

Kofax RPA

Guide de l'administrateur

Version : 11.2.0

Date : 2021-07-09

The logo for KOFAX, consisting of the word "KOFAX" in a bold, blue, sans-serif font.

© 2015–2021 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

Table des matières

Préface	6
Documentation connexe.....	6
Configuration requise.....	7
Formation.....	8
Obtenir de l'aide sur les produits Kofax.....	8
Chapitre 1 : Temps d'exécution	9
RoboServer.....	9
Démarez RoboServer.....	10
Configuration de la production.....	13
Configuration de RoboServer.....	16
Démarrer et entrer la licence dans le Management Console intégré.....	18
Configuration Management Console intégrée.....	19
Gestion des utilisateurs.....	20
Sécurité.....	20
Configuration TLS RoboServer.....	21
Restrictions.....	21
Requête d'authentification.....	22
Configurer la Journalisation RoboServer.....	22
Certificats.....	23
Installer des certificats HTTPS.....	25
Installer des certificats de client HTTPS.....	25
Installer des certificats de client API.....	26
Installer un certificat du serveur API.....	27
Configuration RoboServer – Mode sans tête.....	27
Mode JMS.....	30
Management Console en mode JMS.....	31
Options du RoboServer en mode JMS.....	32
Files d'attente et sujets.....	32
Configuration du serveur JMX.....	34
Projet RoboServer par défaut.....	35
Modifier l'allocation de la RAM.....	35
Dépannage du démarrage de RoboServer Service.....	36
Chapitre 2 : Tomcat Management Console	37
Déploiement de Tomcat.....	37

Installez Management Console sur Tomcat.....	38
Configurer ManagementConsole.war.....	39
Fichiers de configuration Spring.....	39
Dépannage.....	40
Créer une nouvelle base de données.....	40
Créer un fichier contextuel Tomcat.....	42
Démarrer Tomcat.....	44
Saisir les informations relatives à la licence.....	45
Rôles prédéfinis des utilisateurs.....	45
Autorisations de projets.....	48
Sécurité.....	50
Liste de contrôle pour le déploiement.....	51
Outils Docker pour déploiement Kofax RPA.....	52
Remarques pour Windows Docker.....	53
Déployez Kofax RPA en utilisant les fichiers docker-compose.....	55
Exemples de docker-compose.....	56
Optimiser la taille du support de création Docker.....	58
Utilisez la fonction Docker secrets pour stocker les mots de passe.....	59
Configuration de la base de données.....	60
Sauvegarder et restaurer.....	61
Contrôles préalables au démarrage.....	63
Dossiers de données.....	63
Variables d'environnement.....	63
Exécuter sur Docker Swarm avec Management Console en haute disponibilité.....	64
Configuration avancée.....	67
Intégration Single Sign-On LDAP et CA.....	67
Intégration de Single Sign-On SAML.....	73
Configurer le mode JMS.....	79
Haute disponibilité :.....	82
Encodage URI.....	88
Chiffrement du mot de passe.....	88
Vérification des terminaux SSL.....	90
Sessions simultanées pour un compte d'utilisateur.....	92
Utiliser le serveur Microsoft SQL avec sécurité intégrée.....	93
Configurer le fichier WAR de Management Console.....	93
Créer un modèle avec les paramètres de Management Console.....	94
Extraire les paramètres du fichier WAR de Management Console existant.....	95
Appliquer les paramètres au fichier WAR de Management Console.....	95

Configurer le Serveur du système de fichiers du robot.....	96
Exemple : Dossier Carte vers le Système de fichiers du robot.....	96
Chapitre 3 : Exécuter les composants RPA en tant que services.....	99
ServiceInstaller.exe expliqué.....	99
Exécuter RoboServer et la Management Console en tant que service.....	100
Exécuter Synchronizer en tant que service.....	102
Chapitre 4 : WebSphere Management Console.....	104
Chapitre 5 : Journal d'audit pour Management Console.....	106
Référence du journal d'audit.....	108
Chapitre 6 : Scripts SQL pour les tables Kofax RPA.....	110
Annexe A : Modèle de sécurité Kofax RPA.....	111

Préface

Ce guide est destiné aux administrateurs système qui déploient Kofax RPA dans l'environnement de l'entreprise.

Si vous utilisez une des versions précédentes de Kofax RPA, consultez le *Guide de mise à niveau de Kofax RPA* pour les procédures de mise à jour.

Le guide comprend des informations administratives pour Kofax RPA, comme :

- [Temps d'exécution](#)
- [Tomcat Management Console](#)
- [WebSphere Management Console](#)
- [Journal d'audit pour Management Console](#)
- [Scripts SQL pour les tables Kofax RPA](#)

Documentation connexe

Le jeu de documents pour Kofax RPA est disponible ici^{1,2}.

https://docshield.kofax.com/Portal/Products/RPA/11.2.0_ea1ydbmwk9/RPA.htm

En plus de ce guide, le jeu de documents comprend les éléments suivants :

Notes de publication Kofax RPA

Contient des détails de dernière minute et d'autres informations qui ne sont pas disponibles dans votre autre documentation Kofax RPA.

Spécifications techniques Kofax RPA

Contient des informations sur les systèmes d'exploitation pris en charge et les autres exigences du système.

Guide d'installation Kofax RPA

Contient des instructions sur l'installation de Kofax RPA et de ses composants dans un environnement de développement.

¹ : Vous devez être connecté à Internet pour accéder au jeu de documents complet en ligne

² Pour l'accès sans connexion Internet, voir le *guide d'installation*.

Guide de mise à jour Kofax RPA

Contient des instructions sur la mise à jour de Kofax RPA et de ses composants vers une version plus récente.

Aide pour Kofax RPA

Décrit comment utiliser Kofax RPA. L'aide est également disponible en format PDF et connue sous le nom de *Kofax RPA Guide utilisateur*.

Guide des meilleures pratiques Kofax RPA pour la gestion du cycle de vie des robots

Propose des méthodes et des techniques recommandées pour vous aider à optimiser vos performances et à assurer votre succès tout en utilisant la gestion du cycle de vie des robots dans votre environnement Kofax RPA.

Kofax RPA Guide de mise en route pour la création de robots

Fournit un tutoriel qui explique l'utilisation de Kofax RPA pour créer un robot.

Kofax RPA Guide de démarrage Document Transformation

Fournit un tutoriel qui explique comment utiliser la fonctionnalité Document Transformation dans un environnement Kofax RPA, comme l'OCR, l'extraction, le formatage des champs et la validation.

Kofax RPA Guide de configuration du service d'automatisation des postes de travail

Décrit comment configurer le service Desktop Automation nécessaire pour utiliser Desktop Automation sur un ordinateur distant.

Guide du développeur Kofax RPA

Contient des informations sur l'API utilisée pour exécuter les robots sur RoboServer.

Documentation sur l'API d'intégration Kofax RPA

Contient des informations sur l'API Java Kofax RPA et l'API .NET Kofax RPA qui fournissent un accès programmatique au produit Kofax RPA. La documentation de l'API Java est disponible en ligne et hors ligne à l'adresse Kofax RPA, tandis que la documentation de l'API .NET n'est disponible que hors ligne.

Remarque Les API de Kofax RPA comprennent de nombreuses références à RoboSuite, le nom original du produit. Le nom de RoboSuite est conservé dans les API pour assurer la rétrocompatibilité. Dans le contexte de la documentation de l'API, le terme RoboSuite a la même signification que Kofax RPA.

Configuration requise

Pour plus d'informations sur les systèmes d'exploitation pris en charge et les autres exigences système, consultez le document *Kofax RPA Spécifications techniques* sur le [site de la documentation des produits Kofax RPA](#).

Formation

Kofax propose des formations en classe et sur ordinateur pour vous aider à tirer le meilleur parti de votre solution Kofax RPA. Viz le portail de formation Kofax à l'adresse suivante : <https://learn.kofax.com/> pour obtenir des détails sur les formations et les planifications disponibles.

Vous pouvez également consulter le Kofax Intelligent Automation SmartHub à l'adresse suivante : <https://smarthub.kofax.com/> pour explorer d'autres solutions, robots, connecteurs, etc.

Obtenir de l'aide sur les produits Kofax

Le répertoire de la [Base de connaissances Kofax](#) contient des articles qui sont régulièrement mis à jour pour vous tenir informé des produits Kofax. Nous vous encourageons à utiliser cette base de connaissances pour obtenir des réponses à vos questions sur les produits.

Pour accéder à la Kofax Knowledge Base [Base de connaissances Kofax], ouvrez le [Kofaxsite web](#) et sélectionnez **Assistance** sur la page d'accueil.

Remarque La Kofax Knowledge Base [Base de connaissances Kofax] est optimisée pour Google Chrome, Mozilla Firefox ou Microsoft Edge.

La Kofax Knowledge Base [Base de connaissances Kofax] propose :

- Puissantes fonctionnalités de recherche pour vous aider à localiser rapidement les informations dont vous avez besoin.
Saisissez vos termes ou votre phrase à rechercher dans le champ **Search** [Recherche], puis cliquez sur l'icône Loupe.
- Informations produit, détails de configuration et documentation, notamment les actualités des nouvelles versions.
Faites défiler la page d'accueil de la Kofax Knowledge Base [Base de connaissances Kofax] pour localiser une famille de produits. Cliquez ensuite sur le nom d'une famille de produits pour afficher une liste d'articles sélectionnés. Veuillez noter que certaines familles de produits nécessitent un identifiant valide du portail Kofax pour afficher ces articles sélectionnés.
- Accès au Kofax Customer Portal [Portail client Kofax] (pour les clients éligibles).
Cliquez sur le lien **Customer Support** [Assistance client] en haut de la page, puis cliquez sur **Log in to the Customer Portal** [Connexion au portail client].
- Accès au Kofax Partner Portal [Portail partenaire Kofax] (pour les partenaires éligibles).
Cliquez sur le lien **Partner Support** [Assistance partenaire] en haut de la page, puis cliquez sur **Log in to the Partner Portal** [Connexion au portail partenaire].
- Accès aux validations de l'assistance Kofax, aux politiques de cycle de vie, aux détails d'exécution électroniques et aux outils en libre-service.
Faites défiler la page jusqu'à la section **General Support** [Assistance générale], cliquez sur **Support Details** [Détails de l'assistance], puis sélectionnez l'onglet approprié.

Chapitre 1

Temps d'exécution

Kofax RPA propose un certain nombre d'outils permettant d'exécuter les robots que vous avez développés. Les sections suivantes décrivent ces outils :

- RoboServer est une application serveur qui permet à des clients distants d'exécuter des robots. Il est configuré à l'aide des applications Management Console et Paramètres RoboServer (pour la configuration avancée, telle que la sécurité et l'authentification).
- Management Console vous aide à programmer l'exécution des robots, à consulter les journaux et les données extraites. Il fournit également un lieu centralisé où les paramètres des groupes de RoboServers peuvent être configurés.

Important Après avoir modifié les paramètres de configuration de n'importe quel composant de Kofax RPA , redémarrez le composant concerné pour que les changements prennent effet.

Remarque Les définitions des fuseaux horaires sont intégrées dans le JRE groupé. En cas de modification des définitions depuis la date de publication, le JRE peut être mis à jour à l'aide de l'*outil de mise à jour des fuseaux horaires* fourni par Oracle. Pour de plus amples informations, veuillez consulter le site web d'Oracle.

RoboServer

RoboServer fait fonctionner des robots créés dans Design Studio. Les robots peuvent être démarrés de différentes manières : soit planifiés pour fonctionner à des heures précises par un Management Console, appelé via un service web REST, soit par les API Java ou .NET, soit par un Kapplet.

Important L'installation minimale de Linux doit comprendre les paquets suivants pour pouvoir faire fonctionner les robots créés avec le moteur du navigateur par défaut.

- `libX11.so.6`
- `libGL.so.1`
- `libXext.so.6`

Utilisez la commande `yum install` ou `sudo apt-get` pour installer les bibliothèques nécessaires sur une plateforme Linux.

Assurez-vous également que le système dispose de certaines polices de caractères. Cela peut être nécessaire dans le cas d'une installation de Linux sans tête, car certains des paquets d'installation de Linux ne contiennent pas de polices.

Pour pouvoir exécuter le robot, RoboServer doit être activé par un Management Console. Un RoboServer est actif lorsqu'il appartient à un cluster dans un Management Console avec une licence valide, et que suffisamment de KCU ont été attribués au cluster. Un RoboServer reçoit également les paramètres du Management Console où ils sont configurés sur les clusters. Voir le chapitre Management Console dans l'*Aide de Kofax RPA* pour plus d'informations sur l'administration des RoboServers et des clusters.

Démarrez RoboServer

Un RoboServer peut être lancé de plusieurs manières :

- En cliquant sur l'icône du programme RoboServer (ou sur l'icône « Démarrer » du programme Management Console dans le menu Démarrer qui démarre à la fois Management Console et RoboServer).
- En l'invoquant depuis la ligne de commande.
- En l'exécutant comme un service. Voir [Démarrage automatique des serveurs](#).

Pour invoquer un RoboServer à partir de la ligne de commande, ouvrez une fenêtre d'invite de commande, naviguez jusqu'au dossier `bin` dans le dossier d'installation `Kofax RPA` et saisissez :

```
RoboServer
```

Si tous les paramètres nécessaires sont spécifiés dans le fichier de configuration `roboserver.settings`, le RoboServer démarre.

Si l'un des paramètres manque à l'appel, le RoboServer produit une erreur et affiche l'aide et les paramètres disponibles.

Paramètres RoboServer

La ligne de commande pour le démarrage d'un RoboServer peut inclure les paramètres suivants :

```
RoboServer [-s <service:params>] [-mcUrl <url>] [-cl <Cluster Name>]
           [-b <url>] [-jmsNamespace <name>] [-p <port number>]
           [-sslPort <port number>] [-v]
```

Quelle que soit la façon dont vous démarrez un RoboServer, il accepte les paramètres du tableau suivant. Notez que vous pouvez modifier tous les paramètres dans l'utilitaire Paramètres de RoboServer. Voir [Configuration de RoboServer](#) pour plus d'informations.

Paramètre	Description
-mcUrl<arg>	<p>Indiquez à quel Management Console vous souhaitez vous inscrire dans le format suivant :</p> <pre>http[s]:// <nomutilisateur>:<motdepasse>@<nomdote>:<numéro de port></pre> <p>Exemple : -mcUrl http:// admin:password@myserver:8080/ManagementConsole</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Remarque Lors de l'utilisation d'une étape Document Transformation avec l'option Rappel dans un robot, utilisez le RoboServer nom d'hôte ou l'adresse IP dans le paramètre -mcUrl. N'utilisez pas 'localhost', car le service Document Transformation ne pourra pas atteindre la Management Console et le robot de rappel ne sera pas mis en file d'attente.</p> </div>
-cl --cluster <arg>	<p>Ce paramètre optionnel enregistre automatiquement un RoboServer avec le cluster spécifié sur la Management Console. Dans l'exemple suivant, le RoboServer s'inscrit au sein du cluster <i>Production</i>.</p> <p>Exemple : -cl Production</p> <p>Exemple : -mcUrl http:// administrateur:motdepasse@myserver:8080/ ManagementConsole -cl Production</p>
-eh -externalHost <numéro de port>	<p>Spécifie explicitement le nom ou l'adresse IP de l'hôte RoboServer. Le nom ou l'adresse IP et le numéro de port spécifiés pour l'hôte RoboServer ne sont pas valides lorsque vous travaillez en mode JMS.</p> <p>Ce paramètre doit être spécifié lorsque l'adresse de l'hôte est différente de ce qu'un RoboServer découvre sur la machine locale, comme lors d'une exécution avec NAT dans le cloud, ou lorsque vous exécutez le RoboServer dans un conteneur Docker.</p> <p>Exemple : -eh 10.10.0.123</p>
-ep -externalPort <numéro de port>	<p>Spécifie explicitement le numéro de port de l'hôte RoboServer. Le nom ou l'adresse IP et le numéro de port spécifiés pour l'hôte RoboServer ne sont pas valides lorsque vous travaillez en mode JMS.</p> <p>Ce paramètre doit être spécifié lorsque le port de l'hôte est différent de ce qu'un RoboServer découvre sur la machine locale, par exemple comme lors d'une exécution avec NAT dans le cloud, ou lorsque vous exécutez le RoboServer dans un conteneur Docker.</p>
-jmxPass	<p>Définit le mot de passe JMX si vous surveillez un RoboServer avec l'application JMX qui requiert un mot de passe.</p>
-v -verbeux	<p>Ce paramètre optionnel permet à un RoboServer d'afficher l'état et les événements d'exécution.</p>
-V -version	<p>Ce paramètre optionnel permet à RoboServer d'afficher le numéro de version, puis de sortir.</p>

Paramètre	Description
-h --help	Affiche l'aide.
-pause après une erreur de démarrage	Pause si une erreur s'est produite au démarrage.
-b -brokerUrl<url>	L'URL du répartiteur de messages (lorsque vous utilisez un service JMS).
-jmsNamespace <name>	Espace de noms pour les destinations JMS. La valeur par défaut est <code>Kapow</code> .
-jmsTTL	Spécifie la durée de vie des messages JMS en millisecondes au format <code>:<destination>:<tTL></code>
-s <service-name:service-parameter> -service <service-name:service-parameter>	Ce paramètre spécifie un service RQL ou JMX que RoboServer doit démarrer. Ce paramètre doit être spécifié au moins une fois, et peut être spécifié plusieurs fois pour lancer plusieurs services dans le même RoboServer. Les services disponibles dépendent de votre installation. Exemple : <code>-service socket:50000</code> Exemple : <code>-service jmx:50100</code> Voir « Services disponibles » dans le tableau ci-dessous pour plus d'informations.
-p -port <port-number>	C'est l'abréviation de <code>-s socket:<port-number></code> Exemple : <code>-port 50000</code>
-sslPort <numéro de port>	Il s'agit d'une abréviation pour écrire <code>-s ssl:<numéro de port></code>
-nd -NoDoc	Ce paramètre optionnel désactive les demandes de documentation du robot à ce RoboServer.
-sn -serverName	Il s'agit d'un paramètre facultatif qui définit le nom du serveur pour consigner les statistiques RoboServer, qui est ensuite affiché dans Kofax Analytics for RPA. Si vous ne spécifiez pas le nom du serveur, les statistiques sont collectées en fonction de l'adresse IP du serveur.
Services disponibles	
-service socket:<portNumber>	<portNumber> : Le numéro de port du service de socket sur lequel écouter.
-service ssl:<portNumber>	<portNumber> : Le numéro de port du service de socket sur lequel écouter.
-service jmx:<jmx_port_Number>,<jmx_rmi_url>	<jmx_port_Number> : Numéro de port pour le service JMX sur lequel écouter. <jmx_rmi_url>: Hôte et port RMI optionnels pour le service JMX. À utiliser si vous devez vous connecter à travers un pare-feu. Exemple : <code>-service jmx:example.com:51001</code>
-service jms:<number>	<number>: Un numéro qui identifie de manière unique le RoboServer sur cet hôte. Exemple : <code>-service jms:1</code>

Pour définir les mots de passe, utilisez soit l'utilitaire Paramètres de RoboServer, soit l'outil ConfigureRS. Pour obtenir plus d'informations, voir [Configuration de RoboServer](#) et [Configuration RoboServer – Mode sans tête](#).

