement défini par défaut est un accès **refusé**, toutes les *listes d'accès refusés* ne sont pas forcément utiles. L'accès à tous les hôtes est par défaut refusé, sauf si ceux-ci sont inclus de manière explicite dans une liste d'autorisation.

Lorsque le comportement défini par défaut sélectionné est un accès **autorisé**, l'hôte est contacté, sauf s'il figure dans une *liste d'accès refusés*. Ce paramètre est plus tolérant et permet généralement aux utilisateurs de se connecter à partir d'un serveur OpenID légitime sur Internet. Toutefois, lorsque le paramètre par défaut consiste à **autoriser** les connexions, les listes de refus d'accès doivent être configurées avec soin.

Tivoli Federated Identity Manager prend en charge les types de listes suivants :

Listes d'**autorisation** :

- Liste d'expressions régulières représentant des noms d'hôte et configurable par l'utilisateur
- Liste des masques réseau d'adresse IP (IPv4 et IPv6), configurable par l'utilisateur

Listes de **refus** :

- Liste d'expressions régulières représentant des noms d'hôte et configurable par l'utilisateur
- Liste des masques réseau d'adresse IP (IPv4 et IPv6), configurable par l'utilisateur

- Liste intégrée d'expressions régulières représentant des noms d'hôte refusés par défaut
- Liste intégrée d'expressions régulières représentant des masques réseau d'adresses IP refusées par défaut

Remarque : Les listes d'autorisation sont prioritaires sur les listes de refus.

Les listes d'accès aux noms d'hôtes sont conformes à la syntaxe d'expressions régulières Java comme défini dans le classe de modèles. La liste utilise des expressions régulières pour la concordance avec les noms d'hôtes.

Les listes de refus intégrées ne peuvent pas être modifiées par les utilisateurs. Toutefois, la liste peut être substituée afin d'autoriser certaines entrées via l'ajout d'expressions régulières de noms d'hôte ou de masques de réseau aux listes d'autorisation configurables par l'utilisateur.

La figure 28 illustre les noms d'hôte refusés par défaut. Les valeurs par défaut assurent une protection contre les attaques consistant à tenter d'accéder à des adresses URL arbitraires sur le système local.

```
.*\.localdomain
localhost
```

Figure 28. Expressions régulières représentant des noms d'hôte avec refus par défaut

La figure 29 illustre les masques réseau l'adresse IP default-deny. Ces masques de réseau incluent plusieurs adresses IPv4 et IPv6 non transférables. Cette liste peut être remplacée en ajoutant les réseaux pour lesquels vous souhaitez autoriser les connexions avec la liste des masques de réseau IP autorisés.

```
0.0.0.0/8
10.0.0.0/8
127.0.0.0/8
169.254.0.0/16
172.16.0.0/12
192.168.0.0/16
255.255.255.255
::/128
::1/128
::/96
fc00::/7
fe80::/10
ff00::/8
```

Figure 29. Masques réseau des adresses IP default-deny

Plug-in d'accès aux noeuds finaux dynamiques

Un plug-in d'accès aux noeuds finaux dynamiques est un module personnalisé. Un administrateur peut créer le plug-in d'accès aux noeuds finaux dynamiques personnalisés pour vérifier les listes externes d'hôtes de confiance ou non.

Lorsqu'un administrateur sélectionne un **plug-in d'accès aux noeuds finaux dynamiques** dans le module d'autorisation d'accès aux noeuds finaux dynamiques personnalisé, le logiciel vérifie les noeuds finaux spécifiés. Les noeuds finaux indiqués sont vérifiés afin de déterminer s'ils sont dignes de confiance. Vous pouvez utiliser ce paramètre avec la liste d'autorisation ou de refus d'accès.

Toutefois, si vous définissez l'autorisation de noeuds finaux dynamiques dans la liste d'autorisation d'accès par défaut, le logiciel utilise uniquement les noeuds finaux dans les listes de refus ou d'autorisation.

Exemple : autoriser n'importer quel serveur Internet OpenID, refuser l'accès à l'intranet 9.x.x.x

- Pour configurer cet environnement, la règle d'accès par défaut indiquée est **autoriser**
- La liste d'hôtes autorisés est ignorée.
- La liste d'hôtes refusés est ignorée.
- Le masque de réseau de l'adresse IP refusé est 9.0.0.0/8.

  Plusieurs masques de réseau peuvent être ajoutés s'il existe plusieurs réseaux intranet et il convient d'ajouter les équivalents IPv6 si le réseau prend en charge à la fois les protocoles IPv4 et IPv6.

Exemple : autoriser l'accès OpenID uniquement aux sociétés example1 et example2

- Pour configurer cet environnement, la règle d'accès par défaut indiquée est **refuser**
- La liste des hôtes non autorisés et des maques de réseau IP est ignorée.
- La liste des expressions régulières pour les hôtes autorisés est :

  ```
  .*\.example1\.com,openid\.example2\.com,openidserver\.example2\.com
  ```

Les identificateurs OpenID de 'example1' se présentent sous la forme `john.example1.com` et le serveur OpenID résolu par les URL d'identité est le suivant :

```
https://www.example1.com/openidProcessing.action
```

Pour l'exemple 2, les identificateurs OpenID ont la forme `openid.example2.com/<example2_screenname>` et se résolvent en une page HTML pointant vers le serveur OpenID :

```
https://api.screenname.example2.com/auth/openidServer
```

La nécessité de cette adresse URL est la raison pour laquelle ces deux noms d'hôte apparaissent dans la liste.

Exemple : autoriser tout nom d'hôte contenant la chaîne .ibm.com

Cet exemple indique des paramètres autorisant un utilisateur à accéder tout nom d'hôte contenant la chaîne .ibm.com.

- Pour configurer cet environnement, sélectionnez un plug-in personnalisé.
- La liste d'hôtes autorisés est vérifiée.
- La liste d'hôtes refusés est vérifiée.
- Le plug-in de noeuds finaux dynamiques personnalisés est :

```
package com.tivoli.am.fim.demo.ibmaccessapproval;

import java.net.MalformedURLException;
import java.net.URL;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;

import com.tivoli.am.fim.useragent.AccessApproval;
```

```
public class IBMAccessApproval implements AccessApproval {

 final static String CLASS = IBMAccessApproval.class.getName();

 final static Logger _log = Logger.getLogger(CLASS);

 public IBMAccessApproval() {
 }

 public boolean canAccess(Map ctx) {
  String methodName = "canAccess";
  _log.entering(CLASS, methodName, new Object[] { ctx });
  boolean result = false;
  boolean finestLoggable = _log.isLoggable(Level.FINEST);
  try {
   String endpoint = (String) ctx.get(AccessApproval.CTX_ENDPOINT);
   String fedname = (String) ctx
     .get(AccessApproval.CTX_FEDERATION_NAME);
   String fedid = (String) ctx
   .get(AccessApproval.CTX_FEDERATION_ID);

   if (finestLoggable) {
    _log.logp(Level.FINEST, CLASS, methodName, "Fedname: "
      + fedname + " Fedid: " + fedid + " Endpoint: " + endpoint);
   }

   try {
    URL u = new URL(endpoint);
    String hostname = u.getHost();
    if (hostname != null && hostname.indexOf(".ibm.com") > 0) {
     result = true;
    }
   } catch (MalformedURLException e) {
    e.printStackTrace();
   }

  } finally {
   _log.exiting(CLASS, methodName, "" + result);
  }
  return result;
 }
}
```

Extensions OpenID

Extension OpenID Simple Registration Extension

Au cours du processus de connexion, le consommateur peut demander des attributs auprès des fournisseurs d'identité en spécifiant des paramètres additionnels dans le formulaire de connexion. Ces paramètres doivent correspondre aux noms des paramètres décrits dans l'extension OpenID Simple Registration Extension 1.0 ou Attribute Exchange Extension 1.0, selon applicable. L'extension Simple Registration Extension (SREG) est une extension au protocole d'authentification OpenID et prend en charge une simple liste d'informations d'enregistrement utilisateur commun.

Pou plus d'informations, voir la documentation OpenID à l'adresse suivante : http://openid.net/specs/openid-simple-registration-extension-1_0.html

```
<form name="openidLoginForm" method="post"
action="https://sp.example.com/FIM/sps/openidsp/openid/login">
<input name="openid.mode" type="hidden"
value="checkid_setup">
<input name="openid.sreg.required" type="hidden"
value="email">
<input name="openid.sreg.optional" type="hidden"
value="fullname,dob">
<input name="openid.sreg.policy_url" type="hidden"
value="https://sp.example.com/privacy_policy.html">
<input name="TARGET" type="hidden"
value="https://sp.example.com/myapp">
<input name="openid_identifier" type="text">
<input value="OpenID Login" type="submit">
</form>
```

Figure 30. Exemple d'extension Simple Registration Extension

Extension OpenID Attribute Exchange

Les fournisseurs d'identité peuvent utiliser des extensions OpenID pour obtenir les attributs utilisateur et les communiquer aux clients.

L'extension d'échange d'attribut fournit aux fournisseurs d'identité la possibilité de communiquer les attributs utilisateur aux clients.

Le protocole AX (Attribute Exchange Extension) peut être étendu pour accommoder plusieurs types d'attributs et des attributs à plusieurs valeurs. Les attributs sont identité par un URI unique et correspondent en général à des informations d'identité personnelles. Pour plus d'informations, consultez la documentation relative à OpenID à l'adresse suivante : http://openid.net/specs/openid-attribute-exchange-1_0.html

Le protocole Attribute Exchange Extension fournit une compatibilité stricte avec OpenID 2.0. Vous pouvez utiliser l'une ou les deux extensions en même temps. Utilisez le protocole Attribute Exchange Extension à moins que vous ne deviez être compatible avec une ancienne implémentation OpenID 1.1 qui ne prend en charge que SREG.

En tant qu'administrateur, vous pouvez ajouter un ensemble de paramètres au formulaire de connexion OpenID envoyé au noeud final de connexion.

L'exemple montre un formulaire de connexion avec les conditions requises suivantes :

- Demande l'adresse électronique au fournisseur d'identité
- Demande éventuellement le nom complet, la date de naissance, les amis et groupes.

```
<form name="openidLoginForm" method="post"
action="https://sp.example.com/FIM/sps/openidsp/openid/login">
<input name="openid.mode" type="hidden" value="checkid_setup">
<input name="openid.ax.required" type="hidden" value="axemail">
<input name="openid.ax.if_available" type="hidden"
value="axfullname,axdob,axfriends,axgroups">
<input name="openid.ax.type.axemail" type="hidden"
value="http://axschema.org/contact/email">
<input name="openid.ax.type.axfullname" type="hidden"
value="http://axschema.org/namePerson">
<input name="openid.ax.type.axdob" type="hidden"
value="http://axschema.org/birthDate">
<input name="openid.ax.type.axfriends" type="hidden"
value="http://example.com/myschema/friends">
<input name="openid.ax.count.axfriends" type="hidden"
value="5">
<input name="openid.ax.type.axgroups" type="hidden"
value="http://example.com/myschema/groups">
<input name="openid.ax.count.axgroups" type="hidden"
value="unlimited">
<input name="TARGET" type="hidden"
value="https://sp.example.com/myapp">
<input name="openid_identifier" type="text">
<input value="OpenID Login" type="submit">
</form>
```

Figure 31. Exemple d'extension Attribute Exchange Extension

Remarque : Si aucun chiffre explicite n'est demandé pour un paramètre d'échange d'attribut, la valeur maximale par défaut est 1.

Tivoli Federated Identity Manager envoie des paramètres au fournisseur d'identité lors des demandes `checkid_immediate` et `checkid_setup`. Les messages d'extraction envoyés avec la demande extraient les attributs d'identité personnelle de l'utilisateur. Pour plus d'informations sur les messages d'extraction, voir la documentation OpenID : http://openid.net/specs/openid-attribute-exchange-1_0.html#fetch

Paramètres de demande d'extraction Attribute Exchange Extension

Attribute Exchange Extension prend en charge un modèle d'information qui combine un identificateur de sujet, un identificateur de type d'attribut et une valeur. L'inclusion d'autres paramètres relie la demande d'extraction Attribute Exchange à une requête d'authentification standard. Pour permettre au client d'extraire des informations à partir du fournisseur d'identité, spécifiez les paramètres de zone de formulaire suivants dans le formulaire de connexion.

openid.ax.required
: Extrait les attributs requis à partir du fournisseur d'identité. La valeur représente une liste d'alias, qui sont des étiquettes représentant des attributs individuels au niveau du fournisseur d'identité. Reliez chaque alias sur un URI qui identifie l'attribut dans un paramètre `openid.ax.type.`*`alias`* distinct. *(facultatif)*

openid.ax.if_available
: Extrait un attribut disponible à partir du fournisseur d'identité. La valeur a les mêmes exigences que `openid.ax.required`. *(facultatif)*

Remarque : Vous devez spécifier `openid.ax.required` ou `openid.ax.if_available` dans la requête. Chaque alias d'attribut demandé doit avoir un paramètre `openid.ax.type.`*`alias`* associé.

openid.ax.type.*alias*
Relie l'alias à un URI définissant la signification de l'attribut. Vous devez spécifier un paramètre pour chaque alias défini dans `openid.ax.required` ou `openid.ax.if_available`. *(facultatif)*

Un grand nombre d'attributs classiques ont défini des URI de type à http://www.axschema.org/types/

openid.ax.sendalways
Inclut les informations Attribute Exchange Extension OpenID dans les demandes d'authentification au niveau du fournisseur d'identité. L'environnement exécution du client envoie les informations de requête Attribute Exchange Extension si le fournisseur d'identité prend en charge Attribute Exchange Extension avec XRDS. La valeur par défaut est false. *(facultatif)*

Paramètres de réponse d'extraction Attribute Exchange Extension

Une fois les droits d'accès accordés à un fournisseur d'identité, un message de réponse d'extraction fournit les informations dans les paramètres de demande d'extraction. Les paramètres de réponse d'extraction facultatifs suivants spécifient les attributs personnels extraits à partir du fournisseur d'identité.

openid.ax.type.*alias*
Spécifie le type d'URI pour l'attribut d'extraction identifié par l'alias. *(facultatif)*

openid.ax.count.*alias*
Renvoie le nombre de valeurs spécifiées pour l'attribut correspondant à l'alias. Si vous ne spécifiez pas de valeur spécifique, une seule valeur est renvoyée.

openid.ax.value.*alias*
Affecte une valeur spécifiée pour l'attribut correspondant à l'alias. *(facultatif)*

openid.ax.value.*alias***.number**
Affecte une valeur spécifiée pour l'attribut correspondant à l'alias. Ce paramètre est requis si `openid.ax.count.`*`alias`* est envoyé et qu'au moins une valeur est configuré pour l'attribut associé. Il doit y avoir un paramètre distinct pour chaque valeur de l'alias, avec des nombres incrémentiels.

Extension de règle d'authentification de fournisseur OpenID

Utilisez la console d'administration pour configurer l'extension PAPE OpenID.

Lorsqu'un utilisateur lance l'authentification à partir d'une partie de confiance avec un identificateur OpenID, la partie de confiance demande le fournisseur d'identité pour authentifier l'utilisateur.

OpenID Provider Authentication Policy Extension (PAPE) est un mécanisme dans lequel la partie utilisatrice peut effectuer les actions suivantes :

- Demander aux fournisseurs d'identité d'utiliser des règles d'authentification spécifiques lors de l'authentification d'un utilisateur.

- Demander au fournisseur d'identité d'informer la partie de confiance des règles d'authentification utilisées lors de l'authentification.
- Demander au fournisseur d'identité de communiquer les niveaux d'authentification tels que défini dans les ensembles de niveaux d'assurance personnalisés demandés.

Selon votre rôle dans la fédération, certains paramètres sont disponibles dans le panneau des propriétés de configuration de la console d'administration.

Remarque : Les paramètres PAPE peuvent uniquement être configurés APRES la création d'une fédération. Utilisez le panneau de propriétés de fédération pour indiquer les paramètres de la configuration.

Implémentation PAPE de partie de confiance

Utilisez le panneau de propriétés de configuration de la partie de confiance pour activer PAPE. Une fois que le paramètre est activé, les attributs PAPE indiqués sont envoyés au fournisseur d'identité dans la demande d'authentification. Lorsqu'une partie de confiance envoie la demande d'authentification avec les attributs PAPE indiqués, le fournisseur d'identité envoie une réponse. La réponse indique quelles exigences sont remplies et lesquelles ne le sont pas. Grâce à la réponse, la partie de confiance peut déterminer s'il faut authentifier l'utilisateur.

Spécifiez les paramètres suivants dans le panneau de propriétés de configuration de la partie de confiance :

Mode d'application

- Strict

  Indique qu'un utilisateur n'est pas authentifié si les exigences PAPE ne sont pas remplies.
- Lenient

  Indique qu'un utilisateur est authentifié même si les exigences PAPE ne sont pas remplies. La règle de mappage utilisée dans la fédération accède aux informations de réponse. La réponse indique quelles exigences sont remplies et lesquelles ne le sont pas. Ce paramètre permet à l'auteur de la règle de mappage de choisir s'il souhaite connecter l'utilisateur en se basant sur ces informations. La règle de mappage fournit une autorisation plus limitée.

Règles d'authentification

Indique un ensemble d'URI de règles d'authentification. Les URI représentent les règles d'authentification que le fournisseur d'identité doit respecter lors de l'authentification d'un utilisateur. Si plusieurs règles sont requises, le fournisseur d'identité doit en respecter le plus grand nombre possible. Le fournisseur d'identité indique ensuite quelles règles d'authentification ont été respectées dans la réponse.

Age maximal de l'authentification

Indique la période pendant laquelle l'utilisateur doit avoir été authentifié. Si cette période a expiré, le fournisseur d'identité doit authentifier de nouveau l'utilisateur.

Niveaux d'assurance préférés

Indique une liste ordonnée des URI préférés d'espace de nom du niveau d'assurance. Les valeurs d'espace de nom du niveau d'assurance déterminent le niveau du certificat contenu dans l'authentification de

l'utilisateur. Les parties de confiance demandent des informations concernant ces espaces de nom du niveau d'assurance provenant du fournisseur d'identité.

Implémentation PAPE du fournisseur d'identité

Le panneau de propriétés de configuration du fournisseur d'identité indique les conditions sous lesquelles un utilisateur doit s'authentifier.

Remarque : Si vous comptez utiliser la gestion de cookie WebSEAL avec l'implémentation OpenID PAPE, assurez-vous que la liste des cookies gérés n'inclut pas le cookie de session WebSphere. Voir «Configuration de WebSEAL pour gérer les cookies», à la page 591.

Spécifiez les paramètres suivants dans le panneau de propriétés de configuration du fournisseur d'identité :

Authentification imposée à tout âge d'authentification maximal PAPE demandé
Ce paramètre indique qu'un utilisateur doit toujours s'authentifier. Si cette option est sélectionnée, la zone Maximum authentication age allowable clock skew (Décalage horloge maximal autorisé de l'âge d'authentification) est désactivée.

Décalage horloge maximal autorisé de l'âge d'authentification
Lorsqu'un âge d'authentification maximum est demandé par un fournisseur de services au cours d'une connexion unique, la règle de mappage du fournisseur d'identité doit renvoyer la dernière heure d'authentification de l'utilisateur. Ce paramètre permet de calculer le décalage d'horloge entre :

- la dernière heure d'authentification retournée par la règle de mappage du fournisseur d'identité
- l'horloge du fournisseur d'identité

Généralement, l'heure de décalage est un petit nombre, mais qui peut prendre en compte les différences entre la machine du point de contact et la machine d'exécution.

Formulaire de configuration du fournisseur d'identité

Tivoli Federated Identity Manager comprend un assistant qui vous guidera tout au long de la configuration des fédérations OpenID. L'assistant vous invite à renseigner les propriétés nécessaires pour votre déploiement. Ce formulaire décrit les propriétés.

Ce formulaire vous permet de planifier vos propriétés et vous pouvez vous y référer lors de l'exécution de l'assistant.

Nom de la fédération
Le nom peut correspondre à n'importe quelle chaîne de caractères. Par exemple, `openid-idp`. Cette zone est obligatoire.

Rôle de la fédération
Votre rôle est *Fournisseur d'identité*.

Nom de la société
Nom de la société qui crée la fédération. La valeur peut correspondre à n'importe quelle chaîne de caractères. Vous pouvez également utiliser le caractère espace, ainsi que les autres caractères. Cette zone est obligatoire.

Protocole de la fédération
OpenID.

Serveur point de contact
Adresse URL du serveur qui agit en tant que point de contact initial pour les requêtes entrantes. L'adresse est constituée d'une spécification de protocole, du nom d'hôte du serveur et (en option) d'un numéro de port. Lorsque WebSEAL est le serveur point de contact, la jonction WebSEAL est spécifiée.

Exemple de valeur :

```
https://webseald.example.com/FIM
```

Remarque : Pour le support d'OpenID, le serveur point de contact doit utiliser le protocole SSL (Secure Socket Layer). L'adresse URL spécifiée doit être du type `https://`.

Expiration d'association (secondes)
Définit la durée de vie de l'indicateur d'association. Ce fournisseur d'identité contrôle cette valeur. Entrez un nombre positif. La valeur par défaut est de 3 600 secondes.

Délai d'expiration de réponse de l'élément Nonce (secondes)
Indique le nombre de secondes restant à un correspondant associé fonctionnant sans association établie avant de devoir exécuter la demande `check_authentication`. Si ce nombre est paramétré sur un nombre positif, cette option empêche de refaire un `contrôle_d_authentification`. Cette restriction s'applique aux clients disposant de parties de confiance incapables de créer ou de stocker des associations. La valeur par défaut est 30 secondes.

Générateur d'ID
Indique quel générateur d'ID crée une valeur remplaçant `@ID@` dans l'URL d'identité. Des générateurs d'ID différents créent des valeurs différentes pour `@ID@`.

Structure de l'URL d'identité OpenID

Représente l'expression régulière à laquelle les URL d'identité sont comparés pour la fédération. Tivoli Federated Identity Manager remplace la partie `@ID@`. La valeur par défaut est l'URL du nom d'hôte de protocole de connexion unique fournie par l'assistant d'installation.

Par exemple, si vous avez défini le serveur de point de contact suivant dans l'assistant :

```
https://webseald.example.com/FIM
```

la structure d'URL d'identité par défaut est la suivante :

```
https://webseald.example.com/@ID@
```

URL de configuration d'utilisateur
Spécifie l'URL envoyée en réponse à une requête `checkid_immediate` provenant d'un consommateur. L'URL est utilisée lorsque le fournisseur d'identité ne peut pas déterminer si un propriétaire possède une URL d'identité précise.

L'URL par défaut est l'URL du serveur point de contact que vous avez définie sur le panneau Serveur point de contact.
A titre d'exemple, lorsque vous avez précédemment spécifié, dans l'assistant, le serveur point de contact suivant :

https://webseald.example.com/FIM

l'URL de configuration d'utilisateur par défaut est :

https://webseald.example.com/

Gestionnaire de sites de confiance

Sélectionne la classe d'implémentation pour un gestionnaire de sites fiable. L'implémentation contient des données concernant les décisions d'accord d'authentification prises par un utilisateur lors des authentifications OpenID.

Prise en charge de l'identificateur OP

Indique si identifier_select est pris en charge lorsqu'un consommateur lance une connexion unique. Utilisez cette option si un fournisseur d'identité utilise XRDS. Si vous ne sélectionne pas cette option, toutes les autres options de identifier_select sont désactivées.

Vous pouvez utiliser une option de configuration pour activer ou désactiver la prise en charge de l'identificateur OP.

Pour activer l'option de configuration, vous devez modifier <was_config_root>/itfim/<tfim_domain>/etc/feds.xml sur le fournisseur d'identité pour ajouter ce paramètre :

```
<fc:EntityProperty name="OPENID.IPSupportOPIdentifier"

<fim:Value>true</fim:Value>

</fc:EntityProperty>
```

Si la valeur est manquante, la valeur par défaut est false.

Si la valeur de OPENID.IPSupportOPIdentifier est true, ce paramètre additionnel doit également être inclus :

```
<fc:EntityProperty name="OPENID.IPGeneratedClaimedIDPattern">

<fim:Value>see_below</fim:Value>

</fc:EntityProperty>
```

Utilisez un modèle de chaîne pour la valeur. Il contient le modèle@ID@ qui est remplacé par le nom de l'utilisateur.

Remarque : Le paramètre est similaire au paramètre Identity Pattern existant qui est utilisé pour vérifier les connexions de l'identificateur réclamées qui sont régulières (connexions autres que l'identificateur OP). L'exception est que le paramètre OPENID.IPGeneratedClaimedIDPattern n'est pas une expression régulière. Il s'agit simplement d'un modèle qui doit inclure la macro de remplacement @ID@. Par exemple, la valeur peut être la suivante :

https://myidp.com/@ID@

Modèle d'identificateur OP demandé créé

Spécifie une URL valide qui doit contenir la chaîne @ID@. Elle active une partie de confiance qui peut lancer une connexion unique avec un identificateur demandé paramétré sur identifier_select.

L'URL par défaut est l'URL du serveur de point de contact définie lors de la configuration de la fédération.

Par exemple, si l'URL de point de contact a été définie comme :

https://webseal.example.com/FIM

le modèle d'identificateur demandé généré via OP par défaut est le suivant :

```
https://webseal.example.com/@ID@
```

Options de reconnaissance de la partie de confiance
Propose deux options :

Exécuter une reconnaissance RP
Indique s'il faut tenter la reconnaissance de la partie de confiance. Si vous ne sélectionnez pas cette option, toutes les autres options de reconnaissance de la partie de confiance sont désactivées.

Réussite de la reconnaissance RP obligatoire
Indique si Tivoli Federated Identity Manager est interrompu avec un message d'erreur lorsqu'il ne peut pas exécuter la reconnaissance de la partie de confiance pour le fournisseur d'identité. Cette option s'applique uniquement si vous activez Perform RP Discovery (Exécuter la reconnaissance de la partie de confiance).

Délai d'expiration de la mise en cache de la reconnaissance RP
Détermine le nombre de secondes nécessaire pour mettre en mémoire cache les informations reconnues concernant les parties de confiance. Si vous saisissez une valeur inférieure à zéro, les informations ne sont jamais mises en mémoire cache.

Protocoles serveur OpenID autorisés
Définit les protocoles autorisés des serveurs OpenID avec lesquels l'agent d'utilisateur autorise l'établissement d'une connexion. Vous pouvez choisir l'une des valeurs ou les deux. Il est généralement recommandé de définir ce paramètre uniquement sur HTTPS.

Sélectionnez l'une ou l'autre des options suivantes, ou les deux :

- HTTPS
- HTTP

Délai d'attente de connexion HTTP
Spécifie le nombre de secondes avant l'expiration du délai d'attente durant les communications avec le client HTTP. Entrez un nombre positif. Si vous entrez zéro (0), le logiciel utilise les valeurs par défaut Java pour les objets URLConnection. La valeur par défaut est 30 secondes.

Fichier de clés
Indique le fichier de clés utilisé pour valider les certificats des noeuds finaux SSL au cours des communications pour la reconnaissance du correspondant associé. Ce fichier de clés doit contenir les certificats de signataire de droits certifiés de tous les correspondants associés pour lesquels la reconnaissance du correspondant associé doit être effectuée.

Règles de connexion agent d'utilisateur
Spécifie les règles de connexion de l'agent utilisateur. Vous devez sélectionner l'une des options suivantes.

- Autoriser l'accès aux hôtes OpenID par défaut

  L'hôte est contacté, à moins qu'il ne soit compris dans la liste de refus. Ce paramètre est plus tolérant et permet généralement aux utilisateurs de se connecter à partir d'un serveur OpenID légitime sur Internet.
- Refuser l'accès aux hôtes OpenID par défaut

Seuls les hôtes de la liste d'autorisation sont accessibles pour l'agent d'utilisateur. Ce paramètre est restrictif et chaque URL d'identité OpenID et chaque serveur dont vous souhaitez autoriser l'accès doit figurer dans les listes d'autorisation.

Pour passer en vue les différents choix de règles, consultez la rubrique «Règles relatives à l'agent d'utilisateur», à la page 366.

Expressions régulières de nom d'hôte autorisées
Indique une liste d'expressions régulières identifiant les noms d'hôtes auxquels l'agent d'utilisateur peut demander à accéder. Entrez une chaîne par ligne.

Par exemple :

```
.*\.ibm\.com
```

La valeur est facultative.

Adresses IP autorisées / Masques de réseau
Indique les adresses IP ou les masques de réseau auxquels l'agent d'utilisateur peut demander à accéder. Utilisez des expressions régulières, et entrez une chaîne par ligne. Entrez une chaîne par ligne.

Par exemple :

```
10.1.1.0/24
192.168.0.10
```

Cette valeur est facultative.

Module d'autorisation d'accès aux noeuds finaux dynamiques
Spécifie une liste de plug-ins d'accès de point final dynamiques personnalisés. Les plug-ins peuvent vérifier les listes externes d'hôtes dignes de confiance ou non. Ce paramètre est utilisé en plus de la configuration des règles de connexion agent d'utilisateur. Si l'option est définie sur l'approbation d'accès par défaut, les paramètres de configuration spécifiés sous les règles de connexion agent d'utilisateur sont utilisés.

Expressions régulières de nom d'hôte refusées
Indique les noms d'hôte auxquels l'agent d'utilisateur ne peut pas demander à accéder. Utilisez des expressions régulières, et entrez une chaîne par ligne.

- Lorsque l'option Règles de connexion agent d'utilisateur est paramétrée de manière à refuser l'accès aux hôtes OpenID par défaut, cette propriété n'est pas utilisée.
- Lorsque l'option Règles de connexion agent d'utilisateur est paramétrée de manière à autoriser l'accès aux hôtes OpenID par défaut, l'utilisation de cette propriété est facultative.

Par exemple :

```
.*\.example\.com
.*\.example2\.com
```

Adresses IP refusées / masques de réseau
Indique une liste d'expressions régulières identifiant les adresses IP ou les masques de réseau auxquels l'agent d'utilisateur ne peut pas demander à accéder. Entrez une chaîne par ligne.

- Lorsque l'option Règles de connexion agent d'utilisateur est paramétrée de manière à refuser l'accès aux hôtes OpenID par défaut, cette propriété n'est pas utilisée.

- Lorsque l'option Règles de connexion agent d'utilisateur est paramétrée de manière à autoriser l'accès aux hôtes OpenID par défaut, l'utilisation de cette propriété est facultative.

Par exemple :

```
11.12.13.0/24
192.168.0.10
```

Options de mappage d'identité

Sélectionnez une des options suivantes :

- **Utilisez des règles de mappage XSLT ou Javascript pour le mappage d'identité**

  Sélectionnez cette option lorsque vous créez une règle de mappage XSLT ou Javascript qui fournit des règles de mappage d'identité.

  Tivoli Federated Identity Manager fournit un exemple de fichier de règles de mappage d'identité destiné aux fédérations de fournisseurs d'identité OpenID :

  /répertoire_installation/examples/ip_openid.xsl

- **Utilisez Tivoli Directory Integrator pour le mappage**

  Sélectionnez cette option lorsque vous avez une chaîne d'assemblage Tivoli Directory Integrator pour le mappage d'identité requis par votre fédération OpenID.

- **Utiliser une instance de modèle de mappage personnalisé**

  Sélectionnez cette option lorsque vous avez un module de service d'accréditation personnalisé pour le mappage d'identité requis par votre fédération OpenID.

Tableau 109. Formulaire pour les propriétés d'identification d'une fédération

Propriété à spécifier	**Votre valeur**
Nom de la fédération	
Rôle	fournisseur d'identité
Nom de la société	
Protocole de fédération	OpenID
Serveur point de contact	
Délai d'expiration d'association	Valeur par défaut : 3 600 secondes.
Délai d'expiration de réponse de l'élément Nonce (secondes)	Valeur par défaut : 30 secondes
Générateur d'ID (génère la valeur @ID@)	
Structure de l'URL d'identité OpenID	
URL de configuration d'utilisateur	
Prise en charge de l'identificateur OP	Modèle d'identificateur OP demandé créé
Modèle d'identificateur OP demandé créé	
Exécuter une reconnaissance RP	

Tableau 109. Formulaire pour les propriétés d'identification d'une fédération (suite)

Propriété à spécifier	Votre valeur
Réussite de la reconnaissance RP obligatoire	
Délai d'expiration de la mise en cache de la reconnaissance RP	
Protocoles serveur OpenID autorisés	
Délai d'attente de connexion HTTP	
Fichier de clés	
Règles de connexion agent d'utilisateur	
Expressions régulières de nom d'hôte autorisées	
Adresses IP autorisées / Masques de réseau	
Module d'autorisation d'accès aux noeuds finaux dynamiques	
Expressions régulières de nom d'hôte refusées	
Adresses IP / Masques de réseau refusés	
Options de mappage d'identité	Sélectionnez l'une des options suivantes : • Utiliser XSLT ou JavaScript pour le mappage d'identité • Utiliser Tivoli Directory Integrator pour le mappage • Utiliser une instance de modèle de mappage personnalisé
Fichier de règles de mappage d'identité	Si vous utilisez XSLT ou JavaScript pour le mappage d'identité, spécifiez le nom du fichier de règle de mappage :
Module de mappage personnalisé	Si vous utilisez un module de mappage personnalisé, notez le nom du module :

Formulaire de configuration du consommateur

Tivoli Federated Identity Manager comprend un assistant qui vous guidera tout au long de la configuration des fédérations OpenID.

L'assistant vous invite à renseigner les propriétés nécessaires pour votre déploiement. Ce formulaire décrit les propriétés.

Ce formulaire vous permet de planifier vos propriétés et vous pouvez vous y référer lors de l'exécution de l'assistant.

Nom de la fédération
Chaîne arbitraire choisie pour dénommer cette fédération. Par exemple, `openid-consumer`.

Rôle de la fédération
Votre rôle est *Fournisseur de services*. Vous devez sélectionner l'option *fournisseur de services* lors de la configuration du rôle de consommateur.

Nom de la société
Valeur au format chaîne désignant le nom de la société. Vous pouvez éventuellement fournir des informations de contact supplémentaires.

Protocole de la fédération
OpenID.

Serveur point de contact
Adresse URL du serveur qui agit en tant que point de contact initial pour les requêtes entrantes. L'adresse est constituée d'une spécification de protocole, du nom d'hôte du serveur et (en option) d'un numéro de port. Lorsque WebSEAL est le serveur point de contact, la jonction WebSEAL est spécifiée. Exemple de valeur :

```
https://webseald.example.com/FIM
```

Remarque : Pour le support d'OpenID, le serveur point de contact doit utiliser le protocole SSL (Secure Socket Layer). L'adresse URL spécifiée doit être du type `https://`.

Certificat racine digne de confiance annoncé
Cette valeur est une adresse URL qui constitue la *racine* des URL accréditées pour la fédération. Elle renvoie par défaut à l'URL de base de la fédération, qui représente le chemin d'accès au nom d'hôte du service de protocole de connexion unique. Lorsque le numéro de port ne correspond pas à la valeur par défaut, la valeur du port est également incluse. La valeur doit se terminer par une barre oblique (/).

Cette valeur doit être une URL parente du noeud final délégué pour le renvoi du nom de connexion. Elle est utilisée en tant que paramètre openid.trust_root dans la requête de connexion unique adressée au fournisseur d'identité.

A titre d'exemple, lorsque vous avez précédemment spécifié, dans l'assistant, le serveur point de contact suivant :

```
https://webseald.example2.com/FIM
```

le certificat racine digne de confiance par défaut est :

```
https://webseald.example2.com/
```

Activer le protocole Yadis
Indique s'il faut effectuer la reconnaissance Yadis. Pour obtenir les meilleures pratiques, choisissez de désactiver cette option.

Activer des identificateurs XRI
Indique s'il faut résoudre les identificateurs demandés URL ou XRI. Si vous ne sélectionnez pas d'option, le logiciel utilise uniquement les identificateurs demandés URL.

Proxys XRI
Indique une liste d'URL permettant de résoudre les identificateurs XRI. L'URL doit contenir la macro `@XRI@`.

Expiration des informations reconnues
Indique le temps pendant lequel la mémoire cache stocke les informations reconnues. Si vous n'entrez pas de nombre positif, le cache est désactivé et la reconnaissance est effectuée à chaque connexion.

Délai de décalage de réponse de l'élément Nonce
Indique une valeur, en secondes, utilisée pour valider la réponse de l'élément Nonce provenant des fournisseurs d'identité OpenID 2.0. La validation est uniquement assurée si ce décalage est un nombre positif. La validation est assurée en prenant la durée de la réponse de l'élément Nonce et le décalage configuré de la réponse de l'élément Nonce.

Si le nombre de secondes n'est pas compris dans cette plage, la réponse d'authentification est refusée. Si le nombre de secondes est compris dans cette plage, une mémoire cache de la réponse de l'élément Nonce est contrôlée. Le contrôle permet de s'assurer que la réponse d'authentification ne correspond pas à une nouvelle lecture.

Lorsque la validation est effectuée avec succès, la réponse de l'élément Nonce est ajoutée à la mémoire cache de la réponse de l'élément Nonce tant qu'elle est comprise dans la période du décalage. La réponse de l'élément Nonce est ajoutée à la mémoire cache de l'élément Nonce afin de vérifier que les réponses d'authentification à venir ne sont pas de nouvelles lectures.

Protocoles serveur OpenID autorisés
Cette valeur représente l'ensemble des protocoles autorisés sur les serveurs OpenID pour lesquels l'agent d'utilisateur autorise les connexions. Il est généralement recommandé de définir ce paramètre uniquement sur HTTPS.

Sélectionnez l'une ou l'autre des options suivantes, ou les deux :

- HTTPS
- HTTP

La valeur HTTPS est configurée par défaut. Vous devez sélectionner au moins un protocole.

Délai d'attente de connexion HTTP
Cette valeur spécifié le délai d'attente de communication sur le client HTTP. Cette valeur doit être un entier positif valide. La valeur maximale est définie par la valeur entière la plus élevée. La valeur zéro (0) signifie que les valeurs Java par défaut doivent être appliquées aux objets URLConnection. La valeur par défaut est 30 secondes.

Fichier de clés
Cette valeur correspond au nom du fichier de clés qui a été précédemment configuré dans le service de clés de Tivoli Federated Identity Manager. Le magasin de clés doit contenir les certificats de signataires émis par l'autorité de certification uniquement.

Le client HTTP du consommateur utilise ce fichier de clés lors de la communication avec les fournisseurs d'identité compatibles avec SSL. Le magasin de clés est utilisé pour déterminer si l'hôte avec lequel la connexion doit être établie est digne de confiance. Cette vérification a lieu lors du traitement des messages `associate` et `check_authentication`.

Valeur par défaut :
`DefaultTrustedKeyStore`

Règles de connexion agent d'utilisateur
Cette valeur définir les règles des connexions établies par l'agent d'utilisateur. Vous devez sélectionner l'une des options suivantes.

- Autoriser l'accès aux hôtes OpenID par défaut
- Refuser l'accès aux hôtes OpenID par défaut

Pour passer en vue les différents choix de règles, consultez la rubrique «Règles relatives à l'agent d'utilisateur», à la page 366.

Expressions régulières de nom d'hôte autorisées
Liste des expressions régulières qui spécifient les noms d'hôte auxquels l'agent d'utilisateur peut demander l'accès. Entrez une chaîne par ligne. Par exemple :

```
.*\.ibm\.com
```

Cette valeur est facultative.

Adresses IP autorisées / Masques de réseau
Liste des expressions régulières qui spécifient les adresses IP et les masques de réseau auxquels l'agent d'utilisateur peut demander l'accès. Entrez une chaîne par ligne. Par exemple :

```
10.1.1.0/24
192.168.0.10
```

Cette valeur est facultative.

Expressions régulières de nom d'hôte refusées
Liste des expressions régulières qui spécifient les noms d'hôte auxquels l'agent d'utilisateur ne peut pas demander l'accès. Entrez une chaîne par ligne. Par exemple :

```
.*\.example\.com
.*\.example2\.com
```

- Lorsque le paramètre Règles de connexion agent d'utilisateur est défini sur **Refuser l'accès aux hôtes OpenID par défaut**, cette propriété est ignorée.
- Lorsque le paramètre Règles de connexion agent d'utilisateur est défini sur **Autoriser l'accès aux hôtes OpenID par défaut**, l'usage de cette propriété est facultatif.

Adresses IP / Masques de réseau refusés
Liste des expressions régulières qui spécifient les adresses IP et les masques de réseau auxquels l'agent d'utilisateur ne peut pas demander l'accès. Entrez une chaîne par ligne. Par exemple :

```
11.12.13.0/24
192.168.0.10
```

- Lorsque le paramètre Règles de connexion agent d'utilisateur est défini sur **Refuser l'accès aux hôtes OpenID par défaut**, cette propriété est ignorée.
- Lorsque le paramètre Règles de connexion agent d'utilisateur est défini sur **Autoriser l'accès aux hôtes OpenID par défaut**, l'usage de cette propriété est facultatif.

Module d'autorisation d'accès aux noeuds finaux dynamiques
Spécifie une liste de plug-ins d'accès de point final dynamiques personnalisés. Les plug-ins peuvent vérifier les listes externes d'hôtes dignes de confiance ou non. Ce paramètre est utilisé en plus de la configuration des règles de connexion agent d'utilisateur. Si l'option est

définie sur l'approbation d'accès par défaut, les paramètres de configuration spécifiés sous les règles de connexion agent d'utilisateur sont utilisés.

Options de mappage d'identité

Vous êtes invité à sélectionner l'une des options suivantes :

- Utilisez des règles de mappage XSLT ou Javascript pour le mappage d'identité

  Sélectionnez cette option lorsque vous avez créé une règle de mappage XSLTfile ou Javascript qui fournit des règles de mappage d'identité.

  Tivoli Federated Identity Manager fournit un exemple de fichier de règles de mappage d'identité destiné aux fédérations de consommateurs OpenID :

  `/répertoire_installation/examples/sp_openid.xsl`
- Utiliser Tivoli Directory Integrator pour le mappage

  Sélectionnez cette option lorsque vous avez préalablement configuré une chaîne d'assemblage Tivoli Directory Integrator pour le mappage d'identité requis par votre fédération OpenID.
- Utiliser une instance de modèle de mappage personnalisé

  Sélectionnez cette option lorsque vous avez écrit et déployé un module de service d'accréditation personnalisé pour le mappage d'identité requis par votre fédération OpenID.

Tableau 110. Propriétés de configuration de consommateur OpenID

Propriété	**Votre valeur**
Nom de la fédération	
Rôle	fournisseur de services
Nom de la société	
Protocole de fédération	OpenID
URL de serveur point de contact	
Certificat racine digne de confiance annoncé	
Activer le protocole Yadis	
Activer des identificateurs XRI	
Proxys XRI	
Expiration des informations reconnues	
Délai de décalage de réponse de l'élément Nonce	
Protocoles serveur OpenID autorisés	`HTTPS`, `HTTP` ou les deux
Délai d'attente de connexion HTTP (secondes)	Valeur par défaut : 30 secondes
Fichier de clés	
Règles de connexion agent d'utilisateur	
Expressions régulières de nom d'hôte autorisées	

Tableau 110. Propriétés de configuration de consommateur OpenID (suite)

Propriété	Votre valeur
Adresses IP autorisées / Masques de réseau	
Expressions régulières de nom d'hôte refusées	
Adresses IP / Masques de réseau refusés	
Module d'autorisation d'accès aux noeuds finaux dynamiques	
Options de mappage d'identité	Sélectionnez l'une des options suivantes : • Utiliser XSL pour le mappage d'identité • Utiliser Tivoli Directory Integrator pour le mappage • Utiliser une instance de modèle de mappage personnalisé
Fichier de règles de mappage d'identité	Si vous utilisez XSL pour le mappage d'identité, spécifiez le nom de fichier de règle de mappage suivant :
Module de mappage personnalisé	Si vous utilisez un module de mappage personnalisé, notez le nom du module :

Chapitre 25. Configuration de OpenID

Configurez une fédération OpenID en créant la fédération, en configurant le serveur point de contact, et en mettant à jour les informations sur les pages de connexion.

Vérification des dépendances OpenID

Vérifiez que les exigences de création d'une fédération OpenID Card sont respectées.

Avant de commencer

Avant d'utiliser l'assistant de création de fédération, assurez-vous que les conditions relatives aux dépendances OpenID Card sont satisfaites. Effectuez les activités de planification requises en passant en revue le contenu de la section relative à la planification.

Procédure

1. Définissez votre stratégie concernant le mappage d'identité.
   - Si vous utilisez un fichier de règles de mappage, assurez-vous que les règles de mappage XSLT ou Javascript nécessaires ont été inclus afin de répondre aux exigences de votre déploiement.
   - Si vous utilisez une chaîne d'assemblage de Tivoli Directory Integrator, assurez-vous que celle-ci a été construite.
   - Si vous utilisez un module de mappage personnalisé, assurez-vous qu'il a été développé et testé.
2. Assurez-vous que vous avez établi les règles de l'agent d'utilisateur pour le consommateur et le fournisseur d'identité.
3. renseignez le formulaire de la fédération. Procédez à l'une des actions suivantes :
   - «Formulaire de configuration du fournisseur d'identité», à la page 375
   - «Formulaire de configuration du consommateur», à la page 381

Configuration d'une fédération OpenID

L'assistant de fédération permet de créer et de configurer une fédération OpenID.

Avant de commencer

Assurez-vous que vous avez préparé les informations de configuration avant de créer la fédération au moyen de l'assistant.

Pourquoi et quand exécuter cette tâche

Pour utiliser l'assistant de fédération afin de créer et configurer une fédération OpenID, procédez comme suit :

Procédure

1. Connectez-vous à la Integrated Solutions Console.

2. Cliquez sur **Tivoli Federated Identity Manager** > **Configuration de la connexion unique fédérée** > **Fédérations**. Les portlets Domaine en cours et Fédérations s'ouvrent.
3. Cliquez sur **Créer**. L'assistant de fédération démarre. L'assistant affiche une série de panneaux de configuration.
4. Utilisez votre formulaire complété afin d'indiquer des valeurs dans chaque panneau.
5. Entrez les valeurs demandées.
   a. La première série de panneaux vous demande d'indiquer les paramètres relatifs au nom, au rôle et serveur point de contact de la fédération.
   b. Puis, le panneau de configuration OpenID vous invite à indiquer les valeurs requises pour un fournisseur d'identité ou un consommateur OpenID.
   c. La dernière série de panneaux vous invite à indiquer les paramètres de configuration du mappage d'identité.

   Lorsque vous avez terminé d'entrer les paramètres de configuration, le panneau Récapitulatif s'affiche.
6. Cliquez sur **Suivant** pour passer au panneau suivant. Si vous avez besoin de revenir en arrière pour ajuster un paramètre de configuration, cliquez sur **Précédent**. Pour obtenir des informations sur des zones spécifiques, affichez l'aide en ligne.
7. Vérifiez que les paramètres de configuration sont corrects.
8. Cliquez sur **Terminer**. Le portlet Création de fédération terminée s'affiche.

Configuration de l'amélioration des performances pour OpenID

Utilisez les paramètres OpenID pour améliorer les performances dans un scénario de liste blanche.

Pourquoi et quand exécuter cette tâche

Les scénarios de connexion unique où une partie de confiance comprend un fournisseur d'identité OpenID dans une liste blanche peuvent utiliser deux paramètres pour améliorer les performances. Ces paramètres sont les suivants :

- `OPENID.DiscoveredInformationExpirationSeconds` - Durée pendant laquelle une partie de confiance OpenID place en mémoire cache les informations de reconnaissance extraites d'un ID fournisseur OpenID. Ce paramètre peut enregistrer un aller-retour de communications directes pour chaque connexion après la première connexion.

  La reconnaissance n'est effectuée qu'une seule fois au cours de la période d'expiration d'un ID de fournisseur OpenID donné.

  Dans les environnements utilisant un fournisseur OpenID bien connu, ce paramètre peut améliorer les performances d'authentification.
- `OPENID.SkipClaimedIdDiscovery` - Lorsque ce paramètre est défini sur `true`, la partie utilisatrice n'effectue aucune reconnaissance sur un identificateur demandé renvoyé par un fournisseur OpenID au cours d'une connexion d'ID de fournisseur OpenID. Par exemple, lorsque les authentifications utilisent `identifier_select`.

  Pour des raisons de sécurité, n'utilisez ce paramètre que lorsque les seuls fournisseurs que vous pouvez contacter avec la partie utilisatrice sont des fournisseurs de confiance figurant sur la liste blanche.

  L'activation de cette option permet d'enregistrer un aller-retour de communications directes pour chaque connexion. Ce paramètre est activé en

général dans un environnement intranet où de nombreuses parties utilisatrices se servent d'un fournisseur OpenID courant.

Ce paramètre est souvent combiné à `OPENID.DiscoveredInformationExpirationSeconds`.

Procédure

1. Modifiez le fichier `<was_config_root>/itfim/<tfim_domain>/etc/feds.xml`.
2. Ajoutez des paramètres de configuration aux paramètres *Self* pour la fédération de la partie utilisatrice OpenID.
3. Insérez un texte similaire à celui-ci :

```
<fc:EntityProperty name="OPENID.DiscoveredInformationExpirationSeconds">
<fim:Value>604800</fim:Value>
</fc:EntityProperty>
<fc:EntityProperty name="OPENID.SkipClaimedIdDiscovery">
<fim:Value>true</fim:Value>
</fc:EntityProperty>
```

Pour savoir où ajouter les paramètres de configuration, il suffit de rechercher le paramètre `OPENID.AuthenticationMode` et d'ajouter les paramètres de configuration à la suite de celui-ci.

Configuration d'un serveur point de contact WebSEAL pour une fédération Open ID

Si vous envisagez d'utiliser WebSEAL en tant que serveur point de contact, vous devez le configurer pour la fédération OpenID.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Ces instructions ont pour hypothèse que le profil du point de contact WebSEAL est activé.

Pourquoi et quand exécuter cette tâche

Le portlet Création de fédération terminée comporte un bouton qui vous permet d'obtenir un Tivoli Federated Identity Manager utilitaire de configuration. Vous devez obtenir cet outil, puis l'exécuter. Pour configurer WebSEAL en tant que serveur point de contact, procédez comme suit :

Procédure

1. Une fois la fédération créée, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour recharger vos modifications.

Remarque : La console de gestion vous offre la possibilité d'ajouter immédiatement un partenaire, mais pour cette configuration initiale de la fédération, vous devez d'abord exécuter d'autres tâches.

2. Cliquez sur **Terminé** pour revenir au panneau Fédérations.
3. Cliquez sur **Télécharger l'outil de configuration Tivoli Access Manager**.
4. Enregistrez l'outil de configuration sur le système de fichiers de l'ordinateur hébergeant le serveur WebSEAL.
5. Démarrez l'outil de configuration depuis une ligne de commande. La syntaxe est la suivante :

```
java -jar /download_dir/tfimcfg.jar -action tamconfig
-cfgfile webseald-instance_name.conf
```

Remarque : Si la norme FIPS (Federal Information Processing Standards) est activée pour votre environnement, une fabrique de connexions sécurisées doit être indiquée. Par exemple :

```
java -jar /download_dir/tfimcfg.jar -action tamconfig
-cfgfile webseald-instance_name.conf -sslfactory TLS
```

Vous aurez besoin de l'ID (par défaut : sec_master) et du mot de passe de l'utilisateur d'administration Tivoli Access Manager. L'utilitaire configure les noeuds finals sur le serveur WebSEAL, crée une jonction WebSEAL, relie les listes de contrôle d'accès adaptées et active les méthodes d'authentification adéquates.

Exemple

Par exemple, lorsque vous avez mis le fichier tfimcfg.jar dans le répertoire `/tmp` et que le nom de l'instance WebSEAL est `default`, la commande (spécifiée sur une ligne ininterrompue) est la suivante :

```
java -jar /tmp/tfimcfg.jar -action tamconfig
-cfgfile /<chemin_qualifié_complet>/webseald-default
```

Pour plus d'informations, voir Annexe A, «Référence de tfimcfg», à la page 827.

Configuration de WebSphere en tant que serveur point de contact

Tivoli Federated Identity Manager est configuré pour utiliser par défaut WebSphere Application Server WebSEAL en tant que serveur point de contact. Pour configurer WebSphere en tant que serveur point de contact, vous devez procéder à une modification de la configuration.

Procédure

1. Connectez-vous à la console d'administration.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Sélectionnez **WebSphere**.
4. Cliquez sur **Activer**.

Résultats

Le serveur WebSphere est désormais configuré en tant que point de contact.

Configuration des pages de connexion

Lors de la configuration d'un serveur point de contact, il convient de configurer les informations figurant sur les pages de connexion.

- Les consommateurs doivent fournir un formulaire de connexion destinés à être présentés à l'utilisateur final.

  Les administrateurs qui utilisent WebSEAL comme serveur de point de contact peuvent modifier la page login.html par défaut de WebSEAL.
- Les fournisseurs d'identité doivent fournir des informations de reconnaissance à l'aide de la reconnaissance HTML ou Yadis. Les informations de reconnaissance sont fournies dans l'URL d'identité OpenID de l'utilisateur ou l'URL d'identificateur de fournisseur d'identité, ou les deux.

Chapitre 26. Référence OpenID

Cette section contient la liste des références pour OpenID. Elle décrit les algorithmes et les transports pris en charge, ainsi que les modèles de pages qui sont utilisés dans un flux de connexion unique.

Algorithmes et modes de transport pris en charge

Tivoli Federated Identity Manager prend en charge les spécifications OpenID pour le type de session à secret partagé (association) :

- OpenID 1.1
  - Texte en clair
  - DH-SHA1

  Pour des raisons de sécurité, le support (consommateur) du fournisseur de services Tivoli Federated Identity Manager pour OpenID 1.1 émet uniquement des demandes de session de type DH-SHA1.
- OpenID 2.0
  - DH-SHA256
  - DH-SHA1
  - no-encryption

  Le consommateur Tivoli Federated Identity Manager tente le type DH-SHA256 par défaut.

  Lorsqu'un fournisseur d'identité renvoie une erreur indiquant qu'un type de session demandé est non pris en charge, le fournisseur d'identité peut établir quels types de session sont pris en charge. Dans ce cas, le consommateurTivoli Federated Identity Manager tente d'établir le type de session suggéré.

  Remarque : Le consommateur Tivoli Federated Identity Manager tente d'utiliser le type 'no-encryption' uniquement lorsque le serveur OpenID est un noeud final SSL.

Il convient que les noeuds finals de fournisseur d'identité utilisés par les consommateurs pour accéder à OpenID soient configurés en SSL.

Dans la plupart des déploiements, des noeuds finals non protégés (par exemple utilisant le protocole HTTP au lieu de HTTPS) sont utilisés pour la résolution de l'URL d'identité d'un utilisateur. Il convient que les adresses URL suivantes, renvoyées sous forme de liens à en-tête HTML, utilisent le protocole SSL :

- openid.server
- openid2.provider

Il convient que les noeuds finals de consommateur soient configurés en HTTPS (SSL).

Modèle de page pour la promotion d'un serveur OpenID

Les spécifications d'authentification OpenID établissent que lorsqu'un fournisseur d'identité utilise une URL de connexion unique, une notification doit être renvoyée en cas de réception d'une requête HTTP GET dépourvue de paramètres (conformément à la spécification OpenID 1.1). La page à renvoyer doit contenir le texte suivant :

```
Il s'agit d'un noeud final du serveur OpenID. Pour plus d'informations, voir http://openid.net/
```

Tivoli Federated Identity Manager fournit le fichier openid_server.html. Le fichier ne contient aucune macro remplaçable.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <title>Serveur OpenID</title>
  </head>
  <body>
    Il s'agit d'un noeud final du serveur OpenID. Pour plus d'informations,
voir <a href="http://openid.net/">http://openid.net/</a>
  </body>
</html>
```

Figure 32. Modèle de fichier openid_server.html

Ce modèle n'est défini que sur le fournisseur d'identité.

Modèle de page pour le consentement d'authentification

Utilisez le modèle de page pour le consentement d'authentification au niveau du fournisseur d'identité pour déterminer et enregistrer les informations de consentement de l'utilisateur relatives aux droits d'authentification dans un consommateur spécifique. Il est également utilisé pour indiquer les attributs facultatifs à partager avec le consommateur.

Lors s"une opération OpenID checkid_setup, l'utilisateur est redirigé vers le fournisseur d'identité afin de valider l'état de sa connexion. A ce moment-là, le fournisseur d'identité de mander à l'utilisateur la permission de fournir les informations d'authentification et d'attribut au site de consommation. Le fournisseur d'identité de Tivoli Federated Identity Manager délivre un modèle de page HTML appelé consent.html.

Tivoli Federated Identity Manager garde en mémoire les décisions de confiance qu'un utilisateur prend à l'égard d'un site de consommation particulier, en conservant les données sous forme de `trust_root` ou de `realm`. La sauvegarde de ces connaissances permet à Tivoli Federated Identity Manager de ne pas avoir à inviter l'utilisateur à se connecter à chaque fois au même consommateur.

La page de consentement affiche la liste des attributs que la requête de connexion unique (émise par le consommateur) a indiqués comme étant *requis* ou *facultatifs*.

Comme la longueur de ces listes est indéterminée, le modèle prend en charge de multiples copies de sections répétées une fois pour chaque attribut dans la liste concernée. La prise en charge des sections répétitives est permise par la spécification d'une extension d'enregistrement simple.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

Ce fichier modèle prend en charge plusieurs macros de remplacement :

@OPENID_TRUSTURL@
Cette macro est remplacée par le paramètre `openid.trust_root` dans la requête checkid_setup.

@OPENID_POLICYURL@
Cette macro est remplacée par le paramètre `openid.sreg_url` dans la requête checkid_setup lorsque l'adresse URL existe. Si l'adresse URL est inexistante, la valeur est représentée par une chaîne vide.

@OPENID_IDENTITYURL@
Cette macro est remplacée par le paramètre `openid.identity` dans la requête checkid_setup.

@OPENID_SSOURL@
Cette macro est remplacée par le noeud final du délégué de serveur OpenID (noeud final) sur le fournisseur d'identité. Cette valeur est utilisée pour le paramètre d'action FORM afin d'envoyer par requête POST les résultats du formulaire de consentement vers le serveur OpenID.

@OPENID_RETURN_TO_VALIDATED@
Cette macro est remplacée par true ou false pour informer l'utilisateur de toute validation d'URL return_to dans le cadre de la reconnaissance de la partie de confiance.

@REQUIRED_ATTRIBUTE@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT requiredAttrs]`. Les valeurs affichent la liste des attributs obligatoires spécifiés par le fournisseur de services pour l'extension d'enregistrement simple. Cette macro est remplacée par chaque valeur contenue dans le paramètre `openid.sreg.required` de la requête, précédée de la chaîne de préfixe `openid.sreg.`.

@OPTIONAL_ATTRIBUTE@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT optionalAttrs]`. Les valeurs affichent la liste des attributs facultatifs spécifiés par le fournisseur de services pour l'extension d'enregistrement simple. Cette macro est remplacée par chaque valeur contenue dans le paramètre `openid.sreg.optional` de la requête, précédée de la chaîne de préfixe `openid.sreg.`.

Les attributs facultatifs nécessitent une attention particulière. Le fournisseur d'identité permet aux utilisateurs de spécifier individuellement les attributs facultatifs qu'ils peuvent envoyer à un consommateur spécifié. Les préférences utilisateur sont dénotées par les paramètres 'true' ou 'false' pour chaque attribut facultatif, comme spécifié dans le formulaire contenu dans la page HTML de consentement d'authentification. Pour permettre l'activation de cette fonctionnalité, le nom du paramètre doit *obligatoirement* commencer par le préfixe `optattr_` et se finir par le nom complet de l'attribut facultatif.

Par exemple :

```
optattr_openid.sreg.email=true&optattr_openid.sreg.nickname=false
```

La figure suivante illustre un exemple de traitement des attributs facultatifs.

```
Les attributs facultatifs suivants ont été demandés. Veuillez sélectionner
les attributs que vous vous préparer à envoyer, ou sélectionnez l'option
"Tous les attributs facultatifs" :<br /><br />
         <input id="chk_all_optional_attributes" type="checkbox"
          checked="checked" name="all_optional_attributes"
          onClick="allOptionalAttributes()" />
         <label for="chk_all_optional_attributes">Tous les attributs optionnels
         </label><br /><br />
     [RPT optionalAttrs]
             <input id="chk_@OPTIONAL_ATTRIBUTE@" type="checkbox"
name="optattr_@OPTIONAL_ATTRIBUTE@" onClick="oneOptionalAttribute()" />
             <label for="chk_@OPTIONAL_ATTRIBUTE">@OPTIONAL_ATTRIBUTE@
             </label><br />
     [ERPT optionalAttrs]
```

Figure 33. Traitement du consentement lié aux attributs facultatifs individuels

Remarque : Le paramètre d'entrée de la case à cocher présente dans le formulaire génère le nom à l'aide du préfixe `optattr_` et le nom de l'attribut facultatif. Pour chaque attribut facultatif, contenu dans la requête émise par le fournisseur de services, le code responsable du traitement de ce formulaire au niveau du fournisseur d'identité recherche un paramètre de type `optattr_<nom_attribut>`. Le fournisseur d'identité traite ensuite la valeur en tant que `true` ou `false`. Une valeur 'true' indique le consentement vis-à-vis de l'attribut facultatif. Lorsqu'un paramètre est absent dans le formulaire envoyé, le consentement adopte la valeur 'false'.

L'un des scénarios de déploiement possibles consiste à déployer un *portail de données personnelles* destiné aux utilisateurs individuels. Les utilisateurs peuvent avoir plusieurs identités créées lorsqu'ils utilisent un *portail de données personnelles*. Chaque personne peut avoir plusieurs ensembles d'attributs gérés dans un magasin de données externe. Cette fonctionnalité permet à l'utilisateur d'associer des données personnelles particulières à un consommateur OpenID particulier. Cela donne aux utilisateurs la possibilité de sélectionner des attributs personnels lorsque l'utilisateur se connecte au consommateur spécifié.

A titre d'exemple, vous pouvez utiliser le fournisseur d'identité pour créer, nommer et alimenter dynamiquement des ensembles d'attributs pour chaque donnée personnelle.

Ce scénario est possible grâce à l'usage d'un paramètre facultatif FORM appelé `userdata`. `userdata` peut être une liste de menu qui permet à l'utilisateur de sélectionner les données personnelles à partir desquelles les attributs sont renseignés.

Lorsque le paramètre `userdata` figure dans le formulaire d'entrée, la valeur de chaîne de l'adresse URL est incluse dans les réclamations adressées au service STS lors du mappage d'identité.

L'exemple de code suivant illustre le fichier modèle HTML consent.html.

Ce modèle n'est défini que sur le fournisseur d'identité.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>OpenID Consent-to-Authenticate</title>
<script type="text/javascript">

// when "All optional attributes" is selected,
uncheck any checked individual optional attributes
function allOptionalAttributes() {
    var theForm = document.forms[0];
    for (i = 0; i < theForm.elements.length; i++) {
        if (theForm.elements[i].type == "checkbox") {
            var cbName = theForm.elements[i].name;
            if (cbName.indexOf("optattr_") == 0) {
                theForm.elements[i].checked = false;
            }
        }
    }
}

// when an individual optional attribute is selected, be sure to
uncheck "All optional attributes"
function oneOptionalAttribute() {
    document.forms[0].all_optional_attributes.checked = false;
}

// utility function to show a section
function showDiv(f) {
  if (f.style) {
    f.style.display='block';
  }
}

</script>
  </head>
  <body>
    Ce site de consommation a demandé une ouverture de session OpenID pour vous :
 <b>@OPENID_TRUSTURL@</b>
    <p />
    Vous trouverez la stratégie du site de consommation, à l'adresse : <b>@OPENID_POLICYURL@</b>
    <p />


<script type="text/javascript">
    //
    // RP-discovery information
    //
    var txtWarningReturnTo = "WARNING: The return_to URL for the site has not
been successfully validated using relying-party discovery";
    var returntoValidated = @OPENID_RETURN_TO_VALIDATED@;
    if (!returntoValidated) {
        document.write(txtWarningReturnTo);
    }
</script>


    <p />
    Votre URL d'identité est : <b>@OPENID_IDENTITYURL@</b>
<script type="text/javascript">
    //
    // Display claimed identifier if different from identity URL
(e.g. if delegation was being used)
    //
    var txtClaimedID = "Your claimed identifier is: ";
```

```
    var identityurl = "@OPENID_IDENTITYURL@";
    var claimedid = "@OPENID_CLAIMEDID@";
    if (claimedid != identityurl) {
        document.write("<p />");
        document.write(txtClaimedID);
        document.write("<b>");
        document.write(claimedid);
        document.write("</b>");
    }
</script>


    <p />
<script type="text/javascript">
    //
    // PAPE information
    //
    var txtMaxAuthnAge = "Requested Maximum Authentication Age (seconds): ";
    var txtRequestedAuthnPolicies = "Requested Authentication Policies";
    var txtRequestedAssuranceLevels = "Requested Assurance Levels";

    var nopii = false;
    var maxAuthenticationAge = @MAXIMUM_AUTHENTICATION_AGE@;
    if ( maxAuthenticationAge >= 0) {
        document.write("<p/>" + txtMaxAuthnAge + maxAuthenticationAge);
    }

    var strAuthPolicies = "";
    [RPT authenticationPolicies]
        strAuthPolicies += "@REQUESTED_AUTHENTICATION_POLICY@"+",";
    [ERPT authenticationPolicies]
    if (strAuthPolicies.length > 0) {
        // strip last comma and split into array
        strAuthPolicies =
strAuthPolicies.substring(0,strAuthPolicies.lastIndexOf(","));
        var authPolicies = strAuthPolicies.split(",");
        document.write("<p/>");
        document.write("<table border>");
        document.write("<tr><th>" + txtRequestedAuthnPolicies + "</th></tr>");
        for (var i = 0; i < authPolicies.length; i++) {
            document.write("<tr><td>"+authPolicies[i]+"</td></tr>");

            // check if this is the nopii policy
            if (authPolicies[i] ==
"http://www.idmanagement.gov/schema/2009/05/icam/no-pii.pdf") {
                nopii = true;
            }
        }
        document.write("</table>");
    }

    var strAssuranceLevels = "";
    [RPT assuranceLevels]
        strAssuranceLevels += "@REQUESTED_ASSURANCE_LEVEL@"+",";
    [ERPT assuranceLevels]

    if (strAssuranceLevels.length > 0) {
        // strip last comma and split into array
        strAssuranceLevels =
strAssuranceLevels.substring(0,strAssuranceLevels.lastIndexOf(","));
        var assuranceLevels = strAssuranceLevels.split(",");
        document.write("<p/>");
        document.write("<table border>");
        document.write("<tr><th>" + txtRequestedAssuranceLevels + "</th></tr>");
        for (var i = 0; i < assuranceLevels.length; i++) {
            document.write("<tr><td>"+assuranceLevels[i]+"</td></tr>");
        }
```

```
        document.write("</table>");
    }

</script>


    <form action="@OPENID_SSOURL@" method="post">
      <input type="hidden" name="openid.mode" value="consent_to_authenticate" />
      <div id="DIV_ATTRIBUTES" name="DIV_ATTRIBUTES" style="display: none;">
        Les attributs obligatoires suivants ont été demandés :<br />
        <ul>
        [RPT requiredAttrs]
            <li>@REQUIRED_ATTRIBUTE@</li>
        [ERPT requiredAttrs]
        </ul>
        <p />
        Les attributs facultatifs suivants ont été demandés. Veuillez sélectionner
les attributs que vous vous préparer à envoyer, ou sélectionnez l'option
"Tous les attributs facultatifs" :<br /><br />
            <input id="chk_all_optional_attributes" type="checkbox"
checked="checked" name="all_optional_attributes"
onClick="allOptionalAttributes()" />
            <label for="chk_all_optional_attributes">
All Optional Attributes</label><br /><br />
        [RPT optionalAttrs]
                <input id="chk_@OPTIONAL_ATTRIBUTE@" type="checkbox"
name="optattr_@OPTIONAL_ATTRIBUTE@" onClick="oneOptionalAttribute()" />
                <label for="chk_@OPTIONAL_ATTRIBUTE@"
>@OPTIONAL_ATTRIBUTE@</label><br />
        [ERPT optionalAttrs]
      </div>
      <p />
      Souhaitez vous ouvrir une session sur ce site, en envoyant tous les attributs
nécessaires et en sélectionnant les attributs facultatifs ?
      <div>
        <input id="rd_permit_forever" type="radio"
name="consent" value="permit_forever"
checked="checked" /><label for="rd_permit_forever">
Allow Authentication forever
(add to my trusted sites)</label><br/>
        <input id="rd_permit_once" type="radio"
name="consent" value="permit_once" />
<label for="rd_permit_once">Allow Authentication this time only</label><br />
        <input id="rd_deny_once" type="radio" name="consent"
value="deny_once" />
<label for="rd_deny_once">Do not authenticate to this
site this time only</label><br />
        <input id="rd_deny_forever" type="radio" name="consent"
value="deny_forever" />
<label for="rd_deny_forever">Do not ever authenticate to this site
(add to my untrusted sites)</label><br />
      </div>
      <p /><label for="tx_userdata">Données utilisateur ou données personnelles :</label>
<input id="tx_userdata" type="text" name="userdata" />
      <p /><input type="submit" name="submit" value="Submit" />
    </form>
<script type="text/javascript">
  //
  // if the nopii policy was requested, leave the attribute information hidden
(as we shouldn't send it), otherwise show it
  //
  if (!nopii) {
    showDiv(document.getElementById("DIV_ATTRIBUTES"));
  }
```

```
</script>

  </body>
</html>
```

Page de modèle HTML pour la gestion des sites dignes de confiance

Cette page est utilisée en fournisseur d'identité. La page HTML est utilisée pour gérer l'ensemble permanent de sites sécurisés ou non sécurisés. L'utilisateur établit les sites via la page consent.html lors des opérations de connexion.

Les fonctionnalités du fournisseur d'identité OpenID incluent la possibilité de stocker et extraire certains attributs de préférences utilisateur, tels que :

- Site de consommation particulier ou non, tel qu'identifié par la valeur `trust_root`. Les valeurs d'accréditation peuvent être `once`, `never` ou `always`.
- La liste des attributs facultatifs pouvant être envoyés à un consommateur de confiance particulier.
- Les données de préférences utilisateur éventuellement choisies par le fournisseur d'identité lors de la création d'un ensemble d'attributs pour une requête de connexion unique adressée à un consommateur. Le détail optionnel peut par exemple contenir un index d'informations personnelles.

  Tivoli Federated Identity Manager est doté d'un mécanisme de stockage des attributs sous forme de cookies persistants dans le navigateur.

Le serveur Tivoli Federated Identity Manager inclut un modèle de page et le code de prise en charge. Le modèle de page et le code de prise en charge utilisent l'interface pour stocker et extraire des informations sur les sites dignes de confiance. Les utilisateurs peuvent utiliser le modèle de page pour afficher et gérer cette liste.

Le fichier modèle porte le nom sitemanager.html.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

Le modèle comprend les macros de remplacement suivantes :

@USERNAME@
: Cette macro est remplacée par le nom d'utilisateur Tivoli Federated Identity Manager.

@SITE_NAME@
: Cette macro à valeurs multiples est utilisée dans une liste de remplacement répétable `[RPT trustedSites]` ou `[RPT untrustedSites]`. Elle est utilisée pour afficher des informations sur les sites configurés dans un des états suivants :

  - Toujours digne de confiance
  - Refusé définitivement

  Cette macro affiche l'URL `trust_root` du site sécurisé ou non sécurisé.

@REQUIRED_ATTRIBUTES@
: Cette macro à valeurs multiples est utilisée dans une liste de remplacement répétable `[RPT trustedSites]`. La macro est utilisée pour afficher une liste

séparée par des virgules de l'ensemble spécifique d'attributs requis que l'utilisateur doit envoyer au consommateur.

@OPENID_SITEMANAGERURL@
Cette macro est remplacée par l'URL de noeud final du gestionnaire de site délégué servant à traiter l'action `remove` sur les sites dignes de confiance.

@ALL_OPTIONAL_ATTRIBUTES@
Cette macro à valeurs multiples est utilisée dans une liste de remplacement répétable `[RPT trustedSites]` pour le site digne de confiance. La macro est utilisée pour indiquer si l'utilisateur est préparé à l'envoi de tous les attributs facultatifs demandés à ce consommateur. Les valeurs admises sont `true` et `false`.

@LISTED_OPTIONAL_ATTRIBUTES@
Cette macro à valeurs multiples est utilisée dans une liste de remplacement répétable `[RPT trustedSites]`. La macro est utilisée pour afficher une liste séparée par des virgules de l'ensemble spécifique d'attributs facultatifs que l'utilisateur est préparé à envoyer au consommateur. Cette valeur est une chaîne non vide lorsque `@ALL_OPTIONAL_ATTRIBUTES@` a la valeur `false` pour le site digne de confiance. Lorsque `@ALL_OPTIONAL_ATTRIBUTES@` vaut `true`, cette valeur est une chaîne vide.

@USERDATA@
Cette macro à valeurs multiples est utilisée dans une liste de remplacement répétable `[RPT trustedSites]`. La macro est utilisée pour afficher les données utilisateur facultatives. Les données peuvent être spécifiées par un utilisateur lorsqu'il traite la page de consentement d'authentification pour accréditer définitivement ce site. Lorsqu'aucune donnée d'utilisateur n'est spécifiée, la valeur de la macro est une chaîne vide

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Gestionnaire de site OpenID</title>
 </head>
 <body>
  OpenID Site Manager</titleb>@USERNAME@</b>
  <p/>
  Sites de confiance<br />
  <table border="1">
   <tr><td>Site</td><td>Required Attributes</td><td>All Optional
Attributes?</td><td>Permitted Optional Attributes</td><td>User
Data</td><td>Action</td></tr>
[RPT trustedSites]
   <tr>
    <td>@SITE_NAME@</td>
    <td>@REQUIRED_ATTRIBUTES@</td>
    <td>@ALL_OPTIONAL_ATTRIBUTES@</td>
    <td>@LISTED_OPTIONAL_ATTRIBUTES@</td>
    <td>@USERDATA@</td>
    <td><a href="@OPENID_SITEMANAGERURL@?action=
remove&site=@SITE_NAME@">Remove</a></td>
   </tr>
 [ERPT trustedSites]
 </table>
 <p/>
 Sites non sécurisés<br />
 <table border="1">
 <tr><td>Site</td><td>Action</td></tr>
 [RPT untrustedSites]
 <tr>
  <td>@SITE_NAME@</td>
  <td>@OPENID_SITEMANAGERURL@?
action=remove&site=@SITE_NAME@">Remove</a></td>
 </tr>
[ERPT untrustedSites]
 </table>
</body>
</html>
```

Figure 34. Modèle de fichier HTML sitemanager.html

Ce modèle n'est défini que sur le fournisseur d'identité.

Modèle de page pour les erreurs liées à OpenID

Tivoli Federated Identity Manager utilise un modèle de page d'erreur générique pour afficher des informations de texte détaillées liées à l'erreur dans les cas suivants :

- Une erreur interrompt le traitement sur le fournisseur d'identité ou le consommateur.
- L'erreur n'est pas renvoyée.

Par exemple :

- Sur un fournisseur d'identité, cette page est utilisée lorsque le traitement des pages des sites sécurisés ou d'une requête de connexion unique ce comporte aucune adresse URL `return_to` valide.

- Au niveau du consommateur, cette page est appelée lorsque des paramètres erronés sont renvoyés dans la page de connexion.

Le modèle de page porte le nom error.html.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

Les macros de remplacement suivantes sont prises en charge :

@REQ_ADDR@
Cette macro est remplacée par l'adresse URL du noeud final délégué en cours d'appel.

@TIMESTAMP@
Cette macro est remplacée par l'heure actuelle au format de temps universel coordonné.

@DETAIL@
Cette macro est remplacée par la version en support de langue nationale (NLS) du message d'erreur associé à l'erreur.

@EXCEPTION_STACK@
Cette macro est remplacée par la trace de pile des exceptions éventuelles qui ont provoqué l'erreur.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
        <title>Une erreur OpenID s'est produite</title>
    </head>
    <body style="background-color:#ffffff">
        <div>
            <h2 style="color:#ff8800">Une erreur s'est produite</h2>
            <div id="infoDiv" style="background-color:#ffffff;color:#000000">
                <em>@REQ_ADDR@</em> <br />
                <em>@TIMESTAMP@</em> <br />
            </div>
            <br />
<div id="detailDiv" style="background-color:#999999; border-style:solid;
border-width:1px; border-color:#000000">
                <h4>Détails sur l'erreur</h4>
                @DETAIL@
            </div>
            <br />
            <div id="stackDiv" style="background-color:#999999;
border-style:solid; border-width:1px; border-color:#000000">
                <h4>Trace de pile</h4>
                @EXCEPTION_STACK@
            </div>
        </div>
    </body>
</html>
```

Figure 35. Modèle de fichier HTML error.html

Ce modèle est utilisé à la fois sur le fournisseur d'identité et le consommateur.

Modèle de page pour l'envoi indirect de requêtes OpenID 2.0

La norme OpenID 2.0 spécifie que des requêtes POST HTTP peuvent être utilisées à la place des redirections HTTP lors de l'envoi de messages de réacheminement entre le fournisseur d'identité et la partie de confiance (consommateur). Ces messages sont envoyés au navigateur, puis redirigés vers la cible.

Tivoli Federated Identity Manager effectue automatiquement la permutation des messages dans une requête FORM auto-émise via une requête HTTP POST (au lieu d'une redirection 302) lorsque les conditions suivantes sont vérifiées :

- OpenID 2.0 est utilisé
- La taille de message dépasse 2 Ko

Lorsqu'une requête POST est utilisée, une page est chargée. Celle-ci contient une requête FORM auto-émise (au lieu d'une redirection 302), contenant les mêmes paramètres que ceux qui auraient été autrement transmis via la chaîne de requête.

Le fichier du modèle est indirect_post.html.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

Le fichier prend en charge les macros de remplacement suivantes :

@OPENID_PARTNER_URL@
Cette macro est remplacée par l'adresse URL du partenaire cible. Cette valeur est utilisée pour le paramètre d'action FORM.

@PARAM_NAME@ / @PARAM_VALUE@
Il s'agit de macros à valeurs multiples spécifiées à l'intérieur d'une liste de remplacement `[RPT formFields]` répétable. Ces valeurs servent à transmettre les paramètres au destinataire concerné.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Message OpenID</title>
  </head>
  <body>
    <form method="post" name="openid_message" action="@OPENID_PARTNER_URL@">
      [RPT formFields]
        <input type="hidden" name="@PARAM_NAME@" value="@PARAM_VALUE@" />
      [ERPT formFields]
      <noscript>
      <button type="submit">Envoyer le message OpenID</button>
<!-- inclus pour les demandeurs qui ne prennent pas en charge Javascript -->
      </noscript>
    </form>
    <script type="text/javascript">
        var signOnText = 'Envoi du message OpenID en cours...';
        document.write(signOnText);
        setTimeout('document.forms[0].submit()', 0);
    </script>
  </body>
</html>
```

Figure 36. Modèle de fichier indirect_post.html

Ce modèle n'est défini que sur le fournisseur d'identité.

Modèle de page renvoyé pour checkid_immediate

Lorsqu'une requête checkid_immediate est initiée par le fournisseur de services Tivoli Federated Identity Manager et que le fournisseur d'identité renvoie un statut selon lequel la détention de l'URL par l'utilisateur ne peut pas être confirmée, le fournisseur d'identité renvoie également l'un des attributs suivants :

- openid.user_setup_url

  Pour OpenID 1.1
- openid.mode=user_setup_needed

  Pour OpenID 2.0

Lorsque le consommateur Tivoli Federated Identity Manager reçoit ce type de réponse, il renvoie un fichier de modèle de page.

Le fichier du modèle de page porte le nom immediate.html.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

Le fichier comprend la macro de remplacement suivante :

@OPENID_USER_SETUP_URL@
: Cette macro est remplacée par l'adresse URL renvoyée dans le paramètre openid.user_setup_url d'une réponse à la requête checkid_immediate par le fournisseur d'identité. Lorsqu'il s'agit d'une requête OpenID 2.0, ce paramètre peut être une chaîne vide.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <title>Résultats de checkid_immediate</title>
  </head>
  <body>
    <script type="text/javascript">
        var setup_url = "@OPENID_USER_SETUP_URL@";
        if (setup_url) {
            document.write('<a href="');
            document.write(setup_url);
            document.write('">Veuillez cliquer ici pour compléter les exigences
            du fournisseur d'identité </a>');
        } else {
            document.write('Impossible de poursuivre le traitement car une
        authentification est requise par le fournisseur d'identité OpenID.');
        }
    </script>
  </body>
</html>
```

Figure 37. Modèle de page immediate.html

Ce modèle n'est utilisé que par le consommateur.

Modèle de page renvoyé pour les erreurs du serveur

Le serveur du fournisseur d'identité renvoie `openid.mode` défini sur `error` et indique le texte de l'erreur dans le fichier `openid.error` dans les cas suivants :

- Lorsque le consommateur Tivoli Federated Identity Manager envoie une requête checkid_immediate ou checkid_setup
- Lorsque la requête checkid_immediate ou checkid_setup génère une erreur

Dans ce cas, le consommateur renvoie la page server_error.html.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

Le modèle de page prend en charge les macros de remplacement suivantes :

@OPENID_SERVER@
: Cette macro est remplacée par l'URL du serveur OpenID que le consommateur était en train de communiquer au moment où l'erreur s'est produite.

@OPENID_ERROR@
: Texte issu du fichier `openid.error`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <title>Erreur OpenID renvoyée par le serveur</title>
  </head>
  <body>
    Le serveur OpenID : @OPENID_SERVER@ a renvoyé
    le texte d'erreur suivant :<br />
    @OPENID_ERROR@
  </body>
</html>
```

Figure 38. Fichier modèle server_error.html

Ce modèle n'est utilisé que sur le consommateur.

Chapitre 27. Présentation de la planification OAuth

Tivoli Federated Identity Manager prend en charge les protocoles OAuth 1.0 et OAuth 2.0. L'implémentation d'OAuth dans Tivoli Federated Identity Manager respecte scrupuleusement les normes OAuth.

Remarque : Chaque mention du protocole OAuth 1.0 dans le guide fait référence à la version RFC5849.

Il convient que vous soyez familiarisé avec la spécification OAuth avant de mettre en oeuvre une fédération de connexion unique. Vous devez respecter les exigences suivantes pour votre implémentation OAuth 2.0 :

- Les informations que vous devez fournir à vos partenaires commerciaux
- Les informations que votre partenaire doit vous fournir

Concepts OAuth

Cette rubrique présente les concepts principaux d'OAuth 1.0 et OAuth 2.0.

OAuth est un protocole d'autorisation HTTP. Il offre aux applications tierces un accès sectorisé à une ressource protégée pour le compte du propriétaire de la ressource. Il permet un accès sectorisé en créant une interaction d'approbation entre le propriétaire de la ressource, le client et le serveur de ressources. Il offre aux utilisateurs la possibilité de partager leurs ressources privées entre différents sites sans avoir à fournir des noms d'utilisateur et des mots de passe. Les ressources privées peuvent être n'importe quelle ressource, mais les exemples courants incluent les photos, les vidéos, les listes de contacts, etc.

Pour obtenir une description complète des spécifications OAuth, consultez le site Web d'OAuth : http://www.oauth.net.

Les concepts suivants sont communs à OAuth 1.0 et OAuth 2.0.

Propriétaire de la ressource
Entité capable d'autoriser l'accès à une ressource protégée. Lorsque le propriétaire de la ressource est une personne, il est appelé un *utilisateur final*.

client OAuth
Application tiers qui veut accéder aux ressources privées du propriétaire de la ressource. Le client OAuth peut effectuer des demandes de ressources protégées pour le compte du propriétaire de la ressource une fois que ce dernier lui accorde l'autorisation. OAuth 2.0 présente deux types de client : confidentiel et public. Les clients confidentiels sont enregistrés avec un secret client, alors que les clients publics ne le sont pas.

Serveur OAuth
Appelé **Serveur d'autorisation** dans OAuth 2.0. Il offre aux clients OAuth un accès sectorisé à une ressource protégée pour le compte du propriétaire de la ressource. Le serveur envoie un jeton d'accès au client OAuth une fois que les actions suivantes sont exécutées :

- Authentifie le propriétaire de la ressource.
- Valide une demande ou un accord d'autorisation.

- Obtient l'autorisation du propriétaire de la ressource.

Un serveur d'autorisation peut également être le serveur de ressources. Tivoli Federated Identity Manager tient le rôle de ces deux serveurs.

Jeton d'accès
Chaîne représentant l'autorisation accordée au client OAuth par le propriétaire de la ressource. Cette chaîne représente les portées et les durées d'accès spécifiques. Ce jeton est accordé par le propriétaire de la ressource et est appliqué par le serveur OAuth.

Ressource protégée
Ressource restreinte pouvant être accessible à partir du serveur OAuth à l'aide de demandes authentifiées.

Des concepts supplémentaires sont introduits pour le protocole OAuth 2.0. Ces nouveaux concepts sont les suivants :

Serveur de ressources
Serveur qui héberge les ressources protégées. Il peut utiliser des jetons d'accès pour accepter et répondre aux demandes de ressources protégées. Le serveur de ressources peut être le même serveur que le serveur d'autorisation.

Accord d'autorisation
Accord qui représente l'autorisation du propriétaire de la ressource pour accéder à ses ressources protégées. Les clients OAuth utilisent un accord d'autorisation pour obtenir un jeton d'accès. Il existe quatre types d'accord d'autorisation : code d'autorisation, implicite, données d'identification par mot de passe du propriétaire de la ressource, et les données d'identification du client.

Code d'autorisation
Code généré par le serveur d'autorisation lorsque le propriétaire de la ressource autorise une requête.

Jeton de régénération
Chaîne utilisée pour obtenir un nouveau jeton d'accès.

Un jeton de régénération est éventuellement émis par le serveur d'autorisation pour le client OAuth conjointement avec un jeton d'accès. Le client OAuth peut utiliser le jeton de régénération pour demander un autre jeton d'accès à l'aide de la même autorisation, sans impliquer le propriétaire de la ressource.

Noeuds finaux OAuth

Les noeuds finals permettent aux clients OAuth de communiquer avec le serveur OAuth ou le serveur d'autorisation au sein d'une fédération.

Tous les noeuds finals sont accessibles via des adresses URL. La syntaxe des URL dépend spécifiquement du motif de l'accès.

Si vous êtes responsable de l'installation, de la configuration ou de la gestion d'une fédération dans Tivoli Federated Identity Manager, il peut être utile de vous familiariser avec ces noeuds finals et adresses URL.

Fédérations OAuth 1.0

La désignation de la fédération OAuth 1.0 est conforme à la convention de dénomination Tivoli Federated Identity Manager standard pour un identificateur unique ou `protocolID`. La syntaxe est la suivante :

`https://<hostname:port>/FIM/sps/<nom_fédération>/oauth10`

Par exemple :

`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth10`

Le tableau suivant décrit les noeuds finals qui sont utilisés dans une fédération OAuth 1.0.

Tableau 111. Définitions de noeud final OAuth 1.0 et adresses URL

Nom de noeud final	Description	Exemple
Noeud final du gestionnaire de clients	URL permettant aux propriétaires de ressources de gérer leurs clients de confiance. Le propriétaire de la ressource peut utiliser le noeud final du gestionnaire des clients pour accéder à et modifier la liste des clients qui ont été autorisés à accéder à la ressource protégée. Le gestionnaire de clients de confiance affiche le nom du client et la portée autorisée d'un client autorisé. **Remarque :** Cette liste n'affiche pas les clients qui ont été désactivés ou supprimés de la fédération. Le propriétaire de la ressource peut éventuellement supprimer les informations de client de confiance de la liste. Ainsi, le propriétaire de la ressource est invité à accorder son autorisation lors de la prochaine tentative du client OAuth pour accéder à la ressource protégée.	`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth10/clients`
Noeud final de demande de données d'identification temporaires	URL de requête utilisée par le client OAuth pour obtenir un ensemble de données d'identification temporaires.	`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth10/request`
Noeud final d'autorisation du propriétaire de la ressource	URL d'autorisation où le propriétaire de la ressource accorde une autorisation au client OAuth pour accéder à la ressource protégée.	`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth10/authorize`
Noeud final de demande de jeton	Adresse URL d'accès où le client OAuth échange l'ensemble de données d'identification temporaires et le code de vérification contre un ensemble de données d'identification de jeton.	`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth10/access`

Fédérations OAuth 2.0

La désignation de la fédération OAuth 2.0 est conforme à la convention de dénomination Tivoli Federated Identity Manager standard pour un identificateur unique ou `protocolID`. La syntaxe est la suivante :

`https://<hostname:port>/FIM/sps/<nom_fédération>/oauth20`

Par exemple :

`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth20`

Le tableau suivant décrit les noeuds finals qui sont utilisés dans une fédération OAuth 2.0.

Remarque : Tous les types d'accord d'autorisation n'utilisent pas les trois points finaux dans un même flux OAuth 2.0.

Tableau 112. Définitions de noeud final OAuth 2.0 et adresses URL

Nom de noeud final	Description	Exemple
Noeud final du gestionnaire de clients	URL permettant aux propriétaires de ressources de gérer leurs clients de confiance. Le propriétaire de la ressource peut utiliser le noeud final du gestionnaire des clients pour accéder à et modifier la liste des clients qui ont été autorisés à accéder à la ressource protégée. Le gestionnaire de clients de confiance affiche le nom du client et la portée admise d'un client autorisé. **Remarque :** Cette liste n'affiche pas les clients qui ont été désactivés ou supprimés de la fédération. Le propriétaire de la ressource peut éventuellement supprimer les informations de client de confiance de la liste. Ainsi, le propriétaire de la ressource est invité à accorder son autorisation lors de la prochaine tentative du client OAuth pour accéder à la ressource protégée.	`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth20/clients`
Noeud final d'autorisation	URL d'autorisation où le propriétaire de la ressource accorde une autorisation au client OAuth pour accéder à la ressource protégée.	`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth20/authorize`
Noeud final de jeton	URL de demande de jeton où le client OAuth échange un accord d'autorisation contre un jeton d'accès et un jeton de régénération facultatif.	`https://server.oauth.com/FIM/sps/MySocialNetwork/oauth20/token`

Flux de travaux OAuth 1.0

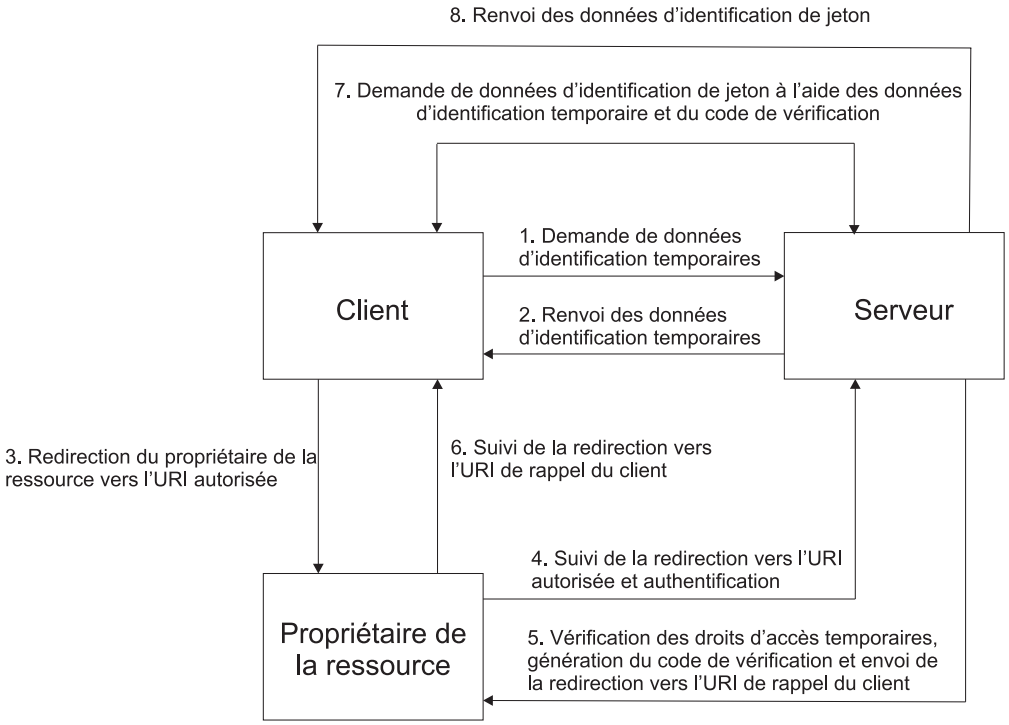
La version RFC5849 d'OAuth 1.0, ou Open Authorization, est un protocole d'autorisation HTTP. La prise en charge d'OAuth 1.0 permet aux utilisateurs de partager leurs ressources privées entre des sites sans avoir à fournir des utilisateurs et des mots de passe. Les ressources privées peuvent être n'importe quelle ressource, mais les exemples courants incluent les photos, les vidéos et les listes de contacts.

La fonction OAuth 1.0 de Tivoli Federated Identity Manager peut être configurée via les méthodes suivantes :

- Console Tivoli Federated Identity Manager
- Interface de ligne de commande

Flux de travaux OAuth 1.0

Un serveur OAuth émet des jetons pour les clients OAuth. Les clients OAuth peuvent accéder aux ressources pour le compte du propriétaire de la ressource à l'aide de jetons dotés de portée, de durées de vie et autres attributs.



Le diagramme de flux de travaux d'exécution du protocole OAuth 1.0 comprend les étapes suivantes :

1. Le client OAuth demande un ensemble de données d'identification temporaires au serveur OAuth afin de démarrer le processus d'authentification. Les données d'identification temporaires permettent de distinguer les requêtes individuelles du client OAuth envoyées au serveur OAuth.
2. Le serveur OAuth valide la requête et renvoie un ensemble de données d'identification temporaires au client OAuth.
3. le client OAuth redirige le propriétaire de la ressource vers l'URI autorisé pour pouvoir accéder à la ressource protégée.
4. Le propriétaire de la ressource s'authentifie auprès du serveur OAuth à l'aide de ses données d'identification client et autorise la requête émise par le client OAuth.
5. Le serveur OAuth valide les données d'identification temporaires et une fois que le propriétaire de la ressource autorise le client OAuth, un code de vérification est généré.
6. Le propriétaire de la ressource est redirigé vers l'URI de rappel fourni par le client OAuth dans la requête précédente.
7. Le client OAuth demande le jeton d'accès à l'aide des données d'identification temporaires et du code de vérification.

8. Le serveur OAuth valide la requête et renvoie un jeton d'accès au client OAuth pour pouvoir accéder à la ressource protégée.

A propos de OAuth deux jambes

Utilisez OAuth deux jambes pour mettre en oeuvre une délégation des droits dans le client.

OAuth deux jambes est également appelé *Signed Fetch*. Dans le scénario OAuth deux jambes, le client OAuth utilise le secret client pour signer la requête et accéder directement à la ressource protégée. Le serveur OAuth compte sur le client OAuth pour fournir des données sans en demander l'autorisation au propriétaire de la ressource.

Interface du service de jeton de sécurité pour flux OAuth à deux jambes

L'interface du service de jeton de sécurité traite les demandes différemment pour un flux OAuth à deux jambes.

Cette section fournit les informations suivantes sur le scénario OAuth à deux jambes :

- Le comportement du service de jeton de sécurité Tivoli Federated Identity Manager.
- La signification du point d'application de l'intercepteur de relations de confiance WebSphere.

Lorsqu'un client OAuth accède à une ressource protégée qui utilise un flux OAuth à deux jambes, aucun jeton token ni propriétaire de ressource d'autorisation n'est associé à la demande. Le client OAuth signe la demande avec ses données d'identification du client, ce qui prouve que la demande a été de ce client. Cette méthode est similaire à une authentification de base traditionnelle, à ceci près que la demande est signé numériquement plutôt que contenant les données d'identification client en texte clair.

Le flux OAuth à deux jambes suit ce processus :

1. Un point d'application OAuth reçoit une demande OAuth à deux jambes.
2. Le point d'application contacte le service de jeton de sécurité pour la validation.
3. Le service de jeton de sécurité traite ensuite la demande OAuth à deux jambes demande envoyée par le point de mise en application, puis valide la signature de demande.
4. Le service de jeton de sécurité renvoie un attribut de réponse`nom_utilisateur` en tant que partie de la réponse de validation. Cet attribut est défini sur la valeur de l'ID client, qui est également appelée clé du client.

   Remarque : Cette méthode est différenre d'un flux OAuth à deux jmabes classique, où l'attribut `nom_utilisateur` est défini sur le nom d'utilisateur du propriétaire de la ressource. L'utilisateur est la même entité que celle qui a autorisé l'accès du client OAuth à la ressource protégée.

Le point d'application OAuth peut utiliser l'attribut de réponse`nom_utilisateur` renvoyé pour l'audit, l'authentification ou l'application de ressource protégée en aval.

Tivoli Federated Identity Manager fournit des exemples de règles de mappage OAuth qui détectent un scénario OAuth à deux jambes. Lorsqu'un tel scénario est détecté, la règle de mappage remplace la valeur de l'attribut `nom_utilisateur` de l'identificateur du client par la valeur prédéfinie `me_guest`.

Vous pouvez modifier la règle de mappage soit pour quitter l'attribut `nom_utilisateur` en tant qu'identificateur du client soir pour le mapper à une autre entité. Ce changement devient important uniquement lorsque votre application ou la ressource protégée en aval du point d'application reposent sur la valeur de l'attribut `nom_utilisateur` renvoyé pour un traitement spécial.

Le point d'application de l'intercepteur de relations de confiance requiert la valeur de l'attribut `nom_utilisateur` renvoyé. Le point d'application effectue une authentification WebSphere en tant qu'attribut `nom_utilisateur` renvoyé par le service de jeton de sécurité. Le registre d'utilisateur WebSphere Application Server doit contenir un utilisateur qui correspond à l'attribut `nom_utilisateur` renvoyé par une règle de mappage personnalisée dans le service de jeton de sécurité.

Si vous utiisez l'exemple de règle mappage en l'état, vous devez créer un utilisateur dans le registre d'utilisateurs WebSphere nommé `me_guest`. Cette étape est nécessaire uniquement pour le flux OAuth à deux jambes avec le point d'application de l'intercepteur de relation de confiance.

Flux de travaux OAuth 2.0

La prise en charge d'OAuth 2.0 dans Tivoli Federated Identity Manager propose quatre façons différentes qui permettent à un client OAuth d'accéder à la ressource protégée.

La fonction OAuth 2.0 de Tivoli Federated Identity Manager peut être configurée via les méthodes suivantes :

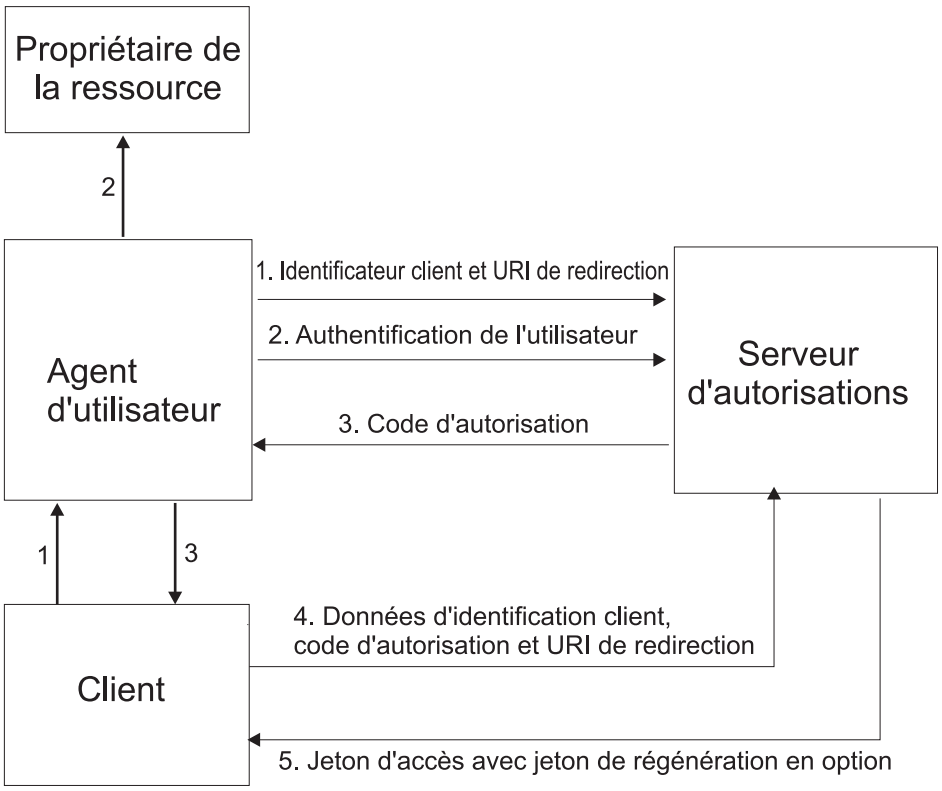
- Console Tivoli Federated Identity Manager
- Interface de ligne de commande

Flux de travaux OAuth 2.0

Tivoli Federated Identity Manager prend en charge les flux de travaux OAuth 2.0 suivants.

Flux de code d'autorisation

Le type d'accord du code d'autorisation est approprié pour les clients OAuth qui peuvent conserver la confidentialité de leurs données d'identification client lors de l'authentification avec le serveur d'autorisation. Par exemple, un client mis en oeuvre sur un serveur sécurisé. En tant que flux basé sur le réacheminement, le client OAuth doit être en mesure d'interagir avec l'agent d'utilisateur du propriétaire de la ressource. Il doit également être en mesure de recevoir des requêtes entrantes via le réacheminement à partir du serveur d'autorisation.

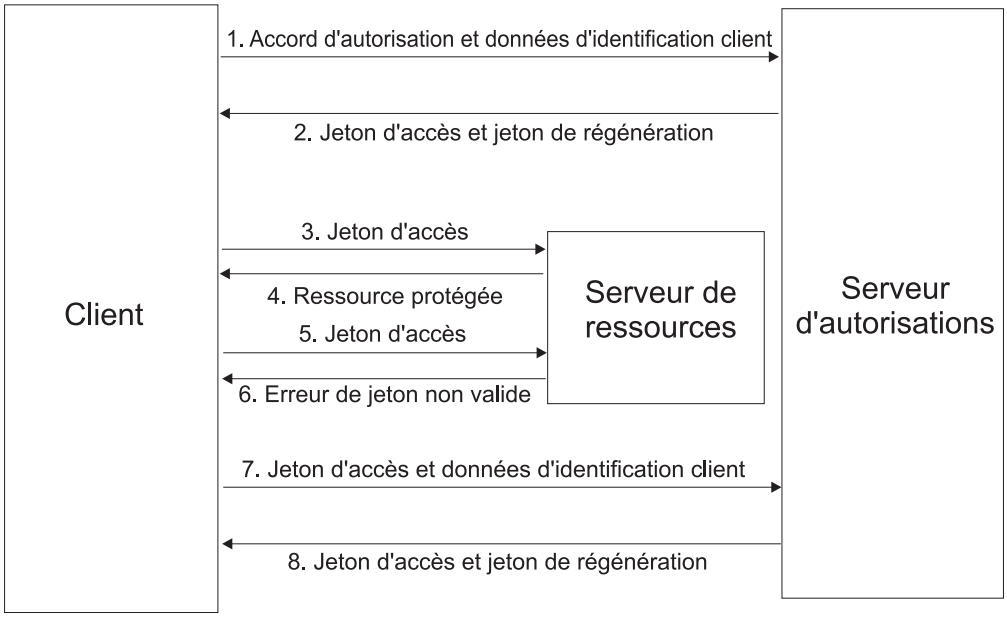


Le diagramme de flux de travaux du code d'autorisation comprend les étapes suivantes :

1. Le client OAuth lance le flux lorsqu'il dirige l'agent d'utilisateur du propriétaire de la ressource vers le noeud final d'autorisation. Le client OAuth inclut son ID client, la portée demandée, l'état local, et un URI de réacheminement. Le serveur d'autorisation renvoie l'agent d'utilisateur vers l'URI de réacheminement une fois que l'accès est accordé ou refusé.
2. Le serveur d'autorisation authentifie le propriétaire de la ressource via l'agent d'utilisateur et détermine si le propriétaire de la ressource accorde ou refuse la demande d'accès.
3. Si le propriétaire de la ressource autorise l'accès, le client OAuth utilise l'URI de réacheminement fourni précédemment pour rediriger l'agent d'utilisateur vers le client OAuth. L'URI de réacheminement comprend un code d'autorisation et un état local précédemment fournis par le client OAuth.
4. Le client OAuth demande un jeton d'accès à partir du serveur d'autorisation via le noeud final du jeton. Le client OAuth s'authentifie à l'aide de ses données d'identification client et inclut le code d'autorisation reçu à l'étape précédente. Le client OAuth inclut également l'URI de réacheminement utilisé pour obtenir le code d'autorisation à des fins de vérification.
5. Le serveur d'autorisation valide les données d'identification du client et le code d'autorisation. Le serveur s'assure également que l'URI de réacheminement reçu correspond à l'URI utilisé pour rediriger le client à l'étape 3. S'il est valide, le serveur d'autorisation répond à l'aide d'un jeton d'accès.

Le serveur d'autorisation peut être le même serveur que le serveur de ressources ou une entité distincte. Un seul serveur d'autorisation peut émettre des jetons d'accès acceptés par plusieurs serveurs de ressources.

Flux de code d'autorisation avec le jeton de régénération



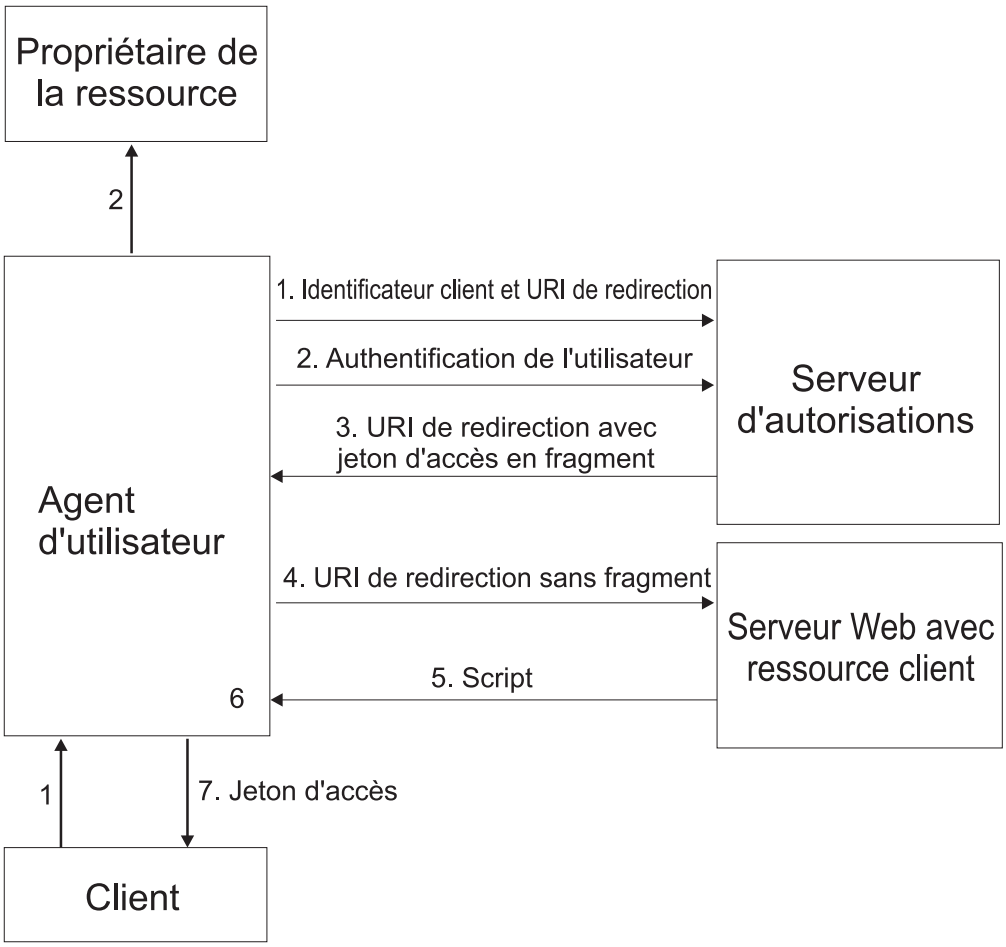
Le diagramme de flux de travaux du code d'autorisation avec un jeton de régénération comprend les étapes suivantes :

1. Le client OAuth demande un jeton d'accès via une authentification auprès du serveur d'autorisation avec ses données d'identification client, et en présentant un accord d'autorisation.
2. Le serveur d'autorisation valide les données d'identification du client et l'accord d'autorisation. S'ils sont valides, le serveur d'autorisation émet un jeton d'accès et un jeton de régénération.
3. Le client OAuth effectue une demande de ressource protégée auprès du serveur de ressource en présentant le jeton d'accès.
4. Le serveur de ressources valide le jeton d'accès. Si le jeton d'accès est valide, le propriétaire de la ressource accepte la demande.
5. Répétez les étapes 3 et 4 jusqu'à ce que le jeton d'accès arrive à expiration. Si le client OAuth sait que le jeton d'accès a expiré, passez à l'étape 7. Sinon, le client OAuth émet une autre demande de ressource protégée.
6. Si le jeton d'accès n'est pas valide, le serveur de ressources renvoie une erreur.
7. Le client OAuth demande un nouveau jeton d'accès via une authentification auprès du serveur d'autorisation avec ses données d'identification client, et en présentant le jeton de régénération.
8. Le serveur d'autorisation valide les données d'identification du client et le jeton de régénération, et en cas de validité, émet un nouveau jeton d'accès et un nouveau jeton de régénération.

Flux d'accord implicite

Le type d'accord implicite est approprié pour les clients qui ne peuvent pas maintenir la confidentialité de leurs données d'identification client pour une authentification auprès du serveur d'autorisation. Un exemple peut se présenter sous la forme d'applications client qui se trouvent dans un agent d'utilisateur, généralement implémentée dans un navigateur à l'aide d'un langage de script tel que JavaScript.

En tant que flux basé sur le réacheminement, le client OAuth doit être en mesure d'interagir avec l'agent d'utilisateur du propriétaire de la ressource, généralement un navigateur Web. Le client OAuth doit également être en mesure de recevoir des requêtes entrantes via le réacheminement à partir du serveur d'autorisation.



Le diagramme de flux de travaux de l'accord implicite comprend les étapes suivantes :

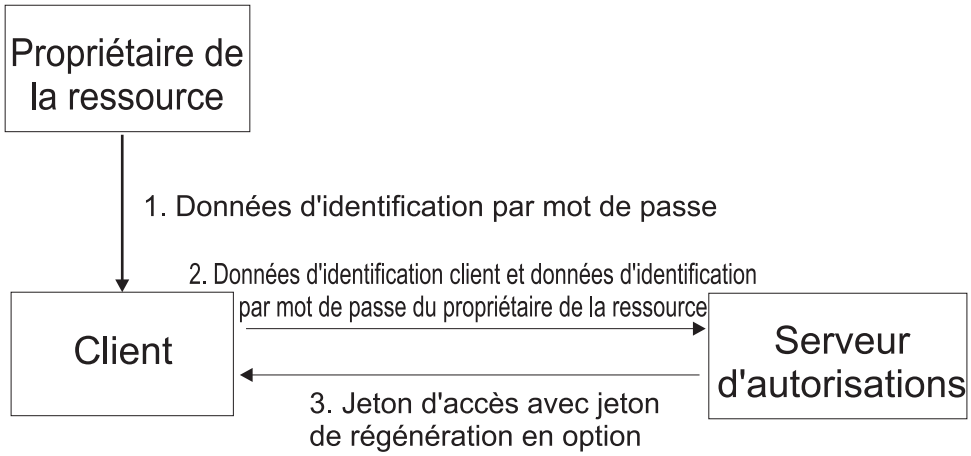
1. Le client OAuth lance le flux en dirigeant l'agent d'utilisateur du propriétaire de la ressource vers le noeud final d'autorisation. Le client OAuth inclut son ID client, la portée demandée, l'état local, et un URI de réacheminement. Le serveur d'autorisation renvoie l'agent d'utilisateur vers l'URI de réacheminement une fois que l'accès est accordé ou refusé.
2. Le serveur d'autorisation authentifie le propriétaire de la ressource via l'agent d'utilisateur et détermine si le propriétaire de la ressource accorde ou refuse la demande d'accès.
3. Si le propriétaire de la ressource autorise l'accès, le serveur d'autorisation redirige l'agent d'utilisateur vers le client à l'aide de l'URI de réacheminement fourni précédemment. L'URI de réacheminement inclut le jeton d'accès dans le fragment d'URI.
4. L'agent d'utilisateur suit les instructions de réacheminement en effectuant une demande au serveur Web sans le fragment. L'agent d'utilisateur conserve les informations de fragment en local.

5. Le serveur Web renvoie une page Web, qui est généralement un document HTML avec un script intégré. La page Web accède à l'intégralité de l'URI de réacheminement, y compris le fragment conservé par l'agent d'utilisateur. Elle peut également extraire le jeton d'accès et d'autres paramètres contenus dans le fragment.
6. L'agent d'utilisateur exécute le script fourni par le serveur Web localement, ce qui entraîne l'extraction du jeton d'accès et sa transmission au client.

Flux de données d'identification par mot de passe du propriétaire de la ressource

Le type d'accord des données d'identification par mot de passe du propriétaire de la ressource est adaptée dans les cas où le propriétaire de la ressource a une relation d'accréditation avec le client. Par exemple, le propriétaire de la ressource peut être le système d'exploitation d'un ordinateur du client OAuth ou une application hautement privilégiée.

Vous pouvez uniquement utiliser ce type d'accord lorsque le client OAuth a obtenu les données d'identification du propriétaire de la ressource. Il est également utilisé pour migrer les clients existants à l'aide des schémas d'authentification directe en convertissant les données d'identification stockées en un jeton d'accès.



Le diagramme de flux de travaux des données d'identification par mot de passe du propriétaire de la ressource comprend les étapes suivantes :

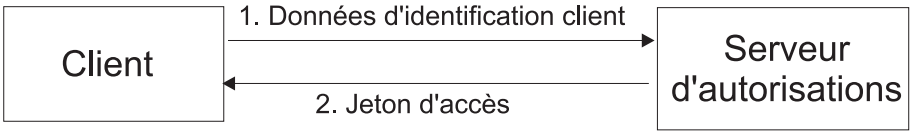
1. Le propriétaire de la ressource fournit au client son nom d'utilisateur et son mot de passe.
2. Le client OAuth demande un jeton d'accès à partir du serveur d'autorisation via le noeud final du jeton. Le client OAuth s'authentifie à l'aide de ses données d'identification client et inclut les données d'identification reçues du propriétaire de la ressource.
3. Une fois que le serveur d'autorisation valide les données d'identification du propriétaire de la ressource et les données d'identification client, il émet un jeton d'accès et, éventuellement, un jeton de régénération.

Flux de données d'identification client

Le flux de données d'identification client est utilisé lorsque le client OAuth demande un jeton d'accès uniquement à l'aide de ses données d'identification client. Ce flux est applicable dans l'une des situations suivantes :

- Le client OAuth souhaite accéder aux ressources protégées sous son contrôle.

- Le client OAuth souhaite accéder à une autre ressource protégée, où l'autorisation a été précédemment convenue avec le serveur d'autorisation.



Le diagramme de flux de travaux des données d'identification client comprend les étapes suivantes :

1. Le client OAuth demande un jeton d'accès au noeud final du jeton via une authentification avec ses données d'identification client.
2. Une fois que le serveur d'autorisation valide les données d'identification du client, il émet un jeton d'accès.

Remarques sur l'authentification du client au niveau du noeud final du jeton OAuth 2.0

Le noeud final du jeton OAuth 2.0 est utilisé pour les communications directes entre un client OAuth et le serveur d'autorisation.

Le noeud final du jeton est utilisé pour obtenir un jeton OAuth. Le type de client, public ou confidentiel, détermine les exigences en matière d'authentification du noeud final de jeton OAuth 2.0.

Les flux de travaux OAuth 2.0 pour les clients confidentiels nécessitant l'authentification client sur le noeud final du jeton, peuvent être configurés de l'une des manières suivantes :

1. Le point de contact Tivoli Federated Identity Manager requiert l'authentification au niveau du noeud final du jeton :
   - Le point de contact est responsable de l'authentification du client.
   - La case **Allow public clients to access the token endpoint** du panneau de propriétés de fédération n'est pas pertinente. Un paramètre `client_secret` ne doit *pas* être envoyé dans la demande de noeud final du jeton.
   - Si un paramètre `client_id` est envoyé dans la demande, il doit correspondre à l'identité du client qui est authentifié par le point de contact.
2. Le point de contact Tivoli Federated Identity Manager permet un accès non authentifié au noeud final du jeton :
   - Le paramètre `client_id` dans la demande de noeud final du jeton est utilisé pour identifier le client.
   - Le partenaire de fédération, également appelé client, doit être activé pour qu'il puisse être identifié.
   - La case **Allow public clients to access the token endpoint** du panneau de propriétés de fédération détermine si un paramètre `client_secret` est requis dans la demande de noeud final du jeton. Un secret client est requis uniquement pour les client confidentiels.

Remarque : Lors de l'exécution de l'authentification client sur le noeud final du jeton, le point de contact doit contenir l'ID client et le secret client dans son registre d'utilisateurs. Le point de contact doit être en mesure de mapper les données d'identification de l'utilisateur authentifié vers le paramètre `id_client` envoyé dans la demande de noeud final du jeton OAuth 2.0.

Sur la base de ces informations, les configurations suivantes sont prises en charge :

Tableau 113. Configurations prises en charge

Types de client	Configurations	Remarques relatives à l'URI de noeud final du jeton point de contact WebSEAL	Remarques relatives à l'URI de noeud final du jeton point de contact WebSphere Application Server	Définition du paramètre «Allow public clients to access the token endpoint»
Clients confidentiels	Le point de contact effectue l'authentification du client.	• Une ACL authentifiée est requise sur le noeud final du jeton. • Le port du noeud final du jeton doit correspondre au port WebSEAL.	• Le port du noeud final du jeton doit correspondre au port SOAP Tivoli Federated Identity Manager. • Le rôle **FIMSoapClient** doit être défini pour le client OAuth.	N/A
Clients confidentiels	Les paramètres **client_id** et **client_secret** dans la demande de noeud final du jeton sont utilisés pour effectuer l'authentification du client.	• Une ACL non authentifiée est requise sur le noeud final du jeton. • Le port du noeud final du jeton doit correspondre au port WebSEAL.	Le noeud final du jeton doit utiliser le même nom d'hôte et port de point de contact que les noeuds finals autorisés et de gestionnaire du client.	Doit avoir la valeur *false*.
Clients publics	Le paramètre **client_id** est utilisé pour effectuer une validation client.	• Une ACL non authentifiée est requise sur le noeud final du jeton. • Le port du noeud final du jeton doit correspondre au port WebSEAL.	Le noeud final du jeton doit utiliser le même nom d'hôte et port de point de contact que les noeuds finals autorisés et de gestionnaire du client.	Doit avoir la valeur *true*.

Utilisation de WebSphere Application Server en tant que point de contact au niveau du noeud final du jeton

Lors de l'exécution de l'authentification au niveau du noeud final de jeton pour un point de contact WebSphere Application Server, l'URL du noeud final de jeton doit utiliser le port SOAP de Tivoli Federated Identity Manager . Cette condition garantit que l'autorisation est appliquée par le rôle **FIMSoapClient**. Le noeud final SOAP Tivoli Federated Identity Manager peut ensuite être configuré pour les mécanismes d'authentification de client appropriés, tels que l'authentification de base ou le certificat client. Pour plus d'informations, voir «Configuration des paramètres d'authentification du noeud final SOAP», à la page 422.

Remarque : Vous devez définir manuellement les propriétés personnalisées de l'environnement d'exécution **`TFIM.SOAP.Port`** et **`SOAP.AuthType`** lorsque vous utilisez WebSphere Application Server de la manière suivante :

- En tant que serveur point de contact dans un cluster
- Pour appliquer l'authentification du noeud final du jeton OAuth 2.0

La case **Allow public clients to access the token endpoint** du panneau de propriétés de fédération n'a aucune incidence sur le traitement de la demande lorsque le point de contact applique l'authentification.

L'URL de noeud final du jeton doit utiliser le même nom d'hôte et port de point de contact que celui autorisé et les noeuds finals des gestionnaires de clients lorsque les conditions suivantes sont applicables :

- WebSphere Application Server est utilisé en tant que point de contact.
- L'accès non authentifié au noeud final du jeton est accepté.

Dans ce cas, le rôle **FIMUnauthenticated** est utilisé. Une autorisation supplémentaire dépend selon si le client est activé. La case **Allow public clients to access the token endpoint** du panneau de propriétés de fédération détermine si le paramètre `client_secret` est requis dans la demande de noeud final du jeton. Il n'est pas obligatoire qu'un client public fournisse un paramètre `client_secret`.

Utilisation de Tivoli Access Manager WebSEAL en tant que point de contact au niveau du noeud final du jeton

Vous pouvez utiliser l'utilitaire `tfimcfg` de Tivoli Federated Identity Manager pour configurer WebSEAL en tant que point de contact d'une fédération OAuth 2.0.

Lors de l'exécution de l'authentification au niveau de WebSEAL pour le noeud final du jeton, utilisez des instances WebSEAL séparées pour le jeton et les noeuds finals d'autorisation. Cette condition permet aux clients de s'authentifier avec des mécanismes d'authentification, tels que l'authentification de base et des certificats client sur le noeud final du jeton. Dans le même temps, les utilisateurs peuvent toujours s'authentifier en utilisant l'authentification par formulaires sur les noeuds finals autorisés et de gestionnaire de clients. Dans ce cas, la configuration de noeud final du jeton dans la fédération OAuth 2.0 doit correspondre au nom d'hôte et au port du noeud final de jeton WebSEAL approprié. Pour plus d'informations sur l'utilisation de l'utilitaire `tfimcfg` pour configurer WebSEAL en tant que point de contact pour une fédération OAuth 2.0, voir «Configuration d'un serveur point de contact WebSEAL pour la fédération OAuth», à la page 444.

Configuration des paramètres d'authentification du noeud final SOAP

Vous pouvez configurer le noeud final SOAP de Tivoli Federated Identity Manager afin qu'il utilise l'authentification standard ou le certificat client comme mécanisme d'authentification de client.

Pourquoi et quand exécuter cette tâche

L'URL du noeud final de jeton utilise le port SOAP de Tivoli Federated Identity Manager lorsque l'authentification d'un point de contact WebSphere Application Server est appliquée.

Pour savoir comment un client est authentifié, sélectionnez un type d'authentification du noeud final SOAP.

Procédure

1. Connectez-vous à la console Integrated Solutions Console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Point de contact**.
3. Sélectionnez le profil du serveur point de contact que vous utilisez dans votre environnement.
4. Cliquez sur **Avancé**. Le panneau Paramètres de sécurité de noeud final SOAP s'ouvre.

5. Sélectionnez le type d'authentification de noeud final SOAP parmi les choix suivants :
   - Authentification de base

     Authentification qui requiert que le client OAuth fournisseur l'identifiant client et le secret partagé.
   - Authentification par certificat client

     Authentification qui requiert que le client OAuth présente un certificat afin que la session authentifiée sécurisée puisse être établie.
6. Cliquez sur **OK**.
7. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**

Enregistrement du client

Un client est ajouté à une fédération OAuth en tant que partenaire. Il ne s'agit ni d'un *Fournisseur de services* ni d'un *Fournisseur d'identité*.

La création d'un partenaire dans une fédération OAuth est identique à l'enregistrement d'un client sur un serveur OAuth ou un serveur d'autorisation. Un serveur OAuth ou un serveur d'autorisation peut comporter plusieurs clients. Par conséquent, une fédération OAuth peut avoir plusieurs partenaires.

Une fédération OAuth peut communiquer avec les clients OAuth qui sont gérés dans Tivoli Federated Identity Manager, ou à partir d'un fournisseur client externe.

La fédération OAuth génère un ensemble unique de données d'identification du client, lors de la création de chaque partenaire. La clé client et le secret client sont des exemples de l'ensemble de données d'identification du client généré par la fédération OAuth. Les clients utilisent ces données d'identification pour s'identifier auprès d'un serveur OAuth ou d'un serveur d'autorisation lors d'une demande d'accès à une ressource protégée.

Gestion d'état

Le paramètre `state_id` du module STSUniversalUser est utilisé en tant que clé pour stocker ou extraire des informations d'état pour chaque appel de la chaîne d'accréditation d'un flux OAuth.

Tivoli Federated Identity Manager fournit des exemples de règles de mappage pour l'application de démonstration qui peuvent placer des informations dans un mappage distribué WebSphere Application Server, IDMappingExtCache. Ces exemples de règle de mappage utilisent l'API de gestion d'état et sont applicables aux protocoles OAuth 1.0 et OAuth 2.0. L'emplacement de ces exemples de règle de mappage est le suivant :

```
/opt/IBM/FIM/examples/demo/demo_rules/
```

Vous pouvez appeler l'API de gestion d'état depuis la règle de mappage XSLT ou JavaScript, ou à partir d'un module de mappage personnalisé. Cette fonction n'est pas disponible dans une règle de mappage Tivoli Directory Integrator. Le serveur Tivoli Directory Integrator s'exécute dans un autre processus Java, et ne peut donc pas utiliser les fonctions IDMappingExtUtil pour accéder à la mappe distribuée WebSphere Application Server.

OAuth 1.0

Les jetons OAuth 1.0 comportent un paramètre `state_id` qui est utilisé dans les règles de mappage du service de jetons de sécurité (STS). Le paramètre `state_id` gère l'état entre les appels de service de jetons de sécurité associés dans un flux OAuth 1.0.

L'attribut `state_id` est établi lorsqu'un client OAuth demande des données d'identification temporaires et reste le même dans l'intégralité d'un flux OAuth 1.0. Il permet de différencier entre les deux flux OAuth d'un seul client OAuth.

L'exemple de règle de mappage ajoute la durée de stockage des jetons à une mappe distribuée et l'extrait au cours d'une requête pour une ressource protégée.

figure 39, à la page 425 présente les sections de l'exemple de règle de mappage XSLT OAuth 1.0 qui illustre l'utilisation de l'API de gestion d'état.

```
xmlns:cache-ext="com.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtCache"
xmlns:mapping-ext="com.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils"
extension-element-prefixes="mapping-ext cache-ext" version="1.0">

...
<!-- Le paramètre token_type pour cette requête. -->
    <xsl:variable name="token_type"
        select="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='token_type']
               [@type='urn:ibm:names:ITFIM:oauth:request']/stsuuser:Value"/>
...
<!-- Descripteur du paramètre state_id pour ce jeton OAuth -->
  <xsl:variable name="state_id">
        <xsl:value-of select="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='state_id']
               [@type='urn:ibm:names:ITFIM:oauth:state']/stsuuser:Value"/>
    </xsl:variable>
...
<!-- Ce modèle dispose d'une correspondance sur ContextAttribute pour le paramètre "state_id"
        avec le type "urn:ibm:names:ITFIM:oauth:state". Si la requête en cours est en mode de demande,
        nous stockons le temps universel coordonné de la requête dans une mémoire cache avec state_id comme clé. Si
        la demande en cours est en mode de validation et que l'autorisation a abouti, nous extrayons la
        valeur d'état stockée (c'est-à-dire, le temps UTC d'origine de la requête pour un jeton temporaire) et
        la mettons dans un attribut nommé recovered_state. Il s'agit ici d'une simple illustration de l'utilisation de
        l'API de gestion d'état.
    -->
    <xsl:template
        match="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='state_id']
               [@type='urn:ibm:names:ITFIM:oauth:state']">

        <!-- Conservez d'abord cet attribut dans la sortie -->
        <stsuuser:Attribute>
            <xsl:attribute name="name">
                <xsl:value-of select="@name" />
            </xsl:attribute>
            <xsl:attribute name="type">
                <xsl:value-of select="@type"/>
            </xsl:attribute>
            <xsl:for-each select="stsuuser:Value">
                <stsuuser:Value>
                    <xsl:value-of select="."/>
                </stsuuser:Value>
            </xsl:for-each>
        </stsuuser:Attribute>

        <!-- Si le mode est "requête", stockez un élément dans le cache de l'état -->
        <xsl:if test="$token_type = 'request'">
            <xsl:variable name="utc_time"
              select="concat('State storage time was: ', mapping-ext:getCurrentTimeStringUTC())"/>

            <!-- Extrayez le cache -->
            <xsl:variable name="cache" select="mapping-ext:getIDMappingExtCache()"/>

             <!--
                  Stockez le temps universel coordonné dans le cache
                  Certains conteneurs ignore la déclaration de variable si la variable n'est pas utilisée.
                  C'est pourquoi un commentaire est inséré ici, même si aucune sortie n'est nécéssaire pour
                  appeler la méthode 'put' sur l'extension du cache.
             -->
            <xsl:comment>
                <xsl:value-of
                    select="cache-ext:put($cache, $state_id, $utc_time, 1000)" />
            </xsl:comment>

        </xsl:if>

        <!-- Si le mode est "valider" et que l'autorisation a abouti,
                                        extrayez un élément du cache de l'état. -->
        <xsl:if test="$token_type = 'validate' and $authorizationResult = 'TRUE'">
            <!-- Extrayez le cache et placez-le dans l'attribut recovered_state -->
            <xsl:variable name="cache" select="mapping-ext:getIDMappingExtCache()"/>

            <stsuuser:Attribute name="recovered_state"
                     type="urn:ibm:names:ITFIM:oauth:response:attribute">
                <stsuuser:Value>
                    <xsl:value-of select="cache-ext:get($cache, $state_id)"/>
                </stsuuser:Value>
            </stsuuser:Attribute>
        </xsl:if>
    </xsl:template>
```

Figure 39. Exemple de code XSL OAuth 1.0 avec la gestion d'état

OAuth 2.0

Les jetons OAuth 2.0 tels que les accords, les jetons d'accès, et les jetons de régénération, ont un paramètre `state_id` qui est utilisé dans les règles de mappage de service de jeton de sécurité. Le paramètre `state_id` gère l'état entre les appels de service de jetons de sécurité associés dans un flux OAuth 2.0.

Tout comme pour OAuth 1.0, la règle de mappage OAuth 2.0 utilise `state_id` comme clé pour émettre un accord d'autorisation. La clé est utilisée pour ajouter le temps de stockage du jeton à un mappage distribué. Le temps de stockage est ensuite extrait de la mémoire cache lors d'une demande pour une ressource protégée.

figure 40, à la page 427 présente les sections de l'exemple de règle de mappage XSLT OAuth 2.0 qui illustre l'utilisation de l'API de gestion d'état.

```
<!--
    Paramètre request_type de cette requête. Si aucun n'est fourni, utilisez "resource"
   -->
  <xsl:variable name="requestType">
    <xsl:choose>
      <xsl:when test="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='request_type']
         [@type='urn:ibm:names:ITFIM:oauth:request']/stsuuser:Value">
        <xsl:value-of select="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='request_type']
         [@type='urn:ibm:names:ITFIM:oauth:request']/stsuuser:Value"/>
      </xsl:when>
      <xsl:otherwise>resource</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <!-- Descripteur du paramètre state_id pour ce jeton OAuth -->
  <xsl:variable name="stateId">
    <xsl:value-of select="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='state_id']
         [@type='urn:ibm:names:ITFIM:oauth:state']/stsuuser:Value"/>
  </xsl:variable>
  <!-- Type d'accord de cette requête OAuth -->
  <xsl:variable name="grantType">
    <xsl:value-of select="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='grant_type']
         [@type='urn:ibm:names:ITFIM:oauth:body:param']/stsuuser:Value"/>
  </xsl:variable>
...

<!-- Ce modèle dispose d'une correspondance sur ContextAttribute pour le paramètre "state_id"
        avec le type "urn:ibm:names:ITFIM:oauth:state". Il effectue les opérations suivantes, si :

request_type = 'authorization' ==> Stockez le temps universel coordonné de la requête dans une mémoire cache
        avec state_id comme clé [authorization_code, implicit]
request_type = 'access_token' && grant_type = 'client_credentials' ==> Stockez le temps universel coordonné
                de la demande dans un cache avec state_id comme clé [client_credentials]
request_type = 'access_token' && grant_type = 'password' ==> Stockez le temps universel coordonné de la requête
                dans un cache avec state_id comme clé [password]
request_type = 'resource' ==> Récupérez le temps stocké et
                placez-le dans un attribut nommé recovered_state

Il s'agit ici d'une simple illustration de l'utilisation de l'API de gestion d'état.
    -->
  <xsl:template match="//stsuuser:ContextAttributes/stsuuser:Attribute[@name='state_id']
                  [@type='urn:ibm:names:ITFIM:oauth:state']">
    <!-- Conservez d'abord cet attribut dans la sortie -->
    <stsuuser:Attribute>
      <xsl:attribute name="name">
        <xsl:value-of select="@name" />
      </xsl:attribute>
      <xsl:attribute name="type">
        <xsl:value-of select="@type"/>
      </xsl:attribute>
      <xsl:for-each select="stsuuser:Value">
        <stsuuser:Value>
          <xsl:value-of select="."/>
        </stsuuser:Value>
      </xsl:for-each>
    </stsuuser:Attribute>
    <xsl:choose>

      <!-- Stockez le temps universel coordonné comme état lorsqu'il s'agit :
       d'une étape d'autorisation pour authorization_code ou flux implicite
       d'une étape de jeton, mais uniquement pour client_credentials ou les flux de données d'identification
du propriétaire de la ressource
      -->
      <xsl:when test="$requestType = 'authorization' or
        ($requestType = 'access_token' and
        ($grantType = 'client_credentials' or $grantType = 'password'))">

        <xsl:variable name="utc_time"
          select="concat('State storage time was: ', mapping-ext:getCurrentTimeStringUTC())"/>
        <!-- Extrayez le cache -->
        <xsl:variable name="cache" select="mapping-ext:getIDMappingExtCache()"/>
        <!--
              Stockez le temps universel coordonné dans le cache.
              Certains conteneurs ignore la déclaration de variable si la variable n'est pas utilisée.
              C'est pourquoi un commentaire est inséré ici, même si aucune sortie n'est nécéssaire pour
              appeler la méthode 'put' sur l'extension du cache.
           -->
        <xsl:comment>
          <xsl:value-of select="cache-ext:put($cache, $stateId, $utc_time, 1000)"/>
        </xsl:comment>
      </xsl:when>
      <xsl:when test="$requestType = 'resource'">
        <!-- Extrayez le cache et placez-le dans l'attribut recovered_state -->
        <xsl:variable name="cache" select="mapping-ext:getIDMappingExtCache()"/>
        <stsuuser:Attribute name="recovered_state"
                 type="urn:ibm:names:ITFIM:oauth:response:attribute">
          <stsuuser:Value>
            <xsl:value-of select="cache-ext:get($cache, $stateId)"/>
          </stsuuser:Value>
        </stsuuser:Attribute>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
```

Figure 40. Exemple de code XSL OAuth 2.0 avec gestion d'état

Gestion des clients de confiance

Tivoli Federated Identity Manager stocke des informations de client de confiance en fonction des décisions d'un propriétaire digne de confiance pour les clients.

La prise en charge du gestionnaire de clients de confiance s'applique aux fédérations OAuth 1.0 et OAuth 2.0.

Dans un flux OAuth, le propriétaire de la ressource est invité à fournir son accord sur les portées demandées par un client OAuth pour accéder à la ressource protégée. Le propriétaire de la ressource peut autoriser ou refuser la requête d'accès du client OAuth.

Le serveur OAuth ou le serveur d'autorisations utilise le noeud final du gestionnaire de clients de confiance pour gérer les informations sur les clients de confiance.

Au cours de l'étape d'autorisation, le serveur OAuth ou le serveur d'autorisation vérifie si un client OAuth avec une portée spécifique est stocké dans le noeud final de gestion des clients de confiance. Si le client OAuth a stocké des informations de portée, il n'est pas nécessaire que le serveur OAuth autorise le propriétaire de la ressource à accéder à la ressource protégée.

Les administrateurs peuvent enregistrer des informations d'un partenaire de client de confiance via les options suivantes :

- Stocker les informations de clients de confiance dans un cookie de navigateur
- Consentir automatiquement à toutes les décisions de confiance dans un environnement d'authentification fermé
- Stocker des informations de clients de confiance en mémoire (*à des fins de test uniquement*)

Par défaut, les informations des clients de confiance sont stockées dans le navigateur du propriétaire de la ressource en tant que cookies permanents. Tivoli Federated Identity Manager fournit un point d'extension `TrustedClientsManager` que les administrateurs peuvent utiliser pour écrire leur propre plug-in personnalisé afin de stocker et extraire les informations.

Présentation d'EAS OAuth

Le service EAS est un plug-in du service d'autorisation modulaire. Les concepteurs système peuvent utiliser l'autorisation IBM Tivoli Access Manager sous la forme d'un module complémentaire pour leurs propres modèles d'autorisation lorsqu'ils possèdent le service EAS.

Le plug-in EAS utilise les fonctions d'OAuth IBM Tivoli Federated Identity Manager.

A l'aide du service EAS d'OAuth, les décisions OAuth peuvent faire partie de l'autorisation standard dans les demandes WebSEAL. Cela signifie que WebSEAL peut être utilisé comme point d'application de l'autorisation pour l'accès aux ressources protégées OAuth. Le plug-in EAS OAuth peut être utilisé pour les protocoles OAuth 1.0 et 2.0.

Pour en savoir plus sur le plug-in EAS, voir la rubrique "External authorization service plug-ins" du centre de documentation e-business IBM Tivoli Access Manager : http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/welcome.htm.

Le service EAS d'OAuth est responsable des actions suivantes dans un processus d'autorisation :

- Réception de la requête d'autorisation.
- Vérification que toutes les données requises sont disponibles dans la requête.
- Construction d'un jeton de sécurité de requête (RST) et son envoi au point de décision de règles, qui est Tivoli Federated Identity Manager.
- Accord ou refus de l'accès à la ressource protégée en fonction de la décision reçue de Tivoli Federated Identity Manager.

Le service EAS d'OAuth communique avec Tivoli Federated Identity Manager via l'interface du service de jeton de sécurité. Pour plus d'informations, voir «Interface de service STS OAuth pour les points d'application d'autorisation», à la page 457.

Données OAuth

Tivoli Federated Identity Manager requiert des informations spécifiques concernant la demande de renvoi d'une décision d'autorisation.

Données de configuration

Le service EAS d'OAuth accepte deux paramètres de configuration facultatifs en tant que paramètres de chaîne de requête dans la requête. Ces paramètres affectent la composition du message envoyé par le service EAS d'OAuth à Tivoli Federated Identity Manager pour obtenir une décision d'autorisation.

Ces paramètres peuvent également être configurés de façon statique dans la section de configuration EAS OAuth. Pour plus d'informations, voir «Section [oauth-eas]», à la page 475.

- Mode (*OAuth10 ou OAuth20Bearer*)
- ID fédération (*l'ID du fournisseur de la fédération OAuth Tivoli Federated Identity Manager associé au client OAuth*)

Données d'autorisation

Les données de requête OAuth suivantes sont obtenues de l'en-tête de l'autorisation, de l'après-corps ou de la chaîne de requête.

Une requête OAuth 1.0 comprend les données suivantes :

- Domaine (*facultatif*)
- Clé de consommateur
- Jeton (*facultatif*)
- Méthode de signature
- Horodatage
- Nonce
- Signature
- Version (*facultatif*)

Une demande OAuth 2.0 requiert le jeton d'accès. Tivoli Federated Identity Manager prend en charge uniquement la spécification liée au jeton bearer.

Informations relatives aux ressources

Ces données sont obtenues à partir de la demande HTTP et sont utilisées par Tivoli Federated Identity Manager pour valider la signature OAuth. Elles sont envoyées à Tivoli Federated Identity Manager, quelle que soit la version OAuth.

Les informations relatives aux ressources incluent les données suivantes :

- Méthode de requête (*par exemple, GET ou POST*)
- Schéma (*par exemple, HTTP ou HTTPS*)
- En-tête de l'hôte issu de la requête
- Chemin de la requête
- Analyse de la demande
- Corps de demande

  Uniquement si les conditions suivantes s'appliquent :
  - Le corps d'entité comprend une seule partie.
  - Le corps d'entité respecte les exigences de codage du type de contenu `application/x-www-form-urlencoded`, comme défini par la spécification W3C HTML 4.0. Voir http://www.w3.org/TR/1998/REC-html40-19980424/.
  - L'en-tête de l'entité de requête HTTP inclut la zone d'en-tête **Content-Type** définie sur `application/x-www-form-urlencoded`.
- Port

  Uniquement si les conditions suivantes s'appliquent :
  - Il fait partie de l'URL de la requête.
  - Il ne s'agit pas du port par défaut du schéma. Par exemple, si le schéma est HTTP et que le port n'est pas 80.

WebSEAL utilise le plug-in EAS pour fournir les données requises et utiliser la fonction OAuth dans Tivoli Federated Identity Manager.

Réponses d'erreur

Une réponse HTTP indique le type d'erreur qui s'est produit lorsqu'une action d'un processus d'autorisation échoue.

Dans certains cas, les réponses d'erreur HTTP suivantes doivent être renvoyées au client :

- 400 Bad Request
- 401 Unauthorized
- 502 Bad Gateway

En ce qui concerne la réponse 401, un autre en-tête `WWW-Authenticate` est ajouté à la réponse dans le format suivant :

```
WWW-Authenticate: OAuth realm = <realm-name>
```

Le composant HTML des réponses est préinstallé à partir des fichiers qui ont été spécifiés dans la configuration EAS.

Voir le document WebSEAL Administration Guide pour plus de détails sur la façon de configurer les fichiers de modèle de réponse pour EAS OAuth.

Informations de configuration des fédérations et des partenaires

Tivoli Federated Identity Manager fournit des assistants pour créer des fédérations OAuth 1.0 et OAuth 2.0, et enregistrer un client en tant que partenaire. Complétez les formulaires appropriés avant d'exécuter les assistants de création de fédération et de partenaire.

Utilisez le formulaire approprié pour collecter des informations lors de la préparation du processus de configuration :

- «Formulaire de fournisseur de services OAuth 1.0»
- «Formulaire du partenaire de fournisseur de services OAuth 1.0», à la page 434
- «Formulaire de fournisseur de service OAuth 2.0», à la page 436
- «Formulaire de partenaire de fournisseur de services OAuth 2.0», à la page 440

Formulaire de fournisseur de services OAuth 1.0

Utilisez ce formulaire pour planifier vos propriétés lors de la création d'une fédération OAuth 1.0 et vous y reporter lors de l'exécution de l'assistant.

Le tableau suivant fournit les descriptions des propriétés de fédération OAuth 2.0, ainsi qu'un espace pour vous permettre d'écrire vos valeurs pour chaque propriété.

Tableau 114. Formulaire pour les propriétés de configuration de fédération OAuth 1.0

Propriété	Description	Votre valeur
Nom de la fédération	Indique le nom de la fédération. (**Obligatoire**) Utilisez un nom décrivant l'objet de la fédération. Entrez une valeur alphanumérique. Par exemple, un site de réseautage social peut être un serveur OAuth. Un service d'impression de photos qui peut imprimer les photos stockées dans le site de réseautage social peut être un client OAuth. Le nom de la fédération peut être : *MonRéseauSocial*	
Mon rôle	Spécifie votre rôle dans la fédération. **Valeur par défaut** : fournisseur de service Un fournisseur de service fournit un service aux utilisateurs. Dans la plupart des cas, le fournisseur de services OAuth protège les ressources et les propriétaires de ressources peuvent autoriser des clients OAuth à accéder à ces ressources protégées.	Fournisseur de services
Nom de la société	Spécifie le nom de la société qui crée cette fédération. La valeur peut correspondre à n'importe quelle chaîne de caractères. Vous pouvez également utiliser le caractère espace. (**Obligatoire**)	

Tableau 114. Formulaire pour les propriétés de configuration de fédération OAuth 1.0 (suite)

Propriété	Description	Votre valeur
Adresse URL de la société	Spécifie l'URL de la société qui crée cette fédération. (**Facultatif**) Par exemple : `http://www.example.com`	
Prénom et nom	Spécifie le nom de la personne à contacter de la société dans cette fédération. (**Facultatif**) Par exemple : `Jean Dupont`	
Adresse électronique	Spécifie l'adresse de courrier électronique de la personne à contacter de la société dans cette fédération. (**Facultatif**) Par exemple : `jeandupont@exemple.com`	
Numéro de téléphone	Spécifie le numéro de téléphone de la personne à contacter de la société dans cette fédération. (**Facultatif**) Par exemple : `+1-555-555-5555`	
Type de contact	Indique le type de contact. (**Facultatif**) Choix possibles : • Technique • Support • Administratif • Facturation • Autre	
Autres informations	Spécifie une zone de texte facultative pour la saisie d'informations de contact supplémentaires sur la fédération. (**Facultatif**)	
Protocole de fédération	Spécifie le protocole de la fédération. **Valeur par défaut** : OAuth 1.0	OAuth 1.0
Serveur point de contact	Indique l'adresse URL du serveur qui fait fonction de point de contact initial pour les demandes entrantes. L'adresse se compose d'une spécification de protocole, du nom d'hôte du serveur et (facultativement) d'un numéro de port. Lorsque WebSEAL est le serveur point de contact, la jonction WebSEAL est spécifiée. (**Obligatoire**) Exemple de valeur : `https://webseald.example.com/FIM`	

Tableau 114. Formulaire pour les propriétés de configuration de fédération OAuth 1.0 (suite)

Propriété	Description	Votre valeur
Fournisseur client OAuth	Indique le fournisseur client pour votre fédération OAuth 1.0. (**Obligatoire**) Vous pouvez sélectionner l'une des options suivantes : • Clients gérés par IBM Tivoli Federated Identity Manager • Clients gérés par un fournisseur client externe **Valeur par défaut** : Clients gérés par IBM Tivoli Federated Identity Manager	
Implémentation du fournisseur client externe	Indique la façon dont les clients OAuth sont gérés en externe. (**Obligatoire si l'option *Clients gérés par un fournisseur client externe* est sélectionnée en tant que Fournisseur client OAuth**) Vous pouvez écrire des implémentations personnalisées pour le point d'extension du fournisseur client OAuth 1.0. Tivoli Federated Identity Manager lit les données de configuration client à partir de votre source de configuration externe. Le plug-in de fournisseur client externe prend en charge GUIXML pour la configuration des paramètres.	
Décalage d'horloge maximum autorisé entre le serveur et le client OAuth (en secondes)	Indique le décalage d'horloge maximal entre le serveur OAuth et le client OAuth. Ce paramètre décrit le décalage d'horloge entre : • l'horloge du serveur OAuth • l'horloge du client OAuth Le valeur du décalage d'horloge est généralement un petit nombre. (**Obligatoire**) **Valeur par défaut** : 300	
Durée de vie des droits d'accès et du code de vérification temporaires (en secondes)	Indique la validité des droits d'accès et du code de vérification temporaires en secondes. (**Obligatoire**) **Valeur par défaut** : 300	
Durée de vie maximale des droits de jeton (en secondes)	Indique en secondes la durée de vie du jeton d'accès OAuth. A l'expiration de cette durée de vie, le client ne peut plus accéder à la ressource protégée. Le propriétaire de la ressource doit réautoriser le client à accéder à la ressource protégée. (**Obligatoire**) **Valeur par défaut** : 604800	

Tableau 114. Formulaire pour les propriétés de configuration de fédération OAuth 1.0 (suite)

Propriété	Description	Votre valeur
Options de mappage d'identité	Indique la façon dont vous voulez effectuer le mappage des identités pour votre fédération OAuth. (**Obligatoire**) Sélectionnez une des options suivantes : • **Utilisez des règles de mappage XSLT ou Javascript pour le mappage d'identité** Sélectionnez cette option lorsque vous créez une règle de mappage XSLT ou Javascript pour le mappage d'identité. • **Utilisez Tivoli Directory Integrator pour le mappage** Sélectionnez cette option lorsque vous disposez d'une ligne d'assemblage Tivoli Directory Integrator pour le mappage d'identité. • **Utiliser une instance de modèle de mappage personnalisé** Sélectionnez cette option lorsque vous disposez d'un module de service d'accréditation personnalisé pour le mappage d'identité.	
Fichier de règles de mappage d'identité	Indique le nom du fichier de règles de mappage si la règle XSLT ou JavaScript pour le mappage d'identité est utilisée.	
Modules de mappage personnalisés	Indique le nom du module si le module de mappage personnalisé est utilisé comme règle de mappage d'identité.	*Prenez note du nom du module :*

Formulaire du partenaire de fournisseur de services OAuth 1.0

Utilisez ce formulaire pour planifier les propriétés de votre partenaire OAuth 1.0 et vous y référer lors de l'exécution de l'assistant.

Remarque : La création et la configuration de partenaire s'appliquent uniquement aux fédérations qui utilisent Tivoli Federated Identity Manager en tant que fournisseur de client.

Le tableau suivant fournit de descriptions des propriétés de partenaire OAuth 1.0, ainsi qu'un espace pour vous permettre d'écrire vos valeurs pour chaque propriété.

Tableau 115. Formulaire pour les propriétés de configuration de partenaire OAuth 1.0

Propriété	Description	Votre valeur
Nom de la société de client OAuth	Indique le nom de la société de ce partenaire. Il peut être constitué de n'importe quel chaîne de caractères. (**Obligatoire**)	
Adresse URL de la société	Indique l'URL de la société de ce partenaire. (**Facultatif**) Par exemple : `http://www.example.com`	

Tableau 115. Formulaire pour les propriétés de configuration de partenaire OAuth 1.0 (suite)

Propriété	Description	Votre valeur
Prénom et nom	Indique le nom de la personne à contacter pour ce partenaire. (**Facultatif**) Par exemple : `Jean Dupont`	
Adresse électronique	Indique l'adresse de courrier électronique de la personne à contacter pour ce partenaire. (**Facultatif**) Par exemple : `jeandupont@exemple.com`	
Numéro de téléphone	Indique le numéro de téléphone de la personne à contacter pour ce partenaire. (**Facultatif**) Par exemple : `+1-555-555-5555`	
Type de contact	Indique le type de contact. (**Facultatif**) Choix possibles : • Technique • Support • Administratif • Facturation • Autre	
Autres informations	Spécifie une zone de texte facultative pour la saisie d'informations de contact supplémentaires sur le partenaire. Vous pouvez utiliser n'importe quelle chaîne de caractères. (**Facultatif**)	
Identificateur du client	Spécifie un identificateur unique fourni au client OAuth pour lui permettre de s'identifier auprès du serveur OAuth. Vous ne pouvez pas changer cette valeur dans cet écran. (**Obligatoire**)	
Secret partagé du client	Indique un secret partagé entre le client OAuth et un serveur OAuth qui est utilisé pour la signature des demandes. Cette zone est générée automatiquement par le serveur OAuth, mais vous pouvez la remplacer par une valeur de votre choix. (**Obligatoire**)	
URI de rappel du client	Indique un identificateur URI de rappel vers lequel est redirigé le propriétaire de la ressource lorsque l'autorisation est terminée. Vous pouvez activer la prise en charge d'une configuration externe pour recevoir des rappels en définissant cette valeur sur `oob`. Si vous n'enregistrez pas un identificateur URI de rappel, Tivoli Federated Identity Manager le traite comme un `oob`. (**Facultatif**)	

Tableau 115. Formulaire pour les propriétés de configuration de partenaire OAuth 1.0 (suite)

Propriété	Description	Votre valeur
Remplacer l'identificateur URI de rappel de client enregistré	Détermine s'il faut remplacer l'identificateur URI de rappel de client enregistré. Cochez cette case pour remplacer l'identificateur URI de rappel de client enregistré par le paramètre URI de rappel dans la demande de droits d'accès temporaires. (**Facultatif**)	

Formulaire de fournisseur de service OAuth 2.0

Utilisez ce formulaire pour planifier vos propriétés lors de la création d'une fédération OAuth 2.0, et vous y référer lors de l'exécution de l'assistant.

Le tableau suivant fournit les descriptions des propriétés de fédération OAuth 2.0, ainsi qu'un espace pour vous permettre d'écrire vos valeurs pour chaque propriété.

Tableau 116. Formulaire pour les propriétés de configuration de fédération OAuth 2.0

Propriété	Description	Votre valeur
Nom de la fédération	Indique le nom de la fédération. (**Obligatoire**) Utilisez un nom décrivant l'objet de la fédération. Entrez une valeur alphanumérique. Par exemple, un site de réseautage social peut être un serveur d'autorisation. Un service d'impression de photos qui peut imprimer les photos stockées dans le site de réseautage social peut être un client OAuth. Le nom de la fédération peut être *MonréseauSocial*.	
Mon rôle	Spécifie votre rôle dans la fédération. (**Obligatoire**) **Valeur par défaut** : fournisseur de service Un fournisseur de service fournit un service aux utilisateurs. Dans la plupart des cas, le fournisseur de services OAuth protège les ressources et les propriétaires de ressources peuvent autoriser des clients OAuth à accéder à ces ressources protégées.	Fournisseur de services
Nom de la société	Spécifie le nom de la société qui crée cette fédération. La valeur peut correspondre à n'importe quelle chaîne de caractères. Vous pouvez également utiliser le caractère espace. (**Obligatoire**)	
Adresse URL de la société	Spécifie l'URL de la société qui crée cette fédération. (**Facultatif**) Par exemple : `http://www.example.com`	
Prénom et nom	Spécifie le nom de la personne à contacter de la société dans cette fédération. (**Facultatif**) Par exemple : `Jean Dupont`	

Tableau 116. Formulaire pour les propriétés de configuration de fédération OAuth 2.0 (suite)

Propriété	Description	Votre valeur
Adresse électronique	Spécifie l'adresse de courrier électronique de la personne à contacter de la société dans cette fédération. (**Facultatif**) Par exemple : `jeandupont@exemple.com`	
Numéro de téléphone	Spécifie le numéro de téléphone de la personne à contacter de la société dans cette fédération. (**Facultatif**) Par exemple : `+1-555-555-5555`	
Type de contact	Indique le type de contact. (**Facultatif**) Choix possibles : • Technique • Support • Administratif • Facturation • Autre	
Autres informations	Spécifie une zone de texte facultative pour la saisie d'informations de contact supplémentaires sur la fédération. (**Facultatif**)	
Protocole de fédération	Spécifie le protocole de la fédération. (**Obligatoire**) **Valeur par défaut** : OAuth 2.0	OAuth 2.0
Serveur point de contact	Indique l'adresse URL du serveur qui fait fonction de point de contact initial pour les demandes entrantes. (**Obligatoire**) L'adresse se compose d'une spécification de protocole, du nom d'hôte du serveur et (facultativement) d'un numéro de port. Lorsque WebSEAL est le serveur point de contact, la jonction WebSEAL est spécifiée. Exemple de valeur : `https://webseald.example.com/FIM`	
Fournisseur client OAuth	Indique le fournisseur client pour votre fédération OAuth 2.0. (**Obligatoire**) Vous pouvez sélectionner l'une des options suivantes : • Clients gérés par IBM Tivoli Federated Identity Manager • Clients gérés par un fournisseur client externe **Valeur par défaut** : Clients gérés par IBM Tivoli Federated Identity Manager	

Tableau 116. Formulaire pour les propriétés de configuration de fédération OAuth 2.0 (suite)

Propriété	Description	Votre valeur
Implémentation du fournisseur client externe	Indique la façon dont les clients OAuth sont gérés en externe. (**Obligatoire si l'option *Clients gérés par un fournisseur client externe* est sélectionnée en tant que Fournisseur client OAuth**) Vous pouvez écrire des implémentations personnalisées pour le point d'extension du fournisseur client OAuth 2.0. Tivoli Federated Identity Manager lit les données de configuration client à partir de votre source de configuration externe. Le plug-in du fournisseur client externe prend en charge GUIXML pour la configuration des paramètres.	
Types d'accord d'autorisation	Indique la liste des types d'accord pris en charge pour une fédération OAuth 2.0. (**Obligatoire**) Vous devez sélectionner au moins un type d'accord : • Code d'autorisation • Accord implicite • Droits d'accès du client • Données d'identification par mot de passe du propriétaire de la ressource **Valeur par défaut** : code d'autorisation et accord implicite	
Durée de vie maximale de l'accord d'autorisation (en secondes)	Indique la durée maximale d'un accord selon lequel le propriétaire a autorisé le client OAuth à accéder à la ressource protégée. (**Obligatoire**) Cette zone s'applique uniquement au code d'autorisation et aux types d'accord de données d'identification par mot de passe du propriétaire de la ressource. Cette durée de vie affecte la validité d'un code d'autorisation, d'un jeton d'accès et d'un jeton d'actualisation. La valeur de cette durée de vie doit être supérieure aux valeurs indiquées pour les durées de vue du code d'autorisation et du jeton d'accès. A l'expiration de cette durée de vie, le propriétaire de la ressource doit réautoriser le client OAuth à obtenir un accord d'autorisation pour accéder à la ressource protégée. **Valeur par défaut** : 604800	
Durée de vie du code d'autorisation (en secondes)	Indique la validité du code d'autorisation en secondes. (**Obligatoire**) Cette option s'applique uniquement à un type d'accord de code d'autorisation. Ce serveur d'autorisation génère un code d'autorisation et le fournit au client OAuth. Le client OAuth utilise le code d'autorisation en échange d'un jeton d'accès. **Valeur par défaut** : 300	

Tableau 116. Formulaire pour les propriétés de configuration de fédération OAuth 2.0 (suite)

Propriété	Description	Votre valeur
Emettre un jeton d'actualisation	Indique si un jeton d'actualisation est transmis au client OAuth. (**Facultatif**) Un jeton d'actualisation permet d'obtenir un nouvel ensemble constitué d'un jeton d'accès et d'un jeton d'actualisation. Cette option s'applique uniquement aux types d'accord de code d'autorisation et de données d'identification par mot de passe du propriétaire de la ressource. **Valeur par défaut** : Non (la case à cocher n'est pas sélectionnée)	
Type de jeton d'accès OAuth	Indique le type de jeton d'accès OAuth utilisé par un client OAuth pour effectuer des demandes de ressources protégées. (**Obligatoire**) La valeur par défaut est **Type de jeton bearer OAuth**. Un jeton bearer est un jeton de sécurité qui accorde la propriété à tout utilisateur qui détient le jeton. **Valeur par défaut** : Type de jeton bearer OAuth	
Durée de vie du jeton d'accès (en secondes)	Indique la validité du jeton d'accès en secondes. (**Obligatoire**) A l'expiration de cette durée de vie, le client OAuth ne peut plus utiliser le jeton d'accès courant pour accéder à la ressource protégée. **Valeur par défaut** : 3600	
Options de mappage d'identité	Indique la façon dont vous voulez effectuer le mappage des identités pour votre fédération OAuth. (**Obligatoire**) Sélectionnez une des options suivantes : • **Utilisez des règles de mappage XSLT ou Javascript pour le mappage d'identité** Sélectionnez cette option lorsque vous créez une règle de mappage XSLT ou Javascript pour le mappage d'identité. • **Utilisez Tivoli Directory Integrator pour le mappage** Sélectionnez cette option lorsque vous disposez d'une ligne d'assemblage Tivoli Directory Integrator pour le mappage d'identité. • **Utiliser une instance de modèle de mappage personnalisé** Sélectionnez cette option lorsque vous disposez d'un module de service d'accréditation personnalisé pour le mappage d'identité.	
Fichier de règles de mappage d'identité	Indique le nom du fichier de règles de mappage si la règle XSLT ou JavaScript pour le mappage d'identité est utilisée.	
Modules de mappage personnalisés	Indique le nom du module si le module de mappage personnalisé est utilisé comme règle de mappage d'identité.	*Prenez note du nom du module :*

Formulaire de partenaire de fournisseur de services OAuth 2.0

Utilisez ce formulaire pour planifier vos propriétés pour votre partenaire OAuth 2.0 et vous y référer lors de l'exécution de l'assistant.

Remarque : La création et la configuration de partenaire sont applicables uniquement aux fédérations qui utilisent Tivoli Federated Identity Manager comme fournisseur client.

Le tableau suivant fournit les descriptions des propriétés du partenaire OAuth 2.0, ainsi qu'un espace pour vous permettre d'écrire vos valeurs pour chaque propriété.

Tableau 117. Formulaire pour les propriétés de configuration de partenaire OAuth 2.0

Propriété	Description	Votre valeur
Nom de la société de client OAuth	Indique le nom de la société de ce partenaire. Il peut être constitué de n'importe quel chaîne de caractères. (**Obligatoire**)	
Adresse URL de la société	Indique l'URL de la société de ce partenaire. (**Facultatif**) Par exemple : `http://www.example.com`	
Prénom et nom	Indique le nom de la personne à contacter pour ce partenaire. (**Facultatif**) Par exemple : `Jean Dupont`	
Adresse électronique	Indique l'adresse de courrier électronique de la personne à contacter pour ce partenaire. (**Facultatif**) Par exemple : `jeandupont@exemple.com`	
Numéro de téléphone	Indique le numéro de téléphone de personne à contacter pour ce partenaire. (**Facultatif**) Par exemple : `+1-555-555-5555`	
Type de contact	Indique le type de contact. (**Facultatif**) Choix possibles : • Technique • Support • Administratif • Facturation • Autre	
Autres informations	Spécifie une zone de texte facultative pour la saisie d'informations de contact supplémentaires sur le partenaire. Vous pouvez utiliser n'importe quelle chaîne de caractères. (**Facultatif**)	

Tableau 117. Formulaire pour les propriétés de configuration de partenaire OAuth 2.0 (suite)

Propriété	Description	Votre valeur
Identificateur du client	Spécifie un identificateur unique fourni au client pour lui permettre de s'identifier auprès du serveur d'autorisation. Cette zone est générée automatiquement par le serveur d'autorisation, mais vous pouvez la remplacer par une valeur alphanumérique de votre choix. (**Obligatoire**)	
Secret partagé du client	Spécifie un secret partagé entre le client OAuth et le serveur d'autorisation. Cette zone est générée automatiquement par le serveur OAuth, mais vous pouvez la remplacer par une valeur de votre choix. Si vous n'enregistrez pas un secret pour le client OAuth, celui-ci devient un client publique. Si vous enregistrez un secret, le client OAuth devient un client confidentiel. (**Facultatif**)	
URI de redirection du client	Spécifie un identificateur URI vers lequel le propriétaire de la ressource est redirigé lorsque l'autorisation est terminée. Le serveur d'autorisation renvoie des accords ou des jetons uniquement à cet identificateur URI de redirection ou son enfant. (**Facultatif**)	

Chapitre 28. Configuration d'une fédération OAuth

Pour configurer une fédération OAuth, vous devez créer la fédération, ajouter votre partenaire à votre fédération, et configurer le point d'application pour la ressource protégée.

Configuration d'une fédération de fournisseurs de services OAuth

Utilisez l'assistant de fédération pour créer et configurer une fédération de fournisseurs de services.

Avant de commencer

Avant de commencer cette procédure, complétez le formulaire approprié pour le protocole OAuth :

- «Formulaire de fournisseur de services OAuth 1.0», à la page 431
- «Formulaire de fournisseur de service OAuth 2.0», à la page 436

Pourquoi et quand exécuter cette tâche

La description des zones de l'assistant de fédération est détaillée dans l'aide en ligne.

Procédure

1. Connectez-vous à la console Integrated Solutions Console.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Les portlets Domaine en cours et Fédérations s'ouvrent.
3. Cliquez sur **Créer** pour lancer l'assistant de fédération.
4. Utilisez le formulaire complété comme guide pour renseigner les zones.
5. Pour passer au panneau suivant, cliquez sur **Suivant**.
   a. (Facultatif) Si vous avez besoin de revenir en arrière pour modifier un paramètre de configuration, cliquez sur **Précédent**.
   b. (Facultatif) Si vous souhaitez mettre fin à la configuration, cliquez sur **Annuler**.

   Lorsque vous avez complété tous les écrans de configuration, le panneau Récapitulatif s'affiche.
6. Vérifiez que les paramètres de configuration sont corrects.
7. Cliquez sur **Terminer**. Le portlet Création de fédération terminée s'affiche.
8. (Facultatif) Si vous utilisez le fournisseur client interne, vous pouvez ajouter votre partenaire maintenant ou ultérieurement.
   - Cliquez sur **Ajouter un partenaire** pour lancer l'assistant Partenaire et enregistrer un client OAuth à votre fédération de fournisseurs de services OAuth. Voir les étapes de la section «Ajout d'un partenaire à une fédération OAuth», à la page 446.
   - Cliquez sur **Terminé** pour ajouter votre partenaire ultérieurement.
9. (Facultatif) Si vous utilisez un fournisseur client externe, cliquez sur **Terminé** pour revenir au portlet Fédérations.

10. Cliquez sur le bouton **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour déployer les modifications.

Activation de la validation Oauth à deux jambes

Configurez les propriétés d'une fédération OAuth 1.0 existante lorsque vous souhaitez activer la validation OAuth à deux jambes dans votre fédération.

Avant de commencer

Pour effectuer cette tâche, une fédération OAuth 1.0 doit déjà exister. Pour configurer une fédération OAuth 1.0 , voir «Configuration d'une fédération de fournisseurs de services OAuth», à la page 443.

Si l'intercepteur de relations de confiance de WebSphere sert de point d'application, vous devez créer dans le registre WebSphere un utilisateur correspondant au nom d'utilisateur renvoyé par votre règle de mappage. Voir le centre documentation de WebSphere Application Server Version 6.0 pour plus d'informations sur la création d'utilisateurs. Pour comprendre comment agissent le point d'application et le service de jeton de sécurité dans une validation OAuth à deux jambes, voir «Interface du service de jeton de sécurité pour flux OAuth à deux jambes», à la page 414.

Procédure

1. Connectez-vous à la console Integrated Solutions Console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Le panneau Fédérations affiche la liste des fédérations configurées.
3. Sélectionnez une fédération OAuth dans le tableau, puis cliquez sur **Propriétés**. Le panneau **Propriétés de la fédération** s'ouvre.
4. Cochez la case **Activer la validation OAuth à deux jambes**.
5. Cliquez sur **OK** pour quitter le panneau.
6. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour recharger vos modifications.

Résultats

La validation OAuth à deux jambes est activée dans la fédération.

Configuration d'un serveur point de contact WebSEAL pour la fédération OAuth

Si vous envisagez d'utiliser WebSEAL en tant que serveur point de contact de votre fédération OAuth, vous devez le configurer à l'aide de l'utilitaire de configuration.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Avant d'effectuer cette procédure :

- Le profil du point de contact WebSEAL doit être activé.
- Vous devez connaître l'ID (par défaut : sec_master) et le mot de passe de l'utilisateur d'administration Tivoli Access Manager.

Pourquoi et quand exécuter cette tâche

L'assistant de fédération comporte un bouton qui vous permet d'ouvrir l'utilitaire de configuration. Cette procédure décrit en détail comment ouvrir et exécuter l'utilitaire. L'utilitaire configure les noeuds finals sur le serveur WebSEAL, crée une jonction WebSEAL, relie les listes de contrôle d'accès adaptées et active les méthodes d'authentification adéquates.

Les étapes indiquées concernent les fédérations OAuth 1.0 et 2.0.

Pour configurer WebSEAL en tant que serveur point de contact, procédez comme suit :

Procédure

1. Une fois la fédération créée, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour recharger vos modifications.
2. Cliquez sur **Terminé** pour revenir au panneau Fédérations.
3. Cliquez sur **Télécharger l'outil de configuration Tivoli Access Manager**.
4. Enregistrez l'outil de configuration sur le système de fichiers de l'ordinateur hébergeant le serveur WebSEAL.
5. Démarrez l'outil de configuration depuis une ligne de commande. La syntaxe est la suivante :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf
   ```

   Remarques :

   - Si la norme FIPS (Federal Information Processing Standards) est activée, une fabrique de connexions sécurisées doit être indiquée. Par exemple :

     ```
     java -jar /download_dir/tfimcfg.jar -action tamconfig
     -cfgfile webseald-instance_name.conf -sslfactory TLS
     ```

   - **Pour les fédérations OAuth 1.0 :** si un client OAuth envoie des paramètres de protocole OAuth via un en-tête d'autorisation HTTP, le serveur OAuth doit être en mesure d'accepter cet en-tête. Utilisez l'option `-b ignore` sur la jonction entre WebSEAL et Tivoli Federated Identity Manager pour transférer l'en-tête d'autorisation HTTP au serveur d'arrière-plan. Cette option n'est pas obligatoire sur la jonction si le client OAuth utilise la chaîne de requête ou la méthode POST.

- **Pour les fédérations OAuth 2.0 :** si un client OAuth accède à un point d'application des règles qui attend un en-tête d'autorisation HTTP, le serveur OAuth doit être en mesure d'accepter cet en-tête. Utilisez l'option `-b ignore` sur la jonction entre WebSEAL et le point d'application des règles pour transmettre l'en-tête d'autorisation HTTP au serveur d'arrière-plan. Cette option n'est nécessaire que si le point d'application des règles qui lit l'en-tête d'autorisation OAuth se trouve sur un serveur situé derrière WebSEAL. Vous n'avez pas besoin d'exécuter l'option `-b ignore` lorsque vous utilisez le point d'application de service EAS WebSEAL pour OAuth 2.0.

Exemple

Par exemple, lorsque vous avez placé le fichier `tfimcfg.jar` dans le répertoire `/tmp` et que le nom de l'instance WebSEAL est `default`, la commande est la suivante :

```
java -jar /tmp/tfimcfg.jar -action tamconfig -cfgfile webseald-default
```

Pour plus d'informations, voir Annexe A, «Référence de tfimcfg», à la page 827.

Configuration de WebSphere en tant que serveur point de contact

Tivoli Federated Identity Manager est configuré pour utiliser par défaut Tivoli Access Manager WebSEAL en tant que serveur point de contact. Pour configurer WebSphere en tant que serveur point de contact, vous devez procéder à une modification de la configuration.

Procédure

1. Connectez-vous à la console d'administration.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Point de contact**.
3. Sélectionnez **WebSphere**.
4. Cliquez sur **Activer**.

Résultats

Le serveur WebSphere est désormais configuré en tant que point de contact.

Ajout d'un partenaire à une fédération OAuth

Vous pouvez ajouter un partenaire à votre fédération OAuth par l'intermédiaire de la console d'administration.

Avant de commencer

Remarque : Vous pouvez ajouter des partenaires à une fédération uniquement si celle-ci est configurée pour utiliser Tivoli Federated Identity Manager comme fournisseur client.

Avant de commencer cette procédure, complétez le formulaire approprié pour le protocole OAuth :

- «Formulaire du partenaire de fournisseur de services OAuth 1.0», à la page 434
- «Formulaire de partenaire de fournisseur de services OAuth 2.0», à la page 440

Pourquoi et quand exécuter cette tâche

Cette procédure s'applique aux fédérations OAuth 1.0 et OAuth 20 comportant des fournisseurs clients internes.

Après avoir complété le fournisseur relatif au partenaire approprié, utilisez l'assistant Partenaire de la console pour ajouter la partenaire.

Pour obtenir une description détaillée des zones de l'assistant de fédération, consultez l'aide en ligne.

Procédure

1. Connectez-vous à la console Integrated Solutions Console.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Configuration de la connexion unique fédérée** > **Partners**. Le portlet Domaine en cours et partenaires de fédération s'ouvre.
3. Cliquez sur **Créer** pour démarrer l'assistant.
4. Sélectionnez la fédération OAuth à laquelle vous voulez ajouter un partenaire.
5. Cliquez sur **Suivant**.
6. Entrez les propriétés de contact.
7. Cliquez sur **Suivant**.
8. Configurez l'enregistrement du client OAuth.
9. Cliquez sur **Suivant**.
10. Cliquez sur **Suivant** pour afficher un récapitulatif de toute les informations que vous avez entrées.
11. Vérifiez que les paramètres sont corrects.
12. Cliquez sur **Terminer**. Le portlet Ajout de partenaire terminé s'ouvre.

    Remarque : Le partenaire a été ajouté à la fédération, mais il a été désactivé par mesure de sécurité.
13. Cliquez sur **Activer le partenaire** pour activer ce partenaire.
14. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour déployer les modifications.

Configuration de l'intercepteur de relations de confiance (TAI) OAuth WebSphere

Vous pouvez configurer le composant intercepteur de relations de confiance WebSphere en tant que point d'application d'autorisation OAuth.

Avant de commencer

Passez en revue les propriétés du point d'application que vous devez configurer pour une fédération OAuth 1.0 ou OAuth 2.0. Pour plus d'informations, voir «Propriétés personnalisées de l'intercepteur de relations de confiance et du filtre de servlet OAuth», à la page 469.

Pourquoi et quand exécuter cette tâche

L'intercepteur de relations de confiance OAuth WebSphere est destiné uniquement à protéger l'accès aux ressources WebSphere pour lesquelles une authentification

est nécessaire. Les intercepteurs de relations de confiance ne sont pas démarrés à chaque requête Web, mais seulement lorsque l'authentification est demandée.

Les étapes concernent les fédérations OAuth 1.0 et 2.0.

Procédure

1. Copiez le fichier `com.tivoli.am.fim.ws.oauth.jar` situé aux emplacements suivants :
   - AIX, Linux ou Solaris

     `/opt/IBM/FIM/tools/oauth/com.tivoli.am.fim.ws.oauth.jar`
   - Windows

     `C:\Program Files\IBM\FIM\tools\oauth\com.tivoli.am.fim.ws.oauth.jar`

   dans les répertoires suivants :
   - AIX, Linux ou Solaris

     `/opt/IBM/WebSphere/AppServer/lib/ext`
   - Windows

     `C:\Program Files\IBM\WebSphere\AppServer\lib\ext`
2. Connectez-vous à la console d'administration de WebSphere.

   Pour WebSphere 6.1 :
   Sélectionnez **Sécurité** > **Administration, applications et infrastructure sécurisées** > **Sécurité Web** > **Relation de confiance**.

   Pour WebSphere 7 et WebSphere 8 :
   Sélectionnez **Sécurité** > **Sécurité globale** > **Sécurité Web et SIP** > **Relation de confiance**.
3. Sélectionnez **Activer la relation de confiance**.
4. Cliquez sur **Appliquer**.
5. Cliquez sur **Intercepteurs**.
6. Cliquez **Nouveau** pour ajouter un nouvel intercepteur.
7. Entrez le nom de classe de l'intercepteur :

   `com.tivoli.am.fim.ws.oauth.tai.OAuthTAI`
8. Cliquez sur **Appliquer**.
9. Cliquez sur **Propriétés personnalisées**.
10. Ajoutez des propriétés personnalisées pour votre environnement. Voir la liste des propriétés dans la rubrique «Propriétés personnalisées de l'intercepteur de relations de confiance et du filtre de servlet OAuth», à la page 469.
11. Sauvegardez la mise à jour de la configuration.
12. Redémarrez WebSphere.

Configuration du filtre de servlet OAuth WebSphere

Vous pouvez configurer le composant filtre de servlet WebSphere en tant que point d'application d'autorisation OAuth.

Pourquoi et quand exécuter cette tâche

Cette rubrique s'applique aux fédérations OAuth 1.0 et OAuth 2.0.

Le filtre de servlet OAuth gère l'authentification des ressources protégées hébergées sur WebSphere de la même façon que le fait l'intercepteur de relations de confiance.

Vous pouvez utiliser le fichier com.tivoli.am.fim.ws.oauth.jar en tant que bibliothèque d'applications du filtre de servlet.

Procédure

1. Ajoutez le fichier com.tivoli.am.fim.ws.oauth.jar à votre application de fichiers d'archive d'entreprise personnalisée.
2. Ouvrez le fichier MANIFEST.MF de l'application WAR.
3. Indiquez com.tivoli.am.fim.ws.oauth.jar dans la zone **class path header**.
4. Enregistrez le fichier manifeste.
5. Ouvrez le descripteur de déploiement de la ressource protégée (fichier web.xml).
6. Ajoutez le filtre avec les paramètres d'initialisation afin qu'il corresponde à votre environnement. Voir la section «Propriétés personnalisées de l'intercepteur de relations de confiance et du filtre de servlet OAuth», à la page 469 pour plus d'informations sur les paramètres d'initialisation. Le filtre de servlet possède les mêmes paramètres que les propriétés de configuration que l'teur de relations de confiance.

Exemple

La figure 41, à la page 450 illustre un exemple de code (web.xml) pour OAuth 1.0 contenant des pages JSP protégées par le filtre de servlet.

```
  <display-name>com.tivoli.am.fim.war.fimivt</display-name>
  <filter>
    <description>Performs OAuth authorization</description>
    <display-name>OAuth servlet filter</display-name>
    <filter-name>OAuth servlet filter</filter-name>
    <filter-class>com.tivoli.am.fim.ws.oauth.sf.OAuthServletFilter</filter-class>
       <init-param>
      <description/>
      <param-name>DefaultMode</param-name>
      <param-value>OAuth10</param-value>
    </init-param>
    <init-param>
      <description/>
      <param-name>ModeParameterName</param-name>
      <param-value>mode</param-value>
    </init-param>
    <init-param>
      <description/>
      <param-name>URIPrefix</param-name>
      <param-value>/fimivt/oauth/sfprotected.jsp</param-value>
    </init-param>
    <init-param>
      <description/>
      <param-name>STSEndpoint</param-name>
      <param-value>http://server.oauth.com/TrustServer/SecurityTokenService</param-value>
    </init-param>
    <init-param>
      <description/>
      <param-name>OAuthRealm</param-name>
      <param-value>https://server.oauth.com//</param-value>
    </init-param>
    <init-param>
      <description/>
      <param-name>PointOfContact</param-name>
      <param-value>https://server.oauth.com/</param-value>
    </init-param>
     <init-param>
      <description/>
      <param-name>DefaultFederationId</param-name>
      <param-value>https://server.oauth.com/FIM/MySocialNetwork/oauth10</param-value>
    </init-param>
  <init-param>
      <description/>
      <param-name>OAuthTokenCacheSize</param-name>
      <param-value>2</param-value>
    </init-param></filter>
  <init-param>
      <description/>
      <param-name>FederationIdRequestParameterName</param-name>
      <param-value>FederationId</param-value>
    </init-param>
</filter>
  <filter-mapping>
    <filter-name>OAuth servlet filter</filter-name>
    <url-pattern>/oauth/sfprotected.jsp</url-pattern>
  </filter-mapping>
```

Figure 41. Exemple de code JavaScript pour OAuth 1.0

La figure 42, à la page 451 illustre un exemple de code (`web.xml`) pour OAuth 2.0 contenant des pages JSP protégées par le filtre de servlet

```
<display-name>com.tivoli.am.fim.war.fimivt</display-name>
<filter>
  <description>Performs OAuth authorization</description>
  <display-name>OAuth servlet filter</display-name>
  <filter-name>OAuth servlet filter</filter-name>
  <filter-class>com.tivoli.am.fim.ws.oauth.sf.OAuthServletFilter</filter-class>
 <init-param>
    <description/>
    <param-name>DefaultMode</param-name>
    <param-value>OAuth20Bearer</param-value>
  </init-param>
  <init-param>
    <description/>
    <param-name>ModeParameterName</param-name>
    <param-value>mode</param-value>
  </init-param>
  <init-param>
    <description/>
    <param-name>URIPrefix</param-name>
    <param-value>/fimivt/oauth/sfprotected.jsp</param-value>
  </init-param>
  <init-param>
    <description/>
    <param-name>STSEndpoint</param-name>
    <param-value>http://server.oauth.com/TrustServer/SecurityTokenService</param-value>
  </init-param>
  <init-param>
    <description/>
    <param-name>OAuthRealm</param-name>
    <param-value>https://server.oauth.com//</param-value>
  </init-param>
  <init-param>
    <description/>
    <param-name>PointOfContact</param-name>
    <param-value>https://server.oauth.com/</param-value>
  </init-param>
  <init-param>
    <description/>
    <param-name>DefaultFederationId</param-name>
    <param-value>https://server.oauth.com/FIM/MySocialNetwork/oauth20</param-value>
  </init-param>
<init-param>
    <description/>
    <param-name>FederationIdRequestParameterName</param-name>
    <param-value>FederationId</param-value>
  </init-param>
<init-param>
    <description/>
    <param-name>OAuthTokenCacheSize</param-name>
    <param-value>2</param-value>
  </init-param></filter>
<filter-mapping>
  <filter-name>OAuth servlet filter</filter-name>
  <url-pattern>/oauth/sfprotected.jsp</url-pattern>
</filter-mapping>
```

Figure 42. Exemple de code JavaScript pour OAuth 2.0

Configuration EAS OAuth WebSEAL

Configurez le service d'autorisation externe (EAS) OAuth, qui représente un point d'application de règles WebSEAL, pour la prise en charge de OAuth 1.0 et OAuth 2.0.

Configurez le service EAS OAuth WebSEAL à l'aide de l'une des méthodes suivantes :

- Configuration manuelle du service EAS OAuth WebSEAL
- «Configuration du service EAS OAuth WebSEAL à l'aide de l'outil tfimcfg», à la page 454

Les décisions OAuth sont incluses à l'autorisation standard des demandes WebSEAL. Les deux méthodes de configuration garantissent la transmission des données adéquates à chaque service EAS OAuth pour chaque demande.

Concepts associés:

«Présentation d'EAS OAuth», à la page 428
Le service EAS est un plug-in du service d'autorisation modulaire. Les concepteurs système peuvent utiliser l'autorisation IBM Tivoli Access Manager sous la forme d'un module complémentaire pour leurs propres modèles d'autorisation lorsqu'ils possèdent le service EAS.

Configuration manuelle du service EAS OAuth WebSEAL

Configurez manuellement le service EAS OAuth lorsque vous ne souhaitez pas utiliser les valeurs par défaut.

Avant de commencer

IBM® Tivoli® Access Manager for e-business version 6.1.1 ou ultérieure doit être installé. Si la version 6.1.1 est installée, le groupe de correctifs 5 ou ultérieur doit être appliqué.

Procédure

1. Activez le services EAS OAuth.
   a. Ouvrez le fichier de configuration par défaut WebSEAL dans n'importe quel éditeur de fichier.

      UNIX ou Linux

      ```
      /opt/pdweb/etc/webseald-default.conf
      ```

      Windows

      ```
      C:\Program Files\Tivoli\PDWeb\etc\webseald-default.conf
      ```

   b. Indiquez l'entrée <*policy-trigger*> dans la section `[aznapi-external-authzn-services]`. Le service EAS OAuth requiert un unique paramètre qui correspond au fichier de configuration contenant les données de configuration du service EAS OAuth. Le nom du plug-in pour le service EAS OAuth est `amwoautheas`. Sa bibliothèque se trouve dans le répertoire `pdwebrte/lib`.

   Par exemple :

   UNIX ou Linux

   ```
   oauth_pop_trigger = /opt/pdwebrte/lib/libamwoautheas.so &
   /opt/pdweb/etc/oauth_eas.conf
   ```

   Windows

   ```
   oauth_pop_trigger = C:\Program Files\Tivoli\PDWebRTE\bin\
   libamwoautheas.dll & C:\Program Files\Tivoli\PDWeb\etc\
   oauth_eas.conf
   ```

   Pour plus d'informations, voir «Section [aznapi-external-authzn-services]», à la page 472.

2. Configurez les données de décision d'autorisation obligatoires.

   Le service EAS OAuth nécessite diverses données de la requête. Vous pouvez indiquer la requête en tant qu'éléments de requête HTTP dans la section `[azn-decision-info]`. Les entrées de configuration suivantes sont requises pour le bon fonctionnement du service EAS OAuth :

   ```
   [azn-decision-info]
   ##
   The following information will be provided to the authorization
   # framework for every authorization request. This information
   # is required by the OAuth EAS when validating an OAuth token.
   #
   HTTP_REQUEST_METHOD = method
   HTTP_REQUEST_SCHEME = scheme
   HTTP_REQUEST_URI = uri
   HTTP_HOST_HDR = header:host
   HTTP_CONTENT_TYPE_HDR = header:content-type
   HTTP_TRANSFER_ENCODING_HDR = header:transfer-encoding
   HTTP_AZN_HDR = header:authorization

   [aznapi-configuration]

   resource-manager-provided-adi = AMWS_pb_
   ```

   Remarque : Les données de requête requises sont identiques quel que soit l'environnement.

3. Créez les fichiers de réponses HTML requis.

   Pour plus d'informations sur les paramètres de configuration du fichier de réponses, voir «Section [oauth-eas]», à la page 475.

4. Configurez les données spécifiques au service EAS supplémentaires.

   Le service EAS OAuth a besoin de données de configuration particulières pour fonctionner correctement. Ces données figurent dans la section `[oauth-eas]` du fichier de configuration. Le nom de ce dernier est indiqué en tant qu'argument dans la section `[aznapi-external-authzn-services]` de l'entrée de configuration `amwoautheas` du fichier de configuration WebSEAL.

   a. Ouvrez le fichier de configuration du service EAS OAuth spécifié à l'étape 1b à l'aide d'un éditeur de fichier.

   b. Indiquez les valeurs des variables *default-fed-id*, *default-mode*, *realm-name*, *custom response files* et *server url*. Voir un exemple de section `[oauth-eas]` et des détails de configuration dans la rubrique «Exemple de données de configuration du service EAS», à la page 475.

   Remarque : `cluster-name` est l'une des entrées de configuration requises dans la section `[oauth-eas]`. Elle indique le nom du cluster Tivoli Federated Identity Manager hébergeant le service OAuth. Vous devez configurer une section `[tfim-cluster:<cluster>]` correspondante pour indiquer le cluster en question.

   Voir «Section [oauth-eas]», à la page 475 pour plus d'informations sur les entrées de configuration requises.

5. Définissez la règle de démarrage du service EAS OAuth. Par exemple :

   ```
   #pdadmin -a sec_master
   Enter password: passw0rd
   pop create test-pop
   pop modify test-pop set attribute eas-trigger oauth_pop_trigger
   pop attach /WebSEAL/webseal.example.com-default/oauth test-pop
   server replicate
   quit
   ```

Remarque : Si des données de paramètre OAuth sont envoyées aux clients OAuth dans le mécanisme d'en-tête de l'autorisation, définissez la balise `-b ignore` pour la jonction associée au protocole POP OAuth. Pour plus d'informations, voir IBM WebSEAL Administration Guide.

6. Redémarrez WebSEAL pour appliquer les modifications de la configuration.

   UNIX ou Linux

   ```
   /opt/pdweb/bin/pdweb_start restart
   ```

   Windows

   Utilisez le panneau de configuration des services :

   Démarrer > **Paramètres** > **Panneau de configuration** > **Outils d'administration** > **Services**

Configuration du service EAS OAuth WebSEAL à l'aide de l'outil tfimcfg

Utilisez l'outil tfimcfg pour facilement configurer le service EAS OAuth WebSEAL.

Avant de commencer

IBM® Tivoli® Access Manager for e-business version 6.1.1 ou ultérieure doit être installé. Si la version 6.1.1 est installée, le groupe de correctifs 5 ou ultérieur doit être appliqué.

Procédure

1. Exécutez l'outil `tfimcfg`.

   Utilisez les attributs `tfimcfg` suivants :

   - Pour un serveur WebSEAL :

     ```
     java -jar tfimcfg.jar -action tamconfig -cfgfile
     WebSEAL_filename
     ```

   - Pour un serveur Web Gateway Appliance :

     ```
     java -jar tfimcfg.jar -action wgaconfig -cfgurl
     Web_Gateway_Appliance_URL
     ```

   Pour obtenir des informations sur les paramètres de l'outil dans le centre de documentation Federated Identity Manager, voir la référence `tfimcfg`.

   Si vous utilisez l'environnement d'exécution Java fourni avec WebSphere Application Server, version 8.0 ou ultérieure, une erreur peut se produire lors de l'utilisation de `-action tamconfig`. Voir la rubrique "Incidents connus et solutions" du centre de documentation Federated Identity Manager pour les paramètres `tfimcfg` requis.

2. Fournissez les informations OAuth PEP spécifiques suivantes :

   a. Pour l'invite **Fédération à configurer**, sélectionnez **OAuth policy enforcement point**.

   b. Indiquez les paramètres OAuth PEP suivants :

      - Bibliothèque de service EAS OAuth
        - Linux :

          ```
          /opt/pdwebrte/lib/libamwoautheas.so
          ```

        - AIX :

          ```
          /opt/pdwebrte/lib/libamwoautheas.a
          ```

        - Windows :

          ```
          C:\Program Files\Tivoli\PDWebRTE\bin\amwoautheas.dll
          ```

- Page de réponse '400 Bad Request'
- Page de réponse '401 Unauthorized'
- Page de réponse '502 Bad Gateway'
- Fédération OAuth par défaut
- Mode OAuth par défaut

c. Indiquez les paramètres de service suivants :

   Indiquez les paramètres de service suivants :

   - ID utilisateur du client ITFIM Security Token Service facultatif
   - Mot de passe du client ITFIM Security Token Service facultatif

d. L'outil tfimcfg tente de se connecter au serveur hébergeant le service OAuth. Si la connexion n'est pas valide, vous êtes invité à entrer à nouveau le paramètre.

3. Associez oauth-pop à la ressource que OAuth PEP doit protéger. Par exemple, si MyJct est la ressource à protéger, exécutez la commande suivante :

```
#pdadmin -a sec_master
Enter password: passw0rd
pop attach /WebSEAL/localhost-default/MyJct oauth-pop
server replicate
quit
```

Chapitre 29. Référence OAuth

Cette rubrique contient des références sur les points d'application et leurs propriétés personnalisées, les sections EAS et les modèles de pages HTML pour les deux. La présente rubrique s'applique à OAuth 1.0 et OAuth 2.0.

Interface de service STS OAuth pour les points d'application d'autorisation

Utilisez l'interface WS-Trust pour contacter directement une chaîne d'accréditation Security Token Service (STS) OAuth dans Tivoli Federated Identity Manager pour valider une demande pour une ressource protégée OAuth. Un point d'application OAuth intercepte les demandes de ressources protégées OAuth. Le point d'application OAuth valide également la demande auprès de Tivoli Federated Identity Manager, et elle est valide, la transmet. Si la demande n'est pas valide, le point d'application refuse l'accès à la ressource protégée.

Présentation du service STS OAuth

Vous pouvez développer votre propre point d'application de règles personnalisé pour qu'il fonctionne avec la chaîne d'accréditation du service de jeton de sécurité (STS, Security Token Service) via l'interface STS. Voici quelques exemples de points d'application des règles personnalisés existants : WebSphere Servlet Filter, Trust Association Interceptor (TAI) et un proxy inverse tel que WebSEAL. Tivoli Federated Identity Manager prenant en charge à la fois les fédérations OAuth 1.0 et OAuth 2.0, vous pouvez développer des points d'application des règles de sorte qu'ils fonctionnent avec l'un ou l'autre type de fédération OAuth. Le diagramme suivant illustre les relations entre la chaîne d'accréditation STS OAuth et d'autres composants OAuth.

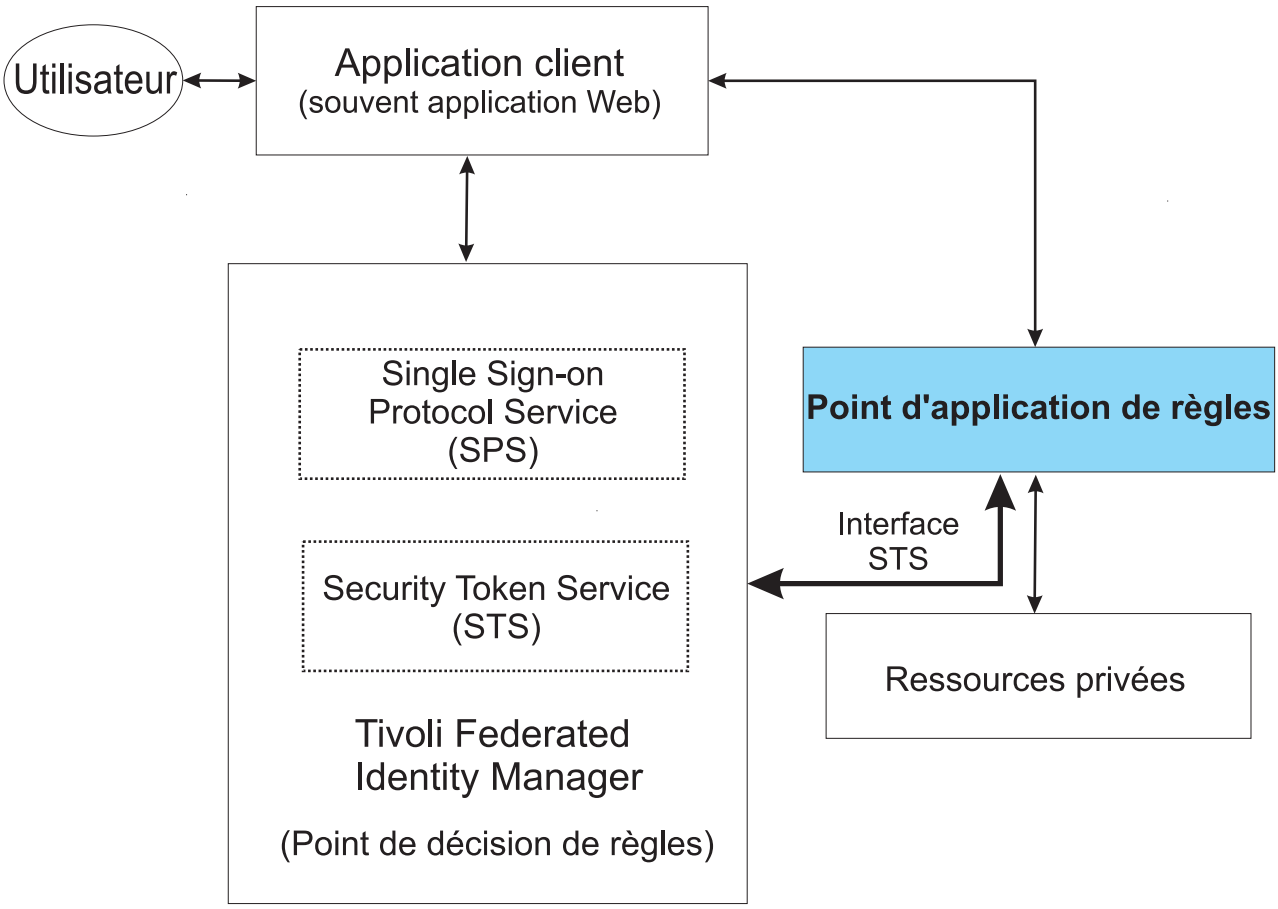


Figure 43. Flux de travaux de chaîne d'accréditation STS OAuth

Cette section décrit la procédure effectuée par un point d'application OAuth pour convertir une demande HTTP de ressource protégée OAuth en message WS-Trust.

Cette conversion permet au service STS Tivoli Federated Identity Manager de valider la demande. Elle décrit également les réponses possibles que peut recevoir un point application du service STS et la manière de les traiter.

L'interface et la structure des messages sont identiques pour OAuth 1.0 et OAuth 2.0. Toutefois, ce document fournit des exemples distinct de chaque cas afin de mettre en évidence les différentes exigences.

Les informations suivantes sur le point de décision de règles dans Tivoli Federated Identity Manager doivent être fournies au point d'application :

- L'adresse URL absolue du noeud final de service d'accréditation Tivoli Federated Identity Manager. (Par exemple : `http://idp.tfim622.com:9080/TrustServer/SecurityTokenService`)
- Le nom d'utilisateur et le mot de passe d'authentification de base pour le service d'accréditation Tivoli Federated Identity Manager (si nécessaire).
- L'ID fournisseur de la fédération Tivoli Federated Identity Manager à laquelle appartient le client, qui est utilisé comme adresse AppliesTo pour les demandes WS-Trust. Facultativement, le point d'application accepte un ID fournisseur du client OAuth en tant que paramètre de demande pour servir simultanément plusieurs fédérations.

Demande de décision d'autorisation (OAuth 1.0)

Configuration

Pour les demandes OAuth 1.0, le point d'application doit également connaître le préfixe d'adresse de l'émetteur Tivoli Federated Identity Manager OAuth 1.0 (`urn:ibm:ITFIM:oauth:consumer:`).

Demande HTTP

Lorsqu'un client OAuth 1.0 extrait une ressource protégée à l'aide de son jeton d'accès, il crée une demande similaire aux exemples suivants. Ces trois exemples représentent logiquement la même demande.

Exemple OAuth 1.0 1 (en-tête d'autorisation)

```
POST /fimivt/oauth/sfprotected.jsp?username=steve HTTP/1.1
  Host: idp.tfim622.com:9443
  Content-Type: application/x-www-form-urlencoded
  Authorization: OAuth oauth_consumer_key="YvMhsjmtEEi2gjv8Tqsl",
       oauth_token="YPxa78JggdW7hvcFRJph",
       oauth_signature_method="HMAC-SHA1",
       oauth_timestamp="1302828764",
       oauth_nonce="xWlY1PbsxjpiSZ4lVGVf",
       oauth_signature="Jpo6apiLE9hVSa8GqBSHUjFt7lg="
```

Exemple OAuth 1.0 2 (corps du post)

```
POST /fimivt/oauth/sfprotected.jsp?username=steve HTTP/1.1
  Host: idp.tfim622.com:9443
  Content-Type: application/x-www-form-urlencoded

    oauth_consumer_key=YvMhsjmtEEi2gjv8Tqsl&oauth_token=YPxa78JggdW7hvcFRJph&
    oauth_signature_method=HMAC-SHA1&oauth_timestamp=1302828764&
    oauth_nonce=xWlY1PbsxjpiSZ4lVGVf&oauth_signature=Jpo6apiLE9hVSa8GqBSHUjFt7lg%3D
```

Exemple OAuth 1.0 3 (chaîne de requête)

```
POST /fimivt/oauth/sfprotected.jsp?username=steve&
    oauth_consumer_key=YvMhsjmtEEi2gjv8Tqsl&oauth_token=YPxa78JggdW7hvcFRJph&
    oauth_signature_method=HMAC-SHA1&oauth_timestamp=1302828764&
    oauth_nonce=xWlY1PbsxjpiSZ4lVGVf&oauth_signature=Jpo6apiLE9hVSa8GqBSHUjFt7lg%3D HTTP/1.1
  Host: idp.tfim622.com:9443
  Content-Type: application/x-www-form-urlencoded
```

Demande de décision d'autorisation

Le point d'application OAuth 1.0 réalise les actions suivantes :

- Convertir la demande HTTP en message SOAP WS-Trust.
- Envoyer le message SOAP WS-Trust au service STS Tivoli Federated Identity Manager pour validation de la demande.

Le demande HTTP est convertie en message SOAP WS-Trust suivant :

Demande de validation de jeton OAuth 1.0 (jeton de sécurité de demande)

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP-ENV:Body>
        <wst:RequestSecurityToken xmlns:wst="http://schemas.xmlsoap.org/ws
        /2005/02/trust">
            <wst:RequestType xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
                http://schemas.xmlsoap.org/ws/2005/02/trust/Validate
            </wst:RequestType>
            <wst:Issuer xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
                <wsa:Address xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08
                /addressing">
                    urn:ibm:ITFIM:oauth:consumer:YvMhsjmtEEi2gjv8Tqsl
                </wsa:Address>
            </wst:Issuer>
            <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
                <wsa:EndpointReference xmlns:wsa="http://schemas.xmlsoap.org
                /ws/2004/08/addressing">
```

```
                    <wsa:Address>https://idp.tfim622.com:9443/sps/oauth10fed
                    /oauth10</wsa:Address>
                </wsa:EndpointReference>
            </wsp:AppliesTo>
            <wst:Base xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
                <stsuuser:STSUniversalUser xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser">
                    <stsuuser:Principal/>
                    <stsuuser:AttributeList/>
                    <stsuuser:ContextAttributes>
                        <stsuuser:Attribute name="oauth_token"
                          type="urn:ibm:names:ITFIM:oauth:param">
                            <stsuuser:Value>YPxa78JggdW7hvcFRJph</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="port"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>9443</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="method"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>POST</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="username"
                          type="urn:ibm:names:ITFIM:oauth:query:param">
                            <stsuuser:Value>steve</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="path"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>/fimivt/oauth/sfprotected.jsp
                             </stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="scheme"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>https</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="oauth_nonce"
                          type="urn:ibm:names:ITFIM:oauth:param">
                            <stsuuser:Value>xWlY1PbsxjpiSZ4lVGVf</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="host"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>idp.tfim622.com</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="oauth_timestamp"
                          type="urn:ibm:names:ITFIM:oauth:param">
                            <stsuuser:Value>1302828764</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="oauth_consumer_key"
                          type="urn:ibm:names:ITFIM:oauth:param">
                            <stsuuser:Value>YvMhsjmtEEi2gjv8Tqsl</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="oauth_signature"
                          type="urn:ibm:names:ITFIM:oauth:param">
                            <stsuuser:Value>Jpo6apiLE9hVSa8GqBSHUjFt7lg=
                            </stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="oauth_signature_method"
                        type="urn:ibm:names:ITFIM:oauth:param">
                            <stsuuser:Value>HMAC-SHA1</stsuuser:Value>
                        </stsuuser:Attribute>
                    </stsuuser:ContextAttributes>
                </stsuuser:STSUniversalUser>
            </wst:Base>
        </wst:RequestSecurityToken>
    </soapenv:Body>
</soapenv:Envelope>
```

Les attributs suivants sont définis par la spécification WS-Trust. Il sont utilisés par Tivoli Federated Identity Manager pour identifier la fédération associée à cette demande et le client OAuth 1.0.

L'élément d'adresse de l'émetteur (mis en évidence en *italiques*) doit être défini sur le préfixe d'adresse de l'émetteur OAuth 1.0 Tivoli Federated Identity Manager OAuth (`urn:ibm:ITFIM:oauth:consumer:`) auquel est ajouté à la fin la clé de consommateur.

L'élément d'adresse AppliesTo (mis en évidence par un soulignement) doit être défini sur l'ID fournisseur de la fédération OAuth 1.0 sur Tivoli Federated Identity Manager. Cet élément se trouve dans la page des propriétés de la fédération.

Les attributs suivants sont définis par le protocole OAuth 1.0. Les attributs qui ne sont pas marqués comme facultatifs sont obligatoires dans le message WS-Trust qui est envoyé à Tivoli Federated Identity Manager.

Ces attributs doivent être ajoutés à la section **ContextAttributes** de **STSUniversalUser** dans le jeton de sécurité et doivent avoir le type `urn:ibm:names:ITFIM:oauth:param`. Si l'un des paramètres obligatoires est manquant dans la demande du client OAuth 1.0, le point d'application ne valide pas la requête auprès de Tivoli Federated Identity Manager. Il peut renvoyer instantanément un code d'état HTTP 400 Bad Request, et son corps peut également inclure une description de l'erreur.

- **consumer_key**
- **nonce**
- **realm** (facultatif)
- **signature**
- **signature_method**
- **timestamp**
- **token** (facultatif uniquement si la validation OAuth à deux jambes est activée)
- **version** (facultatif)

Les attributs suivants sont définis par le protocole OAuth 2.0. Ces attributs sont obligatoires dans le message WS-Trust envoyé à Tivoli Federated Identity Manager. Ces attributs sont également utilisés pour recréer l'identificateur URI de chaîne de base de signature originale de la demande.

Ces attributs peuvent être ajoutés à la section **ContextAttributes** de **STSUniversalUser** dans le jeton de sécurité de la demande WS-Trust et doivent avoir le type `urn:ibm:names:ITFIM:oauth:request`.

- **host** : en-tête de l'hôte extrait de la demande
- **method** : méthode HTTP de la demande (GET/POST)
- **path** : chemin demandé
- **port** : numéro de port sur l'hôte (uniquement si la demande est reçue sur un port HTTP/HTTPS non standard.)
- **scheme** : (HTTP/HTTPS)

Tous les paramètres supplémentaires éventuels que le point d'application OAuth 1.0 trouve dans la demande doivent être ajoutés à la section **ContextAttributes** section de **STSUniversalUser** dans le jeton de sécurité de demande WS-Trust. Des paramètres supplémentaires peuvent être une requête, ou des paramètres de corps de post qui n'appartiennent pas à OAuth 1.0. La valeur du type est déterminée par le tableau suivant.

Emplacement du paramètre HTTP	Valeur du type d'attribut
Paramètres de chaîne de requête d'URL	`urn:ibm:names:ITFIM:oauth:query:param`
Paramètres de corps de requête HTTP	`urn:ibm:names:ITFIM:oauth:body:param`

Les paramètres de corps de post doivent être inclus uniquement si les conditions suivantes sont remplies :

- Le corps d'entité ne comporte qu'une seule partie.
- Le corps d'entité suit les exigences de codage du type de contenu «application/x-www-form-urlencoded» définies par [W3C.REC-html40-19980424].
- L'en-tête d'entité de la demande HTTP inclut le champ d'en-tête «Content-Type» défini sur «application/x-www-form-urlencoded»

Demande de décision d'autorisation (OAuth 2.0)

Configuration

Pour les demandes OAuth 2.0, le point d'application doit également connaître le préfixe d'adresse de l'émetteur OAuth 2.0 Tivoli Federated Identity Manager (`urn:ibm:ITFIM:oauth20:token:`).

Demande HTTP

Lorsqu'un client OAuth 2.0 extrait une ressource protégée à l'aide de son jeton d'accès, il crée une demande similaire à l'un des exemples suivants. Ces trois exemples représentent logiquement la même demande. La seule différence concerne le mécanisme de transmission (en-tête HTTP, chaîne de requête, corps du post) utilisé pour envoyer le jeton d'accès bearer OAuth 2.0 :

Exemple OAuth 2.0 1 (jeton d'accès dans l'en-tête de l'autorisation)

```
POST /fimivt/oauth/sfprotected.jsp HTTP/1.1
  Host: idp.tfim622.com:9443
  Authorization: Bearer YPxa78JggdW7hvcFRJph
  Content-Type: application/x-www-form-urlencoded

username=steve
```

Exemple OAuth 2.0 2 (jeton d'accès dans le corps du post)

```
POST /fimivt/oauth/sfprotected.jsp HTTP/1.1
  Host: idp.tfim622.com:9443
  Content-Type: application/x-www-form-urlencoded

username=steve&access_token=YPxa78JggdW7hvcFRJph
```

Exemple OAuth 2.0 3 (jeton d'accès dans la chaîne de requête)

```
POST /fimivt/oauth/sfprotected.jsp?access_token=YPxa78JggdW7hvcFRJph HTTP/1.1
  Host: idp.tfim622.com:9443
  Content-Type: application/x-www-form-urlencoded

username=steve
```

Demande de décision d'autorisation

Le point d'application OAuth 2.0 réalise les actions suivantes :

- Convertir les demandes HTTP en un message SOAP WS-Trust.
- Envoyer le message SOAP WS-Trust au service STS Tivoli Federated Identity Manager pour la validation de la demande.

Le demande HTTP est convertie en message SOAP WS-Trust suivant :

Demande de validation de jeton OAuth 2.0 (jeton de sécurité de demande)

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP-ENV:Body>
        <wst:RequestSecurityToken xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
            <wst:RequestType xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
                http://schemas.xmlsoap.org/ws/2005/02/trust/Validate
```

```
            </wst:RequestType>
            <wst:Issuer xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
                <wsa:Address xmlns:wsa="http://schemas.xmlsoap.org/ws/2004
                /08/addressing">
                    urn:ibm:ITFIM:oauth20:token:bearer
                </wsa:Address>
            </wst:Issuer>
            <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004
            /09/policy">
                <wsa:EndpointReference xmlns:wsa="http://schemas.xmlsoap
                .org/ws/2004/08/addressing">
                    <wsa:Address>https://idp.tfim622.com:9443/sps/oauth20fed/oauth20</wsa:Address>
                </wsa:EndpointReference>
            </wsp:AppliesTo>
            <wst:Base xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
                <stsuuser:STSUniversalUser xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser">
                    <stsuuser:Principal/>
                    <stsuuser:AttributeList/>
                    <stsuuser:ContextAttributes>
                        <stsuuser:Attribute name="access_token"
                          type="urn:ibm:names:ITFIM:oauth:param">
                            <stsuuser:Value>YPxa78JggdW7hvcFRJph</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="username"
                          type="urn:ibm:names:ITFIM:oauth:body:param">
                            <stsuuser:Value>steve</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="port"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>9443</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="method"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>POST</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="path"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>/fimivt/oauth/sfprotected.jsp</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="scheme"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>https</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="host"
                          type="urn:ibm:names:ITFIM:oauth:request">
                            <stsuuser:Value>idp.tfim622.com</stsuuser:Value>
                        </stsuuser:Attribute>
                    </stsuuser:ContextAttributes>
                </stsuuser:STSUniversalUser>
            </wst:Base>
        </wst:RequestSecurityToken>
    </soapenv:Body>
</soapenv:Envelope>
```

Les attributs suivants sont définis par la spécification WS-Trust. Ils sont utilisés par Tivoli Federated Identity Manager pour identifier la fédération associée à cette demande et le type de jeton d'accès OAuth 2.0 utilisé.

- L'élément d'adresse de l'émetteur (mis en évidence en **gras**) doit être défini sur le préfixe d'adresse de l'émetteur OAuth 2.0 Tivoli Federated Identity Manager (`urn:ibm:ITFIM:oauth20:token:`). Le type de jeton doit être ajouté à la fin, séparé par un caractère deux-points. Pour le moment, le seul type de jeton pris en charge est *bearer*, ce qui signifie que l'adresse de l'émetteur doit être définie sur `urn:ibm:ITFIM:oauth20:token:bearer`.
- L'élément d'adresse AppliesTo (mis en évidence en *italique*) doit être défini sur l'ID fournisseur de la fédération OAuth sur Tivoli Federated Identity Manager. Cet élément figure dans la page des propriétés de la fédération.

L'attribut **access_token** de type `urn:ibm:names:ITFIM:oauth:param` est obligatoire dans le message WS-Trust envoyé à Tivoli Federated Identity Manager. Il doit être ajouté à la section **ContextAttributes** de **STSUniversalUser** dans le jeton de sécurité de demande WS-Trust.

Si l'attribut **access_token** est manquant dans la demande provenant du client OAuth 2.0, le point d'application ne valide pas la demande auprès de Tivoli Federated Identity Manager. Il peut renvoyer instantanément un code d'état HTTP 400 Bad Request et son corps peut inclure en option un description de l'erreur.

Remarque : Si le jeton d'accès est inclus dans l'en-tête de l'autorisation dans le format `Authorization: Bearer <token>`, il peut être toutefois ajouté à la section **ContextAttributes** du STSUU. Le format à utiliser est identique à celui qui serait employé pour envoyer le jeton d'accès via une chaîne de requête ou un corps de post.

Les attributs suivants ne sont *pas* obligatoires dans le message WS-Trust envoyés à Tivoli Federated Identity Manager pour OAuth 2.0. Toutefois, ils peuvent être utiles pour une règle de mappage personnalisée exécutée par Tivoli Federated Identity Manager.

Les attributs suivants doivent être ajoutés à la section **ContextAttributes** de **STSUniversalUser** dans le jeton de sécurité de demande WS-Trust et doivent avoir le type `urn:ibm:names:ITFIM:oauth:request`.

- **method** : méthode HTTP de la demande (GET/POST)
- **scheme** : (http/https)
- **host** : en-tête de l'hôte extrait de la demande
- **port** - numéro de port sur l'hôte (uniquement s'il s'agit d'un port non standard ; par exemple, différent de 80 si la méthode est HTTP ou différent de 443 si la méthode est HTTPS)
- **path** : chemin demandé

Tous les autres paramètres trouvés dans la demande par le point d'application 2.0, tels que les paramètres de la requête ou du corps du post n'appartenant pas à OAuth 2.0, doivent être ajoutés à la section **Context Attribute** de **STSUniversalUser** dans le jeton de sécurité de requête WS-Trust. La valeur de type est déterminée par le tableau suivant.

Dans une demande OAuth 1.0, des paramètres de demande supplémentaires doivent être ajoutés à **STSUniversalUser** pour calculer la signature de demande correcte. Dans les demandes OAuth 2.0, ces paramètres ne sont PAS obligatoires. Toutefois, ils peuvent être utiles pour une règle de mappage personnalisée exécutée par Tivoli Federated Identity Manager, et doivent donc être ajoutés dans ce cas.

Emplacement du paramètre HTTP	Valeur du type d'attribut
Paramètres de chaîne de requête d'URL	`urn:ibm:names:ITFIM:oauth:query:param`
Paramètres de corps de requête HTTP	`urn:ibm:names:ITFIM:oauth:body:param`

Les paramètres de corps de post doivent être inclus uniquement si les conditions suivantes sont remplies :

- Le corps d'entité ne comporte qu'une seule partie.
- Le corps d'entité respecte les exigences de codage du type de contenu «application/x-www-form-urlencoded» définies par [W3C.REC-html40-19980424].
- L'en-tête d'entité de la demande HTTP inclut le champ d'en-tête «Content-Type» défini sur«application/x-www-form-urlencoded».

Réponse de décision d'autorisation (OAuth 1.0 et OAuth 2.0)

La réponse du message SOAP provenant de Tivoli Federated Identity Manager (indépendamment de la version d'OAuth) reflète tous les attributs de contexte envoyés dans la demande d'origine et quelques attributs de contexte de réponse supplémentaires.

Réponse de validation de jeton (RSTR) OAuth

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP-ENV:Body>
        <wst:RequestSecurityTokenResponse wsu:
        Id="uuid56a54e7c-012f-1207-9133-c24cad886d75"
        xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401
        -wss-wssecurity-utility-1.0.xsd">
            <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004
            /08/addressing"
            xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
                <wsa:EndpointReference>
                    <wsa:Address>https://idp.tfim622.com:9443/sps/oauth10fed
                    /oauth10</wsa:Address>
                </wsa:EndpointReference>
            </wsp:AppliesTo>
            <wst:RequestedSecurityToken>
                <stsuuser:STSUniversalUser xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser">
                    <stsuuser:Principal/>
                    <stsuuser:AttributeList/>
                    <stsuuser:ContextAttributes>
                        <stsuuser:Attribute name="authorized"
                        type="urn:ibm:names:ITFIM:oauth:response:decision">
                            <stsuuser:Value>TRUE</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="expires" type="urn:ibm
                        :names:ITFIM:oauth:response:decision">
                            <stsuuser:Value>2011-04-22T00:52:18Z</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="scope" type="urn:ibm
                        :names:ITFIM:oauth:response:attribute">
                                <stsuuser:Value>email</stsuuser:Value>
                                <stsuuser:Value>first</stsuuser:Value>
                                <stsuuser:Value>last</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="username" type="urn:ibm
                        :names:ITFIM:oauth:response:attribute">
                            <stsuuser:Value>wasadmin</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="username_is_self"
                        type="urn:ibm:names:ITFIM:oauth:response:attribute">
                            <stsuuser:Value>FALSE</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="oauth_token" type="urn:ibm
                        :names:ITFIM:oauth:response:attribute">
                            <stsuuser:Value>YPxa78JggdW7hvcFRJph</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="recovered_state" type="urn:ibm
                        :names:ITFIM:oauth:response:attribute">
                            <stsuuser:Value>State storage time was:
                            2011-04-15T00:52:18Z</stsuuser:Value>
                        </stsuuser:Attribute>
                        <stsuuser:Attribute name="state_id" type="urn:ibm
                        :names:ITFIM:oauth:state">
                            <stsuuser:Value>2cJsZ3QhXV5rDVZHNePp</stsuuser:Value>
                        </stsuuser:Attribute>
                    </stsuuser:ContextAttributes>
                    <stsuuser:AdditionalAttributeStatement id=""/>
                </stsuuser:STSUniversalUser>
            </wst:RequestedSecurityToken>
            <wst:Status>
                <wst:Code>http://schemas.xmlsoap.org/ws/2005/02/trust/status
                /valid</wst:Code>
```

```
            </wst:Status>
        </wst:RequestSecurityTokenResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Les attributs de contexte suivants renvoyés au point d'application par Tivoli Federated Identity Manager s'appliquent à la décision d'autorisation. Ils possèdent le type d'attribut `urn:ibm:names:ITFIM:oauth:response:decision` mis en évidence en *italique* dans l'exemple RSTR précédent. Il incombe au point d'application de décider si ces attributs doivent être propagés en aval jusqu'à la ressource protégée OAuth.

Ces attributs sont principalement destinés à être utilisés par le point d'application lui-même pour déterminer le statut d'autorisation.

Attributs de contexte	Description
autorisé	La valeur est définie sur TRUE si la demande OAuth est valide ou autorisée. Sinon, la valeur est FALSE.
expire	L'heure en temps universel que le jeton d'accès a utilisé dans la demande n'est plus valide. Cet attribut n'est pas présent dans la validation OAuth 1.0 à deux jambes, car le flux n'utilise pas de jeton d'accès.

Les attributs de contexte suivants renvoyés au point d'application par Tivoli Federated Identity Manager doivent être propagés en du point d'application à la ressource protégée OAuth. Ils peuvent être ajoutés à la demande HTTP d'origine de toute manière jugée appropriée par le point d'application et la ressource protégée. De cette manière, la ressource protégée peut les extraire (par exemple, en tant qu'en-têtes HTTP supplémentaires).

Ces attributs de contexte ont le type d'attribut `urn:ibm:names:ITFIM:oauth:response:attribute` (mis en évidence en **gras** dans l'exemple RSTR précédent).

Les règles de mappage personnalisées qui sont exécutées après la chaîne d'accréditation OAuth peuvent également ajouter des attributs de ce type. Par conséquent, tous les attributs de ce type doivent être propagés en aval jusqu'à la ressource protégée demandée.

Attributs de contexte	Description
access_token (OAuth 2.0)	Jeton d'accès OAuth utilisé dans la demande de ressource protégée.
client_type (OAuth 2.0)	Type de client pour lequel ce jeton a été émis, peut être public ou confidentiel. Les clients publics sont des clients qui ne possèdent pas de droits d'accès client et ne peuvent par conséquent pas être authentifiés auprès du serveur d'autorisation.
oauth_token_client_id (OAuth 2.0)	Identificateur unique du client pour lequel le jeton d'accès courant a été émis. Ce paramètre n'est pas renvoyé pour les demandes OAuth 1.0, car la clé de consommateur est envoyée dans la demande initiale. Par conséquent, il reste dans le STSUU avec le nom`consumer_key` et le type `urn:ibm:names:ITFIM:oauth:request`.
oauth_token (OAuth 1.0)	Jeton d'accès OAuth utilisé dans la demande de ressource protégée. Ce attribut n'est pas présent pour la validation OAuth 1.0 à deux jambes, car le flux n'utilise pas de jeton d'accès.

Attributs de contexte	Description
scope	Liste de chaînes qui représente la portée de la ressource autorisée par l'utilisateur à l'étape d'autorisation du propriétaire de la ressource OAuth. La ressource protégéeOAuth peut utiliser cet attribut pour déterminer les ressources à retourner dans la réponse. Cet attribut est présent uniquement pour les flux OAuth qui incluent une étape d'autorisation.
username	Nom de l'utilisateur qui a autorisé le jeton OAuth à accéder à leurs ressources protégées en son nom. Avec les flux OAuth qui n'impliquent pas un propriétaire de ressource distinct, cette valeur est l'identificateur du client.

Des attributs supplémentaires ayant le type `urn:ibm:names:ITFIM:oauth:response:attribute` sont parfois ajoutés par une règle de mappage personnalisée, ce qui est le cas avec **recovered_state** et **username_is_self** dans l'exemple.

L'attribut de contexte**state_id** renvoyé au point d'application par Tivoli Federated Identity Manager est utilisée par une règle de mappage personnalisées qui est exécutée après la chaîne d'accréditation OAuth. Il a le type d'attribut `urn:ibm:names:ITFIM:oauth:state` (mis en évidence par un soulignement) et peut être ignoré par le point d'application.

L'attribut**state_id** est un identificateur uniquement pour le jeton OAuth courant utilisé pour stocker les informations d'état.

Si l'attribut **state_id** est requis par la ressources protégée OAuth, une règle de mappage personnalisée peut être implémentée pour effectuer une copie de cet attribut. Ce type peut être changé de la règle de mappage personnalisée en `urn:ibm:names:ITFIM:oauth:response:attribute` pour garantir sa propagation en aval jusqu'à la ressource.

Réponses aux erreurs

Un point d'application peut soumettre les demandes OAuth à la quantité de validation de son choix. Toute validation effectuée est répétée par Tivoli Federated Identity Manager. L'exécution de cette validation avant l'envoi d'une demande d'autorisation à Tivoli Federated Identity Manager peut améliorer la performance. La validation suivante doit être effectuée par le point d'application avant l'envoi d'une demande à Tivoli Federated Identity Manager.

- Vérification de la présence de certaines données OAuth. Si ces données sont absentes, renvoie un code d'état HTTP 401 Unauthorized.
- Vérification de la présence de tous les paramètres OAuth obligatoires. Si l'un de ces paramètres n'est pas présent dans la demande, renvoie un code d'état HTTP 400 Bad Request.
- Vérification de l'unicité de tous les paramètres OAuth obligatoires dans la demande. Les paramètres obligatoires doivent également être présents uniquement dans le composant unique de la demande ; par exemple, la chaîne de requête ou l'en-tête de l'autorisation. En cas d'échec de la validation, renvoie un code d'état HTTP 400 Bad Request.

Le point d'application doit renvoyer un code d'état HTTP 401 Unauthorized au client OAuth si les scénarios suivants se produisent :

- Le point d'application envoie une demande d'autorisation à Tivoli Federated Identity Manager.
- Le point d'application reçoit un message SOAP avec un attribut de contexte autorisé de valeur FALSE.

Le point d'application doit renvoyer un code d'état HTTP 503 Service Unavailable au clientOAuth si les scénarios suivants se produisent :

- Tivoli Federated Identity Manager rencontre une erreur.
- Tivoli Federated Identity Manager ne renvoie pas un message SOAP construit ou le message SOAP ne contient pas d'attribut de contexte autorisé.

Le point d'application peut également facultativement un en-tête WWW-Authenticate HTTP pour indiquer la prise en charge de la validation OAuth.

Organigramme

Le diagramme suivant représente le flux de travaux prévu d'un point d'application de l'autorisation OAuth.

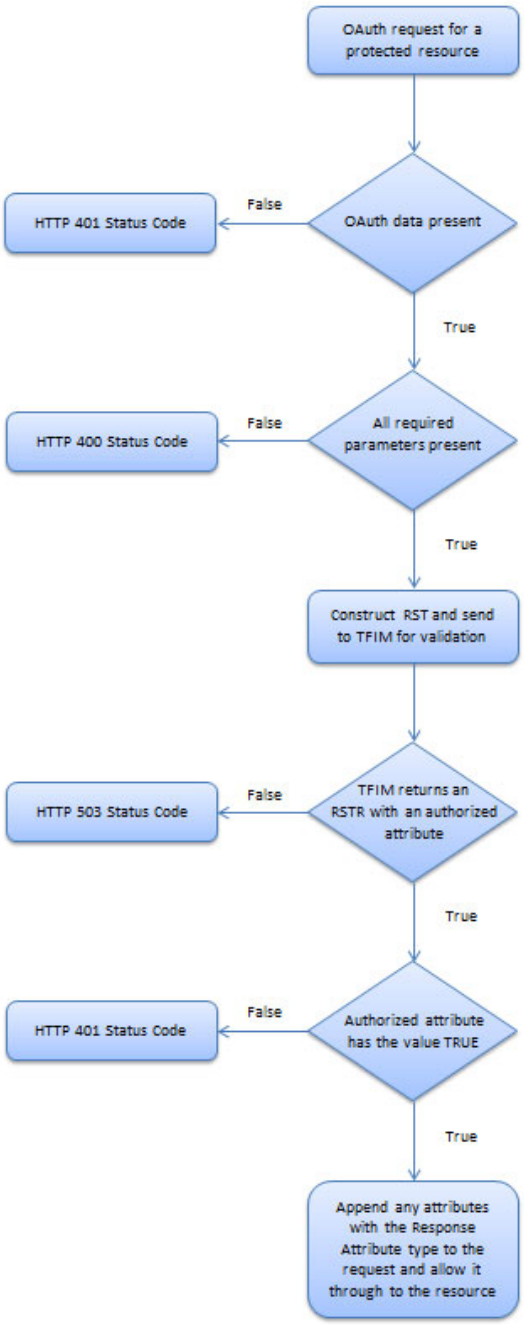


Figure 44. Flux de travaux du point d'application de l'autorisation OAuth

Propriétés personnalisées de l'intercepteur de relations de confiance et du filtre de servlet OAuth

Vous devez personnaliser la propriété du composant WebSphere TAI (Trust Association Interceptor, intercepteur de relations de confiance ou SF (Servlet Filter, filtre de servlet) en tant que point d'application de votre fédération OAuth.

Les propriétés des points d'application sont utilisées pour appeler le service STS (Security Token Service) de Tivoli Federated Identity Manager pour la validation et l'autorisation.

Cette rubrique répertorie les propriétés de configuration des composants WebSphere TAI et SF à la fois pour les fédérations OAuth 1.0 et OAuth 2.0.

Tableau 118. Propriétés de l'intercepteur de relations de confiance et du filtre de servlet

Nom de la propriété	Description	Exemple

FederationIdRequestParameterName	Indique le nom du paramètre de demande. (**Facultative**)
	
	La valeur du paramètre de demande d'environnement d'exécution est utilisée comme adresse AppliesTo dans les appels au service STS. Elle doit correspondre à l'ID fournisseur de la fédération à laquelle appartient le client OAuth.
	
	La personnalisation du paramètre de demande peut être effectuée par l'intermédiaire de cette propriété. Vous pouvez modifier l'URL de la ressource protégée pour inclure un paramètre de chaîne de requête à l'aide des éléments suivants :
	• un nom correspondant à la valeur de cette propriété de configuration et
	• une valeur correspondant à l'ID fournisseur de la fédération à laquelle appartient le client OAuth.
	
	Cette propriété permet à un point d'application de servir les demandes de plusieurs fédérations simultanément.
	
	Si cette propriété n'est pas fournie, la valeur de la propriété **DefaultFederationId** est utilisée comme valeur d'ID fournisseur statique dans les appels au service STS.
		`FederationId`
		
		Exemple d'utilisation :
		`sfprotected.jsp?FederationId=https://server.oauth.com/FIM/MySocialNetwork/oauth20`

DefaultFederationId	Définit la valeur par défaut de l'ID fournisseur de fédération utilisée pour les communications avec le service STS. (**Obligatoire**) Cette propriété est utilisée dans les cas suivants : • La propriété **FederationIdRequestParameterName** n'est pas fournie. • La demande entrante ne comporte aucun paramètre dont le nom correspond à la valeur de la propriété **FederationIdRequestParameterName**.	`https://server.oauth.com/FIM/MySocialNetwork/oauth20`

DefaultMode	Détermine le mode de validation d'une demande par rapport à OAuth 1.0 ou OAuth 2.0. (**Obligatoire**)
	
	Cette propriété permet de distinguer les différentes versions d'un protocole OAuth. Le type de jeton pris en charge pour un protocole OAuth 2.0 est également indiqué dans la valeur.
	
	Elle est utilisée dans les cas suivants :
	• La propriété **ModeParameterName** n'est pas fournie.
	• La demande entrante ne comporte pas de paramètre de demande dont le nom correspond à la valeur de la propriété **ModeParameterName**.
		Pour OAuth 1.0 :
		`OAuth10`
		
		Pour OAuth 2.0 :
		`OAuth20Bearer`

Tableau 118. Propriétés de l'intercepteur de relations de confiance et du filtre de servlet (suite)

Nom de la propriété	Description	Exemple

ModeParameterName	Indique le nom du paramètre de demande. (**Facultative**)
	
	Le nom du paramètre de demande peut être personnalisé pour comporter la valeur du mode. Vous pouvez modifier l'URL de la ressource protégée pour inclure un paramètre de chaîne de requête avec les éléments suivants :
	• un nom correspondant à la valeur de cette propriété de configuration et
	• une valeur correspondant à l'ID fournisseur de la fédération à laquelle appartient le client OAuth.
	
	Un point d'application de règles unique peut servir simultanément des fédérations OAuth 1.0 et OAuth 2.0 si les situations suivantes se produisent :
	• La propriété **ModeParameterName** a été utilisée avec la propriété **FederationIdRequestParameterName**.
	• Les clients OAuth envoient l'identificateur FederationId et les paramètres de mode dans la demande de ressource protégée.
	
	Si cette propriété n'est pas fournie, la valeur de la propriété **DefaultMode** est utilisée pour déterminer s'il est nécessaire de valider la demande entrante en tant que OAuth 1.0 ou OAuth 2.0.
		`mode`
		
		Exemple de syntaxe :
		Pour OAuth 1.0 :
		`sfprotected.jsp?mode=OAuth10`
		Pour OAuth 2.0 :
		`sfprotected.jsp?mode=OAuth20Bearer`

OAuthRealm	Indique le domaine inclus dans l'en-tête WWW-Authenticate qui est renvoyé à une demande qui ne contient pas de jeton OAuth autorisé. (**Obligatoire**)	`https://server.oauth.com/FIM/`
OAuthTokenCacheSize	Indique la taille maximale d'un cache. Ce cache est utilisé pour mapper des jetons bearer OAuth 2.0 à des résultats, tels que l'existence et la durée de validité des jetons, à partir de l'appel aux services de jeton de sécurité. (**Facultative**)	`2`
PointOfContact	Indique l'URL point de contact pour les clients du serveu. Le serveur IBM HTTP Server ou WebSEAL peut être utilisé en amont de WebSphere, auquel cas, l'URL n'aura pas le même aspect que dans l'exemple. (**Facultative**)	`https://server.oauth.com/FIM/`
STSEndpoint	Indique le noeud final WS-Trust 1.2 du service STS. (**Facultative**)	`https://server.oauth.com/FIM/`
STSUsername	Indique le nom d'utilisateur d'authentification de base pour la communication avec le service STS. (**Obligatoire** selon la sécurité du rôle `TrustClientInternalRole` dans l'environnement d'exécution `ITFIMRuntime`.)	`wasadmin`
STSPassword	Indique le mot de passe d'authentification de base pour la communication avec le service STS. (**Obligatoire** selon la sécurité du rôle `TrustClientInternalRole` dans l'environnement d'exécution `ITFIMRuntime`.)	`mot de passe`
STSSSLConfiguration	Indique un objet de configuration SSL WebSphere qui contient des clés adaptées à l'authentification SSL serveur, et si nécessaire client, de l'URL WS-Trust. (**Obligatoire** uniquement si l'URL HTTPS du noeud final STS est utilisée.)	`myssl cfg`
URIPrefix	Indique une chaîne qui est comparée avec le début de l'identificateur URI de la demande pour vérifier si l'intercepteur de relations de confiance ou le filtre de servlet doit protéger cette demande. Pour protéger TOUTES les ressources, utilisez /. (**Obligatoire**)	`/snoop`

Référence de section du service EAS OAuth

Cette rubrique contient la référence de section pour la configuration du service EAS OAuth.

- section [aznapi-external-authzn-services]
- section [azn-decision-info]
- section [aznapi-configuration]
- section [oauth]

Section [aznapi-external-authzn-services]

policy-trigger

Syntaxe

```
policy-trigger = plug-in_location [-weight N [& plug-in_parameters]]
```

Description

Définit le service d'autorisation externe.

Options

policy-trigger

Toute chaîne qui est reconnue comme un nom d'indicatif valide. Les noms d'indicatif de section ne peuvent pas contenir d'espace ni de caractères de crochet ouvrant ([) et de crochet fermant (]). Les caractères crochet sont utilisés pour définir de nouveaux noms de section. La section policy-trigger est sensible à la casse pour les définitions de jeux d'actions, car les actions elles-mêmes sont sensibles à la casse. Toutefois la section policy-trigger est insensible à la casse si l'attribut est un attribut POP (protected object policy, règles d'objet protégé).

plug-in_location

Chemin d'accès de la bibliothèque partagée ou du module DLL qui contient l'implémentation du plug-in pour le déclencheur de règles indiqué. Le chemin d'accès peut être spécifié sous une forme tronquée si le service d'autorisation externe doit être chargé par des clients sur plusieurs plateformes. Dans ce cas, le distributeur de service recherche le plug-in à l'aide de préfixes et de suffixes spécifiques à la plateforme correspondant à des noms DLL.

Le nom du plug-in EAS OAuth est **amwoautheas** et sa bibliothèque réside dans le répertoire `pdwebrte/lib`. Par exemple :

```
/opt/pdwebrte/lib/libamwoautheas.so
```

N

Le paramètre weight est un paramètre facultatif de valeur **size_t** non signée. Cette valeur indique le poids affecté à toute décision retournée par ce service d'autorisation externe dans l'ensemble du processus de décision.

plug-in_parameters

Facultativement, des informations d'initialisation supplémentaires peuvent être transmises au service d'autorisation externe sous la forme d'arguments. Les arguments doivent être précédés du caractère perluète "&". Le service d'autorisation examine le reste de la chaîne après le caractère perluète &,

fractionne la chaîne en jetons distincts séparés par des espaces et transmet ces jetons directement à l'interface d'initialisation du service d'administration, `azn_svc_initialize()`, dans le paramètre de matrice **argv**. Le nombre de chaînes contenues dans la matrice **argv** est indiqué par le paramètre de fonction **argc**.

Un paramètre unique est requis par le service EAS OAuth. Ce paramètre correspond au nom du fichier de configuration EAS OAuth, c'est-à-dire, le fichier qui contient la section **[oauth-eas]** et la section **[tfim-cluster:<cluster>]** correspondante.

Utilisation

Cette entrée de section est obligatoire lors de la configuration de l'authentification EAS OAuth.

Valeur par défaut

Aucune.

Exemple

L'exemple suivant est un déclencheur basé sur les opérations associé à un groupe d'actions défini par l'utilisateur Imprimante et aux actions rxT au sein de ce groupe. Pour spécifier le groupe d'actions principal, indiquez seulement : rxT. Le groupe d'actions principal peut être représenté par un nom de groupe d'actions vide ou la chaîne "primary" peut être utilisée explicitement. Toutes les lettres en minuscules sont requises si la chaîne "primary" est utilisée explicitement. Tout déclencheur de règles ne contenant pas de caractère deux-points (`:`) est interprété comme un nom d'attribut POP.

```
Printer:rxT = eas_plugin -weight 60 & -server barney
```

L'exemple suivant concerne un attribut POP trigger dénommé **webseal_pop_trigger**. Lorsqu'une règle POP qui contient une référence à cette chaîne est rencontrée, le service d'autorisation externe approprié est appelé pour participer à la décision d'accès.

```
webseal_pop_trigger = eas_plugin_2 -weight 70 & -hostname fred
```

Notez que pour que l'attribut POP trigger fonctionne, la configuration des règles POP doit avoir été précédemment effectuée par l'administrateur de domaine sécurisé à l'aide des commandes **pdadmin pop**.

L'exemple présenté ci-dessous est un exemple de configuration pour le service EAS OAuth, où le fichier`/opt/pdweb/etc/oauth_eas.conf` contient la section **[oauth-eas]** et le section correspondante **[tfim-cluster:<cluster>]**. Cet exemple est entré sous la forme d'une ligne unique dans le fichier de configuration WebSEAL :

```
webseal_pop_trigger = /opt/pdwebrte/lib/libamwoautheas.so & /opt/pdweb/etc
/oauth_eas.conf
```

Section [azn-decision-info]

azn-decision-info

Syntaxe

<attr-name> = <http-info>

Description

Cette section définit les informations supplémentaires disponibles pour la structure d'autorisation lors de la prise des décisions d'autorisation. Ces informations supplémentaires peuvent être obtenues à partir de divers éléments de la demande HTTP, à savoir :

- Méthode HTTP
- Schéma HTTP
- URI de la demande
- En-têtes HTTP
- Données POST

Si l'élément demandé ne pas contenu dans la demande HTTP, aucun attribut correspondant n'est ajouté à l'information de décision d'autorisation.

Options

<attr-name>
Nom de l'attribut qui contient les informations HTTP.

<http-info>
Source des informations. Il peut s'agir de l'une des valeurs suivantes :

- Méthode
- Schéma
- Identificateur URI
- En-tête :<header-name>
- Données de post :<post-data-name>

Utilisation

Cette entrée de section est obligatoire lors de la configuration de l'authentification EAS OAuth et doit contenir les éléments suivants :

```
HTTP_REQUEST_METHOD = method
HTTP_REQUEST_SCHEME = scheme
HTTP_REQUEST_URI = uri
HTTP_HOST_HDR = header:host
HTTP_CONTENT_TYPE_HDR = header:content-type
HTTP_TRANSFER_ENCODING_HDR = header:transfer-encoding
HTTP_AZN_HDR = header:authorization
```

Valeur par défaut

N/A

Exemple

```
HTTP_REQUEST_METHOD = method
HTTP_HOST_HEADER= header:Host
```

Section [aznapi-configuration]

resource-manager-provided-adi

Syntaxe

`resource-manager-provided-adi = *préfixe*`

Description

Liste de préfixes de chaînes identifiant les informations ADI (Access Decision Information) qui doivent être fournies par le gestionnaire de ressource (dans ce cas, WebSEAL).

Options

préfixe Les paramètres par défaut indiquent au moteur d'autorisation que lorsqu'il nécessite des informations ADI avec les préfixes AMWS_hd_, AMWS_qs_ ou AMWS_pb_ pour évaluer une règle d'autorisation booléenne et que ces informations ADI ne sont disponibles ni dans l'accréditation ni dans le contexte d'application transmis avec l'appel de décision d'accès, ce moteur doit mettre en échec la décision d'accès et demander au gestionnaire de ressources de retenter la demande et de fournir les données requises dans le contexte d'application de la demande suivante.

Utilisation

Cette entrée de section est obligatoire lors de la configuration de l'authentification EAS OAuth.

Valeur par défaut

```
AMWS_hd_, AMWS_pb_, AMWS_qs_
```

Exemple

```
resource-manager-provided-adi = AMWS_hd_
resource-manager-provided-adi = AMWS_pb_
resource-manager-provided-adi = AMWS_qs_
```

Section [oauth-eas]

Vous pouvez configurer la section [oauth-eas] pour prendre en charge les décisions d'autorisation OAuth dans le cadre des demandes WebSEAL.

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Remarque : Cette section peut être incluse dans un fichier de configuration distinct qui est spécifié pour **amwoautheas** dans la section **[aznapi-external-authzn-services]**.

Exemple de données de configuration du service EAS

La section **[oauth-eas]** contient les informations de configuration relatives au service EAS OAuth.

Cet exemple présente les données de configuration du service EAS OAuth avec l'ID de fédération par défaut et le mode défini pour une fédération OAuth 1.0.

```
[oauth-eas]

# The maximum number of OAuth 2.0 bearer token authorization decisions to cache.
# This EAS has a built in cache for storing authorization decisions so that
# repeated use of the same OAuth 2.0 bearer token does not require repeated
```

```
# requests to TFIM. Bearer token decisions can be cached because they do not
# require signing of the request, unlike OAuth 1.0 requests. The lifetime of the
# cache entry is based on the Expires attribute returned by TFIM. If this
# attribute is not returned, the decision will not be cached.
#
# This EAS implements a Least Recently Used cache, meaning the decision
# associated with the least recently used bearer token will be forgotten when a
# new bearer token decision is cached. A cache-size of 0 will disable caching of
# authorization decisions
cache-size = 0

# The Provider ID of the default OAuth federation at TFIM. If a Provider ID
# is not provided in the request using the fed-id-param option, this provider
# ID will be used for OAuth requests. The Provider ID of a federation can be
# found on the federation properties page.
default-fed-id = https://server.oauth.com/FIM/sps/oauthfed/oauth10

# The name of the request parameter that can be used to override the
# default-fed-id option configured above. By deleting this configuration
# option, you can enforce that the default fed id is always used.
fed-id-param = FederationId

# The default OAuth mode that this EAS will operate under. It affects the
# validation of request parameters, as well as the construction of the RST
# sent to TFIM. The default mode can be overriden for an individual request
# by providing a valid mode value [OAuth10|OAuth20Bearer] in a request
# parameter with the name specified in the mode-param option below.
default-mode = OAuth10

# The name of the request parameter that can be used to override the
# default-mode option configured above. By deleting this configuration
# option, you can enforce that the default mode is always used.
mode-param = mode

# The name of the OAuth realm which will be used in a 401 request
# for OAuth data.
realm-name = oauth-realm

# The name of the file which contains the body used when constructing a
# '400 Bad Request' response.  This response will be generated when
# required OAuth elements are missing from a request.
bad-request-rsp-file = /EAS/oauth_eas/400.html

# The name of the file which contains the body used when constructing a
# '401 Unauthorized' response.  This response will be generated when:
#     - all OAuth data is missing from a request, or
#     - the OAuth data fails validation.
unauthorized-rsp-file = /EAS/oauth_eas/401.html

# The name of the file which contains the body used when constructing a
# '502 Bad Gateway' response.  This response will be generated when
# TFIM fails to process the request.
bad-gateway-rsp-file = /EAS/oauth_eas/502.html

# The name of the TAM trace component which is used by the EAS.
trace-component = pdweb.oauth

# Should the native TAM ACL policy still take affect, in addition to the
# OAuth authorization?
apply-tam-native-policy = false

# The name of the TFIM cluster which houses this OAuth service.  There should
# also be a corresponding [tfim-cluster:<cluster>] stanza which contains the
# definition of the cluster.
cluster-name = oauth-cluster

[tfim-cluster:oauth-cluster]
```

```
#
# This stanza contains definitions for a particular cluster of TFIM
# servers.
#

#
# A specification for the server which is used when communicating with a
# single TFIM server which is a member of this cluster.  Values for this
# entry are defined as follows:
#
#       {[0-9],}<URL>
#
# Where the first digit (if present) represents the priority of the server
# within the cluster (9 being the highest, 0 being lowest).  If the priority
# is not specified, a priority of 9 is assumed.  The <URL> can be any
# well-formed HTTP or HTTPS URL.
#
# Multiple server entries can be specified for failover and load balancing
# purposes.  The complete set of these server entries defines the
# membership of the cluster for failover and load balancing.
#
# server = 9,http://tfim.example.com/TrustServerWST13/services/RequestSecurityToken

...
```

cache-size

Syntaxe

```
cache-size = taille_cache
```

Description

Nombre maximal de décisions d'autorisation de jeton bearer OAuth 2.0 à mettre en cache. Ce cache stocke les décisions d'autorisation de sorte que l'utilisation répétée du même jeton ne nécessite pas des demandes répétées à Tivoli Federated Identity Manager. Un paramètre `cache-size` de 0 désactive la mise en cache des décisions d'autorisation.

Options

taille_cache
: Taille du cache de jetons OAuth.

Utilisation

Cette entrée de section est obligatoire lors de la configuration du service EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
cache-size = 2
```

default-fed-id

Syntaxe

```
default-fed-id = id_fournisseur
```

Description

L'ID du fournisseur est la fédération OAuth par défaut dans Tivoli Federated Identity Manager. Si un ID fournisseur n'est pas indiqué dans la demande à l'aide de l'option **fed-id-param**, cet ID fournisseur est utilisé pour les demandes OAuth. L'ID fournisseur d'une fédération figure dans la page des propriétés de la fédération.

Options

id_fournisseur
ID fournisseur de la fédération OAuth.

Utilisation

Cette entrée de section est obligatoire lors de la configuration de l'authentification EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
default-fed-id = https://server.oauth.com/FIM/MySocialNetwork/oauth20
```

fed-id-param

Syntaxe

```
fed-id-param = nom_param_demande
```

Description

Nom du paramètre de demande pouvant être utilisé pour remplacer l'option **default-fed-id**. Cette valeur doit correspondre à l'ID fournisseur de la fédération pour laquelle le client OAuth est un membre.

Supprimez cette option de configuration pour imposer l'utilisation systématique de l'ID fédération par défaut.

Options

nom_param_demande
Nom du paramètre de demande.

Utilisation

Cette entrée de section est facultative. Si elle n'est pas fournie, la valeur de l'option **default-fed-id** est utilisée en tant que valeur d'ID fournisseur statique dans les appels au service STS.

Valeur par défaut

Aucune.

Exemple

```
fed-id-param = FederationId
```

default-mode

Syntaxe

```
default-mode = valeur_mode
```

Description

Mode OAuth par défaut selon lequel fonctionne le service EAS. Il a une incidence sur la validation des paramètres de demande, ainsi que sur la construction du jeton RST envoyé à Tivoli Federated Identity Manager. Fournissez une valeur de mode valide dans un paramètre de demande avec le nom spécifié dans l'option **mode-param** pour remplacer le mode par défaut pour une demande individuelle.

Options

valeur_mode
La valeur de mode valide pour le protocole OAuth 1.0 est `OAuth10` et la valeur de mode valide pour le protocole OAuth 2.0 est `OAuth20Bearer`.

Utilisation

Cette entrée de section est obligatoire lors de la configuration de l'authentification EAS OAuth.

Valeur par défaut

Aucune.

Exemple

Pour OAuth 1.0 :

```
default-mode = OAuth10
```

Pour OAuth 2.0 :

```
default-mode = OAuth20Bearer
```

mode-param

Syntaxe

```
mode-param = nom_param_demande
```

Description

Nom du paramètre de demande pouvant être personnalisé pour refléter la valeur de mode. Cette option de configuration peut être utilisée pour remplacer l'option **default-mode**. Supprimez cette option de configuration pour imposer l'utilisation systématique du mode par défaut.

Options

nom_param_demande
Nom du paramètre de demande.

Utilisation

Cette entrée de section est facultative. Si elle n'est pas fournie, la valeur de l'option **default-mode** est utilisée pour déterminer la validation ou non de la demande

entrante en tant que OAuth 1.0 ou OAuth 2.0.

Valeur par défaut

Aucune.

Exemple

```
mode-param = mode
```

realm-name

Syntaxe

```
realm-name = nom_domaine
```

Description

Nom du domaine OAuth utilisé dans une demande 401 pour des données OAuth.

Options

nom_domaine
Nom du domaine OAuth.

Utilisation

Cette entrée de section est obligatoire lors de la configuration de l'authentification EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
realm-name = realmOne
```

bad-request-rsp-file

Syntaxe

```
bad-request-rsp-file = nom_fichier
```

Description

Nom qualifié complet du fichier contenant le corps du texte utilisé lors de la construction d'une réponse "400 Demande incorrecte". Cette réponse est générée lorsque des éléments OAuth obligatoires sont manquants dans une demande.

Options

nom_fichier
Nom du fichier de réponse 400 Demande incorrecte.

Utilisation

Cette entrée de section est obligatoire lors de la configuration du service EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
bad-request-rsp-file = /tmp/bad_rqst.html
```

Voici un exemple de fichier de réponse HTML :

```
<html>
<body>
400 Bad Request
</body>
</html>
```

unauthorized-rsp-file

Syntaxe

```
unauthorized-rsp-file = nom_fichier
```

Description

Nom complètement qualifié du fichier contenant le corps de texte utilisé lors de la construction de la réponse "401 Non autorisé". Cette réponse est générée dans les cas suivants :

- Toutes les données OAuth sont manquantes dans une demande ou
- La validation des données OAuth échoue.

Options

nom_fichier
Nom du fichier de réponse 401 Non autorisé.

Utilisation

Cette entrée de section est obligatoire lors de la configuration du service EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
unauthorized-rsp-file = /tmp/unauth_response.html
```

Voici un exemple de fichier de réponses HTML :

```
<html>
<body>
401 Unauthorized
</body>
</html>
```

bad-gateway-rsp-file

Syntaxe

```
bad-gateway-rsp-file = nom_fichier
```

Description

Nom qualifié complet du fichier qui contient le corps de texte utilisé lors de la construction d'une réponse "502 Passerelle incorrecte". Cette réponse est générée lorsque Tivoli Federated Identity Manager ne parvient pas à traiter la demande.

Options

nom_fichier
Nom de fichier de réponse 502 Passerelle incorrecte.

Utilisation

Cette entrée de section est obligatoire lors de la configuration du service EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
bad-gateway-rsp-file = /tmp/bad_gateway.html
```

Voici un exemple de fichier de réponse HTML :

```
<html>
<body>
502 Bad Gateway
</body>
</html>
```

trace-component

Syntaxe

```
trace-component = nom_composant
```

Description

Nom du composant de trace Tivoli Access Manager qui est utilisé par le service EAS.

Options

nom_composant
Nom du composant de trace Tivoli Access Manager.

Utilisation

Cette entrée de section est obligatoire lors de la configuration de l'authentification EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
trace-component = pdweb.oauth
```

apply-tam-native-policy

Syntaxe

```
apply-tam-native-policy = <true | false>
```

Description

Détermine si la règle Tivoli Access Manager ACL native continue à s'appliquer, en plus de l'autorisation OAuth.

Options

vrai La règle Tivoli Access Manager ACL native continue à s'appliquer.

false La règle Tivoli Access Manager ACL native ne s'applique pas.

Utilisation

Cette entrée de section est obligatoire lors de la configuration du service EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
apply-tam-native-policy = false
```

cluster-name

Syntaxe

```
cluster-name = nom_cluster
```

Description

Nom du cluster Tivoli Federated Identity Manager hébergeant ce service OAuth. Il doit également exister une section **[tfim-cluster:<cluster>]** correspondante contenant la définition du cluster.

Options

nom_cluster
Nom du cluster Tivoli Federated Identity Manager dans lequel le service OAuth est hébergé.

Utilisation

Cette entrée de section est obligatoire lors de la configuration du service EAS OAuth.

Valeur par défaut

Aucune.

Exemple

```
cluster-name = oauth-cluster
```

Pour cet exemple, un section **[tfim-cluster:oauth-cluster]** correspondante doit être présente pour définir le cluster.

Modèles de pages OAuth 1.0 et OAuth 2.0 pour la gestion des clients de confiance

Tivoli Federated Identity Manager fournit un modèle de page HTML pouvant être utilisé par les propriétaires de ressources pour afficher et gérer les informations des clients de confiance pour les fédérations OAuth 1.0 et OAuth 2.0.

Il existe plusieurs modèles de pages de gestion des clients de confiance pour chaque protocole OAuth. Ces pages ont la même présentation et utilisent les mêmes macros de remplacement. Les modèles de page pour OAuth 1.0 et OAuth 2.0 sont tous deux nommés `clients_manager.html`.

Le propriétaire de la ressource établit le client OAuth via la page `user_consent.html` pendant les demandes d'autorisation.

Les modèles comprennent les macros de remplacement suivantes :

@USERNAME@
Cette macro est remplacée par le nom d'utilisateur Tivoli Federated Identity Manager.

@OAUTH_CLIENT_COMPANY_NAME@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT trustedClients]`. Ces valeurs sont remplacées par le nom de la société qui souhaite accéder à la ressource protégée.

@PERMITTED_SCOPES@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT trustedClients]`. Les valeurs sont remplacées par les portées de jeton auxquelles le client OAuth peut accéder.

@DENIED_SCOPES@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT trustedClients]`. Les valeurs sont remplacées par les portées de jeton auxquelles le client OAuth ne peut *pas* accéder.

@OAUTH_CUSTOM_MACRO@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT trustedClients]`. Ces valeurs sont remplacées par des informations de client de confiance qui contiennent des informations additionnelles sur un client OAuth autorisé.

@OAUTH_CLIENTMANAGERURL@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT trustedClients]`. Les valeurs sont remplacées par le noeud final du gestionnaire de clients de confiance.

@UNIQUE_ID@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT trustedClients]`. Les valeurs sont remplacées par un identificateur unique qui identifie les informations des clients de confiance pour chaque entrée de la liste.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Gestionnaire de clients OAuth</title>
 </head>
 <body>
  Nom d'utilisateur : <b>@USERNAME@</b>
  <p/>
  Clients de confiance<br />
  <table border="1">
    <tr><td>Client</td><td>Portées autorisées</td><td>Portées refusées</td>
         <td>Informations supplémentaires</td><td>Action</td></tr>
<!-- START NON-TRANSLATABLE -->
   [RPT trustedClients]
<!-- END NON-TRANSLATABLE -->
    <tr>
      <td>@OAUTH_CLIENT_COMPANY_NAME@</td>
      <td>@PERMITTED_SCOPES@</td>
      <td>@DENIED_SCOPES@</td>
      <td>@OAUTH_CUSTOM_MACRO@</td>
      <td><a href="@OAUTH_CLIENTMANAGERURL@?action=remove&id=
          @UNIQUE_ID@">Supprimer</a></td>
    </tr>
<!-- START NON-TRANSLATABLE -->
   [ERPT trustedClients]
<!-- END NON-TRANSLATABLE -->
  </table>
 </body>
</html>
```

Figure 45. Modèle pour clients_manager.html

Modèle de page OAuth 1.0 pour l'accord d'autorisation

Le serveur OAuth utilise cette page pour déterminer et stocker les informations d'accord de l'utilisateur concernant les clients OAuth qui sont autorisés à accéder à la ressource protégée. Cette page indique également la portée qui est demandée par le client OAuth.

Tivoli Federated Identity Manager fournit un modèle de page HTML appelé `user_consent.html`.

Tivoli Federated Identity Manager stocke les décisions prises par le propriétaire de la ressource concernant les clients OAuth de confiance. Le propriétaire de la ressource n'est pas invité à entrer ces décisions à chaque fois que le même client demande l'autorisation d'accéder à la ressource protégée.

La demande d'autorisation du client OAuth affiche la liste des portées approuvées et la liste des portées à approuver. Ces listes sont affichées dans la page de consentement et peuvent être de longueur indéterminée. Le modèle prend en charge de multiples copies de sections répétées une fois pour chaque portée dans la liste concernée.

Ce fichier modèle prend en charge plusieurs macros de remplacement :

@OAUTH_AUTHORIZE_URI@
Cette macro est remplacée par l'identificateur URI du noeud final d'autorisation du propriétaire de la ressource .

@OAUTH_CLIENT_CALLBACK@
Cette macro est remplacée par l'URI de rappel utilisé par le serveur OAuth pour envoyer le code de vérification. La valeur dépend des éléments suivants :

- L'URI de rappel entré pendant l'enregistrement du partenaire.
- Le paramètre `oauth_callback` figurant dans la demande pour des données d'identification temporaires.
- Le remplacement du paramètre d'URI de rappel du client enregistré.

@OAUTH_CLIENT_COMPANY_NAME@
Cette macro est remplacée par le nom de la société qui souhaite accéder à la ressource protégée.

@OAUTH_CUSTOM_MACRO@
Cette macro est remplacée par des informations client dignes de confiance qui contiennent des informations additionnelles sur un client OAuth autorisé.

@USERNAME@
Cette macro est remplacée par le nom d'utilisateur Tivoli Federated Identity Manager.

@OAUTH_OTHER_PARAM_REPEAT@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT oauthOtherParamsRepeatable]`. Les valeurs indiquent la liste des noms de paramètre supplémentaires.

@OAUTH_OTHER_PARAM_VALUE_REPEAT@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT oauthOtherParamsRepeatable]`. Les valeurs indiquent la liste des valeurs de paramètre supplémentaires.

@OAUTH_TOKEN_SCOPE_REPEAT@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT oauthTokenScopePreapprovedRepeatable]` ou `[RPT oauthTokenScopeNewApprovalRepeatable]`. Les valeurs dans `[RPT oauthTokenScopePreapprovedRepeatable]` affichent la liste des portées de jeton qui ont été précédemment approuvées par le propriétaire de la ressource. Les valeurs contenues dans `[RPT oauthTokenScopeNewApprovalRepeatable]` affichent également la liste des portées de jeton qui n'ont *pas* encore été approuvées par le propriétaire de la ressource.

@CONSENT_FORM_VERIFIER@
Cette macro est remplacée par un identificateur unique pour la valeur du paramètre `consent_form_verifier`. La valeur du paramètre `consent_form_verifier` est automatiquement générée par le serveur OAuth. Le nom et la valeur du paramètre ne doivent pas être modifiés.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>OAuth - Accord d'autorisation</title>
  </head>
  <body>
      <h1>OAuth - Accord d'autorisation</h1>
      <br />

       <p>Le site suivant demande l'accès à une ressource protégée OAuth :</p>
       <p>@OAUTH_CLIENT_CALLBACK@</p>
       <p>Nom de l'entreprise : @OAUTH_CLIENT_COMPANY_NAME@</p>
       <p>Informations supplémentaires : @OAUTH_CUSTOM_MACRO@</p>
       <br />

       <p>Nom d'utilisateur : @USERNAME@</p>
       <br />

       <form action="@OAUTH_AUTHORIZE_URI@" method="post">
           <p>Le client a fourni les paramètres de requête supplémentaires suivants :</p>
           <!-- START NON-TRANSLATABLE -->
           [RPT oauthOtherParamsRepeatable]
           @OAUTH_OTHER_PARAM_REPEAT@=@OAUTH_OTHER_PARAM_VALUE_REPEAT@
           <input type="hidden" name="@OAUTH_OTHER_PARAM_REPEAT@"
                 value="@OAUTH_OTHER_PARAM_VALUE_REPEAT@" />
           [ERPT oauthOtherParamsRepeatable]
           <!-- END NON-TRANSLATABLE -->
           <br />

           <p>Le client a demandé les portées de jeton suivantes
                           qui ont été validées précédemment :</p>
           <!-- START NON-TRANSLATABLE -->
           <ul>
           [RPT oauthTokenScopePreapprovedRepeatable]
           <li>@OAUTH_TOKEN_SCOPE_REPEAT@</li>
           <input type="hidden" name="scope" value="@OAUTH_TOKEN_SCOPE_REPEAT@" />
           [ERPT oauthTokenScopePreapprovedRepeatable]
           </ul>
           <!-- END NON-TRANSLATABLE -->
           <br />

           <p>Le client a demandé les portées de jeton suivantes
                           qui n'ont pas encore été validées :</p>
           <!-- START NON-TRANSLATABLE -->
           [RPT oauthTokenScopeNewApprovalRepeatable]
           <input type="checkbox" name="scope" value="@OAUTH_TOKEN_SCOPE_REPEAT@"
                 checked="checked"/> <label>@OAUTH_TOKEN_SCOPE_REPEAT@</label><br />
           [ERPT oauthTokenScopeNewApprovalRepeatable]
           <!-- END NON-TRANSLATABLE -->
           <br />

        <p>Voulez-vous valider cet accès ?</p>
           <br />
   <input type="hidden" name="consent_form_verifier" value="@CONSENT_FORM_VERIFIER@" />

           <!--
               Les paramètres de la portée peuvent être :
               1. Demandés dans le cadre du réacheminement pour l'autorisation par le client
                  en les ajoutant à l'URL d'autorisation en tant que paramètres de chaîne de requête, ou
               2. S'ils ne sont pas demandés par le client, et que vous savez quelles options d'autorisation
                  sont valides pour les ressources protégées OAuth demandées, vous pouvez
                  également les demander manuellement dans ce modèle de page comme illustré
                  par l'exemple suivant
           -->
           <!--
           <table>
               <tr>
                   <td>Portées à autoriser : </td>
                   <td>Portée 1</td><td><input type="checkbox" name="scope"
                        value="token_scope_1" /></td>
                   <td>:: Portée 2</td><td><input type="checkbox" name="scope"
                        value="token_scope_2" /></td>
                   <td>:: Portée 3</td><td><input type="checkbox" name="scope"
                        value="token_scope_3" /></td>
               </tr>
           </table>
           -->

         <table>
           <tr><td>Autoriser </td><td><input type="radio" name="trust_level"
                    value="permit" checked /></td></tr>
           <tr><td>Refuser </td><td><input type="radio" name="trust_level"
                    value="deny" /></td></tr>
         </table>
         <br />

         <input type="submit" name="submit" value="Submit" style="width:80px"/>
       </form>
  </body>
</html>
```

Figure 46. Modèle pour user_consent.html

Modèle de page OAuth 1.0 pour les réponses

Utilisez cette page HTML lorsque l'URI de rappel est défini sur **oob** dans la demande de données d'identification temporaires ou dans l'enregistrement du partenaire.