Important À partir de Kofax RPA version 10, tous les RoboServers doivent s'enregistrer automatiquement sur le Management Console. Par conséquent, l'URL et les identifiants de la Management Console ainsi que le nom de cluster doivent être spécifiés au démarrage du RoboServer (soit sur la ligne de commande comme dans l'exemple suivant, soit en utilisant l'application [Paramètres de RoboServer](#) dans l'onglet Général, sous l'option S'inscrire sur une Management Console).

```
RoboServer.exe -mcUrl http://nomutilisateur:motdepasse@myserver:8080/  
ManagementConsole -cluster Production -service socket:50000
```

Le nom d'utilisateur et le mot de passe par défaut sont `admin:admin`.

Démarrage automatique des serveurs

Si votre installation comprend une fonctionnalité de serveur, vous pouvez la configurer pour qu'elle démarre automatiquement les serveurs.

Par « fonctionnalité du serveur », on entend RoboServer et Management Console (serveur de licences). En fait, ces deux fonctionnalités sont fournies par le même programme serveur, RoboServer, en fonction des arguments qui lui sont fournis au démarrage.

La section [Paramètres de RoboServer](#) contient une description détaillée des arguments de la ligne de commande du programme RoboServer. Pour permettre au programme RoboServer d'exécuter des robots, spécifiez l'argument `-service`. De même, l'argument `-MC` permet d'activer la fonctionnalité Management Console (voir Management Console (Serveur de licences) dans le guide d'installation).

Pour plus d'informations sur le démarrage du RoboServer et d'autres composants RPA en tant que services, voir [Exécuter les composants RPA en tant que services](#).

Fermer RoboServer

RoboServer peut être fermé à l'aide de l'outil en ligne de commande `ShutDownRoboServer`. Exécutez `ShutDownRoboServer` sans arguments pour voir les différentes options permettant d'arrêter le serveur, en particulier comment gérer les robots actuellement exécutés sur le serveur.

Configuration de la production

RoboServer fait fonctionner des robots créés avec Design Studio. Les robots peuvent être démarrés de différentes manières, en étant soit planifiés pour fonctionner à des moments précis par la Management Console, soit appelés via un service web REST, soit par les API Java ou .NET, soit encore à partir d'un Kaplet.

Pour obtenir un environnement de production stable et performant, vous devrez peut-être modifier certains des paramètres par défaut de RoboServer. Nous examinerons les options de configuration suivantes :

- Nombre d'instances de RoboServer
- Nombre de robots simultanés

- Allocation de mémoire
- Détection automatique de la surcharge de la mémoire

Nombre d'instances de RoboServer

RoboServer s'exécute sur la machine virtuelle Java (JVM) d'Oracle, qui à son tour s'exécute sur un système d'exploitation (OS), qui fonctionne sur votre matériel. Les JVM et les OS sont patchés, l'architecture matérielle change, et chaque nouvelle itération vise à apporter de meilleures performances. Bien que nous puissions donner quelques indications générales sur les performances, la seule façon de s'assurer que vous avez la configuration optimale est de la tester.

En règle générale, vous obtenez un peu plus de résultats en lançant deux instances de RoboServer. La JVM utilise une gestion de la mémoire connue sous le nom de « garbage collection » (GC). Sur la plupart des matériels, un seul cœur de CPU est actif pendant la GC, ce qui laisse 75 % du CPU inactif sur un CPU quadricœur. Si vous démarrez deux instances de RoboServer, l'une peut toujours utiliser la totalité de l'unité centrale tandis que l'autre fonctionne en mode GC. Cependant, veuillez noter que l'utilisation du CPU de garbage collector dépend de la spécification du JDK sur laquelle votre environnement fonctionne, de sorte que plusieurs cœurs de CPU peuvent être utilisés pendant le processus de GC.

Nombre de robots simultanés

Le nombre de robots simultanés qu'un RoboServer peut faire fonctionner dépend de la quantité de CPU disponible et de la vitesse à laquelle vous pouvez obtenir les données que RoboServer doit traiter. Le nombre de robots simultanés est configuré dans les paramètres du cluster de la Management Console. Un robot s'exécutant face à un site web lent utilisera beaucoup moins de CPU qu'un robot s'exécutant face à un site web avec un temps de réponse rapide, et voilà pourquoi. La quantité de CPU utilisée par un programme peut être décrite à l'aide de la formule suivante

$$\text{CPU (noyau)\%} = 1 - \text{Temps d'attente} / \text{Temps total}$$

Si un robot prend 20 secondes pour s'exécuter, mais que 15 secondes sont passées à attendre le site web, il n'exécute que 5 secondes, donc pendant les 20 secondes il utilise en moyenne 25 % (d'un noyau CPU). Les étapes d'un robot sont exécutées en séquence, ce qui signifie qu'un seul robot en exécution n'utilise qu'un seul noyau de CPU à la fois. La plupart des CPU modernes ont plusieurs cœurs, de sorte qu'un robot qui s'exécute en 20 secondes, mais qui attend 15 secondes, n'utilise en fait qu'environ 6 % d'un CPU quadricœur.

Par défaut, RoboServer est configuré pour faire fonctionner 20 robots simultanément. Le nombre de robots simultanés est configuré dans les paramètres du cluster Management Console. Si tous vos robots utilisent 6 % de l'unité centrale, celle-ci est pleinement utilisée lorsque vous faites fonctionner 16 à 17 robots en même temps. Si vous démarrez simultanément 33 de ces 6 % de robots, vous surchargez RoboServer; comme la quantité de CPU disponible est constante, le résultat est que chaque robot met deux fois plus de temps à terminer. Dans le monde réel, l'utilisation du processeur d'un robot peut représenter entre 5 et 95 % du noyau du processeur, selon la logique du robot et le site web avec lequel il interagit. Il est donc difficile de deviner ou de calculer la valeur correcte pour le nombre maximum de robots simultanés. La seule façon de s'assurer que vous avez la bonne valeur est de faire un test de charge et de surveiller l'utilisation du CPU de RoboServer, ainsi que le temps d'exécution du robot lorsque la charge augmente.

Allocation de mémoire

Un autre paramètre qui peut affecter le nombre de robots simultanés que chaque RoboServer peut gérer est la quantité de mémoire. La quantité de mémoire utilisée par les robots peut varier de quelques mégaoctets (Mo) à des centaines de Mo. Par défaut, RoboServer est configuré pour utiliser 2048 Mo sur un système 64 bits. Consultez le site [Modifier l'allocation de la RAM](#) pour voir comment contrôler l'allocation de la mémoire. Pour éviter une erreur de mémoire insuffisante, fournissez suffisamment de mémoire à un RoboServer. Pour garantir une allocation correcte de la mémoire, surveillez l'utilisation

de la mémoire pendant vos tests de charge. La JVM n'alloue pas toute la mémoire disponible, mais elle la réserve depuis le système d'exploitation. Une fois que la JVM commence à utiliser la mémoire, elle n'est pas restituée au système d'exploitation. Pour trouver l'allocation optimale de mémoire, effectuez une série de tests de charge qui poussent le CPU à 100 %. Après chaque test, vérifiez quelle quantité de la mémoire réservée a été effectivement utilisée par la JVM (le processus java.exe). Si les 2048 Mo (par défaut) ont tous été utilisés, augmentez la mémoire (en attribuant généralement le double) et refaites le test. À un moment donné, la JVM n'utilise pas toute la mémoire réservée, le numéro de la mémoire utilisée reflète le besoin réel de mémoire et doit être spécifié pour le RoboServer.

Détection de la surcharge de la mémoire

Comme RoboServer tombe en panne en cas de mémoire insuffisante, RoboServer essaie d'empêcher une surcharge de mémoire. Avant qu'un RoboServer ne démarre un nouveau robot, il vérifie l'utilisation de la mémoire. Si l'utilisation de la mémoire est supérieure à 80 %, le robot est mis en file d'attente au lieu de démarrer ; cela réduit considérablement le risque de tomber en panne RoboServer si l'allocation mémoire est mal configurée. Ce mécanisme est souvent appelé le seuil de mémoire de 80 %. Une fois que l'utilisation de la mémoire tombe en dessous de 80 %, le RoboServer continue de démarrer de nouveaux robots. Lorsqu'un RoboServer arrête de lancer de nouveaux robots, il enregistre la ligne suivante dans le journal :

```
« RoboServer utilise » + « memoryUsagePercentage + » pourcentage de la mémoire disponible et mettra en file d'attente toutes les nouvelles exécutions de robot.
```

Cela se produit généralement si vous exécutez trop de robots simultanés sur le serveur.

Lorsqu'un robot démarre après que l'utilisation de la mémoire est tombée en dessous de 80 %, un RoboServer consigne le message suivant :

```
robotLogger.logServerInfo("Utilisation normale de la mémoire", "L'utilisation de la mémoire est à nouveau inférieure au seuil et les exécutions du robot ne seront plus mises en file d'attente.")
```

Important Kofax ne recommande pas de modifier le paramètre `Kapow.memoryThreshold`, car un RoboServer peut devenir instable. Avant de le modifier, essayez ce qui suit :

- Diminuez le nombre de robots simultanés (voir « Nombre de robots simultanés » ci-dessus).
- Augmentez la mémoire disponible (voir « Allocation mémoire » ci-dessus).

Ne modifiez le paramètre `Kapow.memoryThreshold` que s'il est requis. Par exemple, si RoboServer est à 79 % de l'utilisation de la mémoire et que l'un de vos robots utilise 22 % ou plus de la mémoire allouée, le RoboServer tombera en panne en essayant d'exécuter un tel robot. Dans ce cas, vous pouvez diminuer la valeur du seuil à 70 % pour éviter les erreurs. La valeur du seuil est configurée dans la propriété `Kapow.memoryThreshold` du fichier `roboserver.conf`. Pour modifier la valeur à 70 %, ouvrez le fichier `roboserver.conf` dans un éditeur de texte, insérez la ligne suivante et redémarrez le RoboServer.

```
wrapper.java.additional.1=-Dkapow.memoryThreshold=70
```

Le nombre après `wrapper.java.additional` est un compteur, donc ajustez-le si vous avez d'autres paramètres supplémentaires dans le fichier.

Configuration de RoboServer

Vous pouvez configurer RoboServer via l'application de paramétrage RoboServer. Les paramètres RoboServer peuvent être lancés à partir du menu Démarrer de Windows.

The screenshot shows the 'Serveur JMX' configuration window. It includes the following fields and options:

- Activer le service connecteur:** (unchecked)
- Port:** 50000
- Activer le service connecteur SSL:** (unchecked)
- Port:** 50001
- Server name:** [Empty text field]
- Activer le service JMS:** (unchecked)
- URL du serveur de messagerie:** [Empty text field]
- ID du RoboServer pour JMS:** 1
- Nom d'utilisateur:** [Empty text field]
- Mot de passe:** [Empty text field]
- Utiliser SSL sur JMS:** (unchecked)
- Emplacement du magasin de clés:** [Empty text field] with a 'Parcourir...' button
- Mot de passe du magasin de clés:** [Empty text field]
- Emplacement du magasin de confiance:** [Empty text field] with a 'Parcourir...' button
- Mot de passe du magasin de confiance:** [Empty text field]
- Enregistrer sur une Management Console:** (unchecked)
- URL de la Management Console:** [Empty text field]
- Nom d'utilisateur:** [Empty text field]
- Mot de passe:** [Empty text field]
- Cluster:** [Empty text field]
- Hôte externe:** [Empty text field]
- Port externe:** 0
- Rejeter la demande de documentation:** (unchecked)
- Prolix:** (unchecked)

Buttons at the bottom: 'Aide', 'OK', and 'Annul...'.

Fenêtre principale des paramètres

Cette application vous permet de configurer les éléments suivants :

- **Général** : Options du service de socket, activer et configurer les options **Service JMS**, les options de connexion Management Console, y compris le nom et le mot de passe du super-utilisateur `admin`, les paramètres de l'hôte RoboServer et l'option Verbose.
- **Sécurité** : Les **paramètres de sécurité** tels que l'authentification et les autorisations.
- **Certificats** : L'utilisation de **certificats**.

- **Projet** : L'emplacement du [projet par défaut](#).
- **Serveur JMX** : [Configuration du serveur JMX](#).
- **Management Console** : [configuration de la Management Console intégrée](#).

Après avoir modifié l'un des paramètres, cliquez sur OK pour enregistrer les nouveaux paramètres, puis redémarrez les RoboServers pour que les changements prennent effet.

À partir de Kofax RPA version 10, tous les RoboServers doivent s'enregistrer automatiquement sur le Management Console. Par conséquent, l'URL et les identifiants de Management Console ainsi que le nom de cluster doivent être spécifiés au démarrage de RoboServer (soit en [ligne de commande](#), soit en utilisant l'application **Paramètres de RoboServer**).

Le nom ou l'adresse IP et le numéro de port de l'hôte RoboServer doivent être spécifiés lorsque ces paramètres sont différents de ce qu'un RoboServer découvre sur la machine locale, comme lorsque vous exécutez du NAT dans le cloud, ou lorsque vous exécutez le RoboServer dans un conteneur Docker.

Kofax RPA contient plusieurs outils en ligne de commande pour vous aider à modifier les paramètres en mode batch. Par exemple, vous pouvez créer plusieurs utilisateurs avec des autorisations spécifiques. Voir [Configuration RoboServer – Mode sans tête](#) pour plus d'informations.

Si vous devez modifier la quantité maximale de RAM que RoboServer peut utiliser, voir [Modifier l'allocation de la RAM](#).

Démarrer et entrer la licence dans le Management Console intégré

Avant de pouvoir saisir les informations relatives à la licence sur Management Console, vous devez le lancer. Si vous utilisez un Management Console intégré, commencez comme suit. Consulter [Déploiement de Tomcat](#) pour obtenir des informations sur Tomcat Management Console.

Avant de lancer un Management Console, procédez comme suit :

1. Lancez l'application [Paramètres du RoboServer](#) dans le menu Démarrer.
2. Dans l'onglet **Général**, sélectionnez **Enregistrer sur une Management Console**, et fournissez toutes les informations nécessaires, y compris le nom du super-utilisateur et le mot de passe de l'administrateur, pour vous connecter à l'adresse Management Console. Le nom et le mot de passe par défaut du super-utilisateur admin :
 - Nom d'utilisateur : `admin`
 - Mot de passe : `admin`

Windows

Utilisez l'élément **Démarrer la Management Console** dans le menu Démarrer.

Pour lancer le Management Console à partir de la ligne de commande, exécutez la commande suivante dans le sous-dossier bin du dossier d'installation.

```
RoboServer.exe -p 50000 -MC
```

Vous pouvez également utiliser la ligne de commande pour lancer un RoboServer et l'enregistrer sur un Management Console :

```
RoboServer.exe -p 50000 -mcUrl http://username:password@ServerName:port -  
cl la commande « Production » lance un RoboServer sur le port 50000 et l'enregistre dans le
```

Management Console sur `ServerName:port` sous le cluster de `Production` avec le nom d'utilisateur et le mot de passe spécifiés.

Linux

Démarrez Management Console à partir de la ligne de commande. Il fait partie de la planification RoboServer qui se trouve dans le répertoire `bin`, sous le répertoire d'installation.

```
$. /RoboServer -p 50000 -MC
```

Démarrage automatique

Comme alternative, si vous configurez plus tard le démarrage automatique du Management Console comme décrit dans [Démarrez RoboServer](#), vous pouvez choisir de le faire maintenant au lieu de lancer Management Console manuellement.

Une fois que le site Management Console est lancé, ouvrez-le dans un navigateur. Sous Windows, cliquez sur l'élément Management Console dans le menu Démarrer. Sur toutes les plateformes, vous pouvez ouvrir un navigateur et vous rendre sur `http://localhost:50080/`. Connectez-vous au site Management Console en utilisant les identifiants par défaut de l'utilisateur `admin`, acceptez les termes de la licence et saisissez vos informations de licence, y compris vos clés de licence. Si vous avez besoin de modifier les informations de la licence plus tard, vous pouvez le faire dans **Administration > Licence**.

Configuration Management Console intégrée

Les paramètres sont disponibles dans l'onglet « **Management Console** » de l'application « Paramètres du RoboServer ».