Lorsque le client OAuth n'indique pas d'URI de rappel ou ne peut pas recevoir de rappels, le serveur OAuth ne sait pas où rediriger le propriétaire de la ressource une fois le processus d'autorisation terminé. En conséquence, le client OAuth ne reçoit pas le code de vérification qu'il doit échanger contre un ensemble de droits de jeton.

Tivoli Federated Identity Manager fournit un modèle de page HTML appelé `user_response.html`. Cette page contient le jeton OAuth et le code de vérification que le propriétaire de la ressource peut fournir à un client OAuth sécurisé.

Le modèle comprend les macros de remplacement suivantes :

@OAUTH_TOKEN@
Cette macro est remplacée par le paramètre `oauth_token` indiqué dans la requête pour des droits temporaires.

@OAUTH_VERIFIER@
Cette macro est remplacée par le paramètre `oauth_verifier` indiqué dans la réponse d'autorisation.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>OAuth - Réponse</title>
 </head>
 <body>
   <h1>OAuth - Réponse</h1>   <br />
    <p>Votre client OAuth n'a pas indiqué une URL de rappel.
      Fournissez ces valeurs à votre client :</p>
    <br />
    <p>Jeton OAuth : <span class="client">@OAUTH_TOKEN@</span></p>
    <br />
    <p>Code de vérification OAuth : <span class="client">
                @OAUTH_VERIFIER@</span></p>
   </div>
  </div>
 </body>
</html>
```

Figure 47. Modèle pour user_response.html

Modèle de page OAuth 1.0 pour l'accord refusé

Utilisez la page HTML d'accord refusé lorsque le propriétaire de la ressource n'a pas accordé au client OAuth l'accès à la ressource protégée.

Tivoli Federated Identity Manager fournit le fichier `user_consent_denied.html`.

Le modèle ne comporte aucune macro de remplacement.

```
!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>OAuth - Accord refusé</title>
 </head>
 <body>
   <h1>OAuth - Accord refusé</h1>
   <br />
   <div id="content">
    <p>Vous avez refusé l'accord concernant l'accès à vos ressources protégées.<p>
   </div>
  </div>
 </body>
</html>
```

Figure 48. Modèle pour user_consent_denied.html

Modèle de page OAuth 1.0 pour les erreurs

Tivoli Federated Identity Manager utilise un modèle de page d'erreur générique pour afficher des informations de texte détaillées lorsqu'une erreur se produit dans un flux OAuth 1.0.

Le modèle de page est `user_error.html`.

La macro de remplacement suivante est prise en charge :

@OAUTH_ERROR@
: Cette macro est remplacée par la version en support de langue nationale (NLS) du message d'erreur associé à l'erreur.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>OAuth - Erreur</title>
 </head>
 <body>
   <h1>OAuth - Erreur</h1>
   <br />
   <p>L'erreur suivante s'est produite lors du traitement de votre demande OAuth : </p>
   <p><span class="client">@OAUTH_ERROR@</span></p>
 </body>
</html>
```

Figure 49. Modèle pour user_error.html

Modèle de page OAuth 2.0 pour l'accord d'autorisation

Le serveur d'autorisation utilise cette page pour déterminer et stocker les informations d'accord de l'utilisateur concernant les clients OAuth qui sont autorisés à accéder à la ressource protégée. Cette page indique également les portées demandées par le client OAuth.

Tivoli Federated Identity Manager fournit un modèle de page HTML appelé `user_consent.html`. Les macros figurant dans le modèle sont spécifiquement destinées à un flux OAuth 2.0.

Tivoli Federated Identity Manager stocke les décisions prises par le propriétaire de la ressource concernant les clients OAuth de confiance. Le propriétaire de la ressource n'est pas invité à entrer ces décisions à chaque fois que le même client OAuth demande l'autorisation d'accéder à la ressource protégée.

La demande d'autorisation du client OAuth affiche la liste des portées approuvées et la liste des portées à approuver. Ces listes sont affichées dans la page de consentement et peuvent être de longueur indéterminée. Le modèle prend en charge de multiples copies de sections répétées une fois pour chaque portée dans la liste concernée.

Ce fichier modèle prend en charge plusieurs macros de remplacement :

@OAUTH_AUTHORIZE_URI@
Cette macro est remplacée par l'URI du noeud final d'autorisation.

@OAUTH_CLIENT_COMPANY_NAME@
Cette macro est remplacée par le nom de la société qui souhaite accéder à la ressource protégée.

@CLIENT_ID@
Cette macro est remplacée par le paramètre `client_id` indiqué dans la requête d'autorisation.

@REDIRECT_URI@
Cette macro est remplacée par l'URI de réacheminement utilisé par le serveur d'autorisation pour envoyer le code d'autorisation. La valeur dépend des éléments suivants :

- L'URI de réacheminement saisi pendant l'enregistrement du partenaire
- Le paramètre `oauth_redirect` spécifié dans la requête d'autorisation

@STATE@
Cette macro est remplacée par le paramètre `state` spécifié dans la requête d'autorisation.

@RESPONSE_TYPE@
Cette macro est remplacée par le paramètre `response_type` spécifié dans la requête d'autorisation.

@OAUTH_CUSTOM_MACRO@
Cette macro est remplacée par des informations client dignes de confiance qui contiennent des informations additionnelles sur un client OAuth autorisé.

@USERNAME@
Cette macro est remplacée par le nom d'utilisateur Tivoli Federated Identity Manager.

@OAUTH_OTHER_PARAM_REPEAT@
Macro à valeurs multiples contenue dans une liste de remplacement répétable `[RPT oauthOtherParamsRepeatable]`. Les valeurs indiquent la liste des noms de paramètre supplémentaires.

@OAUTH_OTHER_PARAM_VALUE_REPEAT@
Macro à valeurs multiples contenue dans une liste de remplacement

répétable [RPT oauthOtherParamsRepeatable]. Les valeurs indiquent la liste des valeurs de paramètre supplémentaires.

@OAUTH_TOKEN_SCOPE_REPEAT@

Macro à valeurs multiples contenue dans une liste de remplacement répétable [RPT oauthTokenScopePreapprovedRepeatable] ou [RPT oauthTokenScopeNewApprovalRepeatable]. Les valeurs dans [RPT oauthTokenScopePreapprovedRepeatable] affichent la liste des portées de jeton qui ont été précédemment approuvées par le propriétaire de la ressource. Les valeurs contenues dans [RPT oauthTokenScopeNewApprovalRepeatable] affichent également la liste des portées de jeton qui n'ont *pas* encore été approuvées par le propriétaire de la ressource.

@CONSENT_FORM_VERIFIER@

Cette macro est remplacée par un identificateur unique pour la valeur du paramètre consent_form_verifier. La valeur du paramètre consent_form_verifier est générée automatiquement par le serveur d'autorisation. Le nom et la valeur du paramètre ne doivent pas être modifiés.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>OAuth 2.0 - Accord d'autorisation</title>
  </head>
  <body>
    <form action="@OAUTH_AUTHORIZE_URI@" method="GET">
      <h1>OAuth 2.0 - Accord d'autorisation</h1>

      <p>Le site suivant demande l'accès à une ressource protégée OAuth 2.0 :</p>
      <p><b>@OAUTH_CLIENT_COMPANY_NAME@</b></p>

      <p>Le client a fourni les paramètres de requête OAuth 2.0 suivants :</p>
      <ul>
        <li>ID client : @CLIENT_ID@</li>
        <li>URI de réacheminement : @REDIRECT_URI@</li>
        <li>Etat : @STATE@</li>
        <li>Type de réponse : @RESPONSE_TYPE@</li>
      </ul>

      <p>Informations supplémentaires : @OAUTH_CUSTOM_MACRO@</p>

      <p>En validant cette requête, vous allez fournir
                                une autorisation déléguée pour le compte de :</p>
      <p><b>@USERNAME@</b></p>

      <p>Le client a fourni les paramètres de requête supplémentaires suivants :</p>
      <!-- START NON-TRANSLATABLE -->
      <ul>
      [RPT oauthOtherParamsRepeatable]
      <li>@OAUTH_OTHER_PARAM_REPEAT@=@OAUTH_OTHER_PARAM_VALUE_REPEAT@</li>
      <input type="hidden" name="@OAUTH_OTHER_PARAM_REPEAT@"
                                  value="@OAUTH_OTHER_PARAM_VALUE_REPEAT@" />
      [ERPT oauthOtherParamsRepeatable]
      </ul>
      <!-- END NON-TRANSLATABLE -->

      <p>Le client a demandé les portées de jeton suivantes
                                    qui ont été validées précédemment :</p>
      <!-- START NON-TRANSLATABLE -->
      <ul>
      [RPT oauthTokenScopePreapprovedRepeatable]
      <li>@OAUTH_TOKEN_SCOPE_REPEAT@</li>
      <input type="hidden" name="scope" value="@OAUTH_TOKEN_SCOPE_REPEAT@" />
      [ERPT oauthTokenScopePreapprovedRepeatable]
      </ul>
      <!-- END NON-TRANSLATABLE -->

      <p>Le client a demandé les portées de jeton suivantes
                                     qui n'ont pas encore été validées :</p>
      <!-- START NON-TRANSLATABLE -->
      [RPT oauthTokenScopeNewApprovalRepeatable]
      <input type="checkbox" name="scope" value="@OAUTH_TOKEN_SCOPE_REPEAT@"
                 checked="checked"/><label>@OAUTH_TOKEN_SCOPE_REPEAT@</label><br />
      [ERPT oauthTokenScopeNewApprovalRepeatable]
      <!-- END NON-TRANSLATABLE -->
      <br />

      <p>Voulez-vous valider l'accès à cette portée ?</p>
      <input type="hidden" name="consent_form_verifier" value="@CONSENT_FORM_VERIFIER@" />

            <!--
                Les paramètres de la portée peuvent être :
                1. Demandés dans le cadre du réacheminement pour l'autorisation par le client
                   en les ajoutant à l'URL d'autorisation en tant que paramètres de chaîne de requête,
                   et/ou
                2. S'ils ne sont pas demandés par le client, et que vous savez quelles options d'autorisation
                   sont valides pour les ressources protégées demandées, vous pouvez
                   également les demander manuellement dans ce modèle de page comme illustré
                   par l'exemple suivant
            -->
            <!--
            <table>
                <tr>
                    <td>Portées à autoriser : </td>
                    <td>Portée 1</td><td><input type="checkbox" name="scope"
                               value="token_scope_1" /></td>
                    <td>:: Portée 2</td><td><input type="checkbox" name="scope"
                               value="token_scope_2" /></td>
                    <td>:: Portée 3</td><td><input type="checkbox" name="scope"
                               value="token_scope_3" /></td>
                </tr>
            </table>
            -->

      <table>
        <tr>
          <td>Autoriser </td>
          <td><input type="radio" name="trust_level" value="permit" checked /></td>
        </tr>
        <tr>
          <td>Refuser </td>
          <td><input type="radio" name="trust_level" value="deny" /></td>
        </tr>
      </table>
      <br />

      <input type="submit" name="submit" value="Submit" style="width: 80px" />
    </form>
  </body>
</html>
```

Figure 50. Modèle pour user_consent.html

Modèle de page OAuth 2.0 pour les réponses

Utilisez cette page HTML pour afficher le code d'autorisation d'un client OAuth qui n'a pas spécifié d'URI de réacheminement lors de l'enregistrement du partenaire.

Lorsque le client OAuth n'indique pas d'URI de réacheminement ou ne peut pas recevoir des réacheminements, le serveur d'autorisation ne sait pas où envoyer le propriétaire de la ressource une fois le processus d'autorisation terminé. Le client OAuth ne reçoit pas le code d'autorisation requis pour l'échanger contre un jeton d'accès ou un jeton de régénération.

Tivoli Federated Identity Manager fournit un modèle de page HTML appelé `user_response.html`. Cette page contient le code d'autorisation que le propriétaire de la ressource peut fournir à un client OAuth sécurisé.

La macro de remplacement suivante est prise en charge :

@OAUTH_CODE@
: Cette macro est remplacée par le paramètre `oauth_code` indiqué dans la réponse d'autorisation.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>OAuth - Réponse</title>
  </head>
  <body>
      <h1>OAuth - Réponse</h1>
      <br />
        <p>Votre client OAuth n'a pas indiqué d'URI de réacheminement.
                Fournissez cette valeur à votre client :</p>
        <br />
        <p>Code d'autorisation OAuth : <span class="client">@OAUTH_CODE@</span></p>
  </body>
</html>
```

Figure 51. Modèle pour user_response.html

Modèle de page OAuth 2.0 pour les erreurs

Tivoli Federated Identity Manager utilise un modèle de page d'erreur générique pour afficher des informations de texte détaillées lorsqu'une erreur se produit dans un flux OAuth 2.0.

Le modèle de page est `user_error.html`.

La macro de remplacement suivante est prise en charge :

@ERROR_CODE@
: Cette macro est remplacée par des caractères qui identifient de manière unique l'erreur.

@ERROR_DESCRIPTION@
: Cette macro est remplacée par la version en support de langue nationale (NLS) du message d'erreur associé à l'erreur.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>OAuth 2.0 - Erreur</title>
  </head>
  <body>
    <h1>OAuth 2.0 - Erreur</h1>

    <p>L'erreur suivante a été détectée lors du traitement de votre demande OAuth :</p>

    <p>Code d'erreur : <b>@ERROR_CODE@</b></p>
    <p>Description de l'erreur : <b>@ERROR_DESCRIPTION@</b></p>
  </body>
</html>
```

Figure 52. Modèle HTML pour user_error

Chapitre 30. Planification d'une fédération Liberty

Vous devez, lors de la configuration d'une fédération Liberty, spécifier des valeurs pour les propriétés de la fédération. Gardez toutefois à l'esprit que la prise en charge du protocole Liberty est obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.

Familiarisez-vous avec la documentation relative aux normes Liberty avant de mettre en oeuvre une fédération de connexion unique. Les normes définissent les échanges de données et le traitement de messages. Vous devez déterminer les informations que vous devez fournir à vos partenaires, ainsi que les informations que votre partenaire doit vous remettre.

Liberty Alliance

`http://www.projectliberty.org`

L'assistant de fédération vous invite à indiquer des valeurs pour un certain nombre de propriétés. La plupart d'entre elles peuvent être modifiées ultérieurement, après la création de la fédération.

Le choix du ou des profils à utiliser dépend à la fois des décisions en matière de règles commerciales et de l'architecture du réseau du point de vue de la sécurité. Les partenaires de la fédération doivent se mettre d'accord sur le choix des profils afin d'activer la connexion unique sur l'ensemble de la fédération. Ce choix doit être fait avant la configuration de la fédération.

La norme Liberty prend en charge une gamme unique de profils de connexion unique. Les profils s'étendent au-delà des spécifications, pour permettre une connexion unique fédérée, et peuvent inclure d'autres fonctions, comme la déconnexion unique, la notification de résiliation de la fédération ou l'identification du nom de registre.

Rôles du fournisseur d'identité et du fournisseur de services

Au sein d'une fédération, chaque partenaire a un rôle. Il s'agit du rôle **Fournisseur d'identité** ou **Fournisseur de services**. La présente section contient les descriptions des deux rôles.

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

- Fournisseur d'identité

  Le fournisseur d'identité est un partenaire de fédération qui garantit l'identité des utilisateurs. Il authentifie un utilisateur et transmet un jeton d'authentification au fournisseur de services.

  Le fournisseur d'identité authentifie directement l'utilisateur en accomplissant l'une des tâches suivantes :

  – en validant un nom d'utilisateur et un mot de passe,
  – en authentifiant indirectement l'utilisateur,
  – en validant une assertion concernant l'identité de l'utilisateur, telle que représentée par un autre fournisseur d'identité.

Le fournisseur d'identité traite également la gestion des identités utilisateur afin de dégager le fournisseur de services de cette responsabilité.

- Fournisseur de services

  Un fournisseur de services est un partenaire de fédération qui propose des services à l'utilisateur. En général, il n'authentifie pas les utilisateurs, mais demande à un fournisseur d'identité de prendre les décisions liées à l'authentification. Les fournisseurs de services comptent sur les fournisseurs d'identité pour affirmer l'identité d'un utilisateur et pour gérer les identités des utilisateurs pour la fédération.

  Les fournisseurs de services peuvent gérer un compte local pour l'utilisateur, compte qui peut être désigné par un identificateur.

Profils de connexion unique Liberty

Liberty prend en charge plusieurs profils de connexion unique. Vous devez sélectionner au moins un profil. Gardez toutefois à l'esprit, que la prise en charge du protocole Liberty est obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.

Vous avez également la possibilité de configurer les profils Artefact du navigateur (Browser Artifact) et POST du navigateur (Browser POST) lors de la configuration d'un fournisseur d'identité. Vous ne pouvez sélectionner qu'un seul profil lors de la configuration d'un fournisseur de services.

Artefact du navigateur (Browser Artifact)
L'artefact du navigateur utilise un canal de retour SOAP pour échanger un artefact au cours de l'établissement et de l'utilisation d'une session sécurisée entre un fournisseur d'identité, un fournisseur de services et un client (navigateur).

Vous avez également la possibilité de configurer le profil Artefact du navigateur (Browser Artifact) lors de la configuration d'un fournisseur d'identité ou d'un fournisseur de services.

Lorsque vous sélectionnez le profil Artefact du navigateur, entrez le nom d'une clé de chiffrement pour la session sécurisée. Spécifiez une clé même si vous choisissez de rendre facultative la signature d'assertions pour les autres communications par message Liberty.

POST du navigateur (Browser POST)
Le profil du navigateur POST utilise un formulaire qui renvoie l'action à lui-même (self-posting form) pour échanger un artefact au cours de l'établissement et de l'utilisation d'une session sécurisée entre un fournisseur d'identité, un fournisseur de services et un client (navigateur).

Vous avez également la possibilité de configurer le profil POST du navigateur lors de la configuration d'un fournisseur d'identité ou d'un fournisseur de services.

Remarque : Lors de la configuration d'un fournisseur d'identité, vous pouvez sélectionner à la fois le profil Artefact du navigateur (Browser Artifact) et le profil POST du navigateur (Browser POST). Toutefois, lors de la configuration d'un fournisseur de services, vous ne pouvez sélectionner qu'un seul profil, c'est-à-dire soit l'artefact du navigateur, soir le POST du navigateur.

Profil de connexion unique LECP (Liberty-enabled client/proxy)
Un client activé pour Liberty (Liberty Enabled Client) ou un proxy activé pour Liberty (Liberty-Enabled Proxy) a ou sait comment obtenir les informations

requises pour se connecter au fournisseur d'identité que l'utilisateur (principal) veut utiliser avec ce fournisseur de services. Un proxy activé pour Liberty est un proxy HTTP, par exemple une passerelle WAP (Wireless Application Protocol), qui émule un client activé pour Liberty.

Fournisseurs LECP
Liste séparée par des virgules de variables d'en-tête utilisées par LECP. Cette propriété se définit lors de la configuration des fournisseurs d'identité et des fournisseurs de services. Il n'existe pas de valeur par défaut.

Exemple de variable d'en-tête unique :

```
ibm_msisdn
```

Exemple de variables d'en-tête multiples :

```
ibm_msisdn,x_msisdn
```

Identificateur RNI (Register Name Identifie) pour Liberty

Ce profil met à jour l'identificateur d'un utilisateur ou principal. Gardez toutefois à l'esprit, que la prise en charge du protocole Liberty est obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.

Liberty requiert que les fournisseurs d'identité et les fournisseurs de services échangent un alias (ou identificateur) pour chaque compte utilisateur. L'échange d'alias est effectué au lieu d'échanger le nom de compte réel de l'utilisateur. L'échange d'alias permet de relier des comptes entre eux tout en masquant les noms de compte utilisateur.

La configuration du profil RNI est facultative.

Lorsqu'il est sélectionné, l'administrateur doit sélectionner les liaisons de communication à utiliser entre les fournisseurs. Ces liaisons peuvent être définies séparément pour le fournisseur d'identité et pour le fournisseur de services. Les liaisons prises en charge sont les suivantes :

- Réacheminement HTTP

  Le fournisseur d'identité et le fournisseur de services communiquent en envoyant des requêtes HTTP 302 au navigateur. Les identificateurs sont mis à jour l'un après l'autre à l'aide des réacheminements. Le réacheminement HTTP est la liaison par défaut pour les fournisseurs d'identité et de services.
- SOAP/HTTP

  Les identificateurs sont mis à jour via des échanges directs entre les fournisseurs via une connexion SOAP.

Les noeuds finals sont les suivants :

URL de service RNI
Le noeud final d'URL est utilisé pour les protocoles RNI fondés sur l'agent utilisateur. Une valeur par défaut est fournie. Par exemple :

```
https://idp.exemple.com/FIM/sps/libertyfed/liberty/rni
```

URL de retour RNI
Le noeud final d'URL est utilisé à des fins de réacheminement une fois que l'enregistrement du nom HTTP a été effectué. Une valeur par défaut est fournie. Par exemple :

```
https://idp.exemple.com/FIM/sps/libertyfed/liberty/rnireturn
```

Cette valeur est requise pour RNI lors de l'utilisation des communications avec réacheminement HTTP. Elle n'est pas requise pour les communications SOAP/HTTP.

Notification FTN (Federation Termination Notification) pour Liberty

Ce profil met fin aux liaisons associées au compte d'un utilisateur dans la fédération. Par défaut, il est désactivé.

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

La configuration de ce profil est facultative. Lorsque ce profil est sélectionné, vous devez sélectionner les liaisons de communication à utiliser entre les fournisseurs. Ces liaisons peuvent être définies séparément pour le fournisseur d'identité et pour le fournisseur de services. Les liaisons prises en charge sont les suivantes :

- Réacheminement HTTP

  Le fournisseur d'identité et le fournisseur de services communiquent en envoyant des requêtes HTTP 302 au navigateur. Les associations de comptes à la fédération sont résiliées l'une après l'autre à l'aide des réacheminements. Le réacheminement HTTP est la liaison par défaut pour les fournisseurs d'identité et de services.
- SOAP/HTTP

  Les associations de comptes à la fédération sont résiliées l'une après l'autre à l'aide des échanges directs entre les fournisseurs via une connexion SOAP.

Les noeuds finals sont les suivants :

URL de service FTN
URL, sur le fournisseur, vers laquelle les processus FTN uniques sont envoyés. Une valeur par défaut est fournie. Par exemple :

```
https://idp.exemple.com/FIM/sps/libertyfed/liberty/ftn
```

URL de retour FTN
Adresse URL utilisée par le fournisseur d'identité ou le fournisseur de services lors du réacheminement de l'agent utilisateur à la fin du processus de notification de résiliation de la fédération basé sur l'agent utilisateur.

```
https://idp.exemple.com/FIM/sps/libertyfed/liberty/ftnreturn
```

FTN requiert cette valeur lors de l'utilisation de la communication par réacheminement HTTP.

Fermeture de session unique Liberty

Ce profil met fin à toutes les sessions de connexion associées à un utilisateur au sein de la fédération. Par défaut, il est désactivé.

Remarque : La prise en charge du protocole Liberty sera obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.

La configuration de ce profil est facultative. Lorsqu'il est sélectionné, l'administrateur doit sélectionner les liaisons de communication à utiliser entre les fournisseurs. Ces liaisons peuvent être définies séparément pour le fournisseur d'identité et pour le fournisseur de services. Les liaisons prises en charge sont les suivantes :

- Réacheminement HTTP

  Le fournisseur d'identité et le fournisseur de services communiquent en envoyant des requêtes HTTP 302 au navigateur. Les sessions utilisateur sont déconnectées l'une après l'autre à l'aide des réacheminements. Le réacheminement HTTP est la liaison par défaut pour les fournisseurs d'identité et de services.
- HTTP GET

  Les fournisseurs d'identité peuvent, à l'aide des balises Image, obliger le navigateur à utiliser HTTP GET pour communiquer les demandes de déconnexion aux fournisseurs de services. Ces demandes de déconnexion sont traitées simultanément et non pas l'une après l'autre. En cas d'échec d'une demande de déconnexion, les autres demandes ne sont pas affectées et sont envoyées au fournisseur de services approprié. A l'inverse, lorsque des demandes de déconnexion sont traitées l'une après l'autre (en série, avec des réacheminements HTTP), l'échec d'une demande de déconnexion entraîne l'annulation des demandes restantes.

  Remarque : Cette option n'est spécifiée que sur les fournisseurs d'identité. Les fournisseurs de services ne peuvent pas la définir.
- SOAP/HTTP

  La déconnexion des sessions utilisateur s'effectue par des échanges directs entre les fournisseurs via une connexion SOAP.

Les noeuds finals sont les suivants :

URL de service SLO
URL à laquelle le fournisseur de services envoie les demandes de déconnexion d'utilisateurs. Une valeur par défaut est fournie. Par exemple :

```
https://idp.exemple.com/FIM/sps/libertyfed/liberty/slo
```

URL de retour SLO
URL utilisée par le fournisseur de services lors du réacheminement de l'agent utilisateur vers le fournisseur d'identité à la fin du processus du profil de déconnexion unique (SLO). Une valeur par défaut est fournie. Par exemple :

```
https://idp.exemple.com/FIM/sps/libertyfed/liberty/sloreturn
```

Cette valeur est requise pour SLO lors de l'utilisation des communications avec réacheminement HTTP.

Présentation du fournisseur d'identité Liberty

Le profil IPI (Identity Provider Introduction, ou présentation du fournisseur d'identité) permet à un fournisseur de services de connaître les fournisseurs d'identité utilisés par un utilisateur (principal).

La prise en charge du protocole Liberty sera obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.

Ce profil se fonde sur un cookie écrit dans un domaine commun aux fournisseurs d'identité et aux fournisseurs de services dans un réseau de fédération d'identité.

Ce profil n'est configuré que sur un fournisseur d'identité.

Domaine DNS commun

Le domaine DNS commun est un domaine virtuel dans lequel un composant est configuré pour définir ou extraire un cookie. L'utilisation de ce domaine commun permet aux fournisseurs d'identité et aux fournisseurs de services, qui se trouvent généralement dans des domaines séparés, d'accéder à un cookie. Cette propriété de configuration peut être définie avant même l'existence du domaine. Vous devrez toutefois créer ce domaine pour qu'un utilisateur puisse effectuer une connexion unique en s'appuyant sur le profil IPI. Cette propriété n'est définie que lors de la configuration d'un fournisseur d'identité. Il n'existe pas de valeur par défaut. Par exemple :

```
cot.projectliberty.org
```

La configuration IPI exige que vous entrez une valeur pour cette zone.

Nom d'hôte de domaine commun

Nom d'un système hôte dans le domaine DNS commun. Cet hôte reçoit les demandes de définition ou de lecture du cookie du domaine commun utilisé par le profil IPI. Cette propriété n'est définie que lors de la configuration d'un fournisseur d'identité. Il n'existe pas de valeur par défaut. Par exemple :

```
idp.cot.projectliberty.org
```

La partie nom de domaine de ce nom d'hôte doit correspondre à la valeur définie dans le domaine DNS commun. Par exemple, la valeur de ce système hôte doit comprendre `cot.projectliberty.org`.

La configuration IPI exige que vous entrez une valeur pour cette zone.

Sécurité des messages Liberty

L'assistant de création de fédération vous demande si vous souhaitez signer les messages Liberty. Si vous choisissez de signer les messages Liberty, vous devez indiquer une clé ou un certificat à utiliser.

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Options de signature numérique

Dans certains cas, vous devez néanmoins entrer une clé ou un certificat si vous ne sélectionnez pas **Signer les messages Liberty**. Par exemple :

- Vous devez indiquer une clé à utiliser pour signer les messages envoyés dans l'ensemble du canal de retour.
- Vous devez spécifier une clé ou un certificat lorsque vous sélectionnez un des profils facultatifs, et spécifier la communication SOAP initiée par le fournisseur de services.

Fournissez les informations de configuration suivantes si vous devez entrer une clé ou un certificat :

Nom du fichier de clés
L'assistant présente une sélection des fichiers de clés que vous avez configurés avant de commencer la configuration de la fédération de connexion unique.

Mot de passe du fichier de clés
Vous devez indiquer le mot de passe utilisé pour accéder au fichier de clés.

Nom de la clé
Vous devez spécifier la clé à utiliser.

Propriétés des communications Liberty

Vous devez savoir comment remplir les propriétés des communications Liberty pour votre fournisseur d'identité et votre fournisseur de services. Gardez toutefois à l'esprit, que la prise en charge du protocole Liberty est obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.

Durée maximale des messages Liberty

Nombre entier correspondant à la durée de validité, en secondes, d'un message Liberty. Cette propriété est définie sur le fournisseur d'identité et sur le fournisseur de services.

Valeur minimale : 60 secondes

Valeur maximale : pas de valeur maximale autre que celle, sous forme de nombre entier, prise en charge par le type de données.

Valeur par défaut : 60 secondes

Durée des artefacts Liberty

Nombre entier correspondant à la durée, en secondes, pendant laquelle un fournisseur de services peut extraire une assertion d'un fournisseur d'identité. Le fournisseur de services utilise un artefact pour extraire cette assertion. Le fournisseur d'identité conserve le mappage entre l'artefact et l'assertion en cache pendant la durée définie. Si le fournisseur de services ne récupère pas l'artefact au terme de cette durée, l'artefact est purgé du cache et la connexion du fournisseur de services échoue.

Cette propriété ne se définit que lors de la configuration d'un fournisseur d'identité à l'aide du profil de connexion unique Artefact du navigateur (Browser Artifact).

Remarque : Cette valeur n'est pas utilisée avec le profil POST du navigateur (Browser POST).

Valeur minimale : 120 secondes

Valeur par défaut : 120 secondes.

Nécessite un accord pour fédérer

Détermine si le fournisseur d'identité demande à l'utilisateur s'il accepte de rejoindre la fédération. Cette propriété n'est définie que sur le fournisseur d'identité. Ce message s'affiche lors de la fédération du compte utilisateur. Par défaut, cette option est désactivée. Cochez la case pour activer l'émission de l'invite.

SOAP, noeud final

Emplacement du noeud final SOAP (Simple Object Access Protocol) sur le fournisseur de services ou le fournisseur d'identité auquel sont envoyés les messages Liberty SOAP.

Ce paramètre est requis lorsqu'au moins l'une des deux conditions suivantes se vérifie :

- Le profil de connexion unique Artefact du navigateur (Browser Artifact) est sélectionné dans la fenêtre des profils Liberty.
- Un ou plusieurs profils Liberty facultatifs sont sélectionnés, de même que la communication SOAP/HTTP initialisée par au moins un des fournisseurs de services.

Par exemple :

`https://idp.exemple.com/FIM/sps/libertyfed/liberty/soap`

La connexion unique est passive (pas d'interaction entre le fournisseur d'identité et l'utilisateur)

Détermine s'il peut ou non y avoir une interaction entre le fournisseur d'identité et le principal (l'utilisateur) et si ce fournisseur peut prendre le contrôle de l'interface utilisateur à la place du fournisseur d'identité. Cette propriété n'est définie que lors de la configuration d'un fournisseur de services. Cochez cette case pour activer cette option. Valeur par défaut : désactivée.

Forcer le fournisseur d'identité à authentifier l'utilisateur

Détermine si le fournisseur d'identité doit authentifier un utilisateur (principal), que ce dernier soit ou non déjà authentifié. Cette valeur ne se définit que lorsque la case **La connexion unique est passive (pas d'interaction entre le fournisseur d'identité et l'utilisateur)** est désactivée. En outre, elle ne se définit que lors de la configuration d'un fournisseur de services.

Si ce paramètre n'est pas sélectionné, le fournisseur d'identité ne doit authentifier l'utilisateur (principal) que si ce dernier ne l'a pas encore été.

- Cochez la case **Forcer le fournisseur d'identité à authentifier l'utilisateur** pour activer cette option.
- Dans le cas contraire, désactivez-la.

Modules de jetons Liberty

Lorsque vous créez une fédération de connexion unique, vous devez configurer une instance de module de jeton de sécurité pour cette fédération. Gardez toutefois à l'esprit, que la prise en charge du protocole Liberty est obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.

Le module de jeton correspond à un type de jeton de sécurité qui définit le format du jeton chiffré contenant les données d'identification des utilisateurs.

Le jeton est échangé entre le fournisseur d'identité et le fournisseur de services dans le cadre des services d'authentification et d'autorisation pour le traitement de chaque requête d'accès d'utilisateur.

Lorsque vous utilisez l'assistant de création de fédération, un type de jeton est automatiquement sélectionné en fonction du choix du protocole de connexion unique.

La configuration du module de jeton Liberty est requise uniquement par le fournisseur d'identité. Aucune configuration n'est nécessaire lors du déploiement d'un fournisseur de services.

La propriété de configuration est identique pour les jetons Liberty v1.1 et les jetons Liberty v1.2.

Durée de validité (en secondes) de la vérification après émission

Nombre entier correspondant à la durée de validité, en secondes, de l'assertion. Cette valeur d'entier est spécifiée pour les jetons Liberty. La valeur minimale est 120 secondes. La valeur maximale est 300 secondes.

Mappage d'identité Liberty

L'assistant de création de fédération vous invite à spécifier soit l'authentification par fichier de règle de mappage XSLT, soit via une instance de module de mappage personnalisée. Gardez toutefois à l'esprit, que la prise en charge du protocole Liberty est obsolète dans les versions ultérieures d'IBM Tivoli Federated Identity Manager.


Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Le fichier de mappage XSLT ou l'instance de module de mappage personnalisée doivent être préparés avant la configuration de la fédération.

Transformation XSL pour le mappage d'identité
Sélectionnez ce bouton sur l'assistant si vous pouvez fournir un fichier XSL contenant le mappage d'identité. Entrez le nom d'un fichier du système de fichiers local.

Instance de module de mappage personnalisée
Sélectionnez ce bouton sur l'assistant si vous pouvez fournir une instance de module de mappage personnalisée que vous utiliserez à la place du fichier XSL. Vous serez invité à entrer les propriétés de configuration requises par l'instance du module de mappage personnalisée.

Mappage de données d'identification Tivoli Access Manager vers un jeton Liberty ou SAML 2

Ce scénario se produit lors de l'échange de messages entre des partenaires d'une fédération de connexion unique Liberty ou SAML 2, et les informations liées à l'identité de l'utilisateur sont gérées par Tivoli Access Manager

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.


Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Lorsqu'une demande d'utilisateur est reçue (par exemple, pour accéder à une ressource distante), le service d'accréditation prend contact avec Tivoli Access Manager et obtient des données Tivoli Access Manager relatives à l'identité de l'utilisateur.

Dans ce scénario, le module de données d'identification Tivoli Access Manager du service d'accréditation fonctionne en mode de validation. Dans ce mode, il convertit les données d'identification Tivoli Access Manager en un document d'utilisateur universel STS d'entrée (In-STSUUSER). Le document In-STSUUSER créé à partir du module d'accréditation Tivoli Access Manager contient toutes les

informations issues des droits d'accès. Ces informations peuvent éventuellement être utilisées par le module de service d'accréditation qui génère le jeton sortant.

Le service d'accréditation consulte son entrée de configuration correspondant au partenaire de la fédération (par exemple, la destination qui héberge une ressource demandée). La configuration indique le type de jeton à créer.

Le module de mappage d'identité convertit ensuite l'élément In-STSUUSER en un utilisateur universel STS de sortie (Out-STSUUSER). L'élément Out-STSUUSER doit contenir les informations requises par le module de jeton Liberty (ou SAML 2) Tivoli Federated Identity Manager pour générer un jeton Liberty (ou SAML 2).

L'élément Out-STSUUSER doit contenir les informations suivantes qui permettent au module de jeton de générer un jeton valide :

Tableau 119. Entrées Out-STSUUSER servant à générer un jeton Liberty ou SAML 2

Elément Out-STSUUSER	**Informations de jeton**	**Obligatoire/ Facultatif**
Principal Attr: Name	Nom à transmettre au service d'alias	Obligatoire
Attribute: AuthenticationMethod	Méthode d'authentification. N **Remarque :** Cet élément est toujours paramétré sur "password" (nom d'utilisateur/mot de passe) quel que soit le mécanisme d'authentification défini dans les données d'identification de Tivoli Access Manager.	Obligatoire
Liste des attributs	Attributs personnalisés supplémentaires	Facultatif

Le module de mappage est responsable des opérations suivantes :

1. Mappage de l'élément Principal Attr Name dans In-STSUUSER vers une entrée de nom Principal dans Out-STSUUSER.

   Remarque : Lorsque le module de jeton génère le jeton, ce nom de Principal n'est pas utilisé directement. En revanche, la valeur de la zone Name est envoyée en entrée du service d'alias de Tivoli Federated Identity Manager. Le service d'alias obtient l'alias (identificateur de nom) pour le principal et place l'alias renvoyé dans le module de jeton généré.

   Pour obtenir un exemple de fichier de règles de mappage extrait du fichier de mappage d'application de démonstration, ip_liberty.xsl, voir la figure 53, à la page 505. Non

   Remarque : Les jetons Liberty sont des extensions des jetons SAML. Par conséquent, les commentaires de l'exemple de code qui font référence aux jetons SAML sont corrects dans ce contexte.

```
</xsl:template>
<!-- Ce modèle remplace l'intégralité de l'élément Principal par un élément qui ne
     que l'adresse électronique (à partir de ivcred tagvalue_email) et le
     type de données approprié pour SAML. -->
 <xsl:template match="//stsuuser:Principal">
  <stsuuser:Principal>
   <stsuuser:Attribute name="name"
                type="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
    <stsuuser:Value>
    <xsl:value-of
    select="//stsuuser:AttributeList/stsuuser:Attribute[@name='tagvalue_email']
          [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>
  </stsuuser:Principal>
 </xsl:template><!--
```

Figure 53. Exemple de code XSL présentant le mappage d'une valeur des données d'identification Tivoli Access Manager vers un nom Principal pour un jeton Liberty

2. Paramétrage de la méthode d'authentification sur "password" (mot de passe), quelle que soit la valeur obtenue des données d'identification Tivoli Access Manager. Cette action est requise par le module de jeton.

   Pour obtenir un exemple de fichier de règles de mappage extrait du fichier de mappage d'application de démonstration, ip_liberty.xsl, voir figure 54.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
  <!-- Tout d'abord l'attribut de la méthode d'authentification -->
   <stsuuser:Attribute name="AuthenticationMethod"
                           type="urn:oasis:names:tc:SAML:1.0:assertion">
    <stsuuser:Value>urn:oasis:names:tc:SAML:1.0:am:password</stsuuser:Value>
   </stsuuser:Attribute>
          ....
       </stsuuser:AttributeList>
</xsl:template>
```

Figure 54. Exemple de code XSL présentant l'affectation d'une méthode d'authentification sous forme d'attribut pour un jeton Liberty

3. Remplissage de l'instruction d'attribut de la vérification à l'aide des attributs de l'élément AttributeList dans In-STSUUSER. Ces informations deviennent des informations personnalisées du jeton.

   Des attributs personnalisés peuvent être requis par les applications qui utilisent les informations transmises entre les partenaires d'une fédération.

   Pour obtenir un exemple de fichier de règles de mappage extrait du fichier de mappage d'application de démonstration, ip_liberty.xsl, voir la figure 55, à la page 506.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
         ....
      <!-- Puis l'attribut commonName -->
   <stsuuser:Attribute name="commonName"
                        type="http://exemple.com/federation/v1/commonName">
    <stsuuser:Value>
     <xsl:value-of
     select="//stsuuser:AttributeList/stsuuser:Attribute[@name='tagvalue_name']
           [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>
   <!-- Puis l'attribut ssn -->
   <stsuuser:Attribute name="ssn" type="http://exemple.com/federation/v1/ssn">
    <stsuuser:Value>
     <xsl:value-of
     select="//stsuuser:AttributeList/stsuuser:Attribute[@name='tagvalue_ssn']
             [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>
              ....
     </stsuuser:AttributeList>
 </xsl:template>
```

Figure 55. Exemple de code XSL présentant l'affectation d'attributs facultatifs pour un jeton Liberty

4. L'élément GroupList de In-STSUUSER n'est pas lu par le module de jeton. Cependant, les informations contenues dans cet élément peuvent, le cas échéant, servir à remplir les attributs personnalisés de Out-STSUUSER.

   La figure 56 présente l'affectation facultative d'une valeur GroupList à un attribut. Cet exemple de code est issu du fichier de mappage d'application de démonstration, ip_liberty.xsl.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
         ....
      <!-- Puis l'attribut role (peut être multiforme) -->
   <stsuuser:Attribute name="role"
                        type="http://exemple.com/federation/v1/role">
    <xsl:for-each select="//stsuuser:GroupList/stsuuser:Group">
     <stsuuser:Value>
      <xsl:value-of select="@name" />
     </stsuuser:Value>
    </xsl:for-each>
   </stsuuser:Attribute>
              ....
     </stsuuser:AttributeList>
 </xsl:template>
```

Figure 56. Exemple de code XSL présentant l'affectation facultative d'une valeur GroupList à un attribut d'un jeton Liberty

Mappage d'un jeton Liberty ou SAML 2 vers des données d'identification Tivoli Access Manager

Mappez un jeton Liberty ou SAML 2.0 vers des données d'identification Tivoli Access Manager pour un scénario de fédération de connexion unique.

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Le fournisseur de services reçoit un jeton Liberty ou SAML 2. Le module de jeton, en mode de validation, crée un document In-STSUUSER à partir du jeton. Le tableau 120 affiche les informations issues du jeton qui sont converties en un document In-STSUUSER.

Tableau 120. Informations de jeton converties en document d'utilisateur universel STS

Informations de jeton	**Elément In-STSUUSER**	**Obligatoire dans Out-STSUUSER ?**
ID utilisateur obtenu à partir du service d'alias	Principal Attr: Name	Obligatoire
Attributs personnalisés supplémentaires	Liste des attributs	Facultatif

Il est à noter que le module de jeton ne remplit pas l'élément GroupList du document In-STSUUSER.

Le module de jeton lit le jeton et extrait l'élément NameIdentifier. Le module de jeton transmet l'élément NameIdentifier (alias) au service d'alias. Ce dernier convertit l'alias reçu en ID utilisateur Tivoli Access Manager local. Le module de jeton place l'ID utilisateur dans l'élément Principal du document In-STSUUSER.

Le service d'accréditation doit convertir ces informations en données d'identification Tivoli Access Manager de manière à prendre une décision d'autorisation sur la demande de l'identité utilisateur.

- L'alias NameIdentifier renvoyé sert à remplir l'attribut `name` de l'élément Principal. Il s'agit de l'ID utilisateur local.

  Pour obtenir un exemple d'affectation d'une valeur définie pour le nom de Principal, voir la figure 57. Cet exemple de code est issu du fichier de mappage d'application de démonstration, sp_liberty.xsl.

```
<!-- Cela remplace le nom de Principal (correspondant à l'adresse électronique
     dans l'assertion SAML) par l'utilisateur "me_guest". -->
   <xsl:template match="//stsuuser:Principal/stsuuser:Attribute[@name='name']">
           <stsuuser:Attribute name="name"
                               type="urn:ibm:names:ITFIM:5.1:accessmanager">
            <stsuuser:Value>
<xsl:value-of
 select="//stsuuser:Principal/stsuuser:Attribute[@name='name']/stsuuser:Value" />
            </stsuuser:Value>
          </stsuuser:Attribute>
       </xsl:template>
```

Figure 57. Exemple de code XSL présentant l'affectation d'une valeur pour le nom de Principal d'un jeton Liberty

- D'autres informations issues du jeton servent à remplir la zone Attributes de l'élément AttributeList.

Pour obtenir un exemple d'affectation facultative de valeurs supplémentaires aux attributs, voir la figure 58. Cet exemple de code est issu du fichier de mappage d'application de démonstration, sp_liberty.xsl.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
          ....
               ....
      <!-- Attribut tagvalue_sso -->
      <stsuuser:Attribute name="tagvalue_sso"
                          type="urn:ibm:names:ITFIM:5.1:accessmanager">
                    <stsuuser:Value>isSingleSignOn</stsuuser:Value>
      </stsuuser:Attribute>
      <!-- Attribut tagvalue_fedname -->
      <stsuuser:Attribute name="tagvalue_fedname"
                         type="urn:ibm:names:ITFIM:5.1:accessmanager">
                     <stsuuser:Value>libertyfed</stsuuser:Value>
      </stsuuser:Attribute>
               ....
               ....
     </stsuuser:AttributeList>
 </xsl:template>
```

Figure 58. Exemple de code XSL présentant l'affectation facultative d'attributs pour un jeton Liberty

Service d'alias Liberty

Les normes Liberty relatives aux protocoles de connexion unique exigent l'utilisation d'alias lorsqu'une identité d'utilisateur est envoyée dans un message échangé entre les partenaires d'une fédération de connexion unique. Les alias sont requis pour mieux garantir la confidentialité de l'utilisateur final lors de l'accès aux ressources d'un fournisseur de services.

Dans les spécifications, les alias sont appelés *identificateurs de nom*. L'identificateur du nom de tel ou tel utilisateur est enregistré lors de la fédération des comptes (liaison des comptes), puis utilisé dans tous les messages transmis entre les partenaires. Les alias sont générés au hasard et ne contiennent aucune information importante sur l'identité des utilisateurs.

Un identificateur de nom différent doit être affecté à chaque utilisateur pour les communications avec chaque partenaire. Le cas échéant, plusieurs identificateurs de nom peuvent être créés en fonction de chaque destination des messages. Par conséquent, l'alias d'un utilisateur varie selon que le fournisseur d'identité contacte le fournisseur de services ou inversement.

Le service d'alias fourni par Tivoli Federated Identity Manager gère les tâches de gestion des alias. Il rend la plupart des tâches de génération et d'échange d'alias invisibles à l'administrateur de fédération. Il assure les fonctions suivantes :

- Génération et association des alias aux utilisateurs locaux
- Recherche de l'identité d'un utilisateur local lorsqu'un alias provient d'un partenaire
- Recherche de l'alias d'un utilisateur local lorsque le fournisseur doit envoyer un message à un partenaire

Le service d'alias Tivoli Federated Identity Manager stocke les informations relatives aux alias dans un registre d'utilisateurs. Le service d'alias prend en charge les registres d'utilisateurs suivants :

- IBM Tivoli Directory Server
- Sun ONE

Pour chacun de ces serveurs LDAP, vous serez amené à définir certains paramètres de configuration après avoir créé la fédération Liberty.

Le service d'alias ne prend pas en charge les registres d'utilisateurs Lotus Domino ou Microsoft Active Directory. Vous pouvez écrire votre propre service d'alias pour l'utiliser avec ces registres.

Chapitre 31. Configuration d'une fédération Liberty

Pour configurer une fédération Liberty, vous devez créer la fédération, lui ajouter un partenaire et fournir les informations de configuration de la fédération à ce partenaire.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Exécutez les tâches suivantes :

Procédure

1. «Création d'un fournisseur d'identité Liberty»
2. «Configuration d'un fournisseur de services Liberty», à la page 514
3. «Configuration d'un serveur point de contact WebSEAL pour la fédération Liberty», à la page 516
4. «Propriétés des propriétés de fédération Liberty», à la page 518
5. «Exportation des informations d'authentification de noeud final SOAP vers un partenaire de fédération Liberty», à la page 518
6. «Obtention des métadonnées auprès d'un partenaire de fédération Liberty», à la page 519
7. «Importation des informations d'authentification de noeud final SOAP à partir d'un partenaire de fédération Liberty», à la page 520
8. «Ajout d'un partenaire dans une fédération Liberty», à la page 522

Création d'un fournisseur d'identité Liberty

Créez un fournisseur d'identité Liberty pour authentifier directement ou indirectement les utilisateurs lorsqu'ils font appel au fournisseur de services.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Pour créer une fédération de fournisseurs d'identité Liberty, procédez comme suit :

Procédure

1. Connectez-vous à la console de gestion.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Les portlets Domaine en cours et Fédérations s'ouvrent.
3. Cliquez sur **Créer**. L'assistant de fédération démarre. Le panneau Informations générales s'ouvre.
4. Indiquez le nom de la fédération et sélectionnez un rôle.
5. Cliquez sur **Suivant**.
6. Entrez les coordonnées du contact.

7. Cliquez sur **Suivant**.
8. Sélectionnez le protocole Liberty 1.1 ou Liberty 1.2.
9. Cliquez sur **Suivant**. Le panneau Serveur point de contact s'ouvre.
10. Entrez les informations sur l'adresse du point de contact et cliquez sur **Suivant**.
11. Indiquez les profils à utiliser avec cette fédération.
    a. Sélectionnez au moins un des profils de connexion unique Liberty.

       Liberty prend en charge trois profils de connexion unique. Vous devez sélectionner au moins un profil. Vous avez également la possibilité de sélectionner les profils Artefact du navigateur (Browser Artifact) et POST du navigateur (Browser POST) lors de la configuration d'un fournisseur d'identité. Vous ne pouvez sélectionner qu'un profil lors de la configuration d'un fournisseur de services.
    b. Sélectionnez n'importe lequel des profils facultatifs à configurer :
       - **Register Name Identifier (RNI)**
       - **Federation Termination Notification (FTN)**
       - **Déconnexion unique**
       - **Identity Provider Introduction**.

         Dans le cas de fournisseurs d'identité uniquement.
12. Une fois que vous avez terminé, cliquez sur **Suivant**. Le panneau Options de signature numérique s'ouvre.
13. Cochez ou décochez la case **Signer les messages Liberty**. Si vous choisissez de signer les messages Liberty, vous devez indiquer une clé ou un certificat à utiliser.

    Dans certains cas, lorsque vous ne cochez pas la case **Signer les messages Liberty**, vous devez néanmoins entrer une clé ou un certificat. Par exemple :
    - Lors de la sélection du profil Artefact du navigateur, vous devez indiquer une clé à utiliser pour signer les messages envoyés dans l'ensemble du canal de retour pour l'artefact.
    - Lorsque vous sélectionnez un des profils facultatifs et que vous indiquez que les communications SOAP doivent être lancées par le fournisseur de services, vous devez spécifier une clé ou un certificat.
14. Sélectionnez un fichier de clés et entrez le mot de passe de ce fichier si vous devez saisir une clé ou un certificat. Cliquez sur **Liste des clés** pour afficher les clés ou les certificats contenus dans le fichier de clés concerné et sélectionnez une clé.
    - Le mot de passe par défaut du fichier de clés par défaut **DefaultKeyStore** est `testonly`.
    - Un exemple de clé est fourni à des fins de test uniquement. N'utilisez pas cette clé dans un environnement de production.
15. Cliquez sur **Suivant**.
16. Configurez les propriétés des données Liberty :
    a. Une valeur par défaut est fournie pour le **noeud final SOAP**. Utilisez cette valeur, sauf en cas de conflit de noeud final sur votre hôte.
    b. Définissez l'option **Durée maximale des messages Liberty**.
    c. Définissez l'option **Durée des artefacts Liberty**.
    d. Cochez ou décochez la case **Demander l'accord de fédération**.
    e. Lorsque le profil LECP a été sélectionné, indiquez des **fournisseurs LECP**.

f. Lorsque Présentation du fournisseur d'identité est sélectionné, entrez les valeurs appropriés dans les zones **Domaine DNS commun** et **Nom d'hôte de domaine commun**.

g. Cliquez sur **Suivant**.

Le panneau panneau de configuration du module de jeton Liberty s'affiche. Son contenu est identique pour les jetons Liberty v1.1 et les jetons Liberty v1.2.

17. Indiquez une valeur dans la zone **Durée de validité (en secondes) de l'assertion après émission**.
18. Cliquez sur **Suivant**.
19. Le panneau Options de mappage d'identité s'ouvre. Sélectionnez l'un des boutons d'option suivants.
    - Utiliser la transformation XSL pour le mappage d'identité

      Indique que vous devez fournir un fichier XSL contenant le mappage d'identité requis.

      a. Lorsque vous sélectionnez cette option et cliquez sur **Suivant**, le panneau Mappage d'identité s'affiche. Dans la zone **Fichier XSLT contenant une règle de mappage d'identité**, entrez le nom d'un fichier du système de fichiers local qui contient la règle de mappage d'identité.

         Il s'agit du fichier que vous avez préparé avant de procéder à cette installation.

         Vous pouvez également localiser le fichier sur le système de fichiers local à l'aide du bouton **Parcourir**.

      b. Cliquez sur **Suivant**.

         Une erreur s'affiche lorsque le fichier est introuvable ou ne contient aucune donnée XSLT (eXtensible Stylesheet Language Transform) valide.

    - Utiliser l'instance de module de mappage personnalisée

      Indique que vous devez fournir une instance de module de mappage personnalisée, que vous utiliserez à la place du fichier XSL.

      a. Lorsque vous sélectionnez l'option **Utiliser l'instance de module de mappage personnalisée**, un tableau des instances de module s'affiche. Cliquez sur le bouton d'option correspondant à l'instance de module à utiliser et cliquez sur **Suivant**.

      b. Le cas échéant, vous serez alors invité à indiquer des valeurs pour les propriétés de l'instance de module de mappage personnalisée. Sinon, le panneau affiche un message indiquant qu'aucune propriété ne doit être configurée pour l'instance de module indiquée.

    Le panneau Récapitulatif s'affiche.
20. Vérifiez que les paramètres de configuration sont corrects.
21. Cliquez sur **Terminer**. Le portlet Création de fédération terminée s'affiche.

Que faire ensuite

Si vous utilisez WebSEAL en tant que serveur point de contact, procédez maintenant à sa configuration. Ne quittez pas la console de gestion. Pour plus détails, voir «Configuration d'un serveur point de contact WebSEAL pour la fédération Liberty», à la page 516.

Configuration d'un fournisseur de services Liberty

Créez un fournisseur de services Liberty pour demander des décisions d'authentification de votre fournisseur d'identité.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Procédure

1. Connectez-vous à la console de gestion.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Les portlets Domaine en cours et Fédérations s'ouvrent.
3. Cliquez sur **Créer**. L'assistant de fédération démarre. Le panneau Informations générales s'ouvre.
4. Indiquez le nom de la fédération et sélectionnez un rôle.
5. Cliquez sur **Suivant**.
6. Entrez les coordonnées du contact.
7. Cliquez sur **Suivant**.
8. Sélectionnez le protocole Liberty 1.1 ou Liberty 1.2.
9. Cliquez sur **Suivant**. Le panneau Serveur point de contact s'ouvre.
10. Entrez les informations sur l'adresse du point de contact et cliquez sur **Suivant**.
11. Indiquez les profils à utiliser avec cette fédération.
    a. Sélectionnez au moins un des profils de connexion unique Liberty.

       Liberty prend en charge trois profils de connexion unique. Vous devez sélectionner au moins un profil. Vous avez également la possibilité de sélectionner les profils Artefact du navigateur (Browser Artifact) et POST du navigateur (Browser POST) lors de la configuration d'un fournisseur d'identité. Vous ne pouvez sélectionner qu'un profil lors de la configuration d'un fournisseur de services.

    b. Sélectionnez n'importe lequel des profils facultatifs à configurer :
       - **Register Name Identifier (RNI)**
       - **Federation Termination Notification (FTN)**
       - **Déconnexion unique**
       - **Identity Provider Introduction**.

         Dans le cas de fournisseurs d'identité uniquement.
12. Cliquez sur **Suivant** lorsque vous avez terminé. Le panneau Options de signature numérique s'ouvre.
13. Cochez ou décochez la case **Signer les messages Liberty**. Si vous choisissez de signer les messages Liberty, vous devez indiquer une clé ou un certificat à utiliser.

    Dans certains cas, lorsque vous ne cochez pas la case Signer les messages Liberty, vous devez néanmoins entrer une clé ou un certificat. Par exemple :
    - Lors de la sélection du profil Artefact du navigateur, vous devez indiquer une clé à utiliser pour signer les messages envoyés dans l'ensemble du canal de retour pour l'artefact.

- Lorsque vous sélectionnez un des profils facultatifs et que vous indiquez que les communications SOAP doivent être lancées par le fournisseur de services, vous devez spécifier une clé ou un certificat.

14. Si vous avez besoin d'entrer une clé ou un certificat, sélectionnez un fichier de clés et tapez le mot de passe qui lui correspond. Cliquez sur **Liste des clés** pour afficher les clés ou les certificats contenus dans le fichier de clés concerné et sélectionnez une clé.
    - Le mot de passe par défaut du fichier de clés par défaut **DefaultKeyStore** est `testonly`.
    - Un exemple de clé est fourni à des fins de test uniquement. N'utilisez pas cette clé dans un environnement de production.
15. Configurez les paramètres du fournisseur de services de profil Liberty :
    a. Si vous avez sélectionné un profil Liberty facultatif (Register Name Identifier, Federation Termination Notification ou Déconnexion unique) et que vous avez choisi SOAP/HTTP comme protocole de communication, vous devez indiquer un noeud final SOAP. Une valeur par défaut est fournie. Vous pouvez valider la valeur par défaut, sauf si vous disposez d'une configuration spécifique qui nécessite un noeud final SOAP différent.
    b. Indiquez une valeur dans la zone **Durée maximale des messages Liberty (en secondes)**.
    c. Sélectionnez ou désélectionnez l'option **La connexion unique est passive (pas d'interaction entre le fournisseur d'identité et l'utilisateur)**.
    d. Sélectionnez ou désélectionnez l'option **Forcer le fournisseur d'identité à authentifier l'utilisateur**.
    e. Si vous avez sélectionné le profil de connexion unique LECP, indiquez un **fournisseur LECP**. Cliquez sur **Suivant**.
16. Cliquez sur **Suivant**.
17. Le panneau Options de mappage d'identité s'ouvre. Sélectionnez l'un des boutons d'option suivants.
    - Utiliser la transformation XSL pour le mappage d'identité

      Indique que vous comptez fournir un fichier XSL contenant le mappage d'identité requis.
      a. Lorsque vous sélectionnez cette option et cliquez sur **Suivant**, le panneau Mappage d'identité s'affiche. Dans la zone **Fichier XSLT contenant une règle de mappage d'identité**, entrez le nom d'un fichier du système de fichiers local qui contient la règle de mappage d'identité.

         Il s'agit du fichier que vous avez préparé avant de procéder à cette installation.

         Vous pouvez également localiser le fichier sur le système de fichiers local à l'aide du bouton **Parcourir**.
      b. Cliquez sur **Suivant**.

         Une erreur s'affiche lorsque le fichier est introuvable ou ne contient aucune donnée XSLT (eXtensible Stylesheet Language Transform) valide.
    - Utiliser l'instance de module de mappage personnalisée

      Indique que vous devez fournir une instance de module de mappage personnalisée, que vous utiliserez à la place du fichier XSL.
      a. Lorsque vous sélectionnez l'option **Utiliser l'instance de module de mappage personnalisée**, un tableau des instances de module s'affiche. Cliquez sur le bouton d'option correspondant à l'instance de module à utiliser et cliquez sur **Suivant**.

b. Le cas échéant, vous serez alors invité à indiquer des valeurs pour les propriétés de l'instance de module de mappage personnalisée. Sinon, le panneau affiche un message indiquant qu'aucune propriété ne doit être configurée pour l'instance de module indiquée.

Le panneau Récapitulatif s'affiche.

18. Vérifiez que les paramètres de configuration sont corrects.
19. Cliquez sur **Terminer**. Le portlet Création de fédération terminée s'affiche.
20. Cliquez sur **Redémarrer WebSphere**.

Que faire ensuite

Si vous utilisez WebSEAL en tant que serveur point de contact, procédez maintenant à sa configuration. Ne quittez pas la console de gestion. Voir aussi :

- «Configuration d'un serveur point de contact WebSEAL pour la fédération Liberty»

Configuration d'un serveur point de contact WebSEAL pour la fédération Liberty

Si vous envisagez d'utiliser WebSEAL en tant que serveur point de contact, vous devez le configurer pour la fédération Liberty.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Ces instructions ont pour hypothèse que le profil du point de contact WebSEAL est activé.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Le portlet Création de fédération terminée comporte un bouton qui vous permet d'obtenir un Tivoli Federated Identity Manager utilitaire de configuration. Vous devez obtenir cet outil, puis l'exécuter.

Pour configurer WebSEAL en tant que serveur point de contact, procédez comme suit :

Procédure

1. Une fois la fédération créée, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour recharger vos modifications.

   Remarque : La console de gestion vous offre la possibilité d'ajouter immédiatement un partenaire, mais pour cette configuration initiale de la fédération, vous devez d'abord exécuter d'autres tâches.

2. Cliquez sur **Terminé** pour revenir au panneau Fédérations.
3. Cliquez sur **Télécharger l'outil de configuration Tivoli Access Manager**.
4. Enregistrez l'outil de configuration sur le système de fichiers de l'ordinateur hébergeant le serveur WebSEAL.
5. Démarrez l'outil de configuration depuis une ligne de commande. La syntaxe est la suivante :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf
   ```

   Remarque : Si la norme FIPS (Federal Information Processing Standards) est activée pour votre environnement, une fabrique de connexions sécurisées doit être indiquée. Par exemple :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf -sslfactory TLS
   ```

   Vous aurez besoin de l'ID (par défaut : sec_master) et du mot de passe de l'utilisateur d'administration Tivoli Access Manager. L'utilitaire configure les noeuds finals sur le serveur WebSEAL, crée une jonction WebSEAL, relie les listes de contrôle d'accès adaptées et active les méthodes d'authentification adéquates.

Exemple

Par exemple, lorsque vous avez mis le fichier tfimcfg.jar dans le répertoire `/tmp` et que le nom de l'instance WebSEAL est `default`, la commande est la suivante :

```
java -jar /tmp/tfimcfg.jar -cfgfile webseald-default -action tamconfig
```

Pour plus d'informations, accédez à l'adresse suivante :

- Annexe A, «Référence de tfimcfg», à la page 827

Que faire ensuite

La prochaine tâche consiste à exporter les propriétés de votre fédération Liberty dans un fichier. Voir «Propriétés des propriétés de fédération Liberty», à la page 518.

Configuration de WebSphere en tant que serveur point de contact

Tivoli Federated Identity Manager est configuré par défaut pour utiliser Tivoli Access Manager WebSEAL en tant que serveur point de contact. Pour configurer WebSphere en tant que serveur point de contact, vous devez procéder à une modification de la configuration.

Procédure

1. Connectez-vous à la console d'administration.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Sélectionnez **WebSphere**
4. Cliquez sur **Activer**.

Résultats

Le serveur WebSphere est désormais configuré en tant que point de contact.

Propriétés des propriétés de fédération Liberty

Lorsque vous souhaitez rejoindre une fédération hébergée par l'un de vos partenaires, vous devez fournir les propriétés de configuration de votrefédération Liberty.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Utilisez la console de gestion pour générer un fichier de métadonnées contenant les propriétés de votre fédération. Transmettez ce fichier à votre partenaire de fédération.

Procédure

1. Connectez-vous à la console de gestion.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**. Le panneau Fédérations s'affiche.
3. Sélectionnez votre fédération Liberty dans le tableau.
4. Cliquez sur **Exporter**. Le navigateur affiche un message vous invitant à sauvegarder le fichier contenant les données exportées.
5. Cliquez sur **OK**. La fenêtre de téléchargement du navigateur vous invite à entrer un emplacement de sauvegarde du fichier.
6. Sélectionnez un répertoire et un nom de fichier. Placez ce fichier dans un endroit facilement accessible.
7. Cliquez sur **Sauvegarder**.

Exportation des informations d'authentification de noeud final SOAP vers un partenaire de fédération Liberty

Fournissez à votre partenaire les clés, certificats, noms d'utilisateur ou mots de passe nécessaires pour établir des communications SSL sur les ports SOAP.

Avant de commencer

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Pourquoi et quand exécuter cette tâche

Remarque : La sécurisation des ports SOAP avec le protocole SSL, des clés, certificats, noms d'utilisateur ou mots de passe n'est pas obligatoire, mais elle est généralement recommandée afin d'optimiser la sécurité des réseaux.

Liberty est doté d'un canal de retour SOAP utilisé avec le profil de connexion unique Artefact du navigateur et peut éventuellement être utilisé avec d'autres profils Liberty qui prennent en charge les liaisons SOAP. Le canal de retour SOAP peut, en option, être protégé à l'aide de communications SSL (via des noeuds finals HTTPS). L'utilisation de SSL est commune à tous les noeuds finals SOAP.

Pour les fédérations Liberty, il se peut que vous ayez également besoin de fournir des informations d'authentification (certificats et informations d'authentification de base) à votre partenaire, à des fins d'accès aux noeuds finals SOAP membres de la fédération Liberty.

Cette tâche est exécutée en dehors de console de gestion.

Remarque : Si votre fédération n'utilise pas le protocole SSL pour sécuriser les ports SOAP, vous pouvez ignorer cette tâche.

Procédure

1. Transmettez à votre partenaire un certificat de validation. Ce certificat valide la communication SSL que le fournisseur de votre fédération envoie au noeud final SOAP du partenaire.
2. Si vous souhaitez que votre partenaire s'authentifie en tant que client, vous devez indiquer s'il est tenu de recourir à une authentification par certificat client ou à une authentification de base. Une seule forme d'authentification peut être spécifiée.

   Authentification par certificat client

   - Lorsque vous exigez une authentification par certificat client, votre partenaire et vous-même devez déterminer quel certificat doit être présenté lors de l'établissement de la session SSL. Le choix d'un certificat est une décision métier. Il peut s'agir soit d'un certificat qui est déjà en possession de votre partenaire, soit d'un certificat que vous lui fournissez.

   Authentification de base

   - Lorsque vous exigez une authentification de base, vous devez fournir un nom d'utilisateur et un mot de passe à votre partenaire afin d'établir une session authentifiée.

Obtention des métadonnées auprès d'un partenaire de fédération Liberty

Si vous souhaitez ajouter un partenaire à votre fédération de connexion unique Liberty, vous devez obtenir auprès de celui-ci les informations de configuration nécessaires relatives à sa fédération Liberty.

Avant de commencer

Une fédération Liberty doit avoir été déjà installée et configurée par votre partenaire. La fédération de votre partenaire joue le rôle opposé à celui de votre fédération. Si, par exemple, votre fédération est configurée en tant que fournisseur d'identité, la fédération de votre partenaire l'est en tant que fournisseur de services.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Procédure

1. Votre partenaire doit exporter les informations de configuration sur la fédération Liberty dans un fichier de métadonnées.

   Le partenaire doit utiliser la console de gestion Tivoli Federated Identity Manager pour exporter les paramètres de configuration vers le fichier de

métadonnées. Il s'agit de la même fonctionnalité que celle utilisée pour fournir vos paramètres de configuration au partenaire.

La fonction d'exportation définit les fichiers de métadonnées d'après une convention de dénomination reposant sur le nom de cette fédération, complété d'un horodatage. L'administrateur peut remplacer le nom par défaut du fichier de métadonnées et définir un nom arbitraire.

2. Votre partenaire doit vous fournir le fichier de métadonnées.

   Cette action a lieu en dehors de console de Tivoli Federated Identity Manager. Votre partenaire doit utiliser tout processus convenu dans le cadre de l'accord d'échange préalablement négocié entre les partenaires.

3. Placez le fichier de métadonnées sur le système de fichiers local dans lequel la configuration de la fédération Liberty est conservée. Vous pouvez indiquer n'importe quel emplacement pour le fichier de métadonnées. Vous avez à présent terminé l'étape préparatoire. Vous utiliserez ultérieurement la console de gestion de Tivoli Federated Identity Manager pour ajouter le partenaire à votre fédération Liberty. La console est dotée d'un assistant d'ajout de partenaire qui vous invite à spécifier le nom du fichier contenant les métadonnées du partenaire. La documentation vous guidera tout au long de cette tâche le moment venu.

Importation des informations d'authentification de noeud final SOAP à partir d'un partenaire de fédération Liberty

Liberty est doté d'un canal de retour SOAP utilisé avec le profil de connexion unique Artefact du navigateur et avec d'autres profils Liberty qui prennent en charge les liaisons SOAP. Le canal de retour SOAP peut, en option, être protégé à l'aide de communications SSL (via des noeuds finals HTTPS). L'utilisation de SSL est commune à tous les noeuds finals SOAP.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Pour les fédérations Liberty, il se peut que vous ayez également besoin d'obtenir des informations d'authentification (certificats et informations d'authentification de base) de la part de votre partenaire. Utilisez les informations d'authentification lorsque vous souhaitez accéder aux noeuds finals SOAP sur la fédération de votre partenaire.

Cette tâche est exécutée en dehors de console de gestion.

Remarque : Si la fédération de votre partenaire n'utilise pas le protocole SSL pour sécuriser les ports SOAP, vous pouvez ignorer cette tâche.

Procédure

1. Demandez à votre partenaire le certificat de validation. Utilisez ce dernier pour valider les communications SSL ou les messages que votre partenaire envoie sur votre noeud final SOAP.
2. Demandez à votre partenaire les exigences d'authentification. Utilisez-les pour l'authentification lorsque votre client contacte le noeud final SOAP de votre partenaire.

Si votre partenaire souhaite voir votre client s'authentifier, il doit vous indiquer si vous devez recourir à une authentification par certification client ou à une authentification de base. Une seule forme d'authentification peut être spécifiée.

Authentification par certificat client

- Lorsque votre partenaire exige une authentification de certificat client, le partenaire et vous-même devez déterminer quel certificat doit être présenté lors de l'établissement de la session SSL. Le choix d'un certificat est une décision métier. Il peut s'agir soit d'un certificat que vous possédez déjà, soit d'un certificat que vous fournit votre partenaire.

Authentification de base

- Lorsque le partenaire exige une authentification de base, il doit vous fournir le nom d'utilisateur et le mot de passe à présenter afin d'établir une session authentifiée.

3. Lorsque votre partenaire utilise des communications SSL destinées à des ports SOAP, vous devez importer le certificat obtenu auprès de votre partenaire. Vous pouvez importer le certificat dans n'importe quel fichier de clés géré par le service de clés de Tivoli Federated Identity Manager.

   Remarque : Tivoli Federated Identity Manager fournit un fichier de clés par défaut (DefaultTrustedKeystore) qui contient quelques certificats de CA courants que vous pouvez utiliser sous forme de certificats de validation. Cependant, vous devez dans la plupart des cas importer un certificat obtenu auprès de votre partenaire.

   a. Cliquez sur **Tivoli Federated Identity Manager** > **Service de clés**.

      Le panneau Fichiers de clés s'affiche.

   b. Sélectionnez un fichier de clés dans le tableau Fichier de clés. Le bouton **Afficher les clés** est activé.

   c. Cliquez sur **Afficher les clés**. Le panneau Clés s'ouvre. Il répertorie les éléments du fichier de clés sélectionné.

   d. Cliquez sur le bouton **Importer**. L'assistant de clés démarre et affiche le panneau Format de fichier de clés.

   e. Sélectionnez le **format de fichier de clés** adapté au fichier à importer.

      (PEM)
      (Privacy-Enhanced Message) Certificat public

      PKCS#12
      Public Key Cryptography Standard 12 : norme d'échange d'informations personnelles

   f. Pour PKCS#12, indiquez si le fichier de clés contient plusieurs paires de clés.

      1) Sélectionnez la zone **Contient plusieurs paires de clés**, le cas échéant.
      2) Désélectionnez (supprimez la marque de sélection) la zone **Contient plusieurs paires de clés** lorsqu'une seule paire de clés est disponible. L'assistant importe automatiquement la clé.

   g. Cliquez sur **Suivant**. Le panneau Importer la clé s'affiche.

   h. Entrez un chemin d'accès complet dans la zone **Emplacement du fichier de clés**.

      Cette zone s'affiche pour tous les types de format.

      Vous pouvez également cliquer sur **Parcourir** pour localiser le fichier sur le système de fichiers.

   i. Si vous y êtes invité, entrez le **mot de passe** du fichier de stockage de clés.

Remarque : Cette zone ne s'affiche que pour le format PKCS#12.

j. Si vous y êtes invité, entrez le nom de la clé dans la zone **Entrez le nom de la clé à importer**.

Remarque : Cette zone ne s'affiche que pour les fichiers au format PKCS#12 qui contiennent plusieurs paires de clés.

k. Entrez une chaîne désignant la nouvelle clé dans la zone **Intitulé de la nouvelle clé**.

Cette zone s'affiche pour tous les types de format.

l. Cliquez sur **Terminer** pour quitter l'assistant.

4. Lorsque votre partenaire requiert l'authentification de base client, vous devez conserver le nom d'utilisateur et le mot de passe. Après création de votre fédération, lorsque vous utilisez la console de gestion pour ajouter un partenaire, l'assistant de partenaire vous invite à indiquer ces valeurs. Elles ne sont pas nécessaires dans les autres cas.

Ajout d'un partenaire dans une fédération Liberty

Vous pouvez ajouter un partenaire à la fédération Liberty que vous avez créée.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Procédure

1. Copiez le fichier de métadonnées du partenaire vers un emplacement facilement accessible sur votre ordinateur. Par exemple, `/tmp`.

   Remarque : Lorsque le partenaire utilise également Tivoli Federated Identity Manager, ce fichier a été créé sur l'ordinateur du partenaire via la console de gestion afin d'exporter les propriétés de la fédération.
2. Connectez-vous à IBM Integrated Solutions Console.
3. Sélectionnez **Tivoli Federated Identity Manager** > **Configuration de la connexion unique fédérée** > **Partenaires**. L'assistant Partenaire de fédération lance et ouvre l'écran Sélection de fédération.
4. Sélectionnez le bouton d'option à côté de la fédération Liberty.
5. Cliquez sur **Suivant**.
6. Entrez le chemin d'accès complet du fichier de métadonnées sur l'ordinateur local dans la zone **Fichier de métadonnées Liberty du partenaire**. Par exemple :

   ```
   /tmp/libertyfed11_metadata_sp.xml
   ```
7. Cliquez sur **Suivant**.
8. L'étape de configuration suivante est déterminée par la présence ou l'absence dans les métadonnées importées d'informations sur une clé ou un certificat.

   En général, les métadonnées importées contiennent une clé que vous devez importer vers un fichier de clés existant.

   - Si les métadonnées importées contiennent des informations relatives à une clé ou un certificat, l'écran Clé partenaire s'ouvre. Passez à l'étape 9, à la page 523.

- Si les métadonnées importées ne contiennent *pas* d'informations relatives à une clé ou un certificat, l'écran Validation de certificat serveur pour SOAP s'ouvre. Passez à l'étape 10.

9. Entrez les informations requises dans la zone Clé partenaire.

   Remarque : Cette clé ou ce certificat sert à signer les messages Liberty et à signer ou valider les jetons Liberty. Cette clé ne permet pas de sécuriser les communications SOAP sur HTTPS.

   a. Sélectionnez un fichier de clés dans le tableau.
   b. Entrez le mot de passe dans la zone **Mot de passe du fichier de clés**.
   c. Entrez une valeur dans la zone **Entrez un intitulé pour la clé de votre partenaire**. Par exemple :

      `benefits.exemple.com`
   d. Sélectionnez ou désélectionnez la zone **Demander au partenaire de signer les messages Liberty**.
   e. Cliquez sur **Suivant**.

10. L'étape de configuration suivante est déterminée par la présence ou l'absence dans les métadonnées importées d'un noeud final SOAP indiqué pour l'utilisation de HTTPS. Choisissez l'une des actions suivantes :
    - Lorsque les métadonnées importées contiennent un noeud final SOAP indiqué pour l'utilisation de HTTPS, vous êtes invité à définir les clés ou certificats à utiliser. Passez à l'étape 11.
    - Si le noeud final SOAP n'utilise pas HTTPS, il n'est pas nécessaire de définir des clés ou certificats. Passez à l'étape 15, à la page 524.

    Remarque : Dans un déploiement standard, vous devez indiquer les clés ou certificats à utiliser avec le noeud final SOAP. Pour optimiser la sécurité, il convient généralement de sécuriser ce noeud final via HTTPS.

11. Lorsque l'écran Validation de certificat serveur pour SOAP s'ouvre, procédez comme suit :

    a. Sélectionnez un fichier de clés dans le menu déroulant **Fichier de clés**.

       Tivoli Federated Identity Manager fournit un fichier de clés **DefaultTrustedKeyStore**.

       Si vous utilisez un des certificats de CA par défaut (en fonction du contrat que vous avez conclu avec votre partenaire), vous pouvez sélectionner ce fichier de clés. Dans le cas contraire, accédez au fichier de clés dans lequel vous avez placé le certificat que vous avez obtenu auprès de votre partenaire, afin de l'utiliser avec la communication SSL entre les noeuds finals SOAP.

       Dans un environnement de test ou de prototype, vous pouvez sélectionner **DefaultTrustedKeyStore**.

    b. Entrez le mot de passe dans la zone **Mot de passe du fichier de clés**.

       Le mot de passe par défaut de **DefaultTrustedKeyStore** est `testonly`.

    c. Cliquez sur **Liste des clés**.
    d. Cliquez sur le bouton d'option correspondant au certificat désiré, comme indiqué par la valeur de la colonne **Alias** dans le tableau des clés.

       Dans un environnement de test ou de prototype, vous pouvez sélectionner `testwebseal`.

    e. Cliquez sur **Suivant**. L'écran Authentification de client pour SOAP s'ouvre.

12. Vous êtes invité à indiquer si le partenaire requiert *l'authentification par certificat client* ou *l'authentification de base client*.

Le partenaire ne peut exiger qu'une seule de ces méthodes d'authentification. Lorsqu'un des types d'authentification affiché dans l'assistant est sélectionné, les entrées correspondant à l'autre type d'authentification sont désactivées.

- Si le partenaire requiert l'authentification par certificat client, passez à l'étape 13.
- Si le partenaire requiert l'authentification de base client, passez à l'étape 14.

13. Indiquez les valeurs pour l'authentification par certificat client.
    a. Cochez la case **Le partenaire requiert l'authentification par certificat client**.
    b. Sélectionnez un élément dans le menu **Fichier de clés**.

       Le fichier de clés sélectionné est l'emplacement auquel vous avez placé le certificat à utiliser pour l'authentification par certificat client.
    c. Entrez le mot de passe dans la zone **Mot de passe du fichier de clés**.
    d. Cliquez sur **Liste des clés**.
    e. Sélectionnez le bouton d'option correspondant à la clé appropriée dans le tableau des clés.
    f. Cliquez sur **Suivant**.
    g. Passez à la section 15.
14. Indiquez les valeurs pour l'authentification standard des clients.
    a. Cochez la case **Le partenaire requiert l'authentification de base client**.
    b. Entrez le **nom d'utilisateur** et le **mot de passe** que vous avez obtenus de votre partenaire.
    c. Cliquez sur **Suivant**.
    d. Passez à la section 15.
15. L'écran suivant qui s'ouvre est déterminé par votre rôle dans la fédération (fournisseur d'identité ou fournisseur de services) et de votre version de Liberty (1.1 ou 1.2). Dans la plupart des cas, vous devez indiquer des propriétés pour le jeton Liberty. Sélectionnez l'instruction suivante qui correspond à votre configuration :
    - Si vous ajoutez un fournisseur de services partenaire à une fédération de fournisseurs d'identité, passez à la section 16.
    - Si vous ajoutez un fournisseur d'identité partenaire à une fédération de fournisseurs de service, passez à la section 17.
16. Indiquez les données de configuration du module de jeton pour l'ajout d'un fournisseur de services partenaire à une fédération de fournisseurs d'identité. Les données requises sont les mêmes pour Liberty version 1.1 et Liberty version 1.2. L'écran Configuration du module de jeton Liberty 1.1 ou Configuration du module de jeton Liberty 1.2 s'affiche.
    a. Dans la zone **Entrez les types d'attributs suivants (si une étoile "*" est indiquée, tous les types sont inclus)**, définissez les types d'attributs à inclure dans le jeton Liberty.

       Vous pouvez accepter l'entrée par défaut de l'astérisque (*) pour inclure tous les types ou spécifiez des types d'attributs.
    b. Cliquez sur **Suivant**.
    c. Passez à la section 15.
17. Indiquez les données de configuration du module de jeton pour l'ajout d'un fournisseur d'identité partenaire à une fédération de fournisseurs de services. Sélectionnez une action correspondant à la version du protocole Liberty (version 1.1 ou 1.2).

- Lors de l'ajout d'un fournisseur d'identité partenaire à une fédération de fournisseurs de services, à l'aide de Liberty 1.1, aucune configuration de module de jeton n'est requise. L'écran Mappage d'identité s'affiche. Passez à la section 18.
- Lorsque vous ajoutez un fournisseur d'identité partenaire à une fédération de fournisseur de services, à l'aide de Liberty 1.2, procédez comme suit :

a. Vous pouvez éventuellement indiquer une valeur dans la zone **Nom d'utilisateur à employer pour les utilisateurs anonymes**. Si vous n'utilisez pas cette fonction de Liberty, vous pouvez laisser cette zone à blanc.
b. Cliquez sur **Suivant**. L'écran Mappage d'identité s'affiche.
c. Passez à l'étape 18.

18. Choisissez l'action correspondant à l'utilisation que vous faites d'une règle de mappage d'identité :
    - Si vous voulez utiliser la règle de mappage d'identité par défaut du fichier de mappage que vous avez entré dans l'assistant de création de fédération, procédez comme suit :
      a. Ne renseignez pas la zone de la règle de mappage d'identité.
      b. Cliquez sur **Suivant**.
    - Si vous utilisez un fichier de mappage personnalisé avec ce partenaire, procédez comme suit :
      a. Entrez le chemin d'accès au fichier.
      b. Cliquez sur **Importer**.
      c. Cliquez sur **Suivant**.

    Le panneau récapitulatif s'ouvre.
19. Vérifiez que les paramètres sont corrects.
20. Cliquez sur **Terminer**.

    L'écran Ajout de partenaire terminé s'ouvre.
21. Cliquez sur **Activer le partenaire** pour activer ce partenaire.

    Le partenaire a été ajouté à la fédération, mais il est désactivé par défaut par mesure de sécurité. Vous devez activer le partenaire.

Configuration du service d'alias pour Liberty

Le service d'alias doit être configuré pour le même registre d'utilisateurs que le service de gestion Tivoli Federated Identity Manager. N'oubliez pas toutefois que cette prise en charge du protocole Liberty sera obsolète dans les versions à venir de IBM Tivoli Federated Identity Manager.

Pourquoi et quand exécuter cette tâche

Les instructions suivantes décrivent la configuration du serveur LDAP IBMTivoli Directory Server.

Procédure

1. «Création d'un suffixe LDAP pour le service d'alias»
2. «Configuration des paramètres du serveur LDAP», à la page 526

Création d'un suffixe LDAP pour le service d'alias

Vous devez créer un suffixe LDAP `cn=itfim` pour permettre au service d'alias d'accéder au registre d'utilisateurs LDAP.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Pour créer un suffixe LDAP pour le service d'alias, procédez comme suit :

Procédure

1. Arrêtez le serveur LDAP d'IBM.

   UNIX

   ```
   # ibmdirctl -D cn=root -w passw0rd stop
   ```

   Windows

   Utilisez l'icône Services.

2. Ajoutez le suffixe :

   ```
   # idscfgsuf -s "cn=itfim"
   ```

3. Démarrez le serveur LDAP d'IBM.

   UNIX

   ```
   # ibmdirctl -D cn=root -w passw0rd start
   ```

   Windows

   Utilisez l'icône Services.

4. Utilisez **ldapmodify** pour mettre à jour le fichier du schéma LDAP. Par exemple, sous UNIX ou Linux :

   - IBM Tivoli Directory Server :

     ```
     ldapmodify -D cn=root -w passw0rd -f
       /opt/IBM/FIM/etc/itfim-secuser.ldif
     ```

   - Serveur Sun ONE Directory :

     ```
     ldapmodify -D cn=root -w passw0rd -f
        /opt/IBM/FIM/etc/itfim-secuser-sunone.ldif
     ```

Configuration des paramètres du serveur LDAP

Vous devez configurer le service d'alias avec les paramètres LDAP appropriés. N'oubliez pas toutefois que cette prise en charge du protocole Liberty sera obsolète dans les versions à venir de IBM Tivoli Federated Identity Manager

Pourquoi et quand exécuter cette tâche

Le service d'alias est utilisé par le protocole Liberty. Le service d'alias communique avec le serveur du registre d'utilisateurs (LDAP) pour manipuler les informations relatives aux identités utilisateur. Vous devez configurer le service d'alias avec les paramètres LDAP appropriés.

Procédure

1. Cliquez sur **Tivoli Federated Identity Manager → Gestion des domaines → Paramètres du service d'alias**. Le panneau Paramètres du service d'alias s'affiche.
2. Dans la zone Suffixe principal, sous **Paramètres de recherche LDAP**, spécifiez la propriété pour le service d'alias à utiliser lors des recherches dans le registre utilisateur LDAP.

Tableau 121. Propriétés de recherche LDAP

Propriété	Description
Suffixe principal	Indique le suffixe racine dans lequel des paramètres du service d'alias sont écrits. Cette propriété ne peut comporter qu'une seule valeur (suffixe). Par exemple : `cn=itfim`

3. Définissez les propriétés de communication du service d'alias devant être utilisées lors de la communication avec les serveurs LDAP. Utilisez les options de menu de la section **Environnement LDAP** de la fenêtre pour définir les propriétés de communication.

Tableau 122. Propriétés de l'environnement LDAP

Propriété	Description
SSL activé	Cochez cette case pour indiquer si les communications établies entre le service d'alias et les serveurs LDAP doivent être sécurisées à l'aide du protocole SSL (Secure Socket Layer). Si les serveurs LDAP sont configurés pour utiliser SSL, le service d'alias doit également utiliser ce protocole lorsqu'il communique avec eux. Sélectionnez **SSL activé** lorsque vous utilisez le protocole SSL. Dans le cas contraire, désactivez cette case.
Fichier de clés	Lorsque que la case **SSL activé** est cochée, sélectionnez un fichier de clés dans le menu **Fichier de clés**. Il s'agit du nom du fichier de clés sécurisé contenant le certificat de CA du serveur LDAP. **Remarque :** Les certificats de CA de tous les serveurs LDAP doivent se trouver dans le même fichier de clés.

4. Définissez les paramètres de configuration relatifs à chaque serveur LDAP. Utilisez la section **Serveurs LDAP** de la fenêtre pour configurer les propriétés des serveurs LDAP utilisés par le service d'alias.

   Vous pouvez effectuer plusieurs actions de configuration à partir de cette section de la fenêtre. Pour chaque serveur LDAP, vous pouvez définir les valeurs d'un certain nombre de propriétés de configuration.

   - Cliquez sur **Ajouter** pour activer les zones de configuration LDAP du serveur sélectionné.
   - Cliquez sur **Sauvegarder** pour sauvegarder les propriétés LDAP que vous avez entrées dans les zones de configuration d'un serveur. Lorsque vous sauvegardez ces propriétés, la console entre le nom d'hôte et le numéro de port dans la zone **Hôtes LDAP**.

Tableau 123. Propriétés du serveur LDAP

Propriété	Description
Nom d'hôte LDAP	La boîte **Hôtes LDAP** contient la liste des serveurs configurés, par ordre de préférence. Le service d'alias tente d'abord de contacter le serveur figurant au début de la liste. S'il ne parvient pas à établir le contact, il essaie avec le serveur suivant. A l'aide de la flèche vers le haut et de la flèche vers le bas situées à droite de la boîte, déplacez les serveurs LDAP vers le haut ou vers le bas, par ordre de priorité.

Tableau 123. Propriétés du serveur LDAP (suite)

Propriété	Description

Port	Port sur lequel le serveur LDAP écoute.
	
	Port par défaut pour les communications non-SSL :
	`389`
	
	Port par défaut pour les communications SSL :
	`636`

Nom distinctif de la connexion	Nom distinctif utilisé par le service d'alias pour établir une liaison avec le serveur LDAP. Valeur par défaut : `cn=root`
Mot de passe BIND	Mot de passe du nom distinctif indiqué dans la zone **DN Bind**.
Nom de la clé	Nom de la clé de chiffrement à utiliser lors de l'établissement de la communication SSL. Sélectionnez un nom de clé dans la liste des noms. Ces noms proviennent du fichier de clés indiqué dans la zone **Fichier de clés** de la section **Environnement LDAP** de cette fenêtre de configuration.
Nombre minimal de connexions	Nombre de connexions (binds ou liaisons) minimal que le service d'alias peut établir avec le serveur LDAP. La plus petite valeur acceptée est zéro (0). La valeur maximale acceptée correspond à la valeur maximale prise en charge par le type de données. La valeur par défaut est 2. Utilisez-la, sauf si vous devez l'augmenter.
Nombre maximal de connexions	Nombre de connexions (binds ou liaisons) maximal que le service d'alias peut établir avec le serveur LDAP. La valeur maximale acceptée correspond à la valeur maximale prise en charge par le type de données. La valeur par défaut est 10. Utilisez-la, sauf si vous avez besoin de l'augmenter.

5. Cliquez sur **OK** pour sauvegarder les propriétés de configuration et quitter la fenêtre.

Chapitre 32. Configuration d'une fédération de connexion unique WS-Federation

Vous devez, lors de la configuration d'une fédération WS-Federation, spécifier des valeurs pour les propriétés de la fédération.

Le protocole WS-Federation constitue une solution Web de connexion unique normalisée et multifournisseur qui repose sur un ensemble de normes de services Web intégrés (WS*), parmi lesquels WS-Security, WS-Trust et WS-Federation. Lorsque vous configurez Tivoli Federated Identity Manager, sélectionnez le profil passif WS-Federation.

Il convient que vous soyez familiarisé avec les documents relatifs aux normes WS-Federation avant de mettre en oeuvre une fédération de connexion unique. Les normes définissent les échanges de données et le traitement de messages. Vous devez déterminer les informations que vous devez fournir à vos partenaires commerciaux, ainsi que les informations que vos partenaires doivent vous remettre.

Web Services Federation Language (WS-Federation) :
`http://www.ibm.com/developerworks/library/ws-fed`

L'assistant de fédération vous invite à indiquer des valeurs pour un certain nombre de propriétés. La plupart d'entre elles peuvent être modifiées ultérieurement, après la création de la fédération.

Le choix du ou des profils à utiliser dépend à la fois des décisions en matière de règles commerciales et de l'architecture du réseau du point de vue de la sécurité. Les partenaires de la fédération doivent se mettre d'accord sur le choix des profils afin de permettre la connexion unique sur l'ensemble de la fédération. Ce choix doit être fait avant la configuration de la fédération.

SAML 2 prend en charge une gamme unique de profils de connexion unique. Les profils s'étendent au-delà des spécifications, pour permettre une connexion unique fédérée, et peuvent inclure d'autres fonctions, comme la déconnexion unique et l'arrêt de la fédération.

Rôles du fournisseur d'identité et du fournisseur de services

Au sein d'une fédération, chaque partenaire a un rôle. Il s'agit du rôle Fournisseur d'identité ou Fournisseur de services. Découvrez le comportement de chaque rôle.

- Fournisseur d'identité

  Le fournisseur d'identité est un partenaire de fédération qui garantit l'identité des utilisateurs. Il authentifie un utilisateur et transmet un jeton d'authentification au fournisseur de services.

  Le fournisseur d'identité est responsable des tâches suivantes :

  – Il authentifie directement l'utilisation en validant un nom d'utilisateur et un mot de passe.
  – Il authentifie indirectement l'utilisateur en validant une assertion concernant l'identité de l'utilisateur, telle que représentée par un autre fournisseur d'identité.

Le fournisseur d'identité traite la gestion des identités utilisateur afin de libérer le fournisseur de services de cette responsabilité.

- Fournisseur de services

  Un fournisseur de services est un partenaire de fédération qui fournit des services à l'utilisateur final. En général, les fournisseurs de services n'authentifient pas les utilisateurs, mais demandent à un fournisseur d'identité de prendre les décisions liées à l'authentification. Les fournisseurs de services comptent sur les fournisseurs d'identité pour affirmer l'identité d'un utilisateur et pour gérer les identités des utilisateurs pour la fédération.

  Les fournisseurs de services peuvent gérer un compte local pour l'utilisateur, compte qui peut être désigné par un identificateur.

Profils de connexion unique WS-Federation

Les profils de connexion unique permettent à un client utilisant un navigateur Web d'accéder aux ressources d'une fédération WS-Federation 1.0 par une connexion unique.

Généralement, l'utilisateur souhaite accéder à une ressource proposée par un fournisseur de services et doit s'authentifier auprès d'un fournisseur d'identité pour bénéficier de cet accès.

Le profil offre à l'utilisateur Web un profil d'obtention d'une assertion d'authentification qui permet d'établir un contexte de sécurité dans la fédération. L'établissement du contexte de sécurité permet à un utilisateur d'accéder à plusieurs ressources de la fédération sans avoir à s'authentifier plusieurs fois.

WS-Federation prend en charge deux profils utilisables avec les sessions de connexion unique :

POST du navigateur (Browser POST)
Le profil du navigateur POST utilise un formulaire qui renvoie l'action à lui-même (self-posting form) pour échanger un artefact au cours de l'établissement et de l'utilisation d'une session sécurisée entre un fournisseur d'identité, un fournisseur de services et un client (navigateur).

Par défaut, WS-Federation prend en charge le profil POST du navigateur. Aucune configuration n'est requise.

Single logout (SLO, Déconnexion unique)
Ce profil met fin à toutes les sessions de connexion associées à un utilisateur spécifique au sein de la fédération. WS-Federation prend en charge la déconnexion unique par défaut. Aucune configuration n'est requise.

Propriétés de connexion unique WS-Federation

Domaine WS-Federation
Nom du domaine WS-Federation. Ce nom est un identificateur unique pour cette instance de Tivoli Federated Identity Manager. Il figure dans les assertions envoyées aux partenaires de fédération. Les partenaires n'acceptent ces assertions que s'ils trouvent un nom de domaine connu (c'est-à-dire défini). Une valeur par défaut est fournie. Par exemple :

```
https://idp.exemple.com/FIM/sps/wsfed/wsf
```

Dans l'exemple ci-dessus, la chaîne `wsfed` correspond au nom de la fédération. Le noeud final est automatiquement créé. Vous pouvez accepter le nom par défaut.

noeud final WS-Federation
Noeud final de toutes les demandes de service WS-Federation. Une valeur par défaut est fournie. Par exemple :

```
https://idp.ibm.com/FIM/sps/wsfed/wsf
```

Dans l'exemple ci-dessus, la chaîne `wsfed` correspond au nom de la fédération. Le noeud final est automatiquement créé. Vous pouvez accepter le nom par défaut.

Propriétés des jetons WS-Federation

Lorsque vous créez une fédération de connexion unique, vous devez configurer une instance de module de jeton de sécurité pour cette fédération. Le module de jeton correspond à un type de jeton de sécurité qui définit le format du jeton chiffré contenant les données d'identification des utilisateurs.

Le jeton est échangé entre le fournisseur d'identité et le fournisseur de services dans le cadre des services d'authentification et d'autorisation pour le traitement de chaque requête d'accès d'utilisateur.

Lorsque vous utilisez l'assistant de création de fédération pour générer une fédération de connexion unique WS-Federation, le type de jeton SAML 1 est automatiquement sélectionné à votre place.

Lorsque vous configurez un fournisseur d'identité, vous êtes invité à spécifier les propriétés du module de jeton. Si vous configurez un fournisseur de services, la spécification des propriétés du module de jeton n'est pas nécessaire.

Durée de validité (en secondes) d'une vérification avant sa date d'émission.
Elle est indiquée lors de la configuration de jeton, uniquement sur le fournisseur d'identité. Valeur par défaut : 60 secondes. Aucune valeur minimale ou maximale n'est requise.

Durée de validité (en secondes) de la vérification après émission
Nombre entier correspondant à la durée de validité, en secondes, de l'assertion. La valeur par défaut est de 60 secondes. Aucune valeur minimale ou maximale n'est requise. Il est indiqué lors de la configuration de jeton, uniquement sur le fournisseur d'identité.

Mappage d'identité WS-Federation

L'assistant de création de fédération vous invite à spécifier soit l'authentification par fichier de règle de mappage XSLT, soit via une instance de module de mappage personnalisée.

Le fichier de mappage XSLT ou l'instance de module de mappage personnalisée doivent être préparés avant la configuration de la fédération.

Transformation XSL pour le mappage d'identité
La sélection de ce bouton sur l'assistant indique que vous comptez fournir un fichier XSL contenant le mappage d'identité. Entrez le nom d'un fichier du système de fichiers local.

Instance de module de mappage personnalisée
La sélection de ce bouton sur l'assistant indique que vous comptez fournir une instance de module de mappage personnalisée, que vous utiliserez à la place du fichier XSL. Vous serez éventuellement invité à entrer les propriétés de configuration requises par l'instance du module de mappage personnalisée.

Mappage de données d'identification Tivoli Access Manager vers un jeton SAML 1

Mappez les données d'identification Tivoli Access Manager vers un jeton SAML 1 lorsque des messages sont échangés entre les partenaires d'une fédération SAML 1.0, SAML 1.1 ou WS-Federation. Vous devez également mapper les données d'identification lorsque Tivoli Access Manager gère les informations liées à l'identité de l'utilisateur.


Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Lorsqu'une demande d'utilisateur est reçue (par exemple, pour accéder à une ressource distante), le service d'accréditation prend contact avec Tivoli Access Manager et obtient des données d'identification Tivoli Access Manager relatives à l'identité de l'utilisateur.

Dans ce scénario, le module de données d'identification Tivoli Access Manager du service d'accréditation fonctionne en mode de validation. Dans ce mode, il convertit les données d'identification Tivoli Access Manager en un document d'utilisateur universel STS d'entrée (In-STSUUSER).

Le document In-STSUUSER créé à partir du module d'accréditation Tivoli Access Manager contient toutes les informations issues des données d'identification (voir le tableau 124). Ces informations peuvent éventuellement être utilisées par le module de service d'accréditation qui va générer le jeton sortant.

Tableau 124. Entrées In-STSUUSER générées à partir de données d'identification Tivoli Access Manager

Données d'identification Tivoli Access Manager	**Elément In-STSUUSER**
ID utilisateur	Principal Attr: name
Domaine	Principal Attr: domain
ID registre	Principal Attr: registryid
UUID de l'utilisateur	Principal Attr: uuid
Nom du groupe	Nom du groupe
ID registre du groupe	Group Attr: registryid
UUID du groupe	Group Attr: uuid

Tableau 124. Entrées In-STSUUSER générées à partir de données d'identification Tivoli Access Manager (suite)

Données d'identification Tivoli Access Manager	**Elément In-STSUUSER**
Autres entrées de données d'identification xxx	Attrlist Attr: xxx

Le service d'accréditation consulte son entrée de configuration correspondant au partenaire de la fédération (par exemple, la destination qui héberge une ressource demandée). La configuration indique le type de jeton à créer. Dans ce cas, le type de jeton est SAML.

Le module de mappage d'identité convertit ensuite l'élément In-STSUUSER en un utilisateur universel STS de sortie (Out-STSUUSER). L'élément Out-STSUUSER doit contenir les informations requises par le module de jeton SAML de Tivoli Access Manager pour générer un jeton SAML.

L'élément Out-STSUUSER doit contenir les informations suivantes qui permettent au module de jeton SAML de générer un jeton SAML valide :

Tableau 125. Entrées Out-STSUUSER servant à générer un jeton SAML

Elément Out-STSUUSER	**Informations de jeton SAML**	**Obligatoire/ facultatif**
Principal Attr: Name	AuthenticationStatement/Subject/NameIdentifier	Obligatoire
Liste des attributs	Attributs personnalisés supplémentaires	Facultatif

Le module de mappage est responsable des opérations suivantes :

1. Mappage de l'élément Principal Attr Name dans In-STSUUSER vers une entrée de nom Principal dans Out-STSUUSER.

   Le type doit être valide pour SAML. Par exemple :

   ```
   urn:oasis:names:tc:SAML:1.0:assertion#emailAddress
   ```

   Pour obtenir un exemple de fichier de règles de mappage extrait du fichier de mappage d'application de démonstration, ip_saml_10.xsl, voir la figure 59, à la page 534.

```
<!--
 Ce modèle remplace l'intégralité de l'élément Principal par un élément qui ne
 contient que l'adresse électronique (à partir de ivcred tagvalue_email) et le
 type de données approprié pour SAML.
-->
<xsl:template match="//stsuuser:Principal">
 <stsuuser:Principal>
  <stsuuser:Attribute name="name"
                  type="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
   <stsuuser:Value>
    <xsl:value-of
    select="//stsuuser:AttributeList/stsuuser:Attribute[@name='tagvalue_email']
            [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
   </stsuuser:Value>
  </stsuuser:Attribute>
 </stsuuser:Principal>
</xsl:template>
```

Figure 59. Exemple de code XSL présentant le mappage d'une valeur des données d'identification Tivoli Access Manager vers un nom Principal pour un jeton SAML

2. Paramétrage de la méthode d'authentification sur "password" (mot de passe), quelle que soit la valeur obtenue des données d'identification Tivoli Access Manager. Cette action est requise par la norme SAML.

   Pour obtenir un exemple de fichier de règles de mappage extrait du fichier de mappage d'application de démonstration, ip_saml_10.xsl, voir la figure 60.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
               ....
   <!-- D'abord, l'attribut AuthenticationMethod -->
   <stsuuser:Attribute name="AuthenticationMethod"
                          type="urn:oasis:names:tc:SAML:1.0:assertion">
    <stsuuser:Value>urn:oasis:names:tc:SAML:1.0:am:password</stsuuser:Value>
   </stsuuser:Attribute>
               ....
               ....
     </stsuuser:AttributeList>
 </xsl:template>
```

Figure 60. Exemple de code XSL présentant l'affectation d'une méthode d'authentification sous forme d'attribut pour un jeton SAML

3. Remplissage de l'instruction d'attribut de l'assertion SAML à l'aide des attributs de l'élément AttributeList dans In-STSUUSER. Ces informations deviennent des informations personnalisées du jeton SAML.

   Des attributs personnalisés peuvent être requis par les applications qui vont utiliser les informations à transmettre entre les partenaires d'une fédération.

   Pour une description du mappage des attributs personnalisés de l'exemple de fichier de mappage de l'application de démonstration Tivoli Federated Identity Manager, voir la figure 61, à la page 535.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
          ....
               ....
       <!-- Puis l'attribut commonName -->
   <stsuuser:Attribute name="commonName"
                          type="http://exemple.com/federation/v1/commonName">
    <stsuuser:Value>
     <xsl:value-of
     select="//stsuuser:AttributeList/stsuuser:Attribute[@name='tagvalue_name']
             [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>

   <!-- Puis l'attribut ssn -->
   <stsuuser:Attribute name="ssn"
                          type="http://exemple.com/federation/v1/namevalue">
    <stsuuser:Value>
     <xsl:value-of
      select="//stsuuser:AttributeList/stsuuser:Attribute[@name='tagvalue_ssn']
             [@type='urn:ibm:names:ITFIM:5.1:accessmanager']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>
               ....
    </stsuuser:AttributeList>
 </xsl:template>
```

Figure 61. Exemple de code XSL présentant l'affectation d'attributs facultatifs pour un jeton SAML

4. Remplissage des attributs personnalisés. L'élément GroupList de In-STSUUSER n'est pas lu par le module de jeton SAML. Cependant, les informations contenues dans cet élément peuvent, le cas échéant, servir à remplir les attributs personnalisés de Out-STSUUSER.

   La figure 62 présente l'affectation facultative d'une valeur GroupList à un attribut. Cet exemple de code est issu du fichier de mappage d'application de démonstration, ip_saml_10.xsl.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
          ....
       <!-- Puis l'attribut role (peut être multiforme) -->
   <stsuuser:Attribute name="role" type="http://exemple.com/federation/v1/role">
    <xsl:for-each select="//stsuuser:GroupList/stsuuser:Group">
     <stsuuser:Value>
      <xsl:value-of select="@name" />
     </stsuuser:Value>
    </xsl:for-each>
   </stsuuser:Attribute>
               ....
    </stsuuser:AttributeList>
 </xsl:template>
```

Figure 62. Exemple de code XSL présentant l'affectation facultative d'une valeur GroupList à un attribut d'un jeton SAML

Mappage d'un jeton SAML 1 vers des données d'identification Tivoli Access Manager

Le fournisseur de services reçoit un jeton SAML. Le module de jeton SAML, en mode de validation, crée un document In-STSUUSER à partir du jeton SAML.

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

tableau 126 affiche les informations issues du jeton qui sont converties en un document In-STSUUSER.

Remarque : Cette rubrique s'applique aux jetons SAML 1.0 et SAML 1.1.

Tableau 126. Informations de jeton SAML converties en document d'utilisateur universel STS

Informations de jeton SAML	Elément In-STSUUSER	Obligatoire pour Out-STSUUSER ?
AuthenticationStatement/Subject/ NameIdentifier	Principal Attr: Name	Obligatoire
Attributs personnalisés supplémentaires	Liste des attributs	Facultatif

Remarque : Le module de jeton SAML ne remplit pas l'élément GroupList du document In-STSUUSER.

Le service d'accréditation doit convertir ces informations en données d'identification Tivoli Access Manager de manière à prendre une décision d'autorisation sur la demande de l'identité utilisateur. Le module de mappage d'identité convertit les données In-STSUUSER en un fichier XML Out-STSUUSER.

- L'élément NameIdentifier sert à remplir l'attribut `name` de l'élément Principal.

  Pour obtenir un exemple d'affectation d'une valeur définie pour le nom de Principal, voir la figure 63. Cet exemple de code est issu du fichier de mappage d'application de démonstration, sp_saml_1x.xsl.

```
<!--
  Ce modèle remplace le nom de principal (correspondant à l'adresse électronique
  dans la vérification SAML) par l'utilisateur "me_chris".
 -->
 <xsl:template match="//stsuuser:Principal/stsuuser:Attribute[@name='name']">
  <stsuuser:Attribute name="name" type="urn:ibm:names:ITFIM:5.1:accessmanager">
            <stsuuser:Value>me_chris</stsuuser:Value>
            </stsuuser:Attribute>
 </xsl:template>
```

Figure 63. Exemple de code XSL présentant l'affectation d'une valeur pour le nom de Principal d'un jeton SAML

- D'autres informations issues du jeton servent à remplir la zone Attributes de l'élément AttributeList.

  Pour obtenir un exemple d'affectation facultative de valeurs supplémentaires aux attributs, voir la figure 64, à la page 537. Cet exemple de code est issu du fichier de mappage d'application de démonstration, sp_saml_1x.xsl.

```
<xsl:template match="//stsuuser:AttributeList">
  <stsuuser:AttributeList>
        ....
      <!-- Attribut tagvalue_name -->
   <stsuuser:Attribute name="tagvalue_name"
                        type="urn:ibm:names:ITFIM:5.1:accessmanager">
    <stsuuser:Value>
    <xsl:value-of
  select="//stsuuser:AttributeList/stsuuser:Attribute[@name='commonName']
      [@type='http://exemple.com/federation/v1/commonName']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>

   <!-- Attribut tagvalue_ssn -->
   <stsuuser:Attribute name="tagvalue_ssn"
                        type="urn:ibm:names:ITFIM:5.1:accessmanager">
    <stsuuser:Value>
     <xsl:value-of
              select="//stsuuser:AttributeList/stsuuser:Attribute[@name='ssn']
         [@type='http://exemple.com/federation/v1/namevalue']/stsuuser:Value" />
    </stsuuser:Value>
   </stsuuser:Attribute>
              ....
     </stsuuser:AttributeList>
 </xsl:template>
```

Figure 64. Exemple de code XSL présentant l'affectation facultative d'attributs pour un jeton SAML

Chapitre 33. Configuration d'une fédération de connexion unique WS-Federation

Pour configurer une fédération à connexion unique WS-Federation, vous devez créer la fédération, y ajouter votre partenaire, puis fournir à celui-ci les informations de configurations issues de votre nouvelle fédération.

1. «Création d'une fédération de connexion unique WS-Federation»
2. «Configuration de WebSEAL en tant que serveur point de contact», à la page 540
3. «Exportation des propriétés WS-Federation», à la page 542
4. «Obtention des informations de configuration auprès d'un partenaire WS-Federation», à la page 542
5. «Ajout d'un partenaire dans une fédération de connexion unique WS-Federation», à la page 545

Création d'une fédération de connexion unique WS-Federation

Le protocole passif WS-Federation permet de créer et de configurer une fédération dans l'assistant de fédération.

Procédure

1. Connectez-vous à la Integrated Solutions Console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configuration de la connexion unique fédérée** > **Fédérations**. Les portlets Domaine en cours et Fédérations s'ouvrent.
3. Cliquez sur **Créer**. L'assistant de fédération démarre. Le panneau Informations générales s'ouvre.
4. Indiquez le nom de la fédération et sélectionnez un rôle.
5. Cliquez sur **Suivant**.
6. Entrez les coordonnées du contact.
7. Cliquez sur **Suivant**.
8. Sélectionnez le protocole WS-Federation Passive Protocol.
9. Cliquez sur **Suivant**. Le panneau Serveur point de contact s'ouvre.
10. Entrez l'adresse du point de contact.
11. Cliquez sur **Suivant**.
12. Choisissez l'une des options suivantes :
    - Si vous voulez configurer un fournisseur de services, l'étape suivante est le mappage d'identité. Passez à l'étape 14.
    - Lorsque vous configurez un fournisseur d'identité, le panneau Configuration du jeton de sécurité s'affiche. Indiquez les propriétés de jeton requises.

      Voir Chapitre 32, «Configuration d'une fédération de connexion unique WS-Federation», à la page 529.
13. Cliquez sur **Suivant**. Le panneau Options de mappage d'identité s'ouvre.
14. Sélectionnez l'un des boutons d'option suivants.
    - Utiliser la transformation XSL pour le mappage d'identité

Indique que vous comptez fournir un fichier XSL contenant le mappage d'identité requis.

a. Lorsque vous sélectionnez cette option et cliquez sur **Suivant**, le panneau Mappage d'identité s'affiche. Dans la zone **Fichier XSLT contenant une règle de mappage d'identité**, entrez le nom d'un fichier du système de fichiers local qui contient la règle de mappage d'identité.

   Il s'agit du fichier que vous avez préparé avant de procéder à cette installation.

   Vous pouvez éventuellement localiser le fichier sur le système de fichiers local à l'aide du bouton **Parcourir**.

b. Cliquez sur **Suivant**.

   Une erreur s'affiche lorsque le fichier est introuvable ou ne contient aucune donnée XSLT (eXtensible Stylesheet Language Transform) valide.

- Utiliser l'instance de module de mappage personnalisée

  Indique que vous comptez fournir une instance de module de mappage personnalisée, que vous utiliserez à la place du fichier XSL.

  a. Lorsque vous sélectionnez l'option Utiliser l'instance de module de mappage personnalisée, un tableau des instances de module s'affiche. Cliquez sur le bouton d'option correspondant à l'instance de module à utiliser et cliquez sur **Suivant**.

  b. Le cas échéant, vous serez alors invité à indiquer des valeurs pour les propriétés de l'instance de module de mappage personnalisée. Sinon, le panneau affiche un message indiquant qu'aucune propriété ne doit être configurée pour l'instance de module indiquée.

Le panneau Récapitulatif s'affiche.

15. Vérifiez que les paramètres de configuration sont corrects.
16. Cliquez sur **Terminer**. Le portlet Création de fédération terminée s'affiche.
17. Cliquez sur **Redémarrer WebSphere**.

Que faire ensuite

Si vous utilisez WebSEAL en tant que serveur point de contact, procédez maintenant à sa configuration. Ne quittez pas la console de gestion. Voir aussi :

- «Configuration de WebSEAL en tant que serveur point de contact»

Configuration de WebSEAL en tant que serveur point de contact

Si vous envisagez d'utiliser WebSEAL en tant que serveur point de contact, vous devez le configurer pour la fédération de connexion unique WS-Federation.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Ces instructions ont pour hypothèse que le profil du point de contact WebSEAL est activé.

Pourquoi et quand exécuter cette tâche

L'assistant de fédération comporte un bouton qui vous permet de récupérer l'outil de configuration Tivoli Federated Identity Manager. Vous devez obtenir cet outil, puis l'exécuter. Pour configurer WebSEAL en tant que serveur point de contact, procédez comme suit :

Procédure

1. Une fois la fédération créée, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager** pour recharger vos modifications.

   Remarque : La console de gestion vous offre la possibilité d'ajouter immédiatement un partenaire, mais pour cette configuration initiale de la fédération, vous devez d'abord exécuter d'autres tâches.
2. Cliquez sur **Terminé** pour revenir au panneau Fédérations.
3. Cliquez sur **Télécharger l'outil de configuration Tivoli Access Manager**.
4. Enregistrez l'outil de configuration sur le système de fichiers de l'ordinateur hébergeant le serveur WebSEAL.
5. Démarrez l'outil de configuration depuis une ligne de commande. La syntaxe est la suivante :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf
   ```

   Remarque : Si la norme FIPS (Federal Information Processing Standards) est activée pour votre environnement, une fabrique de connexions sécurisées doit être indiquée. Par exemple :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf -sslfactory TLS
   ```

   Vous aurez besoin de l'ID (par défaut : sec_master) et du mot de passe de l'utilisateur d'administration Tivoli Access Manager. L'utilitaire configure les noeuds finals sur le serveur WebSEAL, crée une jonction WebSEAL, relie les listes de contrôle d'accès adaptées et active les méthodes d'authentification adéquates.

Exemple

Par exemple, lorsque vous avez mis le fichier tfimcfg.jar dans le répertoire `/tmp` et que le nom de l'instance WebSEAL est `default`, la commande est la suivante :

```
java -jar /tmp/tfimcfg.jar -action tamconfig -cfgfile webseald-default
```

Pour plus d'informations, voir Annexe A, «Référence de tfimcfg», à la page 827.

Que faire ensuite

La prochaine tâche consiste à exporter les propriétés de votre fédération WS-Federation dans un fichier. Voir «Exportation des propriétés WS-Federation», à la page 542.

Configuration de WebSphere en tant que serveur point de contact

Tivoli Federated Identity Manager est configuré pour utiliser par défaut Tivoli Access Manager WebSEAL comme serveur point de contact. Pour configurer WebSphere en tant que serveur point de contact, vous devez procéder à une modification de la configuration.

Procédure

1. Connectez-vous à la console d'administration.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Sélectionnez **WebSphere**.
4. Cliquez sur **Activer**.

Résultats

Le serveur WebSphere est désormais configuré en tant que point de contact.

Exportation des propriétés WS-Federation

Pour rejoindre une fédération hébergée par un autre partenaire commercial, vous devez fournir vos propriétés de configuration de fédération. Vous pouvez facilement exporter les propriétés et les fournir à votre partenaire.

Pourquoi et quand exécuter cette tâche

Dans le cas des fédérations WS-Federation, vous devez préparer manuellement un fichier contenant les propriétés de configuration. Transmettez ce fichier à votre partenaire de fédération.

Procédure

1. Connectez-vous à la console de gestion.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configuration de la connexion unique fédérée** > **Fédérations**. Le panneau Fédérations s'affiche.
3. Sélectionnez votre fédération à connexion unique WS-Federation dans le tableau.
4. Afficher les propriétés de la fédération. Obtenez les propriétés indiquées dans la liste de la rubrique «Propriétés WS-Federation à échanger avec votre partenaire», à la page 543
5. Délivrez le fichier à votre partenaire suivant la méthode convenue dans l'accord conclu entre votre société et celle de votre partenaire.

Que faire ensuite

Vous devez le fournir à votre partenaire lorsque ce dernier souhaite ajouter les informations de votre configuration à sa fédération de connexion unique WS-Federation.

Obtention des informations de configuration auprès d'un partenaire WS-Federation

Vous devez obtenir ces informations auprès de votre partenaire WS-Federation.

Avant de commencer

Si vous souhaitez ajouter votre partenaire commercial en tant que partenaire de votre fédération de connexion unique WS-Federation, vous devez obtenir auprès de celui-ci les informations de configuration nécessaires relatives à sa fédération WS-Federation.

Une fédération WS-Federation doit avoir été déjà installée et configurée par votre partenaire. La fédération de votre partenaire joue le rôle opposé à celui de votre fédération. Si, par exemple, votre fédération est configurée en tant que fournisseur d'identité, la fédération de votre partenaire l'est en tant que fournisseur de services.

Vous pouvez obtenir ces informations en faisant en sorte que votre partenaire assemble manuellement les propriétés de configuration de sa fédération. Le partenaire doit ensuite vous fournir les informations en employant la méthode convenue dans le cadre de l'accord conclu avec votre partenaire.

Procédure

1. Votre partenaire doit collecter les propriétés de sa fédération via la console de gestion. Il convient que le partenaire vous fournisse les propriétés indiquées à la rubrique «Propriétés WS-Federation à échanger avec votre partenaire».
2. Il convient que votre partenaire pour délivre le fichier suivant la méthode convenue dans l'accord conclu entre votre société et celle de votre partenaire.

Propriétés WS-Federation à échanger avec votre partenaire

Vous et votre partenaire devez vous assurer que vous disposez des informations que vous devez échanger avant de pouvoir ajouter votre partenaire à une fédération.

Propriétés de la fédération

Tableau 127. Propriétés WS-Federation

Propriété	Description
Nom de la fédération	Chaîne de caractères désignant la fédération
Rôle	Fournisseur d'identité ou fournisseur de services
Protocole	profil passif WS-Federation
Nom de la société	Nom de la société qui a créé la fédération. (obligatoire)
Adresse URL de la société	Adresse URL de la société qui a créé la fédération. (facultatif)
Prénom et nom	Nom de la personne au sein de la fédération pouvant être contactée par d'autres sociétés. (facultatif)
Adresse électronique	Adresse électronique de la personne pouvant être contactée par d'autres sociétés de la fédération. (facultatif)
Numéro de téléphone	Numéro de téléphone de la personne au sein de la fédération pouvant être contactée par d'autres sociétés. (facultatif)

Tableau 127. Propriétés WS-Federation (suite)

Propriété	Description
Type de contact	chaîne décrivant le type de rôle dans l'entreprise, par exemple Conseiller technique ou Personnel d'assistance. (facultatif)

Données relatives à la WS-Federation

Tableau 128. Données relatives à la WS-Federation

Propriété	Description
Domaine WS-Federation	Nom unique du domaine WS-Federation. Par exemple : `https://idp.exemple.com/FIM/sps/wsfed/wsf`
noeud final WS-Federation	Noeud final de votre partenaire pour toutes les demandes de services WS-Federation. Par exemple : `https://idp.exemple.com/FIM/sps/wsfed/wsf`
Durée maximale d'une requête (en secondes)	Durée maximale de validité, en secondes, d'une demande ou d'un message envoyé par un partenaire WS-Federation.

Configuration du module de jeton SAML

Tableau 129. Propriétés du module de jeton SAML

Propriété	Description
Activer la signature des assertions	Indique si le fournisseur d'identité signe des assertions avant de les envoyer au fournisseur de services partenaire.
Sélectionner la clé de signature des assertions	Indiquez le nom de la clé à utiliser lors de la signature d'assertions. Cette option est spécifiée pour les partenaires des fournisseurs de services.
Algorithme de signature pour la signature des assertions SAML	Indique l'algorithme de signature à utiliser pour signer l'assertion SAML. Sélectionnez l'une des options suivantes : • RSA-SHA1 • DSA-SHA1 • RSA-SHA256
Entrez les types d'attributs suivants (si une étoile '*' est indiquée, tous les types sont inclus)	Types d'attributs à inclure dans le module de jeton SAML. Cette option est spécifiée pour les partenaires des fournisseurs de services.
Activer la validation des signatures	Lorsque cette option est sélectionnée, elle indique que le fournisseur de services valide la signature sur les assertions provenant du fournisseur d'identité partenaire. Option spécifiée pour les partenaires des fournisseurs d'identité.
Sélectionner la clé de validation	Indiquez le nom de la clé à utiliser pour la validation des signatures. Spécifiée pour un partenaire de fournisseur d'identité.

Ajout d'un partenaire dans une fédération de connexion unique WS-Federation

Vous pouvez utiliser la console d'administration pour ajouter un partenaire dans une fédération de connexion unique WS-Federation.

Pourquoi et quand exécuter cette tâche

La procédure de configuration d'ajout est identique lors de l'ajout de tous les partenaires. Les propriétés de configuration sont différentes pour des partenaires de fournisseur d'identité et de services. L'assistant Partenaire vous invite à entrer les propriétés nécessaires.

Procédure

1. Connectez-vous à la IBM Integrated Solutions Console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configuration de la connexion unique fédérée** > **Partenaires**. Le panneau Partenaires de la fédération s'ouvre.
3. Cliquez sur **Créer**. Le panneau Sélection de fédération s'ouvre.
4. Sélectionnez la fédération à laquelle vous souhaitez ajouter un partenaire.
5. Cliquez sur **Suivant**. Le panneau Personne à contacter s'ouvre.
6. Entrez les propriétés du contact.

   Le nom de l'entreprise est obligatoire. Les autres zones sont facultatives.
7. Cliquez sur **Suivant**. Le panneau Données relatives à WS-Federation s'ouvre.
8. Entrez les propriétés demandées.
9. Cliquez sur **Suivant**. Le panneau Configuration du jeton de sécurité s'ouvre.
10. Entrez les propriétés de configuration du jeton de sécurité fédéré.

    Les propriétés de configuration sont propres au rôle du partenaire :
    - Lors de l'ajout d'un fournisseur d'identité partenaire :
      a. Si les assertions doivent être signées, cliquez sur **Activer la signature des vérifications**. Si vous cochez cette case, vous devez indiquer une clé à utiliser pour la signature des assertions. Sélectionnez l'option **Fichier de clés**, tapez la valeur de la zone **Mot de passe du fichier de clés**, cliquez sur **Liste des clés** et sélectionnez une clé dans le tableau.
      b. Vous pouvez aussi indiquer des attributs dans la zone : **Inclure les types d'attributs suivants (si un * est indiqué, tous les types sont inclus).**.
      c. Cliquez sur **Suivant**.
    - Lors de l'ajout d'un fournisseur de services partenaire :
      a. Si les signatures doivent être validées, cliquez sur **Activer la validation des signatures**. Si vous cochez cette case, vous devez indiquer la clé à utiliser pour la validation des signatures. Sélectionnez l'option **Fichier de clés**, tapez la valeur de la zone **Mot de passe du fichier de clés**, cliquez sur **Liste des clés** et sélectionnez une clé dans le tableau.
      b. Cliquez sur **Suivant**.

    Le panneau Options de mappage d'identité s'ouvre.
11. Sélectionnez l'un des boutons d'option suivants.
    - Utiliser la transformation XSL pour le mappage d'identité

Indique que vous comptez fournir un fichier XSL contenant tous les mappages d'identité requis.

a. Lorsque vous sélectionnez cette option et cliquez sur **Suivant**, le panneau Mappage d'identité s'affiche. Ne remplissez pas cette zone si vous souhaitez utiliser la règle de mappage d'identité par défaut que vous avez entrée dans l'assistant de création de fédération.

   Si vous voulez remplacer la règle de mappage par défaut par une règle spécifique au partenaire, entrez le nom d'un fichier du système de fichiers local contenant la règle de mappage d'identité concernée dans la zone **Fichier XSLT contenant une règle de mappage d'identité**.

   Vous pouvez éventuellement localiser le fichier sur le système de fichiers local à l'aide du bouton **Parcourir**.

b. Cliquez sur **Suivant**.

- Utiliser l'instance de module de mappage personnalisée

  Indique que vous comptez fournir une instance de module de mappage personnalisée, que vous utiliserez à la place du fichier XSL.

  a. Lorsque vous sélectionnez l'option Utiliser l'instance de module de mappage personnalisée, un tableau des instances de module s'affiche. Cliquez sur le bouton d'option correspondant à l'instance de module à utiliser et cliquez sur **Suivant**.

  b. Le cas échéant, vous serez alors invité à indiquer des valeurs pour les propriétés de l'instance de module de mappage personnalisée. Sinon, le panneau affiche un message indiquant qu'aucune propriété ne doit être configurée pour l'instance de module indiquée.

Le panneau Récapitulatif s'affiche.

12. Vérifiez que les paramètres sont corrects, puis cliquez sur **Terminer**. Le panneau Ajout de partenaire terminé s'ouvre.
13. Cliquez sur **Activer le partenaire** pour activer ce partenaire.

    Le partenaire a été ajouté à la fédération, mais il est désactivé par défaut par mesure de sécurité. Vous devez activer le partenaire.

Partie 4. Configuration de la gestion de sécurité des services Web

Les rubriques de la section Configuration vous guident pas à pas lorsque vous configurez la gestion de sécurité des services Web pour Tivoli Federated Identity Manager.

Veuillez consulter d'abord la section suivante :

Chapitre 34, «Configuration de la gestion de sécurité des services Web», à la page 549

Chapitre 34. Configuration de la gestion de sécurité des services Web

La configuration de la gestion de sécurité des services Web commence par l'établissement d'un domaine Tivoli Federated Identity Manager. Lorsque le domaine est établir, vous pouvez configurer le composant de gestion de la sécurité des services Web.

La procédure de configuration de la gestion de sécurité des services Web est la suivante :

1. Configuration d'un domaine Tivoli Federated Identity Manager.

   Le déploiement d'un scénario Tivoli Federated Identity Manager nécessite la création d'un domaine Tivoli Federated Identity Manager.

   vous devez créer et configurer un domaine avant de pouvoir configurer le composant gestion de la sécurité des services Web.

   Voir Chapitre 3, «Configuration de domaine», à la page 25.
2. Configuration du composant gestion de la sécurité des services Web.

   La configuration du composant peut s'effectuer de différentes manières afin de refléter les scénarios de déploiement. Pour une description détaillée de la configuration, voir : *IBM Tivoli Federated Identity Manager - Guide de gestion de la sécurité des services Web*.

Partie 5. Configuration du service STS (Security Token Service)

Les rubriques de la section Configuration vous guident pas à pas lors de la configuration d'un service de jeton de sécurité (STS) dans le cadre d'une solution intégrée de gestion des identités d'utilisateur dans un environnement réseau réparti.

La présente section décrit le déploiement d'un module de service STS sur une délégation Kerberos pour les besoins de prise en charge d'une solution de jonctions Kerberos fournie par l'association entre Tivoli Federated Identity Manager et Tivoli Access Manager for e-Business WebSEAL, ainsi que des produits WebSphere et autres composants additionnels.

Veuillez d'abord consulter la présentation du scénario de déploiement :

Chapitre 35, «Présentation de la délégation contrainte Kerberos», à la page 553

Chapitre 35. Présentation de la délégation contrainte Kerberos

Tivoli Federated Identity Manager fournit un service de jeton de sécurité, ou STS (Security Token Service) qui permet d'échanger les formats des jetons de sécurité. Cette fonction sert à déplacer les données d'identification de l'utilisateur entre différents formats de jeton, suivant les besoins des différentes applications.

Remarque : IBM a déprécié le client Tivoli Federated Identity Manager Security Token Service (STS) dans cette version.

Si vous utilisez WebSphere 6.X, vous pouvez continuer de vous servir du client Tivoli Federated Identity Manager Security STS tant que Tivoli Federated Identity Manager prend en charge WebSphere 6.X. Lorsque Tivoli Federated Identity Manager arrêtera son support pour WebSphere 6.X, vous devrez utiliser WebSphere Application Server version 7 Update 11 et version ultérieure. Voir API client WS-Trust et WS-Trust Clients pour plus d'informations.

Le service STS fait partie intégrante des solutions de connexion unique de Tivoli Federated Identity Manager, mais est également utilisable de manière autonome. Tivoli Federated Identity Manager peut être intégré dans divers déploiements réseau hétérogènes en raison de la fonctionnalité autonome.

L'un de ces types de déploiement est un environnement qui exploite l'authentification intégrée de Microsoft Windows (SPNEGO) avec des tickets Kerberos. Dans cet environnement, Tivoli Federated Identity Manager peut être déployé de manière à accepter les données d'identification de l'utilisateur et à les convertir suivant le format Kerberos nécessaire.

Pour utiliser cette fonction, Tivoli Federated Identity Manager inclut un module de service de jeton de sécurité destiné spécifiquement à la délégation contrainte Kerberos. Le module de la délégation Kerberos facilite l'émission des tickets du service d'application pour la délégation contrainte Kerberos, également appelé S4U2Proxy ("Service for User To Proxy").

Le module prend uniquement en charge l'*émission* des jetons et *excusivement* les tickets du service d'application Kerberos de Windows via le modèle de délégation à contrainte.

Une fonction principale du modèle de délégation contrainte Kerberos est que le mot de passe de l'utilisateur pour lequel le ticket de service Kerberos doit être obtenu peut être inconnu de l'application qui génère le ticket. Dans ce cas, l'application est WebSphere plus Tivoli Federated Identity Manager. L'application a uniquement besoin de connaître le nom de l'utilisateur, ainsi que le nom principal (SPN) du service Kerberos cible.

Le module STS de la délégation contrainte Kerberos vise principalement à permettre à Tivoli Access Manager WebSEAL de prendre en charge la connexion unique sur les jonctions Kerberos. Il s'agit de jonctions avec un serveur Web configuré pour l'authentification intégrée SPNEGO sous Windows.

WebSEAL permet de maintenir une session utilisateur au moyen de n'importe quel mécanisme d'authentification sélectionné par ses soins, puis de se connecter à un serveur Web (par exemple le serveur IIS) par le biais du flux d'authentification SPNEGO. Ce flux d'authentification exploite un ticket Kerberos.