RoboServer contient un serveur web intégré qui gère le Management Console. Le serveur web fait partie de RoboServer, mais il n'est activé que lorsqu'un RoboServer est lancé avec l'option `-MC`. Par défaut, le serveur web interagit avec le port 50080, et l'interface web Management Console est donc disponible sur :

```
http://host:50080/
```

Protocoles et ports

Vous pouvez configurer le serveur web pour qu'il soit accessible par HTTP et HTTPS sur des ports séparés. Si un protocole est activé, un numéro de port doit être choisi ; les valeurs par défaut sont le port 50080 (HTTP) et le port 50443 (HTTPS).

Pour activer le HTTPS, un certificat de serveur au format JKS doit être stocké dans un fichier appelé `webserver.keystore` dans le dossier `Certificates/web` de l'installation. Si un mot de passe de certificat autre que le mot de passe par défaut (*changeit*) doit être utilisé, saisissez-le dans le champ Mot de passe du certificat.

Vous pouvez également limiter les personnes autorisées à télécharger le pilote JDBC sur le site Management Console (pour plus d'informations, voir « Pilotes de base de données » dans *l'aide du site Kofax RPA*). Les choix possibles sont « **Non autorisé** », où personne ne peut télécharger les pilotes JDBC, « Administration à partir de l'hôte local », ce qui signifie que l'utilisateur `admin` peut télécharger les pilotes lorsqu'il accède à Management Console depuis la machine locale ; et enfin, « Administration à partir d'un hôte », ce qui signifie que l'utilisateur `admin` peut toujours télécharger les pilotes JDBC.

Gestion des utilisateurs

Management Console est accessible non seulement à partir du même ordinateur (localhost), mais aussi à partir d'autres ordinateurs. L'un des avantages d'une Management Console est qu'elle coordonne l'exécution des robots, et doit donc être accessible à de nombreux clients.

Afin d'atténuer le risque potentiel de sécurité lié à l'accès au Management Console à partir d'autres ordinateurs, la gestion des utilisateurs est activée par défaut en mode intégré et le mot de passe par défaut du super-utilisateur `admin` est disponible (nom d'utilisateur - `admin`, mot de passe - `admin`). Vous devez utiliser ces identifiants pour enregistrer un RoboServer sur un Management Console, lorsque vous publiez un robot sur le Management Console à partir du Design Studio, et lorsque vous accédez à l'interface web à partir d'un navigateur. Voir [Rôles prédéfinis des utilisateurs](#) pour plus d'informations.

Sécurité

Dans l'onglet RoboServer paramètres de **Sécurité**, vous indiquez la configuration TLS de RoboServer, les restrictions générales de sécurité, si une authentification est nécessaire pour accéder à RoboServer et les préférences de journalisation des audits.

Autoriser l'accès au système de fichiers et à la ligne de commande

Permet à RoboServer de créer et de modifier des fichiers sur l'ordinateur où tourne RoboServer.

Important Lorsqu'ils utilisent la base de données intégrée de Derby, les robots peuvent créer et modifier des fichiers sur les ordinateurs lorsque cette option n'est pas sélectionnée. Nous vous recommandons d'utiliser MySQL ou une autre base de données d'entreprise dans votre environnement réseau.

Autoriser l'utilisation de connecteurs

Lors d'une exécution sur RoboServer, ce paramètre permet l'utilisation de connecteurs personnalisés dans les robots de l'ordinateur sur lequel s'exécute RoboServer. Utilisez des connecteurs personnalisés dans l'étape Activité personnalisée dans les robots. Pour plus de détails, consultez *l'aide du site Kofax RPA*.

Accepter les pilotes JDBC de la Management Console

Distribue les pilotes JDBC du site Management Console au site RoboServer.

Délai dépassé pour la commande

Indique combien de temps le RoboServer doit attendre une réponse à une commande sur un dispositif distant. Cette option ne s'applique qu'à l'automatisation des terminaux et à la navigation sur les sites web à l'aide de robots.

Une commande est une instruction envoyée à un dispositif d'automatisation, comme par exemple cliquer sur un bouton de la souris, ouvrir une application, ajouter un garde de localisation trouvé, etc. Si une commande ne peut être exécutée dans un délai déterminé, le service envoie une notification et l'exécution du robot s'arrête.

Notez que dans le cas d'une étape Choix contrôlé, ce paramètre s'applique à l'appel du garde dans le workflow, mais attendre que le garde soit satisfait n'est pas lié à ce délai, et l'attente peut être indéfinie. Une situation similaire se produit lors de l'utilisation des étapes Déplacer la souris et Extraire. Les

commandes doivent être invoquées sur le dispositif dans le délai d'attente spécifié dans ce champ, mais le robot attend jusqu'à 240 secondes pour les exécuter.

Configuration TLS RoboServer

Kofax RPA fournit les moyens d'établir une communication TLS entre le dispositif d'automatisation et RoboServer. La communication utilise des certificats pour chiffrer la communication. Le chiffrement utilise une structure de clé publique - privée pour sécuriser la connexion.

Dans les **Paramètres de configuration TLS** de l'onglet **Sécurité**, vous pouvez spécifier si vous voulez utiliser l'ensemble de certificats intégré ou en spécifier un autre.

- Pour utiliser les certificats Kofax RPA, sélectionnez **Utiliser les valeurs par défaut**
- Pour utiliser d'autres certificats, décochez la case **Utiliser les valeurs par défaut** et indiquez les chemins d'accès suivants aux clés privées et publiques ainsi qu'au dossier Certificats de confiance dans les champs correspondants.

Voir « Utilisation des communications TLS » dans l'*aide de Kofax RPA* pour plus d'informations.

Restrictions

Vous pouvez préciser si le RoboServer est autorisé à accéder au système de fichiers et à la ligne de commande. Par défaut, cela n'est pas autorisé. Si vous l'activez, cependant, les robots qui tournent sur RoboServer sont autorisés à accéder au système de fichiers et, en utilisant l'étape Exécuter la ligne de commande, à exécuter des commandes arbitraires sur la machine qui exécute un RoboServer.

Remarque Permettre l'accès au système de fichiers et à la ligne de commande représente un risque pour la sécurité, et vous devriez examiner attentivement si cela est nécessaire. Si vous le faites, vous devez vous assurer que la machine n'est pas accessible depuis l'extérieur du réseau local, et/ou vous devez exiger une authentification de l'utilisateur. Le fait d'avoir un RoboServer avec un système de fichiers et un accès en ligne de commande fonctionnant sur une machine accessible depuis l'Internet et ne nécessitant pas d'authentification, ouvre la machine à l'extérieur, et chacun peut modifier le système de fichiers en fonction des droits d'accès de l'utilisateur exécutant RoboServer.

Vous pouvez également désactiver l'acceptation des pilotes JDBC à partir du Management Console. En activant les RoboServers, le Management Console leur envoie également des paramètres. Par défaut, cela inclut tous les pilotes JDBC qui ont été téléchargés sur le site Management Console. Si un utilisateur malveillant a obtenu un accès administrateur au site Management Console, il peut charger des fichiers JAR tout aussi malveillants qui seront ensuite envoyés aux RoboServers. Si l'utilisateur admin Management Console n'est autorisé à télécharger des pilotes JDBC qu'à partir de l'hôte local, ce qui précède ne se produira que si l'attaquant est en fait assis devant la machine qui exécute le Management Console ou s'il a obtenu l'accès à un VPN, par exemple (auquel cas vous aurez probablement des problèmes plus importants). En général, il ne devrait donc pas être nécessaire de désactiver l'acceptation des conducteurs JDBC. Si vous le faites, vous pouvez mettre les pilotes JDBC à la disposition du RoboServer en les plaçant manuellement dans le répertoire `lib/jdbc` sous Linux et `lib` sous Windows dans le dossier d'installation Kofax RPA.

Requête d'authentification

Pour protéger votre RoboServer contre tout accès non autorisé, vous pouvez activer l'authentification. Cela s'applique à tous les RoboServers exécutés à partir de votre installation Kofax RPA, y compris un RoboServer lancé en tant que service ou à partir d'une ligne de commande.

Pour activer l'authentification, cochez la case Demander RoboServer Authentification. Pour faire fonctionner des robots sur un RoboServer dont l'authentification est activée, vous devez ajouter des utilisateurs en cliquant sur le bouton d'ajout. Cela permettra d'insérer un nouvel utilisateur anonyme. Vous pouvez ensuite remplir les informations sur l'utilisateur, y compris le nom d'utilisateur, qui sera alors affiché dans la liste des utilisateurs.

Un utilisateur est configuré en utilisant les propriétés du tableau suivant.

Propriétés de l'utilisateur

Propriété	Description
Nom d'utilisateur	Il s'agit du nom d'utilisateur utilisé pour accéder au RoboServer.
Mot de passe	Il s'agit du mot de passe utilisé pour accéder au RoboServer.
Commentaires	Ici, vous pouvez écrire un commentaire sur l'utilisateur.
Démarrer le robot	Cela permet à l'utilisateur de lancer des robots sur le RoboServer.
Arrêter le robot	Cela permet à l'utilisateur d'arrêter les robots sur le RoboServer.
Fermeture de RoboServer	Cela permet à l'utilisateur de fermer le RoboServer à partir du Management Console.

Pour faire fonctionner des robots sur un RoboServer configuré avec une autorisation, l'appelant doit fournir les identifiants appropriés. Dans les RoboServers, cela se fait dans les paramètres. Lorsque vous utilisez un robot via l'API Java, les identifiants sont fournis dans **Management Console > Paramètres > Général > Authentification RoboServer**.

Configurer la Journalisation RoboServer

Pour que chaque requête HTTP et FTP effectuée par RoboServer soit automatiquement enregistrée, sélectionnez l'option Journaliser le trafic HTTP/FTP. Cela permettra d'enregistrer les requêtes HTTP et FTP en utilisant le logger Log4J2 spécifié par logger.auditlog. Log4J2 est configuré par le fichier log4j2.properties dans le dossier Configuration du dossier des données d'application.

Le journal d'audit comprend toutes les requêtes adressées aux deux pages web et à toutes ses ressources. Le journal contient les informations suivantes :

- Horodatage
- Niveau du journal (INFO)
- Enregistreur

- Nom du robot
- Nom d'hôte
- URL de la requête
- Code de réponse HTTP/FTP
- Nombre d'octets chargés à partir du corps de réponse

Certificats

Un problème clé dans l'établissement d'une communication sécurisée est de s'assurer que vous communiquez avec la bonne partie. Il ne suffit pas de demander des informations d'identité à l'autre partie, il doit également y avoir un moyen de vérifier ces informations avant de pouvoir s'y fier. Les certificats apportent une solution à ce problème, car ils contiennent à la fois les informations d'identité d'une partie (le nom d'hôte et la clé publique de cette partie) et la signature d'un tiers de confiance qui garantit l'exactitude des informations d'identité. Un certificat doit être approuvé uniquement :

- Si le nom d'hôte indiqué dans le certificat correspond au nom d'hôte du site dont il provient (sinon, il s'agit d'un enregistrement de l'identité de quelqu'un d'autre, ce qui revient à utiliser de faux identifiants).
- Et si le certificat est signé par une tierce partie en qui vous avez confiance.
- En outre, la date d'expiration du certificat et autres éléments similaires doivent être vérifiés, mais nous ne discutons pas ces détails.

Le processus de signature est fondé sur la cryptographie à clé publique/privée. La signature d'un certificat est le résultat (assez compact) d'un calcul complexe qui implique à la fois le contenu du certificat et la clé privée du signataire, qui ne peut être reproduite sans cette même clé privée. La signature peut être vérifiée en faisant un autre calcul qui implique le contenu du certificat, la signature et la clé publique du signataire. Ce calcul permet de savoir si le certificat correspond à la signature et est donc authentique. Veuillez remarquer que la clé publique vous permet uniquement de vérifier une signature, et non d'en générer une. Donc, la clé publique ne permet à personne de falsifier un certificat.

Le signataire ne doit jamais donner la clé privée à qui que ce soit, mais il peut distribuer la clé publique le plus largement possible. Toutefois, il reste encore un problème à résoudre avant de pouvoir faire confiance à une signature : Vous devez vous assurer que vous disposez d'une copie authentique de la clé publique du signataire. Les clés publiques d'autorités de signature bien connues comme VeriSign sont distribuées avec chaque navigateur et Java Runtime, et votre confiance envers la signature (et donc envers le certificat) repose en fait sur votre confiance du mode d'installation du navigateur ou le Java Runtime.

Il est possible de créer votre propre autorité de signature en créant une paire de clés publique/privée et en distribuant la clé publique. Cela se fait en intégrant la clé publique dans un certificat (appelé certificat auto-signé). Bien sûr, une partie qui reçoit un tel certificat ne lui fait pas confiance en fonction de sa signature, mais parce qu'il vous fait confiance et il se fie au mode de communication de ce certificat.

Afin de rendre ce schéma plus flexible dans la mise en œuvre réelle, il est possible de déléguer le pouvoir de signer. En d'autres termes, la signature d'un certificat peut ne pas provenir d'une tierce partie à laquelle vous faites confiance, mais plutôt d'une autre partie qui peut afficher un certificat auquel vous pouvez faire confiance. Cela peut être étendu à n'importe quel nombre de niveaux, représentant une chaîne de confiance. Pour que la vérification soit pratique, le certificat signé contient des copies de tous les

certificats associés à la chaîne de confiance (le dernier étant un certificat racine auto-signé). Au moins un des certificats de cette chaîne doit avoir été préalablement validé par vous.

Les certificats sont utilisés de quatre manières différentes sur RoboServer, ce qui correspond aux quatre propriétés de l'onglet « Certificats ». Deux d'entre eux concernent la manière dont les robots accèdent aux serveurs web dans le cadre de leur exécution.

Vérifier les certificats HTTPS

Un robot peut avoir besoin de vérifier l'identité d'un serveur web auquel il accède (via HTTPS). Une telle vérification est effectuée régulièrement (et de manière invisible) par les navigateurs ordinaires afin de détecter les attaques de phishing. Toutefois, la vérification n'est souvent pas nécessaire lorsque les robots collectent des informations, car ils n'accèdent qu'aux sites web bien connus pour lesquels ils ont été conçus. Ainsi, la vérification n'est pas activée par défaut.

Si elle est activée, la vérification se fait de la même manière qu'un navigateur le ferait, c'est-à-dire que le certificat du serveur web est vérifié en fonction de l'ensemble des certificats HTTPS de confiance installés, semblables à ceux que vous pouvez configurer dans un navigateur.

Certificats client HTTPS

C'est presque la même chose que ce qui est décrit ci-dessus, mais dans la direction opposée. Un robot peut avoir besoin d'accéder à un serveur web qui veut vérifier l'identité du client (robot) qui y accède. Le serveur web contient vraisemblablement des données confidentielles ou commerciales qui ne doivent être transmises qu'à des clients dont l'identité est prouvée. Cette identité est représentée par un certificat client HTTPS.

Deux autres propriétés ont trait à l'authentification dans la communication entre RoboServer et les clients API qui veulent que les robots soient exécutés sur RoboServer. Ces propriétés s'appliquent uniquement lorsque les clients se connectent à RoboServer via le protocole RQL basé sur un socket sécurisé. Le but principal du protocole sécurisé est de chiffrer la communication, mais avec une légère configuration, il permettra également l'authentification, c'est-à-dire la vérification de l'identité :

Vérifier les certificats client de l'API

RoboServer est en mesure de vérifier l'identité de tout client qui s'y connecte afin d'exécuter des robots. Cette vérification est désactivée par défaut.

Si elle est activée, la mécanique de vérification est la même que pour les certificats HTTPS, même si l'objectif est tout à fait différent. Le certificat du client qui se connecte est vérifié sur la base d'un ensemble installé de certificats client de l'API de confiance.

Certificat du serveur API

RoboServer dispose également d'un certificat de serveur qu'il présentera aux clients qui se connectent. Ce certificat a deux objectifs : Il fait fonctionner le côté chiffrement du protocole SSL (c'est pourquoi RoboServer est livré avec un certificat auto-signé par défaut), et il identifie ce RoboServer particulier aux clients.

Le certificat par défaut du serveur API est le même pour tous les RoboServers et n'est donc pas approprié pour l'identification. Si vos clients ont besoin de vérifier l'identité du RoboServer auquel ils se connectent, comme décrit dans le protocole SSL, vous devez créer et installer un certificat du serveur API unique pour chaque RoboServer.

Installer des certificats HTTPS

Lorsqu'un robot accède à un site web via HTTPS, il vérifie le certificat du site (si la case « Vérifier les certificats HTTPS » est cochée). La vérification est effectuée sur la base de deux ensembles de certificats de confiance : l'ensemble des certificats racine et un ensemble supplémentaire de certificats de serveur.

Les certificats racine sont installés avec Kofax RPA tout comme les certificats racine sont installés avec votre navigateur. Ils se trouvent dans le dossier Certificats/Root dans le dossier Application Data.

Certains sites HTTPS peuvent utiliser des autorités de certification qui ne sont pas incluses par défaut. Dans ce cas, vous devez installer les certificats appropriés pour que Kofax RPA puisse être chargé à partir de ces sites. Le plus souvent, ils sont installés dans le dossier Certificates/Server du dossier Application Data.

Pour le traitement des sites HTTPS, il importe peu que vous ajoutiez des certificats à l'ensemble des certificats racine ou à l'ensemble des certificats de serveur. Toutefois, il faut noter que les certificats racine ont une portée plus large, car ils sont également utilisés lors de la vérification des certificats client de l'API.

Pour installer un certificat, obtenez le certificat sous forme de chaîne de certificats PKCS#7, de chaîne de certificats Netscape ou de certificat codé DER. Vous installez le certificat en le copiant dans l'un des deux dossiers mentionnés ci-dessus. Le nom du fichier contenant le certificat n'a pas d'importance.

L'exemple suivant explique comment installer un certificat de serveur pour le site web <https://www.foo.com>. L'exemple est basé sur Internet Explorer.