L'utilisation de droits d'accès Kerberos pour la connexion unique sur des jonctions offre les capacités suivantes :

- Les droits d'accès Kerberos sont aisément utilisables par les applications Web ASP.NET sans nécessiter de déploiement de code supplémentaire.
- Les droits d'accès Kerberos peuvent être transmis d'une application à l'autre tout en maintenant une signature cryptographique, ce qui permet de renforcer la sécurité.

Remarque : ce module diffère du module STS Kerberos Java natif de Tivoli Federated Identity Manager. Le module Kerberos Java prend en charge l'émission et la validation générique des autres tickets Kerberos.

Vous trouverez plus d'informations sur les extensions Kerberos de Windows à l'emplacement suivant :

- http://technet2.microsoft.com/WindowsServer/en/Library/c312ba01-318f-46ca-990e-a597f3c294eb1033.mspx
- http://msdn2.microsoft.com/en-us/library/aa480585.aspx
- http://msdn.microsoft.com/msdnmag/issues/03/04/SecurityBriefs/

Présentation de la délégation contrainte Kerberos avec des jonctions WebSeal

Tivoli Federated Identity Manager utilise le module STS de la délégation contrainte Kerberos pour générer les jetons Kerberos.

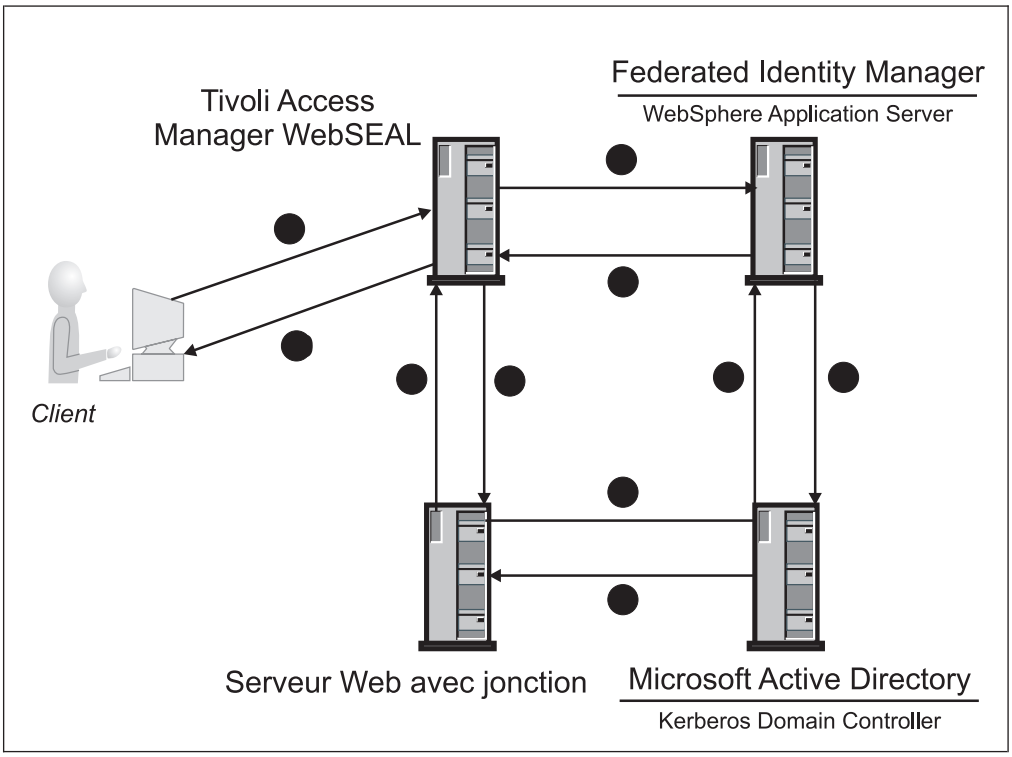


Figure 65. Délégation contrainte Kerberos avec une jonction WebSEAL

WebSEAL extrait les jetons Kerberos en déléguant la demande de jeton au module STS.

La figure 65 illustre un exemple de déploiement des applications impliquées dans la réalisation de ce type de connexion unique. Le diagramme illustre également la manière dont les messages circulent entre les différents composants physiques.

1. Le client s'appuie sur le processus d'authentification standard de Tivoli Access Manager pour s'authentifier sur WebSEAL via le protocole HTTPS ou HTTP, puis demander un objet sur le serveur comportant les jonctions. WebSEAL autorise la demande émise par le client, puis détermine qu'un ticket Kerberos est requis pour accéder à l'application de la jonction.
2. WebSEAL génère un message d'émission de clé WS-Trust destiné au serveur Tivoli Federated Identity Manager. Une clé d'émission WS-Trust unique peut être utilisée pour demander des jetons Kerberos multiples. WebSEAL ouvre une connexion auprès du serveur WebSphere exécutant Tivoli Federated Identity Manager. WebSEAL adresse la requête SOAP au serveur WebSphere.
3. Le serveur Tivoli Federated Identity Manager exécuté sur l'instance du serveur WebSphere vérifie que le serveur WebSEAL est autorisé à démarrer le service STS (Security Token Service).

   Le service STS démarre ensuite un module d'accréditation Tivoli Federated Identity Manager afin de demander le nombre configuré de tickets Kerberos requis pour le serveur Web de la jonction au titre du client. Le module

d'accréditation utilise Kerberos, sur le port TCP ou UDP 88 pour communiquer avec le contrôleur de domaine Active Directory.

4. Le contrôleur de domaine Active Directory vérifie que le serveur Tivoli Federated Identity Manager est autorisé à demander des tickets pour le serveur Web joint, au titre de l'utilisateur concerné. Le contrôleur de domaine Active Directory renvoie le nombre de tickets Kerberos configuré vers le composant d'exécution de Tivoli Federated Identity Manager.
5. Le composant d'exécution de Tivoli Federated Identity Manager renvoie les tickets sous forme de réponse SOAP adressée au serveur WebSEAL.
6. Le serveur WebSEAL met en cache les tickets Kerberos et transmet l'un d'entre eux avec la demande client adressée au serveur Web de la jonction, via le protocole HTTP ou HTTPS.
7. Ce serveur Web de jonction demande la validation du ticket Kerberos de la part du contrôleur de domaine Kerberos (KDC). Le contrôleur KDC est décrit ici comme étant le même système que le serveur Active Directory.
8. Le contrôleur KDC vérifie que le ticket Kerberos est valide. Le ticket Kerberos est utilisé comme preuve de l'identité du client et peut également servir à d'autres contrôles d'autorisation.
9. Le serveur Web de jonction renvoie une réponse HTTP à WebSEAL.
10. WebSEAL renvoie une réponse HTTP au client.

Un nouveau ticket Kerberos est envoyé avec la demande au serveur Web de jonction dans les cas suivants :

- lors de chaque requête ultérieure à partir du même client, et
- au cours de la même session de connexion au même serveur Web de jonction.

Le nouveau ticket Kerberos est soit extrait du cache WebSEAL des tickets Kerberos, soit une demande est envoyée au serveur WebSphere exécutant Tivoli Federated Identity Manager afin d'obtenir une nouvelle série de tickets Kerberos.

Présentation du déploiement

Le déploiement Kerberos requiert des logiciels spécifiques. Cette section répertorie les prérequis et les tâches impliquées dans le déploiement.

Logiciels prérequis

- Tivoli Federated Identity Manager doit être exécuté sur Windows 2003 Server Service Pack 2 ou version supérieure.

  Le module de mise à jour est requis en raison d'un problème connu de fuite de mémoire affectant le processus Windows Isass.exe sur les versions antérieures. Voir http://support.microsoft.com/kb/907524/.
- Le serveur WebSEAL peut être exécuté sur n'importe quelle plateforme prise en charge.

  Le serveur WebSEAL *ne doit pas nécessairement* faire partie du domaine Active Directory.
- Le déploiement de WebSphere peut être effectué soit en mode autonome, soit sous forme de cluster. Il convient que tous les serveurs WebSphere membres du cluster soient installé sur des systèmes Windows et fassent partie du domaine.
- Lorsque les utilisateurs de Tivoli Access Manager sont stockés dans Active Directory, le serveur de règles Tivoli Access Manager doit être hébergé sous Windows et être membre du domaine.

- Il convient que tous les contrôleurs du domaine Active Directory soient exécutés au niveau fonctionnel de Windows Server 2003.
- La prise en charge de Tivoli Federated Identity Manager support pour les modules de délégation Kerberos n'est pas incluse dans le produit Tivoli Federated Identity Manager Business Gateway.

Présentation des tâches de déploiement

1. Activez l'authentification Windows intégrée
2. Configurez Active Directory et WebSphere pour la délégation contrainte
3. Installez et configurez un domaine et un composant d'exécution Tivoli Federated Identity Manager
4. Configurez une instance de module Kerberos et une chaîne d'accréditation pour le module STS de la délégation contrainte Kerberos
5. Configurez WebSEAL pour la prise en charge d'une jonction Kerberos

Tableau 130. Exemples de noms d'hôte de serveur utilisés dans cette documentation

Rôle du serveur	**Exemple de valeur**
Serveur dorsal (serveur Web à jonctions)	mydataserver.example.com
Serveur WebSEAL	websealhost.example.com
Nom d'hôte Active Directory	activedirectoryhost.example.com

Chapitre 36. Activation de l'authentification Windows intégrée

Ces instructions expliquent comment configurer Microsoft IIS en vue de l'authentification SPNEGO.

Avant de commencer

Ces instructions supposent que vous avez déployé Windows Server 2003 avec Active Directory. Ces étapes doivent être terminées avant la configuration de la délégation contrainte.

Procédure

1. Sur le contrôleur de domaine, sélectionnez **Démarrer** > **Programmes** > **Outils d'administration** > **Utilisateurs et ordinateurs Active Directory**.
2. Créez un utilisateur agissant en tant que proxy pour le serveur IIS. Par exemple, `iisuser`.
3. Indiquez que le mot de passe de l'utilisateur `n'expire jamais`.
4. Ouvrez une invite de commande.
   a. Accédez au répertoire `C:\Program Files\Support Tools`.
   b. Exécutez la commande **ktpass** appropriée.

      Syntaxe de la commande ktpass :

      ```
      ktpass -princ HTTP/nom_serveur_IIS.nom_domaine@DOMAIN_NAME
        -mapuser nom_utilisateur_IIS -mapOp set
      ```

      où :

      - `-princ` est le nom principal, indiqué sous la forme *user@REALM*
      - `-mapuser` correspond à la valeur -princ pour ce compte d'utilisateur. Cette option est définie par défaut.
      - `-mapOp` spécifie le mode de définition de l'attribut de mappage : `set` *set_value*
5. Affichez les propriétés du compte pour `iisuser`. Vérifiez que la zone **Nom de connexion de l'utilisateur** est définie sur la valeur suivante :

   `HTTP/`*nom_serveur_IIS.nom_domaine*

   Par exemple :

   `HTTP/mydataserver.example.com`
6. Configuration de l'identité du pool d'applications
   a. Sur le système du serveur IIS, sélectionnez **Démarrer** > **Programmes** > **Outils d'administration** > **Gestionnaire IIS (Internet Information Service)**.
   b. Sélectionnez *nom_de_votre_serveur/nom IIS* > **Programmes** > **Pools d'applications** > **Pool d'applications par défaut**.
   c. Cliquez avec le bouton droit et sélectionnez **Propriétés**.
   d. Sélectionnez l'onglet d'identité et spécifiez l'identité de domaine de l'utilisateur IIS (exemple : `iisuser`).

   Pour obtenir des instructions détaillées sur la tâche Windows de *configuration de l'identité du pool de connexions avec IIS 6.0*, voir :

   http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/f05a7c2b-36b0-4b6e-ac7c-662700081f25.mspx?mfr=true.
7. Ouvrez l'Explorateur Windows.

a. Accédez au répertoire `C:\WINNT\Microsoft.NET\Framework\v1.1.4322\Temporary ASP.NET Files`.
b. Sélectionnez **Propriétés**.
c. Sélectionnez l'onglet **Sécurité**.
d. Attribuez à l'utilisateur du domaine `iisuser` le contrôle d'accès complet au répertoire.

8. Accédez au système IIS et sélectionnez **Démarrer** > **Programmes** > **Outils d'administration** > **Gestion de l'ordinateur**.
   a. Ouvrez **Utilisateurs et groupes locaux**.
   b. Ouvrez **Groupes**.
   c. Cliquez avec le bouton droit sur le groupe local `IIS_WPG`.
   d. Sélectionnez les propriétés.
   e. Sélectionnez **Ajouter**.
   f. Ajoutez l'utilisateur du domaine (soit dans le cas présent, `iisuser`) à ce groupe local.
9. Sur le système IIS, ouvrez les règles de sécurité locales du serveur.
   a. Cliquez sur **Démarrer** > **Exécuter** et entrez `secpol.msc`.
   b. Développez les règles locales et accédez à la section Attribution des droits utilisateur.
   c. Ouvrez la stratégie intitulée **Ouverture de session en tant que service**.

      Il est à noter que n'importe quel compte ou groupe figurant dans cette liste peut ouvrir une session en tant que service.
   d. Cliquez sur **Ajouter un utilisateur ou un groupe**.
   e. Entrez (ou recherchez) le compte du nom de domaine `iisuser`.
   f. Une fois les droits accordés, réamorcez le serveur.

      Le réamorçage du système est nécessaire, car les paramètres de sécurité sont appliqués durant la phase de démarrage de toute machine Windows 2003 Server.
10. Dans le système IIS, sélectionnez **Démarrer** > **Programmes** > **Outils d'administration** > **Gestionnaire IIS (Internet Information Service)**.
    a. Ouvrez l'ordinateur local.
    b. Cliquez avec le bouton droit sur DefaultAppPool.
    c. Sélectionnez **Recycler** pour redémarrer le pool.
11. Ouvrez un navigateur et accédez à `http://`*`serveur_web`*.

    S'il s'agit d'un nouveau serveur IIS dépourvu de contenu, la page `Under Construction` du serveur doit normalement s'afficher. Lorsqu'un contenu est présent sur le serveur IIS, il convient que celui-ci soit visible.
12. Dans le système IIS, sélectionnez **Démarrer** > **Programmes** > **Outils d'administration** > **Gestionnaire IIS (Internet Information Service)**.
    a. Cliquez avec le bouton droit sur le site Web par défaut.
    b. Sélectionnez Propriétés, puis l'onglet Sécurité de répertoire.
    c. cliquez sur le bouton **Modifier** en regard de l'entrée `Permettre l'accès anonyme` et modifiez les messages d'authentification liés à cette ressource.
    d. Désactivez l'accès anonyme.
    e. Activez l'authentification Windows intégrée
    f. Cliquez sur **OK**.
    g. Cliquez à nouveau sur **OK**.

13. Ouvrez votre navigateur et accédez à http://*serveur_web*. Vous êtes invité à vous connecter.
14. Entrez un utilisateur de domaine valide. Exemple : user@mydomain.com. Lorsque la connexion aboutit, vous pouvez visualiser le contenu IIS.

Chapitre 37. Configuration d'Active Directory et WebSphere pour la délégation contrainte

Vous devez configurer Active Directory et WebSphere pour que votre délégation Kerberos fonctionne.

Pourquoi et quand exécuter cette tâche

L'agent de noeud WebSphere qui héberge le composant d'exécution de Tivoli Federated Identity Manager doit être exécuté via un compte spécial d'Active Directory pour bénéficier de la permission d'obtenir des tickets Kerberos destinés à d'autres utilisateurs, ainsi qu'une série de cibles restreintes. Pour que votre chaîne d'accréditation de délégation Kerberos puisse fonctionner, vous devez exécuter les tâches suivantes :

- Créez le compte.
- Définissez les options appropriées.
- Modifiez le service WebSphere pour qu'il utilise le compte.

Les instructions suivantes décrivent la manière dont vous devez accomplir ces tâches.

Procédure

1. Vérifiez que le serveur DNS est correctement configuré sur le contrôleur de domaine Active Directory.

   Le serveur DNS doit être configuré à la fois en vue de permettre les recherches incursives et récursives. Chaque hôte du domaine Active Directory doit être configuré en vue d'utiliser l e serveur DNS du contrôleur de domaine.

   Pour vérifier ceci, exécutez les commandes **nslookup** à la fois sur le nom d'hôte et l'adresse IP de l'ordinateur membre du domaine. Il convient que le résultat des commandes **nslookup** indiquent que la partie correspondant au domaine dans le nom résolu est celui du contrôleur de domaine.
2. Assurez-vous que les services de synchronisation des horloges sont actifs sur toutes les machines situées dans le domaine Active Directory et que les horloges de toutes ces machines sont synchronisées.
3. Vérifiez que le système Windows Server 2003 (ou les systèmes multiples, en cas de déploiement dans un cluster WebSphere) est configuré dans le domaine Active Directory. Le serveur peut, en option, être configuré en tant que contrôleur de domaine.
4. Vérifiez que tous les contrôleurs du domaine soient exécutés au niveau fonctionnel de Windows Server 2003. Pour ce faire, procédez comme suit :
   a. Ouvrez le panneau de configuration des utilisateurs et ordinateurs d'Active Directory.
   b. Cliquez sur le nom du domaine avec le bouton droit et sélectionnez **Augmenter le niveau fonctionnel du domaine**.
   c. Sélectionnez `Windows Server 2003` et cliquez sur **OK**.

   La fenêtre Augmenter le niveau fonctionnel du domaine s'affiche. Elle contient normalement les messages suivants :

```
Niveau fonctionnel du domaine actuel
Windows Server 2003

Ce domaine fonctionne au niveau fonctionnel le plus élevé possible.
```

5. Sur le contrôleur de domaine, créez un utilisateur pour la délégation dans Active Directory. Le serveur WebSphere qui héberge le composant d'exécution de Tivoli Federated Identity Manager s'exécute en tant qu'identité pour cet utilisateur.
   a. Créez un utilisateur. Par exemple, `tfimdeleguser`. Vous pouvez indiquer une autre identité d'utilisateur. Ce nom d'utilisateur sera utilisé dans les présentes instructions.
   b. Cochez la case **Le mot de passe n'expire jamais**.

      Remarque : Vous pouvez, en option, définir l'expiration du mot de passe. Si tel est le cas, lorsque vous le modifierez le moment venu, vous devrez également réinitialiser le mot de passe de l'agent de noeud WebSphere dans le service Windows.
6. Sur le contrôleur de domaine, ajoutez l'utilisateur `tfimdeleguser` au groupe d'administrateurs du domaine. Pour vérifier les paramètres :
   a. Sélectionnez **Utilisateurs et ordinateur Active Directory**.
   b. Pour le domaine, cliquez sur **Utilisateurs**, puis sur **Administrateurs de domaine**.
   c. Sélectionnez l'onglet **Membres**. Vérifiez que l'utilisateur tfimdeleguser figure parmi la liste des membres du groupe.
7. Assurez-vous que les outils Microsoft Support Tools sont installés sur le contrôleur de domaine. A titre d'exemple, pour obtenir les outils de support 32 bits pour Windows Server 2003 Service Pack 1 :

   http://www.microsoft.com/downloads/details.aspx?FamilyId=6EC50B78-8BE1-4E81-B3BE-4E7AC4F0912D&displaylang=en
8. Sur le contrôleur de domaine, créer un nom principal de service (SPN) pour l'utilisateur `tfimdeleguser`. Pour ce faire, procédez comme suit :
   a. Ouvrez une invite de commande sur le contrôleur de domaine sur lequel les outils de support sont installés.
   b. Entrez la commande **setspn**.

      La syntaxe de la commande est la suivante :

      `setspn -A tfim/<utilisateur_délégation_tfim> <utilisateur_délégation_tfim>`

      Par exemple :

      ```
      setspn -A tfim/tfimdeleguser tfimdeleguser
      ```
9. Sur le contrôleur de domaine, ouvrez **Utilisateurs et ordinateurs Active Directory** et accédez aux propriétés de l'utilisateur `tfimdeleguser`.
   a. Sélectionnez l'onglet **Délégation**.

      Remarque : Si l'onglet **Délégation** n'apparaît pas, revenez à l'étape précédente et assurez-vous que la commande setspn a abouti.
   b. Sélectionnez l'option **N'approuver cet utilisateur que pour la délégation aux services spécifiés**.
   c. Sélectionnez le bouton d'option **Utiliser n'importe quel protocole d'authentification**.
   d. Cliquez sur le bouton **Ajouter** dans l'onglet Délégation.

e. Ajoutez les services cibles pouvant être délégués à tfimdeleguser. Il s'agit des services cible de la délégation contrainte. Dans cet exemple, le serveur IIS Web est exécuté en tant qu'utilisateur.

f. Cliquez sur le bouton **Utilisateurs ou ordinateurs** pour rechercher des services particuliers.

g. Sélectionnez l'utilisateur du domaine (service) sous lequel le serveur IIS est exécuté pour la jonction Kerberos WebSEAL.

Lorsque vous avez terminé, l'onglet **Délégation** doit indiquer un service cible dans la fenêtre **Services auxquels ce compte peut présenter des données d'identification déléguée**.

La fenêtre peut, par exemple, indiquer un **Type de service** `HTTP`, tandis que la section **Utilisateur ou ordinateur** indique un nom d'hôte ou de domaine, tel que `mydataserver.example.com`

Sélectionnez l'entrée `HTTP/mydataserver.example.com`. Appuyez sur **OK** pour continuer.

10. Ajoutez tfimdeleguser aux groupes d'accès d'autorisation Windows. Pour ce faire, procédez comme suit :

a. Ouvrez le panneau **Utilisateurs et ordinateurs Active Directory**.

b. Sélectionnez l'objet **Builtin** sous le domaine.

c. Recherchez l'objet **Groupes d'accès d'autorisation Windows**.

d. Cliquez avec le bouton droit et sélectionnez **Propriétés**. Sélectionnez l'onglet **Membres**.

e. Cliquez sur **Ajouter** et ajoutez l'utilisateur de la délégation (dans notre exemple, il s'agit de `tfimdeleguser`) en tant que membre.

11. Octroyez à l'utilisateur de la délégation (tfimdeleguser) les privilèges **Agir en tant que partie du système d'exploitation**.

Le processus effectif qui doit être exécuté en tant que service Windows dépend de l'environnement WebSphere :

- Le nom du service défini dans un environnement *autonome* correspond à l'instance WebSphere Application Server qui héberge le composant d'exécution de Tivoli Federated Identity Manager
- Le nom du service défini dans un environnement *groupé* est le serveur WebSphere Application qui exécute l'agent de noeud WebSphere pour le composant d'exécution de Tivoli Federated Identity Manager.

Remarque : Dans un environnement groupé, cette étape doit être répétée sur toutes les machines hébergeant un membre de noeud du cluster WebSphere qui exécute le composant d'exécution Tivoli Federated Identity Manager. Pour ce faire, procédez comme suit :

a. Accédez au menu approprié suivant votre déploiement :
   - Sur le contrôleur de domaine, sélectionnez **Démarrer -** > **Programmes** > **Outils d'administration -** > **Stratégie de sécurité du domaine**.
   - Sur une machine autre que le contrôleur de domaine, sélectionnez **Démarrer** > **Programmes** > **Outils d'administration** > **Stratégie de sécurité locale**.

b. Développez l'option Stratégies locales.

c. Sélectionnez **Attribution des droits utilisateur** > **Agir en tant que partie du système d'exploitation**

d. Cliquez avec le bouton droit et sélectionnez **Propriétés**.

e. Cochez la case **Définir les paramètres de cette stratégie**.

f. Cliquez sur **Ajouter un utilisateur ou un groupe** afin d'ajouter l'utilisateur de la délégation (tfimdeleguser) à la liste des utilisateurs autorisés à agir en tant que partie du système d'exploitation.

g. Cliquez sur **OK**.

12. Octroyez à l'utilisateur de la délégation (tfimdeleguser) les privilèges nécessaires :

    - Lorsque l'application Tivoli Federated Identity Manager est exécutée sur un membre du domaine, octroyez à l'utilisateur les privilèges **Ouvrir une session en tant que service** sur la machine locale.
    - Lorsque l'application Tivoli Federated Identity Manager est exécutée sur le contrôleur de domaine, octroyez à l'utilisateur les privilèges **Ouvrir une session en tant que service** sur ce contrôleur de domaine.

    a. Revenez au menu des stratégies de sécurité ouvert à l'étape précédente.

    b. Sélectionnez **Attribution des droits utilisateur** > **Ouvrir une session en tant que service**.

    c. Cliquez avec le bouton droit et sélectionnez **Propriétés**.

    d. Cochez la case **Définir les paramètres de cette stratégie**.

    e. Cliquez sur **Ajouter un utilisateur ou un groupe** afin d'ajouter l'utilisateur de la délégation (tfimdeleguser) à la liste des utilisateurs autorisés à agir en tant que partie du système d'exploitation.

    f. Cliquez sur **OK**.

13. Permettez au processus WebSphere qui exécute l'application Tivoli Federated Identity Manager de s'exécuter en tant que service Windows.

    Exécutez la commande **wasservice**. Emplacement par défaut :

    ```
    C:\Program Files\IBM\WebSphere\AppServer\bin
    ```

    Exemple de commande :

    ```
    C:\Program Files\IBM\WebSphere\AppServer\bin>wasservice -add ndagentwinser
    -servername nodeagent
    -profilePath "C:\Program Files\IBM\WebSphere\AppServer\profiles\Custom01"
    -wasHome "C:\Program Files\IBM\WebSphere\AppServer"
    -logfile "c:\Program Files\IBM\WebSphere\AppServer\profiles\
       Custom01\logs\ws_startserver.log"
    -logRoot "c:\Program Files\IBM\WebSphere\AppServer\profiles\
       Custom01\logs\nodeagent"
    -restart true
    ```

    Exemple de sortie de la commande :

    ```
    Adding Service: ndagentwinser
      Config Root:
      C:\Program Files\IBM\WebSphere\AppServer\profiles\Custom01\config
      Server Name: nodeagent
      Profile Path: C:\Program Files\IBM\WebSphere\AppServer\profiles\Custom01
      Was Home: C:\Program Files\IBM\WebSphere\AppServer\
      Start Args:
      Restart: 1
    IBM WebSphere Application Server V6.1
       - ndagentwinser service successfully added
    ```

    Pour obtenir un message relatif à la syntaxe de la commande **wasservice**, entrez :

    ```
    > WASService.exe
    ```

    sans spécifier d'argument.

14. Si l'exécution a lieu dans un environnement groupé, modifiez le service WebSphere depuis l'étape précédente afin qu'il démarre en tant qu'utilisateur de la délégation (tfimdeleguser)

a. Ouvrez les **Services** dans le Panneau de configuration et recherchez le service correspondant soit au composant d'exécution Tivoli Federated Identity Manager, soit à l'agent de noeud du composant d'exécution Tivoli Federated Identity Manager dans le cas d'un environnement groupé.
b. Sélectionnez l'onglet **Connexion**.
c. Indiquez l'utilisateur de la délégation `tfimdeleguser`.
d. Indiquez le mot de passe de l'utilisateur de la délégation.
e. Cliquez sur **OK**.

15. Redémarrez l'agent de noeud WebSphere.

    Cette étape est nécessaire pour assurer que le gestionnaire de noeud Websphere démarre bien les noeuds sous la nouvelle identité.

    a. Connectez-vous à la console WebSphere.
    b. Sélectionnez **Serveurs** > **Serveurs d'applications** dans le cas d'un environnement autonome ou **Serveurs** > **Clusters** dans le cas d'un environnement cluster.
    c. Cochez la case définissant le redémarrage du serveur ou du cluster et appuyez sur le bouton **Arrêt** pour un environnement autonome, ou sur le bouton **Démarrage en cascade** pour un environnement groupé.
    d. Dans un environnement autonome, une fois que l'arrêt du serveur a eu lieu, cochez la case définissant le redémarrage du serveur ou du cluster et appuyez sur le bouton **Démarrer**.

Que faire ensuite

Autres informations :

- Principes de configuration de Microsoft :

  http://technet2.microsoft.com/windowsserver/en/library/c312ba01-318f-46ca-990e-a597f3c294eb1033.mspx?mfr=true
- Instructions de configuration :

  http://technet2.microsoft.com/windowsserver/en/library/e5d4cdbd-f071-4a1a-b24e-92713f7fafc11033.mspx?mfr=true
- Instructions IBM pour la configuration de l'exécution de WebSphere en tant que compte autre que **Système local**.

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.base.doc/info/aes/ae/tsec_actwindows.html

Chapitre 38. Configuration de Tivoli Federated Identity Manager pour un scénario de jonction Kerberos

Avant de configurer la délégation Kerberos, assurez-vous d'avoir créé un domaine, tel que décrit au Chapitre 3, «Configuration de domaine», à la page 25.

Etapes de configuration :

1. «Planification de configuration de la chaîne d'accréditation»
2. Exécution de l'étape «Formulaire de configuration de chaîne d'accréditation», à la page 573
3. «Création d'une instance de module de délégation contrainte Kerberos», à la page 575
4. «Création d'une chaîne d'accréditation pour la délégation contrainte Kerberos», à la page 576

Planification de configuration de la chaîne d'accréditation

Planification de la configuration du déploiement d'une chaîne d'accréditation pour la délégation contrainte Kerberos.

Pour déployer une chaîne d'accréditation destinée à la délégation contrainte Kerberos, vous devez accomplir les deux tâches suivantes :

1. Créez une instance du module de service d'accréditation pour la délégation contrainte Kerberos.
2. Créez une chaîne d'accréditation pour la délégation contrainte Kerberos.

Tivoli Federated Identity Manager fournit des assistants de configuration pour chaque tâche. Les assistants vous invitent à spécifier les valeurs des propriétés de configuration requises.

Instance de module de délégation Kerberos

L'ensemble des modules d'accréditation par défaut de Tivoli Federated Identity Manager n'inclut aucune instance du type de module de délégation contrainte Kerberos. Vous devez créer cette instance.

Bien que la création de plusieurs instances soit possible, il convient de n'en créer qu'une seule pour chaque domaine Tivoli Federated Identity Manager. Cette instance peut être utilisée par n'importe quelle chaîne de modules requise.

Les raisons de cette limitation à une seule instance est liée au fait que ce module de délégation contrainte Kerberos charge une bibliothèque DLL native (bibliothèque Windows chargée dynamiquement) qui est partagée par toutes les instances du module. Toutes les instances partagent les mêmes paramètres de configuration.

Lorsque plusieurs instances de module sont créées, seul le *dernier* module à être initialisé détermine la taille de la mémoire cache utilisateur créée dans le code natif. Pour éviter toute confusion, la meilleure pratique consiste à ne créer qu'une seule instance de module.

Type de module
Cette propriété obligatoire est demandée dans le panneau relatif au type de module. Le type de module à utiliser est le suivant :

```
com.tivoli.am.fim.trustserver.sts.modules.KerberosDelegationSTSModule
```

Nom de l'instance de module
Cette propriété obligatoire est demandée dans le panneau relatif au nom de l'instance de module. Entrez une chaîne de votre choix. Par exemple :

```
MyKerberosDelegationInstance
```

Description de l'instance de module
Cette propriété facultative est demandée dans le panneau relatif au nom de l'instance de module. Vous pouvez entrer une chaîne qui décrit cette instance.

Taille maximale de la mémoire cache des données d'identification de l'utilisateur

Cette propriété obligatoire est demandée dans le panneau de configuration du module de délégation Kerberos. Ce nombre détermine les indicateurs de personnification et les données d'identification de l'utilisateur placées en cache dans le fichier DLL chargé par le module.

La mise en cache est effectuée dans le but d'améliorer les performances. Définissez cette valeur sur le nombre approximatif attendu d'utilisateurs finals simultanés du service pour les transactions de gros volumes.

Le paramètre par défaut est 100.

Remarque : Plus ce chiffre est élevé, plus la quantité de mémoire utilisable par le module d'exécution de Tivoli Federated Identity Manager est importante.

Chaîne d'accréditation de délégation Kerberos

Nom du mappage de chaîne

Cette propriété obligatoire est demandée dans le panneau relatif à l'identification du mappage de chaîne. Vous pouvez indiquer n'importe quel nom pour la chaîne. Par exemple :

```
ivcred_to_kerberos
```

Description de chaîne
Cette propriété facultative est demandée dans le panneau relatif à l'identification du mappage de chaîne. cette description peut correspondre à n'importe quelle chaîne de caractères.

Créer une chaîne dynamique
Cette propriété est demandée dans le panneau relatif à l'identification du mappage de chaîne. Cette option n'est pas utilisée avec les chaînes d'accréditation de délégation Kerberos. Désélectionnez cette option.

Type de requête
Cette propriété obligatoire est demandée dans le panneau relatif à la recherche du mappage de chaîne. Sélectionnez **Issue Oasis URI**.

Type de recherche
Sélectionnez le bouton d'option **Utiliser les éléments WS-Trust habituels (AppliesTo, Issuer et Token)**.

(AppliesTo) Adresse
Cette propriété obligatoire est demandée dans le panneau relatif à la

recherche du mappage de chaîne. Entrez une **Adresse** qui corresponde à la propriété **applies-to** définie dans la section [tfimsso:*jct_name*du fichier de configuration WebSEAL. Par exemple :

```
http://websealhost.example.com/kerbjct
```

(AppliesTo) Nom du service
Cette propriété obligatoire est demandée dans le panneau relatif à la recherche du mappage de chaîne.

Cette propriété comporte deux zones.

Dans la première zone, définissez un astérisque (*) pour que cette valeur renvoie à tous les noms de service, ou spécifiez la valeur de la propriété service-name dans la section [tfimsso:*jct name*] du fichier de configuration WebSEAL.

Dans la seconde zone, définissez toujours un astérisque (*) pour cette valeur.

(AppliesTo) Type de port
Cette propriété est demandée dans le panneau relatif à la recherche du mappage de chaîne.

Cette propriété admet deux zones.

Laissez les deux zones vides.

(Issuer) Adresse
Cette propriété obligatoire est demandée dans le panneau relatif à la recherche du mappage de chaîne. Dans la zone **Adresse**, entrez :

```
amwebrte-sts-client
```

(Issuer) Nom du service
Cette propriété est demandée dans le panneau relatif à la recherche du mappage de chaîne. Ne renseignez pas cette zone.

(Issuer) Type de port
Cette propriété est demandée dans le panneau relatif à la recherche du mappage de chaîne. Ne renseignez pas cette zone.

Type de jeton
Cette propriété obligatoire est demandée dans le panneau relatif à la recherche du mappage de chaîne. Sélectionnez **Kerberos GSS V5**.

Initialiser la chaîne au démarrage de l'exécution
Cette propriété obligatoire est demandée dans le panneau relatif à l'identification de chaîne. *Ne sélectionnez pas* cette option

Instances de modules et modes
Ces propriétés obligatoires sont demandées dans le panneau Assemblage de chaîne.

Le panneau Assemblage de chaîne vous invite à spécifier les valeurs des Instances de module dans la chaîne. Pour chaque instance de module, vous devez sélectionner un mode. Ensuite, vous cliquez sur un bouton pour ajouter la paire de valeurs instance/mode à la chaîne.

Dans le cas de la délégation contrainte Kerberos, vous pouvez configurer une séquence particulière de modules de service d'accréditation :

1. La première instance de module est **Jeton IVCred par défaut**. Choisissez un mode **validation**

2. La seconde instance est l'instance de module de la délégation que vous avez créée, et qui est nommée d'après la propriété **Nom d'instance de module** via l'assistant correspondant. Par exemple :

   ```
   MyKerberosDelegationInstance
   ```

   Sélectionnez l'onglet **Emission**.

Remarque : L'assistant vous avertira que la chaîne ne contient aucune module en mode **mappage**. Pour une délégation contrainte Kerberos, le mode mappage n'est pas obligatoire.

vous pouvez ajouter un mode de mappage si votre déploiement le nécessite. Un module de mappage est nécessaire su le nom d'utilisateur de Tivoli Access Manager doit être mappé avec un autre nom du registre Active Directory.

Dans un déploiement typique, ce mappage n'est pas obligatoire. Dans de nombreux déploiements, par exemple, Tivoli Access Manager est installé de manière à utiliser le registre Active Directory. Dans pareil cas, il n'existe qu'une identité par utilisateur.

Activer la validation des signatures
Cette propriété est demandée dans le panneau de configuration du module Access Manager Credential (IVCred). *Ne sélectionnez pas* cette option

Nom principal du service cible par défaut
Cette propriété est demandée dans le panneau de configuration du module de délégation Kerberos en tant que propriété de Partenaire.

Dans un déploiement typique, vous pouvez laisser cette zone vide.

cette valeur peut être utilisée pour les clients WS-Trust qui n'envoient pas le SPN (Service Principal Name) cible dans l'élément AppliesTo/ ServiceName du jeton RST (RequestSecurityToken). De plus, les clients ne disposent d'aucune règle de mappage pour configurer le SPN cible en tant qu'attribut de contexte pour l'utilisateur STSUU (Security Token Service Universal User).

Options permettant d'ajouter un nom d'utilisateur Tivoli Access Manager pour les besoins de l'authentification Kerberos
Ces options permettent d'indiquer si un module va automatiquement ajouter un suffixe au nom d'utilisateur STSUniversalUser. Elle sont utiles lors du déploiement du module de délégation Kerberos via un déploiement Tivoli Access Manager WebSEAL. Les options incluent :

- Ne pas ajouter de suffixe au nom d'utilisateur.

  Cette option laisse le nom d'utilisateur tel quel.

- Ajouter le domaine DNS de la machine en tant que suffixe au nom d'utilisateur.

  Cette option ajoute automatiquement le suffixe du domaine DNS pour la machine d'exécution Tivoli Federated Identity Manager au nom principal dans STSUniversalUser avant d'appeler l'API Windows pour obtenir un ticket Kerberos. Le nom de domaine DNS est lu à partir de la clé de registre Windows suivante :

  ```
  SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Domain
  ```

  Cette option optimise le comportement du module en vue d'une utilisation dans les configurations Tivoli Access Manager utilisant les jonctions Kerberos. L'ajout du domaine DNS permet à l'API Windows

d'établir une correspondance correcte entre le nom d'utilisateur et l'enregistrement utilisateur du registre d'utilisateurs Active Directory.

Notez que ce module ajoute automatiquement le nom de domaine DNS lorsque le nom principal STSUniversalUser ne contient *pas* déjà le caractère @. Cela signifie que si une règle de mappage a été utilisée pour ajouter un suffixe contenant le caractère @ au nom principal d'utilisateur ou si le nom d'utilisateur Tivoli Access Manager contient le caractère @, ce paramètre n'a aucun effet.

- Ajouter le suffixe configuré au nom d'utilisateur

  Cette option optimise le comportement du module en vue d'une utilisation dans les configurations Tivoli Access Manager utilisant les jonctions Kerberos.

  Cette option permet à l'administrateur d'indiquer le suffixe manuellement. Cette option est réservée aux cas particuliers où l'attribut userPrincipalName de l'utilisateur ne correspond pas au nom de domaine DNS de la machine Windows exécutant l'environnement d'exécution de Tivoli Federated Identity Manager. Cette option n'a aucun effet lorsque le nom principal contient déjà le caractère @.

  Suffixe à ajouter en cas d'utilisation d'un suffixe configuré
  Par exemple :

  ```
  @mondomaine.com
  ```

Formulaire de configuration de chaîne d'accréditation

Complétez les formulaires avant de configurer la chaîne d'accréditation.

Les propriétés des formulaires sont décrites à la rubrique «Planification de configuration de la chaîne d'accréditation», à la page 569.

Formulaire d'instance de module Kerberos

Les tableaux suivants correspondent aux panneaux présentés par l'assistant de création des instances de module.

Tableau 131. Propriétés des panneaux d'identification de module

Propriété	Valeur
Type de module	com.tivoli.am.fim.trustserver.sts.modules.KerberosDelegationSTSModule
Nom de l'instance de module	
Description de l'instance de module	

Tableau 132. Propriété du panneau pour la configuration du module de délégation Kerberos

Propriété	Votre valeur
Taille maximale des données d'identification de l'utilisateur	Valeur par défaut : 100

Formulaire de chaîne d'accréditation pour le module Kerberos

L'assistant de chaîne d'accréditation affiche une série de panneaux de configuration. Les tableaux suivants correspondent à chaque panneau.

Tableau 133. Propriétés d'identification de mappage de chaîne

Propriété	**Votre valeur**
Nom du mappage de chaîne	
Description de chaîne	
Créer une chaîne dynamique	*Cette option doit être désélectionnée*

Tableau 134. Propriétés de recherche de mappage de chaîne

Propriété	**Votre valeur**
Type de requête	Emettre URI Oasis
Type de recherche	Utiliser les éléments WS-Trust habituels (AppliesTo, Issuer et TokenType)
(AppliesTo) Adresse	
(AppliesTo) Nom du service	*Deux zones* Utilisez un astérisque (*) pour chaque zone
(AppliesTo) Type de port	*Deux zones* Laissez les deux zones vides
(Issuer) Adresse	
(Issuer) Nom du service	*Deux zones* Laissez les deux zones vides
(Issuer) Type de port	*Deux zones* Laissez les deux zones vides
Type de jeton	Kerberos GSS V5

Tableau 135. Panneau d'identification de chaîne

Propriété	**Votre valeur**
Initialiser la chaîne au démarrage de l'exécution	*Ne sélectionnez pas cette option*

Tableau 136. Panneau d'assemblage de chaîne

Propriété	**Votre valeur**
Première instance de module	Jeton IVCred par défaut
Premier mode de module	valider
Seconde instance de module	*Nom de l'instance de module Kerberos*:
Second mode de module	Issue (Emission)

Tableau 137. Propriété de configuration du module Tivoli Access Manager Credential

Propriété	Votre valeur
Activer la validation des signatures	*Désélectionnez cette option*

Tableau 138. Propriété de configuration (mode Emission) du module de délégation Kerberos

Propriété	Votre valeur
Nom principal du service cible par défaut	
Options pour l'ajout d'un nom d'utilisateur Tivoli Access Manager pour l'authentification Kerberos : • Ne pas ajouter de suffixe au nom d'utilisateur. • Ajouter le domaine DNS de la machine en tant que suffixe au nom d'utilisateur. • Ajouter le suffixe configuré au nom d'utilisateur **Suffixe à ajouter en cas d'utilisation d'un suffixe configuré** Par exemple : `@mondomaine.com`	

Création d'une instance de module de délégation contrainte Kerberos

Apprenez à créer une instance de module pour une délégation contrainte Kerberos.

Pourquoi et quand exécuter cette tâche

Un assistant vous guide tout au long de la création de l'instance de module. Pour plus d'informations sur chacune des propriétés requises, voir «Planification de configuration de la chaîne d'accréditation», à la page 569.

vous pouvez également consulter le document «Formulaire de configuration de chaîne d'accréditation», à la page 573.

Procédure

1. Connectez-vous à la console WebSphere.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer Service d'accréditation** > **Instances de module**. Le portlet Instances de module s'ouvre.
3. Cliquez sur **Créer**. L'assistant Instance de module démarre et le panneau Type de module s'affiche.
4. Sélectionnez **com.tivoli.am.fim.trustserver.sts.modules.KerberosDelegationSTSModule**.
5. Cliquez sur **Suivant**. Le panneau Nom d'instance du module s'ouvre.
6. Entrez un nom dans la zone **Nom d'instance du module**.

   Par exemple :

   `Jonction Kerberos`

7. Si vous le souhaitez, vous pouvez ajouter une description dans la zone **Description de l'instance de module**.
8. Cliquez sur **Suivant**. Le panneau Configuration du module de délégation Kerberos s'affiche.
9. Entrez une valeur dans la zone **Taille maximale de la mémoire cache des données d'identification de l'utilisateur**.
10. Cliquez sur **Terminer**. Le panneau Instances de module s'ouvre. Le portlet Domaine en cours s'affiche également et vous invite à spécifier les nouvelles modifications de la configuration.
11. Cliquez sur le bouton **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.
12. Passez à l'étape «Création d'une chaîne d'accréditation pour la délégation contrainte Kerberos».

Création d'une chaîne d'accréditation pour la délégation contrainte Kerberos

Vous devez créer une chaîne d'accréditation et configurer ses propriétés pour la délégation contrainte Kerberos à l'aide de l'assistant de chaîne d'accréditation.

Avant de commencer

Le domaine doit contenir une instance du module de service d'accréditation pour la délégation contrainte Kerberos avant que la chaîne d'accréditation ne soit créée. Si vous n'avez pas encore créé d'instance, procédez à cette opération maintenant. Voir «Création d'une instance de module de délégation contrainte Kerberos», à la page 575.

Pourquoi et quand exécuter cette tâche

Pour configurer correctement la chaîne d'accréditation, vous devez vous assurer que les propriétés sont alignées sur les propriétés de configuration WebSEAL. Avant d'exécuter l'assistant de chaîne d'accréditation, il convient de d'accomplir les tâches suivantes :

- consultez la rubrique «Planification de configuration de la chaîne d'accréditation», à la page 569
- Exécutez la procédure indiquée à la rubrique «Formulaire de configuration de chaîne d'accréditation», à la page 573

Procédure

1. Connectez-vous à la console WebSphere.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer Service d'accréditation** > **Chaînes du service d'accréditation**. Le portlet Chaînes du service d'accréditation s'affiche.
3. Cliquez sur **Créer**. L'assistant de configuration s'ouvre.
4. Cliquez sur **Suivant**. Le panneau Identification du mappage de chaîne s'ouvre.
5. Entrez les valeurs demandées.
   a. Entrez un nom dans la zone **Nom du mappage de chaîne**.
   b. Si vous le souhaitez, vous pouvez ajouter une description dans la zone **Description**.
   c. *Ne sélectionnez pas* la zone **Créer une chaîne dynamique**

d. Cliquez sur **Suivant**. Le panneau **Recherche de mappage de chaîne** s'ouvre.

6. Entrez les valeurs demandées.
   a. Définissez le **Type de requête** sur **Emettre URI Oasis**.

      La valeur d'URI du type de requête correspondante est automatiquement entrée par l'assistant.
   b. Définissez l'option **Type de recherche** sur **Utiliser les éléments WS-Trust habituels (AppliesTo, Issuer et TokenType)**.
   c. Entrez des valeurs dans la section **AppliesTo**.
      - Entrez une **adresse**.

        Par exemple :

        `http://websealhost.example.com/krbjct`
      - Entrez le **nom du service**.

        Définissez par exemple les deux zones avec un astérisque (*).
      - Ne renseignez pas les zones **Type de port**.

      Pour obtenir de l'aide, voir «Planification de configuration de la chaîne d'accréditation», à la page 569.
   d. Entrez des valeurs dans la section **Emetteur**.
      - Dans la zone **Adresse**, entrez :

        `amwebrte-sts-client`
      - N'indiquez aucune valeur dans les zones **Nom du service** et **Type de port**.
   e. Pour le **Type de jeton**, sélectionnez **Kerberos GSS V5**.
   f. Cliquez sur **Suivant**.

      Le panneau Identification de chaîne s'affiche.
7. *Ne sélectionnez pas* l'option **Initialiser la chaîne au démarrage de l'exécution**. Cliquez sur **Suivant**.

   Le panneau **Assemblage de chaîne** s'ouvre.
8. Créez la chaîne d'accréditation :
   a. Pour l'instance de module, sélectionnez l'option **Jeton IVCred par défaut**.
   b. Pour le mode, choisissez **validation**
   c. Cliquez sur **Ajouter l'instance de module sélectionnée à la chaîne**.
   d. Pour l'instance de module, sélectionnez le nom d'instance de module que vous avez spécifié à la rubrique «Création d'une instance de module de délégation contrainte Kerberos», à la page 575. Par exemple :

      `Jonction Kerberos`
   e. Pour le mode, choisissez **émission**
   f. Cliquez sur **Ajouter l'instance de module sélectionnée à la chaîne**.
9. Cliquez sur **Suivant**.

   Remarque : Un avertissement s'affiche pour vous signaler qu'un module est manquant dans votre chaîne en mode mappage. Vous pouvez ignorer cet avertissement. Pour plus d'informations, voir «Planification de configuration de la chaîne d'accréditation», à la page 569.

   L'écran Configuration du module de droits d'accès Access Manager (IVCred) s'affiche.
10. *Ne sélectionnez pas* l'option **Activer la validation des signatures**. Cliquez sur **Suivant**.

Le panneau Configuration du module de délégation Kerberos s'affiche.

11. Si nécessaire, définissez le nom principal du service cible par défaut ou changez les options relatives à l'ajout d'un suffixe au nom d'utilisateur Tivoli Access Manager pour l'authentification Kerberos.

    Remarque : Dans la plupart des cas, vous pouvez laisser cette zone vide et conserver la sélection par défaut de ces options. Voir «Planification de configuration de la chaîne d'accréditation», à la page 569.
12. Cliquez sur **Suivant**. Le panneau Récapitulatif s'affiche.
13. Cliquez sur **Terminer**.
14. Dans le portlet Domaine en cours, cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Résultats

La configuration de la chaîne d'accréditation est à présent terminée.

Remarques sur la configuration de Tivoli Federated Identity Manager

La configuration d'un scénario de jonction Kerberos peut vous obliger à vérifier certains paramètres de configuration. Cette section contient des remarques sur les éléments que vous devez vérifier.

Vérification de la configuration de la chaîne d'accréditation de Tivoli Federated Identity Manager

Vérification de l'aptitude du gestionnaire de déploiement WebSphere à communiquer avec l'instance WebSphere Application Server hébergeant Tivoli Federated Identity Manager.

Pour ce faire, accédez à l'adresse URL :

```
http://<serveur_IHS>/TrustServerWST13/RequestSecurityToken
```

Un modèle de réponse semblable au suivant s'affiche :

```
RequestSecurityToken ... Hi there this is an AXIS service!
 Perhaps there will be a form for invoking the service here...
```

Vérification des mappages de modules WebSphere

Assurez-vous que les mappages de module WebSphere Application Server et des hôtes virtuels ont été propagés. Pour ce faire, accédez à l'adresse URL :

```
http://<serveur_IHS>/Info/InfoService
```

Un modèle de réponse semblable au suivant s'affiche :

```
Hi there this is a Web service!
```

Haute disponibilité dans une configuration en clusters

De multiples serveurs WAS sont déployés dans un cluster WAS afin d'obtenir une haute disponibilité. Les noeuds WAS individuels membres du cluster reçoivent des instructions de configuration en provenance d'un gestionnaire de déploiement.

La plupart des tâches administratives sont accomplies en communiquant avec le gestionnaire de déploiement. Cependant, tous les flux de protocole nécessaires aux demandes de service pour le service d'accréditation TFIM sont servis par les noeuds WAS individuels. En cas de défaillance du gestionnaire de déploiement, ces flux de protocole n'en subissent aucune conséquence. En revanche, en cas de défaillance des noeuds WAS, un impact a lieu sur le flux de protocole.

Chapitre 39. Configuration de WebSEAL

Vous devez installer et configurer le serveur de règles Tivoli Access Manager avant d'installer WebSEAL.


Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Ces instructions supposent que vous avez installé et configuré avec succès le serveur de règles.

Effectuez une installation standard de WebSEAL. La procédure exacte dépend de votre environnement de déploiement. Pour plus d'informations, voir le document *IBM Tivoli Access Manager Installation Guide*.

Présentation des tâches :

1. «Vérification d'une installation WebSEAL»
2. «Planification de la configuration des jonctions WebSEAL Kerberos», à la page 582
3. Remplissage d'un formulaire «Formulaire de configuration de jonction Kerberos», à la page 587
4. «Débogage d'une jonction WebSEAL Kerberos», à la page 587

Vérification d'une installation WebSEAL

La présente rubrique explique comment vérifier que la configuration de base du serveur WebSEAL est correcte, afin que vous puissiez étendre la configuration à la prise en charge des jonctions Kerberos.

Avant de commencer


Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Ces instructions supposent que vous avez installé et configuré IBM Tivoli Access Manager for e-business. Elles supposent également que vous avez installé et configuré avec succès le serveur WebSEAL.

Pourquoi et quand exécuter cette tâche

Pour vérifier la configuration de base, créez une jonction WebSEAL normale et vérifiez que Tivoli Access Manager affiche correctement une invite de connexion utilisateur.

Procédure

1. Obtenez le nom du serveur WebSEAL.

   Le nom de serveur dépend du nom d'hôte. Si par exemple le nom d'hôte est `websealhost` :

   ```
   pdadmin sec_master> server list
     default-webseald-websealhost
   ```

2. Créez une jonction simple.

   A titre d'exemple, lorsque le serveur protégé est `mydataserver`, la commande suivante crée une jonction sur `/jct` :

   ```
   pdadmin sec_master> server task default-webseald-websealhost
    create -t tcp -h mydataserver/jct
   ```

3. Obtenez la liste de valeurs de l'objet /WebSEAL.

   Cette valeur est nécessaire pour joindre correctement une liste de contrôle d'accès (ACL) :

   ```
   pdadmin sec_master> object list /WebSEAL
     /WebSEAL/websealhost-default
   ```

4. Rattachez une liste de contrôle d'accès à la nouvelle jonction.

   La liste de contrôle d'accès a pour rôle de contrôler les actions pouvant être exécutées par les utilisateurs spécifiés au sein de l'espace d'objets protégés de Tivoli Access Manager. Cette étape suppose l'existence d'une liste de contrôle d'accès intitulée `testacl`.

   ```
   pdadmin sec_master> acl attach /WebSEAL/websealhost-default/testacl
   ```

5. Pour confirmer que la jonction et la liste de contrôle d'accès sont correctement configurées, procédez comme suit :

   a. Placez un fichier d'essai dans le répertoire documentRoot du serveur Web protégé.

      Par exemple, dans le répertoire documentRoot de `mydataserver`, créez un répertoire de test et ajoutez le fichier index.html pour afficher un contenu quelconque. A titre d'exemple, ajoutez le fichier suivant sous le point de jonction :

      ```
      /testdir/index.html
      ```

   b. Accédez au contenu protégé :

      ```
      https://websealhost.example.com/jct/testdir/index.html
      ```

   c. WebSEAL vous invite à vous connecter. Connectez-vous avec une identité et un mot de passe valides pour Tivoli Access Manager.

      Si l'opération aboutit, vous pouvez afficher le contenu de la page `testdir/index.html`.

Planification de la configuration des jonctions WebSEAL Kerberos

Avant de pouvoir configurer WebSEAL pour les jonctions Kerberos, vous devez déterminer les valeurs requises par votre déploiement pour chaque propriété.

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Les propriétés de configuration WebSEAL sont spécifiées dans le fichier de configuration WebSEAL. Le fichier de configuration par défaut est webseald-default.conf. Par exemple, sur les systèmes UNIX or Linux :

```
/opt/pdweb/etc/webseald-default.conf
```

Le fichier de configuration contient les propriétés qui prennent en charge le déploiement des jonctions Kerberos.

Les propriétés sont regroupées en deux sections :

```
[tfimsso:jct-id]
[tfim-cluster:cluster]
```

Dans certains cas, l'introduction d'une connexion unique Kerberos sur des serveurs comprenant des jonctions peut influer sur les performances. Chaque jeton Kerberos est valable uniquement pour une authentification Kerberos.

WebSEAL doit demander un nouveau jeton Kerberos pour chaque transaction séparée. Les performances peuvent être également être réduites par le canal de communications, qui nécessite l'obtention par WebSEAL de jetons via une requête SOAP adressée à Tivoli Federated Identity Manager.

Section [tfimsso:*jct_id*]

Section [tfimsso:*jct_id*]

La section `[tfimsso:<jct-id>]` permet de définir les options de configuration de la connexion unique Kerberos. Cette section contient les informations de configuration de connexion unique Tivoli Federated Identity Manager pour une seule jonction.

- Pour les jonctions standard, le nom de la section doit être qualifié au moyen du nom du point de jonction, y compris la barre oblique. Par exemple :

  ```
  [tfimsso:/kerbjct]
  ```

- Pour les jonctions d'hôte virtuel, le nom de la section doit être qualifié au moyen du nom du libellé de l'hôte virtuel, par exemple :

  ```
  [tfimsso:www.example.com]
  ```

always-send-tokens

Propriété booléenne. Cette propriété peut être utilisée pour optimiser les performances lorsque le serveur dorsal (comprenant des jonctions) est capable de maintenir l'état de session. Dans ce cas, vous pouvez indiquer si WebSEAL doit envoyer un jeton Kerberos pour chaque requête HTTP ou si WebSEAL doit attendre une réponse 401 avant de demander le jeton.

Une réponse 401 signifie qu'une autorisation est requise. Lorsque l'état de sessionb est maintenu, il n'est pas nécessaire de décerner une autorisation avant chaque requête. Afin de limiter le nombre d'extractions de jetons Kerberos au nombre de fois où une autorisation est requise, définissez la valeur

```
always-send-tokens = false
```

Lorsque le serveur dorsal ne peut maintenir l'état de session et qu'un jeton de sécurité doit être envoyé pour chaque requête HTTP, définissez :

```
always-send-tokens = true
```

applies-to

Cette propriété définit les critères de recherche à appliquer pour localiser le module de service de jeton de sécurité approprié dans Tivoli Federated Identity Manager.

La valeur est généralement un chemin d'accès spécifié selon le format suivant :

`http://hôte_serveur_webseal/nom_jonction`

Par exemple :

```
http://websealhost.example.com/kerbjct
```

service-name

Cette propriété essentielle remplit deux objectifs :

1. Spécifier le nom principal de service utilisé lors de la génération d'un jeton Kerberos.

   Cette valeur est employée par Tivoli Federated Identity Manager lors de la recherche de correspondances avec la chaîne d'accréditation. La configuration de la chaîne Tivoli Federated Identity Manager inclut une section Applies-to contenant une propriété de nom de service. La valeur du paramètre service-name de WebSEAL est comparée à la propriété du nom de service.

   Pour garantir une correspondance exacte, il convient que `service-name` renvoie à la propriété Nom de service dans la configuration de Tivoli Federated Identity Manager.

   Remarque : L'un des moyens permettant de trouver une correspondance consiste à utiliser, dans la configuration de Tivoli Federated Identity Manager, un caractère générique tel qu'un astérisque (*).

2. Spécifier le nom principal de service de l'utilisateur délégataire lors de la création du jeton Kerberos. Le nom principal de service (SPN) est défini sur le système Microsoft Windows.

   Pour déterminer le SPN, accédez au serveur Windows et exécutez la commande **setspn**. Par exemple :

   `setspn -L nom_utilisateur`

   Le serveur Web comportant des jonctions est exécuté avec l'identité *nom_utilisateur*. Par exemple, iisuser.

La syntaxe de cette propriété est la suivante :

`service-name=nom_principal_service`

Le format est le suivant :

`HTTP/nom_serveur_IIS.nom_domaine`

Par exemple :`service-name = HTTP/B16INTEL3.tamad.com`

renewal-window

Période, en secondes, pendant laquelle la durée de validité d'un jeton de sécurité est réduite. Cette entrée sert à compenser les différences ente les heures des systèmes et à permettre la mise en place de temps de transmission pour les jetons de sécurité.

```
renewal-window = 15
```

tfim-cluster-name

Nom du cluster WebSphere sur lequel le service Tivoli Federated Identity

Manager est déployé. Il convient que cette valeur corresponde à une autre entrée de section [tfim-cluster:<*cluster*>], où *cluster* désigne **tfim-cluster-name**.

Par exemple :

```
tfim-cluster-name = STSCluster2
```

token-collection-size

Pour optimiser les performances, WebSEAL peut demander plusieurs jetons Kerberos à Tivoli Federated Identity Manager au sein d'une même requête SOAP. Pour cela, la spécification de service Web WS-Trust est requise. Les jetons sont mis en cache dans la session de l'utilisateur, puis réutilisés lors des requêtes ultérieures.

WebSEAL ne demande des jetons de sécurité supplémentaires à Tivoli Federated Identity Manager qu'après l'utilisation complète ou l'expiration de tous les jetons placés en cache.
Vous pouvez spécifier le nombre de jetons à extraire de Tivoli Federated Identity Manager. Le nombre de requêtes envoyées à Tivoli Federated Identity Manager décroît à mesure que ce nombre augmente, mais la taille (et le temps de traitement) de chaque requête augmente également. Il peut arriver que les jetons Kerberos soient relativement volumineux.
Si vous spécifiez une valeur élevée pour cette propriété, pour pouvez augmenter de façon significative la taille de la session et l'utilisation de la mémoire pour WebSEAL.
La valeur par défaut est 10 :

```
token-collection-size = 10
```

token-type

Le seul type de jeton pris en charge est le type `kerberos`. Il s'agit de la valeur par défaut. Utilisez cette valeur. Ne la modifiez pas.

Section tfim-cluster

[tfim-cluster:*cluster***]**

Cette valeur définit le nom du cluster WebSphere pour le service Tivoli Federated Identity Manager. Le nom du *cluster* dans cette section doit correspondre à l'option **tfim-cluster-name** définie dans une section **[tfimsso:*jct-id*]**.

server Spécifie le niveau de priorité et l'adresse URL d'un serveur unique Tivoli Federated Identity Manager membre du cluster identifié pour cette section.

Il est possible de faire figurer plusieurs entrées **server** dans la section. Ceci vous permet de spécifier plusieurs entrées de serveur pour les besoins de la reprise en ligne et de l'équilibrage de charge entre WebSEAL et le serveur proxy de WebSphere Application Server.

Lorsque le cluster Tivoli Federated Identity Manager est configuré, WebSEAL vérifie l'état du serveur Web Proxy de Tivoli Federated Identity Manager toutes les minutes.

Lorsque vous disposez de plusieurs serveurs, vous pouvez utiliser le niveau de priorité de manière à spécifier l'ordre d'accès aux serveurs lors du traitement. Le niveau de priorité est un nombre entier compris dans la plage [0-9].

Lorsqu'un seul serveur existe, vous pouvez omettre le niveau de priorité. Lorsqu'aucun niveau de priorité n'est spécifié, il est supposé égal à 9 (valeur maximale).

Syntaxe :

```
server = [0-9],URL_serveur
```

Exemple :

```
9,http://mydataserver.example.com/TrustServerWST13/services
/RequestSecurityToken
```

handle-pool-size
Définit le nombre maximal de descripteurs en cache utilisés lors de la communication avec Tivoli Federated Identity Manager.

Valeur par défaut : 10

handle-idle-timeout
Durée, en secondes, avant le retrait d'un descripteur inutilisé de la mémoire cache du pool de descripteurs.

Valeur par défaut : 240 secondes

timeout
Durée d'attente, en secondes, d'une réponse de la part de Tivoli Federated Identity Manager.

Valeur par défaut : 240 secondes

ssl-keyfile
Nom du fichier de la base de données de clés contenant le certificat client à utiliser.

Les entrées SSL, ainsi que les suivantes, sont facultatives. Elles ne sont requises que dans les cas suivants :

- Au moins une entrée de serveur indique que le protocole SSL (HTTPS) doit être utilisé.
- Un certificat est requis autre que celui utilisé par ce serveur lors des communications avec le serveur de règles.

Remarque : Cette valeur, ainsi que les valeurs d'entrée SSL suivantes, doivent être partagées par toutes les variables du serveur qui exploitent le protocole HTTPS. Lors du déploiement sur un clusterWebSphere, les valeurs doivent être les mêmes pour chaque serveur du cluster qui exploite le protocole HTTPS.

ssl-keyfile-stash
Nom du fichier de mot de passe secret pour le fichier de la base de données de clés.

ssl-keyfile-label
Libellé du certificat client dans la base de données de clés.

ssl-valid-server-dn
Cette entrée de configuration spécifie le nom distinctif du serveur (obtenu à partir du certificat SSL) qui sera accepté. Lorsqu'aucune entrée n'est configurée, tous les noms distinctifs sont considérés comme valides. Il est possible de définir plusieurs noms distinctifs en incluant des entrées de configuration multiples correspondant à ce nom.

ssl-fips-enabled
Cette entrée contrôle l'activation et la désactivation des communications FIPS avec Tivoli Federated Identity Manager. Lorsqu'aucune entrée de configuration n'est présente, le paramètre global FIPS, tel que déterminé par le serveur de règles TAM, entre en vigueur.

Remarque : Pour obtenir une description complète de chaque propriété de la section, reportez-vous au document *IBM Tivoli Access Manager WebSEAL Administration Guide*. Consultez également les commentaires contenus dans le fichier de configuration WebSEAL.

Formulaire de configuration de jonction Kerberos

Ce formulaire permet d'assembler les valeurs à ajouter au fichier de configuration WebSEAL.

Tableau 139. Propriétés des sections tfimsso et tfim-cluster

Propriété	**Votre valeur**
[tfimsso:*ID_jonction*]	
always-send-tokens	Valeur par défaut : false
applies-to	
service-name	
renewal-window	Valeur par défaut : 15
tfim-cluster-name	
token-collection-size	Valeur par défaut : 10
token-type	kerberos
[tfim-cluster:*cluster*]	
serveur	
handle-pool-size	Valeur par défaut : 10
handle-idle-timeout	Valeur par défaut : 240
timeout	Valeur par défaut : 240
ssl-keyfile	
ssl-keyfile-stash	
ssl-keyfile-label	
ssl-valid-server-dn	
ssl-fips-enabled	

Conseils pour la configuration :

- Assurez-vous que la propriété **service-name** correspond à la configuration de la chaîne d'accréditation Tivoli Federated Identity Manager.
- Assurez-vous que la propriété **tfim-cluster-name** correspond à la propriété *cluster* de la section [tfim-cluster:*cluster*].
- Assurez-vous que la propriété *cluster* de la section [tfim-cluster:*cluster*] correspond au nom du cluster WebSphere.

Débogage d'une jonction WebSEAL Kerberos

Configurez la jonction WebSEAL Kerberos en modifiant le fichier de configuration WebSEAL et en utilisant la commande pdadmin pour créer la jonction, puis joignez les listes de contrôle d'accès.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Pourquoi et quand exécuter cette tâche

La configuration d'une jonction WebSEAL s'effectue en deux étapes :

- Editez le fichier de configuration WebSEAL.

  Vous devez spécifier dans le fichier de configuration WebSEAL les propriétés qui permettent de prendre en charge les jonctions spécifiques nécessaires à la connexion unique Kerberos avant de pouvoir créer la jonction via la commande 'pdadmin'.
- Utilisez la commande pdadmin pour créer la jonction et joindre les listes de contrôle d'accès (ACL) requises.

  Pour créer une jonction standard activée pour la connexion unique Kerberos, exécutez la commande de création de jonction ('server task create') en spécifiant l'option -Y. L'option -Y spécifie que la connexion unique SPNEGO/Kerberos est requise pour la jonction.

  Pour créer une jonction d'hôte virtuel activée pour la connexion unique Kerberos, exécutez la commande de création d'hôte virtuel ('server task create') en spécifiant l'option -Y.

  WebSEAL prend en charge de nombreuses options pour la création de jonctions. vous pouvez associer l'option -Y à d'autres options, suivant les besoins de votre déploiement. Pour obtenir des informations complètes sur les options liées aux jonctions WebSEAL, reportez-vous au document *IBM Tivoli Access Manager WebSEAL Administration Guide*.

Procédure

1. Utilisez un éditeur de texte pour éditer le fichier de configuration WebSEAL.

   Spécifiez les valeurs regroupées dans le formulaire pour le support de jonction Kerberos.

   Pour plus d'informations, voir : «Planification de la configuration des jonctions WebSEAL Kerberos», à la page 582
2. Utilisez la commande pdadmin pour créer la jonction Kerberos et joindre les listes de contrôle d'accès requises.

   Vous pouvez créer soir des jonctions Kerberos classiques, soit des jonctions Kerberos d'hôte virtuel.

   Remarque :

   - Le nom de la jonction doit correspondre à la valeur *jct_id* définie dans la section [tfimsso:*jct_id*] du fichier de configuration WebSEAL.
   - Assurez-vous que vous avez bien configuré le fichier de configuration WebSEAL conformément au type de jonction que vous prévoyez d'utiliser. Si vous n'avez pas édité le fichier de configuration WebSEAL, la commande d'administration échoue et renvoie un message d'erreur.

   Jonctions Kerberos classiques

   a. Créez la jonction :

```
pdadmin sec_master>  server task default-webseald-websealhost
  create -t tcp -h mydataserver.example.com -Y /kerbjct
```

L'hôte mydataserver.example.com correspond au serveur d'arrière plan IIS.

b. Rattachez la liste de contrôle d'accès :

```
pdadmin sec_master> acl attach /WebSEAL/websealhost-default/kerbjct testacl
```

Jonctions Kerberos d'hôte virtuel

a. Créez la jonction :

```
pdadmin sec_master> server task default-webseald-websealhost virtualhost
  create -t tcp -h mydataserver.example.com -v website.example.com
  -Y kerbvirtjct
```

b. Rattachez la liste de contrôle d'accès :

```
pdadmin sec_master> acl attach /WebSEAL/websealhost-default/kerbvirtjct
  testacl
```

Résultats

Les messages d'erreur sont consignés dans le fichier journal de configuration WebSEAL. Par exemple, sous UNIX ou Linux :

```
/opt/pdweb/log/msg__webseald-default.log
```

Remarques sur la configuration de WebSEAL

Utilisez les remarques suivantes sur la configuration de WebSEAL pour la communication entre WebSEAL et le client.

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Remarques concernant la configuration entre WebSEAL et le client

- La haute disponibilité pour le serveur WebSEAL s'obtient généralement en plaçant un équilibreur de charge en amont du serveur WebSEAL. Consultez l'article du document IBM Developer Works intitulé *Load Balancers for Tivoli Access Manager* :

  http://www-128.ibm.com/developerworks/tivoli/library/t-tlb/index.html
- La sécurité des canaux de communication entre le client et WebSEAL est généralement assurée grâce à l'acquisition d'un certificat SSL destiné au serveur WebSEAL.
- Les clients peuvent s'authentifier à WebSEAL via n'importe quelle méthode prise en charge.
- Aucune modification de ces configurations standard n'est nécessaire pour prendre en charge les jonctions Kerberos.

Remarques concernant la configuration des communications entre WebSEAL et la jonction

- La haute disponibilité pour le serveur de la jonction s'obtient généralement en configurant des serveurs de jonctions multiples pour le point de jonction. Consultez l'article du document IBM Developer Works intitulé *Load Balancers for Tivoli Access Manager* :

http://www-128.ibm.com/developerworks/tivoli/library/t-tlb/index.html

- La sécurité des canaux de communication entre WebSEAL et la jonction est généralement assurée via l'authentification mutuelle de certificats SSL.
- Aucune modification de ces configurations standard n'est nécessaire pour prendre en charge les jonctions Kerberos.

Synchronisation temporelle entre WebSphere et WebSEAL

Vérifiez que les paramètres temporels sont synchronisés entre le système hébergeant l'instance WebSphere Application Server qui exécute Tivoli Federated Identity Manager et le système hébergeant l'instance Tivoli Access Manager WebSEAL.

Pour afficher les paramètres, procédez comme suit :

1. Sur le système WebSphere, sélectionnez **Default Domain Security Settings** > **Account Policies** > **Kerberos Policy**.
2. Examinez la tolérance maximale définie pour la synchronisation de l'horloge système.

Si l'écart horaire entre l'instance WebSphere Application Server et le serveur WebSEAL est importante, il se peut que les jetons de sécurité générés parTivoli Federated Identity Manager expirent avant de pouvoir être utilisés.

Messages d'erreur de configuration

Les messages d'erreur suivants s'affichent lorsqu'une ou plusieurs des conditions suivantes sont vérifiées :

- La propriété service-name ne correspond pas à la configuration de la chaîne d'accréditation de Tivoli Federated Identity Manager.
- WebSEAL extrait des jetons de Tivoli Federated Identity Manager, mais ceux-ci ont expiré. Cette situation peut par exemple se produire lorsque les paramètres temporels des serveurs ne sont pas synchronisés.
- Le navigateur renvoie une erreur. Par exemple :

```
Erreur du serveur
Access Manager WebSEAL could not complete your request due to an
unexpected error.
Diagnostic Information
Method: GET
URL: /kjct/index.html
Error Code: 0x38cf027c
Error Text: DPWWA0636E No TFIM single sign-on tokens were available.
```

- Le fichier journal WebSEAL contient des erreurs. Par exemple (certaines lignes sont scindées pour d es raisons de mise en forme) :

```
DPWWA2852E   An error occurred when attempting to communicate with the SOAP
  server URL
http://d06win13.testlab.example.com/TrustServerWST13/services/
  RequestSecurityToken: +JNI:
Error running InitializeSecurityContext for HTTP/d02jlnx.testlab.example.com:
-2146893042 (No credentials are available in the security package).
File h:\fim620\src\kerberoswin32\KerbUserState.cpp,
 line 641  (error code: 71/0x47).
2008-03-04-13:08:10.080-06:00I----- 0x38CF027C
webseald ERROR wwa sso ThirdPartyJunction.cpp 4124 0x00000070
DPWWA0636E   No TFIM single sign-on tokens were available.
```

Débogage d'une jonction Kerberos

Pour effectuer le débogage d'un déploiement de jonction Kerberos, activez la fonction de trace sur Tivoli Federated Identity Manager et Tivoli Access Manager. Un point de trace approprié pour Tivoli Access Manager et WebSEAL est : pdweb.sso.tfim.

Par exemple, dans un environnement Linux ou UNIX :

```
pdadmin> server task default-webseald-c1sun1 trace set pdweb.sso.tfim 9
  file path=/var/pdweb/log/debug.log
```

Pour désactiver la fonction de trace, définissez la valeur du niveau de trace sur 0.

Configuration de WebSEAL pour gérer les cookies

Par défaut, WebSEAL ne supprime aucun cookie à la déconnexion. Si vous comptez configurer WebSEAL pour gérer les cookies, la liste des cookies gérés ne doit pas inclure le cookie de session WebSphere.

Chapitre 40. Tâche de configuration SSL pour un déploiement de jonctions Kerberos

Pour une sécurité optimale, configurez les communications SSL entre serveurs dans un déploiement de jonctions Kerberos.

Cette rubrique présente les étapes de configuration d'un environnement en cluster WebSphere en vue d'utiliser des communications SSL entre WebSEAL, IBM HTTP Server (IHS), un plug-in WebSphere Application Server, WebSphere Application Server et Tivoli Federated Identity Manager. Ces procédures ne concernent pas les communications SSL entre le client et WebSEAL, ni avec le serveur Web d'arrière-plan. Aucune modification de ces configurations SSL standard n'est nécessaire pour prendre en charge les jonctions Kerberos.

Conseil : Envisagez le déploiement d'une configuration opérationnelle sans SSL, avant d'ajouter le protocole SSL.

Pour chaque composant, créez une paire de clés publique/privée et extrayez la clé publique vers un emplacement connu.

Sur le serveur WebSEAL :

1. Copiez la clé publique IHS sur le système WebSEAL.
2. Exécutez l'utilitaire **ikeyman** pour ajouter la clé publique IHS. Lorsque l'environnement comprend plusieurs instances Proxy IHS, accomplissez cette procédure pour chaque serveur IHS.
3. Configurez les valeurs appropriées pour les variables `[tfim-cluster:cluster]` suivantes : server, ssl-keyfile, ssl-keyfile-stash. Eventuellement, configurez la variable ssl-valid-server-dn le cas échéant.

   Pour plus d'informations, voir «Planification de la configuration des jonctions WebSEAL Kerberos», à la page 582.
4. Redémarrez WebSEAL pour activer les modifications apportées au fichier de configuration WebSEAL.

Sur le serveur IBM HTTP Server :

1. Copiez la clé publique WebSEAL sur le système IHS.
2. Exécutez l'utilitaire **ikeyman** sur IHS pour ajouter la clé publique WebSEAL.
3. Copiez la clé publique WebSphere du système WebSphere Deployment Manager (dmgr) vers le système IHS.
4. Exécutez l'utilitaire **ikeyman** sur IHS pour ajouter la clé publique WebSphere.
5. Mettez à jour le fichier `httpd.conf` pour configurer ou ajouter un hôte virtuel prenant en charge les connexions SSL.
6. Redémarrez l'instance IHS pour activer les modifications.
7. Lorsque votre déploiement comprend des instances de proxy IHS multiples, répétez la procédure ci-dessus pour chaque proxy IHS.

Dans le module d'extension WebSphere situé sur le serveur IHS :

1. Copiez la clé publique WebSphere sur le système hébergeant le module d'extension.

2. Exécutez l'utilitaire **ikeyman** pour le module d'extension afin d'ajouter la clé publique WebSphere.
3. Copiez la clé publique WebSphere du noeud WebSphere vers le serveur hébergeant le module d'extension.
4. Exécutez l'utilitaire **ikeyman** pour le module d'extension afin d'ajouter la clé publique du noeud WebSphere.
5. Lorsque votre déploiement comprend de multiples modules d'extension, répétez la procédure ci-dessus pour chaque module d'extension.

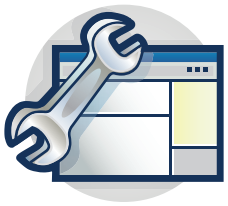
Sur l'instance WebSphere Network Deployment Manager (dmgr) :

1. Assurez-vous que la clé publique du module d'extension est référencée sous un chemin accessible via la console d'administration WebSphere.
2. Utilisez la console WebSphere pour ajouter la clé publique du module d'extension à CellDefaultTrustStore.
3. Lorsque votre déploiement comprend de multiples modules d'extension, répétez la procédure ci-dessus pour chaque module d'extension.
4. Assurez-vous que la clé publique du noeud est référencée sous un chemin accessible via la console d'administration WebSphere.
5. Utilisez la console WebSphere pour ajouter la clé publique du noeud à CellDefaultTrustStore.
6. Lorsque votre déploiement comprend de multiples noeuds, répétez la procédure ci-dessus pour chaque noeud.
7. Si nécessaire, configurez l'authentification client pour votre déploiement.

Sur le noeud WebSphere :

1. Assurez-vous que la clé publique du gestionnaire de déploiement (DMGR) est référencée sous un chemin accessible via la console d'administration WebSphere.
2. Utilisez la console WebSphere pour ajouter la clé publique du gestionnaire de déploiement à NodeDefaultTrustStore.
3. Lorsque votre déploiement comprend de multiples noeuds, répétez la procédure ci-dessus pour chaque noeud.

Partie 6. Configuration de User Self Care



Les rubriques de la section Configuration vous guident pas à pas lors de la configuration de User Self Care.

Cette section présente le déploiement de User Self Care. Veuillez d'abord consulter la présentation de la fonction User Self Care :

Chapitre 41, «Découverte de User Self Care», à la page 597

Chapitre 41. Découverte de User Self Care

User Self Care fournit une méthode grâce à laquelle les utilisateurs peuvent être mis à disposition dans des environnements entreprise à client.

User Self Care effectue cette mise à disposition en fournissant un ensemble d'opérations que les utilisateurs peuvent utiliser pour créer et gérer leurs propres comptes. Ces opérations incluent :

- Création d'un compte
- Création et mise à jour des attributs associés au compte
- Modification des mots de passe
- Récupération des mots de passe et ID utilisateur oubliés
- Suppression de comptes

User Self Care est basé sur la technologie STS Tivoli Federated Identity Manager.

Remarque : IBM a déprécié le client Tivoli Federated Identity Manager Security Token Service (STS) dans cette version.

Si vous utilisez WebSphere 6.X, vous pouvez continuer de vous servir du client Tivoli Federated Identity Manager Security STS tant que Tivoli Federated Identity Manager prend en charge WebSphere 6.X. Lorsque Tivoli Federated Identity Manager arrêtera son support pour WebSphere 6.X, vous devrez utiliser WebSphere Application Server version 7 Update 11 et version ultérieure. Voir API client WS-Trust et WS-Trust Clients pour plus d'informations.

Grâce à l'infrastructure STS, les administrateurs peuvent connecter leurs propres modules de création et de consommation de jetons. User Self Care utilise l'infrastructure STS et les composants HTTP de Tivoli Federated Identity Manager, mais il n'est pas utilisé pour la consommation et création de jeton.

Les utilisateurs accèdent aux opérations de User Self Care via une interface HTTP. Les utilisateur interagissent avec les pages Web qui demandent des entrées, collectent des données et fournissent du feedback. User Self Care fournit un petit ensemble d'URL servant de noeuds finaux pour accéder aux opérations.

Vous pouvez personnaliser User Self Care. Les plug-ins de modules STS démarrés de manière séquentielle dans une chaîne implémentent la logique commerciale. Pour d'autres fonctions pour chaque chaîne, vous pouvez remplacer des modules individuels ou en ajouter de nouveaux. Vous pouvez modifier ou remplacer des formulaires HTML selon nécessaire.

User Self Care utilise les fonctions de cluster, de distribution, de mise à l'échelle et de configuration fournies par WebSphere. User Self Care utilise également le composant WebSphere Federated Repositories pour rendre les adaptateurs de registre disponibles à l'environnement d'exploitation. Les administrateurs peuvent ajouter ou remplacer des registres.

User Self Care s'intègre à Tivoli Access Manager WebSEAL. WebSEAL fournit une authentification et une autorisation pour les transactions entreprise à client.

L'illustration montre les parties logicielles composant la solution User Self Care.

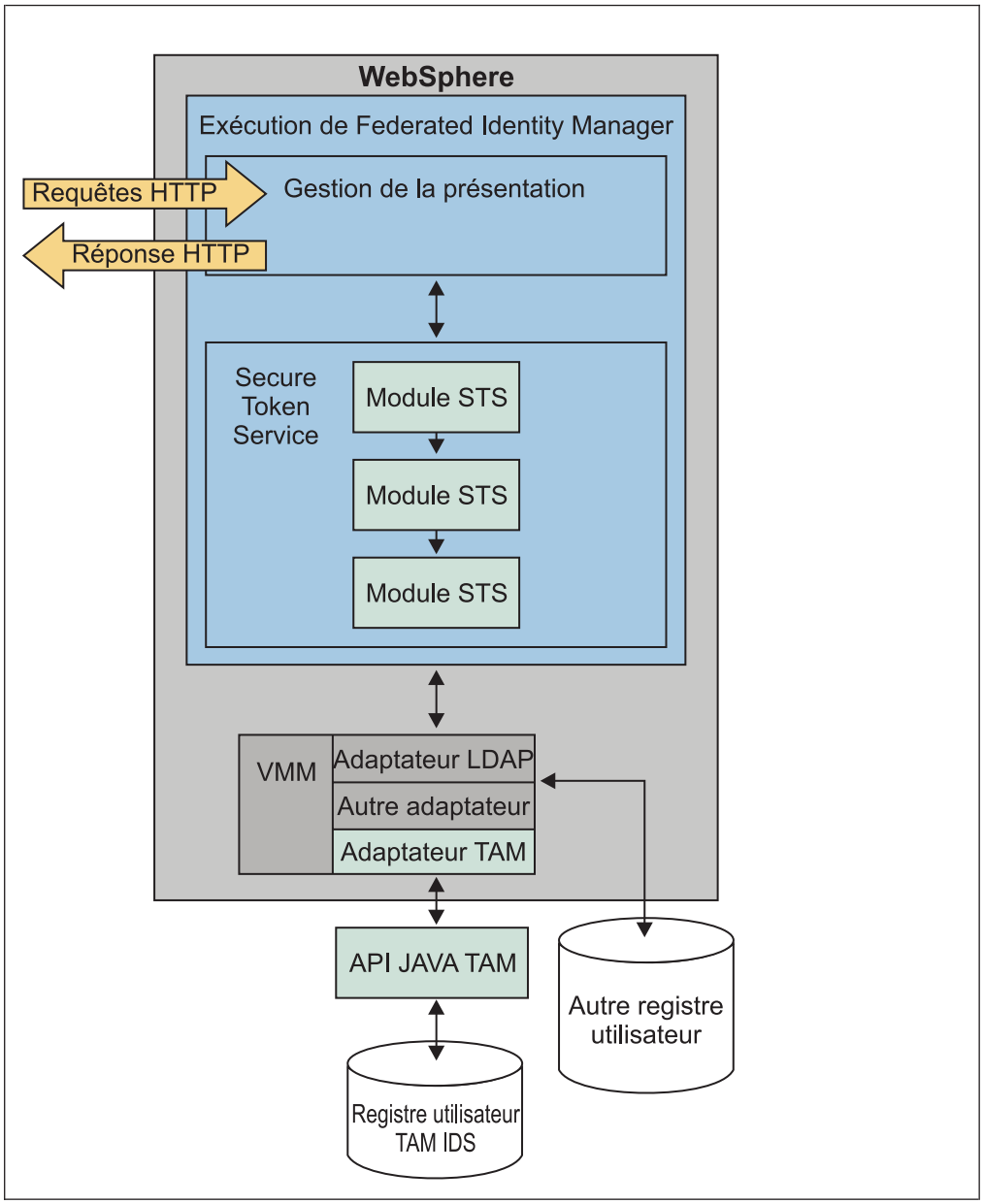


Figure 66. Solution User Self Care

- WebSphere fournit l'infrastructure pour la plupart des parties logicielles.
- L'exécution de Tivoli Federated Identity Manager fournit deux composants qui prennent en charge User Self Care :

  Gestion de la présentation User Self Care
  Fournit un ensemble de pages par défaut. Les utilisateurs interagissent avec ces pages en demandant des URL User Self Care. L'infrastructure de gestion prend en charge la personnalisation et le remplacement de ces pages. Cette prise en charge inclut la capacité à substituer (personnaliser) les macros des pages.

  Chaînes d'accréditation STS (Secure Token Service)
  Prend en charge la création des chaînes dynamiques des modules de plug-in pour effectuer la logique commerciale. La prise en charge User Self Care inclut un nombre de chaînes STS. Chaque chaîne se mappe sur

une opération User Self Care. Vous pouvez étendre ces chaînes. Vous pouvez remplacer ou modifier les modules de composant de chaque chaîne. La validation de l'entrée utilisateur et l'envoi d'u courrier électronique de confirmation constitue un exemple d'opération de chaîne User Self Care.

- Les modules STS utilise le réferentiel WebSphere Federated Repository pour communiquer avec le registre d'utilisateurs. Lorsque le registre d'utilisateurs cible est Tivoli Access Manager, User Self Care utilise l'adaptateur de produit pour communiquer avec le registre Tivoli Access Manager via l'API Tivoli Access Manager Registry Direct Java.

User Self Care fonctionne avec plusieurs registres utilisateur. Chaque registre a une syntaxe unique pour la réalisation des opérations de gestion. Le composant WebSphere Federated Repositories permet à User Self Care d'émettre une commande de gestion, telle que **user create**, avec une syntaxe cohérente. Le composant Federated Repositories transmet alors la requête à l'adaptateur de registre approprié, qui traduit la commande en syntaxe spécifique au registre.

Etant donné que WebSphere Federated Repositories fournit une interface de plug-in pour les adaptateurs, vous pouvez ajouter de nouveaux registres sans modifier User Self Care.

Personnalisation efficace de User Self Care

Les déploiements de User Self Care sont en général personnalisés selon les besoins spécifiques de la société. Vous pouvez personnaliser votre déploiement de la manière la plus efficace possible seulement si vous comprenez bien comment les différentes parties de User Self Care fonctionnent ensemble.

1. Découvrez la technologie User Self Care.
   - User Self Care est basé sur une série d'opérations. Voir «Découverte des opérations User Self Care», à la page 600.
   - Les utilisateurs interagissent avec les fonctions de User Self Care via des échanges de demandes et réponses HTTP. Les pages HTML en URL dirigent les échanges. Les pages HTML sont des modèles pour les informations que vous souhaitez échanger avec vos utilisateurs. Vous puvez (et devez) personnaliser les pages HTML de sorte à refléter les besoins de votre société. Pour plus d'informations sur les pages HTML par défaut, voir «URL User Self Care», à la page 608.
   - Un grand nombre de sites Internet utilise des tests de réponse Captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) pour se protéger des attaques informatiques. Cette technologie fait partie des nombreux déploiements User Self Care. Le produit User Self Care fournit un module Captcha de démonstration. Voir «Démonstration Captcha» , à la page 611.
2. Déployez Tivoli Federated Identity Manager et configurez User Self Care.

   Ce document présente les étapes de configuration que vous devez faire dans un ordre spécifique. Voir Chapitre 42, «Déploiement de User Self Care», à la page 613.
3. Découvrez les méthodes de réglage des caches distribués pour optimiser les performances. Voir Chapitre 43, «Réglage de User Self Care», à la page 679.

Découverte des opérations User Self Care

Une *operation* User Self Care est la série d'étapes requises pour accomplir une tâche.

La récupération d'un mot de passe oublié par un utilisateur est un exemple de tâche. Pour exécuter cette tâche, l'utiliateur doit procéder aux étapes suivantes :

1. Envoyer un formulaire Web avec son ID utilisateur.
2. Envoyer un second formulaire qui l'invite à répondre à sa question secrète et à entrer un nouveau mot de passe.
3. Cliquez sur un lien dans un courrier électronique qui lui est envoyé.

Cette série d'étapes constitue une *opération*.

Chaque action lancée par un utilisateur effectuée dans le cadre de User Self Care se trouve sous la forme d'une requête HTTP. Une requête peut être par exemple une demande de page, l'envoi d'un formumaire ou l'activation d'un lien dans un courrier électronique. Chaque requête HTTP a une réponse HTTP correspondante. Les réponses peuvent être par exemple l'envoi à l'utilisateur du formulaire d'entrée ou d'une notification l'informant de l'envoi d'un courrier électronique. Chaque opération User Self Care est composée d'un ou plusieurs échanges requête-réponse.

La plupart du temps, chaque échange requête-réponse est un événement atomique. Par exemple, lorsqu'un utilisateur demande la page Gestion des profils, User Self Care termine une opération discrète en renvoyant la page. User Self Care ne conserve pas d'état ni aucun signe indiquant que l'utilisateur a lancé la requête initiale. Il existe des exceptions à ce comportement d'état utilisateur ; elles sont décrites dans les rubriques liées aux opérations individuelles de cette documentation.

Cette documentation regroupe les échanges requête-réponse selon leur association à une opération. Chaque opération est associée à une chaîne d'accréditation de service de jeton sécurisé particulier. Les chaînes d'accréditation STS réalisent la base du travail en traitant une opération User Self Care.

Remarque : IBM a déprécié le client Tivoli Federated Identity Manager Security Token Service (STS) dans cette version.

Si vous utilisez WebSphere 6.X, vous pouvez continuer de vous servir du client Tivoli Federated Identity Manager Security STS tant que Tivoli Federated Identity Manager prend en charge WebSphere 6.X. Lorsque Tivoli Federated Identity Manager arrêtera son support pour WebSphere 6.X, vous devrez utiliser WebSphere Application Server version 7 Update 11 et version ultérieure. Voir API client WS-Trust et WS-Trust Clients pour plus d'informations.

Echange requête-réponse typique

Le flux typique lorsqu'un utilisateur soumet une requête dans User Self Care est le suivant :

1. L'utilisateur demande une URL User Self Care indiquant un formulaire HTML.
2. Le composant de gestion de présentation User Self Care renvoie le formulaire HTML approprié.

   Si le module Captcha est utilisé, la chaîne STS Captcha est démarrée afin d'obtenir l'image affichée à l'utilisateur à des fins de validation.

3. L'utilisateur fournit des données pour le formulaire, puis l'envoie.
4. Le composant de gestion de présentation envoie la requête HTTP qui en résulte à la chaîne d'accréditation STS.
5. Les modules d'accréditation STS User Self Care de la chaîne sont démarrés dans un ordre particulier, pour réaliser des tâches telles que :
   - Validation des données
   - Mappage des attributs
   - Interaction avec les registres
   - Envoi de courrier électronique
6. Les modules STS renvoient les résultats de traitement au composant de gestion de présentation.
7. Le composant de gestion de présentation renvoie une réponse HTTP à l'utilisateur. Voir la liste suivante de réponses type :
   - Un autre formulaire
   - Le même formulaire avec un message
   - Une page d'erreur
   - Une page d'informations

   Selon l'opération, la tâche de l'utilisateur est finie ou bien elle nécessite un autre étape. Si une autre étape est nécessaire, la séquence précédente ou identique est répétée.

Opérations et chaînes STS

Chaque opération de User Self Care se mappe à une chaîne STS unique. Lors de l'opération, la chaîne STS peut être démarrée plusieurs fois. User Self Care détermine la phase de l'opération en cours et contrôle le comportement en conséquence.

Par exemple, la validation Captcha peut être réalisée lorsqu'un utilisateur soumet le formulaire d'inscription initial. Toutefois, elle n'est pas réalisée lorsque l'utilisateur clique sur le lien de l'e-mail. Dans les deux cas, la même chaîne STS est démarrée et le module STS Captcha est présent au début de la chaîne. Dans le second cas, le module Captcha n'est pas censé effectuer d'action et transmet la requête au module STS suivant dans la chaîne.

Vous pouvez utiliser la console d'administration pour afficher chaque chaîne d'accréditation. Les chaînes d'accréditation correspondent à une ou plusieurs opérations User Self Care. Lorsque vous affichez les chaînes d'accréditation, vous voyez les modules STS accomplissant l'opération. Vous pouvez alors personnaliser les modules et chaînes pour votre déploiement.

Remarque : Pour des informations sur la personnalisation de User Self Care, voir le Wiki Tivoli Federated Identity Manager :

http://www.ibm.com/developerworks/wikis/display/tivolifederatedidentitymanager/Home

Opération de vérification d'existence d'ID utilisateur

A la plage d'inscription initiale, l'utilisateur entre un ID utilisateur dans une zone spécifique. User Self Care fournit une icône sur laquelle l'utilisateur peut cliquer si l'ID existe dans le registre.

L'opération d'existence d'ID utilisateur est une exception à la règle d'unicité de chaîne STS par opération. Cette opération se mappe sur la même chaîne d'accréditation STS en opération d'inscription. Toutefois, elle est conceptuellement différente et elle utilise une URL différente.

Flot de tâches d'opération :

1. L'utilisateur entre son ID utilisateur demandé dans une zone de formulaire.
2. L'utilisateur clique sur l'icône.
3. La chaîne STS de création de compte démarre.
   - Le registre est interrogé pour déterminer si l'ID utilisateur existe.
   - Le cache interne est également interrogé.

     La vérification du cache interne est décrite à la rubrique «Opération d'inscription».

Opération d'inscription

L'opération d'inscription est effectuée en deux échanges question-réponse : lorsque vous obtenez des informations utilisateur pour l'envoi d'un courrier électronique de validation et lorsque l'utilisateur valide l'opération en cliquant sur un lien dans le courrier électronique.

Demande d'inscription initiale

Flot de tâches d'opération :

1. L'utilisateur demande et reçoit un formulaire de demande d'inscription. L'utilisateur renseigne les zones du formulaire avec des détails d'inscription comme :
   - ID utilisateur
   - Adresse électronique
   - Mot de passe
   - Choix d'attribut de profil, y compris l'attribut de question secrète.
2. L'utilisateur soumet le formulaire de demande d'inscription.
3. La chaîne STS de création de compte démarre.
   - Si des erreurs surviennent, elles sont renvoyées à l'utilisateur. Les erreurs s'affichent en tant que message du formulaire que l'utilisateur a traité.
   - Si aucune erreur ne se produit, un courrier électronique est envoyé à l'utilisateur pour validation. User Self Care affiche une page informant l'utilisateur de la présence du message électronique.
4. Une entrée est créée dans un cache interne qui conserve les informations d'inscription de l'utilisateur lors de la validation. Ce cache interne conserve également l'ID utilisateur de sorte qu'aucun autre utilisateur ne puisse l'utiliser pour l'inscription. Vous pouvez configurer les limites de temps concernant la période pendant laquelle les données sont conservées dans le cache interne.

Validation de l'inscription

Le courrier électronique qui a été envoyé lors de la demande d'inscription initiale contient un lien auquel est attachée une chaîne de requête. La chaîne de requête contient une clé dans l'entrée de cache interne pour que les données soumises initialement par l'utilisateur puissent être récupérées et que l'inscription se finalise.

Flot de tâches :

1. L'utilisateur clique sur un lien dans le courrier électronique de validation.
2. La chaîne STS de création de compte démarre.
3. Si des erreurs se produisent, elles s'affichent dans une page envoyée à l'utilisateur. Si aucune erreur ne se produit, User Self Care :
   a. Crée une entrée dans le registre pour le nouveau compte d'utilisateur.
   b. Supprime l'entrée du cache interne.
   c. Envoie un message indiquant une réussite à l'utilisateur.

Opérations de gestion du mot de passe

Il existe deux opérations de gestion de mot de passe : un changement de mot de passe initié par l'utilisateur et un changement du mot de passe requis après l'expiration d'un mot de passe existant.

Modification du mot de passe lancée par l'utilisateur

Flot de tâches :

1. L'utilisateur demande l'URL de formulaire Modifier le mot de passe.
2. User Self Care fournit à l'utilisateur un formulaire dans lequel il entre son ancien mot de passe, puis il entre deux fois son nouveau mot de passe.
3. L'utilisateur soumet le formulaire dans l'URL Modifier le mot de passe.
4. User Self Care démarre la chaîne STS Modifier le mot de passe.
   - En cas d'erreur, User Self Care envoie à l'utilisateur une page d'informations contenant lesdites erreurs.
   - Si aucune erreur ne se produit, le mot de passe est modifié. User Self Care envoie alors à l'utilisateur une page indiquant la réussite de l'opération.

Modification du mot de passe après son expiration

Le flux de tâches est identique que celui de la rubrique de modification de mot de passe lancée par l'utilisateur, sauf :

- L'utilisateur fait une requête initiale pour une ressource protégée
- Le serveur point de contact demande à l'utilisateur de modifier son mot de passe.

La requête initiale de l'utilisateur est interceptée par le serveur point de contact d'authentification, tel que WebSEAL ou WebSphere Application Server. Le serveur point de contact gère le flux de communications et doit diriger l'utilisateur vers User Self Care de sorte qu'il modifie son mot de passe.

User Self Care indique des suggestions et améliorations de déploiement pour ce faire à l'aide de WebSEAL en serveur point de contact. Pour plus d'informations, voir «Intégration de User Self Care à WebSEAL», à la page 673.

User Self Care peut fonctionner en tant que composant à appeler pour une fonction telle que Tivoli Access Manager Local Response Redirect. Cette fonction redirige l'utilisateur vers le gestionnaire de User Self Care pour réaliser une opération de modification de mot de passe. L'utilisateur est alors redirigé à la fin de l'opération.

Opérations de gestion de profil

Vous pouvez utiliser la gestion des profils pour gérer les informations étendues spécifiques à votre compte.

Voici des exemples de ces informations :

- Adresse
- Numéro de téléphone
- Question secrète

Requête de gestion des profils initiale

Flot de tâches :

1. L'utilisateur soumet des requêtes pour l'URL de formulaire de gestion des profils. Cette adresse URL doit être une ressource protégée.
2. L'identité de l'utilisateur est obtenue à partir du contexte authentifié.
3. User Self Care démarre la chaîne STS de gestion des profils et fournit l'identité de l'utilisateur.
   - Si des erreurs se produisent, elles s'affichent dans une page d'informations envoyée à l'utilisateur.
   - Si aucune erreur ne se produit, le module STS extrait les attributs à partir du registre.
4. User Self Care présente à l'utilisateur le formulaire de gestion des profils contenant ses attributs existants. L'utilisateur peut alors mettre à jour les informations de profil, y compris sa question secrète.

Soumission de la mise à jour de profil

Flot de tâches :

1. L'utilisateur modifie les zones de son choix et soumet la forme.
2. L'identité de l'utilisateur est obtenue à partir du contexte authentifié.
3. User Self Care démarre la chaîne STS de gestion des profils.
   - Si des erreurs se produisent, elles s'affichent dans une page d'informations envoyée à l'utilisateur.
   - Si aucune erreur ne se produit, le registre est mis à jour. User Self Care envoie une page indiquant une réussite à l'utilisateur.

Opération d'ID utilisateur oublié

Il est toujours possible de récupérer un ID oublié à l'aide de cette procédure.

Flot de tâches d'opération :

1. L'utilisateur clique sur l'URL ID oublié. Cette URL ne doit pas être une ressource protégée.
2. Le formulaire d'ID oublié est renvoyé à l'utilisateur.
3. L'utilisateur entre son adresse e-mail.

   Une solution personnalisée peut utiliser un attribut de registre différent, tel qu'un numéro de compte client, par exemple. Le formulaire User Self Care par défaut utilise l'adresse e-mail.
4. L'utilisateur soumet le formulaire.
5. User Self Care transmet le contenu du formulaire à la chaîne STS ID oublié. Les modules de cette chaîne extraient du registre tous les ID utilisateur associés à l'adresse e-mail, et les envoie par messagerie électronique à l'utilisateur.
   - Si des erreurs se produisent, elles s'affichent dans une page d'informations envoyée à l'utilisateur.

- Si aucune erreur ne se produit, User Self Care envoie la page d'informations d'accusé de l'ID oublié à l'utilisateur. La page informe l'utilisateur que les ID utilisateur ont été envoyés à son adresse e-mail.

Opération de mot de passe oublié

L'opération liée au mot de passe oublié a lieu dans le cadre de plusieurs échanges questions-réponses.

Flot de tâches :

1. L'utilisateur demande l'URL Mot de passe oublié. Cette URL ne doit pas être une ressource protégée.
2. User Self Care envoie le formulaire de mot de passe oublié à l'utilisateur.
3. L'utilisateur entre son ID utilisateur et envoie le formulaire.
4. User Self Care transmet le contenu du formulaire à la chaîne STS ID oubliée de sorte à extraire la question secrète.
5. Le module STS envoie le formulaire de question secrète du mot de passe oublié à l'utilisateur. Le formulaire contient une question secrète et une zone dans laquelle l'utilisateur doit entrer la réponse. Le formulaire fournit également deux zones de capture du nouveau mot de passe.
6. L'utilisateur édite et soumet le formulaire Question secrète.
7. User Self Care transmet le contenu du formulaire à la chaîne STS ID oublié de sorte à procéder à la validation de la question secrète.
   a. Le module STS assure le suivi des tentatives infructueuses dans un cache interne. Si le nombre dépasse la limite configurée, le module STS envoie une erreur à l'utilisateur.
   b. Le module STS stocke la demande de modification de mot de passe dans un cache interne.
   c. Le module STS envoie à l'utilisateur un e-mail contenant un lien vers l'URL de formulaire de validation de mot de passe oublié. L'e-mail contient un lien auquel s'ajoute une chaîne de requête. La chaîne de requête contient une clé vers l'entrée de cache interne. La clé est utilisée de sorte que l'utilisateur soumis puisse être récupéré et la modification du mot de passe achevée.
8. L'utilisateur demande le lien dans le courrier électronique.
9. User Self Care transmet la requête à la chaîne STS ID oublié. Les modules de chaînes récupèrent les données à partir du cache interne et tentent de modifier le mot de passe.
   - Si des erreurs se produisent, elles s'affichent dans une page d'informations envoyée à l'utilisateur.
   - Si aucune erreur ne survient, User Self Care envoie à l'utilisateur la page d'informations d'accusé de mot de passe oublié. Cette page indique à l'utilisateur que le mot de passe a été modifié.

Opération de suppression de compte

L'opération de suppression de compte suit une opération de flux de tâches.

Flot de tâches d'opération :

- L'utilisateur demande la page de suppression de compte. Cette page doit être une ressource protégée.
- L'utilisateur clique sur un lien de la page.
- L'identité de l'utilisateur est obtenue à partir du contexte authentifié.

- La chaîne STS de suppression de compte est démarrée.
- La chaîne STS de suppression de compte finalise la suppression du compte utilisateur.
- User Self Care renvoie à l'utilisateur la page d'informations de réussite de suppression de compte.

Opération Captcha

Captcha ne représente pas une opération User Self Care distincte. En effet, l'opération Captcha est implémentée en tant que module STS Captcha.

Vous pouvez placer le module STS Captcha d'abord dans des chaînes d'accréditation de service de jeton sécurisé utilisées par User Self Care.

Remarque : IBM a déprécié le client Tivoli Federated Identity Manager Security Token Service (STS) dans cette version.

Si vous utilisez WebSphere 6.X, vous pouvez continuer de vous servir du client Tivoli Federated Identity Manager Security STS tant que Tivoli Federated Identity Manager prend en charge WebSphere 6.X. Lorsque Tivoli Federated Identity Manager arrêtera son support pour WebSphere 6.X, vous devrez utiliser WebSphere Application Server version 7 Update 11 et version ultérieure. Voir API client WS-Trust et WS-Trust Clients pour plus d'informations.
Lorsque le module Captcha est présent, la validation Captcha est effectuée avant l'exécution de toute autre opération.

Pour plus d'informations, voir «Démonstration Captcha», à la page 611.

Opérations d'attributs de registre

User Self Care ne donne pas la possibilité de modifier le schéma de registre utilisateur. Vous devez modifier votre schéma de registre selon nécessaire pour créer les attributs de registre requis pour la prise en charge de vos profils. Vous devez également modifier le schéma pour prendre en charge l'attribut *question secrète*.

User Self Care fournit un exemple de fonction. User Self Care utilise l'attribut LDAP `businessCategory` pour stocker l'attribut de profil de question secrète. L'exemple d'implémentation utilise également l'attribut LDAP `mobile` pour stocker un numéro de téléphone portable pour l'utilisateur.

Lorsque vous déployez User Self Care, vous devez créer un schéma qui peut contenir les attributs de profil que vous devez fourni à vos utilisateurs. Dès que vous avez identifié et défini ces attributs, vous pouvez personnaliser les formulaires HTML et modules STS de sorte à les utiliser.

Dans le cadre d'un déploiement complet, il est nécessaire de créer un schéma qui peut contenir les attributs de profil que vous devez fourni à vos utilisateurs. Lorsque ces attributs sont sélectionnés, vous devez personnaliser les formulaires HTML et les modules STS afin d'utiliser les nouveaux attributs.

Pour plus d'informations, voir le wiki Tivoli Federated Identity Manager :

http://www.ibm.com/developerworks/wikis%2Fdisplay%2Ftivolifederatedidentitymanager%2Fhome.

Opération relative à la question secrète

La *question secrète* représente un mot de passe et un conseil secondaires stockés dans le registre utilisateur en attribut utilisateur. User Self Care considère la gestion de la question secrète comme un autre élément de profil.

User Self Care fournit un exemple d'implémentation de la question secrète question en utilisant l'attribut LDAP businessCategory pour stocker le profil de question secrète. Vous pouvez personnaliser cette implémentation de sorte qu'elle corresponde au mieux à vos besoins.

Les rubriques suivantes décrivent le fonctionnement de l'exemple d'implémentation.

Sélection de question secrète durant l'inscription

Un formulaire d'inscription User Self Care fournit un menu qui permet à un utilisateur de sélectionner une des questions suivantes :

- Nom de jeune fille de la mère
- Lieu de naissance
- Nom du premier animal animal domestique

La sélection d'un ou plusieurs éléments renseigne une zone de formulaire avec des valeurs numériques correspondant à l'index de l'entrée dans la liste. Le nom de cette zone sur les formulaires HTML fourni avec User Self Care est `usc.form.profile.secret.question`.

Une zone de formulaire distinct est utilisé pour spécifier la réponse à la question du texte. Le nom de cet attribut sur les formulaires HTML fourni avec User Self Care est `usc.form.profile.secret.question.answer`.

Lorsque l'utilisateur soumet le formulaire d'inscription, chacun de ces paramètres est transmis à la chaîne d'accréditation STS d'inscription. L'index et la réponse sont concaténés ensemble et stockés dans l'attribut LDAP `businessCategory`.

Affichage de la question secrète durant la gestion des profils

Lorsque l'utilisateur demande le formulaire de gestion des profils, User Self Care extrait les attributs, y compris la question secrète, à partir du registre. Le module STS de gestion des profils analyse l'attribut et détermine l'index spécifiant la question secrète que l'utilisateur a précédemment sélectionnée. User Self Care utilise alors cette valeur d'index pour afficher la valeur appropriée du menu du formulaire de gestion des profils.

Utilisation de la question secrète pour valider l'identité utilisateur

Lorsque l'utilisateur soumet le formulaire de mot de passe oublié, User Self Care utilise l'ID utilisateur pour extraire l'attribut de registre `businessCategory`. Le module STS Mot de passe oublié analyse alors la valeur de l'attribut et renvoie l'index au composant de gestion de présentation. Ce composant utilise l'index pour effectuer une substitution de macro. La substitution fournit une valeur à JavaScript qui mène la sélection de la question secrète correspondante.

Astuce d'implémentation de question secrète

La sécurité de l'approche de la question secrète est améliorée si les utilisateurs peuvent créer leur propre question secrète. Une liste de menu peut s'avérer pratique, mais il existe un risque de fournir des informations d'identification. Les informations sont souvent réutilisées sur plusieurs sites Internet.

Les valeurs par défaut fournies par User Self Care servent d'exemples uniquement. Elles incluent les valeurs souvent utilisées, telles que le nom de jeune fille de la mère, le couleur préférée et le nom du premier animal domestique. Il est recommandé de ne pas utiliser ces valeurs dans un déploiement d'entreprise.

URL User Self Care

User Self Care fournit un ensemble de pages HTML par défaut pour communiquer avec l'utilisateur. Les pages HTML facilitent l'échange de demandes HTTP et réponses.

- «Requêtes HTTP User Self Care»
- «Réponses HTTP User Self Care», à la page 610

Requêtes HTTP User Self Care

La table suivante répertorie les URL demandées par les utilisateurs lors de leur interaction avec User Self Care. Certaines URL sont présentées pour plusieurs requêtes. Chaque URL est unique à une opération User Self Care et est mappée à une chaîne STS. User Self Care détermine la phase de l'opération effectuée en examinant le contenu de la requête.

Remarque : L'authentification utilisateur est requise pour certaines URL. Si la description ne mentionne pas d'authentification utilisateur, cette dernière n'est pas nécessaire.

Tableau 140. Requêtes HTTP

Nom	Méthode HTTP	URI et description de la requête
Page principale	GET	Page personnalisée facultative non hébergée par User Self Care. Vous pouvez décider de créer une page contenant des liens vers les opérations User Self Care, mais qui n'est pas hébergée par User Self Care.
Formulaire de demande d'inscription	GET	/sps/*federation_name*/usc/self/account/create Demande le formulaire d'inscription.
Soumettre la demande d'inscription	POST	/sps/*federation_name*/usc/self/account/create Soumet le formulaire d'inscription.
Extraire l'ID utilisateur	POST	/sps/*federation_name*/usc/global/userid/search Se mappe sur une opération User Self Care distincte qui détermine si un ID utilisateur existe. Cette page apparaît une fois que vous avez cliqué sur un lien du formulaire de demande d'inscription.

Tableau 140. Requêtes HTTP (suite)

Nom	Méthode HTTP	URI et description de la requête
Validation de l'inscription	POST	/sps/*federation_name*/usc/self/account/create/validate Indique l'URL de base dans l'e-mail envoyé à l'utilisateur durant la validation de l'inscription. Une chaîne de requête est ajoutée à l'URL finale.
Formulaire Modifier le mot de passe	GET	/sps/*federation_name*/usc/self/password/update Authentification requise Demande le formulaire de modification du mot de passe.
Soumettre la modification du mot de passe	POST	/sps/*federation_name*/usc/self/password/update Authentification requise Soumet le formulaire de modification du mot de passe.
Formulaire ID oublié	GET	/sps/*federation_name*/usc/self/account/recover/userid Demande le formulaire d'ID oublié.
Soumettre ID oublié	POST	/sps/*federation_name*/usc/self/account/recover/userid Soumet le formulaire d'ID oublié.
Formulaire Mot de passe oublié	GET	/sps/*federation_name*/usc/self/account/recover/password Demande le formulaire de mot de passe oublié.
Formulaire Mot de passe oublié	POST	/sps/*federation_name*/usc/self/account/recover/password Soumet le formulaire de mot de passe oublié.
Formulaire de question secrète du mot de passe oublié	POST	/sps/*federation_name*/usc/self/account/ recover/password/secretquestion Soumet le formulaire de validation de la question secrète. Ce formulaire est présenté à l'utilisateur après soumission du formulaire de mot de passe oublié.
Formulaire de validation du mot de passe oublié	POST	/sps/*federation_name*/usc/self/account/ recover/password/validate Indique l'URL de base dans l'e-mail envoyé à l'utilisateur durant la validation du mot de passe oublié. Une chaîne de requête est ajoutée à l'URL finale.
Formulaire de mise à jour de profil	GET	/sps/*federation_name*/usc/self/profile/update Authentification requise Demande le formulaire de mise à jour de profil.
Soumettre la mise à jour du profil	POST	/sps/*federation_name*/usc/self/profile/update Authentification requise Soumet le formulaire de mise à jour de profil.
Formulaire Suppression de compte	GET	/sps/*federation_name*/usc/self/account/delete Authentification requise Demande le formulaire de suppression de compte.

Tableau 140. Requêtes HTTP (suite)

Nom	Méthode HTTP	URI et description de la requête
Soumettre la suppression de compte	POST	/sps/*federation_name*/usc/self/account/delete Authentification requise Soumet le formulaire de suppression de compte.

Réponses HTTP User Self Care

Cette rubrique répertorie l'ensemble de pages que User Self Care présente à l'utilisateur.

Cet ensemble entre dans les catégories suivantes :

Info
Page d'informations présentant des instructions, des erreurs ou une indication de réussite.

Formulaire
Un formulaire HTML dans lequel l'utilisateur entre des données.

Réacheminement
Un réacheminement HTTP.

Tableau 141. Réponses HTTP

Nom	Type	Description
Formulaire de demande d'inscription	Formulaire	Rassemble les informations suivantes : • ID utilisateur requis • Adresse électronique • Mot de passe • Confirmation du mot de passe • Attributs du profil • Entrée Captcha (facultatif)
Validation de l'inscription	Formulaire	Informe l'utilisateur qu'un e-mail a été envoyé à des fins de validation ou qu'une erreur est survenue.
Résultat de l'inscription	Info	Informe l'utilisateur que son compte a été créé ou qu'une erreur est survenue.
Modifier le mot de passe	Formulaire	Rassemble les informations suivantes : • Ancien mot de passe • Nouveau mot de passe • Confirmation du nouveau mot de passe
Résultat de modification du mot de passe	Info	Informe l'utilisateur que son mot de passe a été modifié ou qu'une erreur est survenue.
ID oublié	Formulaire	Rassemble les éléments suivants pour aider à utilisateur à récupérer un ID utilisateur oublié : • Adresse électronique • Entrée Captcha. Cette valeur est facultative.

Tableau 141. Réponses HTTP (suite)

Nom	Type	Description
Mot de passe oublié	Formulaire	Rassemble les informations suivantes : • ID utilisateur • Entrée Captcha Cette valeur est facultative.
Question secrète du mot de passe oublié	Formulaire	Affiche la question secrète. Rassemble les informations suivantes : • Réponse à la question secrète • Nouveau mot de passe • Confirmation du nouveau mot de passe • Entrée Captcha Cette valeur est facultative.
Post ID oublié	Info	Présente une erreur ou une indication de réussite après une tentative de récupération d'un ID oublié.
Mettre à jour le profil	Formulaire	Présente à l'utilisateur ses détails de profil actuels et rassemble les modifications dans les zones.
Post gestion des profils	Info	Présente une erreur ou une indication de réussite après des opérations de gestion des profils.
Suppression de compte	Formulaire	Présente une icône que l'utilisateur peut utiliser pour supprimer son compte.
Post suppression de compte	Info	Affiche une indication d'erreur ou de réussite après la suppression du compte.

Validation de contenu du formulaire

Considérez l'utilisation d'une validation d'entrée côté client pour vérifier que les zones du formulaire contiennent les données appropriées pour leur type. Les pages HTML de User Self Care contiennent plusieurs exemples.

Démonstration Captcha

Le module STS de démonstration Captcha fournit un exemple d'intégration de Captcha à User Self Care.

User Self Care fournit des pages HTML qui prend en charge les opérations User Self Care. Plusieurs de ces pages sont bonnes pour le type de validation d'entrée fournie par Captcha. Vous pouvez configurer ces pages pour inclure une macro pour Captcha.

L'application User Self Care peut remplacer la valeur de la macro par la source HTML nécessaire de sorte à prendre en charge la démonstration Captcha. Lorsque Captcha n'est pas configuré, la macro n'est pas substituée et les éléments Captcha ne s'affichent pas sur la page.

Lorsqu'un utilisateur demande initialement une page contenant un challenge Captcha, le module STS Captcha est contacté. Le module sélectionne de manière aléatoire une image de l'ensemble des images configurées. Cette image construit la macro sur la page HTML affichée à l'utilisateur.

Après la substitution de la macro, une bloc de code tel que l'exemple ci-dessous s'affiche sur la page.

```
<label for="demo_captcha">
    Veuillez entrer le ou les mots de vérification affichés ci-dessous
(obligatoire)
</label>
<br />
<img src="http://myserver/public/captcha_test/hello.jpg" border="0" />
<br />
<input type="hidden"
       name="usc.demo.captcha.challenge.field"
       id="usc.demo.captcha.challenge.field"
       value="http://myserver/public/captcha_test/hello.jpg" />
<input style="background-color:#F8F8C8;"
       type="text"
       name="usc.demo.captcha.response.field"
       id="usc.demo.captcha.response.field" />
```

Figure 67. Exemple Captcha

Ce bloc fournit une balise `src` et deux zones d'entrée. La balise `src` affiche l'image à l'utiliateur. La première zone d'entrée fournit le nom de l'image. La seconde rassemble l'entrée utilisateur, à savoir le texte situé dans l'image.

Lorsque le formulaire est soumis, les deux zones d'entrée sont fournies dans le module STS Captcha de démonstration. Ce module compare la réponse de l'utilisateur avec la chaîne associée à cet image. Si une correspondance est correcte, la validation est terminée.

Remarque : La première zone d'entrée spécifie une valeur qui est l'URL d'un serveur hébergeant les images s'affichant à l'utilisateur.

Le module de démonstration Captcha est situé dans le répertoire :
Federated_Identity_Manager_installation_directory`/examples/demo/captcha`

Ce répertoire contient :

- Un fichier readme
- Un fichier `com.tivoli.am.fim.demo.sts.captcha.jar` contenant à la fois le code compilé et le code source pour le module de démonstration STS Captcha.
- Un répertoire `captchaTestImages` contenant :
  - Un ensemble de six images JPEG
  - Un fichier `DemoCaptchaImagesInfo.txt` qui affiche le mappage entre les noms de fichiers d'images et la chaîne de texte que l'utilisateur doit entrer lorsqu'il voir l'image associée.

Pour des instructions concernant la configuration, voir «Configuration de la démonstration Captcha», à la page 627.

Chapitre 42. Déploiement de User Self Care

Tivoli Federated Identity Manager installe automatiquement User Self Care en tant que partie du composant d'exécution. Vous n'avez pas besoin d'installer d'autres logiciels, sauf si vous prévoyez d'utiliser Tivoli Access Manager en tant que registre d'utilisateurs cible.

Les administrations qui souhaitent déployer User Self doivent maîtriser la gestion de :

- WebSphere Application Server, y compris l'interface d'administration **wsadmin**.
- Les modules STS Tivoli Federated Identity Manager et chaînes d'accréditation.
- Tivoli Directory Server LDAP.

Les administrations qui souhaitent utiliser Tivoli Access Manager en registre utilisateur cible ou WebSEAL en serveur point de contact doivent maîtriser l'administration de Tivoli Access Manager for e-business.

La liste suivante récapitule les tâches de déploiement de User Self Care et l'ordre dans lequel les effectuer. Avant de commencer une tâche, vérifiez que vous avez procédé aux tâches préalables requises.

1. Configurez un domaine Tivoli Federated Identity Manager. Les étapes de configuration incluent la configuration de la gestion de l'exécution.

   Les étapes de cette tâche sont identiques pour tous les scénarios Tivoli Federated Identity Manager. Aucune tâche de cette rubrique n'est unique à User Self Care. Les liens associés aux tâches vous transfèrent vers les rubriques des tâches courantes du guide de configuration de *Tivoli Federated Identity Manager.*

   «Configuration d'un domaine Tivoli Federated Identity Manager», à la page 614
2. Intégrez User Self Care au registre utilisateur pour votre déploiement. User Self Care prend en charge les registres Tivoli Directory Server et Tivoli Access Manager. Vous êtes dirigé vers les instructions correspondant au type de votre registre.

   «Configuration d'un registre utilisateur», à la page 617
3. La configuration User Self Care est basée sur des valeurs obtenues d'un fichier de réponse. Dans cette tâche, vous renseignez un fichier de réponses avec des valeurs applicables à votre déploiement.

   «Configuration d'un fichier de réponses», à la page 625
4. Utilisez le fichier de réponses créé lors de la tâche précédente pour configurer votre déploiement User Self Care. Cette étape décrit comment afficher les chaînes d'accréditation préconfigurées à partir de l'interface d'administration. Cette étape décrit également comment utiliser l'interface de ligne de commande Tivoli Federated Identity Manager pour déployer votre environnement User Self Care. Eventuellement, vous pouvez configurer la démonstration Captcha.

   «Configuration de User Self Care», à la page 626
5. Lorsque votre déploiement inclut le serveur WebSEAL Tivoli Access Manager en serveur point de contact, vous devez intégrer des fonctions User Self Care à WebSEAL. Ces tâches vous expliquent comment procéder à l'intégration.

   «Intégration de User Self Care à WebSEAL», à la page 673

Configuration d'un domaine Tivoli Federated Identity Manager

Vous devez configurer un domaine Tivoli Federated Identity Manager.

Avant de commencer

Installez les composants Tivoli Federated Identity Manager suivants :

- Gestion de l'environnement d'exécution
- Console d'administration

Procédure

1. Connectez-vous à la console d'administration.
2. Créez un domaine. Pour plus d'informations, voir Chapitre 3, «Configuration de domaine», à la page 25.

Que faire ensuite

Passez à l'étape «Configuration d'un registre utilisateur», à la page 617.

Configuration de domaine

Un domaine Tivoli Federated Identity Manager est un déploiement du composant d'exécution Tivoli Federated Identity Manager sur un serveur WebSphere unique ou sur un cluster WebSphere.

Il existe un domaine par cluster WebSphere. Un environnement comportant un serveur unique ne peut contenir qu'un seul domaine.

Chaque domaine est géré indépendamment. Vous pouvez utiliser l'installation de la console de gestion Tivoli Federated Identity Manager pour gérer plusieurs domaines. Vous ne pouvez gérer qu'un seul domaine à un moment donné. Le domaine à gérer est désigné par *domaine actif*.

Une fois que Tivoli Federated Identity Manager est installé, aucun domaine n'existe. Utilisez la console de gestion pour créer un domaine. Une fois Tivoli Federated Identity Manager installé, le service de gestion est déployé sur un serveur WebSphere (mode serveur unique) ou sur un gestionnaire de déploiement WebSphere (mode cluster WebSphere).

Connectez-vous au service de gestion et sélectionnez un serveur ou cluster WebSphere sur lequel déployer le composant d'exécution Tivoli Federated Identity Manager. Une fois que le module d'exécution est déployé et configuré, vous êtes prêt à configurer les fonctionnalités complémentaires telles que la connexion unique fédérée ou la gestion de la sécurité des services Web.

Dans un environnement WebSphere Network Deployment, le déploiement et la configuration du module d'exécution Tivoli Federated Identity Manager sur les membres du cluster constituent un processus automatisé. Il n'est pas nécessaire d'installer également les logiciels Tivoli Federated Identity Manager ou Tivoli Access Manager sur les ordinateurs WebSphere en cluster.

Le service de gestion Tivoli Federated Identity Manager utilise les services de déploiement d'application de WebSphere Deployment Manager pour déployer et configurer le module d'exécution sur les membres de cluster distribués.

La console de gestion offre un assistant qui vous guidera tout au long de la création du domaine. Les sections ci-après répertorient les propriétés que l'assistant vous invite à spécifier.

Propriétés des noeuds finals de service de gestion de domaine

Hôte Nom de domaine complet de l'**hôte** sur lequel WebSphere Application Server est en cours d'exécution. Par exemple :

```
idp.exemple.com
```

Port de connexion SOAP
Le port SOAP (autonome) par défaut WebSphere Application Server est 8880. Lors de la création d'un domaine pour un serveur WebSphere Application Server qui fait partie d'un cluster WebSphere, le numéro de port SOAP peut s'avérer différent. Par exemple, 8879. Si vous avez un doute sur le numéro de port SOAP correct, utilisez la console d'administration de WebSphere Application Server pour déterminer le port.

Propriétés de sécurité globale WebSphere

WebSphere Application Server est doté d'une option d'activation de sécurité globale. Lorsque la sécurité globale est activée, les propriétés de sécurité doivent être configurées pour le service de gestion Tivoli Federated Identity Manager. La sécurité globale est activée dans la plupart des déploiements.

Nom de l'utilisateur d'administration
Nom de l'administrateur WebSphere Application Server. Par exemple :

```
wsadmin
```

Mot de passe d'administration
Mot de passe de l'administrateur WebSphere Application Server tel qu'il a été spécifié lors de l'installation de WebSphere.

Fichier de clés certifiées SSL
Fichier de clés utilisé par WebSphere Application Server.

Si vous avez installé Tivoli Federated Identity Manager sur un ordinateur hébergeant une installation WebSphere existante, le chemin d'accès par défaut sous Linux ou UNIX est le suivant :

```
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/etc/trust.p12
```

Sous Windows :

```
C:\Program Files\IBM\WebSphere\AppServer\
  profiles\AppSrv01\etc\trust.p12
```

Si vous avez installé l'instance WebSphere intégrée dans le cadre de l'installation de Tivoli Federated Identity Manager, le chemin d'accès par défaut sous Linux ou UNIX est le suivant :

```
/opt/IBM/FIM/ewas/profiles/
 itfimProfile/etc/trust.p12
```

Sous Windows :

```
C:\Program Files\IBM\FIM\ewas\
   profiles\AppSrv01\etc\trust.p12
```

Mot de passe du fichier de clés certifiées SSL
Mot de passe requis pour l'accès au fichier de clés certifiées SSL.

Le mot de passe par défaut pour la clé WebSphere est le suivant :

`WebAS`

Fichier de clés client SSL
Fichier de clés utilisé par WebSphere Application Server.

Ce fichier de clés est un élément de configuration optionnel. Certains déploiements WebSphere ne nécessitent aucun fichier de clés client SSL.

Mot de passe du fichier de clés client SSL
Mot de passe requis pour l'accès au fichier de clés client SSL. Cette zone doit être renseignée lorsque vous avez entré un fichier de clés client SSL.

Nom de serveur ou cluster WebSphere

Lors de la création d'un domaine, l'assistant de domaine vous demande le nom de serveur ou de cluster WebSphere.

Nom du serveur
Nom du serveur WebSphere Application Server sur lequel le service de gestion Tivoli Federated Identity Manager est configuré.

Le serveur correspond à un serveur unique et ne fait pas partie d'un cluster.

Le nom par défaut est créé automatiquement par l'assistant. A titre d'exemple, pour un hôte appelé `host1` :

```
WebSphere:cell=host1Node01Cell,node=host1Node01,server=server1
```

Nom de cluster
Nom du cluster WebSphere Application Server dans lequel le service de gestion Tivoli Federated Identity Manager est configuré.

Propriétés d'environnement Tivoli Access Manager

L'assistant vous invite à préciser su vous devez ou non effectuer la configuration dans un environnement Tivoli Access Manager. N'effectuez *pas* la configuration dans un environnement Tivoli Access Manager si vous utilisez un serveur point de contact autre que WebSEAL. A titre d'exemple, n'effectuez *pas* la configuration dans un environnement Tivoli Access Manager si vous utilisez WebSphere en tant que serveur point de contact.

L'invite affichée par l'assistant est la suivante :

Cet environnement utilise Tivoli Access Manager
Si vous désélectionnez cette case, vous n'aurez besoin de configurer aucune propriété pour Tivoli Access Manager.

Si vous cochez cette case, spécifiez les propriétés indiquées dans le tableau suivant.

Nom d'utilisateur de l'administrateur
Administrateur Tivoli Access Manager. L'ID par défaut est sec_master. Si vous avez choisi un autre ID administrateur lorsque vous avez installé Tivoli Access Manager, entrez l'ID administrateur dans la zone **Administrator Username**.

Mot de passe de l'administrateur
Mot de passe de l'administrateur Tivoli Access Manager.

Nom d'hôte du serveur de règles
Nom d'hôte complet de l'ordinateur qui exécute le serveur de règles de Tivoli Access Manager. Par exemple :

`idp.exemple.com`

Port Numéro de port permettant de communiquer avec le serveur de règles.

Ce nombre correspond au numéro de port que vous avez spécifié lorsque vous avez configuré Tivoli Access Manager. La valeur par défaut est `7135`.

Nom d'hôte du serveur d'autorisations
Nom d'hôte complet de l'ordinateur qui exécute le serveur d'autorisations de Tivoli Access Manager. Par exemple :

`idp.exemple.com`

Port Numéro de port permettant de communiquer avec le serveur d'autorisations.

Ce nombre correspond au numéro de port que vous avez spécifié lorsque vous avez configuré Tivoli Access Manager. La valeur par défaut est `7136`.

Domaine Tivoli Access Manager
Nom du domaine d'administration Tivoli Access Manager que vous avez indiqué lors de la configuration de ce dernier. La valeur par défaut est `Default`.

Configuration d'un registre utilisateur

Intégrez User Self Care au registre utilisateur défini pour votre déploiement.

User Self Care prend en charge ces registres via la configuration de WebSphere Federated Repositories :

- IBM Tivoli Directory Server. Voir «Configuration de Tivoli Directory Server».
- IBM Tivoli Access Manager. Voir «Configuration d'un adaptateur Tivoli Access Manager», à la page 618.
- Microsoft Active Directory. Voir «Configuration d'un serveur Active Directory», à la page 624.

Configuration de Tivoli Directory Server

Configurez WebSphere Federated Repository pour Tivoli Directory Server LDAP.