Remarque Vous devez installer les certificats sur toutes les installations qui doivent être chargées à partir des sites HTTPS particuliers.


1. Ouvrez <https://www.foo.com/> dans Internet Explorer.
2. Sélectionnez **Tools > Internet Options > Content > Certificates**.
La fenêtre **Certificates** apparaît.
3. Dans l'onglet **Personnel**, cliquez sur **Import**.
4. Cliquez sur **Next** et accédez au fichier contenant le certificat.
5. Sélectionnez le fichier de certificat et cliquez sur **Suivant > Terminer**. Quittez la fenêtre du certificat.
Vous avez maintenant installé le certificat dans votre navigateur. L'étape suivante consiste à l'exporter dans un fichier.
6. Revenez à la fenêtre **Certificates**, recherchez le certificat installé et cliquez sur **Export**.
7. Cliquez deux fois sur **Next**.
8. Dans le dossier des données de la demande, enregistrez le certificat sous le nom `foo.cer` dans le sous-dossier `Certificates/Server`.
9. Redémarrez tous vos RoboServers en cours d'exécution.

Installer des certificats de client HTTPS

Lorsqu'un robot accède à un site HTTPS, il peut avoir besoin de fournir son propre certificat au serveur web. Cette option est configurée dans les options par défaut pour un robot ou dans les options

spécifiques à une étape. Une façon de fournir le certificat est de faire référence à l'un de ceux qui ont été configurés dans l'installation Kofax RPA.

Remarque Les certificats client HTTPS doivent être configurés dans toutes les installations Kofax RPA qui doivent faire fonctionner le robot.

1. Dans l'onglet **Certificats** de la fenêtre d'application Paramètres RoboServer, cliquez sur le bouton plus (+) sous la liste des certificats de clients HTTPS.
2. Lorsque vous y êtes invité, sélectionnez un fichier de certificat, qui doit être au format PKCS12.
3. Saisissez le mot de passe utilisé pour chiffrer le fichier de certificat.
4. Vous pouvez éventuellement modifier l'ID unique attribué au certificat en cliquant sur le bouton  Modifier.
L'ID est utilisée plus tard dans le robot pour sélectionner le certificat.

Installer des certificats de client API

Lorsque les clients API se connectent à RoboServer via SSL, RoboServer il vérifie les certificats qui lui sont présentés (si la case « Vérifier les certificats client de l'API » est cochée). La vérification signifie que RoboServer rejette les connexions des clients qui échouent à la vérification, et la vérification est effectuée sur la base de deux séries de certificats de confiance : L'ensemble des certificats racine et un ensemble supplémentaire de certificats client API.

Les certificats racine sont installés avec Kofax RPA tout comme les certificats racine sont installés avec votre navigateur. Ils se trouvent dans le dossier Certificats/Root dans le dossier Application Data. Ce sont les mêmes certificats racine qui sont utilisés pour vérifier les certificats HTTPS ; cependant, les certificats racine joueront probablement un rôle beaucoup plus restreint lors de la vérification des clients API.

En effet, dans la plupart des cas, vous créez vos propres certificats client de l'API auto-signés plutôt que d'utiliser des certificats délivrés par des autorités de signature officielles. Vous devez installer vos certificats client de l'API dans le dossier Certificates/API/TrustedClients du dossier des données d'application afin que RoboServer puisse les reconnaître.

Pour vérifier la connexion des clients API, il importe peu que vous ajoutiez des certificats clients API à l'ensemble des certificats racine ou à l'ensemble des certificats client de l'API. Toutefois, les directives données ici vous aident à éviter les problèmes causés par le fait que les certificats racine sont également (voire principalement) utilisés lors de la vérification des certificats HTTPS.

Vous pouvez générer un certificat auto-signé pour votre client API avec la commande Java keytool comme suit :

```
keytool -genkey -keystore client.p12 -alias client -keyalg RSA -storetype "PKCS12"
```

Vous êtes invité à fournir les informations suivantes : Nom (domaine), nom de l'unité organisationnelle, organisation, ville, état, pays et mot de passe. N'oubliez pas le mot de passe, il n'y a aucun moyen de le récupérer en cas de perte. Cet appel de keytool place le certificat dans le magasin de clés client.p12. Vous devez ensuite l'extraire dans un fichier séparé :

```
keytool -export -keystore client.p12 -alias client -storetype "PKCS12" -file client.pub.cer
```

Vous êtes invité à indiquer le mot de passe utilisé lors de la génération du certificat. Le fichier de sortie `client.pub.cer` est ce qui doit être copié dans le dossier `Certificates/API/TrustedClients` dans le dossier des données d'application.

Installer un certificat du serveur API

Pour des raisons techniques, RoboServer doit disposer d'un certificat de serveur qu'il peut présenter aux clients de l'API lorsqu'ils s'y connectent en utilisant le protocole SSL. Lors de l'installation, un certificat auto-signé par défaut est installé. Ce certificat n'est pas valide pour l'identification car il est le même pour tous les RoboServers et doit être remplacé si les clients de l'API doivent vérifier l'identité d'un RoboServer.

1. Dans l'onglet **Paramètres du RoboServer Certificats**, cliquez sur **Changer**.

2. Dans la fenêtre de sélection des fichiers, sélectionnez le certificat.

3. Lorsque vous y êtes invité, saisissez le mot de passe du certificat.

Le certificat est importé et les propriétés suivantes apparaissent : Délivré à, délivré par, et expire.

4. Utilisez la commande Java keytool suivante pour générer un nouveau certificat auto-signé pour RoboServer :

```
keytool -genkey -keystore server.p12 -alias server -keyalg RSA -storetype "PKCS12"
```

5. Lorsque vous y êtes invité, saisissez les informations suivantes :

- Unité organisationnelle
- Organisation
- Ville
- Statut
- Pays
- mot de passe

Remarque Veillez à noter le mot de passe, car il ne peut être récupéré en cas de perte.

Le certificat est enregistré dans le fichier `server.p12` du magasin de clés, qui peut être importé.

Configuration RoboServer – Mode sans tête

Kofax RPA est livré avec plusieurs utilitaires permettant de configurer votre RoboServer à partir d'une ligne de commande. Les utilitaires se trouvent dans le sous-dossier `bin` du dossier d'installation de Kofax RPA. Notez que les fichiers de configuration dépendent de l'utilisateur et sont stockés dans le dossier de l'utilisateur. Pour plus d'informations, voir la rubrique « Dossiers importants » du Guide d'installation de Kofax RPA.

- `ConfigureRS` : définit le mot de passe JMX et le mot de passe Management Console dans le fichier de configuration RoboServer (`roboserver.settings`).
- `ConfigureMC` : définit les protocoles et l'utilisation des ports, les mots de passe des certificats et les autorisations de téléchargement des fichiers jar JDBC dans le fichier `mc.settings`.
- `ConfigureRSUser` : Ajoute et supprime des utilisateurs, puis met à jour les identifiants d'utilisateur dans le fichier `rsusers.xml`. Les informations contenues dans ce fichier sont utilisées pour authentifier les requêtes API.

Pour obtenir de l'aide sur l'utilisation, exécutez les utilitaires avec l'option `-h`.

Pour établir une connexion avec le site Management Console sur lequel un RoboServer s'enregistre, saisissez la commande suivante :

```
ConfigureRS -mcUrl http://admin:password@localhost:8080/ManagementConsole
```

Remarque Les identifiants par défaut du super-utilisateur `admin` sont les suivants : nom d'utilisateur – `admin`, mot de passe – `admin`.

Pour créer un utilisateur1 avec le mot de passe `Password1` et toutes les autorisations, saisissez la commande suivante :

```
ConfigureRSUser utilisateur1 Motdepasse1 -a
```

Pour activer l'authentification des requêtes API, ouvrez `rsusers.xml` et modifiez la configuration `utilisateur` activée sur `vrai`, comme le montre l'exemple suivant.

Exemple de fichier de configuration `rsusers.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<userConfiguration enabled="true">
  <users>
    <user username="user1"
password_hash="20c7628c31534b8718a1da00435505e4262e3f4dc305">
      <startRobot/>
      <stopRobot/>
      <shutdownRoboServer/>
    </user>
  </users>
</userConfiguration>
```

Exemple de fichier de configuration `roboserver.settings`

```
# Settings file for RoboServer. Some configurations contains encrypted passwords and
should
not be edited by hand, these should be modified using dedicated commandline tools.

# The directory of use on RoboServer when the API used the DefaultRoboLibrary. On
windows \ must be escaped like this
c:\\\\users\\AppData\\Local\\Kofax RPA\\...
defaultProject = /home/TestUser/Kofax RPA/trunk

# Should RoboServer be allowed to access the fileSystem, or call commands/scripts.
Values: true/false
sec_allow_file_system_access = false

# Will RoboServer accept JDBC drivers sent from the Management Console. Values: true/
false
sec_accept_jdbc_drivers = true

# Should RoboServer log all loaded URLs to the log4j audit log. Values true/false
sec_log_http_traffic = false

# If enabled RoboServer will check credentials for API requests. Values: true/false
sec_authenticate_api_requests = false

# If enabled RoboServer generate an error when accessing a https site without a valid
certificate. Values: true/false
cert_verify_https_certificates = false
```

```
# If enabled, RoboServer will only allow SSL connections from trusted client. Values
true/false
cert_verify_api_certificates = false

# Configures if the the JMX service should be enabled
enable_jmx = false

# The port number for the JMX service to listen on.
jmx_port_Number = 50100

# If enabled, input for robots is exposed through JMX. Values: true/false
jmx_show_inputs = true

# Heartbeat notification interval, in seconds
jmx_heartbeat_interval = 0

# Configure if JMX should use RMI
enable_jmx_rmi = false

# Optional RMI host and port for the JMX service. Use if you need to connect through a
firewall. Example: example.com:51001
jmx_rmi_url =

# Enables authentication for JMX requests. Values: true/false
jmx_enable_authentication = true

# The user-name used for JMX authentication
jmx_username =

# The password used for JMX authentication. This should be created using the
ConfigureRS command line tool.
jmx_password =

# Configures if the socket service should be enabled
enable_socket_service = false

# Configures which port the RoboServer should be listening on
port = 50000

# Configures if the ssl socket service should be enabled
enable_ssl_socket_service = false

# Configures which ssl port the RoboServer should be listening on
ssl_port = 50001

# Configures if the JMS service should be enabled
enable_jms_service = false

# Configures which id the RoboServer should have when running JMS
jms_id = 1

# Configures the URL of the message broker when running JMS
broker_url =

# Configures if the RoboServer should register to a Management Console
enable_mc_registration = false

# Specify which Management Console to register to formatted as: http[s]://
<hostname>:<port number>
mc_URL =

# The user name to use for authentication to the Management Console
mc_username =
```

```
# The password to use for authentication to the Management Console
mc_password =

# Specifies which cluster the RoboServer should be registered to
cluster =

# Causes RoboServer to output status and runtime events
verbose = false
```

Exemple de fichier de configuration mc.settings

```
# Settings file for Management Console. Passwords should not be edited by hand, but
using the 'ConfigureMC' command line utility.

# Should the MC web-server start a HTTP listener. Values true/false
mc_http = true

# Configures the port of the http listener.
mc_http_port = 50080

# Should the MC web-server start a HTTPS listener. Values true/false
mc_https = false

# Configures the port of the HTTPS listener.
mc_https_port = 50443

# Password for the certificate used by the HTTPS listener. This should be created
using the ConfigureMC command line tool.
mc_https_cert_password = 3W2MTrL/b2k=

# Configures which hosts are allowed to upload JDBC jar files to MC. Values: NONE,
LOCALHOST, ANY_HOST
mc_allow_jdbc_upload = LOCALHOST
```

Mode JMS

En dehors du mode Socket, les robots peuvent communiquer avec les RoboServers par l'intermédiaire du serveur de messagerie JMS (Java Message Service). Pour démarrer le système en mode JMS, démarrez d'abord le courtier JMS. Les sites Management Console et les RoboServers ne peuvent pas démarrer si le serveur de messagerie n'est pas démarré ou s'il est inaccessible. Nous recommandons de lancer le Management Console avant de lancer les RoboServers.

Il faut un certain temps (en fonction de la vitesse du réseau) pour que les RoboServers s'enregistrent sur le Management Console et pour que le Management Console découvre que les RoboServers sont en ligne.

Voir [Configurer le mode JMS](#) pour savoir comment configurer Kofax RPA pour utiliser le JMS.

Remarque À partir de Kofax RPA version 10, tous les RoboServers doivent s'enregistrer automatiquement sur le Management Console. Par conséquent, l'URL et les identifiants pour le Management Console ainsi que le nom de cluster doivent être spécifiés au [démarrage du RoboServer](#) (soit dans la ligne de commande, soit à l'aide de l'application [Paramètres du RoboServer](#)).

Avantages du fonctionnement en mode JMS

Lorsque Kofax RPA fonctionne en mode JMS, le système est plus résistant aux erreurs de réseau entre les RoboServers, le serveur de messagerie JMS et le Management Console. Bien qu'il ne puisse pas gérer une panne de réseau qui dure des heures, il peut gérer des problèmes de réseau de courte durée en quelques minutes.

Il est possible d'utiliser la fonction standard d'ActiveMQ pour assurer la redondance des courtiers afin d'accroître la stabilité. De plus, la prise en charge de l'écriture des clients personnalisés pour communiquer avec les RoboServers est grandement améliorée. Avec le KCP (Kofax RPA Control Protocol), les clients peuvent être créés dans tous les principaux langages de programmation (voir « Kofax RPA Control Protocol » dans le *guide du développeur Kofax RPA*).

Lorsqu'ils exécutent JMS, les robots ne sont pas mis en file d'attente sur les RoboServers individuels. Au lieu de cela, tous les RoboServers au sein d'un cluster partagent une file d'attente commune, qui est gérée par le serveur de messagerie JMS. Lorsqu'un RoboServer est prêt à lancer une nouvelle course de robots, il choisit un robot dans la file d'attente. Cela présente deux avantages majeurs :

- Les robots en file d'attente ne sont pas perdus en raison d'une défaillance de communication entre la Management Console et les RoboServers.
- Un robot ne peut pas rester bloqué dans une file d'attente sur RoboServer alors que d'autres RoboServers sont inactifs, ce qui permet d'utiliser plus efficacement le groupe de RoboServers.

Contrairement à la configuration en mode Socket, il n'est pas nécessaire en mode JMS d'ouvrir des ports personnalisés pour la communication entre un Management Console et des RoboServers. Les seuls ports nécessaires sont ceux du Management Console UI (par défaut 8080) et le port utilisé pour la communication avec le courtier JMS (par défaut 61616).

Dans une certaine mesure, les robots peuvent être mis en attente sur la file d'attente commune d'exécution alors que les RoboServers ne sont pas disponibles, c'est-à-dire pas encore démarrés. Notez que les requêtes d'exécution ne sont pas temporisées.

Management Console en mode JMS

Pour configurer un Management Console afin qu'il fonctionne en mode JMS, définissez `jmsEnabled` sur `"true"` dans le fichier `Configuration.xml`. Le `brokerUri`, l'espace de nom, le temps de vie des messages et les identifiants des répartiteurs sont également spécifiés dans `Configuration.xml`.

Par exemple, pour modifier l'heure du message à vivre, dans le bean de configuration `JMSC`, ajoutez la propriété `customTimeToLive` avec une entrée pour chaque type de message JMS, afin de modifier l'heure du message à vivre. La clé est le type de message et la valeur est le temps à vivre en millisecondes comme dans l'exemple suivant.

```
<property name="customTimeToLive">
  <map>
    <entry key="Execute" value="1200000"/>
    <entry key="Cancel" value="1300000"/>
  </map>
</property>
```

Après le démarrage du courtier, vous pouvez lancer un Management Console. Si le site Management Console ne parvient pas à joindre le courtier au moment du démarrage, il ne démarre pas. Dans ce cas, corrigez le problème et redémarrez le site Management Console. Une fois le démarrage réussi, le site Management Console indique qu'il est connecté au répartiteur.

Si le site Management Console perd la connexion avec le répartiteur, il le signale dans l'interface utilisateur et enregistre un message. Une fois que le Management Console rétablit la connexion, il reprend son fonctionnement normal.

Options du RoboServer en mode JMS

RoboServer accepte désormais les options suivantes liées à JMS.

Option	Description
-service jms:<id>	Démarrage d'un service JMS comprenant à la fois RQL et KCP. L'identifiant jms associé à l'adresse IP de l'hôte RoboServer identifie de manière unique un RoboServer. Un RoboServer ne peut définir qu'un seul service JMS. Le JMS et le service de socket (SSL ou autre) ne peuvent pas coexister sur le même RoboServer.
-brokerUrl	Précise l'URL du répartiteur. Par exemple, <code>failover://(tcp://localhost:61616)?startupMaxReconnectAttempts=3&timeout=15000</code> Voir la documentation du répartiteur pour le format de ce paramètre.
-jmsNamespace	Précise l'espace nom qu'un RoboServer utilise sur le répartiteur. Il doit correspondre à l'espace nom utilisé par le Management Console. La valeur par défaut est « Kapow ».

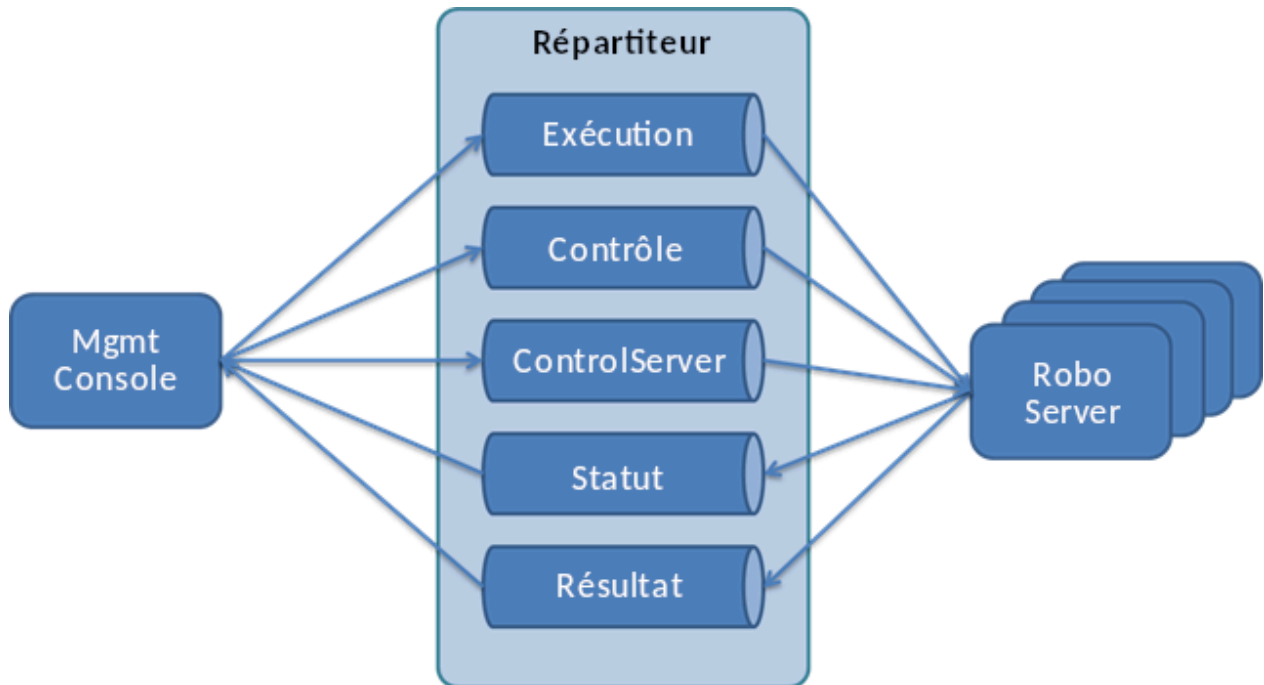
Il est obligatoire de spécifier un service JMS et une URL de courtier lors de l'exécution de JMS. Vous pouvez le faire soit en utilisant les options de la ligne de commande, soit en utilisant l'application [Paramètres du RoboServer](#).

Si un RoboServer ne parvient pas à se connecter au répartiteur lors du démarrage, il ne démarre pas. Si un RoboServer perd la connexion avec le répartiteur en phase de fonctionnement normale, il continue à exécuter les robots, mais ne reçoit pas de nouvelle requête d'exécution jusqu'à ce que la connexion soit rétablie. Si la connexion est perdue pendant une période prolongée, le Management Console considère le RoboServer comme hors ligne jusqu'à ce que la connexion soit rétablie. Sauf indication contraire dans `Configuration.xml`, l'intervalle de grâce est fixé à une minute. Pour un système très chargé ou un réseau lent, augmentez l'intervalle.

Files d'attente et sujets

Les files d'attente JMS et les sujets utilisés par Kofax RPA sont nommés `<namespace>.<protocol>.<cluster>.<destination>` où l'espace de noms est défini dans `Configuraton.xml` pour Management Console et dans `roboserver.settings` ou dans la ligne de commande pour RoboServer (la valeur par défaut est Kapow. Le protocole est soit RQL, qui est utilisé par Management Console, soit KCP pour les clients personnalisés utilisant le protocole de contrôle Kofax RPA basé sur protobuf (voir « Kofax RPA Control Protocol » dans le *guide du développeur Kofax RPA*). Le nom de cluster est défini dans **Administration > RoboServers** dans l'application Management Console. Comme les destinations sont spécifiques à chaque cluster, il n'y a aucun risque que les messages d'un cluster entrent en conflit avec les messages d'autres clusters. Enfin, la destination est l'une des suivantes.

- Exécuter
- Contrôle
- ControlServer
- État
- Résultat



Les destinations sont créées à la volée une fois qu'un Management Console et les RoboServers sont démarrés et peuvent être consultés à l'aide de la console ActiveMQ.

Durée de vie des messages

La durée de vie par défaut des messages est de dix minutes. En d'autres termes, si un message reste dans une file d'attente pendant plus de dix minutes sans être lu, il est supprimé. La configuration de la durée de vie est essentielle pour les requêtes d'exécution. En d'autres termes, si une requête d'exécution n'est pas prise en charge par un RoboServer dans les dix minutes qui suivent son placement dans la file d'attente d'exécution, elle est supprimée, l'exécution du robot est effectivement perdue et un message est écrit dans le journal. L'intention est qu'un Management Console et d'autres clients n'envoient pas de demande d'exécution plus vite que les RoboServers ne peuvent les traiter. Dans ce cas, le nombre de demandes d'exécution devrait être réduit ou la capacité d'exécution des robots des RoboServers devrait être augmentée. Pour déterminer si c'est le cas ou non, surveillez les journaux et la taille de la file d'attente d'exécution dans la section **Administration > RoboServers** du Management Console. Cliquez sur l'icône du menu **☰** à droite et sélectionnez pour afficher les **Robots en file d'attente**.

Exécuter la file d'attente

Tous les RoboServers prêts à accepter sont à l'écoute de la demande d'exécution dans la file d'attente d'exécution. Autrement dit, la file d'attente est commune à tous les RoboServers du cluster. L'un des RoboServers disponibles choisit la demande et les autres continuent d'écouter les autres demandes. Le fait que RoboServer reçoive la requête est arbitraire. En fonction de la durée de fonctionnement du robot, des charges inégales peuvent être observées de temps en temps. Si la durée de fonctionnement du robot est courte ou similaire, la charge devrait être plus régulière.

Sauf configuration contraire dans le répartiteurs, la file d'attente d'exécution est non limitée par défaut. Vous pouvez définir une limite dans le fichier Management Console `Configuration.xml`. La borne ne s'applique qu'à la file d'attente d'exécution utilisée par Management Console. Si un Management Console

est invité à lancer un robot soit par un programme soit par une exécution directe alors que la file d'attente d'exécution est pleine, l'exécution échoue et le message correspondant est écrit dans le journal.

Sujet de contrôle

Le sujet de contrôle est uniquement utilisé pour arrêter l'exécution d'un robot. Tous les RoboServers s'abonnent au thème contrôle. Un Management Console peut diffuser un message à tous les RoboServers pour arrêter l'exécution d'un robot identifié par l'identifiant d'exécution. Le RoboServer qui exécute le robot arrête le robot et signale qu'il est arrêté. Si le robot ne fonctionne plus, le message est ignoré en silence.

File d'attente ControlServer

Chaque RoboServer écoute la file d'attente du serveur de contrôle en utilisant un sélecteur de messages pour recevoir les messages destinés à un RoboServer particulier. Un RoboServer est identifié par son adresse IP et l'identifiant JMS, par exemple, `127.0.0.1:1`. Management Console utilise la file d'attente pour l'extinction d'un RoboServer, permettre le profilage et demander une mise à jour du état des robots en cours d'exécution.

Sujet d'état

Les RoboServers diffuse des messages sur ce thème et un Management Console s'y abonne.

Les RoboServers diffusent un ping toutes les 30 secondes sur le thème pour informer le Management Console qu'ils sont vivants. Si un Management Console n'a pas reçu de ping d'un RoboServer dans un certain intervalle, le Management Console considère que le RoboServer est hors ligne et les exécutions du robot sont perdues. Sauf indication contraire dans `Configuration.xml`, l'intervalle est fixé à une minute. Pour un système très chargé ou un réseau lent, augmentez l'intervalle.

Un site RoboServer fournit également des informations statistiques sur le fonctionnement des robots toutes les 20 secondes. Ces informations sont stockées dans une base de données pour la création de rapports plus avancés en utilisant Kofax Analytics for RPA.

Enfin, si un RoboServer diffuse l'état des robots en cours d'exécution, cet état est indiqué dans la fenêtre de durée d'exécution du robot en bas de la page **Administration > RoboServer** dans la section Management Console.

Liste des résultats

Les RoboServers envoient à cette file d'attente les résultats des événements survenus pendant une exécution de robot, tels que le démarrage, l'échec ou l'achèvement. La file d'attente est également utilisée pour l'envoi des résultats des exécutions du robot.

Configuration du serveur JMX

Vous pouvez utiliser le serveur JMX intégré pour surveiller un RoboServer en cours d'exécution grâce à des outils tels que JConsole. Activez JConsole en fournissant un argument sur la ligne de commande RoboServer.

Cacher les données sensibles des robots

L'option **Afficher les entrées** contrôle si les paramètres d'entrée du robot sont affichés dans l'interface de gestion. Cela permet de dissimuler des informations sensibles en matière de sécurité, comme les mots de passe.

Accès au serveur JMX

Par défaut, un serveur JMX est accessible à tous les clients ayant accès au bon port sur le serveur. En sélectionnant l'option **Utiliser un mot de passe**, le nom d'utilisateur et le mot de passe sélectionnés sont requis lors de la connexion.

Notifications de battements de cœur

Si un intervalle (en secondes) supérieur à 0 est spécifié, le serveur JMX envoie une notification de battement de cœur avec l'intervalle donné, tant qu'un RoboServer est en cours d'exécution et répond aux requêtes.

Projet RoboServer par défaut

Vous pouvez définir l'emplacement du dossier par défaut du projet RoboServer sur l'onglet Projet de paramétrage RoboServer. Par défaut, le dossier est défini sur le projet de robot par défaut créé lors du processus d'installation. Pour obtenir plus d'informations sur les projets de robots, consultez le chapitre Design Studio dans *l'aide pour Kofax RPA*.

Le projet par défaut de RoboServer n'est utilisé que par l'API. Lors de l'exécution d'un robot utilisant l'API, toute référence à des types, des snippets ou d'autres ressources est résolue en regardant dans le projet par défaut.

Modifier l'allocation de la RAM

Tel qu'installé, chaque application Kofax RPA est configurée avec une quantité maximale de RAM qu'elle peut utiliser. Cette quantité est généralement suffisante pour le travail ordinaire, mais si vous exécutez de nombreux robots en parallèle sur un RoboServer, ou si certains robots utilisent beaucoup de RAM, il peut être nécessaire d'augmenter l'allocation.

Vous pouvez modifier l'allocation pour n'importe quelle application en éditant son fichier `.conf` qui se trouve dans le sous-dossier `bin` du dossier d'installation.

Pour RoboServer, éditer : `bin/RoboServer.conf`.

Pour Design Studio, éditer : `bin/DesignStudio.conf`.

Pour modifier le fichier, suivez les étapes suivantes.

1. Ouvrez le fichier `.conf` correspondant dans un éditeur de texte.
2. Trouvez la ligne contenant le paramètre `wrapper.java.maxmemory`.
3. Décommentez la ligne (supprimez le n° de tête) et modifiez sa valeur.

Par exemple, pour permettre à un RoboServer d'utiliser jusqu'à 4 Go de RAM, saisissez ce qui suit : `wrapper.java.maxmemory=4096`.

Remarque Si le fichier `.conf` ne contient pas la ligne `wrapper.java.maxmemory`, ajoutez la ligne entière au fichier. Le fichier `.conf` ne peut être modifié que par l'utilisateur qui a installé Kofax RPA, tel que l'administrateur Windows.

Dépannage du démarrage de RoboServer Service

Si votre service ne démarre pas, recherchez les messages RoboServer dans le journal des événements de Windows. Assurez-vous que vous avez installé le service avec l'argument `wrapper.syslog.loglevel=INFO`. Pour plus d'informations, voir Kofax RPA Configuration initiale dans le *guide d'installation Kofax RPA*.

Chapitre 2

Tomcat Management Console

Pour l'environnement de production, nous recommandons fortement de déployer Management Console comme une application web ordinaire sur un serveur web Tomcat autonome.

Important Si votre installation nécessite un accès au Management Console en dehors de l'intranet de votre entreprise, configurez TLSv1.0 (ou une version ultérieure de TLS) pour qu'il fonctionne avec votre serveur Tomcat.

Le tableau suivant énumère les différences dans l'ensemble des caractéristiques.

Caractéristiques et configuration Management Console

Dossier	Intégré	Conteneur web autonome J2SE
Authentification	Super-utilisateur et utilisateurs <code>admin</code> uniques définis dans Management Console. Utilisateurs et rôles gérés par l'administrateur de Management Console.	Utilisateurs et rôles gérés par l'administrateur de Management Console. Sécurité basée sur les rôles via Active Directory ou un autre fournisseur LDAP. Single Sign-On utilisant Single Sign-On CA.
Management Console data store	Base de données Derby intégrée	Base de données gérée par conteneur (plateformes compatibles)

Remarque Le pilote Derby JDBC n'est pas distribué avec Management Console Entreprise. Voir le site web [Apache Derby](#) pour des informations sur le téléchargement du pilote JDBC Derby. Nous vous recommandons d'utiliser MySQL ou une autre base de données d'entreprise avec votre entreprise Management Console.

Les instructions pour la configuration d'un Management Console intégré sont disponibles dans l'aide en ligne Kofax RPA. Pour démarrer un Management Console intégré, voir « Start Management Console » dans la section « Management Console ».

Déploiement de Tomcat

Ce chapitre fournit des détails sur la façon d'installer manuellement Management Console sur un conteneur web J2SE autonome. Pour ce guide, nous avons choisi Tomcat. Votre conteneur web J2SE doit utiliser le composant d'exécution Java 8 ou une version ultérieure. Visitez le site Oracle Java SE Downloads et téléchargez la dernière version de Java.

Important Si votre installation nécessite un accès au Management Console en dehors de l'intranet de votre entreprise, configurez TLSv1.0 (ou une version ultérieure de TLS) pour qu'il fonctionne avec votre serveur Tomcat.

Installez Management Console sur Tomcat

Conditions préalables

- Installez la dernière mise à jour java depuis le site web d'Oracle : <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Télécharger Tomcat sur le site web d'Apache <https://tomcat.apache.org>.
 - Installez Tomcat et définissez le mot de passe de l'utilisateur.

Installation

1. Téléchargez l'installateur complet de Kofax RPA et procédez à l'installation. Sélectionnez l'option **WAR Management Console** pendant l'installation.
2. Copiez le fichier `ManagementConsole.war` du répertoire `webApps` de l'installation Kofax RPA dans le répertoire `webapps` sur le serveur Tomcat et [configurez](#) le fichier `.war`.
3. Créer un fichier contextuel Tomcat `ManagementConsole.xml` sur le serveur Tomcat à l'adresse `conf/Catalina/localhost/`. Voir [Créer un fichier contextuel Tomcat](#) pour plus d'informations.
4. Modifier le fichier `webapps/manager/WEB-INF/web.xml` sur Tomcat.
 - Pour télécharger les candidatures, modifiez les éléments suivants.

```
<multipart-config>
<!-- 150MB max -->
  <max-file-size>152428800</max-file-size>
  <max-request-size>152428800</max-request-size>
  <file-size-threshold>0</file-size-threshold>
</multipart-config>
```

- Si vos politiques exigent l'utilisation obligatoire des protocoles HTTPS sur votre réseau, activez l'en-tête HTTP Strict Transport Security Header (HSTS Header) sur le serveur Tomcat en utilisant la classe `org.apache.catalina.filters.HttpHeaderSecurityFilter` dans `web.xml`. Voir la documentation sur Apache Tomcat pour obtenir plus de détails.
- Management Console met en œuvre la politique de sécurité du contenu comme une couche de défense supplémentaire qui aide à détecter et à atténuer certaines catégories d'attaques. Un filtre `SecurityHeadersFilter` mappé à toutes les URL est déclaré et configuré dans `web.xml`. Pour configurer la valeur de l'en-tête, définissez la variable `contentSecurityPolicy`. Voici l'exemple de configuration du filtrage des en-têtes dans le fichier `web.xml` :

```
<filter>
  <filter-name>SecurityHeadersFilter</filter-name>
  <filter-
class>com.kapowtech.scheduler.server.servlet.SecurityHeadersFilter</filter-
class>
  <init-param>
    <param-name>contentSecurityPolicy</param-name>
    <param-value>default-src 'self' data: blob: 'unsafe-inline' 'unsafe-
eval'</param-value>
  </init-param>
</filter>
```

```
<filter-mapping>
  <filter-name>SecurityHeadersFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Si vous n'avez pas besoin du filtrage des en-têtes, désactivez ou supprimez le code du filtre et le mappage du filtre correspondant. Pour plus d'informations sur les options possibles et le filtrage des en-têtes, recherchez sur le web la Politique de sécurité du contenu.

5. [Démarrez Tomcat.](#)
6. [Saisir les informations relatives à la licence.](#)

Configurer ManagementConsole.war

L'application Management Console se présente sous la forme d'un fichier WAR (web Application Archive) nommé `ManagementConsole.war`, qui se trouve dans le dossier `/webApps` du dossier d'installation Kofax RPA.

La version de `ManagementConsole.war` livrée avec Kofax RPA est configurée pour fonctionner à l'intérieur de RoboServer. Avant de pouvoir le déployer en tant qu'application autonome sur Tomcat, il doit être reconfiguré pour s'adapter à votre environnement.

Un fichier WAR est compressé à l'aide d'un fichier zip compressé. Pour accéder aux fichiers de configuration, vous devez extraire le fichier zip. Une fois que les fichiers de configuration sont mis à jour, vous redécompressez et déployez `ManagementConsole.war` sur votre serveur Tomcat.

Le tableau suivant contient une liste des fichiers de configuration relatifs à la racine du `ManagementConsole.war` décompressé.