Pourquoi et quand exécuter cette tâche

N'effectuez pas cette tâche si vous utilisez Tivoli Access Manager en tant que registre utilisateur. Voir «Configuration d'un adaptateur Tivoli Access Manager», à la page 618.

Procédure

1. Connectez-vous à la console d'administration.
2. Sélectionnez l'onglet **Sécurité**, puis **Sécurité globale**.
3. Cliquez sur **Configurer**.
   Cette icône se trouve à droite du menu des **référentiels fédérés**.
4. Cliquez sur l'option d'**ajout d'une entrée de base au domaine**.
5. Cliquez sur **Ajouter le référentiel**.
6. Entrez un nom d'**identificateur de référentiel**.
   Vous pouvez indiquer un nom d'identificateur.
7. Entrez les valeurs dans les zones suivantes :
   - **Type de répertoire**

- **Nom d'hôte principal**
- **Port**
- **Nom distinctif Bind**
- **Mot de passe Bind**

Vous pouvez fournir en option des valeurs pour les zones supplémentaires.

8. Cliquez sur **OK** et sauvegardez. Vous pouvez maintenant voir une page qui demande le **nom distinctif d'une entrée de base qui identifie de manière unique cet ensemble d'entrées dans le domaine**.
9. Entrez un nom d'entrée de base.

   Si nécessaire, voir la documentation WebSphere Application Server relative à WebSphere Federated Repository.

   Remarque : Mémorisez le nom de l'entrée de base. Vous devez l'utiliser lors de la configuration de User Self Care.
10. Cliquez sur **OK** et sauvegardez. La page de configuration pour **defaultWIMFileBasedRealm** s'affiche.
11. Examinez la table intitulée **Référentiels du domaine**. Vérifiez que le nouveau domaine s'affiche, et que l'**entrée de base** est définie sur la valeur que vous entrez.
12. Cliquez sur **OK** et sauvegardez. La console d'administration revient à la page **Sécurité globale**.
13. Cochez la case d'**activation de la sécurité d'application**.
14. Cliquez sur **OK** et sauvegardez.

Que faire ensuite

Passez à la section «Configuration d'un fichier de réponses», à la page 625.

Configuration d'un adaptateur Tivoli Access Manager pour WebSphere Federated Repository

Pour configurer un adaptateur Tivoli Access Manager pour User Self Care, vous devez configurer l'adaptateur et l'ajouter à WebSphere Federated Repository en registre personnalisé.

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Exécutez les tâches suivantes :

1. «Configuration d'un adaptateur Tivoli Access Manager».
2. «Configuration d'un adaptateur en tant que registre personnalisé de WebSphere Application Server», à la page 620.

Si nécessaire, consultez les informations de dépannage dans «Traitement des échecs de connexion à WebSphere Application Server», à la page 622.

Configuration d'un adaptateur Tivoli Access Manager

Configurez cet adaptateur lorsque User Self Care gère le registre Tivoli Access Manager.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Pourquoi et quand exécuter cette tâche

Cet adaptateur utilise l'API Tivoli Access Manager Registry Direct Java API pour effectuer des commandes d'administration telles que la création d'utilisateurs et de groupes. L'installation de Tivoli Access Manager fournit cet adaptateur.

Remarque : Si vous n'utilisez pas d'adaptateur Tivoli Access Manager, ne prenez pas ces instructions en compte. Voir «Configuration de Tivoli Directory Server», à la page 617.

Procédure

1. Vérifiez que vous avez installé Tivoli Access Manager.
2. Vérifiez que vous avez installé et configuré Tivoli Access Manager à l'aide de Tivoli Directory Server en registre utilisateur.
3. Vérifiez que vous avez installé le composant d'exécution Tivoli Access Manager 6.1.1 Java.
4. Copiez *TAM_installation_directory*/java/export/rgy/com.tivoli.pd.rgy.jar dans *WebSphere_installation_directory*/lib.
5. Créez une identité utilisateur Tivoli Access Manager exécutant l'API Java.

   Par exemple :

   ```
   pdadmin -a sec_master -p sec_master_password
   pdadmin sec_master> user create -no-password-policy user_name
   cn=user_name,registry_suffix user_name user_name password
   ( SecurityGroup ivacld-servers remote-acl-users )
   pdadmin sec_master> user modify user_name account-valid yes
   ```

   Dans cet exemple, *user_name* est le nom que vous avez choisi pour l'utilisateur. Il est recommandé d'utiliser les conventions de dénomination suivantes :

   ```
   tamVMMAdapter-machine_name
   ```

   La valeur *registry_suffix* est le suffixe du registre où l'utilisateur doit être stocké. Par exemple :

   ```
   o=ibm,c=us
   ```

6. Accédez à l'ordinateur sur lequel l'adaptateur Tivoli Access Manager doit être configuré. Modifiez le répertoire sur *WebSphere_installation_directory*/lib . Exécutez l'outil **com.tivoli.pd.rgy.until.RgyConfig**.

   Utilisez l'environnement d'exécution IBM Java pour exécuter cet outil. Par exemple :

   ```
   <WebSphere install>/AppServer/java/jre/bin/java
   ```

Tableau 142. Utilisation de l'utilitaire com.tivoli.pd.rgy.util.RgyConfig

Syntaxe :

```
java com.tivoli.pd.rgy.util.RgyConfig properties_file_destination create Default
Default "ldaphostname:389:readwrite:5" "DN" DN_password
```

properties_file_destination
Spécifie le chemin complet vers un répertoire existant et le nom d'un fichier créé à l'exécution de cette commande. Placez le fichier dans un répertoire correspondant à votre déploiement WebSphere Application Server :

- Pour un serveur WebSphere Application Server qui n'est pas en cluster :

  WebSphere_application_server/profiles/*server_name*/config/itfim
- Pour un environnement WebSphere Application Server en cluster (répliqué), créez le fichier sur le DMgr :

  WebSphere_application_server/profiles/*DMgr_server_name*/config/itfim

ldaphostname
Le nom d'hôte du serveur LDAP sur lequel Tivoli Access Manager est configuré. Le nom d'hôte est spécifié dans le fichier de configuration d'exécution Tivoli Access Manager :

Tivoli Access Manager_installation_directory/etc/ldap.conf

389
Le port LDAP par défaut. A modifier selon nécessaire pour votre déploiement.

"DN"
Le nom distinctif (DN) spécifié dans la commande de création d'utilisateur **pdadmin**. Vérifiez que la valeur est entourée de guillemets simples.

DN_password
Le mot de passe pour le nom DN.

Exemple de commande :

```
java com.tivoli.pd.rgy.util.RgyConfig
WebSphere_application/profiles/<server>/config/itfim/tamVMMAdapter.properties
 create Default Default "myldapsystem:389:readwrite:5"
"cn=tamVMMAdapter-myhost,o=ibm,c=us" mypasswordmypassword
```

7. Mettez à jour la configuration selon nécessaire pour votre déploiement WebSphere Application Server :
   - Pour un serveur WebSphere Application Server qui n'est pas en cluster, rechargez la configuration Tivoli Federated Identity Manager.
   - Pour un environnement WebSphere Application Server en cluster (répliqué), effectuez une nouvelle synchronisation complète de WebSphere Application Server et rechargez la configuration Tivoli Federated Identity Manager.

Que faire ensuite

Passez à la section «Configuration d'un adaptateur en tant que registre personnalisé de WebSphere Application Server».

Configuration d'un adaptateur en tant que registre personnalisé de WebSphere Application Server

pour procéder à l'intégration dans WebSphere, configurez l'adaptateur Tivoli Access Manager en tant que registre personnalisé de WebSphere Application Server.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Réalisez la tâche «Configuration d'un adaptateur Tivoli Access Manager», à la page 618.

Pourquoi et quand exécuter cette tâche

Après la configuration de l'adaptateur Tivoli Access Manager avec l'environnement d'exécution Tivoli Access Manager, vous devez configurer l'adaptateur Tivoli Access Manager VMM (Virtual Member Manager) dans WebSphere Application Server en registre personnalisé.

Remarque : Pour des informations sur la configuration des registres personnalisés WebSphere Federated Repository, consultez la documentation liée à WebSphere Application Server. Pour WebSphere Application Server Network Deployment 6.1, voir le centre de documentation de WebSphere.

Procédure

1. Arrêtez WebSphere Application Server.
2. Placez-vous dans le répertoire suivant :

   WebSphere_Installation_directory/profiles/*profile_name*/config/cells/*cell_name*/wim/config
3. Utilisez un éditeur de texte pour ouvrir `wimconfig.xml`.

   Remarque : Sauvegardez `wimconfig.xml` avant de le modifier.
4. Ajoutez un nouvel élément `config:repositories` au fichier. Placez cet élément avant l'élément `config:realmConfiguration`.

   Cette entrée spécifie le nom de classe de l'adaptateur et elle définit un identificateur pour le référentiel. Par exemple, pour spécifier un nom de classe de `com.tivoli.pd.vmm.adapter.tam.TAMRegistryAdapter` et pour définir le référentiel `TAMRegistryAdapter` en tant qu'identificateur :

   ```
   <config:repositories
   adapterClassName="com.tivoli.pd.vmm.adapter.tam.TAMRegistryAdapter"
   id="TAMRegistryAdapter"/>
   ```
5. Sauvegardez le fichier wimconfig.xml, et fermez l'éditeur de texte.
6. Copiez le fichier *TAM_installation_directory*/java/export/vmm_tam_adapter/VMMTamAdapter.jar dans *WebSphere_install_directory*/lib.
7. Démarrez **wsadmin** en mode hors **connexion** :

   ```
   wsadmin -conntype none
   ```
8. Désactivez la pagination dans la configuration de référentiel commun. Définissez le paramètre `supportPaging` pour la commande `updateIdMgrRepository` sur `false` pour désactiver le pagination.

   ```
   $AdminTask updateIdMgrRepository {-id TAMRegistryAdapter
    -supportPaging false }
   ```

   Remarque : Un avertissement s'affiche jusqu'à la fin de la configuration de l'exemple de référentiel.

9. Ajoutez une propriété personnalisée à TAMRegistryAdapter.

   ```
   $AdminTask setIdMgrCustomProperty {-id TAMRegistryAdapter
   -name tamConfFile -value "properties_file_destination"}
   ```

   properties_file_destination
   Le fichier de propriétés créé suite à l'exécution de **com.tivoli.pd.rgy.util.RgyConfig** dans le cadre de la tâche prérequise «Configuration d'un adaptateur Tivoli Access Manager», à la page 618.

10. Ajoutez une entrée de base à la configuration de l'adaptateur à l'aide de la commande **addIdMgrRepositoryBaseEntry** pour indiquer le nom de l'entrée de base pour le référentiel spécifié :

    ```
    $AdminTask addIdMgrRepositoryBaseEntry {-id TAMRegistryAdapter
    base-name base_entry_name }
    ```

    base_entry_name
    Ce nom doit correspondre au suffixe utilisé par le registre utilisateur Tivoli Access Manager.

11. Utilisez la commande **addIdMgrRealmBaseEntry** pour ajouter l'entrée de base au domaine. Cette action relie le domaine au référentiel.

    ```
    $AdminTask addIdMgrRealmBaseEntry {-name defaultWIMFileBasedRealm
    -baseEntry base_entry_name }
    ```

    base_entry_name
    Ce nom doit correspondre à la valeur spécifiée dans la commande précédente.

    defaultWIMFileBasedRealm
    Le nom de domaine par défaut est defaultWIMFileBasedRealm. Si le nom de domaine a été modifié, utilisez le vrai nom de domaine plutôt que defaultWIMFileBasedRealm.

12. Sauvegardez les modifications apportées à votre configuration. Entrez les commandes suivantes pour sauvegarder la nouvelle configuration et fermer l'outil **wsadmin** :

    ```
    $AdminConfig save
    exit
    ```

13. Redémarrez WebSphere Application Server.

Que faire ensuite

Sélectionnez une des tâches suivantes :

- Si vous pouvez vous connecter à WebSphere Application Server, poursuivez avec «Configuration d'un fichier de réponses», à la page 625.
- Si vous ne parvenez pas à vous connecter à WebSphere Application Server, consultez la rubrique «Traitement des échecs de connexion à WebSphere Application Server».

Traitement des échecs de connexion à WebSphere Application Server

Si vous ne pouvez pas vous connecter à la configuration WebSphere Application Server suivante de l'adaptateur, revoyez ces astuces d'identification et résolution des problèmes.

Pourquoi et quand exécuter cette tâche

S'il est impossible de contacter un registre, WebSphere Application Server vous empêche de vous connecter. Cette limitation survient même si le compte d'administration WebSphere Application Server est situé dans un registre différent.

Une configuration incorrecte ou le manque de disponibilité d'un registre requis peut impliquer que WebSphere Application Server vous empêche de vous connecter en administrateur.

Si vous rencontrez ce problème après la configuration de l'adaptateur Tivoli Access Manager, essayez les opérations suivantes :

Procédure

1. Vérifiez que le registre Tivoli Access Manager est accessible. Etant donné que l'adaptateur deTivoli Access Manager ne conserve pas de cache d'authentification, une erreur indiquant qu'il vous est impossible de vous connecter s'affiche dès que le registre n'est pas disponible.
   a. Utilisez **pdadmin** pour vous connecter au registre et effectuez une création de test pour confirmer.
   b. Redémarrez le registre et corrigez tout problème de connexion si nécessaire.
   c. Si le problème persiste, passez à l'étape suivante.
2. Ouvrez le fichier `wimconfig.xml` et vérifiez les paramètres du nouveau code que vous créez.

```
<config:repositories adapterClassName="com.tivoli.pd.vmm.adapter.tam.TAMRegistryAdapter"
id="TAMRegistryAdapter" supportPaging="false">
<config:baseEntries name="o=ibm,c=us"/>
<config:CustomProperties
name="tamConfFile"
value="/opt/IBM/WebSphere/AppServer/profiles/dmgr/config/itfim/tamVMMAdapter.properties"/>
</config:repositories>
```

Figure 68. Exemples de paramètres wimconfig.xml

   - Confirmez que l'emplacement ou le nom du fichier de propriétés est correct.
   - Confirmez que le suffixe est correct pour le registre Tivoli Access Manager.

   Remarque : Si vous modifiez le fichier de configuration, vous devez redémarrer WebSphere Application Server. WebSphere Application Server nécessite que vous vous connectiez en tant qu'administrateur pour arrêter WebSphere Application Server. Toutefois, si vous ne pouvez pas vous connecter, vous devez arrêter le processus WebSphere Application Server. Vous pouvez alors redémarrer WebSphere Application Server sans vous connecter.
3. Si vous n'avez pas identifié de problèmes liés au fichier de configuration lors de l'étape précédente, revenez à la copie de sauvegarde de `wimconfig.xml`.
   a. Enregistrez une copie de sauvegarde du nouveau fichier `wimconfig.xml`.
   b. Restaurez la copie de sauvegarde du fichier `wimconfig.xml` initial.
   c. Redémarrez WebSphere Application Server.

      Remarque : WebSphere Application Server nécessite que vous vous connectiez en tant qu'administrateur pour arrêter WebSphere Application Server. Toutefois, si vous ne pouvez pas vous connecter, vous devez arrêter le processus WebSphere Application Server. Vous pouvez alors redémarrer WebSphere Application Server sans vous connecter.

   Si vous pouvez vous connecter après la restauration du fichier sauvegardé, le problème se situe au niveau de la configuration de l'adaptateur Tivoli Access Manager. Revoyez la configuration et corrigez les erreurs.

Configuration d'un serveur Active Directory

Configurez WebSphere Federated Repository pour Microsoft Active Directory.

Pourquoi et quand exécuter cette tâche

N'effectuez pas cette tâche si vous utilisez Tivoli Access Manager en tant que registre utilisateur. Voir «Configuration d'un adaptateur Tivoli Access Manager», à la page 618.

Procédure

1. Connectez-vous à la console d'administration.
2. Sélectionnez l'onglet **Sécurité**, puis **Sécurité globale**.
3. Cliquez sur **Configurer**.
   Cette icône se trouve à droite du menu des **référentiels fédérés**.
4. Cliquez sur l'option d'**ajout d'une entrée de base au domaine**.
5. Cliquez sur **Ajouter le référentiel**.
6. Entrez un nom d'**identificateur de référentiel**.
   Vous pouvez indiquer un nom d'identificateur.
7. Entrez les valeurs dans les zones suivantes :
   - **Type de répertoire**
   - **Nom d'hôte principal**
   - **Port**
   - **Nom distinctif Bind**
   - **Mot de passe Bind**

   Vous pouvez fournir en option des valeurs pour les zones supplémentaires.
8. Dans la console WebSphere Application Server, sélectionnez **Require SSL communications** (Demander aux communications SSL).

   Remarque : La configuration de communications SSL entre un WebSphere Application Server et un registre utilisateur tel qu'Active Directory exige d'autres étapes. Voir la documentation pour votre version de WebSphere Application Server pour des instructions sur la configuration des connexions SSL avec WebSphere Application Server.
9. Cliquez sur **OK** et sauvegardez. Vous pouvez maintenant voir une page qui demande le **nom distinctif d'une entrée de base qui identifie de manière unique cet ensemble d'entrées dans le domaine**.
10. Entrez un nom d'entrée de base.
    Si nécessaire, voir la documentation WebSphere Application Server relative à WebSphere Federated Repository.

    Remarque : Mémorisez le nom de l'entrée de base. Vous devez l'utiliser lors de la configuration de User Self Care.
11. Cliquez sur **OK** et sauvegardez. La page de configuration pour **defaultWIMFileBasedRealm** s'affiche.
12. Examinez la table intitulée **Référentiels du domaine**. Vérifiez que le nouveau domaine s'affiche, et que l'**entrée de base** est définie sur la valeur que vous entrez.
13. Cliquez sur **OK** et sauvegardez. La console d'administration revient à la page **Sécurité globale**.
14. Cochez la case d'**activation de la sécurité d'application**.

15. Cliquez sur **OK** et sauvegardez.

Que faire ensuite

Passez à la section «Configuration d'un fichier de réponses».

Configuration d'un fichier de réponses

Créez un fichier de réponses et remplissez-le avec les valeurs spécifiques à votre déploiement.

Pourquoi et quand exécuter cette tâche

User Self Care charge la configuration à partir d'un fichier de propriétés XML appelé *fichier de réponses*. Ce fichier contient les réponses aux options de configuration. Dans la plupart des cas, le contenu du fichier de réponse est généré par les choix de l'administrateur lors du déploiement initial. Pour User Self Care, le chargement d'un fichier de propriétés est requis dans le cadre de la configuration initiale.

Procédure

1. Créez un fichier de réponses selon nécessaire pour votre déploiement. Utilisez **wsadmin**:

   ```
   $AdminTask manageItfimUserSelfCare {-operation createResponseFile
   -fileId target_location }
   ```

   La valeur *target_location* représente le chemin complet vers un fichier créé.
2. Déterminez la valeur de chaque paramètre dans le fichier de réponses, tel que requis par votre déploiement.

   Eventuellement, utilisez la feuille de travail suivante pour planifier votre fichier de réponses. La feuille de travail identifie les paramètres requis. Dans le fichier de réponses, vous pouvez rechercher la chaîne `REQUIRED` pour trouver ces paramètres.

   Pour plus d'informations sur chaque paramètre de cette feuille de travail, voir Chapitre 44, «Paramètres de fichier de réponses», à la page 683

Tableau 143. Paramètres du fichier de réponses User Self Care

Paramètre du fichier de réponses	Obligatoire/ facultatif	Valeur par défaut	Votre valeur
AccountCreateLifetime	oui	86400	
AccountRecoveryFailureLifetime	non	86400	
AccountRecoveryFailureLimit	non	3	
AccountRecoveryFailureLockoutTime	non	86400	
AccountRecoveryLookupAttribute	non	mail	
AccountRecoveryLookupField	non	aucun	Cette zone est dépréciée.
AccountRecoveryValidationAttributes	non	mail	
AccountRecoveryValidationLifetime	non	86400	
AttributeMappingFilename	oui	aucun	
BaseURL	oui	aucun	
CaptchaSTSModuleId	oui	default-usc-captcha-noop	
DemoCaptchaImageAndKeyList	Oui, en cas d'utilisation de Captcha	Contenu fixe.	Ne pas modifier.
DemoCaptchaImageRootURL	Oui, en cas d'utilisation de Captcha	aucun	
EnrollmentEmailSender	oui	aucun	
EntitySuffix	oui	o=ibm,c=us	

Tableau 143. Paramètres du fichier de réponses User Self Care (suite)

Paramètre du fichier de réponses	Obligatoire/ facultatif	Valeur par défaut	Votre valeur
GroupMembershipGroups	non	aucun	
PasswordRecoveryEmailSender	oui	aucun	
ProfileManagementAttributes	oui	businessCategory roomNumber mobile mail	
SecretQuestionMinimumNumber	non	2	
SecretQuestionMaximumNumber	non	3	
SecretQuestionRequiredForValidationNumber	non	2	
SMTPAuthenticatePassword	Non, à moins que votre serveur SMTP ne l'exige	aucun	
SMTPAuthenticateUsername	Non, à moins que votre serveur SMTP ne l'exige	aucun	
SMTPServerName	oui	aucun	

3. Mettez à jour votre fichier de réponses avec les valeurs.
4. Sauvegardez le fichier.

Que faire ensuite

Poursuivez avec la rubrique : «Configuration de User Self Care».

Configuration de User Self Care

Suivez les étapes de cette procédure pour configurer User Self Care avec le déploiement Tivoli Federated Identity Manager.

Avant de commencer

Vérifiez que vous avez réalisé les tâches de configuration prérequises :

1. «Configuration d'un domaine Tivoli Federated Identity Manager», à la page 614
2. «Configuration d'un registre utilisateur», à la page 617
3. «Configuration d'un fichier de réponses», à la page 625

Pourquoi et quand exécuter cette tâche

Réalisez les étapes suivantes dans l'ordre. Les instructions pour chaque tâche fournissent un lien vers la tâche suivante. La liste des tâches s'affiche ici en présentation générale.

Procédure

1. «Affichage des chaînes d'accréditation», à la page 627
2. «Configuration de la démonstration Captcha», à la page 627

   Ignorez cette étape si vous n'avez pas l'intention d'utiliser la démonstration Captcha.
3. «Utilisation d'un fichier de réponses pour configurer User Self Care», à la page 628
4. «Configuration d'un serveur point de contact», à la page 629
5. «Intégration de User Self Care à WebSEAL», à la page 673

Ignorez cette étape si vous utilisez WebSphere Application Server en tant que serveur point de contact.

Affichage des chaînes d'accréditation

Vous pouvez configurer Tivoli Federated Identity Manager de sorte à afficher les chaînes d'accréditation créées par défaut pour User Self Care.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser la console d'administration pour afficher chaque chaîne d'accréditation. Les chaînes d'accréditation correspondent à une ou plusieurs opérations User Self Care. Lorsque vous affichez les chaînes d'accréditation, vous voyez les modules STS accomplissant l'opération. Vous pouvez alors personnaliser les modules et chaînes selon votre déploiement.

Remarque : Pour des informations sur la personnalisation de User Self Care, voir le wiki Tivoli Federated Identity Manager :

http://www.ibm.com/developerworks/wikis/display/tivolifederatedidentitymanager/Home

Procédure

1. Connectez-vous à la console d'administration.
2. Accédez au panneau Gestion des noeuds d'exécution.
3. Dans la partie de propriété personnalisée du panneau, sélectionnez l'entrée de menu pour `STS.showUSCChains`.
4. Définissez la valeur sur `true`.
5. Sauvegardez la configuration.
6. A l'invite, chargez les modifications de configuration.
7. Redémarrez WebSphere Application Server pour actualiser les commandes de gestion disponibles à **wsadmin**.

Que faire ensuite

Sélectionnez une des étapes suivantes :

- Si vous souhaitez utiliser la démonstration Captcha, poursuivez avec «Configuration de la démonstration Captcha».
- Si vous ne souhaitez pas utiliser la démonstration Captcha, poursuivez avec «Utilisation d'un fichier de réponses pour configurer User Self Care», à la page 628.

Configuration de la démonstration Captcha

Vous pouvez éventuellement configurer la démonstration Captcha dans le cadre de votre déploiement User Self Care.

Avant de commencer

Vérifiez que vous avez réalisé toutes les tâches de configuration prérequises :

- «Configuration d'un domaine Tivoli Federated Identity Manager», à la page 614
- «Configuration d'un registre utilisateur», à la page 617
- «Configuration d'un fichier de réponses», à la page 625
- «Affichage des chaînes d'accréditation»

Procédure

1. Hébergez les fichiers d'images fournis sur un serveur Web accessible à vos utilisateurs.

   Assurez-vous que vous connaissez l'emplacement de l'URL racine des images utilisé lors de la configuration du module STS Captcha. La valeur est stockée dans le paramètre DemoCaptchaImageRootURL du fichier de réponses.
2. Activez le plug-in :
   a. Copiez le fichier jar Captcha dans le répertoire de plug-ins de Tivoli Federated Identity Manager. Par exemple, copiez :

      FIM_install_dir/examples/demo/captcha/com.tivoli.am.fim.demo.sts.captcha.jar

      dans le répertoire :

      TFIM_install_dir/plugins
   b. A l'aide du panneau Gestion des noeuds d'exécution, cliquez sur l'icône **Publier les plug-ins**.
   c. Cliquez sur **Charger les modifications de configuration**.
3. Utilisez le panneau d'instances du module pour créer une instance de DemoCaptchaSTSModule. Définissez le nom de l'instance du module sur la valeur `usc-captcha-demo`.

Que faire ensuite

Passez à la section «Utilisation d'un fichier de réponses pour configurer User Self Care».

Utilisation d'un fichier de réponses pour configurer User Self Care

Utilisez le fichier de réponses que vous avez créé précédemment pour fournir les propriétés nécessaires à la commande de configuration pour User Self Care.

Avant de commencer

Vérifiez que vous avez réalisé les tâches de configuration prérequises :

- «Configuration d'un domaine Tivoli Federated Identity Manager», à la page 614
- «Configuration d'un registre utilisateur», à la page 617
- «Configuration d'un fichier de réponses», à la page 625
- «Affichage des chaînes d'accréditation», à la page 627
- Si vous utilisez la démonstration Captcha, assurez-vous qu'elle est configurée. Voir «Configuration de la démonstration Captcha», à la page 627.

Procédure

1. Obtenez votre fichier de réponses configuré.
2. Exécutez **wsadmin**.

   ```
   wsadmin.sh -username WebSphere_adminstrator_name -password password
   ```
3. Chargez le fichier de réponses :

   ```
   $AdminTask manageItfimUserSelfCare {-operation configure -fimDomainName
   domain_name -federationName federation_name
    -fileId response_file_path }
   ```

   Fournissez ces valeurs :

domain_name
Nom du domaine Tivoli Federated Identity Manager que vous avez créé.

federation_name
Nom de la fédération Tivoli Federated Identity Manager que vous avez créée.

response_file_path
Emplacement de votre fichier de réponses User Self Care.

4. Rechargez la configuration Tivoli Federated Identity Manager.

```
$AdminTask reloadItfimRuntime {-fimDomainName domain_name }
```

Fournissez cette valeur :

domain_name
Nom du domaine Tivoli Federated Identity Manager que vous avez créé.

Que faire ensuite

Poursuivez avec la rubrique : «Configuration d'un serveur point de contact».

Configuration d'un serveur point de contact

Vous devez configurer un serveur point de contact pour User Self Care.

Sélectionnez les instructions pour le type de serveur point de contact utilisé par votre déploiement :

- «Configuration de WebSphere Application Server en tant que serveur point de contact»
- «Configuration de WebSEAL en tant que serveur point de contact», à la page 630

Configuration de WebSphere Application Server en tant que serveur point de contact

Vous pouvez configurer WebSphere Application Server en tant que serveur point de contact pour User Self Care.

Procédure

1. Utilisez **wsadmin** pour activer le type de point de contact **WebSphere**.

   Utilisez les commandes **wsadmin** :

```
$AdminTask manageItfimPointOfContact {-operation activate
-uuid uuid4f3d17d-0106-w412-r36b-a0d5ecc604ba
-fimDomainName your_domain_name}
```

```
$AdminTask reloadItfimRuntime {-fimDomainName your_domain_name}
```

2. Connectez-vous à la console d'administration.
3. Sélectionnez **Enterprise Applications > ITFIMRuntime > Rôle de sécurité pour le mappage utilisateur/groupe**.
4. Mettez à jour le rôle d'application **FIMUserSelfCareAnyAuthenticated** avec **AnyAuthenticated**.
5. Sauvegardez la configuration WebSphere Application Server.
6. Redémarrez WebSphere Application Server.

Que faire ensuite

Consultez les instructions liées au réglage des performances dans le Chapitre 43, «Réglage de User Self Care», à la page 679.

Configuration de WebSEAL en tant que serveur point de contact

Vous pouvez configurer WebSEAL en serveur point de contact pour User Self Care.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Procédure

1. Déterminez l'emplacement de votre fichier **tfimcfg.jar**.

   Ce fichier est situé dans la hiérarchie sous le répertoire d'installation Tivoli Federated Identity Manager. Sous UNIX, le chemin est le suivant :

   ```
   /opt/IBM/FIM/tools/tamcfg/tfimcfg.jar
   ```

2. Exécutez l'outil `tfimcfg`.

   ```
   java -jar tfimcfg.jar -action tamconfig -cfgfile /opt/pdweb/etc/webseald-default.conf
   ```

   Remarque : Si la norme FIPS (Federal Information Processing Standards) est activée pour votre environnement, une fabrique de connexions sécurisées doit être indiquée. Par exemple :

   ```
   java -jar /download_dir/tfimcfg.jar -action tamconfig
   -cfgfile webseald-instance_name.conf -sslfactory TLS
   ```

   Remarques sur l'utilisation :

   - Les chemins de fichier peuvent varier selon votre installation et votre instance WebSEAL.
   - La valeur Le port HTTP Tivoli Federated Identity Manager est le port 9080. Ce port est également le port WC_defaulthost pour WebSphere Application Server.
   - Ne spécifiez pas d'ID utilisateur administrateur ni de mot de passe Tivoli Federated Identity Manager facultatif.
   - Répondez `non` à la question `Utiliser la connexion SSL au serveur ITFIM`.
   - Sélectionnez `uscfed` dans la liste des fédérations à configurer.

Que faire ensuite

Poursuivez avec la rubrique : «Intégration de User Self Care à WebSEAL», à la page 673.

Modification des vérifications d'ID utilisateur et de mot de passe

Les données utilisateur peuvent inclure des informations relatives à l'utilisateur, telles que le nom d'utilisateur, le mot de passe, l'adresse électronique et la question secrète et la réponse. Les données utilisateur sont vérifiées et validées dans les pages HTML, une règle de mappage et un référentiel fédéré. La validation des données utilisateur dans le référentiel fédéré n'est pas abordée dans le présent document.

Les données utilisateur sont vérifiées et validées à trois endroits :

- Dans JavaScript, dans les pages HTML de User Self Care

Dans la configuration par défaut de User Self Care, le langage JavaScript dans les pages HTML valide les données utilisateur suivantes :

- Les valeurs de zones obligatoires manquantes
- Les caractères non valides : `[ ] / < > ( ) , ; : \" = "`
- La présence dans la zone de numéro de mobile de valeurs autres que des chiffres, des espaces et les caractères suivants : `( ) -`

Pour plus d'informations, voir «Présentation de la fonction de validation HTML» , à la page 632.

- Dans la règle de mappage

  Dans la configuration par défaut de User Self Care, la règle de mappage valide les détails suivants dans les données utilisateur :

  - Longueur du mot de passe
    - La longueur minimale de mot de passe est de sept caractères.
    - Le nombre minimal de caractères alphanumériques dans un mot de passe est fixé à quatre.
    - Le nombre minimal de caractères non alphanumériques dans un mot de passe est fixé à un.
    - Le nombre maximal de caractères répétés dans un mot de passe est fixé à deux.
  - Longueur du nom d'utilisateur

    La longueur maximale de nom d'utilisateur est de 256 caractères.
  - Le nom d'utilisateur et le mot de passe doivent contenir uniquement des lettres ou des chiffres.

  Pour plus d'informations, voir «Présentation de la fonction de validation du fichier de mappage», à la page 632.

- Dans le référentiel fédéré

  Le serveur LDAP peut appliquer des limites aux valeurs des données utilisateur. Par exemple, Active Directory. Les règles sur les noms d'utilisateur et les mots de passe qui sont définies dans les stratégies de compte peuvent imposer un mot de passe utilisateur de plus de sept caractères.

Pour modifier les règles sur les noms d'utilisateur ou les mots de passe, un ou plusieurs des fichiers suivants doivent être modifiés. Pour plus d'informations, voir «Modificatino de la validation de nom d'utilisateur et de mot de passe», à la page 634.

Page HTML	**Description de la page**
`enrollment.html`	Formulaire d'inscription.
`changepassword.html`	Formulaire de changement de mot de passe.
`forgotid.html`	Formulaire qui extrait l'ID utilisateur à l'aide de l'adresse électronique enregistrée.
`forgotpassword.html`	Formulaire qui réinitialise le mot de passe en cas d'oubli de celui-ci.
`secretquestion.html`	Page qui s'affiche pour accepter la réponse à la question secrète et le nouveau mot de passe dans le flux de travail mot de passe oublié.
`profile.html`	Formulaire qui met à jour les informations de profil de l'utilisateur.

Présentation de la fonction de validation HTML

La fonction JavaScript `testInput(...)` valide le nom d'utilisateur et le mot de passe sur les pages HTML.

Cette fonction est présente sur toutes les pages HTML qui valident le nom d'utilisateur et le mot de passe.

La fonction suivante est un exemple tiré de la page `enrollment.html`.

```
function testInput(required, fieldName, fieldVal){
   ...
if ( required && fieldVal == "" ) {
    ...
                    return false;
                }
if ( fieldVal.match(illegalChars) ) {
   ...
                    return false;
               }
    return true;
            }
```

Les paramètres d'entrée acceptés sont les suivants :

- `required` : ce paramètre accepte les valeurs true ou false. Si ce paramètre a pour valeur true, `fieldName` est une zone obligatoire.
- `fieldName` : ce paramètre correspond au nom de la zone.
- `fieldVal` : ce paramètre correspond à la valeur qui est fournie par l'utilisateur pour la zone `fieldName`.

Tableau 144. Conditions trouvées dans la fonction de validation HTML

Critères	Description

`if ( required && fieldVal == "" ) {`
`...`
`return false;`
`}`
	Recherche toutes les zones obligatoires.

`if ( fieldVal.match(illegalChars) ) {`
`...`
`return false;`
`}`
	Recherche les caractères qui ne sont pas valides.

Présentation de la fonction de validation du fichier de mappage

La règle de mappage comporte différentes variables et fonctions qui vérifient et valident les données utilisateur dans User Self Care.

variable	Valeur
`MIN_PASSWORD_LENGTH`	7
`MIN_PASSWORD_ALPHA`	0
`MIN_PASSWORD_NON_ALPHA`	0
`MAX_PASSWORD_REPEAT_CHARS`	2

Fonction	Description

`Fonction validUsernameCharacter(cp) {...}`	Renvoie true uniquement lorsque le caractère dans le nom d'utilisateur est une lettre ou un chiffre.
`Fonction validPasswordCharacter(cp) {...}`	Renvoie true uniquement lorsque le caractère dans le mot de passe est une lettre ou un chiffre.

`Fonction checkUsername(helper) {`
` ...`
`    if (userid.length() > MAX_USERNAME_LENGTH) {`
`//Checking for maximum length of username`
` ...`
`return;`
`}`
`  ...`
`for (var i = 0; i < cp.length; i++) {`
`// only allow letters and numbers in usernames`
`if (validUsernameCharacter(cp[i]) == false) {`
`...`
`return;`
`}`
` }`
`}`
	Vérifie la longueur du nom d'utilisateur et si celui-ci comporte des caractères autorisés. Elle appelle
	`validUsername Character`
	
	Elle utilise la valeur dans la variable suivante :
	`MAX_USERNAME _LENGTH`

`Fonction checkPassword(helper) {`
`...`
`if (password.length() < MIN_PASSWORD_LENGTH) {`
`//checking for minimum password length`
`...`
`return;`
` }`
`...`
`    for (var i = 0; i < cp.length; i++) {`
` if (validPasswordCharacter(cp[i]) == false) {`
` //checking for valid characters in the password-only`
` letters and numbers`
`     ...`
`     return;`
` }`
` ...`
`    }`
`    if (alphas < MIN_PASSWORD_ALPHA) {`
` //password should not have less than`
`MIN_PASSWORD_ALPHA alphabets`
` ...`
` return;`
`    }`
`    if (nonalphas < MIN_PASSWORD_NON_ALPHA) {`
` //password should not have less than`
`MIN_PASSWORD_NON_ALPHA`
` ...`
` return;`
`    }`
`    if (repeats > MAX_PASSWORD_REPEAT_CHARS) {`
`  // password cannot have more than`
`MAX_PASSWORD_REPEAT_CHARS repeat     //characters`
` ...`
` return;`
`    }`
`}`
	Vérifie les données de mot de passe suivantes :
	• Longueur minimale
	• Caractères valides
	• Nombre autorisé de caractères alphabétiques et de caractères non alphabétiques
	• Nombre maximal de caractères répétés

Condition	Description
`if (CHECK_PASSWORD[operation] == 1) { System.out.println("USC DEBUG MAPPING: enforcing default password policy."); checkPassword(helper); }`	Ces conditions représentent les points d'entrée dans la règle de mappage.
`if (CHECK_USERNAME[operation] == 1) { System.out.println("USC DEBUG MAPPING: enforcing default username policy."); checkUsername(helper); }`	
`CHECK_PASSWORD[operation]=1`	Par défaut, toutes les opérations sont mappées à cette condition.
`var CHECK_PASSWORD = { "/usc/self/account/create/post": 1, "/usc/self/account/recover/password/secretquestion/post": 1, "/usc/self/password/update/post": 1 };`	Vous pouvez modifier cette condition en changeant le tableau et en ajoutant d'autres conditions pour les gérer.
`if (CHECK_PASSWORD[operation] == 1) { System.out.println("USC DEBUG MAPPING: enforcing default password policy."); checkPassword(helper); }`	

Modificatino de la validation de nom d'utilisateur et de mot de passe

Vous pouvez modifier les différentes méthodes de validation ou de vérification des données utilisateur dans User Self Care.

Pourquoi et quand exécuter cette tâche

La modification de `enrollment.html` est présentée dans cette procédure. Vous devez répéter les étapes pour tous les autres fichiers HTML concernés.

- `enrollment.html`
- `forgotid.html`
- `forgotpassword.html`
- `secretquestion.html`
- `profile.html`

Procédure

1. Editez la page HTML.

   Si vous souhaitez retirer la validation de zone requise pour la zone ID utilisateur

   remplacez l'entrée de la fonction `testInput(required, fieldName, fieldVal` dans

   ```
   if ( ! testInput(true,"User ID", useridTF.value) ){
                       return true;
                   }
   ```

   par

```
if ( ! testInput(false,"User ID", useridTF.value) ){
                    return true;
                }
```

Si vous souhaitez modifier le libellé de la zone ID utilisateur

remplacez

```
<label for="usc.form.userid">
                ID utilisateur (obligatoire)
            </label>
```

par

```
<label for="usc.form.userid">
                ID utilisateur
            </label>
```

Si vous ne souhaitez pas activer la vérification des caractères suivants qui ne sont pas valides dans l'ID utilisateur, l'adresse électronique, le mot de passe et la réponse à une question secrète :

- [
-]
- /
- <
- >
- (
-)
- ,
- ;
- :
- \
- "
- =
- "

recherchez la fonction `testInput(required, fieldName, fieldVal)` et mettez la section en commentaire.

```
if ( fieldVal.match(illegalChars) ) {
      var illegalCharsStr = "The following characters are not
   allowed: [ ] \\ / < > ( ) , ; : \" = ";
       var invalidCharsFieldStr = "The following field contains
   illegal characters: " + fieldName + "<br> " + illegalCharsStr;
        printWarning(invalidCharsFieldStr);
        return false;
    }
```

Si vous ne souhaitez pas vérifier la présence de caractères autres que des chiffres, des espaces, () et - dans la zone de numéro de mobile

Recherchez la fonction `doSubmit()` mettre la section en commentaire.

```
// var mobileCleaned = mobileTF.value.replace(/[\(\)\-\ ]/g, '');
//    if (isNaN(mobileCleaned)) {
//    var illegalMobileCharsStr = "The mobile number can only contain
//    numbers, spaces and the following characters: ( ) - ";
//    printWarning(illegalMobileCharsStr);
//    return true;
// }
```

2. Publiez les pages sur l'environnement d'exécution Tivoli Federated Identity.
3. Modifiez le fichier de règles de mappage.

a. Sauvegardez le fichier de règles de mappage JavaScript.
b. Ouvrez le fichier de règles de mappage JavaScript dans un éditeur de texte.
c. Recherchez les variables suivantes et modifiez-les de façon appropriée.
   - MIN_PASSWORD_LENGTH
   - MIN_PASSWORD_ALPHA
   - MIN_PASSWORD_NON_ALPHA
   - MAX_PASSWORD_REPEAT_CHARS
   - MAX_USERNAME_LENGTH
d. Effectuez des modifications en fonction des scénarios décrits en détail dans le tableau ci-après.

   Si vous souhaitez autoriser des caractères supplémentaires dans la zone Nom d'utilisateur qui soient autres que des lettres et des chiffres

   Recherchez `validUsernameCharacter(cp)` et remplacez la valeur de retour `Character.isLetterOrDigit(cp)` de façon appropriée.

   Remarque : Il se peut qu'une barre oblique inversée (\) soit nécessaire pour certains caractères spéciaux.

   Si vous souhaitez autoriser n'importe quel caractère dans la zone Nom d'utilisateur

   Remplacez `Character.isLetterOrDigit(cp)` par `true`. Le référentiel fédéré que vous utilisez doit autoriser ce caractère spécial.

   Si vous souhaitez avoir des règles différentes pour un flux de travail différent

   Modifiez les variables `CHECK_USERNAME` et `CHECK_PASSWORD`.

   Par exemple, si vous souhaitez définir un paramètre de règles différent lorsque l'utilisateur change le mot de passe, exécutez les étapes ci-après.

   1) Remplacez la valeur de la variable `CHECK_PASSWORD`.

```
var CHECK_PASSWORD = {
"/usc/self/account/create/post": 1,
    "/usc/self/account/recover/password/secretquestion/post": 1,
    "/usc/self/password/update/post": 2
};
```

   2) Introduisez une nouvelle condition.

```
if (CHECK_PASSWORD[operation] == 2) {
     System.out.println("USC DEBUG MAPPING:
enforcing modified password  policy.");
     modifiedCheckPassword(helper);
 }
```

   3) Introduisez la nouvelle fonction `modifiedCheckPassword(helper)`.

```
function modifiedCheckPassword(helper) {
     var password = helper.getNewPassword();
     var pwdattr = [ "usc.form.password.new",
  "usc.form.password.new.confirm" ];

     if (helper.getUserRecoverableError()) {
 return;
     }

     if (password.length() < 10) {
       helper.setUserRecoverableError("Password should
be atleast 10  characters", pwdattr);
```

```
helper.setSTSOutputPageID(FORM_PAGE_IDS[operation]);
return;
    }
}
```

4. Sauvegardez les modifications dans le fichier.
5. Appliquez la règle de mappage modifiée aux chaînes du service d'accréditation.
   a. Connectez-vous à Integrated Solutions Console.
   b. Accédez à **Tivoli Federated Identity Manager** > **Configuration du service d'accréditation** > **Chaînes du service d'accréditation**.
   c. Sélectionnez **USC** sous **Afficher les types de chaîne**.
      1) Si vous modifiez des règles qui sont liées uniquement au nom d'utilisateur, modifiez les chaînes ci-après.
         - **Chaîne par défaut uscCreateAccount**
         - **Chaîne par défaut uscDeleteAccount**
         - **Chaîne par défaut uscForgottenId**
         - **Chaîne par défaut uscProfileManagement** (les deux chaînes)
      2) Si vous modifiez des règles qui sont liées uniquement au mot de passe, modifiez les chaînes ci-après.
         - **Chaîne par défaut uscChangePassword**
         - **Chaîne par défaut uscCreateAccount**
         - **Chaîne par défaut uscDeleteAccount**
         - **Chaîne par défaut uscForgottenPassword** (les deux chaînes)
         - **Chaîne par défaut uscProfileManagement** (les deux chaînes)
      3) Si vous modifiez les règles qui sont liées au nom d'utilisateur et au mot de passe, modifiez les chaînes ci-après.
   d. Sélectionnez le mappage de chaîne que vous souhaitez modifier.
   e. Cliquez sur **Propriétés**.
   f. Sous **Modules de la chaîne du service d'accréditation**, sélectionnez **Module de mappage par défaut**.
   g. Cliquez sur **Propriétés**.
   h. Sous Règle de mappage d'identité pour le chaînage du module partenaire **Chaîne par défaut uscCreateAccount**, cliquez sur **Modifier la règle** .
   i. Cliquez sur **Importer le fichier**.
   j. Cliquez sur **OK**.
   k. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**

   Remarque : Répétez les étapes pour toutes les chaînes que vous souhaitez modifier.

Activation de la fonction des questions secrètes multiples

Utilisez la fonction des questions secrètes multiples pour améliorer la sécurité de la validation des données d'identification de l'utilisateur.

- «Questions secrètes dans User Self Care», à la page 638
- «Mise à jour des paramètres de configuration pour les fédérations User Self Care existantes», à la page 638
  - «A propos de la migration de chaîne STS dans User Self Care», à la page 639
  - Activation de l'utilisation de sel de cryptage et du hachage sur les fédérations User Self Care existantes

Questions secrètes dans User Self Care

Une question secrète est une question à laquelle l'utilisateur doit répondre lors de l'inscription. Il s'agit généralement d'informations personnelles connues uniquement de l'utilisateur. Dans User Self Care, les questions secrètes et les réponses permettent de vérifier l'identité des utilisateurs lorsqu'ils ont oublié leur mot de passe.

Les utilisateurs sont tenus de répondre correctement aux questions secrètes pour que leur identité soit vérifiée. Exemples de questions secrètes :

• Quel est le nom de jeune fille de votre mère ?
• Quel est le nom de votre premier animal domestique ?

Chaque paire ID question-réponse est alors concaténée et stockée dans un attribut à plusieurs valeurs. Par défaut, il s'agit de l'attribut `businessCategory`. Si vous activez la fonction d'utilisation de sel de cryptage et de hachage, la réponse initiale n'est jamais stockée dans le protocole LDAP. Cette fonction permet d'améliorer la sécurité relative au stockage des réponses.

La configuration de la question secrète est stockée dans :

• l'interface utilisateur de stockage des questions secrètes sur les pages HTML de User Self Care ;
• l'interface utilisateur de réponse aux questions secrètes sur les pages HTML de User Self Care ;
• la configuration relative au stockage et à l'extraction des questions secrètes dans et depuis l'attribut LDAP dans le fichier de mappage ;
• les configurations des modules STS (Security Token Service).

Remarque : Par défaut, l'ID de la question secrète est stocké dans l'attribut LDAP. La chaîne de la question secrète est stockée dans les fichiers HTML de User Self Care.

L'identité d'un utilisateur peut être compromise lorsque d'autres personnes connaissent la réponse à la question secrète. Vous pouvez configurer User Self Care pour qu'il utilise plus d'une question secrète pour valider l'identité des utilisateurs. Des questions secrètes multiples permettent de valider les utilisateurs de façon plus sécurisée lorsque ces derniers demandent une réinitialisation de mot de passe.

Mise à jour des paramètres de configuration pour les fédérations User Self Care existantes

Après avoir installé Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4, vous devez mettre à jour la configuration pour les fédérations User Self Care.

A propos de la migration de chaîne STS dans User Self Care :

Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4, met à jour automatiquement les chaînes STS. La migration affecte toutes les fédérations User Self Care existantes.

Les modifications suivantes sont appliquées après l'installation du groupe de correctifs 4 :

- Les chaînes STS sont mises à jour pour prendre en charge les questions secrètes multiples.
- La propriété personnalisée d'exécution `USC.SecretQuestion.SaltAndHash.Enabled` a pour valeur `false`.

Chaînes mises à jour	Modules ajoutés	Propriétés
`uscAccountCreate`	`USCSecretQuestionStoreSTSModule` en mode `validation`	• **Nombre minimal de questions secrètes auxquelles un utilisateur doit répondre**. La valeur par défaut est **1**. • **Nombre maximal de questions secrètes auxquelles un utilisateur peut répondre**. La valeur par défaut est **1**.
`uscProfileManagement`	`USCSecretQuestionStoreSTSModule` en mode `validation`	• **Nombre minimal de questions secrètes auxquelles un utilisateur doit répondre**. La valeur par défaut est **1**. • **Nombre maximal de questions secrètes auxquelles un utilisateur peut répondre**. La valeur par défaut est **1**.
`uscForgottenPassword`	`USCSecretQuestionSTSModule` en mode `émission`	**Nombre minimal de questions secrètes auxquelles un utilisateur doit répondre correctement pour valider son identité**. La valeur par défaut est **1**.

Activation de l'utilisation de sel de cryptage et du hachage sur les fédérations User Self Care existantes :

Vous pouvez renforcer la sécurité relative au stockage des réponses aux questions secrètes en les chiffrant.

Pourquoi et quand exécuter cette tâche

En cryptographie, l'utilisation de sel de cryptage est une méthode qui chiffre des entrées, telles que des mots de passe, en leur ajoutant une chaîne aléatoire. Cette méthode permet de renforcer la sécurité des mots de passe. Le hachage est une méthode qui utilise un algorithme pour convertir des données en une valeur de taille fixe. Dans User Self Care, vous pouvez utiliser ces méthodes de chiffrement pour les réponses aux questions secrètes. L'activation de la fonction d'utilisation de sel de cryptage et de hachage permet de renforcer la sécurité relative au stockage des valeurs de question secrète. Pour activer l'utilisation de sel de cryptage et le hachage sur les valeurs de question secrète existantes, affectez la valeur `true` à la propriété personnalisée d'exécution `USC.SecretQuestion.SaltAndHash.Enabled` dans la console Integrated Solutions Console.

Migration des réponses aux questions secrètes User Self Care dans LDAP à l'aide d'un nouveau format :

Stockez les valeurs de question secrètes à l'aide d'un format plus sécurisé. Utilisez l'outil de migration LDAP pour migrer les réponses aux questions secrètes User Self Care dans un format haché et utilisant du sel de cryptage.

Avant de commencer

- Installez Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4
- Configurez User Self Care

Remarque :

Vous n'avez pas besoin de mettre à jour LDAP si vous configurez User Self Care pour la première fois dans Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4 au minimum.

ATTENTION :
La migration des réponses aux questions secrètes est irréversible. Le hachage d'une réponse à une question secrète est à sens unique. Vous devez sauvegarder les réponses aux questions secrètes d'origine avant d'exécuter la migration. Cette copie de sauvegarde vous permettra de restaurer les valeurs d'origine si la migration échoue. Toutefois, si vous restaurez les entrées de question secrète à l'aide du fichier de sauvegarde, les valeurs hachées migrées ne peuvent pas être extraites.

Pourquoi et quand exécuter cette tâche

En cryptographie, l'utilisation de sel de cryptage est une méthode qui chiffre des entrées, telles que des mots de passe, en leur ajoutant une chaîne aléatoire. Cette méthode permet de renforcer la sécurité des mots de passe. Le hachage est une méthode qui utilise un algorithme pour convertir des données en une valeur de taille fixe. Utilisez cet outil pour appliquer du sel de cryptage et hacher des réponses aux questions secrètes.

Outil de migration LDAP

L'outil de migration LDAP :

- Sauvegarde les réponses aux questions secrètes réponses au format LDIF.

- Crée un fichier LDIF (LDAP Data Interchange Format) pour migration avec les valeurs hachées et auxquelles est appliqué du sel de cryptage qui sont migrées. Ce fichier peut être exécuté ultérieurement pour effectuer la migration LDAP.
- Procède à la migration des valeurs de question secrète LDAP existantes directement dans LDAP.

Les paramètres de l'outil de migration LDAP sont notamment les suivants :

- -h: Indique le nom d'hôte de la machine LDAP.
- -p: (Facultatif) Indique le numéro de port LDAP. Le numéro de port par défaut est 389.
- -D: Indique le nom distinctif utilisateur de liaison. Par exemple, `cn=admin,dc=example,dc=com`.
- -w: Indique le mot de passe de l'utilisateur de liaison.
- -baseDN: Indique le nom distinctif de base à rechercher. Par exemple `dc=example,dc=com`.
- -attribute: Attribut LDAP utilisé pour stocker la question secrète. Par exemple, `businessCategory`.
- -newattribute: (Facultatif) Nouvelle réponse à une question secrète. Il doit s'agir d'un attribut à plusieurs valeurs si vous utilisez la fonction des questions secrètes multiples. Si cet attribut n'est pas spécifié, l'attribut de destination est le même que l'attribut d'origine indiqué dans **-attribute**.
- -ldif: (Facultatif) Ecrit les modifications dans un fichier LDIF spécifié au lieu d'exécuter la migration.
- -deleteOldEntry: (facultatif) ce paramètre fonctionne uniquement si **-attribute** et -newattribute sont spécifiés. Si ce paramètre est présent, l'ancien attribut spécifié dans **-attribute** est supprimé une fois la migration terminée. Si -ldif est également spécifié, l'attribut n'est pas supprimé immédiatement, mais le fichier LDIF contient les commandes permettant de supprimer ces entrées.
- -backup: (Facultatif) Ecrit une copie de sauvegarde de la valeur en cours de l'attribut dans le fichier LDIF spécifié.
- -Z: (Facultatif) Indique si SSL est utilisé pour la connexion à LDAP.

Procédure

1. Activation de l'utilisation de sel de cryptage et du hachage sur les valeurs de question secrète. Voir «Activation de l'utilisation de sel de cryptage et du hachage sur les valeurs de question secrète», à la page 643.
2. Arrêtez l'application d'exécution Tivoli Federated Identity Manager.
   a. Dans Integrated Solutions Console, accédez à **Applications** > **Types d'application** > **Applications d'entreprise WebSphere**
   b. Sélectionnez **ITFIMRuntime**.
   c. Cliquez sur **Arrêt**.
3. Ouvrez une invite de commande.
4. Exécutez les commandes suivantes pour l'outil de migration LDAP :

   Important : Utilisez le paramètre `-backup` pour vérifier que la copie de sauvegarde a été créée.

   ```
   java -classpath <CHEMIN_INSTALL_FIM>\tools\ldap
   com.tivoli.am.fim.ldap.MigrateUSCSecretQuestion [parameters]
   ```

   Par exemple :

```
java -classpath itfim-ldap.jar com.tivoli.am.fim.ldap.MigrateUSCSecretQuestion
-backup /home/user1/Downloads/usc/backup.ldif -h localhost
-D cn=root,dc=example,dc=com -w mercury1 -baseDn dc=example,dc=com
-ldif /home/user1/Downloads/usc/migrate.ldif -attribute description
-newAttribute businessCategory
-deleteOldEntry
```

5. Démarrez l'application d'exécution Tivoli Federated Identity Manager.
   a. Dans Integrated Solutions Console, accédez à **Applications** > **Types d'application** > **Applications d'entreprise WebSphere**
   b. Sélectionnez **ITFIMRuntime**.
   c. Cliquez sur **Démarrer**.

Résultats

Les réponses aux questions secrètes sont soumises au sel de cryptage et hachées dans le répertoire.

Utilisation de plus d'une question secrète dans User Self Care :

Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4, prend en charge la fonction des questions secrètes multiples. Vous devez configurer User Self Care pour l'utilisation de cette fonction.

Avant de commencer

Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4, prend en charge la fonction des questions secrètes multiples pour valider l'identité des utilisateurs. Dans les versions précédentes de User Self Care, une seule question secrète était utilisée. Pour comprendre les différences entre l'ancien et le nouvel exemple de la règle de mappage, voir «Modification du nombre de questions secrètes utilisées dans User Self Care», à la page 644.

Procédure

1. Editez les pages HTML suivantes :
   - `enrollment.html`
   - `secretquestion.html`
   - `profile.html`

   Tivoli Federated Identity Manager versions 6.2.2, groupe de correctifs 4, fournit des exemples de modèle de page. Fusionnez les informations requises issues de l'exemple avec vos pages HTML existantes.

Tableau 145. Pages HTML

Page	Chemin d'accès aux pages	Exemples de modèle de page
`enrollment.html`	`<REP_INSTALL_FIM>/pages/<ENV_LOCAL>/usc/enrollment/enrollment.html`	`<REP_INSTALL_FIM>/pages_template/<ENV_LOCAL>/usc/enrollment/enrollment.html`
`secretquestion.html`	`<REP_INSTALL_FIM>/pages/<ENV_LOCAL>/usc/password/secretquestion.html`	`<REP_INSTALL_FIM>/pages_template/<ENV_LOCAL>/usc/password/secretquestion.html`

Tableau 145. Pages HTML (suite)

Page	Chemin d'accès aux pages	Exemples de modèle de page
profile.html	<REP_INSTALL_FIM>/pages/<ENV_LOCAL>/usc/profile/profile.html	<REP_INSTALL_FIM>/pages_template/<ENV_LOCAL>/usc/profile/profile.html

Pour comprendre les différences entre l'ancien et le nouvel exemple de page, voir «Modification du nombre de questions secrètes utilisées dans User Self Care», à la page 644.

2. **Publiez les pages dans l'environnement d'exécution de Tivoli Federated Identity**.
3. Fusionnez la règle de mappage. Vous devez fusionner les informations fournies dans l'exemple de règle de mappage avec votre règle de mappage existante. L'exemple de règle de mappage se trouve dans <REP_INSTALL_FIM>/examples/js_mappings/usc.js.

Que faire ensuite

Vous devez reconfigurer la fédération User Self Care.

Configuration des nouvelles fédérations dans User Self Care

Après l'installation de Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4, vous devez configurer les nouvelles fédérations dans User Self Care afin de pouvoir utiliser la fonction des questions secrètes multiples.

Une fois que vous avez installé le groupe de correctifs 4, les fédérations User Self Care existantes ne prennent pas en charge la fonction des questions secrètes multiples. Toutefois, si vous créez une fédération User Self Care après avoir installé le groupe de correctifs, cette fédération prend en charge la fonction des questions secrètes multiples par défaut. Les modèles de page HTML peuvent accepter jusqu'à trois réponses aux questions secrètes.

Avant de configurer la nouvelle fédération, remplacez les pages HTML par défaut. Remplacez les pages HTML par défaut contenues dans <REP_INSTALL_FIM>/pages par les modèles de page contenus dans <REP_INSTALL_FIM>/pages_template.

Pour configurer les fédérations User Self Care, voir *IBM Tivoli Federated Identity Manager - Guide de configuration*.

L'utilisation de sel de cryptage et le hachage pour les réponses aux questions secrètes sont désactivés par défaut. Pour activer l'utilisation de sel de cryptage et le hachage, voir «Activation de l'utilisation de sel de cryptage et du hachage sur les valeurs de question secrète».

Activation de l'utilisation de sel de cryptage et du hachage sur les valeurs de question secrète

Pour améliorer la sécurité relative au stockage des valeurs de question secrète, vous pouvez activer l'utilisation de sel de cryptage et le hachage dans User Self Care.

Pourquoi et quand exécuter cette tâche

L'utilisation de sel de cryptage est une méthode qui chiffre des entrées, telles que des mots de passe, en leur ajoutant une chaîne aléatoire. Cette méthode permet de

renforcer la sécurité des mots de passe. Le hachage est une méthode qui utilise un algorithme pour convertir des données en une valeur de taille fixe. L'activation de l'utilisation de sel de cryptage et du hachage est facultative. Toutefois, il est recommandé d'activer ces fonctions pour améliorer la sécurité relative au stockage des réponses aux questions secrètes.

Procédure

1. Connectez-vous à Integrated Solutions Console.
2. Accédez à **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution** > **Propriétés personnalisées de l'environnement d'exécution**.
3. Modifiez le paramètre **USC.SecretQuestion.SaltAndHash.Enabled** et affectez-lui la valeur **true**.
4. Cliquez sur **OK**.
5. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Modification du nombre de questions secrètes utilisées dans User Self Care

User Self Care prend en charge des questions secrètes pour la validation d'utilisateur. Vous pouvez configurer User Self Care pour qu'il utilise plus d'une question secrète, ceci afin d'améliorer la sécurité de la validation des utilisateurs lorsque ces derniers ont oublié leur mot de passe.

Avant de commencer

Utilisez ces instructions pour modifier les questions secrètes présentées aux utilisateurs lors de l'inscription et de la validation et si :

- vous avez créé de nouvelles fédérations après avoir installé Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4 ;
- vous disposez de fédérations User Self Care existantes et vous avez exécuté la tâche «Utilisation de plus d'une question secrète dans User Self Care», à la page 642.

Pourquoi et quand exécuter cette tâche

Vous pouvez modifier le nombre de questions secrètes utilisées pour valider les utilisateurs. Il s'agit notamment de modifier l'interface utilisateur, les configurations de Security Token Service et la règle de mappage de User Self Care.

Remarque : Certains caractères spéciaux ne peuvent pas être utilisés comme attributs LDAP. Les administrateurs doivent établir un niveau de validation suffisant pour empêcher les utilisateurs de saisir ces caractères comme valeurs pour les zones des formulaires HTML de User Self Care.

Pour modifier le nombre de questions secrètes, vous devez modifier les fichiers suivants :

Tableau 146. Pages HTML

Page HTML	Description de la page
`enrollment.html`	Formulaire d'inscription.

Tableau 146. Pages HTML (suite)

Page HTML	Description de la page
`secretquestion.html`	Page qui s'affiche pour accepter la réponse à la question secrète et le nouveau mot de passe dans le flux de travail mot de passe oublié.
`profile.html`	Formulaire d'édition de profil.

Procédure

1. Comprendre les pages HTML liées à la fonction des questions secrètes multiples
2. Modifier les pages HTML liées à la fonction des questions secrètes multiples.
3. Comprendre la règle de mappage liée à la fonction des questions secrètes multiples.
4. Modifier la règle de mappage pour implémenter la fonction des questions secrètes multiples.
5. Appliquer les modifications à la règle de mappage pour la fonction des questions secrètes multiples.
6. Comprendre les modules STS liés à la fonction des questions secrètes multiples.
7. Modifier les modules STS liés à la fonction des questions secrètes multiples.
8. «Reconfiguration de la fédération User Self Care», à la page 655

Résultats

- Le nombre correct d'entrées de question secrète est affiché. Le nombre d'entrées est identique au nombre maximal de nouvelles questions secrètes.
- Les utilisateurs peuvent s'inscrire avec le nombre minimal et le nombre maximal de questions secrètes spécifiés.
- Les utilisateurs peuvent voir toutes les questions auxquelles ils ont répondu au cours de l'inscription lorsqu'ils effectuent une récupération de mot de passe.
- Les utilisateurs peuvent réinitialiser leur mot de passe en répondant correctement aux multiples questions secrètes pour valider leur identité.

A propos des pages HTML liées à la fonction des questions secrètes multiples de User Self Care :

Vous devez éditer des pages HTML pour configurer la fonction des questions secrètes multiples de User Self Care. Il est essentiel de comprendre comment les pages HTML fonctionnent et comment éditer ces pages en fonction de vos besoins.

Trois sections traitent de la question secrète sur les pages HTML relatives aux questions secrètes et à l'inscription. Les pages `enrollment.html` et `secretquestion.html` contiennent les trois sections. La page `profile.html` contient uniquement les deux premières sections.

- Entrée d'interface utilisateur
- Validation des zones obligatoires
- Initialisation de l'option de question secrète

Entrée d'interface utilisateur

L'entrée d'interface utilisateur comprend les éléments HTML `select` et `input`. L'élément `select` fournit la fonction de sélection de question secrète.

Cet élément est le seul endroit où se trouve la chaîne de la question secrète. La liste d'options doit être la même sur la page d'inscription et sur la page de la question secrète.

Code enrollment.html :

```
<!-- Secret Question Example Field -->
<label for="usc.form.profile.secret.question0">
    Please select a secret question and enter an answer. (required)
</label>

<br />

<select name="usc.form.profile.secret.question0"
        id="secret_question0"
        tabindex="8" >
    <option value="0">Mother's maiden name.</option>
    <option value="1">Name of town where you were born.</option>
    <option value="2">Name of first pet.</option>
</select>

<input style="background-color:#F8F8C8;"
       type="text"
       name="usc.form.profile.secret.question0.answer"
       id="secret_question_answer0"
       value=""
       size="60"
       maxlength="60"
       tabindex="9" />
<br />
```

Exemple de processus :

1. L'utilisateur soumet le formulaire d'inscription.
2. `usc.form.profile.secret.question(index)` et `usc.form.profile.secret.question(index)` sont transmis en tant que paramètres d'entrée à Tivoli Federated Identity Manager.
3. `usc.form.profile.secret.question(index).answer` contient uniquement l'index de la question secrète, et non la phrase complète.
4. Les éléments `question select` et `input` sont répétés avec un index différent en fonction du nombre maximal de questions secrètes que l'utilisateur peut entrer.

Code secretquestion.html :

```
<!-- Question 1 -->
<!-- Secret Question -->
<select name="usc.form.profile.secret.question0"
        id="secret_question0"
        disabled="disabled">
    <option value="0">Mother&#8217;s maiden name.</option>
    <option value="1">Name of town where you were born.</option>
    <option value="2">Name of first pet.</option>
</select>
<!-- Secret Question Answer Required Field -->
<input style="background-color:#F8F8C8;"
       type="text"
       name="usc.form.profile.secret.question0.answer"
       id="secret_question_answer0"
       value=""
       size="60"
       maxlength="60"
       tabindex="1" />
<input type="hidden"
       name="usc.form.profile.secret.question0.index"
       id="secret_question_hidden0"
```

```
       value="@USC_FORM_PROFILE_SECRET_QUESTION0@"
       maxlength="2" />
<br />
<!-- End question 1 -->
```

Le fichier secretquestion.html contient des sections de code similaires à enrollment.html, mais usc.form.profile.secret.question(index).index fournit des entrées masquées supplémentaires. Ces pages HTML présentent des différences. L'élément select est désactivé. L'utilisateur ne peut pas modifier la valeur fournie car usc.form.profile.secret.question(index) est désactivé. Il n'est pas transmis comme paramètre d'entrée sur le formulaire de soumission. Lorsque l'utilisateur soumet le formulaire d'inscription, les paramètres de question secrète qui sont transmis en tant que paramètres d'entrée à Tivoli Federated Identity Manager sont usc.form.profile.secret.question(index).answer et usc.form.profile.secret.question(index).index.

Code profile.html :

```
<!-- Secret Question Example Field -->
<label for="usc.form.profile.secret.question0">
    Secret question (Question and answer not displayed for your security.)
</label>
<br />
<select name="usc.form.profile.secret.question0"
        disabled="true"
        id="secret_question0">
    <option value="0">Mother's maiden name.</option>
    <option value="1">Name of town where you were born.</option>
    <option value="2">Name of first pet.</option>
</select>

<input style="background-color:#F8F8C8;"
       type="text"
       disabled="true"
       name="usc.form.profile.secret.question0.answer"
       id="secret_question_answer0"
       value=""
       size="60"
       maxlength="60" />

...
Check to edit the secret question fields:
<input id="edit_secret_question"
       type="checkbox"
       name="control3"
       onclick="enableEditing(this.checked,
               document.forms[0].secret_question0,
               document.forms[0].secret_question_answer0,
               document.forms[0].secret_question1,
               document.forms[0].secret_question_answer1,
               document.forms[0].secret_question2,
               document.forms[0].secret_question_answer2)" />
```

Le fichier profile.html contient une section de code similaire à enrollment.html. Cependant, il contient une case à cocher permettant d'activer et de désactiver les entrées de question secrète. Les éléments usc.form.profile.secret.question et usc.form.profile.secret.question.answer sont initialement désactivés. Si un utilisateur modifie la question secrète et la réponse, il doit cocher la case permettant d'activer ces entrées.

Le gestionnaire onclick de la case à cocher control3 appelle enableEditing pour activer ou désactiver les entrés de question secrète. La

fonction enableEditing prend autant de paramètres que nécessaire. Le premier paramètre est une valeur booléenne, et les autres paramètres sont les éléments qui peuvent être activés ou désactivés. Si le premier élément est true, les autres paramètres sont activés, sinon, ils sont désactivés.

Validation des zones obligatoires

La validation des zones obligatoires se produit dans la fonction javascript doSubmit(). Cette fonction est appelée lorsque l'utilisateur déclenche la soumission du formulaire. Par exemple, lorsque l'utilisateur clique sur Inscrire dans enrollment.html. Si la fonction doSubmit() aboutit, le formulaire est soumis à Tivoli Federated Identity Manager. Sinon, un message d'erreur s'affiche pour l'utilisateur et la soumission du formulaire est annulée. Les parties pertinentes sont les mêmes pour enrollment.html et secretquestions.html.

```
function doSubmit() {

    ...

    var secretQuestionAnswerTF =
            document.getElementById('secret_question_answer');

    ...

    if (!testInput(true,"Secret Question Answer (1st entry)",
            secretQuestionAnswerTF0.value.trim())){
        return true;
    }
    if (!testInput(true,"Secret Question Answer (2nd entry)",
            secretQuestionAnswerTF1.value.trim())){
        return true;
    }
    if (!testInput(false,"Secret Question Answer (3rd entry)",
            secretQuestionAnswerTF2.value.trim())){
        return true;
    }
    ...
}
```

La fonction testInput rend les zones de question secrète obligatoires. Dans enrollment.html et profile.html, les premières questions sont obligatoires et les autres sont facultatives. Le nombre de questions obligatoires doit être égal au nombre minimal de questions auxquelles un utilisateur doit répondre au cours de l'inscription. Pour secretquestion.html, ne définissez aucune des questions comme étant obligatoires afin de permettre aux utilisateurs de choisir la question à laquelle ils souhaitent répondre.

Pour la page profile.html, les entrées de question secrète sont validées et soumises uniquement lorsqu'elles sont activées.

```
function doSubmit() {

    ...

    var secretQuestionAnswerTF0 =
            document.getElementById('secret_question_answer0');
    var secretQuestionAnswerTF1 =
            document.getElementById('secret_question_answer1');
    var secretQuestionAnswerTF2 =
            document.getElementById('secret_question_answer2');

    ...

    if (document.getElementById('edit_secret_question').checked
```

```
            == true) {
        if (!testInput(true,"Secret Question Answer (1st entry)",
                secretQuestionAnswerTF0.value.trim())){
            return true;
        }
        if (!testInput(true,"Secret Question Answer (2nd entry)",
                secretQuestionAnswerTF1.value.trim())){
            return true;
        }
        if (!testInput(false,"Secret Question Answer (3rd entry)",
                secretQuestionAnswerTF2.value.trim())){
            return true;
        }
    }
}
```

Initialisation de l'option de question secrète

Lorsqu'une soumission échoue en raison d'une entrée non valide, le formulaire précédent contenant des zones préremplies est présenté à l'utilisateur pour qu'il le corrige. `setSecretQuestionSelect` affecte la valeur précédemment sélectionnée à l'élément `select`.

Les sections pertinentes sont les mêmes pour `enrollment.html` et `secretquestions.html`. `Profile.html` ne comporte pas cette section pour des raisons de sécurité.

```
function setSecretQuestionSelect() {
    var secretQuestionValue = new Array();
    secretQuestionValue[0] = "@USC_FORM_PROFILE_SECRET_QUESTION0@";
    secretQuestionValue[1] = "@USC_FORM_PROFILE_SECRET_QUESTION1@";
    secretQuestionValue[2] = "@USC_FORM_PROFILE_SECRET_QUESTION2@";

    for (var j = 0; j < 3; j++) {
        if (secretQuestionValue[j].length > 0) {
            var secretQuestion =
                    document.getElementById('secret_question' + j);
            for (i = 0; i < secretQuestion.options.length; i++) {
                if (secretQuestion.options[i].value ==
                        secretQuestionValue[j]) {
                    secretQuestion.options[i].selected = true;
                    break;
                }
            }
        }
    }
    return true;
}
```

Tivoli Federated Identity Manager remplace la valeur `@USC_FORM_PROFILE_SECRET_QUESTION(index)@` , qui est une macro, par le paramètre sortant `usc.form.profile.secret.question(index)`. Par exemple :

1. `enrollment.html` est demandé pour la première fois.
2. La valeur du paramètre sortant `usc.form.profile.secret.question(index)` est vide.
3. `@USC_FORM_PROFILE_SECRET_QUESTION(index)` est remplacé par une chaîne vide.
4. L'utilisateur soumet le formulaire d'inscription.
5. Tivoli Federated Identity Manager copie le contenu du paramètre d'entrée `usc.form.profile.secret.question(index)`. Il contient l'index de question secrète pour le paramètre sortant `usc.form.profile.secret.question(index)`.

6. Si l'inscription échoue en raison d'une entrée incorrecte, Tivoli Federated Identity Manager affiche `enrollment.html` une nouvelle fois. Il ajoute des messages d'erreur. Il remplace également `@USC_FORM_PROFILE_SECRET_QUESTION(index)@` par la valeur du paramètre sortant `usc.form.profile.secret.question(index)`.

Modification des pages HTML liées à la fonction des questions secrètes multiples dans User Self Care :

Lorsque vous modifiez le nombre maximal et le nombre minimal de questions secrètes, mettez à jour les sections d'entrée pour chacun des trois pages.

Avant de commencer

Pour connaître les conditions requises et les pages liées à la fonction des questions secrètes multiples, voir «Modification du nombre de questions secrètes utilisées dans User Self Care», à la page 644. Modifiez les pages HTML liées à la fonction des questions secrètes multiples pour éditer le nombre de questions secrètes présentées aux utilisateurs lors de l'inscription et des mises à jour de profil.

Pourquoi et quand exécuter cette tâche

Editez certaines sections sur les pages HTML pour modifier le nombre de questions secrètes.

Procédure

1. Ouvrez les pages HTML liées à l'aide d'un éditeur de texte.
2. Editez les sections pertinentes des pages HTML. Répétez le nombre de sections d'entrée le même nombre de fois que le nombre maximal de questions secrètes.
3. Mettez à jour l'index en commençant par 0.
4. Définissez les premières questions pour `enrollment.html` et `profile.html` comme des valeurs obligatoires. Le nombre de questions obligatoires doit être identique au nombre minimal de questions secrètes requises.
5. Mettez à jour l'appel vers `enableEditing` avec le nouveau nombre de questions secrètes.

Exemple

```
<input id="secret_enabled"
       type="checkbox"
       name="control3"
       onclick="enableEditing(this.checked,
               document.forms[0].secret_question0,
               document.forms[0].secret_question_answer0,
               document.forms[0].secret_question1,
               document.forms[0].secret_question_answer1,
               document.forms[0].secret_question2,
               document.forms[0].secret_question_answer2);" />
```

Que faire ensuite

Publiez les pages dans l'environnement d'exécution de Tivoli Federated Identity.

Règle de mappage pour la fonction des questions secrètes multiples :

La règle de mappage est implémentée dans la fonction des questions secrètes multiples de User Self Care pour transformer le format d'une entrée utilisateur en un autre format.

Les règles de mappage sont appliquées par le module de mappage par défaut STS (Security Token Service). Par exemple, une règle de ce type consiste à transformer une entrée utilisateur en un autre format. Dans la fonction de question secrète de User Self Care, la règle de mappage peut être utilisée pour concaténer l'index de question secrète et la réponse en une seule chaîne.

Le module STS `module de mappage par défaut` est utilisé dans les chaînes STS `Chaîne par défaut uscCreateAccount` et `Chaîne par défaut uscForgetPassword` pour réaliser l'analyse syntaxique des entrée et sortie de question secrète.

La règle de mappage par défaut se trouve dans `<REP_INSTALL_FIM>\examples\js_mappings\usc.js`. Les règles de mappage définissent le mode de stockage des questions secrètes en interne. Par défaut, la question secrète est stockée dans l'attribut LDAP `businessCategory`. Elle est stockée au format suivant : `'<secretQuestionIndex>::{SSHA2}<salt><hashedSecretQuestionAnswer>'`.

Dans le fichier de règles de mappage par défaut, deux sections concernent la question secrète :

Section sur le mappage de demande entrante

Cette section décrit le paramètre d'entrée fourni par l'utilisateur. Il extrait l'index de question secrète et la réponse à la question secrète à partir de l'entrée de formulaire et les mappe dans l'attribut interne STS pour un traitement ultérieur.

L'attribut LDAP stocke la question secrète et la réponse, et utilise la fonction `helper.setSTSInternalSecretQuestionAttr`.

```
helper.setSTSInternalSecretQuestionAttr("businessCategory");

// Set maximum secret question allowed
var MAX_SECRET_QUESTIONS = 3;
var secretQuestions = java.lang.reflect.Array.newInstance(
        java.lang.String, MAX_SECRET_QUESTIONS);
var secretQuestionsAnswer = java.lang.reflect.Array.newInstance(
        java.lang.String, MAX_SECRET_QUESTIONS);
for (var i = 0; i < MAX_SECRET_QUESTIONS; i++) {
    var secretQuestionInput = helper.getUserInputAttributeValues(
            "usc.form.profile.secret.question" + i);
    if (!supplied(secretQuestionInput)) {
        secretQuestionInput = helper.getUserInputAttributeValues(
                "usc.form.profile.secret.question" + i + ".index");
    }
    var secretQuestionAnswerInput = helper.getUserInputAttributeValues(
            "usc.form.profile.secret.question" + i +".answer);
    if (supplied(secretQuestionInput)
            && supplied(secretQuestionAnswerInput)) {
        secretQuestions[i] = secretQuestionInput[0];
        secretQuestionsAnswer[i] = secretQuestionAnswerInput[0];
    }
}
// Set to STS internal attribute
helper.setSTSInternalAttribute(
        USCCAConstants.USC_STS_INTERNAL_SECRET_QUESTIONS,
        secretQuestions);
helper.setSTSInternalAttribute(
```

```
        USCCAConstants.USC_STS_INTERNAL_SECRET_QUESTIONS_ANSWER,
        secretQuestionsAnswer);
// Set input to normalize and sanitize STS
var normalizeSanitizeInput =
        java.lang.reflect.Array.newInstance(java.lang.String, 2);
normalizeSanitizeInput[0] =
        USCCAConstants.USC_STS_INTERNAL_SECRET_QUESTIONS;
normalizeSanitizeInput[1] =
        USCCAConstants.USC_STS_INTERNAL_SECRET_QUESTIONS_ANSWER;
helper.setSTSInternalAttribute(
        USCCAConstants.USC_STS_INTERNAL_NORMALIZE_AND_SANITIZE_INPUT,
        normalizeSanitizeInput);
// Set input to salt and hash STS
helper.setSTSInternalAttribute(
        USCCAConstants.USC_STS_INTERNAL_SALT_AND_HASH_INPUT,
        USCCAConstants.USC_STS_INTERNAL_SECRET_QUESTIONS_ANSWER);
```

Section sur le mappage de demande sortante

Cette section décrit le paramètre sortant qui remplace les macros sur les pages HTML avant d'envoyer les pages à l'utilisateur. Il extrait la question secrète stockée à partir de l'attribut `USC_FORM_SECRET_QUESTION` de la sortie STS. Il définit ensuite l'attribut `USC_FORM_SECRET_QUESTION(Index)` de la sortie STS.

```
 //
Get the secret question from the registry
   var secretQuestionRA = helper.getSTSOutputAttributeValues(
           USCCAConstants.USC_FORM_SECRET_QUESTION);
   if (secretQuestionRA != null) {
       for (var i = 0; i < secretQuestionRA.length; i++) {
           if (secretQuestionRA[i] != null) {
               helper.setSTSOutputAttribute(
                       "usc.form.profile.secret.question" + i,
                       secretQuestionRA[i]);
           }
       }
   }
```

Modification de la règle de mappage pour la fonction des questions secrètes multiples de User Self Care :

Vous devez modifier les règles de mappage si vous changez le nombre maximal de questions secrètes.

Avant de commencer

Pour comprendre comment la règle de mappage est implémentée dans la fonction des questions multiples de User Self Care, voir «Règle de mappage pour la fonction des questions secrètes multiples», à la page 651.

Procédure

1. Sauvegardez le fichier de règles de mappage.
2. Ouvrez le fichier.
3. Modifiez le nombre maximal de questions secrètes. Par exemple :
   ```
   var MAX_SECRET_QUESTIONS = 3;
   ```
4. Enregistrez le fichier modifié.

Que faire ensuite

Appliquez les modifications à la règle de mappage.

Application des modifications apportées à la règle de mappage pour la fonction des questions secrètes multiples :

Appliquer la règle de mappage modifiée au module de mappage par défaut des trois chaînes STS pour utiliser la fonction des questions secrètes multiples.

Avant de commencer

Utilisez la règle de mappage que vous avez modifiée dans «Modification de la règle de mappage pour la fonction des questions secrètes multiples de User Self Care», à la page 652.

Pourquoi et quand exécuter cette tâche

Pour que la règle de mappage implémente les modifications, elle doit être appliquée aux chaînes suivantes :

- Chaîne par défaut **uscCreateAccount**
- Chaîne par défaut **uscForgottenPassword** (les deux instances de cette chaîne)
- Chaîne par défaut **uscProfileManagement** (les deux instances de cette chaîne)

Procédure

1. Connectez-vous à Integrated Solutions Console.
2. Accédez à **Tivoli Federated Identity Manager** > **Gestion des noeuds d'exécution**.
3. Cliquez sur **Propriétés personnalisées de l'environnement d'exécution**.
4. Affectez la valeur `true` à la propriété **STS.showUSCChains**.
5. Cliquez sur **OK**.
6. Déconnectez-vous d'Integrated Solutions Console.
7. Connectez-vous à Integrated Solutions Console.
8. Accédez à **Tivoli Federated Identity Manager** > **Configuration du service d'accréditation** > **Chaînes du service d'accréditation**.
9. Pour chacune des chaînes, procédez comme suit :
   a. Cochez la case de la chaîne correspondante.
   b. Cliquez sur **Propriétés**.
   c. Cliquez sur **Modifier la règle**.
   d. Cliquez sur **Parcourir** pour sélectionner la règle de mappage modifiée.
   e. Cliquez sur **Importer un fichier**.
   f. Cliquez sur **OK**.
   g. Répétez les étapes pour le second `module de mappage par défaut` de la chaîne.
   h. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Paramètres du fichier de réponses pour la fonction des questions secrètes multiples de User Self Care :

Utilisez les paramètres du fichier de réponses pour la fonction des questions secrètes multiples de User Self Care pour configurer le fichier de réponses.

Les trois paramètres configurables dans le fichier de réponses en rapport avec plusieurs questions secrètes sont les suivants :

SecretQuestionRequiredForValidationNumber
Spécifie le nombre de questions secrètes auxquelles un utilisateur doit répondre correctement pour valider son identité. Au cours de la récupération de mot de passe, les utilisateurs doivent fournir des réponses correctes aux questions secrètes. Le nombre de questions auxquelles ils doivent répondre correctement dépend de ce paramètre.

Remarque : Ce paramètre figure également dans le fichier de réponses, avec la valeur 1 définie par défaut.

SecretQuestionMaximumNumber
Indique le nombre maximal de questions secrètes auxquelles un utilisateur peut répondre pendant l'inscription. Il définit le nombre de questions secrètes auxquelles les utilisateurs peuvent répondre lorsqu'ils ont oublié leur mot de passe ou lors de la mise à jour du profil.

Remarque : Ce paramètre figure également dans le fichier de réponses, avec la valeur 3 définie par défaut.

SecretQuestionMinimumNumber
Indique le nombre minimal de questions secrètes auxquelles un utilisateur doit répondre pendant l'inscription. Il définit le nombre de questions secrètes auxquelles les utilisateurs doivent répondre lorsqu'ils ont oublié leur mot de passe ou lors de la mise à jour du profil.

Remarque : Ce paramètre figure également dans le fichier de réponses, avec la valeur 2 définie par défaut.

Modification des configurations de module STS pour la fonction des questions secrètes multiples :

Vous devez modifier des modules STS spécifiques afin d'utiliser la fonction des questions secrètes multiples de User Self Care.

Pourquoi et quand exécuter cette tâche

Le module STS de magasin de questions secrètes USC par défaut se trouve dans la chaîne `uscAccountCreate` et dans la chaîne `uscProfileManagement`.

Le module STS de question secrète USC par défaut se trouve dans la chaîne `uscForgottenPassword`.

Modifiez le module STS de question secrète USC par défaut et le module STS de magasin de questions secrètes USC par défaut dans les chaînes STS suivantes :

- Chaîne par défaut **uscCreateAccount**
- Chaîne par défaut **uscForgottenPassword** (les deux instances de cette chaîne)
- Chaîne par défaut **uscProfileManagement** (les deux instances de cette chaîne)

Procédure

1. Connectez-vous à Integrated Solutions Console.
2. Accédez à **Tivoli Federated Identity Manager** > **Gestion des noeuds d'exécution**.
3. Cliquez sur **Propriétés personnalisées de l'environnement d'exécution**.
4. Affectez la valeur `true` à la propriété **STS.showUSCChains**.
5. Cliquez sur **OK**.

6. Déconnectez-vous d'Integrated Solutions Console.
7. Connectez-vous à Integrated Solutions Console.
8. Accédez à **Tivoli Federated Identity Manager** > **Configuration du service d'accréditation** > **Chaînes du service d'accréditation**.
9. Pour chacune des chaînes, procédez comme suit :
   a. Cochez la case de la chaîne correspondante.
      - Chaîne par défaut **uscCreateAccount**
      - Chaîne par défaut **uscForgottenPassword** (les deux instances de cette chaîne)
      - Chaîne par défaut **uscProfileManagement** (les deux instances de cette chaîne)
   b. Cliquez sur **Propriétés**.
   c. Sous **Modules de la chaîne du service d'accréditation**, sélectionnez le module et le mode correspondants. Vous devez modifier le module STS de question secrète USC par défaut et le module STS de magasin de questions secrètes USC par défaut.
      - La chaîne par défaut **uscCreateAccount** contient le module STS de magasin de questions secrètes USC par défaut.
      - La chaîne par défaut **uscForgottenPassword** (les deux instances de cette chaîne) contient le module STS de question secrète USC par défaut.
      - La chaîne par défaut **uscProfileManagement** (les deux instances de cette chaîne) contient le module STS de magasin de questions secrètes USC par défaut.
   d. Cliquez sur **Propriétés**.
   e. Modifiez les valeurs de paramètre correspondantes.

      Module STS de question secrète USC par défaut

      SecretQuestionRequiredForValidationNumber - Modifiez la valeur de ce paramètre afin de spécifier le nombre de questions secrètes auxquelles un utilisateur doit répondre correctement pour que son identité soit validée.

      Module STS de magasin de questions secrètes USC par défaut

      - **SecretQuestionMaximumNumber** - Modifier la valeur de ce paramètre afin de spécifier le nombre maximal de questions secrètes auxquelles un utilisateur peut répondre pendant l'inscription.
      - **SecretQuestionMinimumNumber** - Modifiez la valeur de ce paramètre pour spécifier le nombre minimal de questions secrètes requises auxquelles un utilisateur doit répondre pendant l'inscription.
   f. Cliquez sur **OK**.
   g. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**

Reconfiguration de la fédération User Self Care :

Reconfigurez la fédération User Self Care en vue de l'utilisation d'un fichier de réponses modifié.

Procédure

1. Exportez la configuration actuelle de la fédération User Self Care dans un fichier de réponses à l'aide de la commande suivante dans wsadmin :

   Remarque : Entrez la syntaxe suivante sur une seule ligne :

   ```
   $AdminTask manageItfimUserSelfCare {-operation createResponseFile
   -fileId <chemin_fichier> -fimDomainName <nom_domaine> -federationName <nom_fédération>}
   ```

2. Modifiez les paramètres suivants :
   - **SecretQuestionMinimumNumber**
   - **SecretQuestionMaximumNumber**
   - **SecretQuestionRequiredForValidationNumber**

   Pour plus d'informations, voir «Paramètres du fichier de réponses pour la fonction des questions secrètes multiples de User Self Care», à la page 653.
3. Modifiez le paramètre **AttributeMappingFilename** pour qu'il pointe vers la règle de mappage modifiée que vous avez éditée dans «Modification de la règle de mappage pour la fonction des questions secrètes multiples de User Self Care», à la page 652.
4. Sauvegardez le fichier.
5. Annulez la configuration de la fédération User Self Care. Voir Annulation de la configuration de User Self Care.
6. Configurez la fédération User Self Care à l'aide du fichier de réponses modifié. Voir Utilisation d'un fichier de réponses pour configurer User Self Care.

Définition d'un attribut personnalisé

Définissez votre propre attribut personnalisé dans User Self Care de sorte qu'il soit collecté par User Self Care lors de l'inscription.

Les informations utilisateur collectées durant l'inscription sont stockées dans les annuaires d'entreprise pris en charge. Elles sont ensuite mappées aux attributs LDAP de l'entrée d'utilisateur dans un annuaire d'entreprise configuré.

L'inscription d'utilisateur par défaut fournit un nombre limité de zones d'informations qui sont mappées à des attributs LDAP. Vous devez ajouter des zones qui ne sont pas incluses. Vous pouvez utiliser la gestion des profils utilisateur pour ajouter des zones pouvant être mappées à un attribut LDAP existant dans un répertoire d'entreprise.

Par défaut, les zones suivantes sont fournies :

- ID utilisateur
- Mot de passe d'adresse électronique
- Numéro de téléphone portable
- Question secrète

Vous devez effectuer quelques étapes si vous souhaitez ajouter une zone telle que **Company**. La valeur de la nouvelle zone est sauvegardée dans l'attribut LDAP mappé dans l'annuaire d'entreprise configuré.

Pour plus d'informations sur l'utilisation d'un attribut LDAP qui n'est pas défini par défaut, voir «Création d'un attribut pour une nouvelle zone de personnalisation dans User Self Care», à la page 660.

Procédez comme suit pour définir des attributs personnalisés dans User Self Care :

1. Modifiez le fichier HTML pour que la nouvelle zone s'affiche dans le formulaire des informations utilisateur.
2. Modifiez le script Java pour que la nouvelle zone soit mappée aux attributs LDAP.
3. Exécutez les commandes wsadmin pour que vos modifications apparaissent dans IBM Tivoli Federated Identity Manager.

Une fois ces étapes effectuées, l'utilisateur peut s'inscrire.

Modification du fichier HTML pour définir un attribut personnalisé

Modifiez `enrollment.html` et `profile.html` pour définir un attribut personnalisé dans User Self Care.

Pourquoi et quand exécuter cette tâche

Vous pouvez modifier les fichiers HTML suivants :

Tableau 147. Fichiers HTML

Page HTML	Description de la page
`enrollment.html`	Formulaire d'inscription.
`profile.html`	Formulaire qui met à jour les informations de profil de l'utilisateur.

Procédure

1. Ouvrez chacun des fichiers HTML à l'aide d'un éditeur de texte.

   Remarque : Si le fichier HTML n'est pas en anglais, utilisez un éditeur de texte prenant en charge le codage UTF-8. Dans le cas contraire, certaines lettres peuvent apparaître de façon incorrecte et la sauvegarde du fichier peut générer un contenu illisible.
2. Ajoutez les lignes suivantes entre les balises `<form...> ... </form>`.

   ```
   <label for="usc.form.profile.company"> Company </label>
   <input id="company" type="text" tabindex="8" maxlength="60" size="60"
   value="" name="usc.form.profile.company" style="
   background-color: rgb(255, 255, 255);" />
   ```
3. Remplacez le texte `Company` par un nom de zone de personnalisation. Vous pouvez personnaliser n'importe quel attribut mais vous devez prendre soin d'utiliser les mêmes valeurs d'identification.
4. Enregistrez les fichiers modifiés dans le même répertoire que les fichiers d'origine.
5. Connectez-vous à la console **Integrated Solutions Console**.
6. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution**.
7. Cliquez sur **Publier**.
8. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Résultats

La nouvelle zone s'affiche sur la page d'inscription de compte.

Que faire ensuite

Modifiez le fichier de mappage JavaScript.

Modification du fichier de mappage JavaScript

Modifiez le fichier de mappage JavaScript pour que la nouvelle zone soit mappée aux attributs LDAP.

Pourquoi et quand exécuter cette tâche

Le fichier de mappage JavaScript par défaut se trouve dans `<FIMInstallationPath>/examples/js_mappings/usc.js`. Il définit le comportement de mappage. Ajoutez les nouvelles règles de mappage pour la nouvelle zone personnalisée dans le fichier JavaScript. Si vous utilisez un autre fichier JavaScript pour les règles de mappage, prenez soin d'accéder au fichier approprié.

Procédure

1. Sauvegardez le fichier de règles de mappage JavaScript.
2. Ouvrez le fichier de règles de mappage JavaScript dans un éditeur de texte.
3. Ajoutez les lignes suivantes et remplacez la valeur de `Company` par le nom de zone que vous souhaitez ajouter.

```
////////////////////////////////
// INCOMING REQUEST MAPPING
////////////////////////////////
var company = helper.getUserInputAttributeValues("usc.form.profile.company");
if (supplied(company)) {
    // map 'company' to 'company' LDAP attribute
    // change "company" to "organizationName" when using Tivoli Directory Server
    // or as mentioned you can use some other LDAP attribute
    helper.setSTSInternalRegistryInputAttribute("company", company);
}

////////////////////////////////
// OUTGOING RESPONSE MAPPING
////////////////////////////////
// Get the 'company' from the registry.
 // change "company" to "organizationName" when TDS
      // or as mentioned you can use some other LDAP attribute
var companyRA = helper.getSTSInternalRegistryOutputAttributeValues("company");
if (supplied(companyRA)) {
    // Stick it in the output attribute
    helper.setSTSOutputAttribute("usc.form.profile.company", companyRA);
}
```

4. Sauvegardez les modifications.
5. Connectez-vous à la console **Integrated Solutions Console**.
6. Sélectionnez **Tivoli Federated Identity Manager** > **Configurer le service d'accréditation** > **Chaînes du service d'accréditation**.
7. Ouvrez les pages **Propriétés** pour les fichiers suivants :
   - `Chaîne USC pour uscCreateAccount`
   - `Chaîne USC pour uscProfileManagement` (deux instances)

   Remarque : Selon la chaîne que vous êtes en train de modifier, vous devrez modifier plusieurs chaînes pour la règle de mappage à implémenter.
8. Cochez les cases **Chaîne USC pour uscCreateAccount** et **Chaîne USC pour uscProfileManagement**.
9. Exécutez les étapes suivantes pour les trois éléments.

- `Chaîne USC pour uscCreateAccount`
- `Chaîne USC pour uscProfileManagement` (deux instances)

a. Cliquez sur **Propriétés**.
b. Sous **Modules de la chaîne du service d'accréditation**, sélectionnez **Module de mappage par défaut**.
c. Cliquez sur **Propriétés**.
d. Cliquez sur **Modifier la règle**.
e. Cliquez sur **Parcourir** pour sélectionner la règle de mappage modifiée.
f. Cliquez sur **Importer le fichier**.
g. Cliquez sur **OK**.
h. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.
i. Répétez les étapes pour les éléments suivants :
   - Les mappages de chaîne auxquels la règle est associée. Certaines règles peuvent affecter plusieurs mappages de chaîne.
   - Les chaînes portant le même nom.
   - Les modules de mappage par défaut présents dans une chaîne.

Que faire ensuite

Exécutez les commandes wsadmin.

Exécution de commandes wsadmin pour l'implémentation de l'attribut personnalisé

Exécutez les commandes **wsadmin** pour que l'attribut personnalisé apparaisse dans Tivoli Federated Identity Manager User Self Care.

Procédure

1. Connectez-vous à la console wsadmin à l'aide des commandes suivantes :

   Windows : **<WASInstallationPath>\WebSphere\AppServer\profiles\<profileName> \bin\wsadmin.bat\-username username \-password password**

   Linux : **/<WASInstallationPath>/WebSphere/AppServer/profiles/<profileName> /bin/wsadmin.sh\-username username \-password password**

2. Dans la console wsadmin, exécutez les commandes **$AdminTask** suivantes.
   a. **$AdminTask addIdMgrPropertyToEntityTypes {-name company -dataType string -entityTypeNames PersonAccount}**

      Cette commande prend en charge le nouvel attribut personnalisé pour le composant Référentiels fédérés ou pour Virtual Member Manager de WebSphere Application Server. Par défaut, le composant Virtual Member Manager ne prend en charge que quelques attributs de base. Pour utiliser un nouvel attribut, vous devez agrandir le schéma d'entité interne dans le composant Virtual Member Manager. Cette commande ajoute la prise en charge de l'attribut **company** LDAP dans Virtual Member Manager. Vous pouvez modifier **company** pour n'importe quel attribut LDAP.
   b. **$AdminTask addIdMgrPropertyToEntityTypes {-name company -dataType string -entityTypeNames PersonAccount -isMultiValued true}**

      Certains attributs admettent plusieurs valeurs. Vous devez utiliser des attributs à valeurs multiples. Si les attributs doivent avoir plusieurs valeurs, indiquez le paramètre **isMultiValued** avec la valeur `true`.
3. Arrêtez WebSphere Application Server.

4. Redémarrez WebSphere Application Server. Lorsque le serveur est redémarré, les paramètres du composant Virtual Member Manager sont appliqués.

Résultats

Lorsque vous cliquez sur le lien dans le courrier électronique, l'attribut utilisateur est créé dans l'annuaire d'entreprise configuré avec le nouvel attribut.

Que faire ensuite

Après avoir redémarré WebSphere Application Server, accédez à la page d'inscription et vérifiez que la nouvelle zone a été ajoutée. Après que vous avez cliqué sur Inscrire, un courrier électronique est envoyé à l'adresse électronique que vous avez indiquée.

Création d'un attribut pour une nouvelle zone de personnalisation dans User Self Care

Créez un attribut pour une zone de personnalisation dans User Self Care, qui est mappée vers un attribut LDAP que vous souhaitez utiliser.

Avant de commencer

Assurez-vous que Tivoli Federated Identity Manager et WebSphere Application Server sont configurés pour Virtual Member Manager. Pour plus d'informations, voir Tivoli Federated Identity Manager Configuration Guide.

Pourquoi et quand exécuter cette tâche

Exemple 1

Créez un attribut pour une zone de personnalisation, telle que `badge number`, qui est mappée à un attribut, tel que `badgeNo` dans LDAP.

Exemple 2

Vous souhaitez ajouter l'attribut IBM Tivoli Directory Server `ibm-ismanager`. Il ne peut pas être ajouté à une entité LDAP par défaut. Vous devez tout d'abord ajouter l'attribut `ibm-itdPerson` comme l'une des valeurs `objectClasses` de l'entité LDAP. Vous pouvez ensuite ajouter `ibm-ismanager` à une entité LDAP avec une valeur `true` ou `false`.

Aucune modification n'est requise pour les scénarios dans lesquels vous souhaitez ajouter une zone de personnalisation possédant déjà un attribut LDAP disponible. Par exemple, vous pouvez ajouter la zone de personnalisation `Given Name` et la mapper à l'attribut `givenName` dans IBM Tivoli Directory Server.

Procédure

1. Connectez-vous à la console **Integrated Solutions Console**.
2. Sélectionnez **Sécurité** > **Sécurité globale**.
3. Sous **Référentiel de comptes utilisateur**, cliquez sur **Configurer**.
4. Sélectionnez **Référentiels fédérés**.
5. Cliquez sur **Gestion des référentiels**. Une liste de référentiels s'affiche. Procédez comme suit pour chaque référentiel :
   a. Cliquez sur le nom du référentiel. Les détails relatifs au référentiel sont affichés.
   b. Sous **Propriétés supplémentaires**, cliquez sur **Types d'entités LDAP**.

c. Cliquez sur **PersonAccount**.
d. La page **PersonAccount** comporte une zone **Classes d'objets**. Une valeur existante, telle que `user`, existe si le type de référentiel est Active Directory ou IBM Tivoli Directory Server.
e. Dans la zone **Classes d'objets**, ajoutez le nouvel attribut `objectClass` qui définit les attributs que vous souhaitez utiliser. Par exemple, si vous souhaitez utiliser l'attribut `ibm-ismanager`, la classe d'objets qui le définit est `ibm-itdPerson`. Ajoutez l'attribut `ibm-ismanager` dans la zone avec un point-virgule. Par exemple, `netOrgPerson;ibm-itdPerson`.
f. Cliquez sur **OK**.
g. Cliquez sur **Enregistrer les modifications dans le fichier de configuration principale**.

6. Facultatif : Si vous avez configuré et utilisez un adaptateur Tivoli Access Manager, procédez comme suit :
   a. Ouvrez le fichier `tamVMMAdapter.properties`.

      ```
      <Installation_WAS>\profiles\<tfimprofile>\config\itfim\
        tamVMMAdapter.properties
      ```

      Remarque : Le nom de fichier sera peut-être différent. Le nom affiché dans cet exemple est celui qui est défini lors de la configuration de l'adaptateur Tivoli Access Manager.
   b. Ajoutez les lignes suivantes.

      ```
      ldap.user-objectclass=Person;ePerson;inetOrgPerson,organizationalPerson;
      customObjectClass ldap.user-self-care-objectclass=customObjectClass
      ```

7. Arrêtez WebSphere Application Server.
8. Démarrez WebSphere Application Server.

Résultats

Le gestionnaire de membre virtuel crée l'attribut.

Stockage des informations de session User Self Care

Permet à certains flux User Self Care de conserver divers attributs utilisateur dans la session d'un utilisateur. Ces attributs peuvent ensuite être extraits par d'autres modules.

Fournit une option dans le fichier de réponses de User Self Care qui indique quels sont les flux qui peuvent conserver des informations. Vous pouvez configurer les flux qui envoient généralement un e-mail pour stocker des informations dans la session utilisateur.

Vous pouvez configurer les flux suivants pour stocker des informations User Self Care dans la session utilisateur :

- Inscription d'utilisateur
- Mot de passe oublié
- ID utilisateur oublié

Pour stocker des informations dans une session, configurez le paramètre `FlowsWithSessionStorageAndNoEmailDelivery` dans le fichier de réponses.

Ce paramètre est un paramètre de chaîne à plusieurs valeurs qui, par défaut, n'est associé à aucun flux. Il recherche les valeurs suivantes dans le fichier de réponses :

```
<void method="put">
   <string>FlowsWithSessionStorageAndNoEmailDelivery</string>
   <object class="java.util.ArrayList">
      <void method="add">
        <string>USC_ENROLLMENT</string>
      </void>
      <void method="add">
        <string>USC_FORGOT_PASSWORD</string>
      </void>
      <void method="add">
        <string>USC_FORGOT_ID</string>
      </void>
   </object>
</void>
```

Où :

- L'inscription d'utilisateur est `USC_ENROLLMENT`
- Le mot de passe oublié est `USC_FORGOT_PASSWORD`
- L'ID utilisateur oublié est `USC_FORGOT_ID`

Les informations stockées dans la session sont extraites de la chaîne STS (Security Token Service) de User Self Care et placées dans une mappe. La mappe

- se compose de clés de chaîne et de valeurs de type String[].
- Elle est sérialisée, convertie en byte[] et codée en Base64.
- Elle est ensuite stockée dans la session de l'utilisateur.
- Elle peut être extraite dans une règle de mappage à l'aide de la fonction `IDMappingExtUtils.getSPSSessionData()` avec la clé `usc_attributes_session_key`.

La mappe contient divers attributs liés à l'utilisateur et au flux. Etant donné que les attributs admettent plusieurs valeurs, ils sont stockés sous forme de tableau d'objets de type chaîne dans la mappe. Chaque flux conserve un ensemble différent d'attributs.

- Le flux d'inscription d'utilisateur

Tableau 148. Une liste des attributs stockés lors d'un flux d'inscription d'utilisateur

Clé	Description
`usc.flow.id`	Flux exécuté.
`usc.form.userid`	Utilisateur effectuant l'inscription.
`usc.confirmation.code`	Code de confirmation du flux d'inscription.
`usc.validation.url`	Adresse URL de validation du flux d'inscription.

Remarque : Les attributs STSUU de type `usc.user.input.type` se trouvent également dans la mappe et leur clé correspond à leur nom d'attribut STSUU.

- Le flux de mot de passe oublié

Tableau 149. Une liste des attributs stockés lors d'un flux de mot de passe oublié

Clé	Description
`usc.flow.id`	Flux exécuté.
`usc.user.id`	Utilisateur effectuant l'inscription.
`usc.confirmation.code`	Code de confirmation du flux d'inscription.
`usc.validation.url`	Adresse URL de validation du flux d'inscription.

Remarque : Les attributs STSUU de type `usc.sts.internal.registry.output.type` se trouvent également dans la mappe et leur clé correspond à leur nom d'attribut STSUU.

- Le flux d'ID utilisateur oublié

Tableau 150. Une liste des attributs stockés lors d'un flux d'ID utilisateur oublié

Clé	Description
usc.user.id	Liste des ID utilisateur associés au compte oublié.
usc.flow.id	Flux exécuté.

Remarque : Les attributs STSUU de type `usc.sts.output.type` se trouvent également dans la mappe et leur clé correspond à leur nom d'attribut STSUU.

L'exemple JavaScript suivant explique comment une règle de mappage peut extraire les données User Self Care stockées depuis la session de l'utilisateur. Pour ce faire, la mappe sérialisée est extraite de la session à l'aide de la méthode `getSPSSessionData('usc_attribute_session_key')`. Cet objet est désérialisé dans une mappe et les attributs individuels sont accessibles depuis la mappe à l'aide de la clé appropriée.

Pour pouvoir accéder à `SPSSessionData`, vous devez importer le package `com.tivoli.am.fim.trustserver.sts.utilities`. Pour procéder à la désérialisation, vous pouvez importer le package `com.ibm.ws.util`. La règle de mappage suivante en est un exemple.

```
//required import statements
importPackage(Packages.com.tivoli.am.fim.trustserver.sts.utilities);
importPackage(Packages.com.ibm.ws.util);

//
// Returns a String->String[] attribute map that was set by USC using the
//     FlowsWithSessionStorageAndNoEmailDelivery parameter.
//
function getUSCAttributeMap() {
    var dmapKey0 = "usc_attributes_session_key";
    var attrMap = null;
    var serializedAttributes = IDMappingExtUtils.getSPSSessionData(dmapKey0);
    if (serializedAttributes != null) {
        IDMappingExtUtils.traceString("Serialized attributes: " + serializedAttributes);
        // deserialize then add to AttributeList with canned "type"
        var bais = new java.io.ByteArrayInputStream((Base64.decode(serializedAttributes)));
        var ois = new java.io.ObjectInputStream(bais);
        attrMap = ois.readObject();
    } else {
        IDMappingExtUtils.traceString("No session attributes are persisted.");
    }
    return attrMap;
}

//
// Pull the attributes from the session stored by USC
//
var flowId = null;
var userId = null;
var confirmationCode = null;
var validationUrl = null;

// get the attribute map
var uscAttributeMap = getUSCAttributeMap();
if (uscAttributeMap != null) {
    // All values are String[] and you must use index operator.
    flowId = uscAttributeMap.get("usc.flow.id")[0];
    userId = uscAttributeMap.get("usc.user.id")[0];
    confirmationCode = uscAttributeMap.get("usc.confirmation.code")[0];
    validationUrl = uscAttributeMap.get("usc.validation.url")[0];
    IDMappingExtUtils.traceString("Attributes retrieved-> flowId: " + flowId + " userId: " + userId +
        " confirmation code: " + confirmationCode + " validationUrl: " + validationUrl);
}
```

Personnalisation des pages HTML de User Self Care

Personnaliser l'interface User Self Care en ajoutant des fichiers CSS, des images d'arrière-plan et en utilisant les macros fournies par User Self Care.

Avant de commencer

Les administrateurs qui configurent User Self Care doivent connaître les produits et les concepts suivants :

- WebSphere® Application Server, y compris l'interface d'administration `wsadmin`.
- Modules Secure Token Service (STS) et chaînes d'accréditation Tivoli Federated Identity Manager.
- Protocole LDAP Tivoli Directory Server ou autre protocole LDAP pris en charge.
- Connaissance de CSS et HTML

Pourquoi et quand exécuter cette tâche

Vous avez besoin d'un serveur Web, par exemple, IBM HTTP Server, pour héberger les fichiers externes, tels que les fichiers CSS et les fichiers image.

Vous pouvez personnaliser les pages HTML avec les macros et les feuilles de style en cascade fournies par User Self Care. Pour plus de détails, consultez le site :

- «Macros User Self Care»
- «A propos des feuilles de style en cascade de User Self Care», à la page 669

Procédure

1. Connectez-vous à la console Integrated Solution Console.
2. Effectuez l'une des actions suivantes :
   - Utilisez les macros pour formater User Self Care. Pour plus d'informations, voir «Formatage des pages HTML de User Self Care à l'aide de macros», à la page 671.
   - Utilisez les feuilles de style en cascade pour formater User Self Care. Pour plus d'informations, voir «Formatage des pages HTML de User Self Care à l'aide de feuilles de style en cascade», à la page 672.
3. Testez les modifications apportées aux fichiers HTML.

Macros User Self Care

Les macros User Self Care permettent d'ajouter du code ou des valeurs aux pages HTML User Self Care lors de l'exécution.

Les macros sont remplacées par leurs valeurs lors de l'exécution lorsque l'utilisateur accède à User Self Care. Les macros ne sont pas toutes renseignées en même temps. La valeur des macros varie en fonction de l'exploitation et de l'état en cours de cette exploitation.

Les informations suivantes s'affichent :

- Les macros disponibles dans la configuration par défaut de User Self Care.
- L'exploitation et l'état de l'exploitation de ces macros.
- La description de la macro avec des exemples de valeur pour User Self Care. WebSphere est le serveur point de contact.

Le modèle comprend les macros de remplacement suivantes :

@ACTION@

Cette macro est remplacée par l'URL à laquelle les données de formulaire sont envoyées pour soumission. Il s'agit également de l'URL qui est demandée initialement. Par exemple, dans le formulaire d'inscription, l'action de formulaire se présente comme suit :

```
<form action=" (ACTION)">
```

Lors du chargement de la page, cette action est remplacée par

```
<form action="https://company.com:9443/sps/USCFederation/
usc/self/account/create">
```

Cette macro est applicable à tous les flux de travaux.

@VALIDATION_URL@

Cette macro est remplacée par l'URL de validation. Chaque fois que le système oblige l'utilisateur à envoyer une vérification par courrier électronique, VALIDATION_URL est remplacé par l'URL qui doit être envoyée dans le courrier électronique. Au moment de l'exécution, le système envoie un courrier électronique contenant l'URL ajoutée avec un paramètre de code de confirmation unique.

Valeur pour l'inscription d'utilisateur :

Remarque : Entrez la syntaxe suivante sur une seule ligne :

```
https://<nom_hôte>:<port>/sps/<fédération_USC>/usc/self/
account/create/validate
```

Valeur pour l'oubli de l'ID utilisateur :

Remarque : Entrez la syntaxe suivante sur une seule ligne :

```
https://<nom_hôte>:<port>/sps/<fédération_USC>/usc/self/account/recover
/password/validate
```

@USC_SEARCH_USERID_URI@

Sur la page Inscription, cette macro représente l'URL à laquelle la demande est envoyée lorsque l'utilisateur clique sur **ID utilisateur disponible ?**.

Valeur pour l'inscription d'utilisateur :

```
https://<nom_serveur>:<port>/sps/fédération>/usc/global/userid/search
```

@DETAIL@

Cette macro contient le message, le cas échéant, dans la réponse.

Exemple

Si l'utilisateur tente d'accéder à l'URL de validation sans données de validation dans l'URL, la valeur de DETAIL se présente comme suit :

```
FBTUSC0>>E
```

Les données de validation d'inscription doivent être fournies.Cette macro est applicable à tous les flux de travaux.

@EXCEPTION_STACK@

Cette macro est remplacée par la source HTML requise qui prend en charge la démonstration `Captcha` si elle est configurée.

Exemple

Cette macro est remplacée par :

```
<label for="demo_captcha">
  Veuillez entrer le ou les mots de vérification affichés ci-dessous (obligatoire)
  </label>
  <br />
  <img src="http://myserver/public/captcha_test/hello.jpg" border="0" />
  <br />
  <input type="hidden"
  name="usc.demo.captcha.challenge.field"
```

```
id="usc.demo.captcha.challenge.field"
value="http://myserver/public/captcha_test/hello.jpg" />
<input style="background-color:#F8F8C8;"
type="text"
name="usc.demo.captcha.response.field"
id="usc.demo.captcha.response.field" />
```

Cette macro est applicable à tous les flux de travaux.

@USC_STS_CAPTCHA_HTML_STRING@

Cette macro est remplacée par la source HTML requise qui prend en charge la démonstration Captcha si elle est configurée.

Exemple

Cette macro est remplacée par :

```
<label for="demo_captcha">
  Veuillez entrer le ou les mots de vérification affichés ci-dessous (obligatoire)
  </label>
  <br />
  <img src="http://myserver/public/captcha_test/hello.jpg" border="0" />
  <br />
  <input type="hidden"
  name="usc.demo.captcha.challenge.field"
  id="usc.demo.captcha.challenge.field"
  value="http://myserver/public/captcha_test/hello.jpg" />
  <input style="background-color:#F8F8C8;"
  type="text"
  name="usc.demo.captcha.response.field"
  id="usc.demo.captcha.response.field" />
```

Cette macro est applicable dans le flux de travail d'inscription d'utilisateur.

@USC_FORM_USERID@

Cette macro est remplacée par l'ID de l'utilisateur avec une session valide ou l'ID utilisateur fourni dans la demande précédente.

Exemple

Dans le flux de travaux de modification de mot de passe, USC_FORM_USERID est remplacé par l'ID utilisateur authentifié après l'authentification de l'utilisateur.Cette macro est applicable à tous les flux de travaux.

@USC_USERAGENT_IPADDR@

Cette macro contient l'adresse IP à partir de laquelle la demande précédente est reçue.

Cette macro est applicable à tous les flux de travaux.

@USC_USERAGENT_HOSTNAME@

Cette macro contient le nom d'hôte à partir duquel la demande précédente est reçue.

Cette macro est applicable à tous les flux de travaux.

@USC_USERAGENT_TRANSPORT@

Cette macro contient le protocole de transport HTTP qui est utilisé pour la demande précédente.

Cette macro est applicable à tous les flux de travaux.

@USC_USERAGENT_METHOD@

Cette macro est remplacée par la méthode de soumission HTML qui extrait ce formulaire. La valeur de cette macro peut être GET ou POST.

Cette macro est applicable à tous les flux de travaux.

@USC_USERAGENT_URI@

Cette macro est remplacée par la méthode de soumission HTML qui extrait ce formulaire. La valeur de cette macro peut être `GET` ou `POST`.

Cette macro est applicable à tous les flux de travaux.

@USC_USERAGENT_QUERY@

Cette macro est remplacée par la chaîne de requête contenue dans la demande précédente.

L'exemple ci-après illustre la structure qui existe lorsque l'utilisateur accède à l'URL dans le courrier électronique de réinitialisation de mot de passe :

Remarque : Entrez la syntaxe suivante sur une seule ligne :

```
https://<serveur>:<port>/sps/<fédération_USC>/usc/self/account/recover/
password/validate?usc.confirmation.code=<Caractères aléatoires>
```

Après la réinitialisation du mot de passe, `USC_USERAGENT_QUERY` prend la valeur suivante :

```
usc.confirmation.code=<Caractères aléatoires>
```

Cette macro est disponible au cours des opérations suivantes :

- Inscription des utilisateurs.
- Réinitialisation de mot de passe utilisateur.

@USC_USERAGENT_LOCALES@

Cette macro contient l'environnement local du navigateur à partir duquel la demande précédente est reçue.

Exemple

Lorsque la demande est envoyée depuis un environnement local anglais, la valeur de cette macro est `en_US`.Cette macro est applicable à tous les flux de travaux.

@USC_FORM_PASSWORD@

Lorsque l'utilisateur modifie un mot de passe, cette macro est remplacée par la valeur indiquée dans la demande précédente pour la zone **Ancien mot de passe**.

Cette macro est applicable dans le flux de travail de changement de mot de passe.

@USC_FORM_PASSWORD_NEW@

Cette macro est remplacée par la valeur de la zone **Mot de passe** qui est fournie dans la demande précédente.

Exemple

Lorsque l'utilisateur indique la zone de mot de passe dans le formulaire d'**inscription d'utilisateur** et clique sur **Inscrire**, la page de validation affiche `USC_FORM_PASSWORD_NEW_CONFIRM` avec la valeur de la zone de mot de passe qui est fournie par l'utilisateur.

Cette macro est disponible dans les flux de travaux suivants :

- Flux de travail d'inscription d'utilisateur.
- Flux de travail de changement de mot de passe.
- Flux de travail de mot de passe oublié.

@USC_FORM_PASSWORD_NEW_CONFIRM@

Cette macro est remplacée par la valeur de la zone **Confirmer le mot de passe** qui est fournie dans la demande précédente.

Exemple

Lorsque l'utilisateur indique la valeur de confirmation de mot de passe dans le formulaire d'**inscription d'utilisateur** et clique sur **Inscrire**, la page de validation affiche `USC_FORM_PASSWORD_NEW_CONFIRM` avec la valeur de confirmation de mot de passe.Cette macro est disponible dans les flux de travaux suivants :

- Flux de travail d'inscription d'utilisateur.
- Flux de travail de changement de mot de passe.
- Flux de travail de mot de passe oublié.

@USC_FORM_EMAIL_ADDRESS@

Cette macro est remplacée par la valeur d'**adresse électronique** qui est fournie par l'utilisateur dans le formulaire précédent.

Exemple

Lorsque l'utilisateur indique la valeur d'**adresse électronique** dans le formulaire d'**inscription d'utilisateur** et clique sur **Inscrire**, la page de validation affiche `USC_FORM_EMAIL_ADDRESS` avec la valeur de la zone **Adresse électronique** qui est fournie dans le formulaire précédent.Cette macro est disponible dans les flux de travaux suivants :

- Flux de travail d'inscription d'utilisateur.
- Flux de travail de mot de passe oublié.

Cette macro est disponible une fois qu'un utilisateur répond correctement à la question secrète.

@USC_FORM_EMAIL_ADDRESS_CONFIRM@

Cette macro est remplacée par la valeur de **confirmation d'adresse électronique** qui est fournie par l'utilisateur dans le formulaire précédent.

Exemple

Lorsque l'utilisateur indique la valeur de confirmation d'adresse électronique dans le formulaire d'**inscription d'utilisateur** et clique sur **Inscrire**, la page de validation affiche `USC_FORM_EMAIL_ADDRESS_CONFIRM` avec la valeur de la zone de **confirmation d'adresse électronique** qui est fournie dans le formulaire précédent.

Cette macro est disponible dans les flux de travaux suivants :

- Flux de travail d'inscription d'utilisateur.
- Flux de travail de mot de passe oublié.

Cette macro est disponible une fois qu'un utilisateur répond correctement à la question secrète.

@USC_FORM_USERID_AVAILABLE@

Lorsque l'utilisateur clique sur **ID utilisateur disponible ?** dans la page d'**inscription**, la réponse contient le paramètre `USC_FORM_USERID_AVAILABLE` associé à la valeur true ou false, selon que l'ID utilisateur est disponible ou non.

@USC_FORM_SECRET_QUESTION@

Cette macro est remplacée par la question secrète sélectionnée par l'utilisateur dans la demande précédente.

Exemple

Lorsque l'utilisateur sélectionne une question secrète dans le formulaire **d'inscription d'utilisateur** et clique sur **Inscrire**, USC_FORM_SECRET_QUESTION contient la valeur de la question secrète qui est sélectionnée dans la demande précédente si le formulaire est renvoyé en raison d'une erreur.

Cette macro est disponible dans le flux de travail d'inscription d'utilisateur.

@USC_FORM_SECRET_QUESTION_ANSWER@
Cette macro est remplacée par la réponse à la question secrète sélectionnée par l'utilisateur dans la demande précédente.

Exemple

Lorsque l'utilisateur sélectionne une réponse à une question secrète dans le formulaire **d'inscription d'utilisateur** et clique sur **Inscrire**, USC_FORM_SECRET_QUESTION_ANSWER contient la valeur de la réponse à la question secrète qui est fournie dans la demande précédente si le formulaire est renvoyé en raison d'une erreur.

Cette macro est disponible dans le flux de travail d'inscription d'utilisateur.

A propos des feuilles de style en cascade de User Self Care

Une feuille de style en cascade est un langage de feuille de style qui décrit l'apparence et le format d'un document HTML.

Vous pouvez formater les pages HTML de User Self Care avec une feuille de style en cascade. Les feuilles de style en cascade permettent :

- d'améliorer l'accessibilité au contenu ;
- de fournir souplesse et contrôle dans la spécification des caractéristiques de présentation ;
- d'activer plusieurs pages pour le partage du formatage ;
- de réduire la complexité et les répétitions dans le contenu structurel.

Vous devez utiliser un serveur Web (HTTP Server) pour héberger les pages de feuille de style en cascade. Ce document fournit un exemple de fichier de feuille de style en cascade, de fichier image et de pages HTML de User Self Care pour l'anglais uniquement. Les pages HTML sont modifiées pour utiliser les fichiers de feuille de style en cascade et les fichiers image. Le déploiement des fichiers HTML modifie le comportement de la version anglaise et n'affecte pas les pages de User Self Care dans les autres langues.

Exemples

Fichier	Description

enrollment.html	Dans ce fichier, le fichier de feuille de style en cascade est lié au début du document.
	`<link href="../../css/usc.css" rel="stylesheet">`
	
	Dans votre déploiement, remplacez `../../css/usc.css` par l'emplacement du fichier de feuille de style en cascade sur le serveur Web.
	`<link href="" https://company.com/ css/usc.css"" rel="stylesheet">`

	Cette entrée affiche l'image en haut des pages HTML :
	`<div class="header"> <a href="http: //ibm.com" title="Powered by IBM"><img src="@USC_HTTP_SERVER_URL@/images/ IBMlogo.png" alt="IBM"/></a> </div>`
	
	En phase d'exécution, vous pouvez remplacer cette entrée par :
	`<div class="header"> <a href="http: //ibm.com" title="Powered by IBM"><img src="https://company.com/images/ IBMlogo.png" alt="IBM"/></a> </div>`

usc.css	`cssname` est un nom d'élément HTML pour lequel les attributs sont définis sous `{...}`.
	
	Chaque attribut est une paire clé-valeur. Dans cette définition, la couleur d'arrière-plan HTML correspond au code de couleur 336699. Ce style est appliqué à toutes les pages HTML qui font référence au fichier de feuille de style en cascade.
	`html { ; background-color:#336699; }`
	`cssname { key:val; }`

	Le code suivant définit le style de tous les éléments avec `class="container"`. `container{ background:#fff; margin:0 auto <0px auto; width:640px }`

Formatage des pages HTML de User Self Care à l'aide de macros

Utilisez des macros pour formater les pages HTML de User Self Care. Vous pouvez définir votre propre macro à partir de la règle de mappage.

Procédure

1. Placez la macro dans la page HTML. Par exemple, placez la macro dans `enrollment_validation.html` sous forme de commentaire.

   ```
   <!-- Observe the macro being replaced
   @USC_FORM_PROFILE_TEST_MACRO@
   -->
   ```

2. Publiez les pages dans l'environnement d'exécution de Tivoli Federated Identity Manager.
3. Définissez la macro dans la règle de mappage. La règle de mappage par défaut est dans `<INSTALL_FIM>\examples\js_mappings\usc.js`.
   a. Ajoutez le texte suivant au bas de la règle de mappage.

      ```
      testMacro = "Macro replaced";
      helper.setSTSOutputAttribute("usc.http.profile.test.macro", testMacro);
      ```

      L'attribut `helper.setSTSOutputAttribute` permet d'affecter la valeur `Macro replaced` à l'attribut `usc.http.profile.test.macro` qui est représenté dans la page HTML par `@USC_FORM_PROFILE_TEST_MACRO@`.
4. Appliquez les modifications à la règle de mappage.
   a. Connectez-vous à Integrated Solution Console.
   b. Accédez à **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution**.
   c. Cliquez sur **Propriétés personnalisées de l'environnement d'exécution**.
   d. Affectez la valeur **true** à la propriété **STS.showUSCChains**.
   e. Cliquez sur **OK**.
   f. Déconnectez-vous de la console Integrated Solutions Console.
   g. Connectez-vous à la console Integrated Solutions Console.
   h. Accédez à **Tivoli Federated Identity Manager** > **Configuration du service d'accréditation** > **Chaînes du service d'accréditation**.
   i. Sélectionnez **Chaîne USC** pour **uscCreateAccount**.
   j. Cliquez sur **Propriétés**. Deux entrées **Module de mappage par défaut** sont présentes sous **Modules de la chaîne du service d'accréditation**.
   k. Exécutez les tâches suivantes pour les deux mappages de chaînes :
      1) Sélectionnez **Module de mappage par défaut** et cliquez sur **Propriétés**.
      2) Sous **Règle de mappage d'identité** pour le chaînage du module partenaire, cliquez sur **Modifier la règle**.
      3) Cliquez sur **Parcourir** pour sélectionner la règle de mappage modifiée.
      4) Cliquez sur **Importer le fichier**.
      5) Cliquez sur **OK**.
   l. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**
      1) Accédez à `https://myserver/sps/uscfed/usc/self/account/create`.
      2) Replissez le formulaire d'enregistrement d'utilisateur.

Sur la page Validation de l'inscription User Self Care, la macro est remplacée par

```
<!-- Observe the macro being replaced
Macro replaced
-->
```

Remarque : Les macros définies dans la règle de mappage sont remplacées par leurs valeurs uniquement lorsque la demande est reçue par la chaîne STS et que le module de mappage par défaut contient la définition de la macro.

Formatage des pages HTML de User Self Care à l'aide de feuilles de style en cascade

Utilisez des feuilles de style en cascade pour formater les pages HTML de User Self Care

Procédure

1. Hébergez les fichiers de feuille de style en cascade et les fichiers image sur un système IBM HTTP Server.
   a. Accédez au répertoire d'installation d'IHS.
   b. Ouvrez le fichier `httpd.conf` dans `<RACINE_INSTALL_IHS>\conf` à l'aide d'un logiciel de traitement de texte.
   c. Déterminez la valeur de `'DocumentRoot'` à partir de ce fichier. Ce répertoire correspond à l'emplacement de vos documents de feuille de style en cascade et de vos documents image. La valeur par défaut est `<RACINE_INSTALL_IHS>\htdocs`.
   d. Créez deux dossiers, `css` et `images`, sous le dossier `DocumentRoot(htdocs)`.
   e. Placez le fichier `usc.css` sous le dossier `css` et le fichier image `IBMlogo.png` sous le dossier `images`.
2. Remplacez les pages HTML de User Self Care par défaut par les pages HTML modifiées.
   a. Vérifiez que vous avez édité tous les fichiers HTML pour qu'ils pointent vers le fichier `usc.css` et l'image hébergée sur le serveur Web.

      ```
      ...
      <link href=""https://company.com/css/usc.css"" rel="stylesheet">
      ...
      <div class="header">
      <a href="http://ibm.com" title="Powered by IBM">
      <img src="https://company.com/images/IBMlogo.png" alt="IBM"/></a>
      </div>
      ...
      ```

   b. Appliquez les mêmes modifications aux pages HTML suivantes :
      - `captcha.html`
      - `enrollment.html`
      - `generic.html`
      - `password.html`
      - `profile.html`
   c. Sauvegardez le dossier `usc`.
   d. Remplacez `<INSTALL_FIM>\pages\C\usc` par les exemples de fichier fournis dans la note technique 1614886, *Cascading Style Sheets for Tivoli Federated Identity Manager User Self Care*, sur le portail de support pour Tivoli Federated Identity Manager.

Test des modifications apportées aux fichiers HTML

Une fois que vous avez publié les fichiers HTML sur le module d'exécution de Tivoli Federated Identity Manager, vous pouvez accéder aux pages HTML de User Self Care pour voir les modifications.

Procédure

1. Accédez à `https://myserver/sps/USCFED/usc/self/account/create`.
2. Examinez les pages HTML modifiées.

Intégration de User Self Care à WebSEAL

Les déploiements User Self Care ayant un registre Tivoli Access Manager utilisent la plupart du temps WebSEAL en serveur point de contact. Dans ce cas, vous devez intégrer les interactions entre deux composants qui accomplissent la tâche de suppression de compte et de gestion de mot de passe.

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

- Suppression de compte

  Lorsqu'un utilisateur supprime un compte du registre Tivoli Access Manager, dans le cadre d'un déploiement User Self, assurez-vous que la session en cours est terminée. Cette restriction est requise dans le cadre des recommandations de sécurité.

  La suppression de la session utilisateur inclut la session WebSEAL. Par défaut, User Self Care met fin à la session WebSEAL lorsque le compte est supprimé. Toutefois, cet arrêt dépend de votre utilisation antérieure de l'outil tfimcfg pour configurer WebSEAL en serveur point de contact. Si vous avez exécuté cet outil tel que décrit précédemment, aucune configuration spéciale n'est requise.

  Si vous n'avez pas configuré WebSEAL en serveur point de contact, procédez à cette opération maintenant. Voir «Configuration de WebSEAL en tant que serveur point de contact», à la page 630.

- Gestion de mot de passe

  Deux opérations de gestion de mot de passe sont affectées lorsque WebSEAL est le serveur point de contact. Il s'agit des opérations suivantes : Modifier le mot de passe et Mot de passe expiré. Ces deux intégrations nécessitent que vous autorisiez l'accès non authentifié à la page de modification du mot de passe et que vous utilisiez un formulaire Modifier le mot de passe User Self Care.

Intégration de l'opération de modification de mot de passe à WebSEAL

Lorsque WebSEAL est le serveur point de contact et qu'un utilisateur souhaite modifier un mot de passe, l'utilisateur doit fournir les données. Il existe plusieurs méthodes pour ce faire.

Les voici :

- L'utilisateur peut accéder directement à l'URL Modification de mot de passe User Self Care.

- Le formulaire de modification de mot de passe WebSEAL peut rediriger l'utilisateur vers le même formulaire de User Self Care. Vous pouvez ajouter un réacheminement de balise meta dans la page de modification de mot de passe WebSEAL pour prendre en charge cette action.

Intégration de l'opération de mot de passe expiré à WebSEAL

WebSEAL, en tant que serveur point de contact, gère l'authentification, y compris les mots de passe expirés. Toutefois, lorsque User Self Care est intégré à WebSEAL, il doit gérer la gestion des mots de passe expirés.

Dans ce cas, les étapes suivantes se produisent :

1. WebSEAL indique la session authentifié sur `expired` (expirée).
2. L'utilisateur reçoit une version modifiée du formulaire de mot de passe expiré WebSEAL.
3. L'utilisateur entre des données et soumet le formulaire de mot de passe expiré. Cette action transmet les données de mot de passe au l'URI de modification de mot de passe User Self Care.

   Remarque : Les données de mot de passe doivent répondre à certains critères et être envoyées à l'URL cible User Self Care correct. Lorsque l'utilisateur a soumis le formulaire, User Self Care traite le contenu du formulaire et gère les erreurs. Ce traitement peut inclure l'affichage du formulaire de modification de mot de passe User Self Care à l'utilisateur avec les détails concernant les erreurs.
4. User Self Care gère la modification du mot de passe.
5. La session WebSEAL s'arrête.

   Remarque : La session WebSEAL s'arrête car l'entrée de session gérée par WebSEAL est indiquée `expired` (expirée). Jusqu'à la modification de cet indicateur, l'utilisateur voit toujours le formulaire de modification de mot de passe WebSEAL. L'utilisateur ne peut pas continuer, même après modification de son mot de passe dans User Self Care. L'arrêt de la session est également une mesure de sécurité recommandée, car elle nécessite que l'utilisateur se connecte avec son nouveau mot de passe pour continuer.
6. La page de réussite de modification de mot de passe User Self Care s'affiche à l'utilisateur. Cette page peut être modifiée en vue d'un réacheminement vers WebSEAL si vous le souhaitez.

Etapes de configuration

Effectuez chacune des étapes suivantes pour l'opération à intégrer à WebSEAL :

- Pour intégrer l'opération de modification de mot de passe à WebSEAL :
  1. «Autorisation d'accès non authentifié au formulaire de modification de mot de passe de User Self Care», à la page 675
  2. «Modification du formulaire de modification de mot de passe WebSEAL User Self Care», à la page 675
- Pour intégrer l'opération de mot de passe expiré à WebSEAL :
  1. «Autorisation d'accès non authentifié au formulaire de modification de mot de passe de User Self Care», à la page 675
  2. «Modification du formulaire de modification de mot de passe WebSEAL User Self Care», à la page 675

3. «Modification d'un formulaire de mot de passe expiré WebSEAL», à la page 676
4. «Prise en charge du réacheminement vers WebSEAL», à la page 677

Autorisation d'accès non authentifié au formulaire de modification de mot de passe de User Self Care

Pour prendre en charge l'intégration de l'opération de mot de passe WebSEAL, les utilisateurs non authentifiés doivent pouvoir accéder à l'URI de modification de mot de passe via une jonction WebSEAL. La jonction doit être configurée avec SSL pour la confidentialité.


Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Utilisez pdadmin pour autoriser l'accès non authentifié au formulaire de modification de mot de passe de User Self Care situé à l'emplacement suivant :

WebSEAL_server/*fim_junction*/sps/uscfed/usc/self/password/update

Consultez la documentation Tivoli Access Manager pour des informations concernant la commande **pdadmin**.

Lorsque vous modifiez cet accès, vous devez utiliser un nouveau formulaire de modification de mot de passe User Self Care. Poursuivez avec la rubrique «Modification du formulaire de modification de mot de passe WebSEAL User Self Care».

Modification du formulaire de modification de mot de passe WebSEAL User Self Care

L'ID utilisateur doit être fourni dans le formulaire Modifier le mot de passe lors de l'intégration des opérations de modification de mot de passe User Self Care à WebSEAL.

Avant de commencer


Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Pourquoi et quand exécuter cette tâche

L'autorisation de l'accès non authentifié signifie qu'il est possible aux utilisateurs d'accéder au formulaire de modification du mot de passe. D'une perspective de sécurité, cette action utilisateur est acceptable car l'utilisateur doit entrer son ancien mot de passe dans ce formulaire avant qu'il ne procède à sa modification. Toutefois, vous devez modifier modify to the default User Self Care form to activate this function.

Par défaut, User Self Care n'exigent pas que les utilisateurs entrent leur ID utilisateur dans le formulaire de modification de mot de passe. A la place, User Self Care rassemble les informations d'un contexte authentifié. Ce mécanisme ne fonctionne pas si l'utilisateur ne s'authentifie pas avant de demander le formulaire. Si l'utilisateur demande le formulaire sans authentification, User Self Care renvoie un message d'erreur indiquant qu'aucune identité utilisateur authentifiée n'est disponible.

Pour éviter cette erreur, l'ID utilisateur doit être fourni dans le formulaire Modifier le mot de passe lors de l'intégration des opérations de modification de mot de passe User Self Care à WebSEAL.

Procédure

1. Enregistrez une copie de sauvegarde du fichier *FIM_install_dir*/pages/C/usc/password/changepassword.html.
2. Copiez le fichier d'exemple changepassword.html dans le référentiel de pages User Self Care.
   - Le fichier d'exemple est :
     FIM_install_dir/examples/examples/html/usc/password/changepassword.html
   - L'emplacement de destination est :
     FIM_install_dir/pages/C/usc/password/changepassword.html
3. Connectez-vous à la console d'administration.
4. Accédez au panneau **Gestion des noeuds d'exécution**.
5. Cliquez sur l'option d'**actualisation des pages**.
6. Sauvegardez les modifications de configuration.

Que faire ensuite

- Si vous intégrez l'opération de modification de mot de passe, vous avez terminé la tâche.
- Si vous intégré l'opération d'expiration de mot de passe, passez à la rubrique «Modification d'un formulaire de mot de passe expiré WebSEAL».

Modification d'un formulaire de mot de passe expiré WebSEAL

Modifiez le formulaire de mot de passe expiré WebSEAL pour vous assurer une bonne gestion des mots de passe dans User Self Care.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Vérifiez que vous avez réalisé les tâches prérequises :

1. «Autorisation d'accès non authentifié au formulaire de modification de mot de passe de User Self Care», à la page 675
2. «Modification du formulaire de modification de mot de passe WebSEAL User Self Care», à la page 675

Pourquoi et quand exécuter cette tâche

Il existe plusieurs moyens de modifier le formulaire.

Procédure

1. Copiez le fichier User Self Care `changepassword.html` dans le répertoire WebSEAL dans lequel sont situés les pages de gestion. Renommez-le en `usc_changepassword.html`.

   Par exemple :

   `/opt/pdweb/www-default/lib/html/C/usc_changepassword.html`
2. Modifiez le formulaire `usc_changepassword.html` comme suit :
   a. Ajoutez une nouvelle zone masquée :

      ```
      <input type="hidden" name="usc.form.password.expired.flag" value="true" />
      ```

   b. Ajoutez une autre nouvelle zone masquée :

      ```
      <input type="hidden" name="usc.form.userid" value="%USERNAME%" />
      ```

   c. Supprimez ou commentez les deux lignes :

      ```
      <div class="hidden" id="errorDiv"> </div>
      <div class="hidden" id="errorAttrDiv"> </div>
      ```

   d. Remplacez la macro du formulaire `ACTION` par l'URL de la cible de modification de mot de passe User Self Care.

      Par exemple :

      `https://webseal.example.com/fimjct/sps/uscfed/usc/self/password/update`
3. Définissez les droits et appartenance de fichier de `usc_changepassword.html` de sorte à faire correspondre les droits des autres fichiers de gestion WebSEAL.
4. Editez le fichier de configuration WebSEAL. Accédez à la section `acnt-mgt` et modifiez la valeur `passwd-expired = passwd_exp.html` en `passwd-expired = usc_changepassword.html`
5. Redémarrez WebSEAL.

Que faire ensuite

Vous pouvez éventuellement passer à la rubrique «Prise en charge du réacheminement vers WebSEAL».

Prise en charge du réacheminement vers WebSEAL

Eventuellement, vous pouvez rediriger les utilisateurs vers WebSEAL une fois qu'ils ont modifié leur mot de passe.

Avant de commencer

Les informations contenues dans cette section concernent les utilisateurs du package Tivoli Federated Identity Manager. Elles s'appliquent également aux organisations qui possèdent déjà Tivoli Access Manager for e-business dans leur environnement informatique.

Pourquoi et quand exécuter cette tâche

Parfois, vous pouvez décider d'héberger une *page d'atterrissage* avec des liens vers des destinations à partir du système WebSEAL plutôt que le système User Self Care.

Procédure

1. Créez une page de *réussite de modification de mot de passe* dans le répertoire docs WebSEAL.

   Cette page est la page d'atterrissage de WebSEAL. Elle peut par exemple indiquer que `votre mot de passe a été correctement modifié et que vous devez vous connecter à nouveau pour accéder aux pages protégées`.

2. Modifiez la page User Self Care située dans *FIM_install_dir*`/pages/C/usc/password/changepassword_success.html` pour ajouter une balise de redirection meta qui achemine le client vers la page WebSEAL de *réussite de modification de mot de passe*.

Modification d'une fédération User Self Care

Il existe des restreintes quant à la manière de modifier des fédérations User Self Care existantes.

- L'interface de ligne de commande ne prend pas en charge la modification des fédérations User Self Care. Utilisez la console d'administration pour définir la propriété d'exécution `STS.showUSCChains` à `true`. Affichez les chaînes d'accréditation User Self Care et modifiez-les, ainsi que les propriétés, selon nécessaire.

  Vous pouvez également configurer User Self Care en répétant les étapes de déploiement initiales. Dans ce cas, vous devez créer et éditer un nouveau fichier de réponse, puis utilisez l'interface de ligne de commande pour déployer la fédération.
- Il est impossible de capturer, au sein d'un fichier de réponse, les paramètres de configuration spécifiques à une chaîne particulière. Par exemple, les modules STS de mappage d'attributs utilisent un fichier de règle de mappage. Différentes chaînes peuvent avoir différentes règles de mappage. Vous ne pouvez pas spécifier les différentes règles de mappage lors de la création d'un fichier de réponse d'une configuration existante.

  Les paramètres pouvant être spécifiques à une chaîne particulière n'ont pas de valeurs définies dans le fichier de réponse. Lorsque différentes chaînes ont différentes règles de mappage, utilisez la console d'administration afin de modifier les modules de chaîne pour utiliser des fichiers de règles différents.

Annulation de la configuration de User Self Care

Utilisez **wsadmin** pour annuler la configuration de User Self Care.

Pourquoi et quand exécuter cette tâche

Cette tâche supprime les chaînes d'accréditation User Self Care et la fédération User Self Care.

Procédure

1. Démarrez **wsadmin**.
2. Exécutez la commande :

```
$AdminTask manageItfimUserSelfCare {-operation unconfigure
-fimDomainName your_domain_name -federationName uscfed}
```

Chapitre 43. Réglage de User Self Care

Vous pouvez améliorer les performances de User Self Care en ajustant les paramètres pour plusieurs caches distribués.

User Self Care prend en charge trois caches distribués différents :

- Cache de création de compte
- Cache de mot de passe oublié
- Cache d'échec de question secrète

Les caches sont partagés parmi les membres de cluster WebSphere Application Server pour permettre la bonne gestion d'une opération utilisateur. Ce partage est requis au cas où différentes phases de l'opération ont lieu sur différents noeuds.

User Self Care utilise la technologie WebSphere Distributed Object Cache pour implémenter les caches. Voir la documentation WebSphere Application Server pour plus de détails sur cette technologie de mise en cache.

Il existe deux types de paramètres qui affectent chaque cache distribué User Self Care :

Durées de vie d'entrées
Ces paramètres sont définis dans le fichier de réponses. Les entrées de cache sont conservées jusqu'à la fin de la durée de vie ou jusqu'à ce que l'utilisateur termine l'opération demandant l'entrée de cache. Les noms et paramètres de cette configuration spécifique à la mise en cache sont décrits dans les descriptions de réglage de cache individuel plus loin dans ce document.

Tailles de cache
Ces paramètres sont définis dans la console d'administration en accédant à **Ressources** > **Instances de cache** > **Instances de cache de l'objet**. Le paramètre de `taille de cache` contrôle le nombre d'entrées concurrentes conservées dans le cache. Les noms et paramètres de cette configuration spécifique à la mise en cache sont décrits dans les descriptions

Vous devez définir la taille des caches de manière appropriée pour que les utilisateurs puissent effectuer des opérations nécessitant un cache distribué dans la période configurée. Si un cache est trop petit, il peut arriver que les utilisateurs ne puissent pas valider leurs comptes ou récupérer leurs mots de passe durant la période spécifiée. Vous pouvez spécifier la période dans la configuration de durée de vie des entrées du cache.

Par exemple, pour donner à vos utilisateurs deux minutes pour finir une validation de récupération de compte, configurez la durée de vie de l'entrée pour le cache de validation de récupération de compte sur deux minutes. Si vous attendez deux utilisateurs par seconde pour effectuer une opération de récupération de compte, définissez le cache de validation de récupération de compte sur une valeur d'au moins 240.

Déterminez la taille appropriée à l'aide des calculs suivants :

```
120 secondes x 2 utilisateurs/seconde = 240
```

La taille par défaut du cache de validation de récupération de compte est de 1000 entrées. Cette valeur par défaut serait appropriée dans le cas l'exemple précédent. D'autres opérations, telles que la création de compte, peuvent nécessiter une augmentation de la taille du cache.

Selon l'utilisation attendue de votre système, vous pouvez augmenter la taille d'un ou plusieurs caches. Cet ajustement peut affecter la configuration matérielle requise. Les entrées de cache utilisent de la mémoire ; elles doivent être répliquées entre les systèmes du cluster.

Il est recommandé de fournir un tampon pour la taille du cache attendue.

Voir les rubriques suivantes :

- «Cache de création de compte»
- «Cache de mot de passe oublié», à la page 681
- «Cache d'échec relatif à la question secrète», à la page 681
- «Remarques concernant le réglage des caches», à la page 681

Cache de création de compte

Ce cache stocke les données des entrées utilisateur pendant la création de compte et le message envoyé après le processus de validation. Dès que l'utilisateur termine la validation, User Self Care récupère les données à partir du cache pour créer un compte dans le registre.

Tableau 151. Paramètres de cache de création de compte

Paramètre	Description
AccountCreateLifetime	Les durées de vie des entrées sont contrôlées par le paramètre `AccountCreateLifetime` décrit dans la rubrique : Chapitre 44, «Paramètres de fichier de réponses», à la page 683.
itfim-usc_accountcreate	La taille du cache est contrôlée par la taille du cache `itfim-usc_accountcreate`.

Contrairement aux autres opérations, chaque opération de création de compte crée deux entrées de cache. Une entrée est uniquement composée de l'ID utilisateur et d'une clé. La seconde entrée est composée de toutes les données entrées par l'utilisateur dans le formulaire de création de compte.

Vous configurez les durées de vie d'entrée de cache sur 120 secondes. Attendez-vous à une valeur maximale d'utilisateurs s'inscrivant lors d'une nouvelle opération d'application des accès de 10 chacun/seconde. Vous pouvez décider de la taille de cache suivante :

```
10 utilisateurs/seconde x 2 entrées/utilisateur x 120 secondes/entrée =
2400 x 20% tampon ~= 3000.
```

Cache de mot de passe oublié

Ce cache stocke l'ID utilisateur lors de l'opération de validation Mot de passe oublié.

Tableau 152. Paramètres de cache de mot de passe oublié

Paramètre	Description
AccountRecoveryValidationLifetime	Les durées de vie des entrées sont contrôlées par le paramètre `AccountRecoveryValidationLifetime` décrit à la rubrique : Chapitre 44, «Paramètres de fichier de réponses», à la page 683.
itfim-usc_forgottenpassword	La taille du cache est contrôlée par la taille du cache `itfim-usc_forgottenpassword`. Cette entrée est courte, elle est composée de l'ID utilisateur et d'une clé.

Cache d'échec relatif à la question secrète

Ce cache stocke le nombre de tentatives échouées de réponses à la question secrète ayant eu lieu jusque là.

Tableau 153. Paramètres de cache d'échec relatif à la question secrète

Paramètre	Description
AccountRecoveryFailureLifetime	Les durées de vie des entrées sont contrôlées par le paramètre `AccountRecoveryFailureLifetime` décrit à la rubrique : Chapitre 44, «Paramètres de fichier de réponses», à la page 683
itfim-usc_secretquestionfailures	La taille du cache est contrôlée par la taille du cache `itfim-usc_secretquestionfailures`. Cette entrée est composée d'un chiffre et d'une clé.

Remarques concernant le réglage des caches

La configuration des opérations WebSphere Application Server peut améliorer votre réglage des caches.

- Réplication

  WebSphere ne réplique pas automatiquement toutes les données mises en cache entre les noeuds. A la place, il réplique uniquement les clés entre les noeuds et extrait uniquement les données lorsque la requête est effectuée par un noeud particulier. Si une clé est demandée sur un noeud particulier introuvable sur le cache, User Self Care tente une opération de recherche de cache. La tentative fournit du temps à WebSphere Application Server pour terminer toute réplication possible.
- Vidages de cache

  Le redémarrage de WebSphere Application Server efface les caches et les renvoie à un état propre.
- Suppression de caches User Self Care

  Les entrées de cache sont conservées jusqu'à la fin de la durée de vie de l'entrée ou jusqu'à ce que l'utilisateur termine l'opération demandant l'entrée de cache.

Chapitre 44. Paramètres de fichier de réponses

Utilisez les paramètres décrits dans cette section pour configurer les fichiers de réponses pour User Self Care.

AccountCreateLifetime
Indique la durée, en secondes, pendant laquelle Self Care utilisateur reconnaît la demande de création de compte comme étant valide, et conserve la requête dans le cache interne. Si la chaîne d'accréditation Create Account n'effectue pas de création de compte dans la durée spécifiée, la demande est annulée et la création de compte prend fin.

Cette propriété est obligatoire.
Type : entier
Valeur par défaut : 86 400
Maximum :
aucun
Minimum : 0

Un paramètre de '0' désactive les créations de compte car les entrées ne sont pas conservées dans le cache. Des paramètres plus gros peuvent affecter l'exploitation de mémoire et affecter potentiellement les performances dans les environnements répliqués à cause de l'augmentation de données répliquées à l'aide de DynaCache sur les noeuds.

Lors de la définition de cette propriété, considérez également une taille adéquate pour le `cache itfim-usc_accountcreate`. Voir : Chapitre 43, «Réglage de User Self Care», à la page 679.

AccountRecoveryFailureLifetime
Indique pendant combien de temps, en secondes, le programme conserve l'enregistrement d'une tentative de validation de compte ayant échoué. Lorsque la période indiquée est écoulée, l'enregistrement de la tentative ayant échoué est supprimé et le compteur est décrémenté de un.

Type : entier
Valeur par défaut : 86 400
Maximum :
aucun
Minimum : 0. La valeur 0 indique une désactivation du verrouillage.

Lors de la définition de cette propriété, considérez également une taille adéquate pour le `itfim-usc_secretquestionfailures`. Ce paramètre est configuré séparément lors du réglage de User Self Care. Voir : Chapitre 43, «Réglage de User Self Care», à la page 679.

AccountRecoveryFailureLimit
Indique le nombre de fois qu'un utilisateur peut tenter de restaurer en vain l'accès au compte avant que le programme ne verrouille le compte. Si l'utilisateur ne donne pas la bonne réponse à la question secrète, l'accès au compte n'est pas restauré. Si l'utilisateur échoue à restaurer l'accès au compte, la valeur de cette propriété s'incrémente de un. Lorsque la valeur est égale au nombre indiqué, le programme verrouille le compte.

Type : entier
Valeur par défaut : 3
Maximum :
aucun
Minimum : 0

Un paramètre de 0 ou 1 pour la valeur minimum entraîne le verrouillage du compte dès le premier échec.

AccountRecoveryFailureLockoutTime
Indique pendant combien de temps, en secondes, le programme laisse le compte verrouillé une fois que l'utilisateur a dépassé le nombre maximum de tentatives de validation ayant échoué. Si le programme a verrouillé le compte, cette valeur définit le temps qui doit s'écouler avant que le programme ne déverrouille le compte.

Type : entier
Valeur par défaut : 86 400
Maximum :
aucun
Minimum : 0. La valeur 0 désactive le verrouillage.

AccountRecoveryLookupAttribute
Indique un attribut utilisateur utilisé pour la recherche d'ID utilisateur. Cette propriété indique un attribut unique que l'utilisateur saisit dans le formulaire Forgotten user ID (ID utilisateur oublié) pour récupérer son ID utilisateur (identité). Self Care utilisateur se sert de cet attribut de registre comme d'une zone de recherche. Self Care utilisateur recherche une entrée contenant l'attribut fourni par l'utilisateur et retourne l'ID utilisateur correspondant. La valeur de l'attribut est supposée être une adresse e-mail. Un e-mail contenant tous les ID utilisateur oubliés est envoyé à cette adresse.

Type : chaîne
Par défaut : `mail`

AccountRecoveryLookupField
Cette zone est dépréciée. Ne pas la modifier.

AccountRecoveryValidationAttributes
Cette zone est dépréciée. Ne pas la modifier.

AccountRecoveryValidationLifetime
Indique la durée (en secondes) pendant laquelle User Self Care considère que la demande de validation de compte est valide.

Au cours de la récupération de mot de passe, les utilisateurs doivent exécuter une étape de validation avant de récupérer leur mot de passe. L'étape de validation consiste à répondre à un courrier électronique User Self Care indiquant un lien d'accès. Si l'utilisateur ne répond pas dans la temps défini par ce paramètre, le programme invalide le lien du courrier électronique.

Type : entier
Valeur par défaut : 86 400
Maximum :
aucun
Minimum : 0

Un paramètre de 0 pour la valeur minimum désactive la capacité de récupérage d'un compte.

Lors de la définition de cette propriété, considérez également une taille adéquate pour le cache `itfim-usc_forgottenpassword`. Ce paramètre est configuré séparément lors du réglage de User Self Care. Voir : Chapitre 43, «Réglage de User Self Care», à la page 679.

AttributeMappingFilename
Indique le chemin vers l'emplacement d'un fichier contenant les règles de transformation à utiliser avec le module STS de mappage d'attribut. Ce fichier peut être un fichier JavaScript ou XSLT.

User Self Care est livré avec un fichier JavaScript par défaut appelé `usc.js` :
Federated_Identity_Manager_installation_dir`/examples/js_mappings`

Cette propriété est obligatoire.
Type : chaîne
Par défaut : aucun

Exemple :
`/opt/IBM/FIM/examples/js_mappings/usc.js`

BaseURL
Indique une URL complète pour la racine de la fédération User Self Care. User Self Care utilise la racine pour créer des éléments HTML dynamiques. La syntaxe est la suivante :
method`//`*POC_server:port*`/`*FIM_junction*`/sps`

Où :

method
Doit être `http:`ou `https:`

POC_server:port
Le nom d'hôte complet, et numéro de port facultatif, du serveur point de contact.

FIM_junction
Le nom de la jonction WebSEAL. Cette valeur est requise uniquement lors de l'utilisation d'un serveur point de contact WebSEAL.

Cette propriété est obligatoire.
Type : chaîne
Par défaut : aucun

Exemple :
`https://myWebSEALserver.example.com/myTFIMjct/sps`

Remarque : Si vous utilisez WebSEAL en tant que serveur point de contact, vous n'avez probablement pas encore créé de jonction vers le serveur Tivoli Federated Identity Manager. La plupart du temps, vous créez cette jonction à la fin des étapes de configuration de User Self Care. Toutefois, vous devez déterminer le nom de la jonction maintenant, de sorte à pouvoir définir la valeur BaseURL dans le fichier de réponses. Vous devez mémoriser le nom de jonction, pour une utilisation ultérieure lors de l'exécution de la commande **tfimcfg**.

CaptchaSTSModuleId
Indique le module Captcha de démonstration ou un module de marque de réservation qui ne réalise aucune action. Lorsque cette valeur est spécifiée, User Self Care active le module de démonstration Captcha.

Cette propriété est obligatoire.
Type : chaîne
Par défaut : aucun

Il existe deux valeurs valides pour cette zone :

- `usc-captcha-demo`

  Utilisez cette valeur si vous voulez activer le module de démonstration Captcha. Si vous utilisez ce paramètre, vous devez définir les autres paramètres Captcha dans ce fichier de réponses. Pour utiliser la démonstration Captcha, vous devez également configurer le module. Voir : «Configuration de la démonstration Captcha», à la page 627.

- `default-usc-captcha-noop`

  Utilisez cette valeur si vous voulez utiliser le module de marque de réservation `USCNoOpsSTSModule`. Ce module n'effectue aucune action, mais sert de marque de réservation pour un module de validation fourni par le client qui peut être utilisé, par exemple, pour la validation Captcha. USCNoOpsSTSModule facilite aux clients l'entrée de leur propre module sans redéfinir les chaînes d'accréditation.

`DemoCaptchaImageAndKeyList`
Cette zone est requise si vous utilisez le module de démonstration Captcha.

Le contenu est fixe et ne peut pas être modifié.

Remarque : Le paramètre `DemoCaptchaImageAndKeyList` a déjà été défini. Le programme ignore ce paramètre si vous n'utilisez pas le module de démonstration Captcha.

`DemoCaptchaImageRootURL`
Indique l'URL d'un répertoire contenant les images utilisées pour le module de démonstration Captcha fourni avec User Self Care.

Vous devez indiquer une valeur pour cette propriété si vous souhaitez utiliser le module de démonstration Captcha.

Exemple :

```
https://images.example.com/captcha/demo
```

`EnrollmentEmailSender`
Indique une adresse électronique qualifiée complète pour le compte que Self Care utilisateur utilise pour envoyer un message à l'utilisateur. Le message valide l'inscription de l'utilisateur. La plupart du temps, il s'agit d'une adresse électronique qui ne reçoit pas de réponse.

Cette propriété est obligatoire.
Type : chaîne
Par défaut : aucun

Exemple :

```
no-reply@example.com
```

`EntitySuffix`
Indique un suffixe dans lequel les utilisateurs créés sont stockés dans le registre. Ce suffixe doit identifier un identifiant de manière unique le registre que User Self Care utilise pour toutes les opérations.

Cette propriété est obligatoire.
Type : chaîne
Valeur par défaut : `o=ibm,c=us`

GroupMembershipGroups
Indique une liste de groupes auxquels ajouter des utilisateurs qui viennent d'être écrits. Indique un ou plusieurs groupes définis dans le registre d'utilisateurs utilisé par la chaîne d'accréditation Create Account. Les noms de groupes sont spécifiques au registre d'utilisateurs.

Type : chaîne
Par défaut : aucun

Exemple :

```
<void method="add">
 <string>Group1</string>
</void>
<void method="add">
 <string>Group2</string>
</void>
```

PasswordRecoveryEmailSender
Indique une adresse électronique qualifiée complète pour le compte Self Care utilisateur qui envoie un message à l'utilisateur. Self Care utilisateur utilise le message pour valider une opération de récupération de mot de passe. La plupart du temps, cette adresse électronique ne reçoit pas de réponse.

Cette propriété est obligatoire.
Type : chaîne
Par défaut : aucun

Exemple :

```
no-reply@example.com
```

ProfileManagementAttributes
Définit l'ensemble d'attributs de registre utilisés pour les informations de profil. Afin de fournir un prototype opérationnel, la solution User Self Care définit un ensemble d'attributs de registre à utiliser avec la fonction par défaut. User Self Care ne modifie pas le schéma du registre cible. Pour cette raison, le nombre d'attributs de profil est limité ; il utilise des attributs LDAP standard qui sont présents dans la plupart des cas.

Cette propriété est obligatoire. Voici la liste des attributs utilisés :

- businessCategory
- roomNumber
- mobile
- mail

Les attributs sont représentés dans le fichier de configuration comme suit :

```
<object class="java.util.ArrayList">
 <void method="add">
  <string>businessCategory</string>
 </void>
 <void method="add">
  <string>roomNumber</string>
 </void>
 <void method="add">
       <string>mail</string>
     </void>
 <void method="add">
       <string>mobile</string>
     </void>
</object>
```

Figure 69. Attributs de gestion de profil dans le fichier de réponses

SecretQuestionMinimumNumber
Spécifie le nombre minimal de questions secrètes requises auxquelles un utilisateur doit répondre lors de son inscription. Selon la configuration, certaines ou toutes les questions secrètes peuvent être présentées à l'utilisateur à des fins de vérification lorsqu'il a oublié son mot de passe.

Cette propriété est facultative.
Type : entier
Par défaut : 2 Maximum : aucun
Valeur minimale : 1

SecretQuestionMaximumNumber
Spécifie le nombre maximal de questions secrètes auxquelles un utilisateur peut répondre lors de son inscription. Selon la configuration, toutes les questions secrètes sont présentées à l'utilisateur à des fins de vérification lorsqu'il a oublié son mot de passe.

Cette propriété est facultative.

Type : entier

Valeur par défaut : 3

Maximum : aucun

Minimum : la valeur maximale dépend de `SecretQuestionMinimumNumber`. Elle doit être au moins égale à la valeur spécifiée dans le paramètre `SecretQuestionMinimumNumber`.

SecretQuestionRequiredForValidationNumber
Spécifie le nombre de questions secrètes auxquelles un utilisateur doit répondre correctement pour valider son identité.

Au cours de la récupération de mot de passe, les utilisateurs doivent fournir des réponses correctes aux questions secrètes qui leurs sont posées. Le nombre de questions auxquelles ils doivent répondre correctement dépend de ce paramètre.

Exemple de scénario :

Lors de l'inscription, trois questions secrètes sont présentées à l'utilisateur, auxquelles il doit fournir une réponse.

L'administrateur configure le paramètre sur : `SecretQuestionRequiredForValidationNumber=2`

Lorsque l'utilisateur oublie son mot de passe, les 3 questions secrètes lui sont posées. Cependant, comme le paramètre a été défini sur `SecretQuestionRequiredForValidationNumber=2`, l'utilisateur ne doit répondre correctement qu'à 2 des 3 questions. Il peut laisser une zone sans réponse. Si l'utilisateur choisit de répondre à toutes les questions qui lui sont posées, toutes ses réponses doivent être bonnes.

Dans ce scénario, si un utilisateur choisit de répondre à 3 questions, il doit fournir les bonnes réponses pour que les 3 questions soient validées. L'utilisateur ne peut pas être validé s'il ne répond correctement qu'à 1 des 3 questions.

Cette propriété est facultative.

Type : entier

Par défaut : 2

Maximum : aucun

Minimum : la valeur maximale dépend de `SecretQuestionMinimumNumber`. Elle doit être au plus égale à la valeur spécifiée dans le paramètre `SecretQuestionMaximumNumber`.

SMTPAuthenticatePassword
Mot de passe correspondant au compte indiqué par le paramètre SMTPAuthenticateUsername lors de l'utilisation de l'authentification dans le serveur SMTP. Cette propriété est facultative.

Type : chaîne
Par défaut : aucun

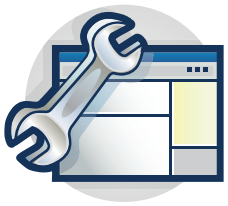
SMTPAuthenticateUsername
Nom d'utilisateur permettant de s'authentifier auprès du serveur SMTP. Cette propriété est facultative.

Type : chaîne
Par défaut : aucun

SMTPServerName
Nom de système hôte qualifié complet du serveur Simple Mail Transport Protocol (SMTP) qui envoie un message électronique pour l'utilisateur. Cette propriété est obligatoire.

Type : chaîne
Par défaut : aucun

Partie 7. Configuration d'un mot de passe à utilisation unique



Les rubriques de la section Configuration vous guident pas à pas lors de la configuration d'un mot de passe à utilisation unique.

La présente section décrit le déploiement d'un mot de passe à utilisation unique. Veuillez d'abord consulter la présentation de la fonction de mot de passe à utilisation unique :

«Présentation du mot de passe à utilisation unique», à la page 693

Chapitre 45. Mot de passe à utilisation unique

Configurez Tivoli Federated Identity Manager pour utiliser le mot de passe à utilisation unique comme facteur d'authentification d'une connexion unique fédérée et dans un scénario d'authentification étendue.

Présentation du mot de passe à utilisation unique

Tivoli Federated Identity Manager fournit plusieurs mécanismes d'authentification dans l'interface de point de contact.

Le serveur point de contact est un proxy ou une application qui interagit avec un utilisateur et gère à la fois l'authentification et les sessions. Dans un déploiement classique, le point de contact est situé au bord d'un réseau protégé et derrière un pare-feu, comme dans une zone démilitarisée, par exemple.

Les méthodes d'authentification disponibles dans un déploiement sont généralement déterminées par la technologie de point de contact utilisée dans l'environnement. Les technologies de point de contact fournissent généralement l'authentification simple telle que l'utilisation d'un nom d'utilisateur et d'un mot de passe.

Dans le cadre d'une authentification renforcée, les utilisateurs qui tentent d'accéder à des ressources sensibles doivent fournir un type de droit d'accès spécifique. Ils peuvent être invités à s'authentifier et à fournir un ensemble de justificatifs supplémentaires pour prouver qu'ils sont autorisés à accéder aux ressources sensibles. L'authentification par mot de passe à utilisation unique peut être utilisée lorsque la sécurité doit être accrue.

Dans le cadre d'une authentification multi-facteur, les utilisateurs doivent fournir plusieurs types de droit d'accès pour accéder à une ressource protégée.

Un mot de passe à utilisation unique est un mot de passe unique qui valide une session de connexion. Il ne peut pas être réutilisé. Ces restrictions le rendent moins vulnérables aux attaques de réexécution et plus sécurisé que des mots de passe statiques.

La fonction d'authentification par mot de passe à utilisation unique dans Tivoli Federated Identity Manager étend la prise en charge du point de contact existant grâce aux fonctions suivantes :

- Détermination des règles d'authentification basées sur le contexte à l'aide d'une règle de mappage
- Génération d'un mot de passe à utilisation unique connectable et validation avec l'implémentation par défaut
- Livraison d'un mot de passe à utilisation unique connectable avec courrier électronique et service SMS comme implémentation par défaut
- Stockage d'un mot de passe à utilisation unique connectable avec prise en charge par défaut de mémoire cache
- Génération d'un mot de passe à utilisation unique basé sur la durée et basé sur un compteur, qui est créé à la fois par le client et le serveur de sorte qu'aucun mécanisme de distribution ne soit nécessaire.

- Stockage et récupération des informations utilisateur connectables pour la génération et la validation d'un mot de passe à utilisation unique nécessitant des informations utilisateur.

Vous pouvez implémenter l'utilisation du mot de passe à utilisation unique dans le protocole fédéré ou le flux d'authentification étendue.

Scénario de connexion unique fédérée
Ce flux consiste à autoriser les opérations avec authentification renforcée et authentification multi-facteur. Il s'appuie sur une authentification par mot de passe à utilisation unique dans le contexte d'un protocole de connexion unique.

Scénario d'authentification étendue
Ce flux consiste à autoriser les opérations avec authentification renforcée et authentification multi-facteur. Il s'appuie sur une authentification par mot de passe à utilisation unique pour étendre les fonctions d'authentification des technologies de point de contact existantes. Ce flux est disponible en dehors du contexte d'une connexion unique fédérée.

Présentation de la configuration du mot de passe à utilisation unique

La fonction de mot de passe à utilisation unique comporte plusieurs composants. Evaluez tout ce que vous devez configurer pour implémenter la fonction afin de répondre à vos besoins.

Vous pouvez implémenter la fonction de mot de passe à utilisation unique dans deux scénarios :

- Scénario de connexion unique fédérée
- Scénario d'authentification étendue

Chaque point de contact est configuré pour être utilisé avec une fédération. Un seul point de contact peut être actif à la fois. Une seule fédération avec un mot de passe à utilisation unique peut être utilisée par plusieurs points de contact.

Deux parties requises doivent être configurées pour activer la fonction de mot de passe à utilisation unique :

- Configurer le profil du point de contact. La configuration de la règle d'authentification basée sur le contexte peut être comprise.
- Configurer la fédération avec un mot de passe à utilisation unique qui établit les noeuds finals Web d'exécution et les modules et chaînes du service de jeton de sécurité (STS) de support. Cette configuration inclut les méthodes de génération du mot de passe à utilisation unique, de validation, et de distribution.

Vous pouvez configurer le point de contact avec le support du mot de passe à utilisation unique dans la console. La fédération avec un mot de passe à utilisation unique ne peut s'effectuer que dans l'interface de ligne de commande.

Une configuration de profil de point de contact permet de configurer plusieurs rappels d'authentification. L'exécution d'un événement d'authentification sur Tivoli Federated Identity Manager se compose de l'ouverture de la liste des rappels d'authentification configurés. L'exécution des flux de mots de passe à utilisation unique se compose de l'ouverture de la liste des rappels d'authentification configurés. Chaque rappel configuré reçoit un niveau d'authentification. Le niveau d'authentification configuré représente le niveau de certitude fourni par chaque rappel d'authentification. Le niveau d'authentification requis par la configuration des règles détermine les rappels configurés qui sont exécutés. Si une session

authentifiée existe lorsque l'événement d'authentification survient, le niveau d'authentification requis détermine si un jeton particulier fourni pour la session authentifiée est satisfaisant.

Le protocole fédéré ou flux User Self Care et le flux d'authentification étendue dépendent des paramètres suivants :

- **Niveau d'authentification requis** détermine le niveau d'authentification requis pour qu'un utilisateur puisse accéder à une ressource protégée. Les niveaux d'authentification sont représentés par un nombre entier. Chaque rappel d'authentification reçoit un niveau d'authentification.

  Pendant un événement d'authentification, les rappels d'authentification configurés sont utilisés. Les règles d'authentification requises sont appliquées. Pour évaluer si une session authentifiée est satisfaisante en fonction des règles, le point de contact Tivoli Federated Identity Manager extrait le niveau d'authentification des droits d'accès. S'il n'existe droit d'accès valide ou satisfaisant, les rappels sont lancés jusqu'à ce qu'un niveau satisfaisant soit atteint. Un administrateur doit affecter un niveau d'authentification à chaque rappel d'authentification configuré.
- **Type d'authentification** détermine le type d'authentification requis pour qu'un utilisateur puisse accéder à une ressource protégée. Il existe deux types d'authentification pris en charge.

  Type d'authentification hiérarchique (renforcée)
  Exécute le rappel d'authentification ayant un niveau d'authentification égal ou supérieur au niveau d'authentification requis. Elle est exécutée jusqu'à ce qu'une authentification satisfaisante soit atteinte.

  Type d'authentification complémentaire (multi-facteur)
  Exécute tous les rappels d'authentification qui sont configurés jusqu'à ce qu'une authentification satisfaisante soit atteinte.
- **Mode d'authentification** détermine le type de mode dans lequel ces rappels d'authentification sont exécutés. Il existe deux types de mode d'authentification pris en charge.

  Mode d'authentification de groupe
  Exécute tous les rappels d'authentification dans le même échange HTTP.

  Mode d'authentification individuelle
  Exécute chaque rappel d'authentification nécessitant une interaction de l'utilisateur dans un échange HTTP distinct.

Vous avez également la possibilité d'utiliser votre propre règle de mappage afin de déterminer la règle d'authentification requise qui est basée sur les attributs de demande. Cette technique est appelée détermination des règles d'authentification basées sur le contexte. Vous pouvez télécharger une règle JavaScript pour déterminer le niveau d'authentification, mode d'authentification et le type d'authentification dans la règle d'authentification générique du point de contact.

Règles d'authentification

- Ensemble de règles qui s'applique au processus d'authentification et à la vérification des données d'authentification.
- Détermine l'application des règles en fonction du contexte de la demande.
- Est constitué du niveau, du mode et du type d'authentification requis.

En l'absence de détermination de la règle d'authentification basée sur le contexte, la règle d'authentification est déterminée de manière statique en fonction de la configuration.

Les règles d'authentification peuvent être définies dans les emplacements suivants :

Règle de mappage des règles d'authentification
Pour plus de détails, voir Personnalisation de la règle de mappage des règles d'authentification.

Chaîne de requête
Pour plus de détails, voir l'étape *Modification de stepuplogin.html* dans «Configuration de l'authentification étendue par mot de passe à utilisation unique avec WebSEAL comme point de contact», à la page 699.

Point de contact
Pour plus de détails, voir le paramètre **allow.authentication.policy.request.overrides** dans «Création de votre propre point de contact de mot de passe à utilisation unique», à la page 704.

Une implémentation par défaut du rappel de la règle d'authentification est fournie. L'implémentation par défaut est configurée pour permettre une configuration de règle statique. Elle peut également reposer sur une règle de mappage configurée par un client afin de déterminer la règle d'authentification à partir des attributs de demande.

Dans le flux d'authentification étendue, un noeud final est fourni pour la technologie de point de contact externe pour démarrer le flux. Une adresse URL cible est fournie dans un paramètre de la chaîne de requête. Les utilisateurs sont redirigés vers cette adresse URL lorsque l'événement d'authentification est terminé.

Les informations de jeton utilisateur d'une authentification existante disponibles dans la demande et les attributs de demande sont collectées. Elles le sont pour déterminer les règles d'authentification basées sur le contexte et le traitement du flux de mot de passe à utilisation unique dans la chaîne STS (Security Token Service).

Tivoli Federated Identity Manager fournit les points d'extension connectables suivants :

- Fournisseur de mot de passe à utilisation unique — génère et valide la valeur du mot de passe à utilisation unique.
- Module de livraison du mot de passe à utilisation unique - fournit la valeur du mot de passe à utilisation unique.
- Module de stockage du mot de passe à utilisation unique - stocke la valeur du mot de passe à utilisation unique.
- Module du fournisseur sur les informations utilisateur du mot de passe à utilisation unique — stocke et récupère les informations utilisateur requises pour calculer la valeur du mot de passe à utilisation unique.
- Rappel des règles d'authentification - détermine la règle d'authentification requise en fonction du contexte de la demande.
- Détermination du fournisseur de mot de passe à utilisation unique dynamique basé sur le contexte de la demande.

Chapitre 46. Déploiement du mot de passe à utilisation unique

Vous devez configurer divers composants, par exemple une fédération de mots de passe à utilisation unique et un point de contact, pour déployer l'authentification par mot de passe à utilisation unique. Le mot de passe à utilisation unique est valide pour le scénario de connexion unique fédérée ou d'authentification étendue.

La liste suivante récapitule les tâches de déploiement du mot de passe à utilisation unique et l'ordre dans lequel les effectuer. Avant de commencer une tâche, vérifiez que vous avez procédé aux tâches préalables requises.

1. Décidez du scénario approprié pour le déploiement de l'authentification par mot de passe à utilisation unique. Voir «Présentation du mot de passe à utilisation unique», à la page 693.
2. Configurez une fédération avec un mot de passe à utilisation unique. Les étapes de configuration incluent la configuration d'un fichier de réponses. Indiquez la configuration des règles de mappage du mot de passe à utilisation unique, des plug-ins du fournisseur de mot de passe à utilisation unique et des plug-ins de livraison de mot de passe à utilisation unique.

   «Configuration d'une fédération avec un mot de passe à utilisation unique»
3. Etape facultative : personnalisez un point de contact de mot de passe à utilisation unique. Vous pouvez créer votre propre profil de point de contact.

   «Création de votre propre point de contact de mot de passe à utilisation unique», à la page 704
4. Activez le point de contact du mot de passe à utilisation unique. Utilisez Integrated Solutions Console pour activer le point de contact du mot de passe à utilisation unique.

   «Activation du point de contact du mot de passe à utilisation unique», à la page 698
5. Selon le scénario dans lequel vous souhaitez déployer l'authentification par mot de passe à utilisation unique, effectuez l'une des tâches suivantes :
   - Personnalisation du mot de passe à utilisation unique pour un scénario de connexion unique fédérée.
   - Configuration de l'authentification étendue par mot de passe à utilisation unique avec WebSEAL comme point de contact

Configuration d'une fédération avec un mot de passe à utilisation unique

Utilisez un fichier de réponses pour configurer votre fédération avec mot de passe à utilisation unique. Indiquez la configuration des règles de mappage du mot de passe à utilisation unique, des plug-ins du fournisseur de mot de passe à utilisation unique et des plug-ins de livraison de mot de passe à utilisation unique.

Procédure

1. Créez un fichier de réponses à l'aide de l'outil wsadmin en entrant les commandes suivantes sur une seule ligne.
   - Création d'un fichier de réponses

     ```
     $AdminTask manageItfimOneTimePassword  { -operation createResponseFile
     -fimDomainName nom_domaine -fileId ID_fichier }
     ```

- Création d'un fichier de réponses basé sur une fédération avec un mot de passe à utilisation unique existante :

  ```
  $AdminTask manageItfimOneTimePassword  { -operation createResponseFile
  -fimDomainName nom_domaine -federationName nom_fédération -fileId ID_fichier }
  ```

  Où :

  domainName est le nom de votre domaine.

  federationName est le nom de la fédération de mot de passe à utilisation unique.

  fileId est le nom du fichier de réponses de mot de passe à utilisation unique.

2. Editez les paramètres contenus dans le fichier de réponses. Voir «Fichier de réponses de mot de passe à utilisation unique», à la page 733.

   Important : Si vous prévoyez d'utiliser `WebSEAL with OTP` en tant que serveur point de contact, vérifiez que le nom de la fédération est `otpfed`.

3. Configurez la fédération de mot de passe à utilisation unique avec votre fichier de réponses en entrant les commandes suivantes sur une seule ligne :

   ```
   $AdminTask manageItfimOneTimePassword { -operation configure
   -fimDomainName nom_domaine -fileId ID_fichier }
   ```

   Où :

   domainName est le nom de votre domaine.

   fileId est le nom du fichier de réponses de mot de passe à utilisation unique créé dans 1, à la page 697

4. Si votre règle de mappage est syntaxiquement valide, mais que Tivoli Federated Identity Manager indique le contraire, ajoutez `STS.validateMappingRules` et définissez la valeur sur `false`. Le message `FBTADM001I Commande exécutée avec succès` est renvoyé.

5. Entrez la commande suivante dans l'outil wsadmin :

   ```
   $AdminTask reloadItfimRuntime { -fimDomainName nom_domaine }
   ```

   Où :

   domainName est le nom de votre domaine.

Activation du point de contact du mot de passe à utilisation unique

Utilisez Integrated Solutions Console pour activer le point de contact du mot de passe à utilisation unique.

Procédure

1. Connectez-vous à la console Integrated Solutions Console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Point de contact**.
3. Sélectionnez **WebSEAL avec mot de passe à utilisation unique** ou créez votre propre point de contact. Pour plus de détails, voir «Création de votre propre point de contact de mot de passe à utilisation unique», à la page 704.
4. Cliquez sur **Activer**.
5. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager.**

Configuration du mot de passe à utilisation unique dans un flux de connexion unique fédérée

Configurez l'authentification par mot de passe à utilisation unique pour activer le flux de connexion unique dans une fédération.

Procédure

1. Exécutez les étapes 1 à 4 de la tâche Deploying one-time password.
2. Facultatif : Si la fédération dans laquelle la fonction de mot de passe à utilisation unique est utilisée a été créée avant Tivoli Federated Identity Manager 6.2.2, groupe de correctifs 4, vous devez exécuter à nouveau la commande `tfimcfg` dans cette fédération afin que l'URL du déclencheur EAI soit ajouté au fichier de configuration WebSEAL. Vous pouvez réutiliser ou recréer la jonction. Pour plus de détails, voir tfimcfg Reference.

Vérification de la configuration de la connexion unique fédérée du mot de passe à utilisation unique

Vérifiez votre configuration de connexion unique fédérée pour vous assurer que l'implémentation avec mot de passe à utilisation unique fonctionne correctement.

Pourquoi et quand exécuter cette tâche

Procédez comme suit pour vérifier que votre configuration avec mot de passe à utilisation unique est correcte dans le flux de connexion unique fédérée.

Procédure

1. Lancez un flux de connexion unique fédérée. Selon le protocole que vous utilisez pour votre fédération, l'URL de noeud final qui servira au lancement de la connexion unique peut différer. Par exemple : `https://idp.example.com/sps/<nom_fédération>/saml20/logininitial?PartnerId=sp.example.com`
2. Vous êtes redirigé vers la page de connexion du fournisseur d'identité.
3. Saisissez votre nom d'utilisateur et votre mot de passe. En fonction de votre configuration, vous pouvez être invité à sélectionner le mode de livraison du mot de passe à utilisation unique.
4. Entrez le mot de passe à utilisation unique fourni avec le mode de livraison que vous avez choisi. Si votre authentification aboutit, vous êtes redirigé vers la ressource protégée dans le fournisseur de services.

Configuration de l'authentification étendue par mot de passe à utilisation unique avec WebSEAL comme point de contact

Configurez WebSEAL pour permettre l'authentification étendue par mot de passe à utilisation unique.

Avant de commencer

Déployez l'authentification du mot de passe à utilisation unique.

Pourquoi et quand exécuter cette tâche

L'entrée de strophe `eai-auth`, qui se trouve dans la strophe `[eai]` du fichier de configuration WebSEAL, active et désactive la fonction de l'interface

d'authentification externe. L'interface d'authentification externe peut être implémentée sur HTTP, HTTPS, ou les deux.

L'interface d'authentification externe est désactivée par défaut.

Procédure

1. Editez le fichier de configuration WebSEAL pour le faire correspondre à la configuration de profil du point de contact Tivoli Federated Identity Manager. Par exemple, `$WEBSEAL_INSTALL_DIRECTORY&/etc/webseald-default.conf`.
   a. Par exemple :

```
[eai]
#----------------------
# EXTERNAL AUTHENTICATION INTERFACE
#----------------------

# Enable EAI authentication. No other EAI parameters will take effect
# if this is set to 'none'.
#
# One of <http, https, both, none>
# Added by FIM TAM autoconfig: Thu Apr 15 12:33:58 CDT 2010
eai-auth = https

# IMPORTANT
# An appropriate authentication library must be configured to handle
# EAI authentication to complete this configuration.  Please
# refer to the "authentication mechanisms and libraries" subsection
# at the end of the authentication section.

# EAI HEADER NAMES

# If eai-auth is not 'none', and WebSEAL has received a trigger URL
# in a request, WebSEAL will examine the corresponding server response for
# the following headers.  These are the headers that
 will contain authentication
# data used to authenticate the user.

# EAI PAC header names
eai-pac-header = am-fim-eai-pac
eai-pac-svc-header = am-eai-pac-svc

# EAI USER ID header names
eai-user-id-header = am-fim-eai-user-id
eai-auth-level-header = am-eai-auth-level
eai-xattrs-header = am-eai-xattrs
# EAI COMMON header names
eai-redir-url-header = am-fim-eai-redir-url
# RETAIN EAI SESSION
# If an already-authenticated EAI client authenticates via an
EAI a second
# time, the existing session and cache entry are completely replaced by
# default.  If retain-eai-session = yes, then the existing session and
# cache entry will be retained, and the credential and relevant data will
# be updated in the existing cache entry.
retain-eai-session = no
eai-redir-url-priority = yes
```

   b. Ajoutez l'URL de déclencheur EAI. Par exemple :

```
# EAI TRIGGER URLS
[eai-trigger-urls]
trigger = /FIM/sps/auth*
```

   c. Ajoutez EAI à la strophe `authentication-mechanisms`. Par exemple :

```
[authentication-mechanisms]
#----------------------
# AUTHENTICATION MECHANISMS AND LIBRARIES
#----------------------
# List of supported authentication mechanisms and
# their associated shared libraries
# Uncomment the line and supply the full path to a library to
# enable a mechanism.

# Username/Password - such as Basic Authentication or Forms
#passwd-cdas       = <passwd-cdas-library>
#passwd-ldap       = <passwd-ldap-library>
#passwd-uraf       = <uraf-authn-library>
passwd-ldap = /opt/PolicyDirector/lib/libldapauthn.so &
-cfgfile [/opt/pdweb/etc/webseald-webseald-ip.conf]
cert-ldap = /opt/PolicyDirector/lib/libcertauthn.so &
-cfgfile [/opt/pdweb/etc/webseald-webseald-ip.conf]

ext-auth-interface = /opt/pdwebrte/lib/libeaiauthn.so
```

Remarque : L'emplacement du fichier de bibliothèque peut être différent sous Windows, par exemple, `C:\Program Files\Tivoli\PDWebRTE\bin\eaiauthn.dll`.

d. Modifiez les paramètres `user session ID`. Par exemple :

```
#----------------------
# USER SESSION IDS
#----------------------
# Enable/disable the creation and handling of user session ids.
user-session-ids = yes

# Include the replica set name in the user session ID.  If set to "yes"
# then the user-session-id will include the replica set.  If set to "no"
# then WebSEAL will not include the replica set in the user-session-id,
# and will assume that all user-sessions specified in
the "terminate session"
# command belong to the standard junction replica set.
user-session-ids-include-replica-set = yes
```

e. Modifiez la strophe `authentication-levels`. Par exemple :

```
[authentication-levels]
0 = unauthenticated
1 = password
2 = ext-auth-interface
```

f. Modifiez la strophe `authentication`. Par exemple :

```
##############################
# AUTHENTICATION
##############################

[ba]
#----------------------
# BASIC AUTHENTICATION
#----------------------

# Enable authentication using the Basic Authentication mechanism
# One of <http, https, both, none>
# Added by FIM TAM autoconfig: Tue Nov 27 10:47:33 SGT 2012
# Enabling forms instead of BA for improved user interface
ba-auth = none



[forms]
#----------------------
# FORMS
#----------------------
```

```
# Enable authentication using the forms authentication mechanism
# One of <http, https, both, none>
# Added by FIM TAM autoconfig: Tue Nov 27 10:47:33 SGT 2012
# Enabling forms instead of BA for improved user interface
forms-auth = https
```

2. Modifiez `stepuplogin.html` afin que la demande d'authentification soit redirigée vers le noeud final d'authentification étendue Tivoli Federated Identity Manager.
   a. Accédez au répertoire qui contient `stepuplogin.html`. Par exemple, `$WEBSEAL_INSTALL_DIRECTORY$/www-default/lib/html/$LOCALE$/stepuplogin.html`.
   b. Insérez le code suivant dans la section `Javascript` du fichier. Par exemple :

```
authnlevel="%AUTHNLEVEL%";
 if (authnlevel == "2")
{
    window.location = "https://<HOST>:<PORT>/<JUNCTION>
/sps/xauth?Target=
%HTTPS_BASE%%URL_ENCODED%[&AuthenticationLevel=<RequiredAuthenticationLevel>]
[&AuthenticationType=
HIERARCHICAL|COMPLEMENTARY][&AuthenticationMode=INDIVIDUAL|GROUP]"
}
```

   Par exemple :

```
authnlevel="%AUTHNLEVEL%";
 if (authnlevel == "2")
{
    window.location = "https://idp.example.com/FIM/sps
/xauth?Target=%HTTPS_BASE%%URL_ENCODED%&AuthenticationLevel=2"
}
```

   Dans cet exemple, la demande d'authentification renforcée au niveau de 2 est réacheminée vers Tivoli Federated Identity Manager.

3. Créez une jonction Tivoli Federated Identity Manager avec les commandes `pdadmin` suivantes de Tivoli Access Manager :

```
pdadmin sec_master> server task <WEBSEAL_SERVER_NAME> create -t tcp
-h <BACKEND_SERVER_HOST_NAME> -p <BACKEND_PORT> -c all -f /FIM
```

   Par exemple :

```
pdadmin sec_master> server task default-webseald-localhost create -t tcp
-h localhost -p 9080 -c all -j -r -q /sps/cgi-bin/query_contents -f /FIM
```

4. Créez une liste de contrôle d'accès non authentifiée relative à l'authentification étendue avec un mot de passe à utilisation unique.
   a. `pdadmin sec_master> acl create xauth_unauth`
   b. `pdadmin sec_master> acl modify xauth_unauth set Group iv-admin TcmdbsvaBRrxl`
   c. `pdadmin sec_master> acl modify xauth_unauth set Group webseal-servers Tgmdbsrxl`
   d. `pdadmin sec_master> acl modify xauth_unauth set User sec_master TcmdbsvaBRrxl`
   e. `pdadmin sec_master> acl modify xauth_unauth set Any-other Tr`
   f. `pdadmin sec_master> acl modify xauth_unauth set Unauthenticated Tr`
   g. `pdadmin sec_master> acl show xauth_unauth`

```
ACL Name: xauth_unauth
Description:
Entries:
```

```
User sec_master TcmdbsvaBRrxl
Any-other Tr
Unauthenticated Tr
Group webseal-servers Tgmdbsrxl
Group iv-admin TcmdbsvaBRrxl
```

5. Liez la liste de contrôle d'accès relative à l'authentification étendue avec un mot de passe à utilisation unique au noeud final de l'authentication étendue à l'aide de la commande suivante :

   ```
   pdadmin sec_master> acl attach /WebSEAL/<WEBSEAL_INSTANCE_ROOT>
   /FIM/sps/xauth xauth_unauth
   ```

   Par exemple :

   ```
   pdadmin sec_master> acl attach /WebSEAL/localhost-webseald-ip
   /FIM/sps/xauth xauth_unauth
   ```

6. Créez une liste de contrôle d'accès authentifiée relative à l'authentification étendue avec un mot de passe à utilisation unique.
   a. `pdadmin sec_master> acl create xauth_anyauth`
   b. `pdadmin sec_master> acl modify xauth_anyauth set Group iv-admin TcmdbsvaBRrxl`
   c. `pdadmin sec_master> acl modify xauth_anyauth set Group webseal-servers Tgmdbsrxl`
   d. `pdadmin sec_master> acl modify xauth_anyauth set User sec_master TcmdbsvaBRrxl`
   e. `pdadmin sec_master> acl modify xauth_anyauth set Any-other Tr`
   f. `pdadmin sec_master> acl modify xauth_anyauth set Unauthenticated T`
   g. `pdadmin sec_master> acl show xauth_anyauth`

      ```
      ACL Name: xauth_anyauth
      Description:
      Entries:
      User sec_master TcmdbsvaBRrxl
      Any-other Tr
      Unauthenticated T
      Group webseal-servers Tgmdbsrxl
      Group iv-admin TcmdbsvaBRrxl
      ```

7. Liez la liste de contrôle d'accès relative à l'authentification étendue avec un mot de passe à utilisation unique au noeud final de l'authentication à l'aide de la commande suivante :

   ```
   pdadmin sec_master> acl attach /WebSEAL/<WEBSEAL_INSTANCE_ROOT>
   /FIM/sps/auth xauth_anyauth
   ```

   Par exemple :

   ```
   pdadmin sec_master> acl attach /WebSEAL/localhost-webseald-ip
   /FIM/sps/auth xauth_anyauth
   ```

8. Redémarrez WebSEAL à l'aide de la commande `pdweb restart`.

Que faire ensuite

Vérification de la configuration de l'authentification étendue par mot de passe à utilisation unique.

Vérification de la configuration de l'authentification étendue par mot de passe à utilisation unique

Vérifiez votre configuration d'authentification étendue pour vous assurer que l'implémentation avec mot de passe à utilisation unique fonctionne correctement.

Pourquoi et quand exécuter cette tâche

Procédez comme suit pour vérifier que votre configuration avec mot de passe à utilisation unique est correcte dans le flux d'authentification étendue. Les commandes fournies dans ces étapes doivent être utilisées dans `pdadmin`. Pour plus d'informations sur l'utilitaire de ligne de commande `pdadmin`, consultez le centre de documentation Tivoli.

Procédure

1. Créez un compte utilisateur test. Par exemple :
   ```
   pdadmin> user create john cn=john,o=ibm,c=us John Doe password
   ```
2. Activez le compte. Par exemple :
   ```
   pdadmin> user modify john account-valid yes
   ```
3. Créez une ressource test protégée par le niveau d'authentification 2 et placez-la dans le répertoire principal de WebSEAL. Par exemple : `/opt/pdweb/www-default/docs/test.html`.
4. Essayez d'accéder à cette ressource via WebSEAL. Par exemple : `https://idp.example.com/test.html`. Un formulaire Web s'affiche pour vous permettre de saisir le nom d'utilisateur et le mot de passe.
5. Entrez les données d'identification que vous avez créées à l'étape 1. Le contenu de la ressource s'affiche.
6. Créez une règle d'objet protégé (POP) avec un niveau d'authentification 2. Par exemple :
   ```
   pdadmin> pop create level2only
   pdadmin> pop modify level2only set ipauth anyothernw 2
   ```
7. Associez la règle à la ressource protégée que vous avez créée à l'étape 3. Par exemple :
   ```
   pdadmin> pop attach /WebSEAL/idp.example.com-default/test.html level2only
   ```
8. Ouvrez une nouvelle session de navigateur et essayez d'accéder à la ressource test à nouveau. Un formulaire Web s'affiche pour vous permettre de saisir le nom d'utilisateur et le mot de passe.
9. Entrez les données d'identification de l'utilisateur test. Vous êtes transféré sur le noeud final d'authentification étendue de Tivoli Federated Identity Manager. Vous démarrez maintenant la fonction de mot de passe à utilisation unique. Selon votre configuration, vous pouvez être invité à sélectionner le mode de livraison du mot de passe à utilisation unique.
10. Entrez le mot de passe à utilisation unique fourni avec le mode de livraison que vous avez choisi. Si votre authentification aboutit, vous êtes redirigé vers la ressource test et vous pouvez accéder à son contenu.

Création de votre propre point de contact de mot de passe à utilisation unique

Tivoli Federated Identity Manager version 6.2.2, groupe de correctifs 4, fournit un profil de point de contact WebSEAL de mot de passe à utilisation unique. Toutefois, vous pouvez créer votre propre point de contact.

Avant de commencer

Avant d'ajouter votre serveur point de contact à votre environnement, vous devez :

- Publier les plug-ins de rappel de point de contact personnalisés.
- Connaître le type de paramètre à utiliser, le cas échéant, et les valeurs correspondantes à transmettre à ces rappels. Voir «Présentation de la configuration du mot de passe à utilisation unique», à la page 694.

Procédure

1. Connectez-vous à la console Integrated Solutions Console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Point de contact**.
3. Sélectionnez votre point de contact.
   - **WebSEAL avec mot de passe à utilisation unique**
4. Cliquez sur **Créer comme**. L'assistant Profil du point de contact s'ouvre.
5. Cliquez sur **Suivant**. Le panneau Nom de profil s'ouvre.
6. Entrez le nom du profil.
7. Facultatif : Entrez une description.
8. Cliquez sur **Suivant**. Le panneau d'ouverture de session s'ouvre.
9. Spécifiez les rappels de connexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

   Etant donné que vous avez créé un point de contact sur la base d'un point de contact existant, les rappels et leurs paramètres sont automatiquement remplis.

   Pour ajouter ou supprimer des rappels, cliquez sur **Ajouter** ou **Supprimer**. Les valeurs de la liste Rappels utilisés sont ceux qui sont utilisés par votre nouveau point de contact.
10. Cliquez sur **Suivant**. Le panneau de fermeture de session s'ouvre.
11. Spécifiez les rappels de déconnexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

    Etant donné que vous avez créé un point de contact sur la base d'un point de contact existant, les rappels et leurs paramètres sont automatiquement remplis.

    Pour ajouter ou supprimer des rappels, cliquez sur **Ajouter** ou **Supprimer**. Les valeurs de la liste Rappels utilisés sont ceux qui sont utilisés par votre nouveau point de contact.
12. Cliquez sur **Suivant**. Le panneau ID local s'ouvre.
13. Spécifiez les rappels à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

    Etant donné que vous avez créé un point de contact sur la base d'un point de contact existant, les rappels et leurs paramètres sont automatiquement remplis.

    Pour ajouter ou supprimer des rappels, cliquez sur **Ajouter** ou **Supprimer**. Les valeurs de la liste Rappels utilisés sont ceux qui sont utilisés par votre nouveau point de contact.
14. Cliquez sur **Suivant**. Le panneau Authentification s'ouvre.
15. Spécifiez les rappels à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

    Etant donné que vous avez créé un point de contact sur la base d'un point de contact existant, les rappels et leurs paramètres sont automatiquement remplis.

Pour ajouter ou supprimer des rappels, cliquez sur **Ajouter** ou **Supprimer**. Les valeurs de la liste Rappels utilisés sont ceux qui sont utilisés par votre nouveau point de contact.

a. Modifiez les paramètres par défaut pour **otpAuthenticateCallback**.
   - **authentication.level** - Ce paramètre indique le niveau d'authentification du rappel. La valeur doit être un nombre entier.
   - **config.federation.name** - Ce paramètre indique le nom de la fédération avec mot de passe à utilisation unique qui est utilisée par le rappel. Cette fédération avec mot de passe à utilisation unique doit exister.

16. Cliquez sur **Suivant**. Le panneau Règle d'authentification s'ouvre.
17. Spécifiez les rappels à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

    Etant donné que vous avez créé un point de contact sur la base d'un point de contact existant, les rappels et leurs paramètres sont automatiquement remplis.

    Pour ajouter ou supprimer des rappels, cliquez sur **Ajouter** ou **Supprimer**. Les valeurs de la liste Rappels utilisés sont ceux qui sont utilisés par votre nouveau point de contact.

    a. Modifiez les paramètres par défaut pour **genericPocAuthenPolicyCallback**.
       - **allow.authentication.policy.request.overrides** - Ce paramètre indique si l'appel doit utiliser la règle d'authentification, qui inclut le niveau d'authentification, le mode d'authentification et le type d'authentification spécifiés dans la chaîne de requête à la place de sa propre règle d'authentification. La valeur doit être une valeur booléenne. Si ce paramètre n'est pas spécifié, la valeur par défaut (false) est utilisée.
       - **authentication.level** - Ce paramètre indique le niveau d'authentification du rappel. La valeur doit être un nombre entier. Si ce paramètre n'est pas spécifié, la valeur par défaut (2) est utilisée.
       - **authentication.mode** - Ce paramètre indique le mode d'authentification du rappel. La valeur doit être *INDIVIDUAL* ou *GROUPAL*. Si ce paramètre n'est pas spécifié, la valeur par défaut (*INDIVIDUAL*), est utilisée.
       - **authentication.type** - Ce paramètre indique le type d'authentification du rappel. La valeur doit être *COMPLEMENTARY* ou *HIERARCHICAL*. Si ce paramètre n'est pas spécifié, la valeur par défaut (*COMPLEMENTARY*) est utilisée.

    b. Facultatif : Téléchargez la règle de mappage **AuthenticationPolicyCallback**. Pour plus d'informations, voir «Personnalisation de la règle de mappage des règles d'authentification», à la page 728.

       Pour télécharger la règle de mappage :

       1) Cliquez sur **Ajouter une règle**.
       2) Cliquez sur **Modifier la règle**.
       3) Cliquez sur **Parcourir** pour rechercher le fichier sur le système.
       4) Cliquez sur **Importer le fichier**.
       5) Cliquez sur **OK**. Le fichier de règles de mappage d'identité est appliqué.

18. Cliquez sur **Suivant**. Le panneau récapitulatif est affiché.
19. Cliquez sur **Terminer**.
20. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Que faire ensuite

Vous devez activer le point de contact du mot de passe à utilisation unique.

Réclamations d'une demande HTTP pour le rappel des règles d'authentification

Configurez le rappel des règles d'authentification afin que les informations de la demande HTTP soient disponibles pendant l'exécution de la règle de mappage pour déterminer la règle d'authentification basée sur le contexte.

Pour activer les paramètres de demande HTTP pour la règle d'authentification basée sur le contexte, vous devez configurer le paramètre suivant :

SPS.http.request.claims.enabled
Lorsqu'il a pour valeur 'true', ce paramètre permet au service SPS (Secure Protocol Service) d'inclure un élément de réclamations WS-Trust. L'élément de réclamations WS-Trust est inclus dans la demande WS-Trust transmise au service STS. L'élément de réclamations contient toutes les informations de demande HTTP reçues par le service SPS qui entraînent l'appel du service STS. Pour éviter les problèmes d'analyse syntaxique XML, les valeurs de la demande sont codées en XML avant d'être incluses en tant que valeurs dans la structure de l'élément de réclamations. Les informations de la demande HTTP suivante sont incluses dans l'élément de réclamations :

- Cookies
- En-têtes HTTP
- Attributs de demande HTTP
- Paramètres de demande HTTP

Exemple de configuration : `SPS.http.request.claims.enabled=true`
Valeur par défaut : False

- Type de valeur : booléen
- Exemple de valeur : true

Les cookies, en-têtes et paramètres d'une demande HTTP peuvent être nombreux et aboutir à un élément de réclamations volumineux. Vous pouvez filtrer les cookies, en-têtes et paramètres de la demande à l'aide d'une propriété personnalisée. Utilisez la propriété personnalisée suivante pour éviter d'inclure des informations qui ne peuvent pas être traitées par le rappel de règle d'authentification :

SPS.http.request.claims.filter.spec
Utilisez ce paramètre de rappel pour spécifier cookies, en-têtes et paramètres de demande à inclure dans l'élément de réclamations.

Pour chaque type de données, vous pouvez choisir d'ajouter toutes les valeurs ou de filtrer les valeurs en fonction du nom de l'élément.

Le filtre par défaut est : **cookies=*:headers=***

Le filtre par défaut inclut tous les cookies et en-têtes et exclut tous les paramètres.

Le format de la syntaxe de spécification du filtre est :

cookies=[*|cookieName1,cookieName2]:

headers=[*|header1,header2]: parameters=[*|param1,param2]

Remarque :

- Pour établir un filtre pour un élément spécifique, définissez la propriété personnalisée avec l'élément spécifique en fonction du type de données auquel elle appartient. Par exemple, si vous souhaitez recevoir un cookie appelé `MyCookie`, spécifiez le filtre comme suit :

  ```
  cookies=MyCookie
  ```

  Pour extraire tous les cookies de la demande mais exclure tous les paramètres et les en-têtes, définissez la propriété personnalisée sur `cookies=*`.
- Les en-têtes, cookies et paramètres peuvent avoir plusieurs valeurs.
- La valeur du cookie comporte la valeur actuelle du cookie, le domaine et le chemin, séparés par un point-virgule (`;`). Par exemple, un cookie nommé `MyCookie` ayant pour valeur `MyValue`, pour chemin `/` et pour domaine `my.domain` est formaté sur le document XML comme suit :

  ```
  <Cookie Name="MyCookie" Type="urn:ibm:names:ITFIM:httprequest:cookies">
      <Value>MyValue; %2F; my.domain</Value>
  </Cookie>
  ```

Exemple d'utilisation de la propriété personnalisée pour activer tous les cookies, en-têtes et paramètres :

```
cookies=*:headers=*:parameters=*
```

L'élément `HTTPRequestClaims` généré est le suivant :

```
<HTTPRequestClaims xmlns="urn:ibm:names:ITFIM:httprequest">
 <Attributes>

  <Attribute Name="remoteAddress"
   Type="urn:ibm:names:ITFIM:httprequest:remoteAddress">
   <Value>127.0.0.1</Value>
  </Attribute>
  <Attribute Name="remoteHost" Type="urn:ibm:names:
      ITFIM:httprequest:remoteHost">
   <Value>fim620</Value>

  </Attribute>
  <Attribute Name="protocol" Type="urn:ibm:names:ITFIM:
    httprequest:protocol">
   <Value>HTTP</Value>
  </Attribute>
  <Attribute Name="method" Type="urn:ibm:names:ITFIM:
      httprequest:method">

   <Value>GET</Value>
  </Attribute>
  <Attribute Name="pathInfo" Type="urn:ibm:names:ITFIM:
      httprequest:pathInfo">
   <Value>/xauth</Value>
  </Attribute>

  <Attribute Name="queryString"
   Type="urn:ibm:names:ITFIM:httprequest:queryString">
   <Value>Target=https://idp.fim.demo.com</Value>
  </Attribute>
  <Attribute Name="requestURI" Type="urn:ibm:names:
      ITFIM:httprequest:requestURI">
   <Value>/sps/xauth</Value>

  </Attribute>
  <Locales>
   <Locale Name="locales" Type="urn:ibm:names:
       ITFIM:httprequest:locales">
```

```
   <Value>en_US</Value>
   <Value>en</Value>

  </Locale>
 </Locales>
</Attributes>
<Headers>
 <Header Name="iv-creds" Type="urn:ibm:names:ITFIM:
   httprequest:headers">
  <Value>Version=1,
   BAKs3DCCBO0MADCCBOcwggT....WgQA
  </Value>

 </Header>
 <Header Name="keep-alive" Type="urn:ibm:names:ITFIM:
   httprequest:headers">
  <Value>115</Value>
 </Header>
 <Header Name="accept-charset" Type="urn:ibm:names:
   ITFIM:httprequest:headers">

  <Value>ISO-8859-1,utf-8;q=0.7,*;q=0.7</Value>
 </Header>
 <Header Name="accept" Type="urn:ibm:names:ITFIM:
  httprequest:headers">
  <Value>text/html,application/xhtml+xml,
     application/xml;q=0.9,*/*;q=0.8
  </Value>
 </Header>

 <Header Name="host" Type="urn:ibm:names:ITFIM:
   httprequest:headers">
  <Value>fim620:9081</Value>
 </Header>
 <Header Name="iv-user" Type="urn:ibm:names:
   ITFIM:httprequest:headers">
  <Value>Unauthenticated</Value>

 </Header>
 <Header Name="referer" Type="urn:ibm:names:ITFIM:
   httprequest:headers">
  <Value>https://saml20ip/FIM/sps/saml20ip/saml20/
      login?SAMLRequest=nVNdT8IwFP0rS....d%2FmV928%3D
  </Value>
 </Header>
 <Header Name="via" Type="urn:ibm:names:ITFIM:
   httprequest:headers">

  <Value>HTTP/1.1 fim620:444</Value>
 </Header>
 <Header Name="content-type" Type="urn:ibm:names:
   ITFIM:httprequest:headers">
  <Value>application/x-www-form-urlencoded</Value>
 </Header>

 <Header Name="iv-groups" Type="urn:ibm:names:ITFIM:
   httprequest:headers">
  <Value />
 </Header>
 <Header Name="iv_server_name" Type="urn:ibm:names:
   ITFIM:httprequest:headers">
  <Value>webseald-sp-webseald-localhost</Value>

 </Header>
 <Header Name="content-length" Type="urn:ibm:names:
   ITFIM:httprequest:headers">
```

```
   <Value>6245</Value>
  </Header>
  <Header Name="accept-language" Type="urn:ibm:names:
    ITFIM:httprequest:headers">
   <Value>en-us,en;q=0.5</Value>
  </Header>

  <Header Name="connection" Type="urn:ibm:names:ITFIM:
    httprequest:headers">
   <Value>close</Value>
  </Header>
 </Headers>
 <Cookies>
  <Cookie Name="jsessionid" Type="urn:ibm:names:ITFIM:
    httprequest:cookies">
   <Value>0000ZOelYEj9RH1aQVymcofXoKc:-1</Value>
  </Cookie>
  <Cookie Name="iv_jct" Type="urn:ibm:names:
     ITFIM:httprequest:cookies">

   <Value>%2FFIM</Value>
  </Cookie>
 </Cookies>
 <Parameters>
  <Parameter Name="Target"
   Type="urn:ibm:names:ITFIM:httprequest:query:param">
   <Value>https://idp.fim.demo.com</Value>
  </Parameter>
 </Parameters>
</HTTPRequestClaims>
```

Remarque : La valeur du type d'attribut de paramètre indique si le paramètre a été reçu à l'aide de la chaîne de requête ou comme partie intégrante du corps de la demande. Concernant les paramètres de la chaîne de requête, le type a la valeur `urn:ibm:names:ITFIM:httprequest:query:param`. Concernant les paramètres reçus comme partie intégrante du corps de la demande, le type a la valeur `urn:ibm:names:ITFIM:httprequest:body:param`.

Dans l'exemple, les cookies, les en-têtes et les paramètres sont filtrés selon les valeurs indiquées.

Cet exemple filtre le cookie `jsessionid`, l'en-tête de l'hôte et le paramètre `RelayState` :

```
cookies=jsessionid:headers=host:parameters=Target
```

Remarque : Les valeurs indiquées pour les paramètres sont sensibles à la casse. Les valeurs des cookies et en-têtes ne sont pas sensibles à la casse.

L'élément `HTTPRequestClaims` généré est le suivant :

```
<HTTPRequestClaims xmlns="urn:ibm:names:ITFIM:httprequest">
 <Attributes>
  <Attribute Name="remoteAddress"
   Type="urn:ibm:names:ITFIM:httprequest:remoteAddress">

   <Value>127.0.0.1</Value>
  </Attribute>
  <Attribute Name="remoteHost"
    Type="urn:ibm:names:ITFIM:httprequest:remoteHost">
   <Value>fim620</Value>
  </Attribute>

  <Attribute Name="protocol"
   Type="urn:ibm:names:ITFIM:httprequest:protocol">
```

```
 <Value>HTTP</Value>
</Attribute>
<Attribute Name="method"
  Type="urn:ibm:names:ITFIM:httprequest:method">
 <Value>GET</Value>

</Attribute>
<Attribute Name="pathInfo"
  Type="urn:ibm:names:ITFIM:httprequest:pathInfo">
 <Value>/xauth</Value>
</Attribute>
<Attribute Name="queryString"
 Type="urn:ibm:names:ITFIM:httprequest:queryString">

 <Value>Target=https://idp.fim.demo.com</Value>
</Attribute>
<Attribute Name="requestURI"
  Type="urn:ibm:names:ITFIM:httprequest:requestURI">
 <Value>/sps/xauth</Value>
</Attribute>

<Locales>
 <Locale Name="locales"
    Type="urn:ibm:names:ITFIM:httprequest:locales">
  <Value>en_US</Value>
  <Value>en</Value>
 </Locale>

 </Locales>
</Attributes>
<Headers>
 <Header Name="host"
  Type="urn:ibm:names:ITFIM:httprequest:headers">
  <Value>fim620:9081</Value>

 </Header>
</Headers>
<Cookies>
 <Cookie Name="jsessionid"
   Type="urn:ibm:names:ITFIM:httprequest:cookies">
  <Value>0000sOnmzkbGcYdIcevoYRuxq0m:-1</Value>

 </Cookie>
</Cookies>
<Parameters>
 <Parameter Name="Target"
  Type="urn:ibm:names:ITFIM:httprequest:query:param">
  <Value>https://idp.fim.demo.com</Value>
 </Parameter>
</Parameters>
</HTTPRequestClaims>
```

Exemple d'élément HTTPRequestClaims, comme illustré dans STSUUSER, pendant l'exécution du mappage de la règle d'authentification :

```
<stsuuser:ContextAttributes>

.......
    <stsuuser:Attribute name="HTTPRequestClaims"
type="urn:ibm:names:ITFIM:httprequest">

        <stsuuser:Value>
            <HTTPRequestClaims xmlns="urn:ibm:names:ITFIM:httprequest">
                    ........
            </HTTPRequestClaims>
```

```
        </stsuuser:Value>
    </stsuuser:Attribute>
.......
</stsuuser:ContextAttributes>
```

Prise en charge du renvoi du mot de passe à utilisation unique

La page de modèle de connexion avec mot de passe à utilisation unique présente un bouton de nouvelle génération et de nouvelle sélection. Le bouton de nouvelle génération permet aux utilisateurs de générer à nouveau un mot de passe à utilisation unique si la valeur du mot de passe à utilisation unique est perdue ou n'a pas été reçue. Le bouton de nouvelle sélection permet aux utilisateurs de resélectionner leur méthode de livraison préférée.

Un clic sur le bouton d'actualisation déclenche les actions suivantes :

1. Invalide l'ancien mot de passe à utilisation unique.
2. Génère une nouvelle valeur de mot de passe à utilisation unique.
3. Utilise le même mécanisme de livraison que celui utilisé lors de la tentative précédente de livraison de la nouvelle valeur.

Un clic sur le bouton de nouvelle sélection déclenche les actions suivantes :

1. Invalide l'ancien mot de passe à utilisation unique.
2. Régénère la liste des méthodes de génération, de livraison et de vérification du mot de passe à utilisation unique.
3. Affiche à nouveau la page de sélection de méthode pour le mot de passe à utilisation unique. L'utilisateur peut alors resélectionner la méthode d'actualisation, de livraison et de vérification du mot de passe à utilisation unique.

Configuration d'un flux de mot de passe à utilisation unique non authentifié

Vous pouvez exécuter le flux de mot de passe à utilisation unique sans que l'authentification préalable de l'utilisateur ne soit nécessaire.

Pourquoi et quand exécuter cette tâche

Avant Tivoli Federated Identity Manager groupe de correctifs 5, le mot de passe à utilisation unique dépendait de l'URL `/sps/auth` pour passer d'une phase du flux à l'autre. L'adresse URL `/sps/auth` était également utilisée pour forcer l'authentification dans les environnements qui utilisent le point de contact WebSEAL. Cette approche rendait impossible l'exécution d'un flux de mot de passe à utilisation unique si l'utilisateur n'était pas authentifié avant l'exécution du flux.

Avec le groupe de correctifs 5 de Tivoli Federated Identity Manager, vous pouvez exécuter le flux de mot de passe à utilisation unique sans que l'authentification préalable de l'utilisateur ne soit nécessaire.

Cette fonction est activée par défaut.

Pour désactiver cette fonction, associez la propriété personnalisée `SPS.POC.use.legacy.auth.url` à `true`. Dans cette configuration, Tivoli Federated Identity Manager aura le comportement existant avant le groupe de correctifs 5.

Cette fonction requiert plusieurs étapes de configuration dans WebSEAL. L'outil `tfimcfg` est activé pour effectuer la configuration nécessaire lorsqu'une fédération de connexion unique est configurée.

Vous devez réexécuter l'outil pour apporter les modifications nécessaires à la configuration.

Dans le cas des environnements qui utilisent le mode d'authentification étendu, les étapes suivantes doivent être effectuées manuellement.

Procédure

1. Configurez une fédération avec un mot de passe à utilisation unique.
2. Ajoutez l'adresse URL `/sps/authservice/authentication` aux adresses URL de déclencheur EAI.

   ```
   # EAI TRIGGER URLS
   [eai-trigger-urls]
   trigger = /FIM/sps/authservice/authentication*
   ```

3. Associez la liste de contrôle d'accès non authentifiée à l'adresse URL `/sps/authservice/authentication`.

   ```
   pdadmin sec_master> acl attach /WebSEAL/<RACINE_INSTANCE_WEBSEAL>/FIM/sps/authservice/authentication xauth_unauth
   ```

   Exemple :

   ```
   pdadmin sec_master> acl attach /WebSEAL/localhost-webseald-ip/FIM/sps/authservice/authentication xauth_unauth
   ```

Migration de fichiers de mots de passe à utilisation unique dans un environnement existant

Mettez à jour les fichiers de mots de passe à utilisation unique pour utiliser des mots de passe à utilisation unique basés sur la durée et basés sur un compteur. Effectuez cette migration après avoir procédé à une mise à niveau de la version 6.2.2, Limited Availability (LA) 5 ou supérieure vers Tivoli Federated Identity Manager, version 6.2.2, groupe de correctifs 7.

Pourquoi et quand exécuter cette tâche

L'installation de Tivoli Federated Identity Manager, version 6.2.2, groupe de correctifs 7 n'écrase pas les pages de mot de passe à utilisation unique et les règles de mappage lorsque vous procédez à une mise à niveau depuis LA 5, ou version supérieure.

Si vous procédez à une mise à niveau à partir du groupe de correctifs 4, les fichiers de mots de passe à utilisation unique sont écrasés. Il n'est donc pas nécessaire de suivre cette procédure pour utiliser des mots de passe à utilisation unique basés sur la durée ou sur un compteur.

Procédure

1. Mettez à jour les pages HTML pour un mot de passe à utilisation unique basé sur la durée et sur un compteur :
   a. Accédez à *FIM_INSTALL_DIR*`/pages_template/`*LOCALE*`/otp/`.
   b. Recherchez les modèles de page HTML suivants et vérifiez que des mises à jour s'appliquent à votre environnement :

Pages HTML mises à jour	Mises à jour dans 6.2.2.7
login.html	• La macro @OTP_METHOD_TYPE@ peut masquer le bouton **Regenerate**. • Si les limites de tentative de connexion sont activées et dépassées, une macro @OTP_LOGIN_DISABLED@ définie dans la règle de mappage otp_verify.js peut désactiver le bouton **Soumettre** lorsqu'elle est utilisée dans une balise d'entrée HTML.
delivery_selection.html	Modification mineure du texte descriptif.

c. Mettez à jour le fichier du même nom dans *FIM_INSTALL_DIR*/pages/*LOCALE*/otp/ en effectuant l'une des actions suivantes :
   - Copiez le nouveau fichier HTML sur le fichier existant si aucune personnalisation n'a été effectuée sur ce fichier.
   - Fusionnez les modifications du nouveau fichier HTML dans le fichier HTML existant pour conserver le contenu personnalisé.

2. Mettez à jour vos règles de mappage existantes avec les informations contenues dans le modèle de règles de mappage :
   a. Accédez à *FIM_INSTALL_DIR*/examples/js_mappings/.
   b. Recherchez les fichiers de règles de mappage suivants et vérifiez que des mises à jour s'appliquent à votre environnement :

Règles de mappage mises à jour	Mises à jour dans 6.2.2.7
otp_get_delivery_methods.js	• L'attribut userInfoType indique le fournisseur d'informations utilisateur User Info Provider pour les mots de passe à utilisation unique basés sur la durée et sur un compteur. S'il n'est pas spécifié ou s'il est défini sur une chaîne vide, aucun fournisseur n'est utilisé. • Une nouvelle méthode de mot de passe à utilisation unique basé sur un compteur a été ajoutée. • Une nouvelle méthode de mot de passe à utilisation unique basé sur la durée a été ajoutée.
otp_deliver.js	Lorsque le mode de livraison est no_delivery, une vérification est effectuée pour ne fournir aucune suggestion de mot de passe à utilisation unique.
otp_verify.js	• Différentes classes sont maintenant importées. • Une exception de limite de tentative de connexion native est utilisée, différente de celle avec IDMappingExtUtils. • Une macro @OTP_LOGIN_DISABLED@ définie par l'utilisateur peut désactiver le bouton **Soumettre** dans login.html lorsque la limite de tentative de connexion est dépassée.

c. Fusionnez les modifications des nouvelles règles de mappage dans les règles de mappage existantes pour conserver le contenu personnalisé.

Personnalisation des mots de passe à utilisation unique

Editez les règles de mappage et les modèles pour personnaliser les mots de passe à utilisation unique.

Personnalisation des règles de mappage des mots de passe à utilisation unique

Editez les règles de mappage des mots de passe à utilisation unique pour personnaliser le traitement des mots de passe à utilisation unique.

Les exemples de règle de mappage se trouvent dans `$FIM_INSTALL_DIR$/examples/js_mappings`. Les règles de mappage des mots de passe à utilisation unique suivantes sont disponibles :

- `otp_get_delivery_methods.js` - Ce fichier contient la règle OTPGetDeliveryMethodsMappingRule.
- `otp_generate.js` - Ce fichier contient la règle OTPGenerateMappingRule.
- `otp_deliver.js` - Ce fichier contient la règle OTPDeliverMappingRule.
- `otp_verify.js` - Ce fichier contient la règle OTPVerifyMappingRule.

Vous pouvez personnaliser les règles de mappage des mots de passe à utilisation unique suivantes :

- «Règle de mappage OTPDeliver»
- «Règle de mappage OTPGenerate», à la page 716
- «Règle de mappage OTPGetDeliveryMethods», à la page 716
- «Règle de mappage OTPVerify», à la page 718

Règle de mappage OTPDeliver

La règle de mappage `OTPDeliver` est exécutée lorsque Tivoli Federated Identity Manager livre le mot de passe à utilisation unique à l'utilisateur.

Utilisez les règles de mappage `OTPDeliver` suivantes :

Générer la suggestion de mot de passe à utilisation unique

Il s'agit d'une séquence de caractères associée au mot de passe à utilisation unique. Tivoli Federated Identity Manager utilise la suggestion de mot de passe à utilisation unique pour indiquer le mot de passe à utilisation unique que l'utilisateur doit soumettre. La suggestion de mot de passe à utilisation unique s'affiche dans la page Connexion avec mot de passe à utilisation unique. Elle est également envoyée à l'utilisateur avec le mot de passe à utilisation unique.

Vous pouvez personnaliser la manière dont la suggestion de mot de passe à utilisation unique est générée en modifiant la section suivante de la règle de mappage `OTPDeliver` par défaut :

```
var otpHint = Math.floor(1000 + (Math.random() * 9000));
```

Remarque : Pour plus de détails, voir les commentaires dans la règle de mappage.

Générer le mot de passe à utilisation unique formaté

Le mot de passe à utilisation unique formaté est la version formatée du

mot de passe à utilisation unique. Tivoli Federated Identity Manager envoie le mot de passe à utilisation unique formaté, au lieu du mot de passe à utilisation unique réel, à l'utilisateur. Par exemple, pour la suggestion de mot de passe à utilisation unique `abcd` et le mot de passe à utilisation unique `12345678`, vous pouvez définir le mot de passe à utilisation unique `abcd-12345678`. Pour la suggestion de mot de passe à utilisation unique `efgh` et le mot de passe à utilisation unique `87654321`, vous pouvez définir le mot de passe à utilisation unique `efgh#8765#4321`.

Vous pouvez personnaliser la manière dont le mot de passe à utilisation unique est généré en modifiant la section suivante dans l'exemple de règle de mappage OTPDeliver :

```
var otpFormatted = otpHint + "-" + otp;
```

Remarque : Pour plus de détails, voir les commentaires dans la règle de mappage.

Modifier le type de livraison de la méthode sélectionnée pour la livraison du mot de passe à utilisation unique

Tivoli Federated Identity Manager utilise le type de livraison pour déterminer le plug-in de livraison de mot de passe à utilisation unique qui livre le mot de passe à utilisation unique à l'utilisateur.

Modifier l'attribut de livraison de la méthode sélectionnée pour livrer Tivoli Federated Identity Manager

L'attribut de livraison est associé au type de livraison. La signification de l'attribut de livraison dépend du plug-in du fournisseur de mot de passe à utilisation unique pour le type de livraison. Par exemple, pour le type de livraison SMS, l'attribut de livraison est le numéro de téléphone portable de l'utilisateur. Pour le type de livraison Courrier électronique, l'attribut de livraison est l'adresse électronique de l'utilisateur.

Remarque : Pour plus de détails, voir les commentaires dans la règle de mappage.

Règle de mappage OTPGenerate

La règle de mappage `OTPGenerate` est une règle de mappage qui est exécutée lorsque Tivoli Federated Identity Manager génère le mot de passe à utilisation unique pour l'utilisateur.

Vous pouvez utiliser la règle de mappage `OTPGenerate` dans la configuration suivante :

Modifier le type de mot de passe à utilisation unique de la méthode sélectionnée pour générer le mot de passe à utilisation unique

Tivoli Federated Identity Manager utilise le type de mot de passe à utilisation unique pour déterminer le plug-in de fournisseur de mot de passe à utilisation unique qui génère le mot de passe à utilisation unique pour l'utilisateur.

Remarque : Pour plus de détails, voir les commentaires dans la règle de mappage.

Règle de mappage OTPGetDeliveryMethods

`OTPGetDeliveryMethods` est une règle de mappage qui s'exécute lorsque le fichier de réponses de mot de passe à utilisation unique extrait les méthodes de livraison du mot de passe à utilisation unique pour l'utilisateur.

Ce modèle de règle de mappage définit les conditions de livraison du mot de passe pour les modes de livraison suivants :

- Par e-mail
- Par SMS
- Aucune livraison

Chaque mode de livraison inclut les attributs suivants ainsi que leur valeur correspondante :

id Indique un ID de mode de livraison unique. Cette valeur remplace la macro `@OTP_METHOD_ID@` sur la page OTP Method Selection. Utilisez une valeur unique pour les différentes méthodes. Par exemple, `sms`.

deliveryType
Indique le plug-in de livraison qui fournit le mot de passe à utilisation unique. La valeur doit correspondre à l'un des types du paramètre `DeliveryTypesToOTPDeliveryModuleIds` du fichier de réponses OTP. Par exemple, `sms_delivery`.

deliveryAttribute
Indique un attribut associé au type de livraison. La valeur dépend du plug-in du fournisseur de mot de passe à utilisation unique pour le type de livraison. Par exemple :

- Pour la livraison par SMS, la valeur est le numéro de téléphone portable de l'utilisateur. Par exemple, `mobileNumber`.
- Pour la livraison par e-mail, la valeur est l'adresse e-mail de l'utilisateur. Par exemple, `adresseElectronique`.
- S'il n'y a aucune livraison, la valeur est une chaîne vide.

label Indique le mode de livraison unique pour l'utilisateur. Pour un mot de passe à utilisation unique basé sur la durée et sur un compteur, utilisez cet attribut pour indiquer la clé secrète de l'utilisateur. Si `label` n'est pas spécifié, le code du mot de passe à utilisation unique basé sur la durée et sur un compteur extrait la clé en appelant le plug-in du fournisseur d'informations sur l'utilisateur. Ce paramètre remplace la macro `@OTP_METHOD_LABEL@` sur la page OTP Method Selection.

otpType
Indique le plug-in du fournisseur de mot de passe à utilisation unique générant et vérifiant le mot de passe. La valeur doit correspondre à l'un des types du paramètre `OTPTypesToOTPProviderModuleIds` du fichier de réponses OTP. Par exemple, `mac_otp`.

userInfoType
Indique le plug-in du fournisseur d'informations sur l'utilisateur à utiliser pour extraire des informations utilisateur requises pour le calcul du mot de passe à utilisation unique. Ce paramètre est uniquement requis si des informations utilisateur sont utilisées pour le calcul du mot de passe à utilisation unique.

Pour personnaliser la livraison du mot de passe à utilisation unique, vous pouvez exécuter l'une des actions suivantes :

- Créer vos propres règles de mappage basées sur le modèle de règle de mappage `OTPGetDeliveryMethods`.
- Modifier le modèle de règle de mappage `OTPGetDeliveryMethods`.

Modèle de règle de mappage OTPGetDeliveryMethods

```
var methods = [];

if (useSMSDelivery) {
    var mobileNumber = new java.lang.String("12345678");
    //var mobileNumber = attributeContainer.getAttributeValueByName("tagvalue_phone");
    var fomattedMobileNumber = mobileNumber;
    if (mobileNumber != null && mobileNumber.length() > 4) {
        formattedMobileNumber = mobileNumber.substring(0, mobileNumber.length() - 4) + "XXXX";
    }
    var method = {
        id: "sms",
        otpType: "mac_otp",
        deliveryType: "sms_delivery",
        deliveryAttribute: mobileNumber,
        label: "SMS to: " + formattedMobileNumber,
        userInfoType: ""
    };
    methods.push(method);
}

if (useEmailDelivery) {
    var emailAddress = new java.lang.String("username@tfim.ibm.com");
    //var emailAddress = attributeContainer.getAttributeValueByName("tagvalue_email");
    var fomattedEmailAddress = emailAddress;
    if (emailAddress != null && emailAddress.length() > 4) {
        fomattedEmailAddress = "XXXX" + emailAddress.substring(4);
    }
    var method = {
        id: "email",
        otpType: "mac_otp",
        deliveryType: "mail_delivery",
        deliveryAttribute: emailAddress,
        label: "Email to: " + fomattedEmailAddress,
        userInfoType: ""
    };
    methods.push(method);
}

if (useTOTP) {
    var method = {
        id: "totp",
        otpType: "totp_otp",
        deliveryType: "no_delivery",
        deliveryAttribute: "SECRET_KEY_GOES_HERE",
        label: "Time Based OTP",
        userInfoType: "file_userinfo"
    };
    methods.push(method);
}

if (useHOTP) {
    var method = {
        id: "hotp",
        otpType: "hotp_otp",
        deliveryType: "no_delivery",
        deliveryAttribute: "SECRET_KEY_GOES_HERE",
        label: "Counter Based OTP",
        userInfoType: "file_userinfo"
    };
    methods.push(method);
}
```

Remarque : Pour plus de détails, voir les commentaires dans la règle de mappage.

Règle de mappage OTPVerify

`OTPVerify` est une règle de mappage qui s'exécute lorsque Tivoli Federated Identity Manager vérifie le mot de passe à utilisation unique soumis par l'utilisateur.

Vous pouvez personnaliser le modèle de règle de mappage `OTPVerify` pour modifier les règles de vérification suivantes :

Modifier le type de mot de passe à utilisation unique de l'utilisateur

Tivoli Federated Identity Manager utilise le type de mot de passe à utilisation unique pour déterminer le plug-in de fournisseur de mot de passe à utilisation unique qui vérifie le mot de passe à utilisation unique soumis par l'utilisateur.

Définir le niveau d'authentification de l'utilisateur

Une fois l'authentification du mot de passe à utilisation unique terminée, des données d'identification contenant le niveau d'authentification de l'utilisateur sont émises. Vous pouvez personnaliser le niveau d'authentification en modifiant la section suivante dans la règle de mappage :

```
var authenticationLevel = contextAttributesAttributeContainer.getAttributeValueByNameAndType
       ("otp.otp-callback.authentication-level", "otp.otp-callback.type");
var attributeAuthenticationLevel = new Attribute("AUTHENTICATION_LEVEL",
       "urn:ibm:names:ITFIM:5.1:accessmanager", authenticationLevel);
attributeContainer.setAttribute(attributeAuthenticationLevel);
```

Imposer le nombre de fois que l'utilisateur peut soumettre le mot de passe à utilisation unique dans la page de connexion avec mot de passe à utilisation unique

Si un utilisateur dépasse le nombre autorisé de soumissions d'un mot de passe à utilisation unique, un message d'erreur s'affiche. Vous pouvez personnaliser le nombre de fois que l'utilisateur peut soumettre le mot de passe à utilisation unique sur la page de connexion avec mot de passe à utilisation unique en modifiant la section suivante dans la règle de mappage :

```
var retryLimit = 5;
```

Par défaut, cette option est définie sur `false`.

Identifier la clé secrète d'un utilisateur

Lorsqu'un utilisateur s'enregistre avec une application à mot de passe à utilisation unique basé sur la durée, une clé secrète lui est affectée. Stockez la clé secrète dans cette règle de mappage pour vérifier l'utilisateur en modifiant le code suivant :

```
var secretStr = new java.lang.String(SECRET_KEY_GOES_HERE);
```

Par défaut, cette option est définie sur `false`.

Pour personnaliser la vérification du mot de passe à utilisation unique, vous pouvez exécuter l'une des actions suivantes :

- Créer vos propres règles de vérification basées sur le modèle de règle de mappage `OTPVerify`.
- Modifier le modèle de règle de mappage `OTPVerify`.

Personnalisation des modèles de page de mot de passe à utilisation unique

Editez les modèles de page de mot de passe à utilisation unique pour personnaliser l'apparence générale des pages affichées par Tivoli Federated Identity Manager à l'utilisateur et pour personnaliser le contenu des SMS et des courriers électroniques envoyés à l'utilisateur par `SMSOTPDelivery` et `EmailOTPDelivery`. `SMSOTPDelivery` et `EmailOTPDelivery` sont des plug-ins de livraison de mot de passe à utilisation unique qui sont fournis avec Tivoli Federated Identity Manager.

Les modèles suivants de page de mot de passe à utilisation unique sont disponibles dans `$FIM_INSTALL_DIR$/pages/$LOCALE$/otp` :

- `login.html`- Modèle de page de mot de passe à utilisation unique pour une connexion.
- `delivery_selection.html` - Modèle de page de mot de passe à utilisation unique pour la sélection de la livraison.
- `errors/allerror.html` - Modèle de page de mot de passe à utilisation unique pour des erreurs générales.
- `errors/error_could_not_validate_otp.html` - Modèle de page de mot de passe à utilisation unique pour une erreur de validation du mot de passe à utilisation unique.
- `errors/error_generating_otp.html` - Modèle de page de mot de passe à utilisation unique pour une erreur de génération de mot de passe à utilisation unique.
- `errors/error_get_delivery_options.html` - Modèle de page de mot de passe à utilisation unique pour une erreur de livraison.
- `errors/error_otp_delivery.html` - Modèle de page de mot de passe à utilisation unique pour une erreur de livraison.
- `errors/error_sts_invoke_failed.html` - Modèle de page de mot de passe à utilisation unique pour une erreur d'opération du service d'accréditation de la sécurité.
- `delivery/sms_message.xml` - Modèle de page de mot de passe à utilisation unique pour service SMS.
- `delivery/email_message.xml` - Modèle de page de mot de passe à utilisation unique pour un courrier électronique.

Après avoir modifié ces pages, vous devez les publier pour que les modifications soient prises en compte. Voir Publication de pages dans l'environnement d'exécution Tivoli Federated Identity.

Modèle de page de mot de passe à utilisation unique pour une connexion

Le modèle de page est utilisé par Tivoli Federated Identity Manager pour afficher l'endroit où l'utilisateur peut saisir le mot de passe à utilisation unique. Cette page est également appelée page de connexion avec un mot de passe à utilisation unique.

Le modèle comprend les macros de remplacement suivantes :

@ERROR_MESSAGE@
Cette macro est remplacée par un message indiquant que le mot de passe à utilisation unique soumis contient des erreurs. Par exemple, le mot de passe à utilisation unique soumis n'est pas valide ou il a été soumis après son expiration.

@MAPPING_RULE_DATA@
Si le mot de passe à utilisation unique soumis comporte une erreur, cette macro est remplacée par la valeur de l'attribut de contexte de l'utilisateur universel STS dont le nom est `@MAPPING_RULE_DATA@` et le type est `otp.sts.macro.type`. Cet attribut de contexte peut être défini dans la «Règle de mappage OTPVerify», à la page 718.

@OTP_HINT@
Cette macro est remplacée par une suggestion de mot de passe à utilisation unique. Il s'agit d'une séquence de caractères associée au mot de passe à

utilisation unique. Cette suggestion est utilisée par Tivoli Federated Identity Manager pour indiquer le mot de passe à utilisation unique que l'utilisateur doit soumettre.

@REGENERATE_ACTION@
Cette macro est remplacée par l'adresse URL à partir de laquelle le bouton de **génération** envoie le formulaire permettant de générer une nouvelle valeur de mot de passe à utilisation unique et de la livrer.

@RESELECT_ACTION@
Cette macro est remplacée par l'adresse URL à partir de laquelle le bouton de **nouvelle sélection** envoie le formulaire permettant de resélectionner la méthode de génération, de livraison et de vérification de la valeur de mot de passe à utilisation unique.

@OTP_METHOD_TYPE@
Cette macro est remplacée par le type de la méthode de génération, de livraison et de vérification du mot de passe à utilisation unique qui est sélectionnée. Ce type est généré par la règle de mappage `OTPGetDeliveryMethods` et a été sélectionné par l'utilisateur.

Modèle de page de mot de passe à utilisation unique pour le choix du mode de livraison

Ce modèle de page affiche la liste des méthodes pour la génération, la livraison et la vérification du mot de passe à utilisation unique. Cette page est également appelée page de sélection de méthode pour le mot de passe à utilisation unique.

@OTP_METHOD_ID@
Cette macro est remplacée par l'ID de la méthode de génération, de livraison et de vérification du mot de passe à utilisation unique. Cet ID est généré par la règle de mappage `OTPGetDeliveryMethods`.

@OTP_METHOD_LABEL@
Cette macro est remplacée par le libellé de la méthode de génération, de livraison et de vérification du mot de passe à utilisation unique. Ce libellé est généré par la règle de mappage `OTPGetDeliveryMethods`.

@OTP_METHOD_CHECKED@
Pour la première méthode, cette macro est remplacée par un attribut de bouton d'option HTML qui entraîne la sélection du bouton d'option. Pour les autres méthodes de génération, de livraison et de vérification, cette macro est remplacée par une chaîne vide.

Modèle de page de mot de passe à utilisation unique pour des erreurs générales

Ce modèle de page est utilisé par Tivoli Federated Identity Manager pour afficher les erreurs générales qui surviennent dans le flux de mot de passe à utilisation unique. Les erreurs générales sont des erreurs qui ne sont pas affichées dans les autres modèles.

Le modèle comprend les macros de remplacement suivantes :

@REQ_ADDR@
Cette macro est remplacée par l'URL dans laquelle la demande de l'utilisateur est envoyée.

@TIMESTAMP@
Cette macro est remplacée par l'horodatage de l'erreur.

@DETAIL@
Cette macro est remplacée par le message d'erreur.

@EXCEPTION_STACK@
Cette macro est remplacée par la trace de pile de l'erreur.

Figure 70. Modèle pour `allerror.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>One-Time Password Error</title>
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color: #ff8800">An error has occurred.</h2>
    <div id="infoDiv" style="background-color:#ffffff;color:#000000">
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em> <br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
     border-style:solid; border-width:1px; border-color:#000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color:#999999;
     border-style:solid; border-width:1px; border-color:#000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

Modèle de page de mot de passe à utilisation unique pour une erreur de génération de mot de passe à utilisation unique

Ce modèle de page est utilisé par Tivoli Federated Identity Manager pour afficher les erreurs qui surviennent lorsque Tivoli Federated Identity Manager génère un mot de passe à utilisation unique.

Le modèle comprend les macros de remplacement suivantes :

@REQ_ADDR@
Cette macro est remplacée par l'URL dans laquelle la demande de l'utilisateur est envoyée.

@TIMESTAMP@
Cette macro est remplacée par l'horodatage de l'erreur.

@DETAIL@
Cette macro est remplacée par le message d'erreur.

@EXCEPTION_STACK@
Cette macro est remplacée par la trace de pile de l'erreur.

Figure 71. Modèle pour `error_generating_otp.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>One-Time Password Error</title>
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color:#ff8800">An error occurred while
     generating the one-time password.</h2>
    <div id="infoDiv" style="background-color:#ffffff;color:#000000">
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em><br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
      border-style:solid; border-width:1px; border-color:#000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color:#999999;
      border-style:solid; border-width:1px; border-color:#000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

Modèle de page de mot de passe à utilisation unique pour une erreur de livraison

Ce modèle de page est utilisé par Tivoli Federated Identity Manager pour afficher les erreurs qui surviennent lorsque Tivoli Federated Identity Manager extrait la liste des méthodes de livraison du mot de passe à utilisation unique à l'utilisateur.

Le modèle comprend les macros de remplacement suivantes :

@REQ_ADDR@
Cette macro est remplacée par l'URL dans laquelle la demande de l'utilisateur est envoyée.

@TIMESTAMP@
Cette macro est remplacée par l'horodatage de l'erreur.

@DETAIL@
Cette macro est remplacée par le message d'erreur.

@EXCEPTION_STACK@
Cette macro est remplacée par la trace de pile de l'erreur.

Figure 72. Modèle pour `error_get_delivery_options.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>One-Time Password Error</title>
```

```
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color:#ff8800">An error occurred while
    obtaining the one-time password delivery options.</h2>
    <div id="infoDiv" style="background-color:#ffffff;color:#000000">
      <em>@REQ_ADDR@</em> <br />
      <em>@TIMESTAMP@</em> <br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
     border-style:solid; border-width:1px; border-color:#000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color:#999999;
     border-style:solid; border-width:1px; border-color:#000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

Modèle de page de mot de passe à utilisation unique pour une erreur de livraison

Ce modèle de page est utilisé par Tivoli Federated Identity Manager pour afficher les erreurs qui surviennent lorsque Tivoli Federated Identity Manager fournit le mot de passe à utilisation unique à un utilisateur.

Le modèle comprend les macros de remplacement suivantes :

@REQ_ADDR@
Cette macro est remplacée par l'URL dans laquelle la demande de l'utilisateur est envoyée.

@TIMESTAMP@
Cette macro est remplacée par l'horodatage de l'erreur.

@DETAIL@
Cette macro est remplacée par le message d'erreur.

@EXCEPTION_STACK@
Cette macro est remplacée par la trace de pile de l'erreur.

Figure 73. Modèle pour `error_otp_delivery.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>One-Time Password Error</title>
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color:#ff8800">An error occurred while
    delivering the one-time password value.</h2>
    <div id="infoDiv" style="background-color:#ffffff;color:#000000">
```

```
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em><br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
     border-style:solid; border-width:1px; border-color:#000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color:#999999;
     border-style:solid; border-width:1px; border-color:#000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

Modèle de page de mot de passe à utilisation unique pour une erreur d'opération du service d'accréditation de la sécurité

Ce modèle de page est utilisé par Tivoli Federated Identity Manager pour afficher les erreurs qui surviennent lorsque Tivoli Federated Identity Manager appelle le service de jeton de sécurité.

Le modèle comprend les macros de remplacement suivantes :

@REQ_ADDR@
: Cette macro est remplacée par l'URL dans laquelle la demande de l'utilisateur est envoyée.

@TIMESTAMP@
: Cette macro est remplacée par l'horodatage de l'erreur.

@DETAIL@
: Cette macro est remplacée par le message d'erreur.

@EXCEPTION_STACK@
: Cette macro est remplacée par la trace de pile de l'erreur.

Figure 74. Modèle pour `error_sts_invoke_failed.html`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>One-Time Password Error</title>
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color: #ff8800">An error occurred while
    invoking the trust service to perfom a one-time password operation.</h2>
    <div id="infoDiv" style="background-color:#ffffff;color:#000000">
      <em>@REQ_ADDR@</em><br />
      <em>@TIMESTAMP@</em><br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
     border-style:solid; border-width:1px; border-color:#000000">
      <h4>Error details</h4>
```

```
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color:#999999;
    border-style:solid; border-width:1px; border-color:#000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
    </div>
  </div>
</body>
</html>
```

Modèle de page de mot de passe à utilisation unique pour une erreur de validation du mot de passe à utilisation unique

Ce modèle de page est utilisé par Tivoli Federated Identity Manager pour afficher les erreurs qui surviennent lorsque Tivoli Federated Identity Manager valide le mot de passe à utilisation unique soumis par l'utilisateur.

Le modèle comprend les macros de remplacement suivantes :

@REQ_ADDR@
Cette macro est remplacée par l'URL dans laquelle la demande de l'utilisateur est envoyée.

@TIMESTAMP@
Cette macro est remplacée par l'horodatage de l'erreur.

@DETAIL@
Cette macro est remplacée par le message d'erreur.

@EXCEPTION_STACK@
Cette macro est remplacée par la trace de pile de l'erreur.

Figure 75. Modèle pour error_could_not_validate_otp.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>One-Time Password Error</title>
</head>

<body style="background-color:#ffffff">
  <div>
    <h2 style="color: #ff8800">The one-time password
     value could not be validated.</h2>
    <div id="infoDiv" style="background-color:#ffffff;color:#000000">
      <em>@REQ_ADDR@</em> <br />
      <em>@TIMESTAMP@</em> <br />
    </div>
    <br />
    <div id="detailDiv" style="background-color: #999999;
    border-style:solid; border-width:1px; border-color:#000000">
      <h4>Error details</h4>
      @DETAIL@
    </div>
    <br />
    <div id="stackDiv" style="background-color:#999999;
    border-style:solid; border-width:1px; border-color:#000000">
      <h4>Stack trace</h4>
      @EXCEPTION_STACK@
```

```
    </div>
  </div>
</body>
</html>
```

Modèle de page de mot de passe à utilisation unique pour service SMS

Ce modèle de page est utilisé par `SMSOTPDelivery` comme contenu du SMS qui est envoyé à l'utilisateur.

Le modèle comprend la macro de remplacement suivante :

@OTP_STRING@
: Cette macro est remplacée par le mot de passe à utilisation unique généré par le plug-in du fournisseur de mots de passe à utilisation unique.

Figure 76. Modèle pour `sms_message.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<Message>
  <Value>
    This is your one-time password @OTP_STRING@.

    Thank you,
    OTP Test
  </Value>
</Message>
</root>
```

Modèle de page de mot de passe à utilisation unique pour un courrier électronique

Ce modèle de page est utilisé par `EmailOTPDelivery` comme contenu du courrier électronique qui est envoyé à l'utilisateur.

Le modèle comprend la macro de remplacement suivante :

@OTP_STRING@
: Cette macro est remplacée par le mot de passe à utilisation unique généré par le fournisseur de mots de passe à utilisation unique.

Figure 77. Modèle pour `email_message.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<Subject>
  <Value>
    One-time password
  </Value>
</Subject>

<Message>
  <Value>
    This is your one-time password @OTP_STRING@.

    Merci,
```

```
    OTP Test
  </Value>
</Message>
</root>
```

Personnalisation de la règle de mappage des règles d'authentification

La règle de mappage des règles d'authentification détermine la règle d'authentification qui est basée sur le contexte de demande.

Les règles d'authentification sont un ensemble de règles qui s'appliquent au processus d'authentification et à la vérification des données d'authentification. L'application des règles d'authentification est déterminée par le contexte de demande.

Les règles d'authentification sont constituées du niveau d'authentification requis, du mode d'authentification et du type d'authentification. Utilisez une règle de mappage JavaScript pour déterminer les paramètres de niveau d'authentification, de mode d'authentification et de type d'authentification. Pour plus d'informations, voir «Présentation de la configuration du mot de passe à utilisation unique», à la page 694.

Les exemples de règle de mappage des règles d'authentification se trouvent dans `$FIM_INSTALL_DIR$/examples/js_mappings`.

Vous pouvez modifier le code suivant dans la règle de `otp_authnpolicy.js`.

```
// Override the authentication mode, level, and type based on the ip address of the user
var ipAddress = stsuu.getAttributeValueByName("AZN_CRED_NETWORK_ADDRESS_STR");
if (ipAddress != null && ipAddress.indexOf("YOUR_IP_ADDRESS_PREFIX") == 0) {
    var contextAttributes = stsuu.getContextAttributes();

    var authModeAttr = new Attribute("AuthenticationMode", null, "INDIVIDUAL");
    contextAttributes.setAttribute(authModeAttr);

    var authLevelAttr = new Attribute("AuthenticationLevel", null, "2");
    contextAttributes.setAttribute(authLevelAttr);

    var authTypeAttr =
    new Attribute("AuthenticationType", null, "HIERARCHICAL");
    contextAttributes.setAttribute(authTypeAttr);
```

Remarque : Pour plus de détails, voir les commentaires dans les règles de mappage.

Concepts associés:

«Présentation du mot de passe à utilisation unique», à la page 693
Tivoli Federated Identity Manager fournit plusieurs mécanismes d'authentification dans l'interface de point de contact.

Création de macros définies par l'utilisateur

La page de sélection de méthode pour le mot de passe à utilisation unique et la page de connexion avec mot de passe à utilisation unique contiennent des macros que vous pouvez utiliser à différentes fins.

Pourquoi et quand exécuter cette tâche

En plus de ces macros, vous pouvez définir vos propres macros. La définition d'une macro est un processus en deux étapes.

Procédure

1. Spécifiez le nom et la valeur des macros dans les règles de mappage de mot de passe à utilisation unique.
   a. Pour définir une macro dans la règle de mappage OTP, ajoutez un attribut dans l'attribut de contexte d'utilisateur universel STS.
   b. Utilisez `otp.sts.macro.type` comme type pour l'attribut.
   c. Utilisez le nom de la macro comme nom pour l'attribut.
   d. Utilisez la valeur de la macro comme valeur pour l'attribut.

   La valeur de la macro doit être une chaîne. Exemple de code JavaScript :

   ```
   var contextUserMacro = new Attribute("@OTP_MAPPING_RULE_DATA@",
   "otp.sts.macro.type", "Data from OTP mapping rule");
   stsuu.getContextAttributesAttributeContainer().setAttribute(contextUserMacro);
   ```

   Dans cet exemple, le nom de la macro est `@OTP_MAPPING_RULE_DATA@` et la valeur de la macro est `Données de la règle de mappage OTP`.
2. Ajoutez le nom des macros dans la page de sélection de méthode et les pages de connexion avec mot de passe à utilisation unique.
   a. Pour afficher des macros dans la page de sélection de méthode pour le mot de passe à utilisation unique, définissez-les dans la règle de mappage `OTPGetDeliveryMethods`.
   b. Pour afficher des macros dans la page de connexion avec mot de passe à utilisation unique, définissez-les dans la règle de mappage `OTPGenerate`, la règle de mappage `OTPDeliver` ou la règle de mappage `OTPVerify`, selon l'opération à l'origine de l'affichage de la page de connexion avec mot de passe à utilisation unique.

      Si la page de connexion avec mot de passe à utilisation unique s'affiche après la génération et la livraison d'un mot de passe à utilisation unique, les macros qui sont définies dans les règles de mappage `OTPGenerate` et `OTPDeliver` sont utilisées. Si la page de connexion avec mot de passe à utilisation unique s'affiche une fois que le plug-in de fournisseur de mot de passe à utilisation unique a déterminé que le mot de passe à utilisation unique soumis n'est pas correct, les macros qui sont définies dans la règle de mappage `OTPVerify` sont utilisées.

manageItfimOneTimePassword

La commande **`manageItfimOneTimePassword`** permet de lister, d'afficher, de configurer, de modifier les fédérations avec un mot de passe à utilisation unique, et d'annuler leur configuration.

Actions

La commande **`manageItfimOneTimePassword`** permet d'effectuer les opérations suivantes :

- Affichage de toutes les fédérations avec un mot de passe à utilisation unique
- Affichage d'une fédération avec un mot de passe à utilisation unique
- Configuration d'une fédération avec un mot de passe à utilisation unique
- Modification d'une fédération avec un mot de passe à utilisation unique
- Annulation de la configuration d'une fédération avec un mot de passe à utilisation unique
- Création d'un fichier de réponses de mot de passe à utilisation unique

Syntaxe

```
$AdminTask manageItfimOneTimePassword {-operation opération -fimDomainName
nom_domaine [paramètres_facultatifs]}
```

Paramètres

Les paramètres opération et nom_domaine sont obligatoires. Les paramètres suivants sont facultatifs :

-federationName *nom_fédération*

-fileId *ID_fichier*

-humanReadable *humanReadable*

L'utilisation de ces paramètres dépend de l'opération à exécuter.

Les paramètres ci-dessous peuvent être utilisés avec la commande **manageItfimOneTimePassword** :

-operation *operation*
Indique l'opération que vous souhaitez effectuer. Le tableau 1 répertorie les opérations prises en charge par cette commande.

Tableau 154. Valeurs du paramètre **-operation**

Valeur	Description et conditions requises
list	Recense toutes les fédérations avec un mot de passe à utilisation unique.
view	Affiche les détails d'une fédération avec un mot de passe à utilisation unique. Lorsque vous utilisez cet opérateur, vous devez également utiliser les paramètres suivants : **federationName** *federationName* Nom de la fédération avec un mot de passe à utilisation unique que vous souhaitez afficher. **humanReadable** *humanReadable* Paramètre qui génère un affichage lisible par l'utilisateur de la fédération avec un mot de passe à utilisation unique.
configure	Configure une fédération avec un mot de passe à utilisation unique. Lorsque vous utilisez cet opérateur, vous devez également utiliser les paramètres suivants : **fileId** *fileId* Nom du fichier de réponses en fonction duquel la nouvelle fédération avec mot de passe à utilisation unique est configurée. Le nom de la nouvelle fédération avec mot de passe à utilisation unique configurée est indiqué dans le fichier de réponses à l'aide de la propriété **FedName**.

Tableau 154. Valeurs du paramètre **-operation** *(suite)*

Valeur	Description et conditions requises
unconfigure	Annule la configuration d'une fédération avec un mot de passe à utilisation unique. Lorsque vous utilisez cet opérateur, vous devez également utiliser les paramètres suivants : **federationName** *federationName* Nom de la fédération avec un mot de passe à utilisation unique dont vous souhaitez annuler la configuration.
modify	Modifie une fédération avec un mot de passe à utilisation unique. Lorsque vous utilisez cet opérateur, vous devez également utiliser les paramètres suivants : **federationName** *federationName* Nom de la fédération avec un mot de passe à utilisation unique que vous souhaitez modifier. **fileId** *fileId* Nom du fichier de réponses de mot de passe à utilisation unique en fonction duquel la fédération avec mot de passe à utilisation unique est modifiée. Pour modifier une fédération avec mot de passe à utilisation unique, procédez comme suit : 1. Créez un fichier de réponses qui est basé sur la fédération avec un mot de passe à utilisation unique que vous souhaitez modifier. 2. Ouvrez le fichier de réponses de mot de passe à utilisation unique dans un éditeur de texte. 3. Modifiez les paramètres que vous souhaitez changer. 4. Sauvegardez et fermez le fichier. 5. Exécutez l'opération *modify* en spécifiant le nom du fichier de réponses dans le paramètre **fileId**. Si vous souhaitez modifier le nom de la fédération avec mot de passe à utilisation unique, utilisez le paramètre **FedName** dans le fichier de réponses de mot de passe à utilisation unique.

Tableau 154. Valeurs du paramètre **-operation** *(suite)*

Valeur	Description et conditions requises
createResponseFile	Crée un fichier de réponses de mot de passe à utilisation unique. Vous pouvez créer un fichier de réponses exemple ou un fichier de réponses basé sur une fédération avec un mot de passe à utilisation unique existante. **Création d'un fichier de réponses exemple** Lorsque vous utilisez cet opérateur, vous devez également utiliser les paramètres suivants : **fileId** *fileId* Nom du fichier de réponses de mot de passe à utilisation unique. **Création d'un fichier de réponses basé sur une fédération avec un mot de passe à utilisation unique existante** Lorsque vous utilisez cet opérateur, vous devez également utiliser les paramètres suivants : • **federationName** *federationName* Nom de la fédération avec un mot de passe à utilisation unique à partir de laquelle est créé le fichier de réponses. • **fileId** *fileId* Nom du fichier de réponses de mot de passe à utilisation unique.

-fimDomainName *nom_domaine_fim*
Indique le nom du domaine dans lequel l'opération s'exécute. Le domaine doit exister.

-federationName *nom_fédération*
Indique le nom de la fédération avec un mot de passe à utilisation unique existante. La fédération doit exister.

-fileId *ID_fichier*
Indique le nom du fichier de réponses de mot de passe à utilisation unique.

-humanReadable *humanReadable*
Indique si l'affichage lisible par l'utilisateur de la fédération avec un mot de passe à utilisation unique est affiché.

Exemples

Les exemples suivants présentent la syntaxe correcte pour plusieurs des tâches pouvant être effectuées avec cette commande :

Recensement de toutes les fédérations avec un mot de passe à utilisation unique existantes :

```
$AdminTask manageItfimOneTimePassword {-operation list -fimDomainName otpdomain}
```

Affichage lisible par l'utilisateur des détails d'une fédération avec un mot de passe à utilisation unique :

```
$AdminTask manageItfimOneTimePassword {-operation view -fimDomainName otpdomain
-federationName otpfed}
```

Affichage machine des détails d'une fédération avec un mot de passe à utilisation unique :

```
$AdminTask manageItfimOneTimePassword {-operation view -fimDomainName otpdomain
-federationName otpfed -humanReadable false}
```

Configuration d'une fédération avec un mot de passe à utilisation unique :

```
$AdminTask manageItfimOneTimePassword {-operation configure
-fimDomainName otpdomain
-fileId /home/user/otp_response.xml}
```

Modification d'une fédération avec un mot de passe à utilisation unique :

```
$AdminTask manageItfimOneTimePassword {-operation modify
-fimDomainName otpdomain
-federationName otpfed -fileId /home/user/otp_response.xml}
```

Annulation de la configuration d'une fédération avec un mot de passe à utilisation unique :

```
$AdminTask manageItfimOneTimePassword {-operation unconfigure
-fimDomainName otpdomain -federationName otpfed}
```

Création d'un exemple de fichier de réponses :

```
$AdminTask manageItfimOneTimePassword  {-operation createResponseFile
-fimDomainName otpdomain
-fileId /home/user/otp_response.xml}
```

Création d'un fichier de réponses basé sur une fédération avec un mot de passe à utilisation unique existante :

```
$AdminTask manageItfimOneTimePassword  {-operation createResponseFile
-fimDomainName otpdomain
-federationName otpfed -fileId /home/user/otp_response.xml}
```

Fichier de réponses de mot de passe à utilisation unique

Créez un fichier de réponses de mot de passe à utilisation unique à l'aide de la commande **manageItfimOneTimePassword** pour configurer une nouvelle fédération avec un mot de passe à utilisation unique ou modifier une fédération avec un mot de passe à utilisation unique existante. Editez-le avec les valeurs appropriées pour votre environnement.

Le fichier de réponses de mot de passe à utilisation unique est un fichier XML qui est utilisé par la commande **manageItfimOneTimePassword** pour configurer et modifier les fédérations avec un mot de passe à utilisation unique. Vous pouvez utiliser la même commande pour créer un exemple de fichier de réponses de mot de passe à utilisation unique ou un fichier de réponses de mot de passe à utilisation unique basé sur une fédération avec mot de passe à utilisation unique existante. Si vous créez un exemple de fichier de réponses de mot de passe à utilisation unique, les valeurs des paramètres dans le fichier de réponses sont remplies avec des exemples de valeur. Si vous créez un fichier de réponses de mot de passe à utilisation unique basé sur une fédération avec un mot de passe à utilisation unique existante, les valeurs des paramètres dans le fichier de réponses sont remplies avec les valeurs utilisées dans cette fédération avec mot de passe à utilisation unique. Pour plus d'informations, voir «manageItfimOneTimePassword», à la page 729.

Exemple de fichier de réponses de mot de passe à utilisation unique

Créez un exemple de fichier de réponses de mot de passe à utilisation unique à l'aide de la commande suivante :

```
$AdminTask manageItfimOneTimePassword {-operation createResponseFile
-fimDomainName otpDomain   -fileId /home/user/otp.response}
```

Fédération avec un mot de passe à utilisation unique existante

Créez un fichier de réponses qui est basé sur une fédération avec un mot de passe à utilisation unique existante à l'aide de la commande suivante :

```
$AdminTask manageItfimOneTimePassword {-operation createResponseFile
-fimDomainName otpDomain  -federationName otpFederation
-fileId /home/user/otp.response}
```

Pour des exemples de fichier de réponses, voir les répertoires suivants :

AIX, Linux ou Solaris :

```
/opt/IBM/FIM/examples/responsefiles
```

Windows :

```
C:\Program Files\IBM\FIM\examples\responsefiles
```

Paramètres

Vous devez spécifier les paramètres dans le fichier de réponses de mot de passe à utilisation unique avant qu'il puisse être utilisé par la commande **manageItfimOneTimePassword**. Les informations suivantes répertorient tous les paramètres dans le fichier de réponses de mot de passe à utilisation unique.

FedName

Nom de la fédération avec un mot de passe à utilisation unique si le fichier de réponses de mot de passe à utilisation unique est utilisé pour configurer la fédération avec un mot de passe à utilisation unique.

Nouveau nom de la fédération avec un mot de passe à utilisation unique si le fichier de réponses de mot de passe à utilisation unique est utilisé pour modifier une fédération avec un mot de passe à utilisation unique existante.

Le nom de la fédération avec un mot de passe à utilisation unique ne doit contenir que des caractères alphanumériques et ne doit pas être utilisé par une autre fédération avec un mot de passe à utilisation unique.

Obligatoire : Oui

Exemple : `otpfed`

OTPGetDelivery MethodsMappingRule

Contenu XML avec caractères d'échappement de la règle de mappage **OTPGetDeliveryMethods** .

Vous devez spécifier ce paramètre ou **OTPGetDelivery MethodsMappingRuleFileName**. Si les deux paramètres sont indiqués, c'est ce paramètre qui est utilisé.

Pour plus d'informations, voir «Règle de mappage OTPGetDeliveryMethods», à la page 716.

Obligatoire :
Oui lorsque le paramètre **OTPGetDelivery MethodsMappingRuleFileName** n'est pas spécifié.
Non lorsque le paramètre **OTPGetDelivery MethodsMappingRuleFileName** est spécifié.

Exemple :
Voir $FIM_INSTALL_DIR$/examples/js_mappings/otp_get_delivery_methods.js.

OTPGetDelivery MethodsMapping RuleFileName

Nom du fichier contenant la règle de mappage **OTPGetDelivery Methods**.

Vous devez spécifier ce paramètre ou **OTPGetDelivery MethodsMappingRule**. Si les deux paramètres sont indiqués, ce paramètre n'est pas utilisé.

Pour plus d'informations, voir «Règle de mappage OTPGetDeliveryMethods», à la page 716.

Obligatoire : Oui, lorsque le paramètre **OTPGetDelivery MethodsMappingRule** n'est pas spécifié.
Non, lorsque le paramètre **OTPGetDelivery MethodsMappingRule** est spécifié.

Exemple : /home/user/otp_get_delivery_methods.js

OTPGetDelivery MethodsMapping RuleType

Type de la règle de mappage **OTPGetDelivery Methods**.

Le type doit avoir la valeur JAVASCRIPT ou XSLT.

Pour plus d'informations, voir «Règle de mappage OTPGetDeliveryMethods», à la page 716.

Obligatoire : Oui

Exemple : JAVASCRIPT

OTPGenerate MappingRule

Contenu XML avec caractères d'échappement de la règle de mappage **OTPGenerate**.

Vous devez spécifier ce paramètre ou **OTPGenerateMapping RuleFileName**. Si les deux paramètres sont indiqués, c'est ce paramètre qui est utilisé.

Pour plus d'informations, voir «Règle de mappage OTPGenerate», à la page 716.

Obligatoire :
Oui lorsque le paramètre **OTPGenerateMapping RuleFileName** n'est pas spécifié.
Non lorsque le paramètre **OTPGenerateMapping RuleFileName** est spécifié.