Fichiers de configuration

Fichier	Configurer	Remarques
WEB-INF/Configuration.xml	Clustering, chiffrement des mots de passe, plug-in REST	Si vous copiez le fichier d'une version antérieure, il sera automatiquement mis à jour dès que vous lancerez le Management Console
WEB-INF/login.xml	Administrateurs et utilisateurs, c'est ici que vous vous intégrez au LDAP	
WEB-INF/classes/log4j2.properties	Journalisation des applications	
WEB-INF/spring/authentication.xml	Authentification des utilisateurs	
WEB-INF/roles.xml	Rôles intégrés dans Management Console.	

Fichiers de configuration Spring

`Configuration.xml`, `login.xml`, `roles.xml` et `authentication.xml` sont tous des fichiers de configuration Spring (www.springsource.org) et partagent la même syntaxe générale décrite ici.

Spring est configuré par une série de beans, et chaque bean a des propriétés qui configurent un morceau de code à l'intérieur de l'application. La syntaxe générale :

```
<bean id="id" class="SomeClass">
  <property name="myName" value="myValue"/>
</bean>
```

Fichier	Configurer
id="id"	L'identifiant du bean est un identificateur interne, que l'application utilise pour se référer au bean. Il est également mentionné à l'aide du nom du bean.
class="SomeClass"	La classe identifie le composant de code que le bean configure.
<property name="myName" value="myValue"/>	Définit une propriété avec le nom myName et la valeur myValue. Cela permet de configurer une propriété sur le composant de code défini par l'attribut de classe.

Dans les versions de Kofax RPA antérieures à la version 9.3, les identifiants d'accès des utilisateurs étaient définis manuellement dans le fichier login.xml. À partir de la version 9.3, la gestion des utilisateurs se fait dans la section Utilisateurs & groupes du menu Administration du site Management Console. Dans la version entreprise de Kofax RPA, la gestion des utilisateurs est activée par défaut. La population d'utilisateurs des installations précédentes peut être effectuée en utilisant la fonctionnalité de sauvegarde de Kofax RPA. Pour les intégrations LDAP et SAML, vous devez encore modifier le fichier login.xml.

Dépannage

Si vous avez des problèmes pendant l'installation, vous devez vérifier le journal Tomcat dans le dossier /logs de votre installation Tomcat. Pendant le processus de configuration, il est souvent plus facile de lancer Tomcat à partir de la ligne de commande, car il imprime les messages d'erreur directement dans la fenêtre de la ligne de commande.

Créer une nouvelle base de données

Nous vous recommandons vivement de créer une nouvelle base de données pour les tables utilisées par Management Console. La base de données doit répondre à deux exigences.

- Prise en charge Unicode
- Comparaison sensible à la casse

La prise en charge de l'Unicode est nécessaire car des caractères non ASCII, comme le danois Æ, l'allemand ß ou le cyrillique Ë peuvent être donnés en entrée aux robots. Cette entrée est stockée dans la base de données, et sans la prise en charge de l'Unicode, ces caractères peuvent être stockés de manière incorrecte.

La comparaison sensible à la casse est nécessaire car il est possible de télécharger un robot nommé a.robot et un autre nommé A.robot. Sans comparaison sensible à la casse, le téléchargement de la dernière version l'emporterait sur la première.

Important Veuillez créer et maintenir les bases de données des produits Kofax RPA conformément aux recommandations de la documentation du produit. Si vous envisagez de modifier ou de personnaliser la base de données, ne procédez pas sans consulter Kofax; sinon, les résultats sont imprévisibles et le logiciel peut devenir inutilisable.

Les serveurs de bases de données traitent très différemment l'Unicode et la comparaison des majuscules et des minuscules. La liste suivante contient des recommandations pour les systèmes de bases de données pris en charge.

Recommandations pour la prise en charge de l'Unicode et de la comparaison sensible à la casse

Base de données	Recommandations
IBM DB2	Créer la base de données à l'aide de CODESET UTF-8
MySQL 5.x	Créer la base de données avec le classement utf8_bin : <pre>CREER UNE BASE DE DONNEES KAPOW_MC COLLATE utf8_bin</pre>
Oracle	Nous utilisons les types NVARCHAR2, NCLOB pour l'Unicode. Pour une comparaison sensible à la casse, assurez-vous que NLS_COMP est réglé sur BINARY
Serveur Microsoft SQL	Nous utilisons les types NVARCHAR, NTEXT pour l'Unicode. Pour une comparaison sensible à la casse, créez la base de données avec un classement sensible à la casse telle que Latin1_General_100_CS_AS_WS : <pre>CREER UNE BASE DE DONNEES KAPOW_MC COLLATE Latin1_General_100_CS_AS_WS</pre> Pour de plus amples informations, consultez Configurer le serveur Microsoft SQL dans Windows Integrated Security ici : Créer un fichier contextuel Tomcat .
Sybase Adaptive Server Enterprise	Nous utilisons les types NVARCHAR, NTEXT pour l'Unicode. Pour une comparaison sensible à la casse, assurez-vous que l'ordre de tri est binaire (voir sp_helpsort). Les scripts suivants utilisent NVARCHAR(255) NOT NULL UNIQUE, en raison des limitations de l'index, cela ne fonctionnera pas sur un ASE avec une taille de page de 2K, à moins que vous ne modifiez les scripts pour utiliser NVARCHAR(200) NOT NULL UNIQUE (ou que vous installiez sur un serveur de pages 4/8/16K)

Les tableaux utilisés par Management Console peuvent être regroupés en 3 catégories : Tableaux des plateformes, tables de journalisation et tables de vue des données. Les tables de la plateforme contiennent des informations exclusives à Management Console telles que les robots chargés et leurs informations de planification, tandis que les tables de journalisation et de vue des données sont partagées avec les RoboServers.

Privilèges des utilisateurs

Lorsqu'un Management Console démarre, il essaie automatiquement de créer les tables de plateforme et les tables de journalisation requises (si elles n'ont pas déjà été créées par un RoboServer). Cela signifie que le compte utilisateur utilisé pour accéder à la base de données doit disposer des privilèges CREATE TABLE et ALTER TABLE ainsi que du privilège CREATE TEMPORARY TABLES pour restaurer une sauvegarde. Les utilisateurs d'Oracle ont également besoin du privilège CREATE SEQUENCE. Si cela n'est pas possible, vous pouvez demander à votre administrateur de base de données de créer les tables en utilisant les scripts suivants.

De plus, l'utilisateur doit être autorisé à SÉLECTIONNER, INSÉRER, METTRE À JOUR, SUPPRIMER pour que le système fonctionne correctement.

Scripts SQL pour les tables Management Console

Les scripts SQL sont inclus avec votre copie de Kofax RPA dans le répertoire `documentation\sql`. Voir [Scripts SQL pour les tables Kofax RPA](#) pour plus d'informations.

Management Console utilise un composant de planification tiers appelé Quartz. Le quartz nécessite également un certain nombre de tables qui doivent se trouver parmi les autres tables de la plateforme. Ces tableaux sont également créés automatiquement au démarrage d'un Management Console, ou peuvent être créés manuellement à l'aide des scripts du [Scripts SQL pour les tables Kofax RPA](#).

Créer un fichier contextuel Tomcat

Dans un environnement d'entreprise, les bases de données sont souvent accessibles via une source de données. Cette section vous montre comment configurer votre Tomcat avec une source de données qui se connecte à un serveur de base de données MySQL local.

Dans Tomcat, les sources de données sont définies dans le contexte des applications. Le contexte peut être déclaré soit intégré, soit externe à l'application. Lorsque le contexte est intégré, il est défini dans le fichier `context.xml` qui doit être situé à l'intérieur du fichier WAR dans le dossier META-INF. Lorsqu'il est déclaré en externe, le fichier doit être situé dans le dossier `/conf/Catalina/localhost` du Tomcat et le nom du fichier doit être `ManagementConsole.xml` (même nom que le fichier WAR déployé). Bien que Tomcat recommande de déployer avec un contexte intégré, car il fournit une unité de déploiement unique, nous utilisons une définition de contexte externe dans ce guide, car cela facilite la modification du fichier. Une fois que vous avez affiné votre configuration, vous pouvez intégrer le fichier de contexte et déployer le fichier War dans votre environnement de production.

Ajouter une source de données de la plateforme

Créez le fichier `ManagementConsole.xml` sur Tomcat dans le dossier `conf/Catalina/localhost` et ajoutez le contenu suivant.

```
<Context useHttpOnly="true">
  <!-- Default set of monitored resources -->
  <WatchedResource>WEB-INF/web.xml</WatchedResource>

  <Resource name="jdbc/kapow/platform" auth="Container"
    type="javax.sql.DataSource"
      maxTotal="100" maxIdle="30" maxWaitMillis="-1"
      validationQuery="/* ping */" testOnBorrow="true"
```

```

        username="MyUser" password="MyPassword"
        driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:3306/KAPOW_MC?
useUnicode=yes&characterEncoding=UTF-8&rewriteBatchedStatements=true"/>
    </Context>

```

Le paramètre `url` suivant est une URL JDBC. Les attributs du nom d'utilisateur et du mot de passe sont utilisés par Tomcat pour créer un pool de connexion utilisé lors de la connexion à la base de données.

Les sources de données sont définies différemment pour les autres bases de données. Par exemple, si vous utilisez le serveur Microsoft SQL, les trois lignes suivantes doivent être utilisées :

```

username="MyUser" password="MyPassword"
        driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
        validationQuery="SELECT 1" testOnBorrow="true"
        url="jdbc:sqlserver://localhost:1433;DatabaseName=MyDbName"/>

```

L'URL `jdbc:mysql://localhost:3306/KAPOW_MC?`
`useUnicode=yes&characterEncoding=UTF-8` fait référence à une base de données nommée `KAPOW_MC` dans votre MySQL local. Pour MySQL, il est recommandé d'ajouter `useUnicode=yes&characterEncoding=UTF-8` à toutes les chaînes de connexion, sinon le pilote JDBC ne peut pas gérer correctement les caractères chinois, japonais ou autres caractères utf-8 sur 3 octets, car "&" ne peut pas apparaître directement dans le fichier xml contextuel, nous devons l'encoder comme ceci « & ».

`rewriteBatchedStatements=true` indique au pilote JDBC MySQL les insertions/mises à jour par lots et devrait permettre d'améliorer les performances d'insertion pour les robots kapplets.

Le paramètre `driverClassName` contrôle quel pilote JDBC est utilisé ; chaque fournisseur de base de données fournit un pilote JDBC pour sa base de données, que vous devez télécharger. Le pilote JDBC, généralement un simple fichier `.jar`, doit être copié dans le dossier `/lib` sur Tomcat.

La `validationQuery` est utilisée par Tomcat pour vérifier que la connexion obtenue à partir du pool de connexion est toujours valide (car le serveur de base de données peut avoir fermé la connexion). La requête de validation est légère et utilise très peu de ressources sur le serveur de base de données, cette liste contient des requêtes de validation pour les bases de données prises en charge.

Demandes de validation

Base de données	Demande
MySQL	<code>/* ping */</code>
Serveur Microsoft SQL	<code>SÉLECTION 1</code>
Sybase Adaptive Server Enterprise	<code>SÉLECTION 1</code>
IBM DB2	<code>VALEURS(1)</code>
Oracle	<code>SÉLECTIONNEZ 1 PARMIS LES DOUBLES</code>

Notez que le pilote MySQL JDBC supporte une requête spéciale `/* ping */` légère, consultez la section 6.1 du manuel de JConnector pour obtenir plus d'informations.

Pour obtenir plus d'informations sur la configuration du contexte et les sources de données, voir les ressources JNDI HOW-TO et la source de données JNDI HOW-TO.

Configurer le serveur Microsoft SQL dans Windows Integrated Security

Si vous utilisez Microsoft SQL Server et Windows Integrated Security, les ordinateurs équipés d'un Design Studio et d'un RoboServer doivent être conformes à ce qui suit :

- Exécution sur Windows
- Exécution avec un compte d'utilisateur qui a reçu l'autorisation d'accéder à la base de données
- Le pilote JDBC doit être installé manuellement comme décrit plus loin dans cette partie

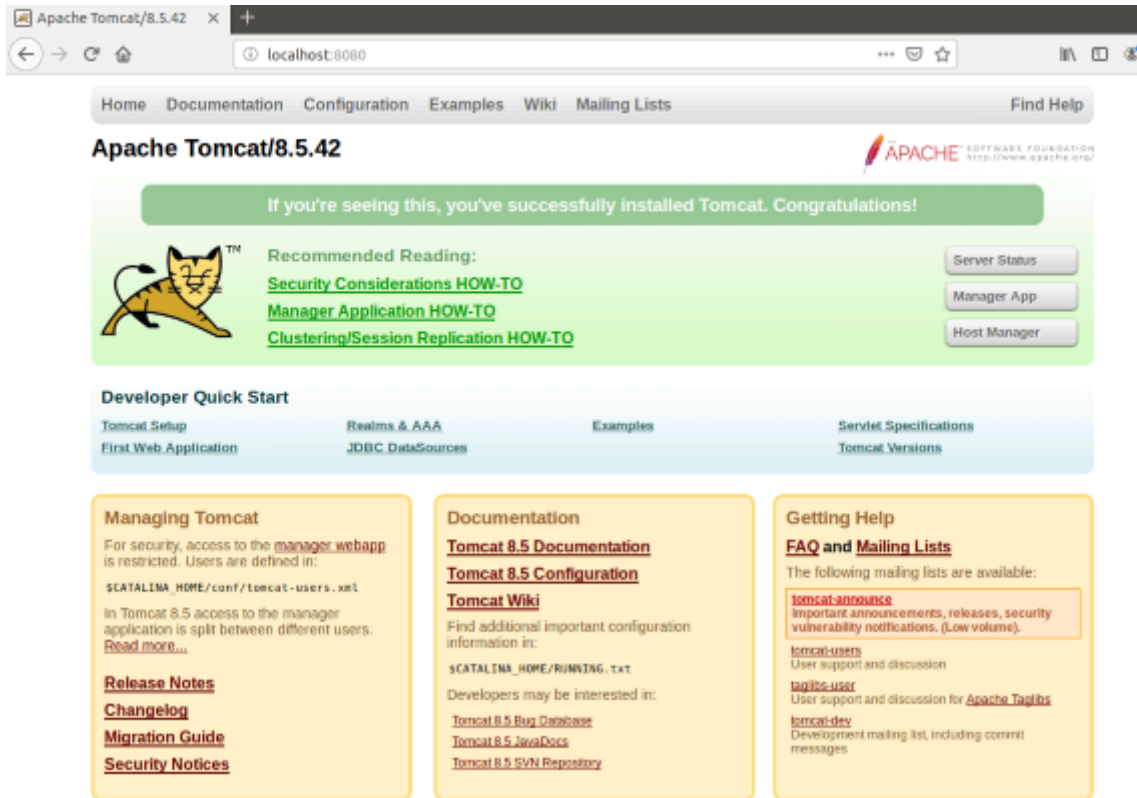
Pour configurer le serveur Microsoft SQL dans Windows Integrated Security, procédez comme suit :

- Installez les pilotes JDBC dans les dossiers Tomcat : copiez le fichier JAR dans le dossier `lib`, copier le fichier DLL dans le dossier `nativelib`.
- Ne téléchargez pas le fichier jar sur Management Console, car il n'est pas possible pour Management Console de distribuer le pilote JDBC.
- Assurez-vous d'ajouter `integratedSecurity=true` à l'URL de connexion, à la fois dans votre `ManagementConsole.xml` (ou `context.xml`) et dans la définition du type de base de données dans un Management Console ainsi que dans toute définition locale de Design Studio. Voir « Ajouter un type de base de données » dans l'aide de Kofax RPA.

Nous sommes maintenant prêts à lancer le serveur Tomcat.

Démarrer Tomcat

Démarrez votre serveur Tomcat, attendez quelques secondes que l'application soit déployée, puis naviguez vers `http://localhost:8080/`. Vous devriez voir la page suivante.



Aucun fournisseur de Page d'accueil Apache

Par défaut, un ensemble d'identifiants est disponible (nom d'utilisateur - `admin`, mot de passe - `admin`) avec des privilèges d'administrateur. Plus tard, vous pourrez changer le mot de passe et créer d'autres utilisateurs et groupes sur la page **Utilisateurs & groupes** sous la rubrique **Administration** du Management Console.

Ouvrez maintenant le site <http://localhost:8080/ManagementConsole>, vous devriez voir l'écran de connexion.

Saisissez `admin` comme nom d'utilisateur et `admin` comme mot de passe et cliquez sur **Connexion**.

Saisir les informations relatives à la licence

Après la connexion, la fenêtre de licence s'affiche.

Saisissez les informations relatives à la licence Kofax RPA et cliquez sur « **Enregistrer** ». Vous devriez voir une boîte de dialogue affichant les fonctionnalités activées par votre clé de licence.

Rôles prédéfinis des utilisateurs

Management Console propose des rôles que les utilisateurs peuvent avoir. Les rôles sont attribués à l'utilisateur d'un groupe de sécurité. Les permissions des utilisateurs sont calculées en fonction des rôles qui sont associés aux groupes de sécurité dont l'utilisateur est membre. Vous pouvez modifier les rôles

intégrés ou ajouter des rôles supplémentaires. Les rôles intégrés sont définis dans le fichier `roles.xml`. Voir [Autorisations de projets](#) pour plus d'informations.

Rôles intégrés

Management Console propose quelques rôles intégrés que les utilisateurs peuvent avoir. Les rôles sont attribués à l'utilisateur d'un groupe de sécurité. Les permissions des utilisateurs sont calculées en fonction des rôles qui sont associés aux groupes de sécurité dont l'utilisateur est membre. Vous pouvez modifier les rôles intégrés ou ajouter des rôles supplémentaires.

- **Administrateur du projet** : Un utilisateur ayant ce rôle administre un ou plusieurs projets et a le droit d'attribuer un rôle à un groupe pour ces projets. Cet utilisateur a un droit de regard sur RoboServer et les paramètres du cluster sans les modifier. L'administrateur de projet n'est pas membre du groupe Administrateurs RPA (de plus amples informations vous sont proposées dans cette section).
- **Développeur** : Les développeurs ont le droit de charger, télécharger et visualiser tous les types de ressources du répertoire. Un utilisateur ayant ce rôle peut créer, modifier et supprimer des planifications, exécuter des robots, consulter des journaux d'exécution et des clusters.
- **Visionneuse** : Les téléspectateurs ont les mêmes droits de regard que les développeurs et le droit de modifier ou d'exécuter n'importe quoi.
- **API** (Cet utilisateur se connecte uniquement en tant que service, en s'authentifiant via l'API) : Un utilisateur ayant ce rôle peut utiliser l'API du répertoire pour lire et écrire dans le répertoire. Un utilisateur ayant ce rôle ne peut pas faire fonctionner des robots en utilisant REST, mais il peut exécuter des robots en utilisant RQL.
- **RoboServer** (Cet utilisateur se connecte uniquement en tant que service, en s'authentifiant via l'API) : Un utilisateur restreint qui ne peut lire qu'à partir du répertoire. Ce rôle est utilisé par les RoboServers pour accéder à un cluster, récupérer des éléments du répertoire et demander des mots de passe au magasin des mots de passe.
- **Administrateur Kapplet** : un utilisateur qui peut créer, visualiser, exécuter et modifier des Kapplets.
- **Utilisateur Kapplet** : un utilisateur qui peut visualiser et exécuter des Kapplets. Un utilisateur ayant ce rôle ne peut pas accéder à Management Console s'il n'a pas d'autres droits.
Pour obtenir plus d'informations sur les rôles des utilisateurs de Kapplets, voir « Rôles des utilisateurs de Kapplets » dans *l'Aide pour Kofax RPA*.
- **Client Magasin des mots de passe** : Un utilisateur ayant ce rôle complémentaire peut accéder au magasin des mots de passe. Le rôle est fourni en plus des autres rôles, tout comme le rôle de développeur. Ce rôle ne donne accès qu'au magasin des mots de passe de Management Console.
- **Utilisateur DAS Client** (Cet utilisateur se connecte uniquement en tant que service, en s'authentifiant via l'API) : Il s'agit d'un utilisateur créé pour les clients distants du Desktop Automation Service (DAS), et qui ne peut accéder qu'à l'API DAS. L'utilisateur du client DAS a le droit d'annoncer un DAS à Management Console, et de récupérer la configuration du DAS.
- **Utilisateur du service VCS** (cet utilisateur se connecte uniquement en tant que service, en s'authentifiant via l'API) : L'utilisateur du service de contrôle de version se voit accorder un ensemble de droits spéciaux pour le Synchronizer. Ce rôle a le droit d'ajouter, de modifier et de supprimer des ressources. C'est le seul rôle qui peut se déployer au nom d'un autre utilisateur pour utiliser la fonction de « déploiement » dans le service de contrôle de version.
- **Process Discovery Client** (cet utilisateur se connecte uniquement en tant que service, en s'authentifiant via l'API) : Ce rôle permet aux composants de Process Discovery d'interagir avec Management Console.

Utilisateur « admin » intégré

admin est un super-utilisateur qui a accès à tout. Il n'est pas membre du groupe Administrateurs RPA et ne peut être membre d'aucun groupe. Le mot de passe par défaut de l'utilisateur admin est disponible (nom d'utilisateur - `admin`, mot de passe - `admin`). Vous pouvez changer le mot de passe de l'utilisateur admin comme décrit dans « Réinitialiser le mot de passe de l'utilisateur » dans l'*Aide pour Kofax RPA*.

Dans une configuration d'intégration LDAP, le groupe **admin** est défini dans le cadre de la configuration LDAP. L'utilisateur **admin** peut alors se connecter et définir quels groupes LDAP doivent être mis en correspondance avec le développeur, l'administrateur de projet, RoboServer, et d'autres rôles.

Dans une configuration d'utilisateur interne, l'utilisateur **admin** est créé au premier démarrage et peut ensuite se connecter et créer des administrateurs, des développeurs et d'autres utilisateurs.

Droits spéciaux pour les utilisateurs « admin » intégrés

Outre le fait d'être l'utilisateur initial, **admin** dispose de droits spéciaux, comme :

- Dans la section RoboServers de Management Console, **admin** peut cliquer sur un nœud RoboServer et demander un tracé de la pile à partir du nœud RoboServer correspondant.
- Seul l'utilisateur **admin** peut créer et importer des sauvegardes.
- Dans le magasin des mots de passe, l'utilisateur **admin** peut déplacer les mots de passe vers un autre projet.

Remarque Lorsque vous restaurez une sauvegarde de Management Console, le super-utilisateur `admin` par défaut est remplacé par un super-utilisateur de la sauvegarde. Utilisez les identifiants spécifiés dans le Management Console restauré.

Groupe intégré

Administrateurs RPA : Les utilisateurs appartenant à ce groupe ont tous les droits pour tous les projets (à l'exception des droits spéciaux des utilisateurs **admin**), tels que la création de nouveaux administrateurs et utilisateurs dans tout type de projet. Pour qu'un utilisateur devienne un administrateur, il doit être ajouté à ce groupe.

Remarque

- Le groupe Administrateurs RPA est visible lorsque la gestion interne des utilisateurs est activée, et il est vide par défaut.
- Lors de la restauration d'une sauvegarde créée sur une version antérieure à 10.7, les utilisateurs avec le rôle d'Administrateur deviennent membres du groupe Administrateurs RPA.

Vérification des tentatives de connexion

Par défaut, la vérification du nombre de tentatives de connexion effectuées par un utilisateur et du temps d'attente avant la prochaine tentative est désactivée. Pour activer cette fonctionnalité, modifiez la section suivante dans le fichier `authentication.xml` qui se trouve dans : `<Tomcat installation folder>`

`\webApps\Management Console\WEB-INF\spring`

```
<bean id="loginAttemptService"
      class="com.kapowtech.scheduler.server.spring.security.LoginAttemptService"
      lazy-init="true">
  <constructor-arg type="boolean" value="false"/>
  <constructor-arg type="int" value="3"/>
  <constructor-arg type="int" value="10"/>
</bean>
```

```
</bean>
```

Réglez la première valeur sur **vrai**. Les deuxième et troisième valeurs concernent respectivement le nombre de tentatives de connexion (3 dans cet exemple) et le temps d'attente en minutes avant la prochaine tentative (10 dans cet exemple).

Autorisations de projets

Le compte d'utilisateur admin utilisé pour se connecter contourne les autorisations normales du projet appliquées aux utilisateurs réguliers, car admin est le super-utilisateur. Le super-utilisateur ne peut être membre d'aucun groupe, et a un accès illimité à tous les projets.


Pour modifier le mot de passe de l'administrateur et créer de nouveaux utilisateurs et groupes, rendez-vous sur Management Console > **Administration** > **Utilisateurs & groupes**. Le modèle de sécurité est défini en fonction du rôle de l'utilisateur ; après avoir créé un utilisateur, vous devez l'ajouter à un ou plusieurs groupes associés à des rôles spécifiques dans un ou plusieurs projets, comme le montrent les exemples de procédures suivants.

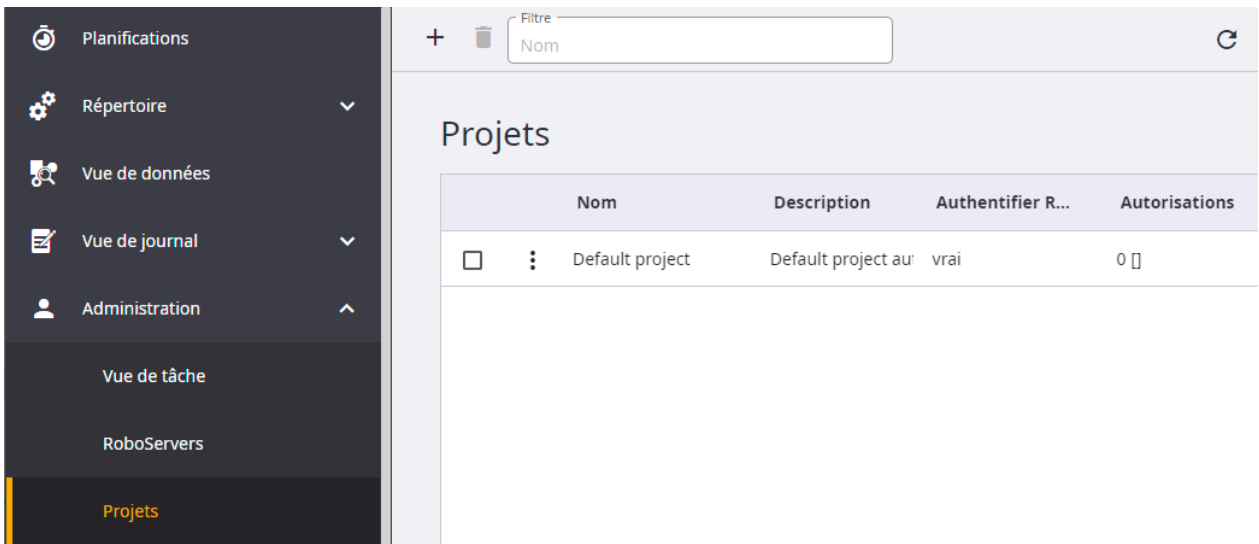
Créer un groupe de « développeurs »

1. Dans l'onglet Groupes, cliquez sur le signe plus.
La boîte de dialogue Créer un groupe apparaît.
2. Précisez le nom « Développeurs » et saisissez une description.
3. Cliquez sur **OK**.
Le groupe apparaît dans le tableau.

Créer un utilisateur « Dev »

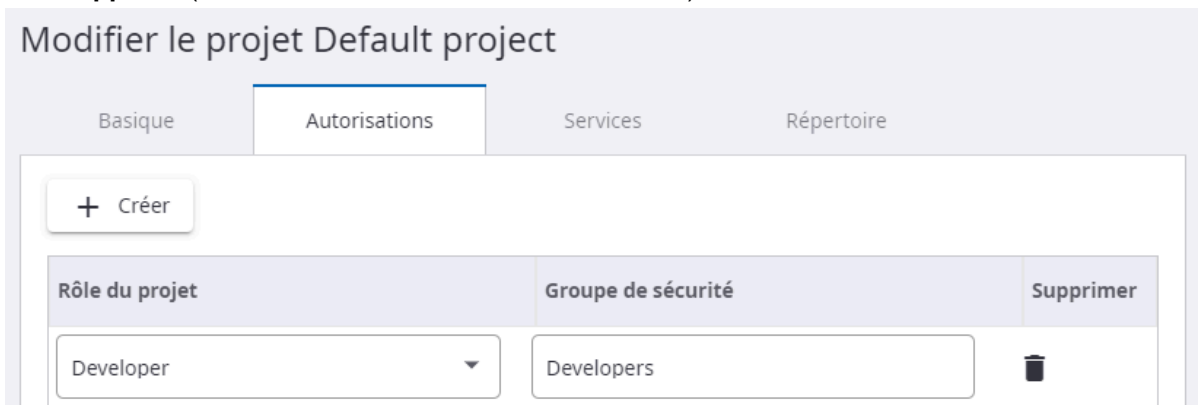
1. Dans l'onglet Utilisateurs, cliquez sur le signe plus.
La boîte de dialogue Créer un utilisateur apparaît.
2. Indiquez le nom d'utilisateur « Dev », le mot de passe, le nom complet et l'e-mail de l'utilisateur, puis sélectionnez le groupe « Développeurs ».
3. Cliquez sur **OK**.
L'utilisateur apparaît dans le tableau.

Vous devez maintenant attribuer des autorisations à l'utilisateur pour qu'il puisse se connecter. Connectez-vous en tant qu'administrateur et allez dans **Administration** > **Projets**. Dans cette section, cliquez sur l'icône du menu  à droite et cliquez pour afficher la colonne **Permissions**. Actuellement, il affiche « 0 » pour les autorisations de ce projet.



Autorisations initiales de projet

1. Cliquez sur **Modifier** dans le menu contextuel de **:** pour le projet par défaut et allez à l'onglet **Permissions**. Il y a une grille avec les colonnes **Rôle dans le projet** et **Groupe de sécurité**. Le rôle de projet détermine un ensemble d'activités qui peuvent être effectuées sur Management Console, comme le téléchargement de robots, la création de planifications, la visualisation de journaux, etc. Dans le cadre d'un projet, vous attribuez un rôle à un groupe de sécurité. De cette façon, tous les utilisateurs du groupe de sécurité sélectionné pourront effectuer les activités autorisées par le rôle de projet qui leur a été attribué.
2. Cliquez sur le signe plus pour ajouter des autorisations dans ce projet. Cela ajoute une nouvelle ligne à la grille, et insère une boîte déroulante nous permettant de sélectionner un rôle de projet. Sélectionnez le rôle de **Développeur** du projet.
3. Cliquez maintenant sur la colonne **Groupe** de sécurité et sélectionnez le groupe de sécurité des **Développeurs** (dont notre utilisateur « Dev » est membre). Il devrait ressembler à cela :



4. Cliquez maintenant sur **Soumettre**.

Tous les membres du groupe « Développeur » peuvent désormais effectuer les activités autorisées par le rôle de développeur. La colonne Permissions du projet par défaut affiche désormais l'autorisation « 1 »

Projets				
	Nom	Description	Authentifi...	Autorisations
<input type="checkbox"/>	⋮ Default project	Default project at	vrai	1 [Developer=Developers]

Ensuite, connectez-vous en tant qu'utilisateur « Dev » et voyez comment les autorisations sont reflétées dans le Management Console. Vous vous déconnectez en cliquant sur le bouton de menu dans le coin supérieur droit, puis vous vous connectez en tant qu'utilisateur dev. Allez maintenant dans la **Vue du journal**, sélectionnez le journal RoboServer dans le panneau de gauche. Remarquez que le bouton de suppression est désactivé et qu'en positionnant votre souris dessus, une info-bulle indiquant que vous n'avez pas la permission de supprimer les messages de RoboServer.

Vous pouvez attribuer plusieurs rôles au même groupe de sécurité, et vous pouvez attribuer le même rôle à plusieurs groupes de sécurité. Si un utilisateur a plusieurs rôles, il peut faire tout ce qu'au moins un des rôles lui permet de faire. Avec plusieurs projets sur Management Console, les utilisateurs de différents projets peuvent être complètement séparés en assignant à leurs groupes des rôles spécifiques à chaque projet.

Les rôles prédéfinis sont des suggestions, mais à l'aide du fichier roles.xml, vous pouvez ajouter un nombre quelconque de rôles supplémentaires ou modifier les rôles existants pour les adapter à vos besoins.

Les activités qui peuvent être effectuées dans les sections Paramètres, Sauvegardes et Licence ne sont disponibles que pour les utilisateurs membres du groupe Administrateurs RPA.

Pour obtenir plus d'information sur l'utilisation de vos comptes d'utilisateur LDAP, veuillez consulter la rubrique [Configuration avancée>Intégration LDAP](#).

Sécurité

Dans le fichier WEB-INF/Configuration.xml, il est possible de configurer certains paramètres de sécurité supplémentaires pour le Management Console.

La section de configuration de la sécurité du fichier ressemble à ceci :

```
<bean id="securityConfiguration" class="com.kapowtech.mc.config.SecurityConfiguration">
  <property name="allowScriptExecution" value="false"/>
  <property name="jdbcDriverUpload" value="LOCALHOST"/>
</bean>
```

Il contient deux paramètres de sécurité qui sont décrits ci-dessous.

Par défaut, seul l'utilisateur admin, lorsqu'il accède au site Management Console à partir de l'hôte local, est autorisé à télécharger les pilotes JDBC. Pour changer ce comportement, modifiez la propriété jdbcDriverUpload. Voici les valeurs possibles que peut avoir la propriété :

- AUCUNE : le téléchargement des pilotes JDBC n'est autorisé pour aucun utilisateur
- LOCALHOST : la valeur par défaut, l'utilisateur admin est autorisé à télécharger des pilotes s'il accède au Management Console depuis l'hôte local
- ANY_HOST : l'utilisateur admin est autorisé à télécharger des pilotes depuis n'importe quel hôte

Liste de contrôle pour le déploiement

Utilisez la liste de contrôle suivante pour vous assurer que toutes les tâches de déploiement sont effectuées dans le bon ordre.

Liste de contrôle pour le déploiement

Élément	Description
Télécharger et installer une version compatible du serveur Apache Tomcat	Si votre configuration requiert un accès au Management Console en dehors de l'intranet de votre entreprise, assurez-vous que le protocole SSL est configuré pour fonctionner avec votre serveur Tomcat.
Télécharger et installer Java (JDK 1.8.0_221 ou version ultérieure)	Management Console ne s'initialise pas correctement si Apache Tomcat est démarré en utilisant une version de Java autre que Java 8.
Assurez-vous que la variable JAVA_HOME pointe vers l'installation Java.	
Démarrez Apache Tomcat et confirmez que le serveur Apache Tomcat sera en ligne avant de tenter de déployer l'application Management Console.	<ul style="list-style-type: none"> • Vous pouvez utiliser la commande d'exécution catalina du répertoire <code>\bin</code> pour démarrer le serveur. • En allant sur <code>http://localhost:8080</code> dans le navigateur, la page de démarrage par défaut d'Apache Tomcat devrait s'afficher.
À partir d'une installation sur Kofax RPA, localisez le fichier <code>ManagementConsole.war</code> qui sera déployé sous le répertoire Apache Tomcat <code>\webapps</code>	Il faut s'attendre à des erreurs lors du démarrage initial de l'application à ce stade, car le serveur n'a pas encore été configuré. Nous décompressons simplement le fichier <code>.war</code> afin de pouvoir accéder facilement aux fichiers que nous devons modifier pour terminer le processus d'installation et de configuration.
Éteignez le serveur Apache Tomcat.	
Créer une nouvelle base de données qui sera utilisée par l'entreprise Management Console pour conserver sa configuration.	<ul style="list-style-type: none"> • Sélectionnez l'une des plateformes de support. • L'installation de Kofax RPA contient les scripts SQL CREATE et DROP nécessaires pour créer toutes les tables de base de données requises. Voir Scripts SQL pour les tables Kofax RPA pour plus d'informations. Notez que ces scripts ne doivent être utilisés que si le compte d'utilisateur que l'application utilisera pour se connecter à la base de données ne dispose pas des privilèges CREATE sur le schéma.