Exemple :
Voir $FIM_INSTALL_DIR$/examples/js_mappings/otp_generate.js.

OTPGenerate MappingRuleFileName

Nom du fichier contenant la règle de mappage **OTPGenerate**.

Vous devez spécifier ce paramètre ou le paramètre **OTPGenerate MappingRule**. Si les deux paramètres sont indiqués, ce paramètre n'est pas utilisé.

Pour plus d'informations, voir «Règle de mappage OTPGenerate», à la page 716.

Obligatoire : Oui, lorsque le paramètre **OTPGenerate MappingRule** n'est pas spécifié.
Non, lorsque le paramètre **OTPGenerate MappingRule** est spécifié.

Exemple : `/home/user/otp_generate.js`

OTPGenerate MappingRuleType

Type de la règle de mappage **OTPGenerate**.

Le type doit avoir la valeur `JAVASCRIPT` ou `XSLT`.

Pour plus d'informations, voir «Règle de mappage OTPGenerate», à la page 716.

Obligatoire : Oui

Exemple : `JAVASCRIPT`

OTPDeliverMappingRule

Contenu XML avec caractères d'échappement de la règle de mappage **OTPDeliver**.

Vous devez spécifier ce paramètre ou le paramètre **OTPDeliver MappingRuleFileName**. Si les deux paramètres sont indiqués, c'est ce paramètre qui est utilisé.

Pour plus d'informations, voir «Règle de mappage OTPDeliver», à la page 715.

Obligatoire : Oui, lorsque le paramètre **OTPDeliver MappingRuleFileName** n'est pas spécifié.
Non, lorsque le paramètre **OTPDeliver MappingRuleFileName** est spécifié.

Exemple :
Voir `$FIM_INSTALL_DIR$/examples/js_mappings/otp_deliver.js`.

OTPDeliver MappingRuleFileName

Nom du fichier contenant la règle de mappage **OTPDeliver**.

Vous devez spécifier ce paramètre ou le paramètre **OTPDeliver MappingRule**. Si les deux paramètres sont indiqués, ce paramètre n'est pas utilisé.

Pour plus d'informations, voir «Règle de mappage OTPDeliver», à la page 715.

Obligatoire : Oui, lorsque le paramètre **OTPDeliver MappingRule** n'est pas spécifié.
Non lorsque le paramètre **OTPDeliver MappingRule** est spécifié.

Exemple : /home/user/otp_deliver.js

OTPDeliver MappingRuleType

Type de la règle de mappage **OTPDeliver**.

Le type doit avoir la valeur JAVASCRIPT ou XSLT.

Pour plus d'informations, voir «Règle de mappage OTPDeliver», à la page 715.

Obligatoire : Oui

Exemple : JAVASCRIPT

OTPVerifyMappingRule

Contenu XML avec caractères d'échappement de la règle de mappage **OTPVerify**.

Vous devez spécifier ce paramètre ou le paramètre **OTPVerify MappingRuleFileName**. Si les deux paramètres sont indiqués, c'est ce paramètre qui est utilisé.

Pour plus d'informations, voir «Règle de mappage OTPVerify», à la page 718.

Obligatoire : Oui, lorsque le paramètre **OTPVerify MappingRuleFileName** n'est pas spécifié.
Non, lorsque le paramètre **OTPVerify MappingRuleFileName** est spécifié.

Exemple :
Voir $FIM_INSTALL_DIR$/examples/js_mappings/otp_verify.js.

OTPVerifyMapping RuleFileName

Nom du fichier contenant la règle de mappage **OTPVerify**.

Vous devez spécifier ce paramètre ou le paramètre **OTPVerify MappingRule**. Si les deux paramètres sont indiqués, ce paramètre n'est pas utilisé.

Pour plus d'informations, voir «Règle de mappage OTPVerify», à la page 718.

Obligatoire : Oui, lorsque le paramètre **OTPVerify MappingRule** n'est pas spécifié.
Non lorsque le paramètre **OTPVerify MappingRule** est spécifié.

Exemple : /home/user/otp_verify.js

OTPVerifyMapping RuleType

Type de la règle de mappage **OTPVerify**.

Le type doit avoir la valeur JAVASCRIPT ou XSLT.

Pour plus d'informations, voir «Règle de mappage OTPVerify», à la page 718.

Obligatoire : Oui

Exemple : JAVASCRIPT

OTPTypesTo
OTPProviderModuleIds

Liste de mappages entre le type de mot de passe à utilisation unique et les ID du module fournisseur de mot de passe à utilisation unique.

Chaque mappage spécifie le plug-in du fournisseur de mot de passe à utilisation unique qui génère et vérifie le mot de passe à utilisation unique pour les utilisateurs ayant le type de mot de passe à utilisation unique spécifié.

Chaque utilisateur peut être associé à un type de mot de passe à utilisation unique. L'ID du module fournisseur de mot de passe à utilisation unique est un ID d'extension du plug-in du fournisseur de mot de passe à utilisation unique.

Requis : Non

Exemple : voir OTPTypesToOTPProviderModuleIds.

DeliveryTypesTo
OTPDeliveryModuleIds

Liste de mappages entre le type de livraison et les ID du module de livraison de mot de passe à utilisation unique.

Chaque mappage spécifie le plug-in de livraison de mot de passe à utilisation unique qui livre le mot de passe à utilisation unique pour les utilisateurs possédant le type de livraison spécifié.

Chaque utilisateur peut être associé au type de livraison. L'ID du module de livraison de mot de passe à utilisation unique est un ID d'extension du plug-in de livraison de mot de passe à utilisation unique.

Requis : Non

Exemple : voir DeliveryTypesToOTPDeliveryModuleIds.

OTPProvider
ModuleConfigs

Liste de mappages entre le type de mot de passe à utilisation unique ou les ID de module de fournisseur de mot de passe à utilisation unique et des configurations.

Le type de mot de passe à utilisation unique ou l'ID de module de fournisseur de mot de passe à utilisation unique doit correspondre à un type de mot de passe à utilisation unique ou à un ID de module de fournisseur de mot de passe à utilisation unique qui est spécifié dans la propriété **OTPTypesTo OTPProviderModuleIds**. Chaque configuration est un mappage entre le nom du paramètre et les valeurs de paramètre.

Chaque mappage spécifie la configuration du module fournisseur de mot de passe à utilisation unique spécifié.

Requis : Non

Exemple : voir OTPProviderModuleConfigs.

OTPDelivery
ModuleConfigs

Liste de mappages entre le type de livraison ou les ID de module de livraison de mot de passe à utilisation unique et des configurations.

Le type de livraison ou l'ID de module de livraison de mot de passe à utilisation unique doit correspondre à un type de livraison ou à un ID de module de livraison de mot de passe à utilisation unique qui est spécifié dans la propriété **OTPTypesTo OTPDeliveryModuleIds**. Chaque configuration est un mappage entre le nom du paramètre et les valeurs de paramètre.

Chaque mappage spécifie la configuration du module de livraison de mot de passe à utilisation unique spécifié.

Requis : Non

Exemple : voir OTPDeliveryModuleConfigs.

Paramètres sensibles

Certaines configurations du module fournisseur de mot de passe à utilisation unique et du module de livraison de mot de passe à utilisation unique peuvent être sensibles.

Un exemple est la configuration `SMTPPassword` du module de livraison de mot de passe à utilisation unique `EmailOTPDelivery`. Vous pouvez déclarer ces configurations comme étant sensibles en ajoutant au nom du paramètre l'annotation `@Sensitive`.

Tivoli Federated Identity Manager masque toutes les configurations déclarées sensibles avant de les stocker dans les fichiers de configuration Tivoli Federated Identity Manager.

Lorsque vous consultez une fédération avec mot de passe à utilisation unique, les configurations sensibles sont affichées sous forme masquée. Lorsque vous créez un fichier de réponses à utilisation unique qui repose sur une fédération avec mot de passe à utilisation unique existante, les configurations sensibles sont affichées sous forme masquée.

Pour obtenir un exemple de configuration sensible, voir OTPDeliveryModuleConfigs.

Exemples

- **Paramètre OTPTypesToOTPProviderModuleIds**

  L'exemple suivant indique la valeur du paramètre **OTPTypesToOTPProviderModuleIds** :

  ```
  <object class="java.util.HashMap">
        <void method="put">
          <string>mac_otp</string>
          <string>MobileAuthCodeOTPModule</string>
        </void>
   </object>
  ```

  Cet exemple contient un seul mappage. Il présente le mappage entre le type de mot de passe à utilisation unique `mac_otp` et l'ID du module fournisseur de mot de passe à utilisation unique `MobileAuthCodeOTPModule`.

-

Paramètre DeliveryTypesToOTPDeliveryModuleIds

L'exemple suivant indique la valeur du paramètre **DeliveryTypesToOTPDeliveryModuleIds** :

```
<object class="java.util.HashMap">
      <void method="put">
         <string>sms_delivery</string>
         <string>SMSOTPDelivery</string>
      </void>
      <void method="put">
        <string>mail_delivery</string>
        <string>EmailOTPDelivery</string>
      </void>
 </object>
```

Cet exemple contient deux mappages.

- Le premier mappage est un mappage entre le type de livraison `sms_delivery` et l'ID du module de livraison de mot de passe à utilisation unique `SMSOTPDelivery`.
- Le deuxième mappage est un mappage entre le type de livraison `mail_delivery` et l'ID du module de livraison de mot de passe à utilisation unique `EmailOTPDelivery`.

•

Paramètre OTPProviderModuleConfigs

L'exemple suivant indique la valeur du paramètre **OTPProviderModuleConfigs** :

```
<object class="java.util.HashMap">
      <void method="put">
        <string>mac_otp</string>
        <object class="java.util.HashMap">
          <void method="put">
            <string>StoreEntryFactoryHashAlgorithm</string>
            <object class="java.util.ArrayList">
              <void method="add">
                <string>SHA-256</string>
              </void>
            </object>
          </void>
          <void method="put">
            <string>GeneratorCharacterSet</string>
            <object class="java.util.ArrayList">
              <void method="add">
                <string>0123456789</string>
              </void>
            </object>
          </void>
        </object>
      </void>
    </object>
```

Cet exemple contient un mappage. Il s'agit du mappage entre le type de mot de passe à utilisation unique `mac_otp` et sa configuration. La configuration contient deux paramètres. Le nom du premier paramètre est **StoreEntryFactoryHashAlgorithm**. Il possède une seule valeur de paramètre : *SHA-256*. Le deuxième paramètre est **GeneratorCharacterSet**. Il possède une seule valeur de paramètre : *0123456789*.

•

Paramètre OTPDeliveryModuleConfigs

L'exemple suivant indique la valeur du paramètre **OTPDeliveryModuleConfigs** :

```
<object class="java.util.HashMap">
      <void method="put">
        <string>mail_delivery</string>
        <object class="java.util.HashMap">
          <void method="put">
            <string>@Sensitive SMTPPassword</string>
            <object class="java.util.ArrayList">
              <void method="add">
                <string>password</string>
              </void>
            </object>
          </void>
        </object>
      </void>
      <void method="put">
        <string>sms_delivery</string>
        <object class="java.util.HashMap">
          <void method="put">
            <string>HTTPParameters</string>
            <object class="java.util.ArrayList">
              <void method="add">
                <string>From=+15127828860</string>
              </void>
              <void method="add">
                <string>To=$DEST_NO$</string>
              </void>
              <void method="add">
                <string>Message=$MSG$</string>
              </void>
            </object>
          </void>
        </object>
      </void>
    </object>
```

Cet exemple contient deux mappages.

- Le premier mappage est un mappage entre le type de livraison **mail_delivery** et sa configuration. La configuration ne contient qu'un paramètre. Le nom de ce paramètre est **SMTPPassword**. Il possède une seule valeur de paramètre *password*. Ce paramètre est un paramètre sensible car il porte l'annotation `@Sensitive`.
- Le deuxième mappage est un mappage entre le type de livraison **sms_delivery** et sa configuration. La configuration ne contient qu'un paramètre. Le nom de ce paramètre est **HTTPParameters**. Il est associé à trois valeurs, *From=+15127828860*, *To=$DEST_NO$* et *Message=MSG*.

manageItfimPointOfContact

Utilisez la commande **manageItfimPointOfContact** pour gérer un profil de point de contact personnalisé pour un domaine spécifique.

Actions

La commande **manageItfimPointOfContact** permet d'effectuer les opérations suivantes sur un profil de point de contact lorsqu'elle est utilisée avec les paramètres appropriés :

- list

- listCallbacks
- create (à l'aide d'un fichier de réponses)
- createResponseFile
- view
- activate

Syntaxe

La syntaxe de la commande est la suivante :

```
$AdminTask manageItfimPointOfContact {-operation opérateur
  -fimDomainName nom [options]}
```

où le paramètre `-operation` et sa valeur *opérateur* ainsi que le paramètre `-fimDomainName` et sa valeur *nom* sont obligatoires. Paramètres facultatifs :

```
-uuid ID
-signInCallbackIds rappel1,rappel2
-signOutCallbackIds rappel1,rappel2
-locaIdCallbackIds rappel1,rappel2
-authenticationCallbackIds rappel1,rappel2
-authenticationPolicyCallbackIds rappel1,rappel2
-fileId fichier_sortie | fichier_entrée
```

L'utilisation de ces paramètres dépend de l'opérateur que vous avez choisi.

Paramètres

Les paramètres suivants peuvent être utilisés avec la commande **manageItfimPointOfContact** :

-operation *opérateur*
Paramètre requis. La valeur utilisée avec ce paramètre indique l'opération à effectuer sur le domaine. Les valeurs valides sont répertoriées dans le tableau suivant.

Tableau 155. Valeurs du paramètre manageItfimPointOfContact **-operation**

Valeur	Description et conditions requises
view	Affiche les propriétés d'un profil de point de contact ainsi que ses rappels. Si vous utilisez cet opérateur, vous devez aussi utiliser les paramètres suivants : **uuid** *ID* Identificateur universel unique du profil de point de contact existant. Vous pouvez déterminer l'**identificateur unique universel** des profils de point de contact existants en exécutant l'opération de listage, comme décrit précédemment.

Tableau 155. Valeurs du paramètre manageItfimPointOfContact **-operation** *(suite)*

Valeur	Description et conditions requises
activate	Active un profil de point de contact spécifique. Si vous utilisez cet opérateur, vous devez aussi utiliser les paramètres suivants : **uuid** *ID* Identificateur universel unique du profil de point de contact existant. Vous pouvez déterminer l'**identificateur unique universel** des profils de point de contact existants en exécutant l'opération de listage, comme décrit précédemment. **Remarque :** Si vous activez un profil WebSphere, les propriétés par défaut suivantes sont utilisées : SOAP Port=9444 Authorization type=Allow Authenticated users to access SOAP endpoints Authentication type=Basic Si votre environnement exige des paramètres différents, vous devez transmettre un fichier texte contenant les paramètres appropriés.
delete	Supprime un profil de point de contact personnalisé spécifique. **Remarque :** Vous ne pouvez pas supprimer un profil de point de contact par défaut. Vous ne pouvez supprimer que les points de contact que vous avez créés. Les profils de points de contact par défaut sont définis en lecture seule pour éviter toute suppression accidentelle. Si vous utilisez cet opérateur, vous devez aussi utiliser les paramètres suivants : **uuid** *ID* Identificateur universel unique du profil de point de contact existant. Vous pouvez déterminer l'**identificateur unique universel** des profils de point de contact existants en exécutant l'opération de listage, comme décrit précédemment.
list	Liste tous les profils de point de contact existants dans un domaine particulier.
listCallbacks	Liste les rappels activés dans un domaine.

Tableau 155. Valeurs du paramètre manageItfimPointOfContact **-operation** *(suite)*

Valeur	Description et conditions requises
createResponseFile	Crée un fichier de réponses que vous pouvez utiliser pour créer un profil de point de contact. Vous pouvez créer un fichier de réponses pour un nouveau profil de point de contact. Vous pouvez également créer un fichier de réponses basé sur un profil de point de contact existant. Nouveau profil de point de contact : Si vous utilisez cet opérateur pour créer un fichier de réponses pour un nouveau profil de point de contact, vous devez également indiquer les paramètres suivants : **signInCallbackIds** *rappel1,rappel2* **signOutCallbackIds** *rappel1,rappel2* **localIdCallbackIds** *rappel1,rappel2* **authenticationCallbackIds** *rappel1,rappel2* **authenticationPolicyCallbackIds***rappel1,rappel2* **fileId** *fichier_sortie* Indiquez le nom et le chemin du fichier de réponses qui sera créé à l'aide de cette commande. Basé sur un profil de point de contact existant : Si vous utilisez cet opérateur pour créer un fichier de réponses qui est basé sur un profil de point de contact existant, vous devez également indiquer les paramètres suivants : **uuid** *ID* Identificateur universel unique du profil de point de contact existant. Vous pouvez déterminer l'**identificateur unique universel** des profils de point de contact existants en exécutant l'opération de listage, comme décrit précédemment. **fileId** *fichier_sortie* Indiquez le nom et le chemin du fichier de réponses qui sera créé à l'aide de cette commande. Après avoir créé un fichier de réponses, ouvrez-le dans un éditeur de texte. Passez en revue les attributs qui y sont définis, effectuez les modifications requises par votre environnement puis sauvegardez et fermez le fichier. Pour plus d'informations sur le contenu du fichier de réponses, voir «Fichier de réponses du serveur point de contact», à la page 746.
create	Crée un profil de point de contact à l'aide d'un fichier de réponses. Si vous utilisez cet opérateur, vous devez aussi utiliser les paramètres suivants : **fileId** *nom_entrée* Ce paramètre indique le nom et le chemin du fichier de réponses que vous utilisez comme entrée. Vous pouvez créer le fichier de réponses à l'aide de l'opérateur `createResponseFile`.

-fimDomainName *nom*
Paramètre requis. La valeur utilisée pour ce paramètre correspond au nom du domaine sur lequel l'opération est effectuée. Ce nom peut correspondre à une chaîne composée de caractères de n'importe quel type.

-uuid *ID*
Un identificateur sous forme de chaîne identifie de manière unique la ressource sur laquelle vous souhaitez effectuer une opération.

-signInCallbackIds *rappel*
Liste de rappels séparée par des virgules qui sera utilisée pour les actions de connexion du point de contact.

-signOutCallbackIds *rappel*
Liste de rappels séparée par des virgules qui sera utilisée pour les actions de déconnexion du point de contact.

-localIdCallbackIds *rappel*
Liste de rappels séparée par des virgules qui sera utilisée comme ID local par le point de contact.

-authenticationCallbackIds *rappel*
Liste de rappels séparée par des virgules qui sera utilisée pour l'authentification du point de contact.

-authenticationPolicyCallbackIds *rappel*
Liste de rappels séparée par des virgules qui sera utilisée pour déterminer la règle d'authentification.

-fileId *fichier_sortie* | *fichier_entrée*
Ce paramètre est obligatoire si vous créez un fichier de réponses ou un profil de point de contact. La valeur utilisée pour ce paramètre correspond au nom et au chemin d'accès du fichier de réponses à partir duquel les données seront lues (fichier d'entrée) ou dans lequel elles seront écrites (fichier de sortie). Le chemin d'accès et le nom de fichier doivent être valides pour le système d'exploitation utilisé.

Exemples

Les exemples suivants présentent la syntaxe correcte pour plusieurs des tâches pouvant être effectuées avec cette commande :

Afficher les détails relatifs à un point de contact :
```
$AdminTask manageItfimPointOfContact {-operation view
  -fimDomainName domain1
  -uuid uuid8f3d17a-0107-w712-q35b-b0c5ecc605ba}
```

Activer un point de contact :
```
$AdminTask manageItfimPointOfContact {-operation activate
  -fimDomainName domain1
  -uuid uuid8f3d17a-0107-w712-q35b-b0c5ecc605ba}
```

Supprimer un profil de point de contact personnalisé :
```
$AdminTask manageItfimPointOfContact {-operation delete
  -fimDomainName domain1
  -uuid uuid3e8de4e8-0119-1a2d-9443-c4944d126cc1}
```

Répertorier tous les profils de point de contact définis dans un domaine :
```
$AdminTask manageItfimPointOfContact {-operation list
  -fimDomainName domain1}
```

Répertorier tous les rappels activés dans le domaine :
```
$AdminTask manageItfimPointOfContact {-operation listCallbacks
  -fimDomainName domain1}
```

Créer un fichier de réponses dans le but de créer un profil de point de contact :
```
$AdminTask manageItfimPointOfContact {-operation createResponseFile
  -fimDomainName domain1
  -signInCallbackIds genericPocSignInCallback,wasPocSignInCallback
  -signOutCallbackIds genericPocSignOutCallback
  -localIdCallbackIds genericPocLocalIdentityCallback
```

```
  -authenticationCallbackIds genericPocAuthenticateCallback
  -authenticationPolicyCallbackIds genericPocAuthnPolicyCallback
  -fileId c:\home\files\temp\empty.xml}
```

Remarque : Le fichier indiqué ici correspond au nom du fichier de réponses que vous créez avec cette commande. Utilisez ce fichier comme entrée pour créer un profil de point de contact. Après avoir créé ce fichier, ouvrez-le dans un éditeur de texte et définissez les attributs qu'il contient afin de les rendre adaptés à votre environnement.

Créer un fichier de réponses basé sur un profil de point de contact existant :

```
$AdminTask manageItfimPointOfContact {-operation createResponseFile
  -fimDomainName domain1
  -uuid uuid8f3d17a-0107-w712-q35b-b0c5ecc605ba
  -fileId c:\home\files\temp\empty.xml}
```

Remarque : Le fichier indiqué ici correspond au nom du fichier de réponses que vous créez avec cette commande. Utilisez ce fichier comme entrée pour créer un profil de point de contact ou pour en modifier les propriétés. Après avoir créé ce fichier, ouvrez-le dans un éditeur de texte pour vous assurer que les attributs qu'il contient sont adaptés à votre environnement.

Créer un profil de point de contact :

Remarque : Le fichier indiqué ici correspond au fichier de réponses et il sera utilisé comme entrée. Ouvrez le fichier de réponses dans un éditeur de texte avant d'exécuter cette commande. Assurez-vous que les attributs qui sont définis dans le fichier sont appropriés à votre environnement.

```
$AdminTask manageItfimPointOfContact {-operation create
  -fimDomainName domain1
  -fileId c:\home\files\temp\empty.xml}
```

Fichier de réponses du serveur point de contact

Vous devez créer un fichier de réponses avant de créer un profil de point de contact à l'aide de la commande **manageItfimPointOfContact**, puis le modifier afin qu'il contienne les valeurs correctes pour votre environnement.

Vous pouvez créer un fichier de réponses lorsque vous créez un partenaire en exécutant la commande suivante :

Nouveau profil de point de contact

```
$AdminTask manageItfimPointOfContact {-operation createResponseFile
  -fimDomainName nom
  -uuid ID
  -fileId nom_de_fichier}
```

Profil de point de contact existant

Pour créer un fichier de réponses permettant de créer un point de contact sur la base d'un point de contact existant, exécutez la commande suivante :

```
$AdminTask manageItfimPointOfContact {-operation createResponseFile
  -fimDomainName nom
  -signInCallbackIds rappel,rappel
  -signOutCallbackIds rappel
  -localIdCallbackIds rappel
  -authenticationCallbackIds rappel
  -authenticationPolicyCallbackIds rappel
  -fileId nom_de_fichier}
```

Un fichier de réponses est créé une fois l'une de ces commandes exécutée. Le contenu du fichier varie en fonction des propriétés spécifiées dans la commande ou dans le point de contact existant.

Remarque : Vous devez procéder comme suit pour vous assurer que les propriétés personnalisées utilisées par vos rappels sont incluses :

1. Ouvrez le fichier de réponses dans un éditeur de texte.
2. Passez en revue les attributs qui y sont définis,
3. Indiquez le type de fédération à créer.
4. Sauvegardez et fermez le fichier.

Pour des exemples de fichier de réponses, voir les répertoires suivants :

AIX, Linux ou Solaris

```
/opt/IBM/FIM/examples/responsefiles
```

Windows

```
C:\Program Files\IBM\FIM\examples\responsefiles
```

Paramètres

Les descriptions ci-après présentent les types de paramètre utilisés dans les fichiers de réponses. Toutefois, les paramètres effectivement utilisés dans votre fichier de réponses XML dépendent de votre environnement et des rappels que vous utilisez. Pour obtenir un exemple de fichier de réponses, voir «Exemples», à la page 748.

Tableau 156. Paramètres utilisés dans un fichier de réponses d'un point de contact

Paramètre	Valeur	Description
profileName=	*nom*	Nom du profil du point de contact.
Description=	*texte*	Description du profil.
signIn.INDEX=	*IDrappel*	Un ou plusieurs ID de rappel sont utilisés pour la connexion. Le paramètre INDEX représente l'ordre d'appel de ce rappel dans la chaîne de connexion. Il commence par 1. L'ID de rappel identifie le module de rappel appelé.
CALLBACKID.PROPERTYNAME=	*valeur*	Désigne un module de rappel et indique qu'une propriété est utilisée avec ce module.
signOut.INDEX=	*IDrappel*,*IDrappel*,	Un ou plusieurs ID de rappel sont utilisés pour la déconnexion. Le paramètre INDEX représente l'ordre d'appel de ce rappel dans la chaîne de connexion. Il commence par 1. L'ID de rappel identifie le module de rappel appelé.
CALLBACKID.PROPERTYNAME=	*valeur*	Désigne un module de rappel et indique qu'une propriété est utilisée avec ce module.

Tableau 156. Paramètres utilisés dans un fichier de réponses d'un point de contact (suite)

Paramètre	Valeur	Description
localId.INDEX=	*IDrappel*,*IDrappel*,	Un ou plusieurs ID de rappel sont utilisés pour l'ID local. Le paramètre INDEX représente l'ordre d'appel de ce rappel dans la chaîne de connexion. Il commence par 1. L'ID de rappel identifie le module de rappel appelé.
CALLBACKID.PROPERTYNAME=	*valeur*	Désigne un module de rappel et indique qu'une propriété est utilisée avec ce module.
authentication.INDEX=	*IDrappel*,*IDrappel*,	Un ou plusieurs ID de rappel sont utilisés pour l'authentification. Le paramètre INDEX représente l'ordre d'appel de ce rappel dans la chaîne d'authentification. Il commence par 1. L'ID de rappel identifie le module de rappel appelé.
CALLBACKID.PROPERTYNAME=	*valeur*	Désigne un module de rappel et indique qu'une propriété est utilisée avec ce module.
authnpolicy.INDEX=	*IDrappel*,*IDrappel*,	Un ou plusieurs ID de rappel sont utilisés pour déterminer la règle d'authentification. Le paramètre INDEX représente l'ordre d'appel de ce rappel dans la chaîne de détermination de règle d'authentification. Il commence par 1. L'ID de rappel identifie le module de rappel appelé.
CALLBACKID.PROPERTYNAME=	*valeur*	Désigne un module de rappel et indique qu'une propriété est utilisée avec ce module.

Remarque : Si le nom de la propriété de rappel se termine par `MappingRuleFileName`, le contenu du fichier est téléchargé en tant que règle de mappage. Le nom de la propriété de l'élément `config` est le texte suivant le suffixe `MappingRuleFileName`. Par exemple, pour créer une propriété nommée `authentication.policy.map.rule`, la propriété sur le fichier de réponses doit être nommé `CALLBACKID.authentication.policy.map.ruleMappingRuleFileName`.

Exemples

Exemple de commande : L'exemple suivant montre comment utiliser la commande de création et indiquer le fichier de réponses :

```
$AdminTask manageItfimPointOfContact {-operation create -fimDomainName domain1
  -fileId c:\home\files\temp\POCprops.xml}
```

Exemple de fichier de réponses : L'exemple suivant présente un fichier de réponses qui peut être utilisé avec l'opération d'activation.

En examinant cet exemple, n'oubliez pas qu'un profil de point de contact est doté d'une hiérarchie comportant les types de rappel suivants :

```
0 ou 4 types de rappel
  chaque type de rappel contient 1 ou plusieurs rappels classés par ordre
   chaque rappel contient 0 ou plusieurs propriétés arbitraires
```

Par exemple :

```
signIn.INDEX=CALLBACKID
  CALLBACKID.PROPERTYNAME1=value
  CALLBACKID.PROPERTYNAME2=value
```

Remarque : INDEX correspond à un chiffre représentant le numéro d'ordre de l'ID de rappel pour ce type spécifique (`signIn` dans l'exemple). Ainsi, il est possible d'ajouter des propriétés à l'ID de rappel en le préfixant au nom de propriété. L'interface de ligne de commande décompose la réponse, ajoute les propriétés au rappel et les attribue au type approprié suivant l'ordre défini. Le fichier de réponses n'a pas l'aspect d'une paire `key=value` au format XML, mais est en fait identique.

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.5.0" class="java.beans.XMLDecoder">
 <object class="java.util.HashMap">
  <void method="put">
   <string>signIn.1</string>
   <object class="java.util.ArrayList">
    <void method="add">
     <string>wasPocSignInCallback</string>
    </void>
   </object>
  </void>
 <void method="put">
   <string>authnpolicy.1</string>
   <object class="java.util.ArrayList">
    <void method="add">
     <string>genericPocAuthnPolicyCallback</string>
    </void>
   </object>
  </void>
  <void method="put">
   <string>wasPocAuthenticateCallback.authentication.macros</string>
   <object class="java.util.ArrayList">
    <void method="add">
     <string>%FEDID%</string>
    </void>
    <void method="add">
     <string>%FEDNAME%</string>
    </void>
    <void method="add">
     <string>%PARTNERID%</string>
    </void>
    <void method="add">
     <string>%ACSURL%</string>
    </void>
    <void method="add">
     <string>%SSOREQUEST%</string>
    </void>
    <void method="add">
     <string>%TARGET%</string>
    </void>
   </object>
  </void>
  <void method="put">
   <string>profileName</string>
   <object class="java.util.ArrayList">
    <void method="add">
     <string>testwaspoc</string>
    </void>
   </object>
  </void>
  <void method="put">
   <string>profileDescription</string>
   <object class="java.util.ArrayList">
    <void method="add">
```

```
    <string>WebSphere Point of Contact Profile</string>
   </void>
  </object>
 </void>
 <void method="put">
  <string>localId.1</string>
  <object class="java.util.ArrayList">
   <void method="add">
    <string>wasPocLocalIdentityCallback</string>
   </void>
  </object>
 </void>
 <void method="put">
  <string>signOut.1</string>
  <object class="java.util.ArrayList">
   <void method="add">
    <string>wasPocSignOutCallback</string>
   </void>
  </object>
 </void>
 <void method="put">
  <string>authentication.1</string>
  <object class="java.util.ArrayList">
   <void method="add">
    <string>wasPocAuthenticateCallback</string>
   </void>
  </object>
 </void>
</object>
</java>
```

Référence du plug-in du fournisseur de mot de passe à utilisation unique

Le plug-in du fournisseur de mot de passe à utilisation unique génère et valide les mots de passe à utilisation unique. Configurez le plug-in du fournisseur de mot de passe à utilisation unique afin qu'il puisse être utilisé par Tivoli Federated Identity Manager.

Tivoli Federated Identity Manager fournit trois plug-in de fournisseur de mot de passe à utilisation unique :

- «MobileAuthCodeOTPModule», à la page 751
- «TOTPModule», à la page 752
- «HOTPModule », à la page 754

`MobileAuthCodeOTPModule` génère le mot de passe à utilisation unique en sélectionnant les caractères un à un et de manière aléatoire à partir du jeu de caractères configuré, jusqu'à atteindre le nombre de caractères défini. `MobileAuthCodeOTPModule` stocke également le mot de passe à utilisation unique généré dans le plug-in du magasin de mots de passe à utilisation unique configuré. Le mot de passe à utilisation unique est soumis au sel de cryptage et haché avant d'être stocké dans le plug-in de ce magasin.

`TOTPModule` génère un mot de passe à utilisation unique à l'aide d'un algorithme spécifié comprenant une application de mot de passe à utilisation unique basée sur la durée. Les mots de passe ne sont pas communiqués ou stockés, mais la correspondances entre le serveur et le client est vérifiée car ils sont régénérés à intervalle régulier.

HOTPModule génère un mot de passe à utilisation unique à l'aide d'un algorithme spécifié comprenant une application de mot de passe à utilisation unique basée sur compteur. Les mots de passe ne sont pas communiqués ou stockés, mais la correspondance incrémentielle entre le serveur et le client est vérifiée.

MobileAuthCodeOTPModule

Vous trouverez ci-après la liste de toutes les configurations de MobileAuthCodeOTPModule. Toutes les configurations sont facultatives.

StoreModuleId
ID extension du plug-in du magasin de mots de passe à utilisation unique dans lequel est stocké le mot de passe à utilisation unique.

La valeur par défaut de StoreModuleId, qui est OTPProviderDynaCacheOTPStore, est utilisée lorsque cette configuration n'est pas spécifiée ou lorsqu'elle fait référence à un plug-in du magasin de mots de passe à utilisation unique qui n'existe pas.

Requis : Non

Valeurs multiples : Non

Exemple : OTPProviderDynaCacheOTPStore

StoreEntryLifetime
Durée de vie du mot de passe à utilisation unique qui est stocké dans le plug-in du magasin de mots de passe à utilisation unique. La durée de vie est exprimée en secondes.

La valeur par défaut de StoreEntryLifetime, qui est 300, est utilisée lorsque cette configuration n'est pas spécifiée ou lorsque sa valeur est inférieure à zéro.

Requis : Non

Valeurs multiples : Non

Exemple: 300

GeneratorCharacterSet
Jeu de caractères à partir duquel les caractères du mot de passe à utilisation unique sont générés.

La valeur par défaut de GeneratorCharacterSet, qui est 0123456789, est utilisée lorsque cette configuration n'est pas spécifiée ou lorsque sa valeur est vide.

Requis : Non

Valeurs multiples : Non

Exemple : 0123456789

GeneratorLength
Longueur des caractères dans le mot de passe à utilisation unique.

La valeur par défaut de GeneratorLength, qui est 8, est utilisée lorsque cette configuration n'est pas spécifiée ou lorsque sa valeur est inférieure à un.

Requis : Non

Valeurs multiples : Non

Exemple : 8

StoreEntryFactoryHashAlgorithm

Algorithme de hachage qui procède au hachage du mot de passe à utilisation unique avant son stockage dans le plug-in du magasin de mots de passe à utilisation unique.

La valeur par défaut de `StoreEntryFactoryHashAlgorithm`, qui est `SHA-256`, est utilisée lorsque cette configuration n'est pas spécifiée ou lorsqu'elle n'est pas prise en charge par l'interface de programme d'application Java Cryptography Extensions (JCE).

Pour connaître la liste des algorithmes de hachage pris en charge, voir l'*annexe A* du document Java Cryptography Architecture(JCA) API Specification & Reference.

Requis : Non

Valeurs multiples : Non

Exemple : `SHA-256`

StoreEntryFactorySaltLength

Longueur du sel de cryptage généré de façon aléatoire qui sert au hachage du mot de passe à utilisation unique avant son stockage dans le plug-in du magasin de mots de passe à utilisation unique.

La valeur par défaut de `StoreEntryFactorySaltLength`, qui est `5`, est utilisée lorsque cette configuration n'est pas spécifiée ou que sa valeur est inférieure à un.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `5`

TOTPModule

La liste suivante décrit toutes les configurations de `TOTPModule`.

OTPLength

Longueur des mots de passe à utilisation unique générés, qui peut être comprise entre 6 et 9 caractères ou nombres.

La valeur par défaut de `OTPLength`, qui est `6`, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `6`

OTPTimeSkewIntervals

Intervalles de décalage de l'algorithme. Les intervalles de décalage prennent compte des éventuels retards de synchronisation entre le serveur et le client qui génère le mot de passe à utilisation unique. Par exemple, un intervalle de décalage de 2 signifie qu'un mot de passe à utilisation unique dans 1 ou 2 intervalles passés ou 2 mots de passe à utilisation unique dans un futur intervalle sont valides. Par exemple, s'il s'agit de l'intervalle 563, et que les intervalles sont de 30 secondes, les mots de passe à utilisation unique pour les intervalles 561 à 565 sont calculés et vérifiés dans un intervalle de 2,5 minutes.

La valeur par défaut de `OTPTimeSkewIntervals`, qui est `1`, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `1`

`OTPGenerationAlgorithm`

Algorithme utilisé pour générer le mot de passe à utilisation unique. Les options valides comprennent les algorithmes suivants : `HmacSHA1`, `HmacSHA256` ou `HmacSHA512`

Remarque : Tous les algorithmes ne sont pas pris en charge par tous les niveaux Java. Par exemple, Java 1.4 prend uniquement en charge `HmacSHA1`. Pour connaître la liste des algorithmes de hachage pris en charge pour votre version Java, voir l'*Annexe A* du document Java Cryptography Architecture (JCA) API Specification & Reference pour Java 1.4 ou Java Cryptography Architecture (JCA) API Specification & Reference pour Java 1.5.

La valeur par défaut de `OTPGenerationAlgorithm`, qui est `HmacSHA1`, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `HmacSHA1`

`OTPGenerationIntervalSeconds`

Durée, en secondes, d'un intervalle. Ce nombre détermine la durée pendant laquelle un mot de passe à utilisation unique est actif avant qu'un autre mot de passe à utilisation unique soit généré.

La valeur par défaut de `OTPGenerationIntervalSeconds`, qui est `30`, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `30`

`OneTimeUseEnforcementEnabled`

Indique s'il faut placer les mots de passe à utilisation unique en mémoire cache s'ils sont utilisés pour une connexion réussie. Si cette option est définie sur `true`, la réutilisation d'un mot de passe à utilisation unique n'est pas possible tant que ce dernier est placé dans le cache.

La valeur par défaut de `OneTimeUseEnforcementEnabled`, qui est `true`, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `true`

`OneTimeUseEnforcementStore`

Cache d'objet à utiliser pour la mise en cache des mots de passe à utilisation unique ayant abouti. Ce stockage n'est pas utilisé si `OneTimeEnforcementEnabled` est défini sur `false`.

La valeur par défaut de OneTimeUseEnforcementStore, qui est OTPProviderDynaCacheOTPStore, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : OTPProviderDynaCacheOTPStore

OTPSecretKeyAttributeName
Nom de l'attribut extrayant la clé confidentielle utilisateur pour la génération d'une valeur de mot de passe à utilisation unique. Toute valeur de chaîne unique identifiant l'attribut est valide.

La valeur par défaut de otp.hmac.secret.key est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : otp.hmac.secret.key

OTPSecretKeyAttributeNamespace
Espace de nom de l'attribut extrayant la clé confidentielle utilisateur pour la génération d'une valeur de mot de passe à utilisation unique. Toute valeur de chaîne identifiant la source de l'attribut est valide. Les valeurs nulles ne sont pas valides.

La valeur par défaut de urn:ibm:security:otp:hmac est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : urn:ibm:security:otp:hmac

HOTPModule

La liste suivante décrit toutes les configurations de HOTPModule.

OTPLength
Longueur des mots de passe à utilisation unique générés, qui peut être comprise entre 6 et 9 caractères ou nombres.

La valeur par défaut de OTPLength, qui est 6, est utilisée lorsque cette configuration n'est pas spécifiée.

Requis : Non

Valeurs multiples : Non

Exemple : 6

OTPGenerationAlgorithm
Algorithme utilisé pour générer le mot de passe à utilisation unique. Les options valides comprennent les algorithmes suivants : HmacSHA1, HmacSHA256 ou HmacSHA512

Remarque : Tous les algorithmes ne sont pas pris en charge par tous les niveaux Java. Par exemple, Java 1.4 prend uniquement en charge HmacSHA1. Pour connaître la liste des algorithmes de hachage pris en charge pour votre version Java, voir l'*Annexe A* du document Java Cryptography

Architecture (JCA) API Specification & Reference pour Java 1.4 ou Java Cryptography Architecture (JCA) API Specification & Reference pour Java 1.5.

La valeur par défaut de `OTPGenerationAlgorithm`, qui est `HmacSHA1`, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `HmacSHA1`

MaxCounterLookahead

Nombre d'incrémentations du compteur nécessaires pour savoir si le mot de passe à utilisation unique est valide avant l'arrêt. Tous les nombres non négatifs sont valides.

La valeur par défaut de `MaxCounterLookahead`, qui est `25`, est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `25`

OTPSecretKeyAttributeName

Nom de l'attribut extrayant la clé confidentielle utilisateur pour la génération d'une valeur de mot de passe à utilisation unique. Toute valeur de chaîne unique identifiant l'attribut est valide.

La valeur par défaut de `otp.hmac.secret.key` est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `otp.hmac.secret.key`

OTPSecretKeyAttributeNamespace

Espace de nom de l'attribut extrayant la clé confidentielle utilisateur pour la génération d'une valeur de mot de passe à utilisation unique. Toute valeur de chaîne identifiant la source de l'attribut est valide. Les valeurs nulles ne sont pas valides.

La valeur par défaut de `urn:ibm:security:otp:hmac` est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `urn:ibm:security:otp:hmac`

OTPCounterAttributeName

Nom de l'attribut stockant et extrayant la valeur de compteur pour la génération d'une valeur de mot de passe à utilisation unique. Toute valeur de chaîne unique identifiant l'attribut est valide.

La valeur par défaut de `otp.hmac.counter` est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `otp.hmac.counter`

OTPCounterAttributeNamespace
Espace de nom de l'attribut stockant et extrayant la valeur de compteur pour la génération d'une valeur de mot de passe à utilisation unique. Toute valeur de chaîne identifiant la source de l'attribut est valide. Les valeurs nulles ne sont pas valides.

La valeur par défaut de `urn:ibm:security:otp:hmac` est utilisée lorsque cette configuration n'est pas spécifiée.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `urn:ibm:security:otp:hmac`

Référence associée:

«Fichier de réponses de mot de passe à utilisation unique», à la page 733
Créez un fichier de réponses de mot de passe à utilisation unique à l'aide de la commande **manageItfimOneTimePassword** pour configurer une nouvelle fédération avec un mot de passe à utilisation unique ou modifier une fédération avec un mot de passe à utilisation unique existante. Editez-le avec les valeurs appropriées pour votre environnement.

«Référence du plug-in de livraison de mot de passe à utilisation unique»
Le module de livraison du mot de passe à utilisation unique est un plug-in qui fournit les mots de passe à utilisation unique aux utilisateurs. Configurez le plug-in de livraison de mot de passe à utilisation unique afin qu'il puisse être utilisé par Tivoli Federated Identity Manager.

Référence du plug-in de livraison de mot de passe à utilisation unique

Le module de livraison du mot de passe à utilisation unique est un plug-in qui fournit les mots de passe à utilisation unique aux utilisateurs. Configurez le plug-in de livraison de mot de passe à utilisation unique afin qu'il puisse être utilisé par Tivoli Federated Identity Manager.

Tivoli Federated Identity Manager fournit les plug-in de livraison de mot de passe à utilisation unique suivants :

- `SMSOTPDelivery`
- `EmailOTPDelivery`
- `NoOTPDelivery`

Le plug-in `SMSOTPDelivery` fournit le mot de passe à utilisation unique par le biais d'un SMS. `SMSOTPDelivery` envoie d'abord le numéro de téléphone de l'utilisateur et le mot de passe à utilisation unique dans une demande **HTTP POST**, dont le type de contenu est `application/x-www-form-urlencoded`, à la passerelle SMS configurée. La passerelle SMS transmet ensuite le mot de passe à utilisation unique à l'utilisateur via un SMS. Tivoli Federated Identity Manager n'est livré avec aucune passerelle SMS. Vous devez configurer votre propre passerelle SMS.

Le plug-in `EmailOTPDelivery` fournit le mot de passe à utilisation unique par courrier électronique. `EmailOTPDelivery` envoie l'adresse électronique de l'utilisateur et le mot de passe à utilisation unique dans un message, dont le type `MIME` est `text/plain`, au serveur SMTP configuré. Le serveur SMTP transmet ensuite le mot de passe à utilisation unique à l'utilisateur par courrier électronique.

Tivoli Federated Identity Manager n'est livré avec aucun serveur SMTP. Vous devez configurer votre propre serveur SMTP.

Le plug-in `NoOTPDelivery` indique qu'il n'y a aucune livraison de mot de passe. Utilisez le plug-in `NoOTPDelivery` avec des applications de mot de passe à utilisation unique basé sur la durée dans lesquelles la communication et le stockage du mot de passe n'est pas nécessaire.

Vous devez configurer les plug-ins `SMSOTPDelivery` et `EmailOTPDelivery` pour qu'ils puissent être utilisés par Tivoli Federated Identity Manager. La configuration de `SMSOTPDelivery` et `EmailOTPDelivery` inclut les paramètres suivants :

Paramètres SMSOTPDelivery

`ConnectionURL`
Adresse URL de la passerelle SMS sur laquelle le numéro de téléphone de l'utilisateur et le mot de passe à utilisation unique sont envoyés.

Requis : Oui

Valeurs multiples : Non

Exemple : `https://smsgateway.tfim.example.com/`

`HTTPParameters`
Liste des paires constituées d'un nom et d'une valeur, incluse dans le corps de la demande **`HTTP POST`** vers la passerelle SMS. Dans chaque paire, le nom et la valeur doivent être séparés par le signe égal.

Tivoli Federated Identity Manager fournit deux macros, **`$DEST_NO$`** et **`$MSG$`**, qui sont remplacées par le numéro de téléphone de l'utilisateur et le contenu du SMS. Ces deux macros peuvent être utilisées uniquement en tant que valeur dans la paire constituée d'un nom et d'une valeur.

Requis : Oui

Valeurs multiples : Oui

Exemple :

- **`From=+0123456789`**
- **`To=`** *$DEST_NO$*
- **`Body=`** *MSG*

`HTTPSTruststore`
Fichier de clés Tivoli Federated Identity Manager qui valide le certificat SSL de la passerelle SMS.

Cette configuration doit être spécifiée uniquement lorsque `SMSOTPDelivery` communique avec la passerelle SMS via `HTTPS`.

Requis : Non

Valeurs multiples : Non

Exemple : `FichierClésCertifiéesParDéfaut`

`BasicAuthUserName`
Nom d'utilisateur utilisé dans l'authentification de base HTTP.

`SMSOTPDelivery` n'effectue pas l'authentification de base HTTP si cette configuration n'est pas spécifiée.

Requis : Non

Valeurs multiples : Non

Exemple : nom utilisateur

BasicAuthPassword
Mot de passe utilisé dans l'authentification de base HTTP.

SMSOTPDelivery n'effectue pas l'authentification de base HTTP si cette configuration n'est pas spécifiée.

Requis : Non

Valeurs multiples : Non

Exemple : mot de passe

ClientAuthKey
Certificat Tivoli Federated Identity Manager utilisé comme certificat client dans l'authentification client SSL. Le certificat est une paire constituée d'un fichier de clés et d'un alias. Le fichier de clés et l'alias doivent être séparés par un trait de soulignement.

SMSOTPDelivery n'effectue pas l'authentification client SSL si cette configuration n'est pas spécifiée.

Requis : Non

Valeurs multiples : Non

Exemple : FichierClésParDéfaut_clétest

SuccessHTTPReturnCode
Code de réponse provenant de la passerelle SMS indiquant que la passerelle SMS a bien traité la demande.

La valeur par défaut de SuccessHTTP
ReturnCode, qui est 200, est utilisée lorsque cette configuration n'est pas spécifiée.

Remarque : La correspondance SuccessHTTP
ReturnCode doit aboutir pour que la correspondance SuccessHTTP
ResponseBody
RegexPattern puisse être effectuée.

Requis : Non

Valeurs multiples : Non

Exemple : 200

SuccessHTTPResponseBodyRegexPattern
Ce paramètre définit le format d'expression régulière Java qui correspond au corps de la réponse HTTP renvoyé par la passerelle SMS. Lorsqu'il y a correspondance, Tivoli Federated Identity Manager considère que la livraison SMS a abouti.

Par défaut, cette valeur n'est pas renseignée.

Par défaut, le corps de la réponse HTTP n'est pas mis en correspondance avec une expression régulière Java et la décision de réussite ou d'échec est fondée sur la valeur de SuccessHTTP
ReturnCode uniquement.

Remarque : Si la réponse HTTP provenant de la passerelle SMS ne contient pas de corps, la correspondance SuccessHTTP ResponseBody RegexPattern n'est pas effectuée.

Requis : Non

Valeurs multiples : Non

Exemple :

- Lorsque le corps de toutes les réponses de la passerelle SMS contient le terme Success ou Failure non suivi d'un caractère de retour à la ligne, la valeur SuccessHTTP Réponse BodyRegex Pattern exemple est

  ```
  Success
  ```

- Lorsque le corps de toutes les réponses de la passerelle SMS contient le texte suivant :

  ```
  MGDID=TTTT
  TTTTTTTTT
  RESPONSE
  CODE=NNN
  SMS=TTTTTTT
  TTTTTTTT
  TTTTTTT
  DATE=NNNNNNNN
  ```

  où chaque ligne se termine par le caractère \n sans être précédée par le caractère \r et que RESPONSECODE est un nombre à trois chiffres de 0 à 199 qui indique la réussite, la valeur SuccessHTTP ResponseBody RegexPattern exemple est

  ```
  (?s).*
  RESPONSE
  CODE=(\d{1,2}
  |[0-1]{1}
  \d{2})\n.*
  ```

Configuration du plug-in EmailOTPDelivery

SMTPHostname

Nom d'hôte du serveur SMTP.

Requis : Oui

Valeurs multiples : Non

Exemple : smtpserver.tfim.example.com

SMTPUsername

Nom d'utilisateur indiqué pour l'authentification SMTP.

Requis : Non

Valeurs multiples : Non

Exemple : nom utilisateur

SMTPPassword

Mot de passe utilisé pour l'authentification SMTP.

Requis : Non

Valeurs multiples : Non

Exemple : `mot de passe`

SenderEmail
Adresse électronique utilisée comme expéditeur du courrier électronique envoyé à l'utilisateur.

Requis : Oui

Valeurs multiples : Non

Exemple : `otp_emailer@example.com`

Référence du plug-in du fournisseur sur les informations utilisateur du mot de passe à utilisation unique

Le plug-in du fournisseur sur les informations utilisateur du mot de passe à utilisation unique extrait des valeurs d'une base de données pour les algorithmes de mot de passe à utilisation unique requérant des informations utilisateur.

Vous pouvez configurer Tivoli Federated Identity Manager pour extraire ces valeurs d'une base de données relationnelle ou basée sur les fichiers. Pour plus d'informations sur la configuration de la base de données, voir les rubriques :

- «Configuration de DB2 pour le stockage d'informations utilisateur sur les mots de passe à utilisation unique», à la page 762
- «Configuration de solidDB pour le stockage d'informations utilisateur sur les mots de passe à utilisation unique», à la page 763

Tivoli Federated Identity Manager est livré avec les plug-in suivants :

JDBCUserInfoModule
Fournit les informations utilisateur d'une base de données basée sur JDBC.

FileUserInfoModule
Fournit les informations utilisateur d'une base de données basée sur des fichiers. Ce plug-in est uniquement pris en charge sur des environnements autonomes ou non configurés en cluster.

Fournisseur d'informations utilisateur JDBC

Les paramètres de configuration suivants sont destinés à `JDBCUserInfoModule`. Tous les paramètres sont facultatifs.

DBDataSource
Nom JNDI de la source de données correspondant à la base de données d'informations utilisateur. Indiquez une source de données définie dans l'environnement WebSphere Application Server.

La valeur par défaut est `jdbc/fim`.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `jdbc/fim`

DBLoggingEnabled
Valeur booléenne permettant un traçage à granularité plus fine sur la connexion à la base de données. Indiquez `true` ou `false`.

La valeur par défaut est `false`.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `false`

Fournisseur d'informations utilisateur fichier

Les paramètres de configuration suivants sont destinés à `FileUserInfoModule`. Tous les paramètres sont facultatifs.

`FileDBFileName`
Nom du fichier de la base de données de fichier d'informations utilisateur. Indiquez le nom de fichier unique et valide. L'utilisateur sous lequel le serveur d'application s'exécute doit disposer d'un accès en écriture à ce fichier. Si le fichier n'existe pas, il est créé par le produit.

La valeur par défaut est `fileUserInfo.properties`.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `fileUserInfo.properties`

`FileDBRootDirectory`
Répertoire principal dans lequel le fichier d'informations utilisateur est stocké. Indiquez un répertoire valide sous lequel le serveur d'application s'exécute et pour lequel l'utilisateur dispose d'un droit en écriture.

La valeur par défaut est le répertoire `/etc` du référentiel de configuration Tivoli Federated Identity Manager *:WAS_ROOT*`/profiles/`*WAS_PROFILE*`/config/itfim/`*FIM_DOMAIN*`/etc`

Obligatoire : Non

Valeurs multiples : Non

Exemple : `/opt/IBM/WebSphere/AppServer/profiles/ip/config/itfim/fimipdomain/etc`

`FileDBKeyTokensDelimeter`
Caractère de séparation des différentes valeurs comprenant la clé d'entrée de la base de données. Indiquez tout caractère se trouvant en dehors des valeurs de nom d'attribut, d'espace de nom d'attribut ou de type de données d'attribut.

La valeur par défaut est `%`.

Obligatoire : Non

Valeurs multiples : Non

Exemple : `%`

Référence associée:

«Fichier de réponses de mot de passe à utilisation unique», à la page 733
Créez un fichier de réponses de mot de passe à utilisation unique à l'aide de la commande **manageItfimOneTimePassword** pour configurer une nouvelle fédération avec un mot de passe à utilisation unique ou modifier une fédération avec un mot de passe à utilisation unique existante. Editez-le avec les valeurs appropriées pour votre environnement.

«Référence du plug-in de livraison de mot de passe à utilisation unique», à la page 756
Le module de livraison du mot de passe à utilisation unique est un plug-in qui fournit les mots de passe à utilisation unique aux utilisateurs. Configurez le plug-in de livraison de mot de passe à utilisation unique afin qu'il puisse être utilisé par Tivoli Federated Identity Manager.

Configuration de DB2 pour le stockage d'informations utilisateur sur les mots de passe à utilisation unique

Vous pouvez configurer DB2 comme votre base de données d'informations utilisateur pour le calcul du mot de passe à utilisation unique.

Avant de commencer

- Vérifiez les exigences de DB2 concernant la prise en charge du fournisseur pour les informations utilisateur. Voir Logiciels supplémentaires.
- Installez la base de données DB2.

Procédure

1. Dans WebSphere Application Server, créez et configurez un contexte JNDI nommé `jdbc/fim`. Voir le centre de documentation WebSphere Application Server. Recherchez *configuring a data source*.
2. Définissez des propriétés de schéma personnalisées en procédant comme suit :
   a. Dans la console d'administration, cliquez sur**Ressources** > **JDBC** > **Sources de données**.
   b. Cliquez sur le nom de la source de données créée à l'étape 1 pour ouvrir la page **Configuration**.
   c. Cliquez sur **Propriétés personnalisées**.
   d. Cliquez sur **currentSchema**.
   e. Dans la zone **Valeur**, entrez `FIM_DB`.
   f. Cliquez sur **OK**.
   g. Cliquez sur **Sauvegarder directement dans la configuration principale**.
3. Exécutez le fichier `.sql` pour créer le schéma de base de données pour des informations utilisateur sur le mot de passe à utilisation unique.

   Systèmes d'exploitation Linux ou UNIX

   Le fichier `.sql` permettant de créer la base de données pour DB2 se trouve dans le répertoire *FIM_HOME*`/dbscripts/db2/`. Par exemple, le répertoire est `/opt/IBM/FIM/dbscripts/db2/`.

   a. Modifiez le fichier `create_schema.sql` et remplacez **&DBUSER** et **$DBPASSWD** par le nom d'utilisateur et le mot de passe de la base de données.
   b. Exécutez le fichier `create_schema.sql` à l'aide de la commande **db2**. Par exemple :

      ```
      db2 -tvf /opt/IBM/FIM/dbscripts/db2/create_schema.sql
      ```

Systèmes d'exploitation Windows

Le fichier `.sql` permettant de créer la base de données pour DB2 se trouve dans le répertoire *FIM_HOME*`\dbscripts\ db2\`. Par exemple, le répertoire est `C:\Program Files\IBM\FIM\dbscripts\db2\`.

a. Modifiez le fichier `create_schema.sql` et remplacez **&DBUSER** et **$DBPASSWD** par le nom d'utilisateur et le mot de passe de la base de données.

b. Exécutez le fichier `create_schema.sql` à l'aide de la commande **db2**. Par exemple :

```
db2 -tvf C:\Progra~1\IBM\FIM\dbscripts\db2\create_schema.sql
```

4. A l'aide de la source de données créée à l'étape 1, à la page 762 et de la console d'administration, testez la connexion à la base de données.

Configuration de solidDB pour le stockage d'informations utilisateur sur les mots de passe à utilisation unique

Vous pouvez configurer solidDB comme votre base de données d'informations utilisateur pour le calcul du mot de passe à utilisation unique.

Avant de commencer

- Vérifiez les exigences de solidDB concernant la prise en charge du fournisseur pour les informations utilisateur. Voir Logiciels supplémentaires.
- Installez la base de données solidDB.

Procédure

1. Dans WebSphere Application Server, créez et configurez un contexte JNDI nommé `jdbc/fim`. Voir le centre de documentation WebSphere Application Server. Recherchez *configuring a data source*.
2. Définissez des propriétés de schéma personnalisées en procédant comme suit :
   a. Dans la console d'administration, cliquez sur**Ressources** > **JDBC** > **Sources de données**.
   b. Cliquez sur le nom de la source de données créée à l'étape 1 pour ouvrir la page **Configuration**.
   c. Cliquez sur **Propriétés personnalisées**.
   d. Cliquez sur **currentSchema**.
   e. Dans la zone **Valeur**, entrez `FIM_DB`.
   f. Cliquez sur **OK**.
   g. Cliquez sur **Sauvegarder directement dans la configuration principale**.
3. Créez une base de données solidDB avec les outils solidDB. Définissez le catalogue de la base de données et le nom d'utilisateur sur `FIM_DB`. Entrez la commande suivante sur une seule ligne :

```
SOLID_BIN_DIRECTORY/solid -UFIM_DB -PDBPASSWORD -CFIM_DB -xexit
  -xdisableallmessageboxes -xhide -CWORKING_DIRECTORY
```

où :

SOLID_BIN_DIRECTORY
Indique le répertoire d'installation `bin` de la base de données solidDB.

DBPASSWORD
Indique le mot de passe de la base de données.

WORKING_DIRECTORY
Indique le répertoire de travail dans lequel se trouvent le fichier solidDB `.ini` et le fichier de licence.

Exemple de commande pour Linux indiquant la licence d'évaluation solidDB :

```
/opt/solidDB/soliddb-7.0/bin/solid -UFIM_DB -Ppassword -CFIM_DB -xexit
  -xdisableallmessageboxes -xhide -C/opt/solidDB/soliddb-7.0/eval_kit/standalone
```

Pour plus d'informations, voir la documentation solidDB.

4. Démarrez la base de données solidDB avec la commande suivante :

```
SOLID_BIN_DIRECTORY/solid -UFIM_DB -PDBPASSWORD -CFIM_DB
  -xdisableallmessageboxes -xhide -CWORKING_DIRECTORY
```

Exemple de commande pour Linux indiquant la licence d'évaluation solidDB :

```
/opt/solidDB/soliddb-7.0/bin/solid -UFIM_DB -Ppassword -CFIM_DB
  -xdisableallmessageboxes -xhide -C/opt/solidDB/soliddb-7.0/eval_kit/standalone
```

Pour plus d'informations, voir la documentation solidDB.

5. Exécutez le fichier `.sql` pour créer le schéma de base de données pour des informations utilisateur sur le mot de passe à utilisation unique.

Systèmes d'exploitation Linux ou UNIX

Le fichier `.sql` permettant de créer le schéma de base de données pour solidDB se trouve dans le répertoire *FIM_HOME*`/dbscripts/soliddb/`. Par exemple, sous Linux, il s'agit du répertoire `/opt/IBM/FIM/dbscripts/soliddb/`.

Exécutez le fichier `create_schema.sql` à l'aide de la commande **solsql**.

```
solsql "NETWORK_NAME" FIM_DB DBPASSWORD
  /opt/IBM/FIM/dbscripts/soliddb/create_schema.sql
```

où :

NETWORK_NAME
Indique le nom de réseau d'un serveur solidDB auquel vous vous connectez.

DBPASSWORD
Indique le mot de passe de la base de données.

Par exemple :

```
solsql "tcpip 1964" FIM_DB password
  /opt/IBM/FIM/dbscripts/soliddb/create_schema.sql
```

Systèmes d'exploitation Windows

Le fichier `.sql` permettant de créer le schéma de base de données se trouve dans le répertoire `C:\Program Files\IBM\FIM\dbscripts\soliddb\`.

Exécutez le fichier `create_schema.sql` à l'aide de la commande **solsql**.

```
solsql "NETWORK_NAME" FIM_DB DBPASSWORD
  C:\Progra~1\IBM\FIM\dbscripts\soliddb\create_schema.sql
```

où :

NETWORK_NAME
Indique le nom de réseau d'un serveur solidDB auquel vous vous connectez.

DBPASSWORD
Indique le mot de passe de la base de données.

Par exemple :

```
solsql.exe "tcpip 1964" FIM_DB password
  C:\Progra~1\IBM\FIM\dbscripts\soliddb\create_schema.sql
```

6. A l'aide de la source de données créée à l'étape 1, à la page 763 et de la console d'administration, testez la connexion à la base de données.

Chapitre 47. Optimisation de mot de passe à utilisation unique

Améliorez les performances du système de mot de passe à utilisation unique en optimisant le composant OTPProviderDynaCacheOTPStore.

OTPProviderDynaCacheOTPStore est le plug-in du magasin de mots de passe à utilisation unique. Il utilise le cache d'objets WebSphere Application Server comme stockage sous-jacent.

Les modules de mots de passe à utilisation unique suivants utilisent ce plug-in de stockage :

- MobileAuthCodeOTPModule : stocke des mots de passe à utilisation unique dans le magasin.
- TOTPModule : utilise le magasin pour l'application de l'utilisation unique.

Vous pouvez régler OTPProviderDynaCacheOTPStore à l'aide de deux approches :

La première consiste à optimiser le cache d'objets WebSphere Application Server.

Le cache d'objets WebSphere Application Server utilisé par OTPProviderDynaCacheOTPStore est itfim-otp. Vous pouvez optimiser ce cache d'objets en modifiant la taille du cache ou en activant le déchargement du disque.

Pour plus de détails, voir la documentation WebSphere Application Server.

La deuxième consiste à optimiser l'utilisation du cache d'objets WebSphere Application Server par OTPProviderDynaCacheOTPStore.

OTPProviderDynaCacheOTPStore extrait le mot de passe à utilisation unique du cache d'objets WebSphere Application Server par le biais d'une interrogation. Si le mot de passe à utilisation unique est indisponible, OTPProviderDynaCacheOTPStore attend un certain temps avant de tenter de l'extraire à nouveau. Ce cycle continue jusqu'à ce que le mot de passe à utilisation unique soit disponible. Si le mot de passe à utilisation unique n'est toujours pas disponible après un certain laps de temps, le délai imparti à OTPProviderDynaCacheOTPStore expire.

Vous pouvez configurer la durée d'attente de OTPProviderDynaCacheOTPStore avant de tenter d'extraire à nouveau le mot de passe à utilisation unique en définissant la propriété personnalisée d'exécution DistributedMap.GetRetryDelay.

Vous pouvez configurer le nombre de nouvelles tentatives avant l'expiration du délai de OTPProviderDynaCacheOTPStore en définissant la propriété personnalisée d'exécution DistributedMap.GetRetryLimit.

Pour plus d'informations, voir Propriétés générales.

Partie 8. Personnalisation

Les rubriques de la section Personnalisation expliquent comment personnaliser les composants et fonctions de Tivoli Federated Identity Manager pour mieux répondre aux besoins de votre environnement.

Chapitre 48. Personnalisation des propriétés de l'environnement d'exécution

Les propriétés personnalisées permettent de personnaliser le service d'exécution de Tivoli Federated Identity Manager afin de répondre à vos besoins spécifiques.

L'utilisation des propriétés personnalisées fait partie des tâches de niveau avancé. Familiarisez-vous avec l'architecture et les services Tivoli Federated Identity Manager pour comprendre comment utiliser les propriétés personnalisées. Pour plus d'informations, reportez-vous au centre de documentation de Tivoli Federated Identity Manager.

Création d'une propriété personnalisée

Vous pouvez personnaliser la configuration d'un domaine en définissant une propriété personnalisée.

Pourquoi et quand exécuter cette tâche

La syntaxe des propriétés personnalisées est la suivante :

nom_propriété = valeur_propriété

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution**. Le panneau Gestion des noeuds d'exécution s'affiche.
3. Cliquez sur **Propriétés personnalisées de l'environnement d'exécution**. Le panneau Propriétés personnalisées de l'environnement d'exécution s'affiche.
4. Sélectionnez la portée de la propriété personnalisée (cellule ou noeud) dans la liste **Portée**. La liste des propriétés de la portée sélectionnée s'affiche.
5. Cliquez sur **Créer**. Un élément est ajouté à la liste des propriétés, avec le nom **nouvelle clé** et la valeur **nouvelle valeur**.
6. Sélectionnez la propriété de la marque de réservation.
7. Entrez une chaîne dans la zone **Nom**. N'insérez pas d'espace dans cette zone.
8. Entrez une chaîne dans la zone **Valeur**. Les espaces sont autorisés dans cette zone.
9. Cliquez sur **OK** pour appliquer les modifications effectuées et quitter le panneau.

Suppression d'une propriété personnalisée

Vous pouvez supprimer un propriété personnalisée en fonction de sa portée.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Gestion des noeuds d'exécution**. Le panneau Gestion des noeuds d'exécution s'affiche.

3. Cliquez sur **Propriétés personnalisées de l'environnement d'exécution**. Le panneau Propriétés personnalisées de l'environnement d'exécution s'affiche.
4. Sélectionnez la portée de la propriété personnalisée (cellule ou noeud) dans la liste **Portée**. La liste des propriétés de la portée sélectionnée s'affiche.
5. Sélectionnez une paire nom-valeur.
6. Cliquez sur **Supprimer**. Le panneau est régénéré et la paire nom-valeur est supprimée de la liste des propriétés personnalisées.
7. Choisissez l'une des actions suivantes :
   - Cliquez sur **Appliquer** pour appliquer les modifications effectuées sans quitter le panneau.
   - Cliquez sur **OK** pour appliquer les modifications effectuées et quitter le panneau.

Liste de référence des propriétés personnalisées

Vous pouvez définir les valeurs de plusieurs propriétés personnalisées. Cette section décrit chacune de ces propriétés.

Pour ajouter des propriétés personnalisées à la configuration de votre domaine, voir «Création d'une propriété personnalisée», à la page 769.

Propriétés générales

DistributedMap.GetRetryLimit

Lorsque cette valeur est définie et supérieure à 0, l'encapsuleur interroge la mappe distribuée autant de fois qu'indiqué dans la configuration avant de renvoyer un message indiquant que les données ne s'y trouvent pas.

- Type de valeur : entier
- Exemple de valeur : 2

DistributedMap.GetRetryDelay

Lorsque le nombre de relances est supérieur à 1, cette valeur définit le temps d'attente entre deux relances (en millisecondes). La valeur par défaut est 2000, soit 2 secondes.

- Type de valeur : entier
- Exemple de valeur : 2000

componentName.**statisticsEnabled**
Lorsque la valeur 'True' est spécifiée, la fonction de suivi statistique d'un composant spécifique est activée et les données collectées peuvent être extraites à l'aide des mécanismes présentés par le composant. Lorsque la valeur 'false' est définie, le suivi statistique n'a pas lieu. En règle générale, cette propriété est définie sur la valeur 'true' pour les composant nécessitant un comptage numérique ou temporel.

- Type de valeur : booléen
- Exemple de valeur : False

Propriétés personnalisées du service de protocole de connexion unique

Utilisez les propriétés personnalisées de la connexion unique pour répondre à vos exigences de déploiement.

requireSoapActionForSoap
Ce paramètre surveille le comportement du service de protocole de connexion unique lorsqu'il reçoit une requête via la méthode POST du navigateur et qu'il doit en déterminer le type (SOAPRequest ou BrowserRequest). Il permet au service de gérer les clients SOAP non conformes qui n'envoient pas l'en-tête SOAPAction nécessaire dans les requêtes.

Valeur par défaut : 'true'

- Type de valeur : booléen
- Exemple de valeur : true

requireContentTypeForSoap
Ce paramètre détermine si une requête SOAP doit contenir ou non un type de contenu `text/xml` ou `application/soap+xml`. Il permet au service de protocole de connexion unique de gérer les clients SOAP non conformes.

Remarque : Lorsque ce paramètre et requestSoapActionForSoap ont tous deux pour valeur 'false', toutes les requêtes reçues sont interprétées comme des requêtes SOAP.

Valeur par défaut : 'true'

- Type de valeur : booléen
- Exemple de valeur : true

POC.allowsCredRefresh
Lorsqu'il a pour valeur 'true', ce paramètre permet d'ignorer LocalLogoutAction au niveau du fournisseur de services lors de la connexion unique et de la fédération. A la place, il régénère les données d'identification. Définissez ce paramètre sur la valeur true pour les plug-ins Web. Sinon, attribuez-lui la valeur false.

Valeur par défaut : 'true'

- Type de valeur : booléen
- Exemple de valeur : true

SPS.PageFactory.HtmlEscapedTokens
Une liste de jetons séparés par des virgules devant utiliser les caractères d'échappement HTML lors de l'affichage sur les pages envoyées au navigateur. En général, cette propriété inclut les macros de la propriété personnalisée d'exécution SPS.PageFactory.Exception2Macro (si elle est utilisée). Cette propriété est une considération de sécurité importante pour empêcher les vulnérabilités de scripts intersites.

- Type de valeur : chaîne
- Exemple de valeur : @TOKEN_A@,@TARGET@

SPS.PageFactory.Exception2Macro
Cette propriété personnalisée d'exécution est une liste séparée par des virgules de paires classname:macro. Le nom de classe représente le nom complet d'une classe d'exception. La macro est la macro de remplacement avec laquelle la classe est mappée. La macro doit commencer et finir par "@", comme indiqué dans les exemples de valeurs.

- Type de valeur : chaîne
- Valeurs d'exemples : com.demo.MyException: @MYEXCEPTION@, com.tivoli.am.fim.trustserver.sts.STSException: @STSEXCEPTION@

SPS.POC.Default.Header.Names.Enabled
cette propriété, lorsqu'elle est spécifiée, permet d'utiliser des noms d'en-tête par défaut comme valeurs d'en-tête du point de contact. Si ce paramètre est défini sur FALSE, les seuls en-têtes qui seront lus ou écrits devront faire partie du fichier de configuration sps.xml.

- Type de valeur : booléen
- Exemple de valeur : false

POC.WebSeal.SignOutInfoDelegate.UserSessionIdHeaderName
Cette valeur se substitue à la valeur par défaut tagvalue_user_session_id.

- Type de valeur : chaîne
- Exemple de valeur : tagvalue_user_session_id

SOAP.AuthType
Type d'authentification à utiliser lors de l'accès au noeud final SOAP. Cette valeur peut être soit `ba`, pour une authentification de base, soit `cert`, pour une authentification basée sur un certificat client.

- Type de valeur : chaîne
- Exemple de valeur : ba

TFIM.SOAP.Port
Ce paramètre est une liste de numéros de port séparée par des virgules.

- Type de valeur : chaîne
- Exemple de valeur : 9443, 9445

SPS.WebSealPoc.ContextPoolSize
Indique le nombre d'objets PDContext disponibles dans le pool. Cette valeur correspond au nombre de clients à autoriser lors de l'utilisation de la connexion unique.

Il peut s'avérer nécessaire d'augmenter cette valeur en fonction de la charge représentée par les déconnexions. En cas de déconnexions simultanées massives, l'environnement d'exécution Tivoli Federated Identity Manager risque de contenir un nombre d'objets PDContext insuffisant, ce qui peut entraîner l'échec des déconnexions. Etant donné que chaque objet PDContext utilise des ressources système, telles que la mémoire et des descripteurs de fichier, cette valeur doit être sélectionnée avec soin. Elle doit être supérieure à 0.
Valeur par défaut : 5

- Type de valeur : entier
- Exemple de valeur : 5

SPS.WebSealPoc.DisablePDSignout
Lorsqu'il est défini sur true, ce paramètre désactive la fonctionnalité de

fermeture de session du client point de contact WebSEAL du service de protocole de connexion unique. Lorsque l'opération de fermeture de session est appelée, il consigne l'absence de fermeture de session et renvoie une réponse positive. Lorsque ce paramètre est activé, la configuration de Tivoli Access Manager Java Runtime (PDJRTE) n'est pas nécessaire pour le service de protocole de connexion unique.

Valeur par défaut : 'false'.

- Type de valeur : booléen
- Exemple de valeur : true

SPS.WebSealPoc.Force.PdAdmin.Task
Lorsqu'il a pour valeur 'true', ce paramètre impose, lors du rappel du point de contact WebSeal, l'utilisation systématique des tâches **pdadmin server** pour déconnecter l'utilisateur.

- Type de valeur : booléen
- Exemple de valeur : false

SPS.WebSealPoc.ContextPoolInitAttempts
Cette valeur représente le nombre de tentatives d'initialisation des objets PDContext. La valeur par défaut est 1 et cette valeur doit être supérieure à 0.

- Type de valeur : entier
- Exemple de valeur : 1

SPS.WebSealPoc.ContextPoolInitTimeout
Cette valeur représente la durée maximale pour l'initialisation des objets PDContext. Une fois le délai expiré, l'initialisation prend fin. La valeur par défaut est 10000 et doit être supérieure à 0. La quantité est exprimée enmillisecondes.

- Type de valeur : entier
- Exemple de valeur : 10000

Propriétés personnalisées du service d'accréditation

Utilisez les propriétés personnalisées du service d'accréditation pour répondre à vos exigences de déploiement.

username.disable.password.validation
Lorsqu'il a pour valeur 'true', ce paramètre permet à UsernameTokenSTSModule d'ignorer la validation du mot de passe.

La valeur par défaut est 'false'.

- Type de valeur : booléen
- Exemple de valeur : true

username.jaas.provider.hostname
Indique un nom pour l'hôte local si WebSphere n'a pas été configuré avec la valeur de localhost pour le nom d'hôte.

La valeur par défaut est localhost.

- Type de valeur : chaîne
- Exemple de valeur : localhost

username.jaas.provider.port
Indique le port configuré pour le service WebSphere NameServer local.

La valeur par défaut est 2809.

- Type de valeur : entier

- Exemple de valeur : 2809

pdjrte.context.min.pool.size
Indique la taille minimale du pool de contextes d'autorisation. Ce paramètre est utilisé par UsernameTokenSTSModule. Définissez ce paramètre uniquement si une évaluation des performances requiert qu'il soit défini.

- Type de valeur : entier
- Exemple de valeur : 5

pdjrte.context.max.pool.size
Indique la taille maximale du pool de contextes d'autorisation. Ce paramètre est utilisé par UsernameTokenSTSModule. Définissez ce paramètre uniquement si une évaluation des performances requiert qu'il soit défini.

- Type de valeur : entier
- Exemple de valeur : 50

ivcred.allow.groupUpdate
S'il est défini sur true, ce paramètre tente de modifier les droits d'accès en ajoutant des groupes.

Remarque : Réservez ce paramètre à des cas particuliers.

- Type de valeur : booléen
- Exemple de valeur : false

ivcred.insert.CRLF76
Lorsque la valeur est true, `IVCred` codé en base64 et généré par le module de service de jeton de sécurité `STSTokenIVCred` est séparé sur plusieurs lignes. Si cette propriété personnalisée est définie sur false, `IVCred` codé en base64 et généré par le module de service de jeton de sécurité `STSTokenIVCred` n'est pas séparé sur plusieurs lignes.

Valeur par défaut : True

- Type de valeur : booléen
- Exemple de valeur : False

saml.use.rst.lifetime
Indique aux modules SAML d'utiliser la durée de vie de l'élément RequestSecurityToken pour déduire celle de la vérification SAML émise. Lorsque ce paramètre est défini sur false, il n'utilise pas la durée de vie de RequestSecurityToken.

Valeur par défaut : false.

- Type de valeur : booléen
- Exemple de valeur : false

passticket.disable.uppercase.principal
Utilise le gestionnaire RACF local pour donner au module PassTicket l'instruction de ne pas convertir l'ensemble du nom principal en majuscule avant de tenter de générer un PassTicket. Lorsque ce paramètre est défini sur false, il convertit toujours le nom du principal en majuscule pour le gestionnaire RACF local.

Valeur par défaut : false.

- Type de valeur : booléen
- Exemple de valeur : false

sts.use.issuer.saml20.sso
Donne au module SAML 2.0 l'instruction d'utiliser la valeur Issuer, à la place de la valeur NameID NameQualifier pour rechercher un alias durant l'opération de connexion unique.

Valeur par défaut : false.

- Type de valeur : booléen
- Exemple de valeur : false

username.wss.namespace.override
Lorsque cette propriété n'est pas précisée, la valeur par défaut est l'espace de nom du profil de jeton WSS 1.1. La clé de cette propriété peut être utilisée comme préfixe pour définir la portée de la propriété sur une chaîne STS spécifique, par exemple, `username.wss.namespace.override.uuid1234`.

- Type de valeur : chaîne
- Exemple de valeur : *<a_URI_namespace>*

STS.validateMappingRules
Indique si la règle de mappage est validée lorsqu'elle est importée via la console ou l'interface de ligne de commande. Si le paramètre **STS.validateMappingRules** est spécifié et que la valeur est égale à la chaîne `false`, sans tenir compte de la casse, la règle de mappage n'est pas validée. Sinon, la règle de mappage est validée.

- Type de valeur : booléen
- Exemple de valeur : false

authorizationsts.initial.num.context
Indique la quantité initiale d'objets de contexte à créer au démarrage. Ce paramètre détermine le nombre de connexions créées et gérées par le pool.

- Type de valeur : entier
- Exemple de valeur : 5

authorizationsts.max.num.context
Indique la quantité maximale d'objets de contexte à créer en tout. Ce paramètre détermine le nombre de connexions créées et gérées par le pool.

- Type de valeur : entier
- Exemple de valeur : 10

Propriétés personnalisées pour OAuth 2.0

Utilisez les propriétés personnalisées OAuth 2.0 pour répondre à vos exigences de déploiement.

OAuth20.DoNotSendXFrameOptionsHeader
Ce paramètre donnent aux noeuds finals OAuth 2.0 l'instruction de ne pas inclure l'en-tête X-Frames-Options: SAMEORIGIN dans les réponses au client OAuth ou au propriétaire de la ressource, en particulier dans l'acceptation de la page d'autorisation. Ce paramètre est désactivé par défaut. Utilisez ce paramètre uniquement lorsque le serveur d'autorisations et le client OAuth ont une relation solide d'accréditation.

Par défaut, l'option **OAuth20.DoNotSendXFrameOptionsHeader** n'existe pas. Pour utiliser ce paramètre, créez le paramètre et définissez la valeur sur **true**.

- Type de valeur : booléen
- Exemple de valeur : true

Propriétés personnalisées pour SAML 1.0

Utilisez les propriétés personnalisées SAML 1.0 pour répondre à vos exigences de déploiement.

saml.use.legacy.clockskew.default
Tivoli Federated Identity Manager utilise par défaut l'horloge locale de l'environnement d'exécution lors de la validation des horodatages des assertions SAML. Définissez ce paramètre sur**true** si vous voulez ajouter un décalage d'horloge de 60 secondes entre le serveur et l'horodotage de l'assertion SAML.

Valeur par défaut : False

- Type de valeur : booléen
- Exemple de valeur : False

Propriétés personnalisées pour SAML 1.1

Utilisez les propriétés personnalisées SAML 1.1 pour répondre à vos exigences de déploiement.

SAML.AllowDebugMessages
Lorsque cette propriété est définie sur true et qu'un échec de résolution d'artefact SAML se produit, les fichiers SystemOut.log et SystemErr.log contiennent un message d'information. De plus, le message contient des informations de débogage supplémentaires sur la demande contenant l'artefact qui a échoué et le motif de l'échec.

Remarque : Ce message est disponible uniquement en anglais.

Valeur par défaut : False

- Type de valeur : booléen
- Exemple de valeur : SAML.AllowDebugMessage=true

saml.use.legacy.clockskew.default
Par défaut, Tivoli Federated Identity Manager ajoute un décalage d'horloge de 60 secondes lors de la validation des horodatages des assertions SAML. Pour désactiver la valeur par défaut de 60 secondes, ajoutez la propriété personnalisée : `saml.use.legacy.clockskew.default = false`

Valeur par défaut : True

- Type de valeur : booléen
- Exemple de valeur : true

Propriétés personnalisées du service de clés

Utilisez des propriétés personnalisées pour le service de clés pour répondre à vos exigences.

kessjksservice.include.keyinfo.x509.certificate.data
Inclut un certificat codé en Base64 dans l'élément KeyInfo de la signature. Lorsque ce paramètre a pour valeur 'true', implicitement ou par l'utilisation explicite de cette propriété, les autres propriétés d'exécution KESS sont ignorées. Lorsque cette propriété n'est pas précisée, la valeur par défaut est 'true'.

- Type de valeur : booléen
- Exemple de valeur : true

kessjksservice.include.keyinfo.x509.subject.key.identifier
Inclut l'identificateur de clé du sujet dans l'élément KeyInfo de la signature lorsque le certificat délivré le prend en charge. Cette propriété peut être utilisée en supplément de issuer.details et de subject.name. Lorsque cette propriété n'est pas précisée, la valeur par défaut est 'false'.

- Type de valeur : booléen
- Exemple de valeur : true

kessjksservice.include.keyinfo.x509.issuer.details
Ajoute des informations sur l'émetteur X509 dans l'élément KeyInfo de la signature. Cette propriété peut être utilisée en supplément de subject.key.identifier et subject.name. Lorsque cette propriété n'est pas précisée, la valeur par défaut est 'false'.

- Type de valeur : booléen
- Exemple de valeur : true

kessjksservice.include.keyinfo.x509.subject.name
Ajoute le nom distinctif du sujet X509 dans l'élément KeyInfo de la signature. Cette propriété peut être utilisée en supplément de subject.key.identifier et issuer.details. Lorsque cette propriété n'est pas précisée, la valeur par défaut est 'false'.

- Type de valeur : booléen
- Exemple de valeur : true

kessjksservice.exclude.inclusive.namespace.prefixes
Liste séparée par des virgules des noms de préfixe. Lorsque cette propriété est définie, les préfixes de cette liste ne sont pas ajoutés à la liste InclusiveNamespaces présente dans l'élément Signature.

- Type de valeur : chaîne
- Exemple de valeur : ds

kessjksservice.supportedalgorithms.signature
Cette propriété d'exécution personnalisée est une liste d'UI d'algorithmes de signature séparée par des virgules. La valeur par défaut est l'ensemble complet des algorithmes de signature pris en charge par IBM Tivoli Federated Identity Manager.

Vous pouvez modifier cette propriété d'exécution personnalisés pour inclure des algorithmes non testés que votre système prend en charge, mais ne pris en charge par IBM. Soyez vigilant lorsque vous modifiez cette propriété pour éviter un échec de la signature.

Valeur par défaut : http://www.w3.org/2000/09/xmldsig#rsa-sha1,http://www.w3.org/2000/09/xmldsig#dsa-sha1,http://www.w3.org/2001/04/xmldsig-more#rsa-sha256

- Type de valeur : chaîne
- Exemple de valeur : http://www.w3.org/2000/09/xmldsig#rsa-sha1,http://www.w3.org/2000/09/xmldsig#dsa-sha1,http://www.w3.org/2001/04/xmldsig-more#rsa-sha256

kessjksservice.supportedalgorithms.messagedigest
Cette propriété d'exécution personnalisée est une liste d'URI d'algorithmes de valeur digest séparée par des virgules. La valeur par défaut est l'ensemble complet des algorithmes de prétraitement pris en charge par IBM Tivoli Federated Identity Manager.

Vous pouvez modifier cette propriété d'exécution personnalisés pour inclure des algorithmes non testés que votre système prend en charge, mais ne pris en charge par IBM. Soyez vigilant lorsque vous modifiez cette propriété pour éviter un échec de la signature.

Valeur par défaut : http://www.w3.org/2000/09/xmldsig#sha1,http://www.w3.org/2001/04/xmlenc#sha256,http://www.w3.org/2001/04/xmlenc#sha512

- Type de valeur : chaîne
- Exemple de valeur : http://www.w3.org/2000/09/xmldsig#sha1,http://www.w3.org/2001/04/xmlenc#sha256,http://www.w3.org/2001/04/xmlenc#sha512

key.selection.criteria
Cette propriété d'exécution personnalisée vous permet de configurer l'ordre des certificats ou des clés. Utilisez ces valeurs pour la propriété personnalisée :

only.alias
Alias uniquement : la clé sélectionnée uniquement, sans substitution automatique. Si la clé est incorrecte, le logiciel indique un échec. Configurez la propriété pour utiliser cette valeur.

shortest.lifetime
Durée de vie la plus courte : pour la signature, une clé valide avec la durée de vie disponible la plus courte. Pour la validation, la disponibilité de la durée de vie de clé fonctionne de la plus courte à la plus longue.

longest.lifetime
Durée de vie la plus longue : pour la signature, une clé valide avec la durée de vie disponible la plus longue. Pour la validation, la disponibilité de la durée de vie de clé fonctionne de la plus longue à la plus courte.

- Type de valeur : chaîne
- Exemple de valeur : only.alias

Propriétés personnalisées d'un client SOAP

Utilisez les propriétés personnalisées SOAP pour répondre à vos exigences de déploiement.

com.tivoli.am.fim.soap.client.jsse.provider
Nom de fournisseur Java Secure Socket Extension (JSSE) devant être utilisé à la place d'IBMJSSE pour les connexions socket client SOAP.

- Type de valeur : chaîne
- Exemple de valeur : IBMJSSE

com.tivoli.am.fim.soap.client.jce.provider
Nom de fournisseur Java Cryptography Extension (JCE) Cryptography Socket Extension (JCE) devant être utilisé à la place d'IBMJCE pour les fichiers de clés certifiées de client SOAP.

- Type de valeur : chaîne
- Exemple de valeur : IBMJCE

com.tivoli.am.fim.soap.client.trust.provider
Nom d'algorithme du fournisseur Java Trust Manager qu'il convient d'utiliser à la place d'IbmX509 pour les gestionnaires d'accréditation SOAP client.

- Type de valeur : chaîne

- Exemple de valeur : IbmX509

Propriétés personnalisées de SAML 2.0

Utilisez les propriétés personnalisées SAML 2.0 pour répondre à vos exigences de déploiement.

SAML.Assertion.IncludeNSPrefixList.DS
Lorsque ce paramètre est défini sur TRUE, `ds` est inclus dans l'attribut `PrefixList` de `InclusiveNamespaces` dans l'assertion SAML.

Valeur par défaut : False

- Type de valeur : booléen
- Exemple de valeur : true

SAML20.LogoutRequest.NotOnOrAfter.Enabled
Lorsque l'attribut NotOnOrAfterAttribute est défini sur TRUE, il est inclus dans les messages LogoutRequest envoyés par le fournisseur d'identité au fournisseur de services.

Valeur par défaut : True

- Type de valeur : booléen
- Exemple de valeur : true

SAML20.LogoutRequest.NotOnOrAfter.Lifetime
Indique le délai à utiliser pour définir l'attribut NotOnOrAfter lors de la demande de déconnexion.

Valeur par défaut : 120

- Type de valeur : entier
- Exemple de valeur : 300

saml.use.legacy.clockskew.default
Par défaut, Tivoli Federated Identity Manager ajoute un décalage d'horloge de 60 secondes lors de la validation des horodatages des assertions SAML. Pour désactiver la valeur par défaut de 60 secondes, ajoutez la propriété personnalisée : `saml.use.legacy.clockskew.default = false`

Valeur par défaut : True

- Type de valeur : booléen
- Exemple de valeur : true

SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding
Lorsque l'attribut est défini sur TRUE, `RelayState`, qui est dans une réponse d'authentification non sollicitée, est codé dans l'URL par le fournisseur d'identité avant d'être envoyé au fournisseur de services. Cette configuration s'applique à une réponse qui est envoyée à l'aide d'une liaison `HTTP POST` et d'une liaison `HTTP ARTIFACT` avec le mode de diffusion d'artefact `HTTP POST`.

Le codage d'URL peut être contrôlé à trois niveaux :

Niveau global
Contrôle le codage d'URL pour toutes les fédérations et tous les partenaires.

Exemple de configuration :`SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding = true`

Niveau de la fédération

Contrôle le codage d'URL pour une fédération spécifique et tous ses partenaires.

Exemple de configuration : `SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID> = true`

Exemple pour SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID>:

```
SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding_
https://idp/sps/fed/saml20 = true
```

Niveau de partenariat

Contrôle le codage d'URL pour une fédération spécifique et un partenaire spécifique.

Exemple de configuration : `SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID>_<PARTNERID>= true`

Exemple pour SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID>_<PARTNERID>:

```
SAML20.IDP.UnsolicitedSSO.RelayState.URLEncoding_https://idp/
sps/fed/saml20_https://sp/sps/fed/saml20 = true
```

Valeur par défaut : True

- Type de valeur : booléen
- Exemple de valeur : False

`<FEDERATION>` représente l'ID du fournisseur de la fédération et `<PARTNER>` l'ID du fournisseur du partenaire. Vous pouvez obtenir l'ID du fournisseur de la fédération à partir de la page Propriétés de la fédération de la console et l'ID du fournisseur du partenaire à partir de la page Propriétés du partenaire de la console.

Vous pouvez utiliser les trois niveaux de contrôle simultanément. Tivoli Federated Identity Manager implémente l'utilisation simultanée en contrôlant les paramètres `RelayState` afin de choisir quelle action prendre dans l'ordre suivant :

1. Paramètre de niveau de partenariat
2. Paramètre de niveau de fédération
3. Paramètre de niveau global

SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding

Lorsque l'attribut est défini sur TRUE, `RelayState`, qui est dans une réponse d'authentification non sollicitée, est décodé dans l'URL par le fournisseur de services après avoir été reçu du fournisseur d'identité.

Le codage d'URL peut être contrôlé à trois niveaux :

Niveau global

Contrôle le codage d'URL pour toutes les fédérations et tous les partenaires.

Exemple de configuration :

```
SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding = true
```

Niveau de la fédération
Contrôle le codage d'URL pour une fédération spécifique et tous ses partenaires.

Exemple de configuration : `SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID> = true`

Exemple pour SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID>:

```
SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding_https://sp/sps/fed/saml20 = true
```

Niveau de partenariat
Contrôle le codage d'URL pour une fédération spécifique et un partenaire spécifique.

Exemple de configuration : `SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID>_<PARTNERID>= true`

Exemple pour SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding_<FEDERATIONID>_<PARTNERID>:

```
SAML20.SP.UnsolicitedSSO.RelayState.URLEncoding_https://sp/sps/fed/saml20_https://idp/sps/fed/saml20 = true
```

Valeur par défaut : True

- Type de valeur : booléen
- Exemple de valeur : False

`<FEDERATION>` représente l'ID du fournisseur de la fédération et `<PARTNER>` l'ID du fournisseur du partenaire. Vous pouvez obtenir l'ID du fournisseur de la fédération à partir de la page Propriétés de la fédération de la console et l'ID du fournisseur du partenaire à partir de la page Propriétés du partenaire de la console.
Vous pouvez utiliser les trois niveaux de contrôle simultanément. Tivoli Federated Identity Manager implémente l'utilisation simultanée en contrôlant les paramètres `RelayState` afin de choisir quelle action prendre dans l'ordre suivant :

1. Paramètre de niveau de partenariat
2. Paramètre de niveau de fédération
3. Paramètre de niveau global

Propriétés personnalisées de la console

STS.showSSOChains
Ce paramètre permet de vérifier si la console autorise un administrateur à gérer ou à modifier des chaînes générées automatiquement pour les transactions de connexion unique. La définition de cette valeur sur `false` ne désactive pas la propriété personnalisée. Vous devez supprimer la paire valeur et clé de la table de propriétés personnalisées.

- Type de valeur : booléen
- Exemple de valeur : true

STS.showUSCChains
Ce paramètre permet de vérifier si la console autorise un administrateur à gérer ou à modifier des chaînes générées automatiquement pour les fédérations

User Self Care. La définition de cette valeur sur `false` ne désactive pas la propriété personnalisée. Vous devez supprimer la paire valeur et clé de la table de propriétés personnalisées.

- Type de valeur : booléen
- Exemple de valeur : true

STS.showAQChains

Ce paramètre permet de vérifier si la console autorise un administrateur à gérer ou à modifier des chaînes générées automatiquement pour les fédérations SAML 2 activant le service de requête d'attribut. La définition de cette valeur sur `false` ne désactive pas la propriété personnalisée. Vous devez supprimer la paire valeur et clé de la table de propriétés personnalisées.

- Type de valeur : booléen
- Exemple de valeur : true

Propriété personnalisée pour OpenID

Utilisez les propriétés personnalisées d'OpenID pour répondre à vos exigences de déploiement.

OpenID.TrustedSitesManagerModuleID

ID de module plug-in associé à un module qui implémente le point d'extension com.tivoli.am.fim.protocols.openid_trusted_sites_manager. Il existe deux exemples d'implémentaton de cette extension :

- TrustedSitesManagerCookieImpl
- TrustedSitesManagerMemoryImpl

Lorsque ce paramètre n'est pas spécifié, la valeur par défaut est TrustedSitesManagerCookieImpl.

- Type : chaîne
- Exemple de valeur : TrustedSitesManagerCookieImpl

OPENID.DiscoveredInformationExpirationSeconds

Indique le nombre de secondes durant lequel sont mises en cache les informations reconnues pour tout identificateur fourni par un utilisateur OpenID. Si cette valeur est inférieure ou égale à zéro, les données ne sont pas mises en cache (comportement par défaut). Ce paramètre contrôle un cache pour les informations reconnues. Utilisez ce paramètre uniquement lorsque les mêmes informations de connexion d'identificateur OP sont fréquemment utilisées par la majorité des utilisateurs du système. Par exemple, dans un déploiement d'intranet.

OPENID.SkipClaimedIdDiscovery

Détermine si les identificateurs revendiqués sont vérifiés lors d'une connexion avec identificateur OP. Ce paramètre est défini sur true uniquement dans un environnement qui utilise un OP certifié avec la partie de confiance. Sinon, il existe un risque pour la sécurité. Ce paramètre est utilisé typiquement dans un environnement intranet.

- Type : Booléen
- Exemple de valeur : False (valeur par défaut)

Propriété personnalisée pour le protocole de sécurité de transport

Définition du protocole de sécurité de transport pour les connexions HTTPS

Le IBM Tivoli Federated Identity Manager crée SSL_TLS en tant que protocole de sécurité par défaut pour les connexions HTTPS. Pour modifier ou remplacer le protocole par défaut, indiquez la propriété personnalisée d'exécution suivante dans le fichier fim.appservers.properties :

```
com.tivoli.am.fim.soap.client.ssl.protocol= PROTOCOL
```

PROTOCOL correspond à l'un des protocoles pris en charge par l'extension Java Secure Socket Extension utilisée par le serveur WebSphere Application Server sous-jacent.

Exemples :

- SSL_TLS
- SSL
- SSLv2
- SSLv3
- TLS
- TLSv1

Remarque : Ces exemples de protocole ne sont pas nécessairement pris en charge.

Propriétés personnalisées pour les jetons LTPA

Spécification des propriétés d'exécution de Tivoli Federated Identity Manager personnalisées qui forcent la génération QName compatible

Les versions 6.0.2 et 6.1 de WebSphere Application Server ne font pas la distinction entre les jetons LTPA v1 et LTPA v2 dans le services Web. Seules les valeurs BinarySecurityToken ValueType sont prises en charge pour les jetons LTPA, et le QName du type de valeur est :

```
http://www.ibm.com/websphere/appserver/tokentype/5.0.2#LTPA
```

Lorsque le module STS de Tivoli Federated Identity Manager émet un jeton LTPA v2, le jeton est créé avec le QName suivant. Ce QName est correct, mais il n'est pas pris en charge par WebSphere Application Server versions 6.0.2 et 6.1 :

```
http://www.ibm.com/websphere/appserver/tokentype#LTPAv2
```

Cet APAR fournit des propriétés d'exécution Tivoli Federated Identity Manager personnalisées qui forcent la génération QName compatible si nécessaire. Pour activer le mode de compatibilité, définissez une ou deux des propriétés d'exécution personnalisées suivantes :

```
ltpa.enable.compat.mode.[chainid_uuid]=true ltpa.enable.compat.mode=true
```

où *chainid_uuid* représente la valeur de l'UUID de la chaîne. Par exemple :

ltpa.enable.compat.mode.[uuideb42e428-011b-1ebc-a0cb-9e6c4b35c1c7]=true

Pour déterminer la valeur de l'UUID de la chaîne, dans la console d'administration, sélectionnez **Trust Service Chains** > **Select Action** > **Show Chain ID in column in table**. Cette sélection d'action entraîne l'affichage d'une nouvelle colonne dans la table qui présente l'ID de chaîne unique.

Chapitre 49. Personnalisation d'un formulaire de connexion d'authentification pour une connexion unique

Personnalisez un formulaire de connexion d'authentification en ajoutant des paramètres à un profil de serveur point de contact WebSphere ou WebSEAL.

Lorsque des requêtes utilisateur accèdent à une fédération de connexion unique, le fournisseur d'identité lance une connexion unique en authentifiant l'utilisateur. Pour authentifier l'utilisateur, le fournisseur d'identité utilise un serveur point de contact afin d'afficher une page de connexion basée sur des formulaires.

Lorsqu'un fournisseur d'identité participe à plusieurs fédérations ou héberge plusieurs partenaires dans une fédération, l'administrateur peut personnaliser le formulaire de connexion par défaut.

En tant qu'administrateur, vous pouvez personnaliser les éléments suivants :

- La page de connexion selon le contenu des requêtes envoyées par les fournisseurs de service.
- L'apparence du formulaire de connexion.
- Le type d'authentification requis.
- Les pages de connexion pour les serveurs point de contact WebSEAL et WebSphere.

Pour personnaliser la page de connexion, utilisez la console d'administration Tivoli Federated Identity Manager pour configurer un nouveau profil de serveur point de contact. Dans le nouveau profil, ajoutez un paramètre à l'appel d'authentification, et indiquez une ou plusieurs valeurs pour le paramètre.

Tivoli Federated Identity Manager fournit des paramètres qui sont toujours disponibles et cohérents sur tous les types de fédération et d'autres qui sont spécifiques au type de fédération.

Les protocoles qui prennent en charge les paramètres spécifiques au protocole sont les suivants :

- SAML 1.x
- SAML 2
- OpenID

L'ensemble de valeurs définies est décrit dans «Macros prises en charge pour la personnalisation d'un formulaire de connexion d'authentification», à la page 786.

Présentation des tâches :

1. Consultez les valeurs prises en charge pour votre type de protocole et identifiez celles que vous souhaitez utiliser. Voir «Macros prises en charge pour la personnalisation d'un formulaire de connexion d'authentification», à la page 786.
2. Créez un nouveau profil de serveur point de contact. Voir «Configuration d'un serveur point de contact pour prendre en charge la personnalisation des pages de connexion», à la page 788.

Macros prises en charge pour la personnalisation d'un formulaire de connexion d'authentification

Cette rubrique décrit l'ensemble de macros permettant de personnaliser un formulaire de connexion d'authentification.

Tivoli Federated Identity Manager fournit des paramètres d'authentification contextuels dans la personnalisation des formulaires de connexion. Lors de l'utilisation de WebSEAL en tant que serveur point de contact, ce sont les paramètres chaîne-requête de la page de connexion. Pour WebSphere, ils sont situés dans le cookie WASReqURL lorsque la page de connexion est chargée. Les paramètres sont des macros dans la configuration de l'appel d'authentification pour le profil de serveur point de contact.

Remarque : Lorsque vous utilisez le point de contact WebSphere, l'URL de la valeur du paramètre de chaîne requête doit être décodée deux fois.

Les macros prises en charge sont les suivantes :

- Macros indépendantes du protocole
- Macros du protocole SAML
- Macros du protocole OpenID
- Macros du protocole OAuth

Remarque : Si la valeur de `authentication.macros` est plis longue que la longueur autorisée du paramètre de chaîne-requête, le cookie WASReqURL ne sera pas présent dans le fournisseur d'identité.

Macros indépendantes du protocole pour la personnalisation d'un formulaire de connexion d'authentification

Les macros suivantes sont indépendantes du protocole et elles peuvent être utilisées quel que soit le type de fédération utilisé.

Tableau 157. Macros indépendantes du protocole prises en charge

Macro	Nom du paramètre de chaîne de requête	Description
%FEDID%	FedId	Spécifie une identificateur unique (UUID) utilisé en interne par Tivoli Federated Identity Manager pour identifier la fédération.
%FEDNAME%	FedName	Spécifie le nom affecté par l'utilisateur de la fédération.

Macros prises en charge par le protocole SAML pour la personnalisation d'un formulaire de connexion d'authentification

Les macros suivantes sont prises en charge pour le protocole SAML. Les macros sont prises en charge pour SAML 1.x et SAML 2.0, sauf mention contraire.

Tableau 158. Macros de protocole SAML prises en charge

Macro	Nom du paramètre de chaîne de requête	Description et valeur
%PARTNERID%	PartnerId	Représente le partenaire SSO employé l'utilisateur pour se connecter. Valeur SAML : la valeur est l'ID fournisseur du partenaire.
%TARGET%	Cible	Représente l'URL cible au niveau du partenaire, si elle est connue. Valeur SAML : la valeur est la valeur du paramètre cible.
%SPRELAYSTATE%	SPRelayState	Pris en charge par SAML 2.0 uniquement. Représente les données RelayState accompagnant la demande SSO, le cas échéant. Valeur SAML : données RelayState accompagnant la demande SAML AuthnRequest.
%ACSURL%	AssertionConsumerURL	Représente l'URL de service d'assertion client du partenaire, si applicable. Valeur SAML : la valeur est l'URL ACS partenaire.
%AUTHNCONTEXT%	AuthnContext	**Prise en charge pour SAML 2.0 uniquement** Représente la valeur AuthnContext dans la demande (si applicable). Valeur SAML : la valeur est une chaîne encodée de base 64 représentant le fichier XML de RequestedAuthnContext dans SAML AuthnRequest (si présent).
%SSOREQUEST%	SSORequest	**Prise en charge pour SAML 2.0 uniquement** Représente la valeur entière SSO (si applicable). Valeur SAML : la valeur est une chaîne encodée de base 64 représentant le fichier XML de toute la valeur SAML AuthnRequest.
%FORCEAUTHN%	ForceAuthn	**Prise en charge pour SAML 2.0 uniquement** La valeur `true` ou `false`. Valeur SAML : si l'indicateur ForceAuthn est défini dans la requête SAML 2 SSO impliquant la ré-authentification de l'utilisateur, la valeur est `true`. Sinon, la valeur est `false`.

Macros prises en charge par OpenID pour la personnalisation d'un formulaire de connexion d'authentification

Les macros suivantes sont prises en charge pour le protocole OpenID.

Tableau 159. Macros prises en charge par le protocole OpenID

Macro	Nom du paramètre de chaîne de requête	Description et valeur
%PARTNERID%	PartnerId	Représente le partenaire SSO employé par l'utilisateur pour se connecter. Valeur OpenID : la valeur du paramètre openid.trustroot.
%TARGET%	Cible	Représente l'URL cible au niveau du partenaire, si elle est connue. Valeur OpenID : la valeur du paramètre openid.return_to.
%SSOREQUEST%	SSORequest	Représente la valeur entière SSO (si applicable). Valeur OpenID : la requête checkid_setup en version encodée basée 64 de la requête SSO codée dans l'URL.
%UNSATISFIEDPAPEPOLICIES%	UnsatisfiedPapePolicies	Représente une liste de chaînes représentant les règles PAPE. Ces chaînes sont renvoyées comme "pas encore satisfaites" par la règle de mappage du fournisseur d'identité dans une fédération de fournisseur d'identité OpenID. Valeur OpenID : règles renvoyées dans l'attribut ContextAttributes openid.pape.to_be_satisfied_auth_policies
%FORCEAUTHN%	ForceAuthn	Spécifie si l'authentification est appliquée au niveau du fournisseur d'identité. Les valeurs sont `true` ou `false`. Valeur OpenID : la valeur est `true` si l'un des critères suivants est satisfait : • la valeur PAPE max_auth_age est zéro (ce qui signifie que l'authentification est réappliquée) • la règle de mappage IDP du fournisseur d'identité OpenID applique l'authentification à cause de règles PAPE non satisfaites • l'heure d'authentification renvoyée par la règle de mappage IDP ne satisfait pas la valeur max_auth_age demandée par le RP (hors zéro) Sinon, la valeur est `false`.

Macros prises en charge par le protocole OAuth pour la personnalisation d'un formulaire de connexion d'authentification

Le tableau suivant indique comment une fédération OAuth remplit les macros d'authentification.

Tableau 160. Macros de protocole OAuth prises en charge

Macro	Nom du paramètre de chaîne de requête	Description et valeur
%PARTNERID%	PartnerId	Identificateur du client unique OAuth.
%TARGET%	Cible	URI de réacheminement du client OAuth.
%SSOREQUEST%	SSORequest	Chaîne codée en base 64 représentant les paramètres de requête et de corps de la requête OAuth.

Configuration d'un serveur point de contact pour prendre en charge la personnalisation des pages de connexion

Cette rubrique décrit comment configurer un serveur point de contact pour prendre en charge la personnalisation d'une page de connexion.

Avant de commencer

Vérifiez que vous :

- Comprenez la manière dont les pages de connexion personnalisées sont prises en charge. Voir Chapitre 49, «Personnalisation d'un formulaire de connexion d'authentification pour une connexion unique», à la page 785.
- Connaissez les macros à spécifier pour le paramètre d'appel d'authentification. Voir «Macros prises en charge pour la personnalisation d'un formulaire de connexion d'authentification», à la page 786.

Remarque : Vous n'avez pas besoin de créer ni de publier un plug-in d'appel point de contact personnalisé avant de spécifier les macros d'authentification. La prise en charge des macros d'authentification est fournie par défaut. Lorsque vous exécutez l'assistant de configuration, vous pouvez ignorer le message indiquant que vous devez publier un plug-in avant d'utiliser l'assistant.

Pourquoi et quand exécuter cette tâche

La procédure qui suit explique comment ajouter un serveur point de contact personnalisé tel que les serveurs point de contact déjà définis dans votre environnement afin de modifier les informations affichées sur la page de connexion.

Procédure

1. Connectez-vous à la console d'administration.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Point de contact**.
3. Sélectionnez le serveur point de contact existant que vous souhaitez utiliser comme base pour votre nouveau serveur point de contact. Vous devez sélectionner un profil pour WebSEAL ou **WebSphere**.
4. Cliquez sur **Créer comme** pour ouvrir le panneau de bienvenue de l'assistant de profil de point de contact.
5. Cliquez sur **Suivant** pour afficher le panneau Nom de profil. Il affiche les informations issues du profil sur lequel vous basez votre nouveau serveur point de contact.
6. Entrez le nom du profil.
7. (Facultatif) Entrez une description.
8. Cliquez sur **Suivant**. Le panneau d'ouverture de session s'affiche.
9. Acceptez les entrées par défaut pour les rappels de connexion, les paramètres pour chaque rappel, ainsi que l'ordre dans lequel ils sont utilisés.
10. Cliquez sur **Suivant**.
11. Acceptez les entrées par défaut du panneau Fermeture de session.
12. Cliquez sur **Suivant**.
13. Acceptez les entrées par défaut du panneau ID local.
14. Cliquez sur **Suivant**.
15. Cliquez sur **Ajouter des paramètres** dans la section Paramètres d'appel du panneau Authentification.
16. Entrez `authentication.macros` dans Nom.

17. Entrez les macros que vous souhaitez utiliser dans Valeurs. Pour spécifier plusieurs valeurs et séparer les macros, placez une barre oblique inversée (\) et une virgule entre les valeurs. Par exemple : `%FEDID%\,%FEDNAME%\, %PARTNERID%`
18. Cliquez sur **Suivant** pour afficher le panneau Récapitulatif. Ce panneau affiche la liste de tous les rappels et des paramètres que vous avez spécifiez au cours des étapes précédentes.
19. Cliquez sur **Terminer** pour achever la configuration ou cliquez sur **Précédent** pour revenir aux panneaux précédents et modifier vos sélections.
20. Cliquez sur le **portlet Domaine en cours**.
21. Cliquez sur **Charger les modifications de configuration dans l'environnement d'exécution de Tivoli Federated Identity Manager**.

Que faire ensuite

«Activation d'un serveur point de contact», à la page 818

Transmission d'un élément de demande SAML au serveur point de contact

Utilisez un paramètre de rappel pour transmettre des éléments de demande SAML spécifiques au serveur point de contact. Transmettez des éléments de demande SAML spécifiques en plus d'autres macros.

Dans un événement de connexion unique typique, des attributs sont transmis en tant que paramètres de la chaîne de requête dans l'URL de redirection. Cette fonction utilise un paramètre de rappel en plus des macros prises en charge existantes. Pour plus d'informations sur la personnalisation d'un formulaire de connexion d'authentification, voir Personnalisation d'un formulaire de connexion d'authentification pour une connexion unique.

Le transfert d'attributs spécifiques à partir d'une requête SAML nécessite le paramètre **`extended.authentication.macros`**. Sa valeur est un ensemble de paires clé-valeur séparées par des barres obliques (\). Pour chaque paire, la clé et la valeur sont séparées par le signe =.

Important : Ne placez pas de blanc avant et après les caractères \ et =.

Chaque paire représente un attribut transmis au point de contact. Chaque attribut est transmis au point de contact en tant que paramètre de la chaîne de requête.

La clé d'une paire est composée de deux parties :

- La première partie est le nom du protocole dans lequel la paire est applicable.
- La seconde partie est le nom du paramètre de la chaîne de requête.

Un point (.) sépare les deux parties. La valeur d'une paire correspond à la méthode spécifique au protocole pour la sélection de l'attribut.

L'exemple suivant présente le paramètre de rappel `extended.authentication.macros` au format BNF (Backus-Naur Form) :

```
<callback-parameter-value>    ::= <key-value-pairs>
<key-value-pairs>             ::= <key-value-pair> | <key-value-pair> "\," <key-value-pairs>
<key-value-pair>              ::= <key> "=" <value>
<key>                         ::= <protocol-name> "." <query-string-parameter-name>
```

```
<protocol-name>              ::= name of the protocol where the pair is applicable
<query-string-parameter-name> ::= name of the query string parameter
<value>                      ::= method for selecting the attribute/element
```

Remarque : Actuellement, seul le protocole SAML 2.0 est pris en charge. Le nom du protocole est `SAML20`. La méthode de sélection de l'attribut est une expression de chemin XPath. Actuellement, seule la forme canonique XPath 1.0 est prise en charge.

L'exemple suivant du paramètre de rappel présente un modèle de valeur pour le paramètre :

extended.authentication.macros

Value:

```
SAML20.AssertionConsumerServiceURL=/samlp:AuthnRequest/@AssertionConsumerServiceURL\,
SAML20.AuthnRequestAttributes=/samlp:AuthnRequest/@*\,
SAML20.Issuer=/samlp:AuthnRequest/saml:Issuer\,
SAML20.AuthnRequestElements=/samlp:AuthnRequest/*
```

Chaque attribut est transmis au point de contact en tant que paramètre de la chaîne de requête. La valeur de la seconde partie de la clé d'une paire est le nom du paramètre de la chaîne de requête. La valeur d'une paire sélectionne l'attribut qui est transmis au point de contact.

Si un seul attribut est sélectionné, la forme canonique de cet attribut est chiffrée à l'aide de BASE-64. Le résultat est utilisé comme valeur du paramètre de la chaîne de requête. Si plusieurs attributs sont sélectionnés, la représentation de chaîne de chaque attribut est chiffrée à l'aide de BASE-64. Les chaînes chiffrées sont concaténées avec une virgule (,) comme séparateur. Le résultat obtenu correspond à la valeur du paramètre de la chaîne de requête. Seule la forme canonique XML-C14 spécifiée dans World Wide Web Consortium est prise en charge.

Le nom du paramètre de la chaîne de requête et la valeur du paramètre de la chaîne de requête sont tous deux chiffrés dans l'URL avant d'être ajoutés dans l'URL de redirection.

Chapitre 50. Personnalisation des pages d'événement de connexion unique

Tivoli Federated Identity Manager génère des fichiers qui s'affichent à la suite d'événements survenus lors des demandes de connexion unique. Il se peut que la réponse affichée corresponde à un formulaire (par exemple, lorsque des informations de connexion sont requises) ou à une instruction d'erreur ou d'information concernant une condition qui s'est produite lors du traitement de la requête.

Vous avez la possibilité de personnaliser les pages d'événement en procédant comme suit :

- Modification de leur aspect ou de leur contenu.
- Spécification de l'environnement local ou linguistique utilisé lors de l'affichage des pages.

Avant de poursuivre la personnalisation, il convient que vous ayez une connaissance approfondie de la manière dont les pages d'événement sont générées et affichées. Voir «Génération des pages d'événement».

Génération des pages d'événement

Les pages d'événement sont affichées à la suite d'événements survenus lors des demandes de connexion unique. Elles contiennent généralement un formulaire (tel qu'une invite d'informations relatives au nom d'utilisateur et au mot de passe) ou du texte (tel qu'un message d'information ou d'erreur).

Les pages d'événement sont des pages dynamiques générées par Tivoli Federated Identity Manager à l'aide des informations suivantes :

Fichiers modèles
Il s'agit de fichiers XML ou HTML fournis avec Tivoli Federated Identity Manager et contenant des éléments tels que des zones, du texte ou des graphiques, et parfois de macros remplacées par des informations propres à la requête afin de répondre à cette dernière.

Identificateurs de page
Informations d'événement correspondant à un ou plusieurs fichiers modèles. Chaque identificateur de page correspond à une condition d'événement spécifique, telle qu'une erreur spécifique ou une condition dans laquelle un message ou un formulaire doit être affiché. Pour créer une page d'événement, les identificateurs de page sont mappés vers un ou plusieurs fichiers modèles. La fonction de mappage permet à plusieurs identificateurs de page de désigner le même fichier modèle.

Catalogue de messages
Texte utilise pour remplacer les macros dans les fichiers modèles.

Lorsqu'une requête est reçue, la page de réponse appropriée est générée comme suit :

1. La requête est traitée et une réponse à un événement est requise.
2. Les fichiers modèles et les identificateurs de page sont lus à partir du système de fichiers.

3. Les macros des fichiers modèles sont remplacées par des valeurs appropriées pour la réponse demandée.
4. Aucune page d'événement n'est générée.
5. La page d'événement générée s'affiche.

Pour plus d'informations sur les relations entre les identificateurs de page et les fichiers modèles, reportez-vous à la rubrique «Identificateurs de page et fichiers modèle».

Identificateurs de page et fichiers modèle

Un identificateur de page spécifie un événement et chaque événement correspond à un ou plusieurs *fichiers modèle*. Certains identificateurs de page sont propres à la spécification (par exemple, SAML 1.x), tandis que d'autres sont généraux.

Pour modifier le texte, les graphiques, ou d'autres éléments de la page qui est affichée pour un événement, exécutez les tâches suivantes :

1. Modifiez le fichier modèle ou copiez un fichier modèle.
2. Utilisez la copie comme base d'un nouveau fichier.
3. Mappez l'événement vers ce nouveau fichier.

Identificateurs de page généraux et fichiers modèles correspondants

Tableau 161. Identificateurs de page généraux et fichiers modèles correspondants

Identificateur de page (événement)	Description	Fichier modèle
/proper/errors/noprotdet	S'affiche lorsque le protocole est inconnu	/proper/errors/noprotdet.html
/proper/errors/missing_component	S'affiche lorsque le protocole est inconnu	/proper/errors/ missingcomponent.html
/proper/errors/protocol_error	S'affiche lorsqu'un module de protocole émet une exception	/proper/errors/protocol_error.html
/proper/errors/need_authentication	S'affiche quand les informations d'URL initiales sont introuvables dans la session utilisateur.	/proper/errors/ need_authentication.html
/proper/errors/access_denied	S'affiche lors d'un refus d'accès.	/proper/errors/access_denied.html
/proper/errors/missing-initial-url.html	S'affiche quand les informations d'URL initiales sont introuvables dans la session utilisateur.	/proper/errors/allerror.html
/proper/errors/unauth-access-to-waspoc-delegate.html	S'affiche lorsqu'un accès au protocole de délégation du point de contact WebSphere a eu lieu sans une authentification appropriée.	/proper/errors/allerror.html
/proper/login/formlogin.html	S'affiche lors de l'utilisation d'une authentification par formulaire. **Avertissement :** Ne modifiez pas la valeur d'action et les noms de paramètre de la page POST. Ils doivent rester inchangés pour que la page fonctionne correctement.	/proper/login/formlogin.html

Tableau 161. Identificateurs de page généraux et fichiers modèles correspondants (suite)

Identificateur de page (événement)	Description	Fichier modèle
/proper/login/formloginerror.html	S'affiche lorsqu'une erreur se produit lors de l'utilisation du fichier formlogin.html. Pour plus d'informations, voir «Personnalisation du formulaire de connexion», à la page 106.	/proper/login/formloginerror.html
/proper/genericpoc/ login_success.html	S'affiche lorsque l'implémentation du point de contact générique effectue une connexion avec succès sans adresse URL cible.	/proper/login/login_success.html
/proper/waspoc/login_success.html	S'affiche lorsque l'implémentation du point de contact WebSphere effectue une connexion avec succès sans adresse URL cible.	/proper/login/login_success.html
/proper/waspoc/login_failure.html	S'affiche lorsqu'une erreur se produit lors d'une connexion via l'implémentation de point de contact WebSphere.	/proper/login/login_failure.html

Identificateurs de page SAML 1.x et fichiers modèles correspondants

Tableau 162. Identificateurs de page SAML 1.x et fichiers modèles correspondants

Identificateur de page (événement)	Description	Fichier modèle
/saml/invalid_request.html	S'affiche lorsqu'une requête n'est pas valide.	/saml/allerror.html
/saml/unknown_sp.html	S'affiche lorsqu'un fournisseur de services inconnu a été détecté.	/saml/allerror.html
/saml/unknown_ip.html	S'affiche lorsqu'un fournisseur d'identité inconnu a été détecté.	/saml/allerror.html
/saml/invalid_ip_request.html	S'affiche lorsqu'un fournisseur d'identité est à l'origine d'une requête non valide.	/saml/allerror.html
/saml/unauth_user.html	S'affiche lorsque l'utilisateur en cours n'a pas été authentifié.	/saml/allerror.html
/saml/cannot_exchange_for_sp.html	S'affiche lorsqu'une erreur est détectée lors de l'échange de jeton.	/saml/allerror.html
/saml/no_ip_post_page.html	S'affiche lorsque le fournisseur d'identité n'a pas de page POST.	/saml/allerror.html
/saml/no_return_token.html	S'affiche lorsqu'il n'existe pas de jeton de retour.	/saml/allerror.html
/saml/ip_post_to_sp.html	Affiche la page HTML POST lorsque le fournisseur d'identité envoie la réponse SAML au fournisseur de services.	/saml/allerror.html
/saml/invalid_response.html	S'affiche lorsqu'un message de réponse non valide a été détecté.	/saml/allerror.html
/saml/ip_response_invalid.html	S'affiche lorsqu'une réponse du fournisseur d'identité n'est pas valide.	/saml/allerror.html

Tableau 162. Identificateurs de page SAML 1.x et fichiers modèles correspondants (suite)

Identificateur de page (événement)	Description	Fichier modèle
/saml/ cannot_exchange_for_resource.html	S'affiche lorsqu'une erreur est détectée lors de l'échange de jeton.	/saml/allerror.html
/saml/missing_context_attribute.html	S'affiche lorsque l'attribut de contexte requis n'est pas représenté.	/saml/allerror.html
/saml/ missing_config_parameter.html	S'affiche lorsqu'un élément de configuration SPS requis est absent.	/saml/allerror.html
/saml/ could_not_retrieve_assertion.html	S'affiche lorsque fournisseur de services ne peut pas extraire l'assertion de la réponse ou du canal de retour SOAP.	/saml/allerror.html
/saml/ could_not_perform_local_auth.html	S'affiche lorsqu'une erreur est détectée lors du renvoi de l'en-tête EAI.	/saml/allerror.html
/saml/ could_not_create_signed_request.html	S'affiche lorsque la génération d'une requête d'assertion SAML est impossible.	/saml/allerror.html
/saml/sp_missing_target.html	S'affiche au niveau du fournisseur de services si la requête initiale adressée au noeud final WAYF ne contient pas de paramètre TARGET.	/saml/allerror.html
/saml/ error_parsing_soap_response.html	S'affiche si une erreur est détectée lorsque le fournisseur de services tente d'extraire l'assertion du noeud final SOAP du fournisseur d'identité.	/liberty/ error_parsing_soap_response.html
/saml/unknown_ip_wayf.html	S'affiche lorsque le cookie WAYF contient un ID de fournisseur d'identité qui n'est pas configuré sur la fédération.	/saml/allerror.html

Identificateurs de page SAML 2.0 et fichiers modèles correspondants

Tableau 163. Identificateurs de page SAML 2.0 et fichiers modèles correspondants

Identificateur de page	Description	Fichiers modèles
/saml20/error_building_msg.html	S'affiche en cas d'erreur de génération des messages SAML 2.	/saml20/error_building_msg.html
/saml20/ error_missing_config_param.html	S'affiche lorsqu'un paramètre de configuration non valide est détecté pendant l'exécution.	/saml20/ error_missing_config_param.html
/saml20/error_sending_msg.html	S'affiche en cas d'erreur d'envoi des messages SAML 2.	/saml20/error_sending_msg.html
/saml20/error_validating_msg.html	S'affiche en cas d'erreur de validation des messages SAML 2.	/saml20/error_validating_msg.html
/saml20/error_validating_art.html	S'affiche en cas d'erreur de validation des artefacts SAML 2.	/saml20/error_validating_art.html
/saml20/invalid_msg.html	S'affiche en cas d'erreur de validation des messages SAML 2.	/saml20/invalid_msg.html
/saml20/invalid_art.html	S'affiche en cas d'erreur de validation des artefacts SAML 2.	/saml20/invalid_art.html

Tableau 163. Identificateurs de page SAML 2.0 et fichiers modèles correspondants (suite)

Identificateur de page	Description	Fichiers modèles
/saml20/authn_failed.html	S'affiche en cas d'échec de l'authentification SAML 2.	/saml20/authn_failed.html
/saml20/logout_failed.html	S'affiche en cas d'échec de la déconnexion.	/saml20/logout_failed.html
/saml20/art_exchange_failed.html	S'affiche en cas d'échec de l'échange d'un artefact SAML contre une réponse.	/saml20/art_exchange_failed.html
/saml20/nimgmt_update_failed.html	S'affiche en cas d'échec de la mise à jour de la gestion des identificateurs de nom.	/saml20/nimgmt_update_failed.html
/saml20/ nimgmt_terminate_failed.html	S'affiche en cas d'échec de l'arrêt de la gestion des identificateurs de nom.	/saml20/ nimgmt_terminate_failed.html
/saml20/ error_validating_msg_signature.html	S'affiche en cas d'erreur de validation de signatures de messages SAML 2.	/saml20/ error_validating_msg_signature.html
/saml20/error_decrypting_msg.html	S'affiche en cas d'erreur de déchiffrement des messages SAML 2.	/saml20/error_decrypting_msg.html
/saml20/error_parsing_msg.html	S'affiche en cas d'erreur d'analyse syntaxique des messages SAML 2.	/saml20/error_parsing_msg.html
/saml20/error_parsing_art.html	S'affiche en cas d'erreur d'analyse syntaxique des artefacts SAML 2.	/saml20/error_parsing_art.html
/saml20/invalid_init_msg.html	S'affiche en cas d'erreur de validation des messages SAML 2.	/saml20/invalid_init_msg.html
/saml20/ error_validating_init_msg.html	S'affiche en cas d'erreur de validation des messages SAML 2.	/saml20/ error_validating_init_msg.html
/saml20/logout_success.html	S'affiche en cas de succès de la déconnexion.	/saml20/logout_success.html
/saml20/logout_partial_success.html	S'affiche en cas de succès de la déconnexion partielle.	/saml20/logout_partial_success.html
/saml20/ nimgmt_terminate_success.html	S'affiche en cas de succès de l'arrêt de la gestion des identificateurs de nom.	/saml20/ nimgmt_terminate_success.html
/saml20/ nimgmt_update_success.html	S'affiche en cas de succès de la mise à jour de la gestion des identificateurs de nom.	/saml20/ nimgmt_update_success.html
/saml20/consent_to_federate.html	S'affiche et invite un utilisateur à donner son accord pour une fédération.	/saml20/consent_to_federate.html
/saml20/saml_post_artifact.html	S'affiche pour l'envoi d'artefacts SAML 2.0 pour les profils POST.	/saml20/saml_post_artifact.html
/saml20/saml_post_request.html	S'affiche pour l'envoi de requêtes SAML 2.0 pour les profils POST.	/saml20/saml_post_request.html
/saml20/saml_post_response.html	S'affiche pour l'envoi des réponses SAML 2.0 pour les profils POST.	/saml20/saml_post_response.html
/saml/ could_not_create_signed_request.html	S'affiche lorsque la génération d'une requête d'assertion SAML est impossible.	/saml/allerror.html

Tableau 163. Identificateurs de page SAML 2.0 et fichiers modèles correspondants (suite)

Identificateur de page	Description	Fichiers modèles
/saml/sp_missing_target.html	Utilisé au niveau du fournisseur de services si la requête initiale adressée au noeud final WAYF ne contient pas de paramètre TARGET.	/saml/allerror.html

Identificateurs de page Liberty

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Tableau 164. Identificateurs de page Liberty

Identificateur de page	Description
/liberty/error_parsing_soap_response.html	Indique que la réponse SOAP ne peut pas être analysée.
/liberty/fed-terminate-success.html	S'affiche lorsque l'arrêt a abouti.
/liberty/lib-cant-modify-alias.html	Indique que la modification d'alias a échoué
/liberty/lib-fed-consent.html	Envoyé pour demander l'accord de l'utilisateur pour une fédération
/liberty/lib-fed-post-request.html	Formulaire utilisé pour envoyer une requête d'authentification.
/liberty/lib-fed-post.html	Formulaire utilisé pour envoyer une réponse
/liberty/lib-internal-error-page.html	Envoyé en cas d'erreur si aucune autre message ne peut être envoyé
/liberty/lib-ipi-consent.html	Permet de demander l'accord de l'utilisateur pour une présentation du fournisseur d'identité aux fournisseurs de services
/liberty/lib-ipi-post.html	Signale la réussite de la présentation de fournisseur d'identité
/liberty/lib-login-failed-page.html	Inutilisé actuellement.
/liberty/lib-logout-failed-page.html	Envoyé à l'utilisateur par le fournisseur d'identité en cas d'échec de la déconnexion pour une raison ou pour une autre
/liberty/lib-logout-page.html	Indique à l'utilisateur destinataire toutes les résiliations de session après déconnexion
/liberty/lib-logout-success-page.html	Envoyé à l'utilisateur par le fournisseur d'identité pour signaler que la déconnexion a abouti
/liberty/logoutFailure.gif	Image indiquant l'échec de la déconnexion si la technique de déconnexion unique HTTP GET est utilisée
/liberty/logoutSuccess.gif	Image indiquant la réussite de la déconnexion si la technique de déconnexion unique HTTP GET est utilisée
/liberty/lib-message-timestamp-failure.html	Envoyé si le temps d'émission dépasse la plage autorisée
/liberty/lib-no-fed-exists.html	Envoyé lorsqu'aucune fédération n'existe
/liberty/lib-no-liberty-assertion.html	Indique que la réponse ne comporte aucune assertion

Tableau 164. Identificateurs de page Liberty (suite)

Identificateur de page	Description
/liberty/lib-no-local-login.html	Indique l'échec de la connexion locale
/liberty/lib-no-service-available.html	Indique l'absence de service d'alias ou d'assertion
/liberty/lib-register-name-identifier-success.html	Indique que l'enregistrement d'un identificateur de nom a abouti
/liberty/lib-request-id-not-matching-resp.html	Indique qu'une réponse ne correspond à aucune requête connue
/liberty/lib-sig-validation-failure.html	Inutilisé actuellement.
/liberty/lib-version-mismatch.html	Inutilisé actuellement.
/pages/itfim/wayf/wayf-html.html	Réponse WAYF HTML

Identificateurs de page WS-Federation

Tableau 165. Identificateurs de page WS-Federation

Identificateur de page	Description
/wsfederation/cannot_exchange_for_resource.html	Indique que la demande IP WS-Trust a échoué sur le fournisseur de services.
/wsfederation/cannot_exchange_for_sp.html	Indique que le fournisseur d'identité ne peut pas échanger un jeton contre le fournisseur de services.
/wsfederation/cannot_local_auth.html	Utilisé lorsque le fournisseur de services ne peut pas valider un jeton.
/wsfederation/invalid_ip_response.html	Indique que le fournisseur de services ne peut pas comprendre une réponse du fournisseur d'identité.
/wsfederation/invalid_request.html	Il ne s'agit pas d'une requête WS-Federation.
/wsfederation/invalid_sp_request.html	S'affiche lorsqu'une requête n'est pas valide.
/wsfederation/ip_post_to_sp.html	Utilisé par WS-Federation pour l'envoi d'informations du fournisseur d'identité au fournisseur de services.
/wsfederation/no_ip_post_page.html	S'affiche lorsque le fournisseur d'identité n'a pas de page post.
/wsfederation/no_return_token.html	Indique que le fournisseur d'identité ne peut pas trouver de jeton à renvoyer au fournisseur de services.
/wsfederation/signout_cleanup_failed.html	Inutilisé actuellement.
/wsfederation/signout_cleanup_failed_no_auth.html	Utilisé lorsque la déconnexion WS-Federation a échoué parce qu'un utilisateur n'a pas été authentifié.
/wsfederation/signout_cleanup_to_sp.html	Le processus de déconnexion unique de WS-Federation l'utilise pour déclencher des déconnexions uniques sur les fournisseurs de services.
/wsfederation/signout_successful.html	Utilisé lorsque la déconnexion WS-Federation a abouti.
/wsfederation/sp_ip_returned_fault.html	Indique l'erreur renvoyée par le fournisseur d'identité au fournisseur de services.

Tableau 165. Identificateurs de page WS-Federation (suite)

Identificateur de page	Description
/wsfederation/unauth_user.html	Indique que l'utilisateur n'a pas été authentifié sur ce fournisseur d'identité.
/wsfederation/unknown_ip_wayf.html	Indique que le fournisseur de services ne peut pas déterminer le fournisseur d'identité.
/wsfederation/unknown_sp.html	Indique que le fournisseur de services est inconnu du fournisseur d'identité.

Identificateurs de page indépendants de bas niveau

Tableau 166. Identificateurs de page indépendants

Identificateur de page	Description
/proper/errors/cannot_process	Utilisé pour les erreurs internes non spécifiées.
/proper/errors/missing_component	S'affiche lorsque le protocole est inconnu.
/proper/errors/noprotdet	S'affiche lorsque le protocole est inconnu.
/proper/errors/not_started	Utilisé lorsque le service SPS n'est pas en cours d'exécution, ce qui indique généralement une erreur de configuration quelconque.
/proper/errors/protocol_error	S'affiche lorsqu'un module de protocole émet une exception.
/pages/itfim/wayf/error-no-ips.html	Signale qu'aucun fournisseur d'identité n'est présent et qu'aucun traitement WAYF ne peut donc être effectué.
/pages/itfim/wayf/error-missing-template.html	Utilisé lorsque le programme ne trouve aucun modèle de page WAYF.
/pages/itfim/wayf/error-invalid-template.html	Utilisé lorsque la page WAYF n'est pas valide.
/pages/itfim/wayf/wayf-html.html	S'affiche lorsqu'une fédération a plusieurs fournisseurs d'identité et que le paramètre de chaîne de requête ITFIM_WAYF_IDP ou le cookie WAYF n'est pas présent.

Emplacement des fichiers modèle

Par défaut, les fichiers modèles sont stockés dans le répertoire suivant :

AIX

`/usr/IBM/FIM/pages/`*`environnement_local`*`/`

Linux ou Solaris

`/opt/IBM/FIM/pages/`*`environnement_local`*`/`

Windows

`C:\Program Files\IBM\FIM\pages\`*`environnement_local`*`\`

Le sous-répertoire de l'environnement local est spécifique à chaque région géographique ou environnement linguistique des fichiers modèle. Le répertoire de

l'environnement local par défaut est intitulé `C` et tous les fichiers sont en anglais. Si un module de langue a été installé, des environnements locaux supplémentaires sont disponibles.

Les fichiers modèles sont publiés depuis leurs sous-répertoires par défaut vers les répertoires de WebSphere Application Server. Voir «Publication des mises à jour», à la page 806.

Avertissement : Si vous devez modifier les fichiers modèle, effectuez la modification sur le serveur Tivoli Federated Identity Manager. *Ne modifiez pas* les fichiers dans les répertoires de WebSphere Application Server.

Contenu des fichiers modèle

Les fichiers modèles HTML peuvent contenir des macros qui sont remplacées par des informations spécifiques au contexte qui sont extraites lorsque la page de réponse est générée et renvoyée. Si votre fichier modèle contient, par exemple, la macro `@EXCEPTION_MSG@`, un message d'exception est inclus dans la page de réponse.

La présence d'une macro dans un fichier modèle ne garantit pas l'affectation d'une valeur réelle à la macro lors de la génération de la page de réponse. Une valeur doit être définie pour la macro lors de l'élaboration de la page, afin que la macro puisse renvoyer une valeur.

Lors de la personnalisation d'un fichier modèle HTML, utilisez uniquement les macros définies dans le fichier modèle. Si vous ajoutez de nouvelles macros dans le fichier modèle, les valeurs des macros ajoutées ne sont pas renvoyées lors de la génération de la page de réponse définitive.

Les macros utilisent le format suivant :

```
@MACRO@
```

Où *MACRO* représente le nom de la macro, par exemple, `@EXCEPTION_MSG@`

Les macros suivantes sont utilisées dans les fichiers modèles.

Tableau 167. Macros utilisées dans les fichiers modèles

Macro de substitution	Description abrégée
@ACTION@	L'action correspond à l'adresse URL vers laquelle le formulaire contenant la réponse POST est envoyé. Utilisée dans un réponse POST HTML envoyée par un fournisseur d'identité à un navigateur pour une demande de service de protocole de connexion unique.
@CAUSE@	Informations relatives à la cause de l'erreur.
@DETAIL@	Informations supplémentaires relatives à une erreur ou une exception qui s'est produite dans le cadre du traitement d'une requête. Dans la mesure où du texte supplémentaire n'est pas toujours disponible, même si la macro @DETAIL@ est utilisée dans un fichier modèle HTML, il n'y a aucune garantie que les macros fournissent du texte supplémentaire concernant l'exception.

Tableau 167. Macros utilisées dans les fichiers modèles (suite)

Macro de substitution	Description abrégée
@EXCEPTION_MSG@	Message de texte décrivant une exception qui s'est produite lors du traitement d'une requête.
@EXCEPTION_STACK@	Pile complète d'une exception qui s'est produite lors du traitement d'une requête.
@FEDERATION_DISPLAY@	Nom de la fédération actuelle, c'est-à-dire celle qui est en cours d'utilisation.
@FEDERATION_ID@	Identificateur unique de la fédération actuelle.
@PARTNER_ID@	Protocole de connexion unique de fédération d'un partenaire de la fédération.
@REQ_ADDR@	Adresse IP (Internet Protocol) du noeud final qui a demandé une action de fédération.
@RESPONSE@	Utilisée dans une réponse POST HTML d'un fournisseur d'identité, remplacée par la réponse SAML.
@SAMLSTATUS@	Collection des valeurs d'état SAML reçues au cours du traitement d'une action de connexion unique.
@SOAP_ENDPOINT@	Adresse URL du noeud final SOAP servant à extraire l'assertion au moyen d'un artefact SAML.
@TARGET@	Permet de fournir la cible de fournisseur de services dans un réponse POST HTML envoyée par un fournisseur d'identité à un navigateur pour une demande de service de protocole de connexion unique.
@TIMESTAMP@	Valeur de l'heure en cours.
@TOKEN:form_action@	Adresse URL vers laquelle le formulaire contenant le message POST est envoyé lors d'une liaison POST.
@TOKEN:IPDisplayName@	Nom unique du fournisseur d'identité.
@TOKEN:IPProviderID@	Identificateur unique du fournisseur d'identité.
@TOKEN:PartnerID@	Identificateur unique du partenaire.
@TOKEN:RelayState@	Valeur RelayState du protocole SAML.
@TOKEN:SamlMessage@	Message SAML codé en base 64 et envoyé dans un formulaire.
@TOKEN:SPDisplayName@	Nom unique du fournisseur de services.
@TOKEN:SPProviderID@	Identificateur unique du fournisseur de services.
@TOKEN:UserName@	Nom de l'utilisateur authentifié qui a soumis l'action de connexion unique.
@WAYF_FEDERATION_DISPLAY_NAME@	Nom affiché de la fédération en cours, comme présenté dans la console. Utilisée dans une page présentant une demande d'authentification WAYF (Where Are You From : D'où venez-vous) et demandant à un utilisateur de choisir un fournisseur d'identité.
@WAYF_FEDERATION_ID@	Identificateur de la fédération en cours dans le fichier de configuration. Utilisée dans une page présentant une demande d'authentification WAYF et demandant à un utilisateur de sélectionner un fournisseur d'identité.

Tableau 167. Macros utilisées dans les fichiers modèles (suite)

Macro de substitution	Description abrégée
@WAYF_FORM@	Informations d'identification relatives au formulaire WAYF HTML qui est présenté à un utilisateur pour acquérir des informations de fournisseur d'identité dans une action SPS, où le fournisseur d'identité du demandeur n'a pas encore été déterminé (il ne figure pas encore dans le cookie présenté).
@WAYF_FORM_ACTION@	Noeud final du service de protocole de connexion unique ; il s'agit de l'adresse initialement demandée (URL). Utilisée dans une page présentant une demande d'authentification WAYF et demandant à un utilisateur de sélectionner un fournisseur d'identité.
@WAYF_FORM_METHOD@	Méthode HTTP utilisée dans une demande qui a donné lieu à un formulaire WAYF sur une page qui invite l'utilisateur à entrer les informations relatives au fournisseur d'identité. La méthode peut être GET, POST ou HEAD.
@WAYF_FORM_PARAM_ID@	Identificateur du paramètre de formulaire pour le fournisseur d'identité en cours ; il s'agit généralement du nom de cookie configuré. Utilisé dans une page présentant une demande d'authentification WAYF et demandant à un utilisateur de sélectionner un fournisseur d'identité.
@WAYF_HIDDEN_NAME@	Nom d'un des paramètres initiaux d'une demande qui donne lieu à un formulaire WAYF ; cette macro est utilisée dans une page qui invite l'utilisateur à entrer les informations relatives au fournisseur d'identité.
@WAYF_HIDDEN_VALUE@	Valeur d'un des paramètres initiaux d'une demande qui donne lieu à un formulaire WAYF ; cette macro est utilisée dans une page qui invite l'utilisateur à entrer les informations relatives au fournisseur d'identité.
@WAYF_IP_DISPLAY_NAME@	Nom affiché configuré du fournisseur d'identité en cours dans une page présentant une demande d'authentification WAYF.
@WAYF_IP_ID@	ID configuration du fournisseur d'identité en cours dans une page présentant une demande d'authentification WAYF.

Modèle de page pour la page WAYF

La page WAYF (Where Are You From) est utilisée au niveau du fournisseur de service. Elle permet aux utilisateurs de sélectionner leur fournisseur d'identité si plusieurs d'entre eux sont configurés dans la fédération.

Lorsqu'un utilisateur arrive au niveau d'un fournisseur de service, un identificateur WAYF peut être envoyé via un cookie ou un paramètre de chaîne de requête avec la demande. L'ID d'entité du fournisseur d'identité est stocké en tant que valeur du cookie ou du paramètre de chaîne de requête. Si le cookie d'identificateur WAYF ou le paramètre de chaîne de requête n'est pas présent, la page WAYF s'ouvre.

Voici un exemple d'URL incluant le paramètre de chaîne de requête pour WAYF :

```
https://sp.host.com/FIM/sps/samlfed/saml20/
logininitial?RequestBinding=HTTPRedirect&ResponseBinding
=HTTPPost&ITFIM_WAYF_IDP=https://idp.host.com/FIM/sps/samlfed/saml20
```

Cet exemple est destiné à une URL de connexion unique SAML 2.0. Le nom du paramètre de chaîne de requête est `ITFIM_WAYF_IDP`. La valeur de l'ID de fournisseur d'identité est `https://idp.host.com/FIM/sps/samlfed/saml20`.

La page WAYF demande à l'utilisateur d'indiquer sa provenance. Si l'utilisateur n'est pas connecté à son fournisseur d'identité, il est invité à se connecter. Selon les attributs analysés, le fournisseur de service peut accorder ou refuser l'accès au service.

Les pages de modèle sont stockées par défaut dans le répertoire suivant :

<FIM_Install_Dir>`/pages/`*<locale>*`/pages/itfim/wayf`

Voir «Identificateurs de page indépendants de bas niveau», à la page 800 pour plus d'informations sur les modèles de pages WAYF.

Les administrateurs peuvent utiliser cette page sans modifications, mais peuvent dans certains cas souhaiter modifier le style du code HTML pour répondre aux exigences spécifiques de leur environnement de déploiement.

Ce fichier modèle prend en charge plusieurs macros de remplacement :

@WAYF_FORM_ACTION@
: Cette macro est remplacée par le noeud final de la requête originale. Cette macro n'appartient pas à la section répétable.

@WAYF_FORM_METHOD@
: Cette macro est remplacée par la méthode HTTP de la requête originale. Cette macro n'appartient pas à la section répétable.

@WAYF_FORM_PARAM_ID@
: Cette macro est remplacée par l'ID utilisé par l'action pour le fournisseur d'identité. Cette macro est répétée une fois pour chaque fournisseur d'identité.

@WAYF_IP_ID@
: Cette macro est remplacée par l'ID unique du fournisseur d'identité. Cette macro est répétée une fois pour chaque fournisseur d'identité.

@WAYF_IP_DISPLAY_NAME@
: Cette macro est remplacée par le nom d'affichage configuré du fournisseur d'identité. Cette macro est répétée une fois pour chaque fournisseur d'identité.

@WAYF_HIDDEN_NAME@
: Cette macro est remplacée par le nom du paramètre masqué. Cette macro est répétée une fois pour chaque paramètre de requête original, puis elle est masquée.

@WAYF_HIDDEN_VALUE@
: Cette macro est remplacée par la valeur du paramètre masqué. Cette macro est répétée une fois pour chaque paramètre de requête original, puis elle est masquée.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<!--
Le modèle html wayf présente les choix sous forme de boutons radio.
-->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
        <title>Where are you from</title>
    </head>
    <body style="background-color:#ffffff">
        <div>
            <!--
            Insert the federation ids here just so we can show some tokens
            [RPT federations]
                @WAYF_FEDERATION_ID@
                @WAYF_FEDERATION_DISPLAY_NAME@
            [ERPT federations]
            -->
            <form id="wayfForm" name="wayfForm"
                    action="@WAYF_FORM_ACTION@" method="@WAYF_FORM_METHOD@">
                <div>
                    <table>

                    [RPT ips]
                        <tr>
                            <td>
                                <input type="radio" id="@WAYF_FORM_PARAM_ID@"
                                    name="@WAYF_FORM_PARAM_ID@"
                                    value="@WAYF_IP_ID@"/>@WAYF_IP_DISPLAY_NAME@
                            </td>
                        </tr>
                    [ERPT ips]

                    </table>
                    <!-- the hidden inputs must be present -->

                    [RPT hidden]
                    <input type="hidden" name="@WAYF_HIDDEN_NAME@"
                        id="@WAYF_HIDDEN_NAME@"
                        value="@WAYF_HIDDEN_VALUE@"/ >
                    [ERPT hidden]

                </div>
                <input type="submit" name="submit" value="Submit" />
            </form>
        </div>
    </body>
</html>
```

Figure 78. Modèle de page wayf-html.html

Modification ou création des fichiers modèles

Pour personnaliser la présentation des pages d'événement, vous pouvez modifier les fichiers modèles ou en créer de nouveaux.

Avant de commencer

Avant de poursuivre cette procédure, assurez-vous de maîtriser le mode de génération des pages d'événement. Voir «Génération des pages d'événement», à la page 793.

Pourquoi et quand exécuter cette tâche

Avertissement : Modifiez les fichiers modèles dans le répertoire du serveur Tivoli Federated Identity Manager (selon la description ci-dessous). Une fois toutes les modifications terminées, publiez-les dans le répertoire de référentiel de configuration WebSphere Application Server. *N'éditez pas* ces fichiers dans le référentiel de configuration.

Procédure

1. Sélectionnez les pages d'événement que vous souhaitez modifier. Reportez-vous à la liste des événements et aux fichiers modèles correspondants dans la rubrique «Identificateurs de page et fichiers modèle», à la page 794.
2. Arrêtez le poste WebSphere Application Server sur lequel le composant d'exécution est installé. Exécutez la commande **stopServer**. Pour plus d'informations, consultez le centre de documentation de WebSphere.
3. Recherchez le fichier modèle qui correspond à la page d'événement que vous souhaitez modifier, ou effectuez une copie d'un fichier modèle existant et utilisez-la pour générer un nouveau fichier.

   Les fichiers modèles sont situés dans un sous-répertoire de l'environnement local spécifique au lieu géographique ou à la langue, sous le répertoire du fichier. Le sous-répertoire de l'environnement local par défaut est intitulé `C` et tous les fichiers sont en anglais. Si un module de langue a été installé, des environnements locaux supplémentaires sont disponibles. Vous pouvez également créer vos propres paramètres nationaux, comme décrit à la rubrique «Création d'un environnement local de page», à la page 807.

   Le répertoire par défaut contenant les fichiers modèles est le suivant :

   AIX
   `/usr/IBM/FIM/pages/`*`locale`*`/`

   Linux ou Solaris
   `/opt/IBM/FIM/pages/`*`environnement_local`*`/`

   Windows
   `C:\Program Files\IBM\FIM\pages\`*`locale`*`\`
4. Utilisez un éditeur de texte pour modifier ou créer des fichiers.
5. Sauvegardez les fichiers à l'emplacement approprié, c'est-à-dire par exemple dans le même répertoire que celui dans lequel vous les avez copiés ou édités.

Que faire ensuite

Une fois que vous avez terminé cette étape, poursuivez la publication des fichiers dans le référentiel de configuration, selon la procédure décrite dans la section «Publication des mises à jour».

Publication des mises à jour

Une fois l'ensemble des mises à jour et ajouts effectués dans les fichiers modèles, vous devez publier les fichiers dans le référentiel de configuration, afin de permettre leur affichage.

Procédure

1. Connectez-vous à la console de gestion.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Pages d'événement**.
3. Localisez les événements que vous souhaitez mapper avec les pages nouvelles ou actualisées.
4. Dans la page **Page HTML affichée** de chaque événement que vous modifiez, indiquez le chemin d'accès et le nom de fichier que vous souhaitez utiliser pour cet événement.
5. Cliquez sur **Valider**. Un message d'avertissement s'affiche pour vous indiquer que vous devez publier les fichiers mis à jour dans le référentiel de configuration.
6. Cliquez sur **Publier les pages** afin de publier immédiatement les modifications.

   Sinon, cliquez sur **Fermer** et, lorsque vous serez prêt à publier le moment venu, cliquez sur **Gestion de domaines** > **Gestion des noeuds d'exécution** et, dans le panneau Gestion des noeuds d'exécution, cliquez sur le bouton **Publier les pages**.

Création d'un environnement local de page

Les fichiers modèles utilisés pour générer des pages d'événement sont situés dans un sous-répertoire de l'environnement local spécifique au lieu géographique ou à la langue, sous le répertoire du fichier. Le sous-répertoire de l'environnement local par défaut est intitulé `C` et tous les fichiers sont en anglais. Des paramètres nationaux, ainsi que les langues correspondantes, sont également disponibles. En outre, vous pouvez créer votre propre environnement local.

Avant de commencer

Avant de poursuivre cette procédure, assurez-vous de maîtriser le mode de génération des pages d'événement. Voir «Génération des pages d'événement», à la page 793.

Procédure

1. Connectez-vous à la console de gestion.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Pages d'événement**. Le panneau Pages d'événement s'affiche.
3. Cliquez sur l'onglet **Environnement local de page** pour ouvrir le panneau correspondant.
4. Cliquez sur **Créer**. Un élément de liste représentant une marque de réservation est ajouté à la liste des environnements locaux de page, caractérisé par le nom **locale** et le répertoire principal **page_root**.
5. Entrez une abréviation d'environnement local pour remplacer la valeur **locale**.
6. Entrez le nom de répertoire de l'environnement local à la place de la valeur **page_root**.
7. Cliquez sur **Appliquer** ou sur **OK**. Un message d'avertissement s'affiche pour vous indiquer que vous devez publier les fichiers mis à jour dans le référentiel de configuration.
8. Cliquez sur **Publier** afin de publier immédiatement les modifications.

Sinon, cliquez sur **Fermer** et, lorsque vous serez prêt à publier le moment venu, cliquez sur **Gestion de domaines** > **Gestion des noeuds d'exécution** et, dans le panneau Gestion des noeuds d'exécution, cliquez sur le bouton **Publier les pages**.

Suppression d'un environnement local sur une page

Vous pouvez supprimer n'importe quel environnement local de page autre que l'environnement C par défaut installé en même temps que Tivoli Federated Identity Manager.

Pourquoi et quand exécuter cette tâche

La suppression d'un environnement local de page empêche l'affichage des pages dans cet environnement.

Procédure

1. Connectez-vous à la console de gestion.
2. Sélectionnez **Tivoli Federated Identity Manager** > **Gestion des domaines** > **Pages d'événement**. Le panneau correspondant s'ouvre.
3. Cliquez sur l'onglet **Environnement local de page** (Page Locale) pour ouvrir le panneau correspondant.
4. Dans la zone **Sélectionner**, choisissez le bouton jouxtant l'environnement local de page à supprimer. Pour obtenir une description des environnements locaux, reportez-vous à l'aide en ligne.
5. Cliquez sur **Supprimer**, puis sur **Appliquer** pour valider vos modifications et rester dans le portlet Environnement local de page, ou cliquez sur **OK** pour valider les modifications et quitter le portlet.

Personnalisation des modèles de page physique à usages multiples

Dans certains cas, vous devez personnaliser les modèles de page physique qui sont désignés par plusieurs identificateurs de page.

Pourquoi et quand exécuter cette tâche

Remarque : Le protocole Liberty est obsolète dans Tivoli Federated Identity Manager version 6.2.2.

Certains modèles de page physique sont désignés par plusieurs identificateurs de page dans le fichier `sps.xml`.

Par exemple : `<sps:PageIdentifierMapping name="/liberty/error_parsing_soap_response.html" location="/liberty/error_parsing_soap_response.html" />` et `<sps:PageIdentifierMapping name="/saml/error_parsing_soap_response.html" location="/liberty/error_parsing_soap_response.html" />`

Si les réponses SAML et Liberty doivent être différentes, modifiez les pages comme suit :

Procédure

1. Pour chaque environnement local affecté, copiez la page Liberty dans le répertoire SAML.

2. Modifiez les deux pages selon vos besoins.
3. Modifiez la seconde expression PageIdentifierMapping de l'exemple ci-dessus comme suit : `<sps:PageIdentifierMapping name="/saml/error_parsing_soap_response.html" location="/saml/error_parsing_soap_response.html"/>`
4. Publiez ces modifications comme expliqué dans la rubrique «Publication des mises à jour», à la page 806.

Personnalisation de l'accord pour fédérer la page pour SAML 2.0

Une *page Accord de fédération* est un formulaire HTML qui invite l'utilisateur à donner son accord pour rejoindre une fédération. Vous pouvez personnaliser la *page Accord de fédération* pour indiquer les informations demandées à un utilisateur.

Avant de commencer

Déterminez les valeurs que vous souhaitez utiliser pour la page d'accord de fédération.

Pourquoi et quand exécuter cette tâche

Lorsqu'un utilisateur accède à une fédération, il donne son accord pour la rejoindre. Le formulaire HTML `consent_to_federate.html` invite à donner cet accord. Vous pouvez personnaliser ce que le formulaire demande en ajoutant des valeurs d'accord. Ces valeurs indiquent la manière dont un utilisateur accepte de rejoindre une fédération et si les fournisseurs de service sont informés de cet accord. Les fournisseurs d'identité reçoivent les valeurs d'accord dans la réponse SAML 2.0.

Les valeurs suivantes déterminent la manière dont un utilisateur rejoint une fédération :

1 Un utilisateur accepte de rejoindre une fédération sans informer le fournisseur de service.

0 L'utilisateur refuse de rejoindre la fédération.

Une valeur URI
Un URI peut indiquer si l'utilisateur accepte de rejoindre une fédération et si vous souhaitez informer le fournisseur de service de l'accord. La table suivante répertorie et décrit les valeurs d'URI prises en charge.

Tableau 168. Valeurs d'accord prises en charge pour la réponse SAML 2.0

Valeur d'accord	URI	Description
Non spécifié	`urn:oasis:names:tc:SAML:2.0:consent: unspecified`	L'accord de l'utilisateur n'est pas spécifié.
Obtenu	`urn:oasis:names:tc:SAML:2.0:consent: obtained`	Indique que l'accord de l'utilisateur est acquis par l'émetteur du message.
Précédent	`urn:oasis:names:tc:SAML:2.0:consent: prior`	Indique que l'accord de l'utilisateur est acquis par l'émetteur du message avant l'action qui a initié le message.
Implicite	`urn:oasis:names:tc:SAML:2.0:consent: current-implicit`	Indique que l'accord de l'utilisateur est acquis de manière implicite par l'émetteur du message au lancement de ce dernier.

Tableau 168. Valeurs d'accord prises en charge pour la réponse SAML 2.0 (suite)

Valeur d'accord	URI	Description
Explicite	`urn:oasis:names:tc: SAML:2.0:consent: current-explicit`	Indique que l'accord de l'utilisateur est acquis de manière explicite par l'émetteur du message lors de l'instance à laquelle le message a été envoyé.
Non disponible	`urn:oasis:names:tc: SAML:2.0:consent: unavailable`	Indique que l'émetteur du message n'a pas pu obtenir l'accord de l'utilisateur.
Inapplicable	`urn:oasis:names:tc: SAML:2.0:consent: inapplicable`	Indique que l'émetteur du message n'a pas besoin d'obtenir ou rapporter l'accord de l'utilisateur.

Procédez comme suit pour personnaliser l'accord de fédération de page.

Important : Modifiez les fichiers modèles sur le serveur Tivoli Federated Identity Manager. Une fois toutes les modifications effectuées, vous pouvez les publier dans le répertoire de référence de configuration de WebSphere Application Server. **N'éditez pas** ces fichiers dans le référentiel de configuration.

Procédure

1. Utilisez la commande `stopServer` pour arrêter le serveur WebSphere Application Server sur lequelTivoli Federated Identity Manager est installé. Pour plus d'informations, voir le centre de documentation de WebSphere.
2. Utilisez un éditeur de texte pour accéder à `consent_to_federate.html`.

   Les fichiers modèles sont situés dans un sous-répertoire de l'environnement local spécifique au lieu géographique ou à la langue. Tous les fichiers sont en anglais. Si vous avez installé un module de langue, des environnements locaux supplémentaires sont disponibles. Le répertoire par défaut dépend du système d'exploitation.

   AIX `/usr/IBM/FIM/pages/locale/saml20/`

   Linux ou Solaris
   `/opt/IBM/FIM/pages/locale/saml20/`

   Windows
   `C:\Program Files\IBM\FIM\pages\locale\saml20\`
3. Ajoutez les valeurs d'accord appropriées pour votre fédération. Voir A propos de cette tâche pour une liste complète des valeurs.
4. Sauvegardez les fichiers à l'emplacement approprié. Il peut s'agir du répertoire dans lequel vous les avez modifiés.
5. Redémarrez WebSphere Application Server.

Exemple

L'exemple suivant présente une URI ajoutée avec une valeur d'accord `Obtenu`:

```
<input type="radio" checked name="Consent"
value="urn:urn:oasis:names:tc:SAML:2.0:consent:obtained"/>
Consent Obtained.br/>
```

Dans cet exemple, l'accord de l'utilisateur est acquis par l'émetteur du message.

Que faire ensuite

Publiez les fichiers dans le référentiel de configuration. Voir «Publication des mises à jour», à la page 806.

Chapitre 51. Développement d'un serveur point de contact personnalisé

Le serveur point de contact de votre environnement Tivoli Federated Identity Manager représente la première entité qui traite une requête visant à accéder à une ressource. Vous pouvez choisir l'une des options fournies pour un serveur point de contact, ou créer un serveur point de contact personnalisé.

Pourquoi et quand exécuter cette tâche

Un serveur point de contact personnalisé est constitué de plusieurs modules de rappel personnalisés d'ouverture qui définissent les paramètres d'ouverture de session, de fermeture de session, d'ID local et d'authentification.

Un serveur point de contact personnalisé peut constituer le choix approprié pour votre environnement si vous souhaitez intégrer une application d'authentification ou de gestion d'accès Web existante à Tivoli Federated Identity Manager.

Un serveur point de contact personnalisé peut s'avérer utile dans les situations suivantes :

- Si vous disposez d'un cookie de connexion unique existant, qui est utilisé sur l'ensemble de l'entreprise, vous pouvez mettre en oeuvre un serveur point de contact personnalisé utilisant un rappel SignIn qui définit le cookie du domaine de connexion unique conformément à votre stratégie de connexion unique.
- Si vous disposez d'un logiciel de gestion d'accès Web qui expose une interface API personnalisée en vue de certifier l'identité d'un utilisateur dans l'environnement, ou d'extraire l'utilisateur actuel pour les besoins de la requête.

  Vous pouvez mettre en oeuvre un serveur point de contact qui exécute un rappel d'identité local (afin d'extraire l'utilisateur lié à la transaction), ou mettre en oeuvre un serveur point de contact personnalisé utilisant un rappel SignIn pour certifier l'identité d'un utilisateur dans l'environnement, ou bien mettre en oeuvre un serveur point de contact qui exploite ces deux types de rappel.

La mise au point d'un serveur point de contact personnalisé nécessite une certaine expérience dans la programmation de modules de rappel, ainsi qu'une bonne connaissance des concepts de programmation de Tivoli Federated Identity Manager. Consultez les liens des documents developerWorks dans le centre de documentation à l'adresse http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc_6.2.2/ic/ic-homepage.html.

Une fois le travail de développement terminé, vous devez effectuer l'intégration de la solution à votre environnement Tivoli Federated Identity Manager en procédant comme suit :

Procédure

1. Publiez les plug-ins de rappel dans le module d'exécution Tivoli Federated Identity Manager. Voir «Publication des plug-ins de rappel», à la page 814.
2. Rassemblez les paramètres nécessaires pour la configuration de chacun des modules de rappel.

3. Créez un nouveau profil de serveur point de contact. Vous avez la possibilité de créer un profil nouveau, ou de réutiliser un profil existant comme base de votre nouveau profil de serveur point de contact. Voir l'une des rubriques suivantes :
   - «Création d'un nouveau serveur point de contact»
   - «Création d'un serveur point de contact comme un serveur existant», à la page 817
4. Activez le serveur point de contact. Voir «Activation d'un serveur point de contact», à la page 818.

Publication des plug-ins de rappel

Si vous avez développé les modules pour un serveur point de contact personnalisé, vous devez publier leurs plug-ins pour pouvoir les utiliser dans votre environnement Tivoli Federated Identity Manager.

Avant de commencer

Avant de poursuivre cette tâche, vérifiez que vous avez développé les modules de rappel appropriés pour votre serveur point de contact. Pour plus d'informations, voir Chapitre 51, «Développement d'un serveur point de contact personnalisé», à la page 813.

Procédure

1. Copiez les plug-ins de rappel dans le répertoire `/plugins` dans lequel vous avez installé Tivoli Federated Identity Manager. Par exemple, sous Windows, il s'agit du répertoire `/opt/IBM/FIM/plugins`.
2. Connectez-vous à la console.
3. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Gestion des noeuds d'exécution**. Le panneau Gestion des noeuds d'exécution s'affiche.
4. Cliquez sur **Publier les plug-ins**.

Que faire ensuite

Après avoir publié les plug-ins, vous pouvez poursuivre avec la création du profil de point de contact.

Création d'un nouveau serveur point de contact

Tivoli Federated Identity Manager fournit plusieurs options pour chaque serveur point de contact, suivant le rôle que vous tenez dans la fédération. De plus, vous avez la possibilité de développer votre propre serveur point de contact. Si vous avez développé votre propre serveur, vous devez l'ajouter à votre environnement via la console.

Avant de commencer

Avant d'ajouter le serveur point de contact personnalisé à votre environnement, vous devez :

- Publier chaque point personnalisé des plug-ins de rappel de contact sur le noeud d'exécution. Voir «Publication des plug-ins de rappel».
- Connaître le type de paramètres à utiliser, le cas échéant, et les valeurs correspondantes qui doivent être transmises à ces rappels.

Pourquoi et quand exécuter cette tâche

La procédure qui suit explique comment ajouter un serveur point de contact personnalisé autre que les serveurs point de contact déjà définis dans votre environnement. Si vous ajoutez un serveur point de contact personnalisé similaire à un autre serveur, appliquez la procédure de la rubrique «Création d'un serveur point de contact comme un serveur existant», à la page 817.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Cliquez sur **Créer**. Le panneau de bienvenue de l'assistant de profil de point de contact s'ouvre.
4. Vérifiez que vous avez exécuté les étapes prérequises.
5. Cliquez sur **Suivant**. Le panneau Nom de profil s'ouvre.
6. Indiquez le nom de profil de votre serveur point de contact personnalisé et, le cas échéant, une description.
7. Cliquez sur **Suivant**. Le panneau d'ouverture de session s'affiche.
8. Dans le panneau Ouverture de session, spécifiez les rappels de connexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.
   a. Sélectionnez une entrée dans la liste **Rappels disponibles**. Cliquez sur **Ajouter** pour incorporer l'entrée à la liste **Rappels utilisés**. Répétez cette étape afin d'ajouter tous les rappels dont vous avez besoin pour le serveur point de contact.
   b. Cliquez sur le bouton **Ajouter des paramètres**. Une section relative aux paramètres de rappel s'affiche pour chaque rappel figurant dans la liste Rappels utilisés. Des zones de paramètres comportant les valeurs par défaut `new key` et `new value` s'affichent.
   c. Ajoutez les paramètres de chaque rappel en remplaçant le nom et la valeur par défaut par les paramètres que vous souhaitez ajouter. Pour ajouter d'autres paramètres, cliquez sur **Créer**. Lorsque vous cliquez sur **Créer**, une nouvelle zone de paramètre contenant les valeurs par défaut est ajoutée à la liste des paramètres.
   d. Répétez les étapes précédentes jusqu'à ce que tous les paramètres aient été ajoutés à tous les rappels.
9. Cliquez sur **Suivant**. Le panneau Fermeture de session s'affiche.
10. Dans le panneau Fermeture de session, spécifiez les rappels de déconnexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.
    a. Sélectionnez une entrée dans la liste **Rappels disponibles**. Cliquez sur **Ajouter** pour incorporer l'entrée à la liste **Rappels utilisés**. Répétez cette étape afin d'ajouter tous les rappels dont vous avez besoin pour le serveur point de contact.
    b. Cliquez sur le bouton **Ajouter des paramètres**. Une section relative aux paramètres de rappel s'affiche pour chaque rappel figurant dans la liste Rappels utilisés. Des zones de paramètres comportant les valeurs par défaut `new key` et `new value` s'affichent.
    c. Ajoutez les paramètres de chaque rappel en remplaçant le nom et la valeur par défaut par les paramètres que vous souhaitez ajouter. Pour ajouter

d'autres paramètres, cliquez sur **Créer**. Lorsque vous cliquez sur **Créer**, une nouvelle zone de paramètre contenant les valeurs par défaut est ajoutée à la liste des paramètres.

d. Répétez les étapes précédentes jusqu'à ce que tous les paramètres aient été ajoutés à tous les rappels.

11. Cliquez sur **Suivant**. Le panneau ID local s'affiche.
12. Dans le panneau ID local, spécifiez les rappels d'ID local à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.
    a. Sélectionnez une entrée dans la liste **Rappels disponibles**. Cliquez sur **Ajouter** pour incorporer l'entrée à la liste **Rappels utilisés**. Répétez cette étape afin d'ajouter tous les rappels dont vous avez besoin pour le serveur point de contact.
    b. Cliquez sur le bouton **Ajouter des paramètres**. Une section relative aux paramètres de rappel s'affiche pour chaque rappel figurant dans la liste Rappels utilisés. Des zones de paramètres comportant les valeurs par défaut `new key` et `new value` s'affichent.
    c. Ajoutez les paramètres de chaque rappel en remplaçant le nom et la valeur par défaut par les paramètres que vous souhaitez ajouter. Pour ajouter d'autres paramètres, cliquez sur **Créer**. Lorsque vous cliquez sur **Créer**, une nouvelle zone de paramètre contenant les valeurs par défaut est ajoutée à la liste des paramètres.
    d. Répétez les étapes précédentes jusqu'à ce que tous les paramètres aient été ajoutés à tous les rappels.
13. Cliquez sur **Suivant**. Le panneau Authentification s'affiche.
14. Dans ce panneau, spécifiez les rappels de déconnexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.
    a. Sélectionnez une entrée dans la liste **Rappels disponibles**. Cliquez sur **Ajouter** pour incorporer l'entrée à la liste **Rappels utilisés**. Répétez cette étape afin d'ajouter tous les rappels dont vous avez besoin pour le serveur point de contact.
    b. Cliquez sur le bouton **Ajouter des paramètres**. Une section relative aux paramètres de rappel s'affiche pour chaque rappel figurant dans la liste Rappels utilisés. Des zones de paramètres comportant les valeurs par défaut `new key` et `new value` s'affichent.
    c. Ajoutez les paramètres de chaque rappel en remplaçant le nom et la valeur par défaut par les paramètres que vous souhaitez ajouter. Pour ajouter d'autres paramètres, cliquez sur **Créer**. Lorsque vous cliquez sur **Créer**, une nouvelle zone de paramètre contenant les valeurs par défaut est ajoutée à la liste des paramètres.
    d. Répétez les étapes précédentes jusqu'à ce que tous les paramètres aient été ajoutés à tous les rappels.
15. Cliquez sur **Suivant**. Le panneau Récapitulatif s'affiche. Ce panneau affiche la liste de tous les rappels et des paramètres que vous avez spécifiez au cours des étapes précédentes.
16. Cliquez sur **Terminer** pour achever la configuration, ou sur **Précédent** pour revenir aux panneaux précédents et passer en revue vos sélections.

Que faire ensuite

Pour rendre ce serveur point de contact actif, poursuivez avec les instructions de la rubrique «Activation d'un serveur point de contact», à la page 818.

Création d'un serveur point de contact comme un serveur existant

Tivoli Federated Identity Manager fournit plusieurs options pour chaque serveur point de contact, suivant le rôle que vous tenez dans la fédération. De plus, vous avez la possibilité de développer votre propre serveur point de contact et de le baser sur un serveur existant. Si vous avez développé votre propre serveur, vous devez l'ajouter à votre environnement via la console.

Avant de commencer

Avant d'ajouter le serveur point de contact personnalisé à votre environnement, vous devez :

- Publier chaque point personnalisé des plug-ins de rappel de contact sur le noeud d'exécution. Voir «Publication des plug-ins de rappel», à la page 814.
- Connaître le type de paramètres à utiliser, le cas échéant, et les valeurs correspondantes à transmettre à ces rappels.

Pourquoi et quand exécuter cette tâche

La procédure qui suit explique comment ajouter un serveur point de contact personnalisé tel que les serveurs point de contact déjà définis dans votre environnement. Si vous ajoutez un serveur point de contact personnalisé différent d'un autre serveur, appliquez la procédure de la rubrique «Création d'un nouveau serveur point de contact», à la page 814.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gestion de la configuration** > **Point de contact**.
3. Sélectionnez le serveur point de contact existant à utiliser comme base du nouveau serveur point de contact.
4. Cliquez sur **Création à l'identique**. Le panneau de bienvenue de l'assistant de profil de point de contact s'ouvre.
5. Vérifiez que vous avez exécuté les étapes prérequises.
6. Cliquez sur **Suivant**. Le panneau Nom de profil et les informations du profil sélectionné s'affichent.
7. Indiquez le nom de profil de votre serveur point de contact personnalisé et, le cas échéant, une description.
8. Cliquez sur **Suivant**. Le panneau d'ouverture de session s'affiche.
9. Dans le panneau Ouverture de session, spécifiez les rappels de connexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

   Etant que vous avez sélectionné un profil pour ce serveur point de contact, les rappels et paramètres affichés pour ce profil seront identiques à ceux en cours d'utilisation.

   Pour ajouter ou supprimer des rappels, utilisez les boutons **Ajouter** et **Supprimer**. Les rappels indiqués dans la liste Rappels utilisés sont ceux qui seront repris par votre nouveau serveur point de contact.

10. Cliquez sur **Suivant**. Le panneau Fermeture de session s'affiche.
11. Dans le panneau Fermeture de session, spécifiez les rappels de connexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

    Etant que vous avez sélectionné un profil pour ce serveur point de contact, les rappels et paramètres affichés pour ce profil seront identiques à ceux en cours d'utilisation.

    Pour ajouter ou supprimer des rappels, utilisez les boutons **Ajouter** et **Supprimer**. Les rappels indiqués dans la liste Rappels utilisés sont ceux qui seront repris par votre nouveau serveur point de contact.
12. Cliquez sur **Suivant**. Le panneau ID local s'affiche.
13. Dans le panneau ID local, spécifiez les rappels de connexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

    Etant que vous avez sélectionné un profil pour ce serveur point de contact, les rappels et paramètres affichés pour ce profil seront identiques à ceux en cours d'utilisation.

    Pour ajouter ou supprimer des rappels, utilisez les boutons **Ajouter** et **Supprimer**. Les rappels indiqués dans la liste Rappels utilisés sont ceux qui seront repris par votre nouveau serveur point de contact.
14. Cliquez sur **Suivant**. Le panneau Authentification s'affiche.
15. Dans ce panneau, spécifiez les rappels de connexion à utiliser, l'ordre dans lequel ces rappels doivent être lancés, ainsi que les paramètres appliqués à chaque rappel.

    Etant que vous avez sélectionné un profil pour ce serveur point de contact, les rappels et paramètres affichés pour ce profil seront identiques à ceux en cours d'utilisation.

    Pour ajouter ou supprimer des rappels, utilisez les boutons **Ajouter** et **Supprimer**. Les rappels indiqués dans la liste Rappels utilisés sont ceux qui seront repris par votre nouveau serveur point de contact.
16. Cliquez sur **Suivant**. Le panneau Récapitulatif s'affiche. Ce panneau affiche la liste de tous les rappels et des paramètres que vous avez spécifiez au cours des étapes précédentes.
17. Cliquez sur **Terminer** pour achever la configuration, ou sur **Précédent** pour revenir aux panneaux précédents et passer en revue vos sélections.

Que faire ensuite

Pour rendre ce serveur point de contact actif, poursuivez avec les instructions de la rubrique «Activation d'un serveur point de contact».

Activation d'un serveur point de contact

Pour utiliser un serveur point de contact comme serveur actif dans votre environnement, vous devez l'activer.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Gérer la configuration** > **Point de contact**.
3. Sélectionnez le serveur point de contact que vous voulez activer.

4. Cliquez sur **Activer**. Le serveur point de contact que vous avez sélectionné est activé et utilisé comme serveur point de contact dans votre environnement.

Chapitre 52. Personnalisation des paramètres des certificats de signature X.509

Lors de la signature de messages ou d'assertions, le certificat X.509 (clé publique) est inclus dans votre signature sous forme de certificat X.509 codé en base 64. Vous pouvez toutefois préciser si ces données doivent être exclues et si d'autres données doivent être ajoutées à vos signatures.

Avant de commencer

Avant d'appliquer cette procédure, vous devez avoir procédé à la configuration de votre fédération. En outre, si vous êtes un fournisseur d'identité dans une fédération SAML 1.x, la configuration de vos paramètres de signature d'assertions a lieu au moment où vous ajoutez vos fournisseurs de services partenaires. Pour modifier les paramètres de votre signature d'assertion, il vous faut avoir déjà configuré un fournisseur de services partenaire.

Procédure

1. Connectez-vous à la console.
2. Cliquez sur **Tivoli Federated Identity Manager** > **Configurer la configuration unique fédérée** > **Fédérations**.

   A l'inverse, si vous êtes un fournisseur d'identité, pour modifier vos paramètres de signature d'assertions SAML 1.x, cliquez sur **Tivoli Federated Identity Manager** > **Configurer la connexion unique fédérée** > **Partenaires**. Le panneau Fédérations affiche la liste des fédérations configurées.
3. Sélectionnez une fédération. Le panneau Partenaires affiche la liste des partenaires configurés.
4. Sélectionnez un partenaire.
5. Cliquez sur **Propriétés**.
6. Sélectionnez les propriétés à modifier. Les propriétés sont décrites dans l'aide en ligne.
7. Une fois les propriétés modifiées, cliquez sur **OK** pour fermer le panneau Propriétés.

Chapitre 53. Exécution de WebSphere Application Server avec Java 2

Si vous exécutez la sécurité Java 2 sur l'instance de WebSphere Application Server sur laquelle Tivoli Federated Identity Manager est installé, vous devez modifier le fichier java.policy afin d'autoriser l'accès aux répertoires de Tivoli Federated Identity Manager situés dans le sous-répertoire temporaire de votre profil WebSphere.

Procédure

1. Localisez le fichier java.policy et ouvrez-le dans un éditeur de texte. Les emplacements par défaut du fichier sont les suivants :

   AIX

   ```
   /usr/IBM/WebSphere/AppServer/java/jre/lib/security/java.policy
   ```

   Linux ou Solaris

   ```
   /opt/IBM/WebSphere/AppServer/java/jre/lib/security/java.policy
   ```

   Windows

   ```
   C:\Program Files\IBM\WebSphere\AppServer
   ```

2. Ajoutez les lignes suivantes au fichier java.policy :

   ```
   grant codeBase "file:${server.root}/temp/nom_noeud/nom_serveur/
     ITFIMManagementService/-" {permission java.security.AllPermission;
   };
   grant codeBase "file:${server.root}/temp/nom_noeud/nom_serveur/
     ITFIMRuntime/-" {permission java.security.AllPermission;
   };
   ```

   nom_noeud désigne le nom du noeud, tel que IBM-FCFB36CC28ENode05

   nom_serveur désigne le nom du serveur, tel que server1

3. Sauvegardez et fermez le fichier `java.policy`.
4. Redémarrez WebSphere Application Server.

Partie 9. Annexes

Annexe A. Référence de tfimcfg

Utilisez la commande **tfimcfg** pour configurer WebSEAL ou Web Gateway Appliance en tant que serveur point de contact ou configurer des paramètres LDAP pour le service d'alias.

Syntaxe de tfimcfg

```
TFIM Autoconfiguration Tool Version 6.2.2.3 [XXXXXXa]

Syntaxe : java -jar tfimcfg.jar [-action <mode>] [options]
L'outil tfimcfg présente plusieurs modes d'opération.  Chaque mode utilise
des options de ligne de commande différentes.

Configuration et suppression de la configuration de serveurs WebSEAL :
   -action tamconfig : configure un serveur WebSEAL.  Il s'agit du mode par défaut.
   Options :
      -cfgfile <fichier>: fichier de configuration WebSEAL.
         Cette option est obligatoire.
      -rspfile <fichier>: fichier de réponses pour une configuration non interactive.
         Par défaut : configuration interactive
      -record : génération du fichier de réponses sans apporter de modifications
à la configuration WebSEAL.
      -sslfactory : spécifie la fabrique de connexions sécurisées à utiliser
(TLS ou SSL).
         Lorsque l'environnement TFIM est activé pour FIPS,
le seul type de fabrique pris en charge
est TLS.
         Si le paramètre n'est pas indiqué, la fabrique par défaut est SSL.

   -action tamunconfig : suppression de la configuration d'un serveur WebSEAL.
   Options :
      -cfgfile <fichier>: fichier de configuration WebSEAL.
         Cette option est obligatoire.
      -rspfile <fichier> : fichier de réponses pour la suppression non interactive
                           d'une configuration.
         Par défaut : configuration interactive

Configuration et suppression de la configuration de serveurs LDAP :
   -action ldapconfig : configure un serveur LDAP.
   Options :
      -rspfile <fichier> : fichier de réponses destiné à contrôler la configuration.
Le fichier de réponses doit être basé sur le fichier exemple
ldapconfig.properties.  Cette option est obligatoire.

   -action ldapunconfig : supprime la configuration d'un serveur LDAP.
   Options :
      -rspfile <fichier> : fichier de réponses destiné à contrôler la configuration.
Le fichier de réponses doit être basé sur le fichier exemple
ldapconfig.properties.  Cette option est obligatoire.

Configuration et déconfiguration des serveurs Web Gateway Appliance :
   -action wgaconfig : configure un serveur Web Gateway Appliance.
   Options :
      -cfgurl <url> : URL de configuration Web Gateway Appliance.
         Cette option est obligatoire.
      -rspfile <fichier>: fichier de réponses pour une configuration non interactive.
         Par défaut : configuration interactive
      -record : génération du fichier de réponses sans
apporter de modifications à la configuration Web Gateway Appliance.
      -sslfactory : spécifie la fabrique de connexions sécurisées à utiliser
(TLS ou SSL).
         Lorsque l'environnement TFIM est activé pour FIPS,
le seul type de fabrique pris en charge est TLS.
Si le paramètre n'est pas indiqué, la fabrique par défaut est SSL.
```

```
-action wgaunconfig : annule la configuration d'un serveur Web Gateway Appliance.
Options :
   -cfgfile <fichier> : URL de configuration Web Gateway Appliance.
      Cette option est obligatoire.
   -rspfile <fichier> : fichier de réponses pour la suppression non interactive
                        d'une configuration.
      Par défaut : configuration interactive
```

Les fichiers journaux de l'outil `tfimcfg.jar` sont enregistrés dans le répertoire temporaire du système. Le répertoire du fichier temporaire du système est indiqué par la propriété système `java.io.tmpdir`.

Configuration de WebSEAL ou de Web Gateway Appliance comme point de contact à l'aide de l'outil tfimcfg

Utilisez l'outil tfimcfg pour configurer WebSEAL ou Web Gateway Appliance comme point de contact.

Avant de commencer

Assurez-vous que votre serveur WebSEAL écoute les connexions sur les adresses et les numéros de port appropriés. Vous pouvez contrôler l'adresse IP et le numéro de port à l'aide du fichier de configuration WebSEAL ou de la console d'administration Web Gateway Appliance. L'adresse IP est contrôlée par l'option de configuration de l'interface réseau [serveur] et les numéros de port sont contrôlés par les options [serveur] `https-port` et [serveur] `http-port`.

Pour utiliser l'outil `tfimcfg`, vous devez remplir les conditions suivantes :

- Obtenir un environnement d'exécution Java IBM® pris en charge par la version installée de PDJrte.
- Pour WebSEAL uniquement : Utiliser PDJrte pour configurer l'environnement d'exécution Java IBM en mode complet.

Pour IBM Security Access Manager WebSEAL, version 7.0 ou ultérieure, vous devez également remplir les conditions suivantes :

- Obtenir la version 6.0, niveau de mise à jour 10 ou suivant, de l'environnement d'exécution Java IBM.
- Configurer *com.ibm.security.cmskeystore.CMSProvider* dans le fichier **java.security** se trouvant dans le répertoire `$JAVA_HOME/lib/security` de l'environnement d'exécution Java IBM.
- S'assurer que l'outil `ikeycmd` contenu dans `$JAVA_HOME/bin` figure dans le chemin.

Lorsque vous configurez WebSEAL en tant que point de contact Tivoli Federated Identity Manager, vous devez exécuter l'outil `tfimcfg.jar` sur l'ordinateur où l'instance WebSEAL est configurée.

Lorsque vous configurez une instance proxy inverse Web Gateway Appliance en tant que point de contact Tivoli Federated Identity Manager, vous pouvez exécuter l'outil `tfimcfg.jar` sur le même ordinateur que celui sur lequel Tivoli Federated Identity Manager est installé.

Pourquoi et quand exécuter cette tâche

L'outil `tfimcfg` utilise le nom d'hôte et le numéro de port de votre serveur WebSEAL pour déterminer quelles URL de la fédération doivent être configurées

sur votre serveur. L'outil `tfimcfg` tente de déterminer le nom d'hôte utilisé par votre serveur WebSEAL selon divers facteurs :

- Le paramètre de configuration WebSEAL du nom d'hôte Web [serveur]
- Le paramètre de configuration WebSEAL de l'interface réseau [serveur]
- Le nom d'hôte locale de votre système
- La résolution du nom d'hôte et de l'adresse IP

Remarque : Il est possible que l'outil `tfimcfg` ne parvienne pas à déterminer précisément le nom d'hôte utilisé par les clients pour contacter votre serveur WebSEAL, particulièrement dans des environnements réseau complexes. Dans tous les cas, l'outil `tfimcfg` vous invite à confirmer le nom d'hôte utilisé par le serveur WebSEAL. Entrez le nom de serveur WebSEAL correct. Pour plus d'informations, voir «Exécution de l'outil tfimcfg», à la page 830.

Selon les options de configuration que vous sélectionnez, l'outil exécute une partie ou toutes les étapes suivantes :

- Mettre à jour la base de données de clés WebSEAL avec le certificat SSL utilisé par le serveur Tivoli Federated Identity Manager.
- Faire une copie de sauvegarde de votre fichier de configuration WebSEAL.
- Enregistrer dans le fichier de configuration WebSEAL les informations nécessaires à l'annulation ultérieure de la configuration de cette fédération à partir du serveur WebSEAL.
- Lier des ACL pour accorder et restreindre l'accès aux URL de la fédération.
- Créer une jonction WebSEAL au serveur Tivoli Federated Identity Manager.
- Configurer un service EAS WebSEAL pour l'accès basé sur les risques ou le point d'application de règles OAuth.
- Configurer WebSEAL pour envoyer des paires HTTP-Tag-Value à la jonction.
- Activer l'authentification EAI pour chaque noeud final utilisé pour la connexion, la déconnexion ou SOAP.
- Activer l'authentification par analyse métier si vous l'avez demandée pour vos noeuds finaux SOAP.
- Activer l'authentification par certificat si vous l'avez demandée pour vos noeuds finaux SOAP.
- Désactiver l'authentification par analyse métier et activer l'authentification par formulaire si vous remplissez les conditions suivantes :
  - C'est la première fois que votre serveur WebSEAL est configuré pour Tivoli Federated Identity Manager.
  - Vous n'avez pas sélectionné l'authentification par analyse métier pour des noeuds finaux SOAP.
- Supprimer toutes les ACL qui sont utilisées ou ne le sont plus.
- Redémarrer votre serveur WebSEAL.
- Enregistrer un fichier de réponses afin de pouvoir répéter la configuration par la suite.
- Enregistrer un fichier journal de toutes les modifications de configuration apportées à votre serveur WebSEAL.

Tivoli Federated Identity Manager fournit un fichier `tfimcfg.jar` qui est utilisé pour modifier la configuration WebSEAL lorsque vous utilisez WebSEAL comme point de contact Tivoli Federated Identity Manager.

Procédure

1. Configurez une variable d'environnement JAVA_HOME pour l'environnement d'exécution Java : Par exemple :

   export JAVA_HOME=/opt/ibm/java-x86_64-60/jre, or

   export JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/jre
2. Ajoutez $JAVA_HOME/bin pour l'exportation de chemin PATH=$JAVA_HOME/bin:$PATH.
3. A partir de la ligne de commande, entrez :
   - Pour un serveur WebSEAL :

     java -jar tfimcfg.jar -action tamconfig -cfgfile *WebSEAL_filename* /opt/pdweb/etc/webseald-default.conf
   - Pour un serveur Web Gateway Appliance :

     java -jar tfimcfg.jar -action wgaconfig -cfgurl *Web_Gateway_Appliance_URL*

Résultats

Une fois que l'outil tfimcfg a exécuté les étapes de configuration de base de Tivoli Federated Identity Manager sur votre serveur WebSEAL, vous pouvez personnaliser la configuration WebSEAL.

Si vous utilisez plusieurs serveurs secondaires WebSEAL pour des raisons de haute disponibilité ou de performances, vous pouvez répéter la configuration à l'aide du fichier de réponses généré par l'outil tfimcfg. Pour répéter la configuration, copiez le fichier de réponses sur les autres serveurs WebSEAL, démarrez l'outil tfimcfg sur un autre dispositif Web Gateway Appliance ou sélectionnez une autre instance de proxy inverse Web Gateway Appliance et exécutez la configuration :

```
# java -jar tfimcfg.jar -rspfile <path-to-response-file> \
-cfgfile <path-to-webseal-config-file>
```

ou

```
# java -jar tfimcfg.jar -action wgaconfig -rspfile <path-to-response-file> \
-cfgurl Web_Gateway_Appliance_URL
```

Vous devez exécuter l'outil tfimcfg pour chaque fédération que vous configurez sur le serveur WebSEAL. Les modifications apportées aux URL de la fédération ou aux profils peuvent nécessiter une nouvelle exécution de l'outil tfimcfg. Ne réexécutez pas l'outil tfimcfg lorsque vous ajoutez des partenaires à une fédération. Lorsque vous configurez plusieurs fédérations sur un seul serveur WebSEAL, les options de configuration qui sont nécessaires pour une fédération peuvent ne pas être appropriées pour les autres fédérations. Par exemple, les fédérations Identity Provider et Service Provider ont des exigences différentes.

Exécution de l'outil tfimcfg

Utilisez l'outil tfimcfg pour configurer une instance WebSEAL.

Avant de commencer

L'environnement d'exécution Java doit être configuré avec l'exécution de Tivoli Access Manager 6.0 pour Java. Assurez-vous que l'environnement d'exécution Java figurant dans le chemin en cours est l'environnement approprié. Par exemple :

```
local:/ # which java
/opt/IBMJava2-142/jre/bin/java
```

Pourquoi et quand exécuter cette tâche

Il est supposé ce qui suit concernant l'environnement dans lequel s'exécute l'outil :

- L'exemple utilise un environnement IBM Tivoli Federated Identity Manager 6.1.0 mais le processus doit être similaire aux éditions suivantes du produit IBM Tivoli Federated Identity Manager.
- IBM HTTP Server est configuré avec le plug-in du serveur Web WebSphere pour réacheminer des demandes vers un serveur WebSphere sur lequel un fournisseur d'identité IBM Tivoli Federated Identity Manager est configuré. Le serveur IHS est configuré avec le nom d'hôte **ihs.myidp.com** et écoute sur le port 80.
- Le point de contact du fournisseur d'identité est un serveur WebSEAL avec le nom d'instance du fournisseur d'identité.
- Le point de contact du fournisseur d'identité n'est pas le même que celui du noeud final SOAP, c'est donc une instance WebSEAL différente qui gère le trafic d'URL du noeud final SOAP.
- Le produit IBM Tivoli Federated Identity Manager est installé dans le répertoire `/opt/IBM/FIMidp`.
- La machine virtuelle Java (IBM Java 2 1.4.2) est installée.
- Tivoli Access Manager est installé et configuré avec l'administrateur **sec_master** et le mot de passe **passw0rd**.

Procédure

1. Démarrez l'utilitaire à l'aide du fichier de configuration de l'instance du fournisseur d'identité WebSEAL et de la commande **–action tamconfig**.

   Vous devez indiquer le fichier de configuration pour que l'instance WebSEAL soit configurée.

   ```
   local:/opt/IBM/FIMidp/tools/tamcfg # java -jar
   ./tfimcfg.jar –action tamconfig –cfgfile
   /opt/pdweb/etc/webseald-idp.conf
   ```

2. Entrez l'ID utilisateur et le mot de passe de l'administrateur Tivoli Access Manager. Comme l'utilisateur administrateur Tivoli Access Manager est par défaut `sec_master` dans cet exemple, appuyez sur **Entrée** sans indiquer de valeur.

   ```
   TAM administrator user-id [sec_master]: <enter>
   TAM administrator password: passw0rd
   Creating TAM administration context...
   TAM administration context created successfully.
   Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
   ```

3. Entrez 1 pour poursuivre le traitement. L'outil vous invite à fournir la liste des URL correspondant aux URL du noeud final de service définies pour la fédération dans Tivoli Federated Identity Manager. Ces URL sont les points de contact fournis par l'instance de serveur WebSEAL. Etant donné que l'outil a correctement identifié le nom d'hôte WebSEAL à partir des données de configuration WebSEAL, appuyez sur **Entrée** sans indiquer de valeur.

   ```
   WebSEAL hostname [www.myidp.com]: <enter>
   WebSEAL URLs:
   http://www.myidp.com/
   https://www.myidp.com/
   ```

4. Entrez le nom d'hôte et le port WebSphere Application Server à l'emplacement où l'application d'exécution IBM Tivoli Federated Identity Manager est installée et en cours d'exécution. WebSphere Application Server doit être actif lorsque l'outil `tfimcfg` s'exécute. Dans cet exemple, SSL n'est pas utilisé pour communiquer avec WebSphere Application Server.

```
Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
ITFIM hostname []: ihs.myidp.com
ITFIM HTTP port: 80
Use SSL connection to ITFIM server (y/n): n
Testing connection to
http://ihs.myidp.com:80/Info/InfoServiceXML.
Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
```

5. Sélectionnez la fédération à configurer. Dans cet exemple, une seule fédération SAML 1.1, saml11Fed, est créée.

```
Federation to configure:
1. saml11Fed
2. Cancel
Enter your choice [1]: 1
```

6. Entrez 1 pour poursuivre la configuration.

```
<p>The following endpoints will be configured on this WebSEAL</p><p>server:
</p><p>https://www.myidp.com/FIM/sps/saml11Fed/saml11/login</
p><p>Press 1 for Next, 2 for Previous,
3 to Repeat, C to Cancel: <b>1</b></p>
```

7. Entrez 1 pour accepter la valeur. L'outil affiche l'URL choisie pour l'accès de tous les utilisateurs authentifiés car elle est reconnue comme noeud final du service de transfert intersite. Ce noeud final correspond à l'URL qui génère un jeton et transfère l'identité de l'utilisateur vers un autre site.

```
<p>URLs allowing all authenticated users access:</p><p>
https://www.myidp.com/FIM/sps/saml11Fed/saml11/login</
p><p>Press 1 for Next, 2 for Previous,
3 to Repeat, C to Cancel: <b>1</b></p>
```

8. Entrez 1 pour poursuivre jusqu'à cette page récapitulative :

```
Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
-------------------------------------------
Planned configuration steps:
A backup of the WebSEAL configuration will be saved as
/opt/pdweb/etc/webseald-idp.conf.2006-03-02-17-08-49.
A junction to the FIM server will be created at /FIM.
ACLs denying access to all users will be attached to:
/WebSEAL/www.myidp.com-idp/FIM
ACLs allowing access to all authenticated users will be
attached to:
/WebSEAL/www.myidp.com-idp/FIM/sps/saml11Fed/saml11/login
/WebSEAL/www.myidp.com-idp/FIM/fimivt
HTTP-Tag-Value header insertion will be configured for the
attributes:
ssn=ssn
name=name
email=email
user_session_id=user_session_id
```

9. Entrez 1 pour poursuivre les modifications de configuration.

```
Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
Beginning configuration...
Configuration backup:/opt/pdweb/etc/webseald-idp.conf.2006-03-
02-17-15-54
Attaching ACLs.
Creating ACL itfim_saml11Fed_nobody.
Creating ACL itfim_saml11Fed_anyauth.
Creating junction /FIM.
Created junction at /FIM
Editing configuration file...
Restarting the WebSEAL server...
Configuration complete.
```

Résultats

L'outil crée un fichier de réponses pour l'application répétée pour cette étape de configuration sur d'autres serveurs WebSEAL. Enfin, il suggère d'éventuelles autres étapes.

Configuration du trafic SOAP à l'aide de l'outil tfimcfg

Utilisez l'outil tfimcfg pour configurer le serveur WebSEAL qui gère le trafic SOAP.

Pourquoi et quand exécuter cette tâche

Dans cet exemple, le fournisseur d'identité utilise la sécurité de transport pour restreindre l'accès à son noeud final SOAP, qui est une instance WebSEAL dédiée configurée comme `idpsoap`. Cette méthode permet l'authentification obligatoire de certificats côté client pour cette instance WebSEAL sans affecter le reste de l'environnement Tivoli Federated Identity Manager.

Le noeud final SOAP du fournisseur d'identité ou le service de résolution d'artefact exige que le client s'authentifie à l'aide d'un certificat indiquant que l'utilisateur est un membre du groupe Tivoli Access Manager `soapusers`.

Procédure

1. Démarrez l'utilitaire de configuration Tivoli Federated Identity Manager pour Tivoli Access Manager.

   ```
   /opt/IBM/FIMidp/tools/tamcfg # java -jar ./tfimcfg.jar -action
   tamconfig -cfgfile /opt/pdweb/etc/webseald-idpsoap.conf
   ```

2. Entrez les données d'administrateur Tivoli Access Manager et le nom d'hôte WebSEAL.

   ```
   TAM administrator user-id [sec_master]: <enter>
   TAM administrator password: passw0rd
   Creating TAM administration context...
   TAM administration context created successfully.
   Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
   WebSEAL hostname [soap.myidp.com]: <enter>
   WebSEAL URLs:
   https://soap.myidp.com/
   ```

3. Entrez le nom d'hôte IBM Tivoli Federated Identity Manager.

   Le nom d'hôte correspond au WebSphere Application Server sur lequel Tivoli Federated Identity Manager s'exécute. Indiquez la fédération en cours de configuration.

   ```
   Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
   ITFIM hostname []: ihs.myidp.com
   ITFIM HTTP port: 80
   Use SSL connection to ITFIM server (y/n): n
   Testing connection to
   http://ihs.myidp.com:80/Info/InfoServiceXML.
   Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
   Federation to configure:
   1. saml11Fed
   2. Cancel
   Enter your choice [1]: 1
   The following endpoints will be configured on this WebSEAL
   server:
   https://soap.myidp.com/FIM/sps/saml11Fed/saml11/soap
   ```

Par exemple, **saml11Fed**. L'outil indique ensuite la liste des adresses URL qui sont identifiées comme associées à la fédération. Cette liste est la liste des URL de noeud final de service reconnues comme associées à cette instance WebSEAL.

4. Entrez 1 pour sélectionner **Certificate authentication**.
5. Entrez **soapusers**. Le noeud final correspondant étant le service de résolution d'artefact, l'outil demande le type d'authentification devant être configuré pour les clients SOAP authentifiés. Ce fournisseur d'identité est configuré avec une règle que seuls les clients avec certificats valides et dont les utilisateurs sont membres du groupe Tivoli Access Manager `soapusers` sont autorisés à accéder au noeud final SOAP Tivoli Federated Identity Manager.

   ```
   Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
   Access type for endpoint URL
   https://soap.myidp.com/FIM/sps/saml11Fed/saml11/soap:
   1. Certificate authentication
   2. User-id/password authentication
   3. Unauthenticated access
   Enter your choice [1]: 1
   Group for SOAP access: soapusers
   URLs used for authenticated SOAP clients:
   https://soap.myidp.com/FIM/sps/saml11Fed/saml11/soap
   Authentication type: certificate
   Access group: soapusers
   ```

6. Entrez `n` dans la demande de configuration IVT.

   La configuration IVT étant déjà effectuée pour le fournisseur d'identité lors de la configuration de `www.myidp.com`, ou du fournisseur d'identité de l'instance WebSEAL, elle n'est pas requise à nouveau.

   Le récapitulatif présente les détails suivants.

   - Une copie de sauvegarde du fichier de configuration WebSEAL est créée.
   - Une jonction appelée **/FIM** est créée entre WebSEAL et l'hôte ou le port contactant l'exécution Tivoli Federated Identity Manager. Le nom **/FIM** est choisi à partir de l'URL du noeud final de service approprié.
   - Une ACL **deny all** est liée à **/FIM**. Cette règle de sécurité permet de s'assurer que tous les accès qui ne sont pas explicitement autorisés ne sont pas autorisés.
   - Une ACL **allow group soapusers only** est liée au noeud final de service de résolution d'artefact.

     ```
     Press 1 for Next, 2 for Previous, 3 to Repeat, C to Cancel: 1
     ---------------------------------------------
     Planned configuration steps:
     A backup of the WebSEAL configuration will be saved as
     /opt/pdweb/etc/webseald-idpsoap.conf.2006-03-02-19-52-54.
     A junction to the FIM server will be created at /FIM.
     ACLs denying access to all users will be attached to:
     /WebSEAL/soap.myidp.com-idpsoap/FIM
     ACLs allowing access to the group soapusers will be attached to:
     /WebSEAL/soap.myidp.com-idpsoap/FIM/sps/saml11Fed/saml11/soap
     Certificate authentication will be enabled.
     HTTP-Tag-Value header insertion will be configured for the
     attributes:
     user_session_id=user_session_id
     ```

7. Entrez **1** pour poursuivre la modification de la configuration et pour afficher les informations suivantes :

   ```
   Configuration backup:/opt/pdweb/etc/websealdidpsoap.
   conf.2006-03-02-19-52-54
   Attaching ACLs.
   Creating ACL itfim_saml11Fed_soapusers.
   Creating junction /FIM.
   ```

```
Created junction at /FIM
Editing configuration file...
Restarting the WebSEAL server...
Configuration complete.
```

Configuration d'un groupe et d'un certificat soapusers

Configurez un certificat côté client qui identifie l'utilisateur en tant que membre d'un groupe Security Access Manager. Configurez le client pour que ce groupe dispose d'un accès selon la règle Tivoli Access Manager.

Pourquoi et quand exécuter cette tâche

Cette tâche fournit des instructions pour la configuration de l'utilisateur et du groupe Tivoli Access Manager ainsi que du certificat pour l'authentification côté client. Grâce au protocole SOAP sur des liaisons HTTP, le fournisseur d'identité peut authentifier le client soap à l'aide de l'une des options suivantes :

- Authentification de base
- Certificat côté client

Procédure

1. Créez une entrée pour le suffixe de l'utilisateur et du groupe dans LDAP.
   - Le suffixe `o=ibm, c=us` est utilisé, vous devez donc créer un fichier LDIF approprié.
   - Le fichier LDIF et la commande **idsldapupdate** pour créer le suffixe.
   - L'administrateur LDAP est `cn=root` avec le mot de passe `passw0rd`.
   - Le serveur LDAP est Tivoli Directory Server.

   ```
   # cat ibmorg.ldif
   dn: o=ibm,c=us
   changetype: add
   objectclass: organization
   o: ibm
   # idsldapmodify -D cn=root -w passw0rd –f /studentfiles/files/ibmOrg.ldif
   ```
2. Créez un utilisateur Tivoli Access Manager `spsoapuser` et un groupe `soapusers`.

   ```
   # pdadmin -a sec_master -p passw0rd <<SOAPUSER
   user create spsoapuser cn=spsoapuser,o=ibm,c=us spsoapuser mssoap passw0rd
   user modify spsoapuser account-valid yes
   group create soapusers cn=soapusers,o=ibm,c=us soapusers
   group modify soapusers add spsoapuser
   SOAPUSER
   ```
3. Créez ou récupérez un certificat qui sera utilisé par un fournisseur de services pour l'authentification de client sur le noeud final SOAP. La clé publique du fournisseur de services doit être importée dans le fichier de clés approprié à l'aide du service de clés IBM Tivoli Federated Identity Manager.

   L'outil iKeyman livré avec Tivoli Access Manager et WebSphere peut être utilisé pour créer un certificat autosigné.

   Le certificat doit indiquer que le nom distinctif de l'objet est l'utilisateur Tivoli Access Manager créé ou `cn=spsoapuser, o=ibm, c=us` dans l'exemple.
4. Configurez la nouvelle instance WebSEAL avec un fichier `.kdb` qui valide un certificat client utilisé par le fournisseur de services.

Restrictions de tfimcfg sous Sun Java 1.4.2.4

Certaines versions de Sun Java sont incompatibles avec tfimcfg.

L'incompatibilité provoque l'erreur suivante :

```
HPDAZ0602E Corrupted file: Insufficient information to contact Policy Server
```

L'incident se produit parce que l'interpréteur JRE de Sun ne peut pas lire les fichiers de clés générés par la commande PDJrteCfg de Tivoli Access Manager. Lorsque cette erreur se produit, vous devez soit utiliser une machine JVM IBM JVM, soit appliquer les derniers correctifs JRE de Sun. Si l'incident persiste après l'application du module de correction de Sun, utilisez une machine virtuelle Java IBM pour la configuration.

Référence des propriétés LDAP tfimcfg

L'utilitaire tfimcfg consulte un fichier de propriétés pour obtenir les valeurs applicables lors de la configuration d'un registre d'utilisateurs LDAP. Le fichier de propriétés contient des valeurs que vous pouvez modifier.

ldap.hostname
Nom d'hôte du serveur LDAP. Valeur par défaut : `localhost`

ldap.port
Numéro du port LDAP. Valeur par défaut : `389`

La valeur par défaut est pour la communication non SSL. Lorsque le serveur LDAP est configuré pour communiquer via SSL, le port par défaut est 636.

ldap.suffix.add
Valeur booléenne qui définit si tfimcfg ajoute des suffixes au serveur LDAP, le cas échéant. Prend uniquement en charge IBM Tivoli Directory Server versions 6.1, 6.0 et 5.2.

Valeur par défaut :

```
ldap.suffix.add=true
```

ldap.suffix.user.configuration
ldap.organization.configuration
Valeurs booléennes qui définissent si tfimcfg crée des conteneurs LDAP pour stocker les utilisateurs et groupes Tivoli Federated Identity Manager. Les utilisateurs et groupes Tivoli Federated Identity Manager sont les suivants :

- Utilisateurs et groupes du serveur Tivoli Federated Identity Manager
- Utilisateurs et groupes d'IVT (Installation Verification Tool) Tivoli Federated Identity Manager

Si vous n'avez pas besoin de ces utilisateurs et groupes, ou si vous avez déjà des conteneurs LDAP à utiliser pour ces utilisateurs et groupes, définissez ces valeurs à `false`.

Lorsque ldap.organization.configuration a la valeur true, tfimcfg crée les objets LDAP `dc=exemple,dc=com`.

Valeur par défaut :

```
ldap.suffix.user.configuration=true
ldap.organization.configuration=true
```

ldap.suffix.alias.configuration
Valeur booléenne qui définit si tfimcfg crée un suffixe LDAP pour stocker les alias de connexion unique. L'alias par défaut est `cn=itfim`.

ldap.suffix.alias.configuration=true

ldap.suffix.tam.configuration
Valeur booléenne qui définit si tfimcfg crée le suffixe secAuthority=Default pour Tivoli Access Manager.

- Si Tivoli Access Manager est déjà configuré, entrez false pour cette valeur.
- Si Tivoli Access Manager n'utilise pas ce serveur LDAP, entrez false pour cette valeur.

ldap.suffix.tam.configuration=true

Remarque : Si le suffixe secAuthority=Default existe, le programme tfimcfg ne tient pas compte de la valeur de la propriété ldap.suffix.tam.configuration.

ldap.fim.configuration
Valeur booléenne qui définit si tfimcfg configure LDAP pour le service d'alias Tivoli Federated Identity Manager.

Valeur par défaut : true.

ldap.ivt.sp.configuration
Valeur booléenne qui définit si tfimcfg crée des utilisateurs et des groupes pour le fournisseur de services dans l'application IVT (Installation Verification Tool).

Valeur par défaut : true.

ldap.ivt.ip.configuration
Valeur booléenne qui définit si tfimcfg crée des utilisateurs et des groupes pour le fournisseur d'identité dans l'application IVT (Installation Verification Tool).

Valeur par défaut : true.

ldap.modify.acls
Valeur booléenne qui définit si tfimcfg connecte les listes de contrôle d'accès appropriées au serveur LDAP. Ces listes octroient l'accès en écriture et en lecture aux administrateurs Tivoli Federated Identity Manager créés par tfimcfg.

Remarque : L'outil tfimcfg connecte des listes de contrôle d'accès pour les serveurs IBM LDAP et Sun ONE. Pour les autres serveurs LDAP, vous devez connecter ces listes manuellement.

Lorsque la valeur est définie sur false, vous devez connecter les ACL manuellement.

Valeur par défaut : true.

ldap.admin.dn
Nom distinctif utilisé par l'administrateur de LDAP pour exécuter les demandes de liaison.

Valeur par défaut : cn=root

ldap.admin.password
Mot de passe de l'administrateur LDAP.

Valeur par défaut : passw0rd

ldap.security.enabled
Valeur booléenne qui définit si la communication avec le serveur LDAP doit utiliser SSL.

Valeur par défaut : false.

ldap.security.trusted.jks.filename
Nom du fichier de clés Java contenant le signataire du certificat SSL présenté par LDAP lors des communications sécurisées.

ldap.suffix.user.dn
ldap.suffix.user.name
ldap.suffix.user.attributes
ldap.suffix.user.objectclasses
Si vous voulez que tfimcfg.jar crée des conteneurs LDAP pour vos utilisateurs, vous pouvez définir ces valeurs pour commander les noms distinctifs utilisés.

Valeurs par défaut :

```
ldap.suffix.user.dn=dc=com
ldap.suffix.user.name=com
ldap.suffix.user.attributes=dc
ldap.suffix.user.objectclasses=domain
```

ldap.suffix.alias.dn
Nom distinctif à utiliser pour stocker les alias de connexion unique. La valeur de cette propriété doit commencer par `cn=`. Modifiez cette valeur lorsque vous ne voulez pas utiliser le nom distinctif par défaut.

Valeur par défaut :

```
ldap.suffix.alias.dn=cn=itfim
```

ldap.organization.dn
ldap.organization.name
ldap.organization.attributes
ldap.organization.objectclasses
Si vous voulez que tfimcfg.jar crée des conteneurs LDAP pour vos groupes, vous pouvez définir ces valeurs pour commander les noms distinctifs utilisés.

Valeurs par défaut :

```
ldap.organization.dn=dc=exemple,dc=com
ldap.organization.name=exemple
ldap.organization.attributes=dc
ldap.organization.objectclasses=domain
```

ldap.user.container.dn
ldap.group.container.dn

Noms distinctifs à utiliser pour les conteneurs des utilisateurs et des groupes.

Valeurs par défaut :

```
ldap.user.container.dn=cn=users,dc=exemple,dc=com
ldap.group.container.dn=cn=groups,dc=exemple,dc=com
```

ldap.fim.server.bind.dn
ldap.fim.server.bind.shortname
ldap.fim.server.bind.password

Nom distinctif, nom abrégé et mot de passe que l'application serveur Tivoli Federated Identity Manager utilise pour se connecter au serveur LDAP.

Par défaut :

```
ldap.fim.server.bind.dn=uid=fimserver,cn=users,dc=exemple,dc=com
ldap.fim.server.bind.shortname=fimserver
ldap.fim.server.bind.password=passw0rd
```

ldap.fim.admin.group.dn
ldap.fim.admin.group.shortname

Nom distinctif et nom abrégé du groupe d'administration Integrated Solutions Console.

Valeur par défaut :

```
ldap.fim.admin.group.dn=cn=fimadmins,cn=groups,dc=exemple,dc=com
ldap.fim.admin.group.shortname=fimadmins
```

ldap.user.objectclasses
ldap.group.objectclasses
ldap.user.shortname.attributes

Valeurs des conteneurs LDAP pour les attributs user.objectclasses, group.objectclasses et user.shortname.

Valeur par défaut :

```
ldap.user.objectclasses=person,organizationalPerson,inetOrgPerson
ldap.group.objectclasses=groupOfUniqueNames
ldap.user.shortname.attributes=cn,sn,uid
```

Fichier ldapconfig.properties par défaut

Le fichier ldapconfig.properties est fourni avec le composant d'exécution et de gestion de services. Certaines propriétés possèdent des valeurs par défaut.

```
ldap.hostname=localhost
ldap.port=389

# If true, new suffixes will be added to the LDAP server as needed.
# Only supported for IDS 5.2 and 6.0
ldap.suffix.add=true

# If true, data for the LDAP user suffix (dc=com, by default) will be
# created.
ldap.suffix.user.configuration=true

# If true, data for the SSO alias suffix (cn=itfim, by default) will be
# created.
ldap.suffix.alias.configuration=true

# If true, create the secAuthority=Default suffix for TAM
ldap.suffix.tam.configuration=true
ldap.fim.configuration=true
ldap.ivt.sp.configuration=true
ldap.ivt.ip.configuration=true
ldap.organization.configuration=true
ldap.modify.acls=true

ldap.admin.dn=cn=root
ldap.admin.password=passw0rd

ldap.security.enabled=false
ldap.security.trusted.jks.filename=

ldap.suffix.user.dn=dc=com
ldap.suffix.user.name=com
ldap.suffix.user.attributes=dc
ldap.suffix.user.objectclasses=domain

# DN to use for storing SSO aliases.  This must begin with cn=
ldap.suffix.alias.dn=cn=itfim

ldap.organization.dn=dc=exemple,dc=com
ldap.organization.name=exemple
ldap.organization.attributes=dc
ldap.organization.objectclasses=domain

ldap.user.container.dn=cn=users,dc=exemple,dc=com
ldap.group.container.dn=cn=groups,dc=exemple,dc=com

ldap.fim.server.bind.dn=uid=fimserver,cn=users,dc=exemple,dc=com
ldap.fim.server.bind.shortname=fimserver
ldap.fim.server.bind.password=passw0rd

ldap.fim.admin.group.dn=cn=fimadmins,cn=groups,dc=exemple,dc=com
ldap.fim.admin.group.shortname=fimadmins

ldap.user.objectclasses=person,organizationalPerson,inetOrgPerson
ldap.group.objectclasses=groupOfUniqueNames
ldap.user.shortname.attributes=cn,sn,uid
```

Figure 79. Valeurs par défaut pour le fichier ldapconfig.properties

Exemple de sortie de la configuration LDAP via tfimcfg

La section suivante illustre un exemple de résultat de l'exécution de tfimcfg.

La commande d'exécution de tfimcfg permettant de configurer les entrées LDAP pour le service d'alias est la suivante :

```
java -jar tfimcfg.jar -action ldapconfig -rspfile /tmp/ldapconfig.properties
```

La figure suivante illustre un résultat de l'exécution de la commande sur un fournisseur d'identité. Cet exemple utilise un fichier `ldapconfig.properties` qui comporte des valeurs par défaut.

```
Configuration du serveur LDAP.
Fournisseur du serveur LDAP : International Business Machines (IBM),
   version 6.0.
Ajout du suffixe LDAP secAuthority=Default.
Rechargement de la configuration IBM Directory Server.
Ajout du suffixe LDAP dc=com.
Rechargement de la configuration IBM Directory Server.
Création de l'objet LDAP dc=com.
Ajout du suffixe LDAP cn=itfim-cmd.
Rechargement de la configuration IBM Directory Server.
Création de l'objet LDAP cn=itfim-cmd.
Création de l'objet LDAP dc=example,dc=com.
Création de l'objet LDAP cn=users,dc=example,dc=com.
Création de l'objet LDAP cn=groups,dc=example,dc=com.
Création de l'objet LDAP uid=fimserver,cn=users,dc=example,dc=com.
Création de l'objet LDAP cn=fimadmins,cn=groups,dc=example,dc=com.
Ajout de l'utilisateur uid=fimserver,cn=users,dc=example,dc=com to group
   cn=fimadmins,cn=groups,dc=example,dc=com.
Création de l'objet LDAP o=identityprovider,dc=com.
Création de l'objet LDAP cn=MEemployee,o=identityprovider,dc=com.
Création de l'objet LDAP cn=MEmanager,o=identityprovider,dc=com.
Création de l'objet LDAP cn=MEexecutive,o=identityprovider,dc=com.
Création de l'objet LDAP cn=elain,o=identityprovider,dc=com.
Création de l'objet LDAP cn=mary,o=identityprovider,dc=com.
Création de l'objet LDAP cn=chris,o=identityprovider,dc=com.
Mise à jour des listes de contrôle d'accès IBM LDAP pour le suffixe CN=ITFIM-CMD.
Mise à jour des listes de contrôle d'accès IBM LDAP pour le suffixe
   SECAUTHORITY=DEFAULT.
Mise à jour des listes de contrôle d'accès IBM LDAP pour le suffixe DC=COM.
Mise à jour de la configuration du serveur LDAP terminée.
```

Figure 80. Exemple de sortie de tfimcfg.jar

Annexe B. Adresses URL pour l'initialisation d'actions de connexion unique

Les spécifications SAML ne contiennent pas ou peu d'informations sur les noeuds finals ou les méthodes que les utilisateurs doivent employer pour initier des actions de connexion unique. Toutefois, dans un environnement Tivoli Federated Identity Manager, des adresses URL sont définies afin de permettre à l'utilisateur d'initier des actions de connexion unique.

La référence est utile pour les architectes ou développeurs d'applications qui implémentent les composants d'interaction utilisateur de leur fédération.

Remarque : Ces adresses URL ne sont pas employées pour les communications inter-partenaires. Pour plus d'informations, voir Chapitre 16, «Noeuds finals SAML et adresses URL», à la page 189.

Adresse URL initiale SAML 1.x

L'adresse URL du service de transfert inter-sites constitue le point de départ du processus de requête de connexion unique dans une fédération SAML 1.x. L'URL d'émission d'une requête de connexion unique se caractérise par la syntaxe suivante :

Syntaxe

```
https://nomhôte_fournisseur_identité:numéro_port/sps/
  nom_fédération/samlxx/login?TARGET=
  id_fournisseur_services/emplacement_application_cible
  [chaînes de requête optionnelles]
```

Eléments

https *ou* **http**
: Schéma d'URI. `https` pour les ressources qui sont protégées par SSL (Secure Socket Layer). `http` pour les ressources qui ne sont pas protégées par SSL.

nomhôte_fournisseur_identité
: Nom d'hôte du serveur point de contact du fournisseur d'identité.

numéro_port
: Numéro de port du noeud final de service de transfert inter-sites. Le valeur par défaut est 9443.

sps
: Désignation du serveur Tivoli Federated Identity Manager. Cet élément ne peut pas être modifié.

nom_fédération
: Nom affecté à la fédération lorsque vous la créez.

saml*xx*
: Désignation du protocole SAML que vous choisissez d'utiliser dans votre fédération. Les valeurs peuvent correspondre à une des suivantes :
  - saml (pour SAML 1.0)
  - saml11 (pour SAML 1.1)

login Cet élément indique le type de noeud final qui utilise le port. **login** est utilisé pour le service de transfert inter-sites.

Utilisez la chaîne de requête **TARGET**. Vous avez la possibilité d'utiliser l'une ou l'autre, les deux ou aucune des deux chaînes de requête facultatives (**SP_PROVIDER**) et (**PROTOCOL**) ; voir les exemples suivants :

TARGET
URL de l'application cible à laquelle un utilisateur peut se connecter via une connexion unique.

SP_PROVIDER_ID
La valeur de cette chaîne de requête indique l'ID du fournisseur de services correspondant à la cible de la demande de connexion unique. Cette chaîne de requête est facultative, mais peut être nécessaire. L'utilisation de cette chaîne de requête évite toute ambiguïté quant au fournisseur de services qui est la cible de la demande de connexion unique.

Sans cette chaîne de requête, le fournisseur de services est déterminé via le mappage de l'*URI://nomhôte[:port]* de l'URL dans la chaîne de requête TARGET vers l'*URI://nomhôte[:port]* de l'ID du fournisseur de services partenaire qui est configuré pour la fédération. Ce paramètre est utilisé pour les requêtes initiées par le fournisseur d'identité.

PROTOCOL
La valeur de ce paramètre indique le type de profil de connexion unique (artefact du navigateur ou POST du navigateur) qui peut être utilisé pour la demande de connexion unique. La syntaxe de l'extension est `PROTOCOL=[BA|POST]`, où BA correspond à 'Browser Artifact' et POST à 'Browser POST'. La chaîne de requête remplace la configuration locale du fournisseur d'identité.

L'utilisation de l'extension est facultative. En l'absence d'extension, le choix du profil est déterminé par les paramètres du fichier de configuration. Pour utiliser cette extension, vous devez activer le paramètre d'extension IBM PROTOCOL lors de la procédure de configuration pour créer une fédération SAML 1.x sur un fournisseur d'identité.

Ces chaînes de requête peuvent être utilisées individuellement ou en association. Par exemple, l'URL servant à déclencher la connexion unique lorsque le paramètre SP_PROVIDER_ID est utilisé alors que l'extension PROTOCOL ne l'est pas, comporte la syntaxe suivante :

```
https://URL_service_transfert_inter-sites?SP_PROVIDER_ID=
  ID_fournisseur_services&TARGET=URL_application_cible
```

Avec le paramètre SP_PROVIDER_ID et l'extension PROTOCOL, l'URL comporte la syntaxe suivante :

```
https://URL_service_transfert_inter-sites?SP_PROVIDER_ID=
  ID_fournisseur_services&TARGET=URL_application_cible
  &PROTOCOL=[BA|POST]
```

Exemples

URL de connexion unique, sans les paramètres facultatifs :
L'exemple suivant présente l'URL de connexion unique d'un fournisseur d'identité utilisant une fédération nommée `ipfed`, le protocole SAML 1.1, un fournisseur de services dont l'ID est `https://sp.example.com:9443`, ainsi qu'une application appelée `snoop` :

```
https://idp.example.com:9443/sps/ipfed/saml11/login?TARGET=
  https://sp.example.com:9443/snoop/
```

URL de connexion unique, lorsque le paramètre SP_PROVIDER_ID et l'extension PROTOCOL *sont* utilisés :

L'exemple ci-dessous présente une URL qui sert à déclencher la connexion unique lorsque l'extension IBM PROTOCOL *est* utilisée. Dans cet exemple, même si le fournisseur d'identité est configuré pour utiliser un profil POST pour le fournisseur de services nommé `sp`, l'utilisation suivante de l'extension PROTOCOL force le fournisseur d'identité à utiliser le profil Artefact du navigateur :

```
https://idp.example.com:9443/sps/ipfed/saml11/login?SP_PROVIDER_ID=
  https://sp.example.com:9443/sps/spfed/saml11&TARGET=
  https://sp.example.com:9443/
  snoop&PROTOCOL=BA
```

URL de connexion unique, lorsque le paramètre SP_PROVIDER_ID est utilisé alors que l'extension PROTOCOL ne l'est *PAS* :

L'exemple suivant présente une URL qui sert à déclencher la connexion unique lorsque le paramètre SP_PROVIDER_ID est utilisé alors que l'extension IBM PROTOCOL ne l'est *PAS* :

```
https://idp.example.com:9443/sps/ipfed/saml11/login?SP_PROVIDER_ID=
  https://sp.example.com:9443/sps/spfed/saml11&TARGET=
  https://sp.example.com:9443/snoop
```

Adresses URL initiales de profil SAML 2.0

Dans l'environnement Tivoli Federated Identity Manager, des adresses URL d'un format spécial peuvent être utilisées pour les actions de connexion unique lancées par l'utilisateur. Ces adresses URL intègrent l'action de connexion unique à exécuter, la liaison à utiliser pour cette action et l'emplacement d'exécution de l'action. Elles sont appelées *URL initiales de profil*.

La spécification SAML 2.0 définit les noeuds finals à utiliser pour les communications partenaire à partenaire. Toutefois, la spécification ne définit pas la manière dont les utilisateurs peuvent initier une connexion unique avec ces noeuds finals.

Les architectes et développeurs d'applications, qui conçoivent et implémentent l'interaction des utilisateurs avec le processus de connexion unique, doivent comprendre la fonction des adresses URL initiales de profil et les incorporer dans leurs applications Web.

Les sections suivantes décrivent le format des URL initiales de profil SAML 2.0 qui sont prises en charge dans un environnement Tivoli Federated Identity Manager.

Adresse URL initiale du service d'assertion client (fournisseur de services)

Dans une fédération SAML 2.0, les URL du service d'assertion client peuvent être initiées au niveau du site du fournisseur d'identité, ou celui du fournisseur de services. La présente rubrique décrit la syntaxe d'initiation de la connexion unique au niveau du fournisseur de services.

Syntaxe d'initialisation de connexion unique sur le fournisseur de services

```
https://nom_hôte_fournisseur:numéro_port/sps/
nom_fédération/saml20/logininitial?RequestBinding=RequestBindingType&
ResponseBinding=ResponseBindingType&
NameIdFormat=NameIDFormatType&
IsPassive=[true|false]&
ForceAuthn=[true|false]&
AllowCreate=[true|false]&
AuthnContextClassRef = ClassReference&
AuthnContextDeclRef = DeclarationReference&
AuthnContextComparison = [exact| minimum | maximum |better]&
Target=target_application_location
```

Eléments

https *ou* **http**
Schéma d'URI. `https` pour les ressources qui sont protégées par SSL (Secure Socket Layer). `http` pour les ressources qui ne sont pas protégées par SSL.

nom_hôte_fournisseur
Nom d'hôte du serveur point de contact du fournisseur.

numéro_port
Numéro de port du noeud final de service de transfert inter-sites. Le valeur par défaut est 9443.

sps Désignation du serveur Tivoli Federated Identity Manager. Cet élément ne peut pas être modifié.

nom_fédération
Nom affecté à la fédération lorsque vous la créez.

saml20
Désignation de la fédération SAML 2.0.

logininitial
Cet élément indique le type de noeud final qui utilise le port. Le paramètre **logininital** initialise le service de connexion unique.

Les chaînes de requête suivantes doivent également être utilisées dans l'adresse URL :

RequestBinding
Liaison utiliser pour envoyer la requête. Les valeurs admises pour l'initialisation de connexion unique sur le fournisseur de services sont les suivantes :

- HTTPPost
- HTTPArtifact
- HTTPRedirect

ResponseBinding
Liaison utilisée par l'émetteur de la réponse renvoyée. Les valeurs admises pour l'initialisation de connexion unique sur le fournisseur de services sont les suivantes :

- HTTPPost
- HTTPArtifact

Target Adresse URL de l'application à laquelle un utilisateur peut se connecter à l'aide de la connexion unique.

NameIdFormat

Format utilisé pour les identificateurs de nom. Les valeurs admises sont :

- Transient (anonyme)
- Persistent
- Encrypted (pour les identificateurs de noms chiffrés)
- Email

Le paramètre 'Persistent' est défini par défaut. Si l'attribut 'NameIdFormat' n'est pas inclus, un ID de nom persistant est spécifié.

AllowCreate

Indique si une nouvelle liaison de compte persistante est appliquée à la requête. La valeur par défaut est `true`.

Remarque : Pour permettre l'utilisation de ce paramètre, **NameIdFormat** doit être défini sur "Persistent".

ForceAuthn

Indique si le fournisseur d'identité authentifie l'utilisateur. Une valeur de `true` signifie que l'utilisateur doit être authentifié. La valeur par défaut est `false`.

Remarque :

- Selon la configuration de fédération, le paramètre le plus restrictif est implémenté. Par exemple, si vous définissez la configuration de fédération pour forcer un utilisateur à l'authentification, la définition de l'élément ForceAuthn sur `false` n'est pas implémentée.
- Si vous comptez utiliser la gestion de cookie WebSEAL avec SAML 2.0 ForceAuthn, assurez-vous que la liste des cookies gérés n'inclut pas le cookie de session WebSphere. Voir «Configuration de WebSEAL pour gérer les cookies», à la page 591.

IsPassive

Indique si le fournisseur d'identité doit prendre le contrôle de l'agent utilisateur si la valeur est définie sur `true`. Le fournisseur d'identité n'est pas autorisé à demander à l'utilisateur ses données d'identification de connexion.

La valeur par défaut est `false`.

Remarque : Selon la configuration de fédération, le paramètre le plus restrictif est implémenté. Par exemple, si vous définissez la configuration de fédération pour empêcher le fournisseur d'identité de prendre le contrôle de l'agent utilisateur, la définition de l'élément IsPassive sur `false` n'est pas implémentée.

AuthnContextClassRef

Indique une ou plusieurs valeurs de chaînes qui identifient les références URI de classe de contexte d'authentification.

Remarque : Utilisez `AuthnContextClassRef` ou `AuthnContextDeclRef`. Si les deux valeurs sont fournies, c'est la valeur `AuthnContextClassRef` qui est utilisée.

AuthnContextDeclRef

Indique une ou plusieurs valeurs de chaînes qui identifient les références URI de déclaration de contexte d'authentification.

Remarque : Utilisez `AuthnContextClassRef` ou `AuthnContextDeclRef`. Si les deux valeurs sont fournies, c'est la valeur `AuthnContextClassRef` qui est utilisée.

AuthnContextComparison
Indique le type de comparaison utilisé pour déterminer les déclarations ou classes de contexte demandées. Le type de comparaison doit être l'une des variables suivantes :

- exact
- minimum
- maximum
- better

La valeur par défaut est exact.

AttributeConsumerSvcIndex
Indique l'index de l'ensemble d'attributs à renvoyer. Cet attribut ne correspond à aucune configuration. Les administrateurs peuvent utiliser l'attribut `AttributeConsumerSvcIndex` pour sélectionner les attributs d'identité utilisateur à inclure dans le jeton utilisateur durant la phase de mappage d'identité.

Cet attribut est pris en charge à la fois sur le fournisseur d'identité et sur le fournisseur de services.

AssertionConsumerSvcIndex
Indique l'index de l'adresse URL du service d'assertion client à laquelle le fournisseur d'identité envoie la réponse. La valeur doit correspondre au noeud final des métadonnées du fournisseur de services.

Cet attribut est pris en charge à la fois sur le fournisseur d'identité et sur le fournisseur de services.

Remarque : Si les attributs `ResponseBinding` et `AssertionConsumerSvcIndex` sont spécifiés, ce dernier prévaut.

Exemple

Adresse URL de connexion unique initialisée au niveau du fournisseur de services :
L'exemple suivant illustre l'adresse URL de connexion unique initialisée au niveau d'un fournisseur de service. Le nom de la fédération est `spfed`, et utilise le protocole SAML 2.0, HTTPPost comme liaison de demandes et liaison de réponses et une application cible à l'adresse `https://sp.example.com:9443/banking`:

```
https://sp.example.com:9443/sps/
spfed/saml20/logininitial?
RequestBinding=HTTPPost&
ResponseBinding=HTTPPost&
NameIdFormat=persistent&
IsPassive=true&
ForceAuthn=true&
AllowCreate=true&
RequestedAuthnContextComparison=minimum&
AuthnContextClassRef=classref1&
AttributeConsumerSvcIndex=1&
Target=https://sp.example.com:9443/banking
```

Adresse URL initiale du service de connexion unique (fournisseur d'identité)

Dans une fédération SAML 2.0, les URL du service de connexion unique peuvent être initiées au niveau du site du fournisseur d'identité, ou celui du fournisseur de services. La présente rubrique décrit la syntaxe d'initiation du service au niveau du fournisseur d'identité.

Syntaxe d'initialisation de connexion unique sur le fournisseur d'identité

```
https://nom_hôte_fournisseur:numéro_port/sps/
nom_fédération/saml20/logininitial?RequestBinding=RequestBindingType&
PartnerId=ID_fournisseur_partenaire_cible
&NameIdFormat=NameIDFormatType&AllowCreate=[true|false]
&Target=emplacement_application_cible
```

Eléments

https *ou* **http**
Schéma d'URI. Utilisez `https` pour les ressources qui sont protégées par SSL (secure sockets layer). Utilisez`http` pour les ressources qui ne sont pas protégées par SSL.

nom_hôte_fournisseur
Nom d'hôte du serveur point de contact du fournisseur.

numéro_port
Numéro de port du noeud final de service de transfert inter-sites. Le valeur par défaut est 9443.

sps Désignation du serveur Tivoli Federated Identity Manager. Cet élément ne peut pas être modifié.

nom_fédération
Nom affecté à la fédération lorsque vous la créez.

saml20
Désignation de la fédération SAML 2.0.

logininitial
Cet élément indique le type de noeud final qui utilise le port. Le paramètre **logininital** initialise le service de connexion unique.

Cible Cet élément est codé dans l'URL et défini comme la valeur du paramètre **`RelayState`** dans la réponse non sollicitée envoyée par le fournisseur d'identité au fournisseur de services. Une Le fournisseur de services Tivoli Federated Identity Manager interprète cette valeur comme l'adresse URL de l'application à laquelle un utilisateur peut se connecter à l'aide d'une connexion unique.

L'URL doit contenir les chaînes de requête suivantes :

RequestBinding
Liaison utilisée pour envoyer la réponse au fournisseur de services. Les valeurs admises pour l'initialisation de connexion unique sur le fournisseur d'identité sont les suivantes :

- HTTPPost
- HTTPArtifact

PartnerId
ID de fournisseur du partenaire cible.

NameIdFormat

Format utilisé pour les identificateurs de nom. Les valeurs admises sont :

- Transient (anonyme)
- Persistent
- Encrypted (pour les identificateurs de noms chiffrés)
- Email

Le paramètre 'Persistent' est défini par défaut. Si l'attribut 'NameIdFormat' n'est pas inclus, un ID de nom persistant est spécifié.

AllowCreate

Indique si une nouvelle liaison de compte persistante doit être appliquée à la demande. La valeur par défaut est False.

Remarque : Pour permettre l'utilisation de ce paramètre, vous devez définir **NameIdFormat** sur "Persistent".

AttributeConsumerSvcIndex

Indique l'index de l'ensemble d'attributs à renvoyer. Cet attribut ne correspond à aucune configuration. Les administrateurs peuvent utiliser `AttributeConsumerSvcIndex` pour sélectionner les attributs d'identité utilisateur à inclure dans le jeton utilisateur durant la phase de mappage d'identité.

Cet attribut est pris en charge à la fois sur le fournisseur d'identité et sur le fournisseur de services.

AssertionConsumerSvcIndex

Indique l'index de l'adresse URL du service d'assertion client à laquelle le fournisseur d'identité envoie la réponse. La valeur doit correspondre au noeud final des métadonnées du fournisseur de services.

Cet attribut est pris en charge à la fois sur le fournisseur d'identité et sur le fournisseur de services.

Remarque : Si les attributs `ResponseBinding` et `AssertionConsumerSvcIndex` sont spécifiés, ce dernier prévaut.

Exemple

Adresse URL de connexion unique initialisée au niveau du fournisseur d'identité :

L'exemple suivant illustre l'adresse URL de connexion unique initialisée au niveau d'un fournisseur d'identité, à l'aide du protocole SAML 2.0. `AssertionConsumerSvcIndex` désigne l'index de l'adresse URL ACS pour l'envoi de la réponse. `AttributeConsumerServiceIndex` désigne l'index ou l'ensemble d'attributs à renvoyer.

```
https://ip/FIM/sps/
saml20/saml20/logininitial?
RequestBinding=HTTPArtifact&
NameIdFormat=persistent&
AllowCreate=true&
AssertionConsumerSvcIndex=0&
AttributeConsumerSvcIndex=1&
PartnerId=https://sp/FIM/sps/saml20/saml20&
Target=https://sp.example.com:9443/banking
```

URL initiale du service SLO

Dans une fédération SAML 2.0, l'adresse URL du service SLO est utilisée par un partenaire afin de contacter le profil Single Logout. L'URL d'initialisation du service comporte la syntaxe suivante :

Syntaxe

```
https://nom_hôte_fournisseur:numéro_port/sps/
  nom_fédération/saml20/sloinitial
..?RequestBinding=RequestBindingType
```

Eléments

https *ou* **http**
: Schéma d'URI. `https` pour les ressources qui sont protégées par SSL (Secure Socket Layer). `http` pour les ressources qui ne sont pas protégées par SSL.

nom_hôte_fournisseur
: Nom d'hôte du serveur point de contact pour le fournisseur de services ou d'identité.

numéro_port
: Numéro de port du noeud final de service de résolution des artefacts. La valeur par défaut est 9444.

sps
: Désignation du serveur Tivoli Federated Identity Manager. Cet élément ne peut pas être modifié.

nom_fédération
: Nom affecté à la fédération lorsque vous la créez.

saml20
: Désignation de SAML 2.0 dans votre fédération.

sloinitial
: Cet élément indique le type de noeud final qui utilise le port. Le paramètre **sloinitial** initialise le service de déconnexion unique.

Les requêtes suivantes doivent également être incluses :

RequestBinding
: Liaison utiliser pour envoyer la requête. Les valeurs admises sont les suivantes :
  - HTTPPost
  - HTTPRedirect
  - HTTPArtifact
  - HTTPSOAP

Exemples

Adresse URL de déconnexion unique initialisée au niveau du fournisseur de services :
: L'exemple suivant illustre l'adresse URL de déconnexion unique initialisée au niveau d'un fournisseur de services au sein d'une fédération nommée `spfed`, qui utilise le protocole SAML 2.0 et le type de liaison de requête HTTPRedirect Artifact :

```
https://sp.example.com:9443/sps/spfed/saml20/sloinitial?
  RequestBinding=HTTPRedirect
```

Adresse URL de déconnexion unique initialisée au niveau du fournisseur d'identité :

L'exemple suivant illustre l'adresse URL de déconnexion unique initialisée au niveau d'un fournisseur d'identité au sein d'une fédération nommée `ipfed`, qui utilise le protocole SAML 2.0 et le type de liaison de requête HTTPArtifact :

```
https://idp.example.com:9444/sps/ipfed/saml20/sloinitial?
  RequestBinding=HTTPArtifact
```

URL initiale du service de gestion des identificateurs de nom

Dans une fédération SAML 2.0, l'adresse URL du service de gestion des identificateurs de noms est utilisée par un partenaire afin de contacter le service Name Identifier Management.

Syntaxe

L'URL d'initialisation du service comporte la syntaxe suivante :

```
https://nom_hôte_fournisseur:numéro_port/sps/
  nom_fédération/mnidsinitial?RequestBinding=RequestBindingType
  &PartnerId=ID_fournisseur_partenaire_cible&NameIdTerminate=[True|False]
```

Eléments

https *ou* **http**

Schéma d'URI. `https` pour les ressources qui sont protégées par SSL (Secure Socket Layer). `http` pour les ressources qui ne sont pas protégées par SSL.

nom_hôte_fournisseur

Nom d'hôte du serveur point de contact pour le fournisseur de services ou d'identité.

numéro_port

Numéro de port du noeud final de service de résolution des artefacts. La valeur par défaut est 9444.

sps

Désignation du serveur Tivoli Federated Identity Manager. Cet élément ne peut pas être modifié.

nom_fédération

Nom affecté à la fédération lorsque vous la créez.

saml20

Désignation de SAML 2.0 dans la fédération.

mnidsinitial

Cet élément indique le type de noeud final qui utilise le port. La propriété **mnidsinitial** est utilisée pour initialiser l'identificateur de noms.

Les chaînes de requête suivantes doivent également être incluses :

RequestBinding

Liaison utiliser pour envoyer la requête au partenaire. Les valeurs admises pour l'initialisation de connexion unique sur le fournisseur d'identité sont les suivantes :

- HTTPPost
- HTTPArtifact
- HTTPRedirect
- HTTPSOAP

PartnerId
ID de fournisseur du partenaire cible.

NameIdTerminate
Valeur indiquant si le flux de gestion des identificateurs de nom doit arrêter le mappage des identificateurs de nom. Les valeurs admises sont :

True Arrête la liaison des comptes.

False Indique si le flux des identificateurs de nom met à jour les identificateurs de nom (alias). La valeur 'False' est définie par défaut si aucune valeur explicite n'est spécifiée.

Exemples

Identificateur de nom initialisé au niveau du fournisseur d'identité :
L'exemple suivant illustre l'adresse URL de l'identificateur de noms initialisée au niveau d'un fournisseur d'identité au sein d'une fédération nommée `ipfed`, qui utilise le protocole SAML 2.0 et le type de liaison de requête HTTP SOAP :

```
https://idp.example.com:9444/sps/ipfed/saml20/mnidsinitial?
  RequestBinding=HTTPSOAP&PartnerId=https://saml20sp:444/sps/
  saml20/saml20&NameIdTerminate=true
```

Identificateur de nom initialisé au niveau du fournisseur de services :
L'exemple suivant illustre l'adresse URL de l'identificateur de noms initialisée au niveau d'un fournisseur de services au sein d'une fédération nommée `spfed`, qui utilise le protocole SAML 2.0 et le type de liaison de requête HTTP Artifact :

```
https://sp.example.com:9444/sps/spfed/saml20/mnidsinitial?
  RequestBinding=HTTPArtifact&PartnerId=https://saml20ip/FIM/sps/
  saml20/saml20&NameIdTerminate=true
```

Annexe C. Utilisation de l'interface de ligne de commande pour la configuration de la prise en charge SHA256 Tivoli Federated Identity Manager

Apprenez à configurer les paramètres requis provenant du fichier de réponses pour la prise en charge de SHA256 dans Tivoli Federated Identity Manager.

Procédure

1. Cliquez sur **Integrated Solutions Console** > **Tivoli Federated Identity Manager** pour effectuer les tâches suivantes :
   a. Créez des fédérations de fournisseur d'identité et de fournisseur de services SAML 2.0. Voir «Création de votre rôle dans la fédération», à la page 234.
   b. Exportez le fichier de métadonnées.
   c. Ajoutez des partenaires. Veillez à bien sélectionner une clé et un algorithme de signature. Voir le tableau 169 pour plus d'informations sur les attributs SHA256.

Tableau 169. Matrice de configuration des paramètres SHA256 SAML 2.0

Attributs SHA256 SAML 2.0	Valeurs	Applicable ?	Remarques
`SigningKeyIdentifier`	Oui	Fournisseurs d'identité et fournisseur de services	Signe les messages SAML sortants et l'assertion SAML. Si l'attribut `AssertionSigningKeyIdentifier` est défini, `AssertionSigningKeyIdentifier` signe alors plutôt l'assertion SAML. Vous pouvez configurer l'attribut dans la console de gestion et le fichier de réponses de la fédération.
`AssertionSigningKeyIdentifier`	Par exemple : `DefaultKeyStore_ dsatestkey`	IP uniquement	Signe l'assertion SAML sortante. L'attribut n'est configurable que dans le fichier de réponses de la fédération.

`SignatureAlgorithm`	`http://www.w3.org/2000/09/xmldsig#dsa-sha1`
	
	`http://www.w3.org/2000/09/xmldsig#rsa-sha1`
	
	`http://www.w3.org/2001/04/xmldsig-more#rsa-sha256`
		Fournisseurs d'identité et fournisseur de services	Signe les messages SAML sortants et l'assertion SAML.
			
			Si l'attribut `AssertionSignatureAlgorithm` est défini, `AssertionSignatureAlgorithm` signe alors plutôt l'assertion SAML.
			
			La valeur `SignatureAlgorithm` doit correspondre au type de clé indiqué dans `SigningKeyIdentifier`.
			
			Si `AssertionSigningKeyIdentifier` est défini et que `AssertionSignatureAlgorithm` ne l'est pas, la valeur `SignatureAlgorithm` doit correspondre au type de clé indiqué dans `AssertionSigningKeyIdentifier`.
			
			Vous pouvez configurer l'attribut dans la console de gestion et le fichier de réponses de partenaire.

Tableau 169. Matrice de configuration des paramètres SHA256 SAML 2.0 (suite)

Attributs SHA256 SAML 2.0	Valeurs	Applicable ?	Remarques
DigestAlgorithm	http://www.w3.org/2000/09/xmldsig#sha1 http://www.w3.org/2001/04/xmlenc#sha256 http://www.w3.org/2001/04/xmlenc#sha512	Fournisseurs d'identité et fournisseur de services	Génère des messages SAML et des valeurs de prétraitement d'assertion SAML. Si AssertionDigestAlgorithm est défini, AssertionDigestAlgorithm hache le prétraitement d'assertion SAML. S'il ne l'est pas, DigestAlgorithm devient : • SHA1, lorsque SignatureAlgorithm correspond à DSA-SHA1 ou RSA-SHA1 • SHA256, lorsque SignatureAlgorithm est RSA-SHA256. L'attribut n'est configurable que dans le fichier de réponses de partenaire.
AssertionSignatureAlgorithm	http://www.w3.org/2000/09/xmldsig#dsa-sha1 http://www.w3.org/2000/09/xmldsig#rsa-sha1 http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	IP uniquement	Signe l'assertion SAML sortante. La valeur doit correspondre au type de clé de l'attribut AssertionSigningKeyIdentifier. SignatureAlgorithm signe l'assertion SAML si AssertionSignatureAlgorithm n'est pas défini. L'attribut n'est configurable que dans le fichier de réponses de partenaire.
AssertionDigestAlgorithm	http://www.w3.org/2000/09/xmldsig#dsa-sha1 http://www.w3.org/2000/09/xmldsig#rsa-sha1 http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	IP uniquement	Signe l'assertion SAML sortante. La valeur doit correspondre au type de clé de l'attribut AssertionSigningKeyIdentifier. SignatureAlgorithm signe l'assertion SAML si AssertionSignatureAlgorithm n'est pas défini. L'attribut n'est configurable que dans le fichier de réponses de partenaire.
AssertionValidateKeyIdentifier	Par exemple : DefaultTrustedKeyStore_IP-validationkey	Fournisseur de services uniquement	Clé utilisée par le fournisseur de services pour valider l'assertion SAML à partit du fournisseur d'identité. AssertionValidateKeyIdentifier doit être identique à la clé publique du fournisseur d'identité lors de la signature d'une assertion SAML. L'attribut n'est configurable que dans le fichier de réponses de partenaire.

2. Dans l'interface de ligne de commande, générez les fichiers de réponses de la fédération du fournisseur et du fournisseur de services, ainsi que ceux des partenaires que vous avez ajoutés. Utilisez les commandes suivantes :

   Fichier de réponse de la fédération du fournisseur d'identité :

   ```
   wsadmin>$AdminTask manageItfimFederation
   {-operation createResponseFile -fimDomainName <fimdomain>
   -federationName <IP_fedname> -fileId output_file}
   ```

   Fichier de réponse de la fédération du fournisseur de services :

   ```
   wsadmin>$AdminTask manageItfimFederation
   {-operation createResponseFile -fimDomainName <fimdomain>
   -federationName <SP_fedname> -fileId output_file}
   ```

   Fichier de réponse du partenaire fournisseur d'identité :

   ```
   wsadmin>$AdminTask manageItfimPartner
   {-operation createResponseFile -fimDomainName <fimdomain>
   -federationName <IP_fedname> -partnerName <Partner_name>
   -fileId output_file}
   ```

Fichier de réponse du partenaire fournisseur de services :

```
wsadmin>$AdminTask manageItfimPartner
{-operation createResponseFile -fimDomainName <fimdomain>
-federationName <SP_fedname> -partnerName <Partner_name>
-fileId output_file}
```

3. Modifiez les attributs SHA256 SAML 2.0 dans les fichiers de réponse du fournisseur d'identité, de la fédération du fournisseur de services et du partenaire en fonction des données disponibles dans le tableau 170.

Tableau 170. Paramètres des fichiers de réponse de la fédération SHA256 du fournisseur d'identité et du fournisseur de services et du partenaire

Paramètres SHA256 SAML 2.0	Fournisseur d'identité	Fournisseur de services
Fédération		
SigningKeyIdentifier	<void method="put"> <string>SigningKeyIdentifier </string> <object class="java.util.ArrayList"> <void method="add"> <string>DefaultKeyStore_<*dsakey*> </string> </void> </object> </void>	<void method="put"> <string>SigningKeyIdentifier </string> <object class="java.util.ArrayList"> <void method="add"> <string>DefaultKeyStore_<rsakey> </string> </void> </object> </void>
AssertionSigningKeyIdentifier	<void method="put"> <string>AssertionSigningKeyIdentifier </string> <object class="java.util.ArrayList"> <void method="add"> <string>DefaultKeyStore_<*rsakey*> </string> </void> </object> </void>	N/A
Partenaire		
AssertionValidateKeyIdentifier	N/A	<void method="put"> <string>AssertionValidateKeyIdentifier </string> <object class="java.util.ArrayList"> <void method="add"> <string>DefaultTrustedKeyStore_ <*IP_publickey*> </string> </void> </object> </void>
SignatureAlgorithm	<void method="put"> <string>SignatureAlgorithm </string> <object class="java.util.ArrayList"> <void method="add"> <string>http://www.w3. org/2000/09/ xmldsig#dsa-sha1 </string> </void> </object> </void>	<void method="put"> <string>SignatureAlgorithm </string> <object class="java.util.ArrayList"> <void method="add"> <string>http://www.w3.org/2001/04/ xmldsig-more#rsa-sha256 </string> </void> </object> </void>

Tableau 170. Paramètres des fichiers de réponse de la fédération SHA256 du fournisseur d'identité et du fournisseur de services et du partenaire (suite)

Paramètres SHA256 SAML 2.0	Fournisseur d'identité	Fournisseur de services
DigestAlgorithm	<void method="put"> <string>DigestAlgorithm </string> <object class="java.util.ArrayList"> <void method="add"> <string>http://www.w3. org/2000/09/ xmldsig#sha1 </string> </void> </object> </void>	<void method="put"> <string>DigestAlgorithm </string> <object class="java.util.ArrayList"> <void method="add"> <string>http://www.w3.org/2001/04/ xmlenc#sha512 </string> </void> </object> </void>
AssertionSignatureAlgorithm	<void method="put"> <string>AssertionSignatureAlgorithm </string> <object class="java.util.ArrayList"> <void method="add"> <string>http://www.w3. org/2001/04/ xmldsig-more#rsa-sha256 </string> </void> </object> </void>	<void method="put"> <string>AssertionSignatureAlgorithm </string> <object class="java.util.ArrayList"/> </void>
AssertionDigestAlgorithm	<void method="put"> <string>AssertionDigestAlgorithm </string> <object class="java.util.ArrayList"> <void method="add"> <string>http://www.w3.org/2001/04/ xmlenc#sha512 </string> </void> </object> </void>	<void method="put"> <string>AssertionDigestAlgorithm </string> <object class="java.util.ArraList"/> </void>

4. Mettez à jour les propriétés du fournisseur d'identité et du fournisseur de services à l'aide du fichier de réponse modifié de la fédération.

   Pour le fournisseur d'identité :

   ```
   wsadmin>$AdminTask manageItfimFederation
   {-operation modify -fimDomainName <fimdomain>
   -federationName <IP_fedname>
   -fileId <Path_to_IP_federation_response_file>}
   ```

   Pour le fournisseur de services :

   ```
   wsadmin>$AdminTask manageItfimFederation
   {-operation modify -fimDomainName <fimdomain>
   -federationName <SP_fedname>
   -fileId <Path_to_SP_federation_response_file>}
   ```

5. Mettez à jour les propriétés du fournisseur d'identité et du partenaire fournisseur de services à l'aide du fichier de réponse modifié du partenaire.

   Pour le fournisseur d'identité :

   ```
   wsadmin>$AdminTask manageItfimPartner
   {-operation modify -fimDomainName <fimdomain>
   -federationName <IP_fedname>
   -partnerName <IP_partner_name>
   -fileId <Path_to_IP_partner_response_file>}
   ```

Pour le fournisseur de services :

```
wsadmin>$AdminTask manageItfimPartner
{-operation modify -fimDomainName <fimdomain>
-federationName <SP_fedname>
-partnerName <SP_partner_name>
-fileId <Path_to_SP_partner_response_file>}
```

6. Activez les partenaires fournisseur d'identité et fournisseur de services.

   Pour le fournisseur d'identité :

   ```
   wsadmin>$AdminTask manageItfimPartner
   {-operation enable -fimDomainName <fimdomain>
   -federationName <IP_fedname>
   -partnerName <IP_partner_name>}
   ```

   Pour le fournisseur de services :

   ```
   wsadmin>$AdminTask manageItfimPartner
   {-operation enable -fimDomainName <fimdomain>
   -federationName <SP_fedname>
   -partnerName <SP_partner_name>}
   ```

7. Effectuez une connexion unique ou une déconnexion unique.

Annexe D. Désactivation de la consignation en vue d'améliorer les performances

Lors de l'utilisation de Tivoli Federated Identity Manager avec Tivoli Access Manager, vous pouvez améliorer les performances obtenues sur un fournisseur de services en désactivation la fonction de consignation du serveur de règles Tivoli Access Manager.

Pour réduire la sollicitation de l'unité centrale (CPU), procédez comme suit :

1. Sauvegardez le répertoire de Policy Director. Par exemple, sous Linux ou UNIX :

   ```
   /opt/IBM/WebSphere/AppServer/java/jre/PolicyDirector
   ```

2. Ouvrez le fichier suivant dans un éditeur de texte:

   ```
   /opt/IBM/WebSphere/AppServer/java/jre/PolicyDirector/PDJLog.properties
   ```

3. Désactivez la consignation des messages en configurant le paramètre suivant sur la valeur 'false' :

   ```
   baseGroup.PDJMessageLogger.isLogging=false
   ```

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, un programme ou un service IBM n'implique pas que seul ce produit, programme ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Toutefois, il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM est susceptible de posséder des brevets ou des applications brevetées en attente qui couvrent le sujet décrit dans ce document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Pour les demandes de licence concernant les informations à deux octets (DBCS), contactez le Département des propriétés intellectuelles IBM de votre pays ou envoyez vos demandes, par courrier, à :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Certaines juridictions n'autorisent pas l'exclusion des garanties implicites ou explicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément au contrat sur les produits et services IBM, aux conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation IBM.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines Corp. dans de nombreux pays. Les autres noms de produits et services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark information" à l'adresse www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript ainsi que toutes les marques incluant Adobe sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.

IT Infrastructure Library est une marque de Central Computer and Telecommunications Agency qui fait désormais partie de The Office of Government Commerce.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium et Pentium sont des marques d'Intel Corporation ou de ses filiales aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans d'autres pays.

ITIL est une marque de The Office of Government Commerce, et est enregistrée au bureau américain U.S. Patent and Trademark Office.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et ou de ses sociétés affiliées.

Cell Broadband Engine est une marque de Sony Computer Entertainment, Inc. aux Etats-Unis et/ou dans d'autres pays et est utilisée sous licence.

Linear Tape-Open, LTO, le logo LTO, Ultrium et le logo Ultrium sont des marques de HP, IBM Corp. et Quantum aux Etats-Unis et dans d'autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Glossaire

accord d'autorisation
Dans le contexte de OAuth, attribution représentant l'autorisation du propriétaire de ressource pour l'accès à ses ressources protégées. Les clients OAuth utilisent un accord d'attribution pour obtenir un jeton d'accès. Il existe quatre types d'accord d'autorisation : code d'autorisation, implicite, droits d'accès par mot de passe de propriétaire de ressource et droits d'accès client.

artefact
Dans le contexte du protocole SAML, un objet de données structurées qui désigne un message de protocole SAML.

artefact du navigateur
Un profil (qui est en fait un ensemble de règles) basé sur la norme SAML et qui indique qu'un artefact est échangé pour établir et utiliser une session digne de confiance standard entre deux partenaires au sein d'une fédération. Par opposition au *POST du navigateur*.

assertion
Dans le contexte du protocole SAML, des données qui contiennent des informations d'authentification et/ou d'attribut dans un message.

canal de retour SOAP
Communications qui ont lieu directement entre deux noeuds finals SOAP.

certificat
Dans le domaine de la sécurité informatique, document numérique associant une clé publique à l'identité du propriétaire d'un certificat. Ce document numérique permet l'authentification du propriétaire du certificat. Un certificat est émis par une autorité de certification et est signé numériquement par cette autorité.

clé privée
En communication sécurisée, schéma algorithmique utilisé pour chiffrer des messages que seule la clé publique correspondante peut déchiffrer. La clé privée est également utilisée pour déchiffrer des messages qui ont été chiffrés par la clé publique correspondante. La clé privée est conservée sur le système utilisateur et est protégée par un mot de passe.

clé publique
En communication sécurisée, schéma algorithmique utilisé pour déchiffrer des messages qui ont été chiffrés par la clé privée correspondante. Une clé publique est également utilisée pour chiffrer des messages que seule la clé privée correspondante permet de déchiffrer. Les utilisateurs diffusent leurs clés publiques à tous ceux avec qui ils doivent échanger des messages chiffrés.

client logiciel ou ordinateur qui demande des services d'un serveur.

client OAuth
Application tierce souhaitant accéder aux ressources privées du propriétaire des ressources. Le client OAuth peut effectuer des demandes de ressources protégées pour le compte du propriétaires des ressources, une fois que le propriétaire des ressources lui en accordé l'autorisation.

code d'autorisation
Dans le contexte de OAuth, code que le serveur d'autorisation génère lorsque le propriétaire de la ressource autorise une requête.

connexion unique
Processus d'authentification par lequel l'utilisateur peut accéder à plusieurs systèmes ou applications en entrant un ID utilisateur et un mot de passe uniques.

demande
Elément qui lance un flux de travaux et les diverses activités d'un flux de travaux.

domaine
Déploiement du composant d'exécution Tivoli Federated Identity Manager sur WebSphere Application Server.

fédération
Relation dans laquelle des entités, telles que des entreprises distinctes, conviennent d'utiliser la même norme

technique (telle que SAML ou Liberty). Cette norme technique permet à chaque partenaire dans la relation d'accéder aux ressources et aux données de l'autre. Voir aussi fournisseur d'identité et fournisseur de services.

fichier de clés
Dans le domaine de la sécurité, fichier ou carte cryptographique matérielle où sont stockées des identités et des clés privées à des fins d'authentification et de chiffrement. Certains fichiers de clés contiennent également des clés certifiées ou publiques.

fichier de réponses
Fichier contenant des valeurs prédéfinies telles que les paramètres et les valeurs utilisées pour contrôler les actions d'un composant d'une façon prédéterminée.

fournisseur de services
Partenaire au sein d'une fédération qui fournit des services aux utilisateurs.

fournisseur d'identité
Partenaire au sein d'une fédération qui a la responsabilité d'authentifier l'identité d'un utilisateur.

gestion de la sécurité des services Web
Composant Tivoli Federated Identity Manager utilisé pour établir et gérer les relations avec les fédérations pour les applications de service Web s'exécutant sur le serveur WebSphere Application Server et utilisant des jetons WS-Security.

jeton Message ou suite de bits spécifique qui indique la permission ou le contrôle temporaire relativement à l'envoi d'informations sur un réseau. Dans le contexte du protocole SAML, jeton est utilisé dans le même sens que le mot *assertion*.

jeton d'accès
Dans le contexte de OAuth, chaîne représentant l'autorisation fournie au client OAuth. Cette chaîne représente des portées et des durées d'accès. Elle est fournie par le propriétaire de la ressource, et appliquée par le serveur OAuth ou le serveur d'autorisation.

jeton d'actualisation
Dans le contexte de OAuth, chaîne utilisée pour obtenir un nouveau jeton d'accès à l'expiration du jeton d'accès en cours.

liaison
Dans le contexte du protocole SAML, la méthode de communication utilisée pour transporter les messages.

mappage d'identité
Processus de modification d'une identité qui est valide dans un contexte d'entrée en une identité qui l'est dans un contexte de sortie.

métadonnées
Données décrivant une information spécifique, comme les paramètres d'une configuration par exemple.

noeud final
Le destinataire final d'une opération.

partenaire
En communication de données, programme d'application distant ou l'ordinateur distant.

POST du navigateur
Profil (c'est-à-dire, ensemble de règles) de la norme SAML qui permet d'utiliser un formulaire à envoi automatique lors de l'établissement et l'utilisation d'une session sécurisée entre deux partenaires d'une fédération. Par opposition à l'*artefact du navigateur*.

profil Dans le contexte de la spécification SAML, une combinaison de protocoles, d'assertions et de liaisons qui sont utilisés ensemble pour créer une fédération et activer la connexion unique fédérée.

propriétaire de la ressource
Dans le contexte de OAuth, type d'utilisateur capable d'autoriser l'accès à une ressource protégée.

protocole
Dans le contexte de la spécification SAML, un type de message de demande et de message de réponse utilisé pour obtenir des données d'identification et pour gérer des identités.

SAML Voir *security assertion markup language*.

section
Groupe de lignes d'un fichier qui ont une fonction commune ou définissent une partie du système. Les sections sont

séparées par des lignes vierges ou des points virgule, et chaque section a un nom.

security assertion markup language
Ensemble de spécifications écrites par le consortium OASIS dans le but de décrire le traitement sécurisé des messages de demande et de réponse basés sur XML et qui contiennent des informations d'autorisation et d'authentification.

serveur d'autorisation
Serveur qui traite l'autorisation et les authentifications.

serveur de ressources
Serveur qui héberge les ressources protégées. Il peut accepter et répondre aux demandes de ressources protégées à l'aide de jetons d'accès. Le serveur de ressources peut être le même serveur que le serveur d'autorisation.

serveur OAuth
Egalement appelé **Serveur d'autorisation** dans OAuth 2.0. Serveur qui fournit aux clients OAuth un accès à portée définie à une ressource protégée pour le compte du propriétaire de la ressource. Un serveur d'autorisation peut également être le serveur de ressources.

serveur point de contact
Dans le contexte d'une fédération, le serveur de proxy ou d'application qui correspond à la première entité traitant une demande d'accès à une ressource.

service d'accréditation
Composant de Tivoli Federated Identity Manager qui gère les jetons de sécurité qui sont échangés entre les domaines de sécurité. Le service d'accréditation est également appelé *Service de jeton de sécurité*.

service d'alias
Composant de Tivoli Federated Identity Manager qui gère les alias ou les identificateurs de nom qui sont transférés entre différents domaines sécurisés.

service d'assertion client
Dans le contexte du protocole SAML, le noeud final d'une fédération qui reçoit des assertions ou des artefacts dans le cadre d'une requête ou d'une réponse à connexion unique.

service de résolution d'artefact
Dans le contexte du protocole SAML, le noeud final d'une fédération où les artefacts sont remplacés par des assertions.

service de transfert inter-sites
Dans le contexte du protocole SAML, le noeud final d'une fédération auquel une requête à connexion unique est envoyée.

service Web
Application modulaire autonome et auto-descriptive qui peut être publiée, découverte et appelée sur un réseau à l'aide de protocoles réseau standard. En général, XML est utilisé pour baliser les données et SOAP est utilisé pour transférer les données. Un fichier WSDL est utilisé pour décrire les services disponibles, et UDDI est utilisé pour répertorier les services qui sont disponibles.

Simple and Protected GSS API Negotiation Mechanism (SPNEGO)
Mécanisme d'authentification qui fournit une fonctionnalité de connexion unique dans les environnements Microsoft Windows.

SOAP Protocole XML simple conçu pour l'échange d'informations dans un environnement décentralisé et distribué. SOAP peut être utilisé pour interroger et renvoyer des informations, et démarrer des services sur Internet.

SPNEGO
Simple and Protected GSS API Negotiation Mechanism

syntaxe
Règles de construction d'une commande ou d'une instruction.

Index

Caractères spéciaux

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

GC11-6781-02