Élément	Description
Créez le compte utilisateur DB à utiliser par l'application pour se connecter à la base de données via JDBC.	
Créez le fichier Tomcat Context : <code>ManagementConsole.xml</code>	<ul style="list-style-type: none"> La requête de validation à spécifier est spécifique à la base de données. L'aide en ligne contient toutes les valeurs acceptées pour toutes les bases de données prises en charge. N'oubliez pas de mettre à jour les paramètres Username, Password et DatabaseName dans la chaîne d'URL JDBC avec les valeurs correctes pour votre base de données.
Avant de démarrer l'application, veuillez noter que la base de données Management Console (comme spécifié dans le <code>ManagementConsole.xml</code>) ne contient pour l'instant aucune table relative au processus Management Console.	<ul style="list-style-type: none"> Lorsque l'application est lancée, les tables de base de données nécessaires sont automatiquement créées si elles ne sont pas présentes, en supposant que le compte DBUser dispose des privilèges CREATE. Si l'utilisateur ne dispose pas des privilèges CREATE, les scripts SQL CREATE se trouvent dans le répertoire <code>documentation\sql</code> de votre répertoire d'installation Kofax RPA. Voir Scripts SQL pour les tables Kofax RPA pour plus d'informations.
N'oubliez pas de déployer le fichier JDBC.jar nécessaire sous le répertoire d'installation d'Apache Tomcat.	<ul style="list-style-type: none"> Le déploiement de <code><APACHE_TOMCAT_INSTALL_DIR>\lib</code> le rend disponible pour TOUTES les applications exécutées sous Tomcat. Le déploiement vers <code><APACHE_TOMCAT_INSTALL_DIR>\webapps\ManagementConsole\WEB-INF\lib</code> rend le fichier JDBC.jar uniquement accessible à l'application Management Console elle-même.
Redémarrer le serveur Apache Tomcat	Confirmez le chargement de la page d'accueil principale du Tomcat : <code>http://localhost:8080</code>
Essayez de vous connecter à l'entreprise Management Console en tant qu'utilisateur - admin, mot de passe - admin	<code>http://localhost:8080/ManagementConsole</code>
Saisissez les clés de licence du logiciel Kofax RPA	Voir Saisir les informations relatives à la licence .

Outils Docker pour déploiement Kofax RPA

Kofax propose des outils Docker pour un déploiement rapide et facile Kofax RPA dans vos environnements Linux et Windows. Outils Docker pour Kofax RPA vous aide à créer des images Docker pour les composants RPA. Actuellement, des images sont disponibles pour les composants suivants.

Remarque En raison de la consommation de ressources nettement plus élevée des conteneurs Windows Docker, la version Linux est recommandée lorsqu'elle est disponible. Les composants qui s'exécutent uniquement sur Windows ne prennent en charge que la version du conteneur Windows Docker.

Composant	Linux	Windows
Management Console	Oui	Oui
RoboServer	Oui	Oui
Système de fichiers du robot	Oui	Oui
Synchronizer	Oui	Oui
Kaplets	Oui	Oui
Document Transformation	Non	Oui
Kofax Analytics for RPA	Non	Oui

Ce chapitre fournit des détails sur les outils Docker et des exemples d'utilisation. Pour obtenir plus d'informations sur Docker, consultez <https://www.docker.com>. Pour obtenir plus d'informations sur le déploiement manuel de Kofax RPA sur un serveur autonome, veuillez consulter [Déploiement de Tomcat](#).

Remarques pour Windows Docker

Pour la production, nous vous recommandons d'exécuter des conteneurs Docker sous le conteneur de système d'exploitation préféré spécifié dans le tableau ci-dessus. Les images de conteneur Windows ne présentent pas la fonctionnalité de haute disponibilité ou les configurations de vérification de l'état. Si votre configuration Kofax RPA intègre un conteneur Windows Docker qui n'est actuellement obligatoire que pour Document Transformation et Kofax Analytics for RPA, nous vous recommandons de déployer cette configuration dans un cluster Kubernetes hybride qui comprend des nœuds Linux et Windows Server 2019 (ou version ultérieure). Les versions précédentes donnent des images beaucoup plus grandes.

Lors de l'exécution de conteneurs Windows Docker, vous devez prendre les décisions suivantes qui sont spécifiques à ce système d'exploitation:

- **Niveau d'isolement**

Décidez du mode d'isolation d'exécution à utiliser: "process" et "Hyper-V" sont disponibles. La valeur par défaut de Windows Server est « processus », tandis que la valeur par défaut de Windows 10 est « Hyper-V ». Lors de l'utilisation du mode d'isolation « processus », la version de base du conteneur doit correspondre à la version du système d'exploitation sur l'hôte. Si vous exécutez les conteneurs sur un hôte Windows Server 2019, les conteneurs doivent être créés en fonction d'une version servercore `1tsc2019`.

Lors de l'utilisation de la version `1tsc2019` ou version ultérieure de servercore, des composants du système d'exploitation doivent être optimisés et ajoutés à l'image RoboServer pour que cela fonctionne. Vérifiez le fichier Docker pour RoboServer situé dans `docker-win\roboboserver` et notez la partie commentée où le script `dxva2.dll` est copié. Déplacez une copie vers le dossier de construction ou modifiez la commande COPY pour qu'elle pointe vers son emplacement comme indiqué dans le commentaire.

De plus, avant d'exécuter le fichier `docker-compose`, assurez-vous de configurer en conséquence l'argument de génération `SERVERCORE_VERSION` dans le Dockerfile requis. Par exemple, si vous utilisez Windows Server 2019, définissez l'argument sur `1tsc2019`.

- **Version**

Choisissez la version de Windows Server à utiliser. Les images de conteneur de base Windows sont considérablement optimisées à partir de la version Windows Server 2019. L'utilisation des versions précédentes n'est pas recommandée en raison de l'utilisation plus élevée des ressources.

Cependant, certains conteneurs Windows ne sont pas encore disponibles dans les versions 2019, comme l'image du serveur Microsoft SQL. L'image officielle de Microsoft ne prend en charge que les conteneurs Docker Linux. Les autres images fournies par Microsoft ne sont pas encore disponibles en tant que versions 2019. Les options disponibles à suivre sont:

- Exécuter en mode d'isolation « processus » sur un système d'exploitation pour lequel toutes les conditions préalables sont disponibles, tel que Windows Server 2016, et créer sur la base de ces images à un coût supplémentaire en ressources d'exécution.
- Exécuter en mode d'isolation « Hyper-V » avec des paramètres optimaux pour chaque conteneur avec la surcharge de l'exécution de machines virtuelles.
- Déployez des images tierces qui ne sont pas disponibles dans la version optimale (MS SQL Server) sur un autre ordinateur. Dans ce cas, l'image officielle Microsoft (Linux) est recommandée.
- Attendez qu'une version du serveur Microsoft SQL soit disponible en fonction de « servercore: 2019ltsc » ou version ultérieure.
- Attendez que Windows Docker prenne en charge l'exécution des conteneurs Windows et Linux en même temps sur un ordinateur en mode d'isolation « processus ».

Étant donné que l'exécution de tous les composants sur un ordinateur n'est pas destinée à la production, nous vous recommandons d'utiliser un cluster Kubernetes hybride ou une forme différente de configuration hybride si des conteneurs Windows Docker sont nécessaires.

Étapes prérequis pour Kofax Analytics for RPA

Avant d'exécuter le fichier docker-compose avec Kofax Analytics for RPA inclus, procédez comme suit :

1. Dans le Kofax RPA dossier d'installation, accédez au dossier `docker-win\kafrpa`, recherchez et exécutez le script `copy_fonts.ps1` pour copier les polices système Windows dans le dossier `docker-win\kafrpa\Insight\Fonts`.
2. Renommez le regroupement de projets Kofax Analytics for RPA à **kafrpa_bundle.zip** et copiez-le dans le dossier `docker-win\kafrpa\Insight`.
3. Renommez le paquet d'installation Insight **KofaxInsightSetup.msi** et copiez-le dans le dossier `docker-win\kafrpa\Insight`.
4. Obtenez le fichier de licence `Altosoft.Insight.License.xml` requis et copiez-le dans le dossier `docker-win\kafrpa`.

Étapes prérequis pour Document Transformation Service

Avant d'exécuter le fichier docker-compose avec Kofax Analytics for RPA inclus, procédez comme suit :

1. Dans le dossier d'installation Kofax RPA, allez dans `docker-win\compose-examples`, ouvrez le fichier `docker-compose-dt.yml`, puis modifiez la ligne suivante selon le cas.

```
DT_LICENSE_SERVER = mettez-votre-serveur-de-licences-ici.exemple.com #  
Serveur de licences Kofax
```
2. Copiez le modifié `docker-compose-dt.yml` fichier à `docker-win \ documenttransformation`.
3. Copiez les bundles PNL `KofaxTransformation-6.3.1.0_NLP-LanguageBundles` (`KofaxTransformation_Salience6.4_LanguageBundle_extended.msi`, `KofaxTransformation_Salience6.4_LanguageBundle_western-default.msi`, et `KofaxTransformation_Salience6.4_LanguageBundle_western-extended.msi`) à `docker-win \ documenttransformation`.
4. Copiez `KofaxRPA_DocumentTransformation-11.2.0.0.msi` vers `docker-win \documenttransformation`.

Déployez Kofax RPA en utilisant les fichiers docker-compose

Cette rubrique fournit les étapes de base du déploiement Kofax RPA sur un serveur autonome sous un système Linux ou Windows. Pour obtenir plus d'informations sur la composition des exemples de fichiers docker-compose pour Windows et Linux fournis par Kofax RPA, consultez la section suivante.

Pour déployer Kofax RPA à l'aide d'un fichier docker-compose, procédez comme suit :

1. Installez Kofax RPA sur votre ordinateur.
2. Téléchargez Docker depuis <https://www.docker.com> et installez-le sur votre ordinateur.
3. *Applicable uniquement à Linux.* Ajoutez un nouvel utilisateur au groupe Docker. Par exemple, pour ajouter un utilisateur « Kofax » et autoriser l'utilisateur à accéder aux conteneurs Docker, remplacez `docker:x:<n>` avec `docker:x:Kofax` dans le fichier `/etc/group`. Déconnectez-vous et connectez-vous ou redémarrez l'ordinateur pour mettre à jour les privilèges.
4. Dans le dossier `compose-examples`, sélectionnez le [fichier docker-compose le mieux adapté à vos besoins](#). Copiez le fichier du dossier `compose-examples` à la racine du dossier d'installation Kofax RPA, et renommez le fichier comme suit : `docker-compose.yml` si cela est nécessaire.
Par exemple, si vous travaillez sous Windows, copiez le fichier `docker-compose.yml` de `C:\Program Files\Kofax RPA 11.2.0.0\docker-win\compose-examples` dans `C:\Program Files\Kofax RPA 11.2.0.0`.
5. Modifiez le fichier `docker-compose` en fonction de votre environnement et de vos besoins.
Vous pouvez ajouter des données de licence à `docker-compose.yml` dans les variables `CONFIG_LICENSE_`, puis démarrez les services Kofax RPA. À partir du dossier d'installation du

produit, exécutez la commande suivante pour créer une image, démarrer les services et définissez le nom « kapow » pour le conteneur actuel.

```
docker-compose -p kapow up -d --build
```

Remarque Une fois que Management Console est en cours d'exécution, vous ne pouvez pas modifier la licence. Pour modifier les paramètres de licence, arrêtez la composition et mettez à jour les variables de licence dans le fichier docker-compose. Si les paramètres de licence ne sont pas inclus dans le fichier docker-compose, vous pouvez les modifier dans Management Console sans arrêter le conteneur.

Remarque Utilisez le nom et le mot de passe par défaut du super-utilisateur admin pour vous connecter à Management Console :

Nom d'utilisateur : admin

Mot de passe : admin

La première fois que vous créez l'image, sa préparation peut prendre un certain temps.

Vous pouvez utiliser des commandes distinctes pour créer une image et démarrer les services comme suit.

- Pour construire une image.
 - Pour Linux : `docker build -f docker/managementconsole/Dockerfile -t managementconsole:11.2.0.0 .`
 - For Windows : `docker build -f docker-win\managementconsole\Dockerfile -t managementconsole:11.2.0.0 .`

Notez l'espace et le point en fin de ligne. Le point fait référence au répertoire courant.

- Pour démarrer les services.
 - Pour Linux et Windows : `docker-compose -p kapow up -d`

Après avoir exécuté le fichier docker-compose, dès qu'il démarre les conteneurs, vous devriez pouvoir ouvrir votre hôte Docker, qui, par défaut, se trouve dans `http://localhost`, et constater qu'un Management Console fonctionne avec un RoboServer dans un conteneur séparé. De plus, vous devriez pouvoir accéder à d'autres composants Kofax RPA en conteneur inclus dans le fichier docker-compose.

Les sections suivantes fournissent des informations détaillées sur les exemples de docker-compose et les paramètres de configuration.

Exemples de docker-compose

Kofax RPA inclut plusieurs fichiers docker-compose avec quelques configurations simples dans le dossier `docker/compose-exemples`. Pour Windows, les fichiers se trouvent dans `docker-win \ compose-exemples`.

Pour Linux, tous les exemples suivants s'appuient sur MySQL comme base de données de configuration de Management Console. Bien que Management Console puisse fonctionner sur d'autres bases de données, MySQL est recommandé pour les données de configuration Management Console. La documentation des images du docker MySQL est disponible sur https://hub.docker.com/_/mysql/. Pour Windows, les exemples de composition reposent sur le serveur Microsoft SQL comme base de données de configuration.

Pour la journalisation et le stockage des données du robot, l'une des bases de données prises en charge peut être utilisée. Voir « Plateformes prises en charge » dans le *Kofax RPA Guide d'installation*.

Les fichiers docker-compose suivants sont fournis pour un **Environnement Linux**.

docker-compose-basic.yml

Cette configuration démarre une Management Console, un RoboServer, une base de données MySQL, et les connecte. Avant de commencer cette configuration, vous pouvez entrer vos données de licence dans la configuration de composition pour éviter d'avoir à les saisir au démarrage de Management Console. Pour mettre à l'échelle la quantité de RoboServers (dans le cluster « Production »), utilisez la commande suivante.

```
docker-compose -p kapow up -d --scale roboserver-service=2
```

docker-compose-ha.yml

Cette configuration démarre une Management Console, un RoboServer, un MySQL et un load balancer fondé sur l'image Traefik légère.

Management Console est configuré pour s'exécuter avec la Haute disponibilité activée à l'aide de la découverte de multidiffusion (nécessite une licence d'entreprise). Pour plus d'informations, voir [Exécuter sur Docker Swarm avec Management Console en haute disponibilité](#).

Remarque La découverte de multidiffusion nécessite une superposition de réseau qui prend en charge la multidiffusion UDP.

Pour que cette configuration fonctionne de manière optimale, saisissez vos informations de licence dans le fichier docker-compose avant d'activer les services. Modifiez également la ligne suivante pour l'adapter au réseau attribué à vos conteneurs:

```
- CONFIG_CLUSTER_INTERFACE = 172.20.0.*
```

Exécutez les étapes suivantes pour savoir quel réseau est utilisé.

1. Commencez la composition.

```
docker-compose -p kapow up -d
```

2. Répertoriez les conteneurs démarrés.

```
conteneur docker ls
```

3. Utilisez la commande suivante pour obtenir le nom d'hôte.

```
docker exec kapow_managementconsole-service_1 hostname -i
```

4. Arrêtez la composition avant de modifier le fichier docker-compose.

```
docker-compose -p kapow down
```

Des caractères génériques peuvent être utilisés, de sorte que l'adresse IP peut être similaire à 172.*.*.*. Notez que Hazelcast n'accepte pas les caractères génériques au lieu de tous les nombres, et donc *.*.*.* n'est pas autorisé.

Lorsque vous commencez la composition, vous pouvez mettre à l'échelle le nombre d'instances Management Console en cours d'exécution en utilisant la commande suivante.

```
docker-compose -p kapow up -d --scale managementconsole-service=2
```

L'exécution de plus de deux instances de Management Console est possible, mais cela augmente la charge de la base de données. Le load balancer est configuré pour utiliser des sessions persistantes.

docker-compose-jms.yml

Cette configuration démarre une Management Console, un RoboServer et une base de données MySQL qui communiquent via un répartiteur ActiveMQ.

docker-compose-kaplets.yml

Cette configuration démarre une Management Console, un RoboServer, une base de données MySQL, et ajoute également Kofax RPA Kaplets et les configure.

docker-compose-ldap.yml

Ceci est un exemple de configuration qui utilise LDAP. Il commence le `OpenLDAP` commande dans un conteneur, qui n'est normalement pas nécessaire mais est incluse à titre d'exemple.

Un fichier connexe, `ldap_ad_content.ldif`, est également inclus à des fins de test. Testez cette composition en exécutant la commande suivante.

```
docker-compose -p kapow up -d --build && docker -vv cp ./docker/compose-examples/ldap_ad_content.ldif kapow_ldap-service_1:/ldap_ad_content.ldif &&
docker exec kapow_ldap-service_1 ldapadd -x -D "cn=admin,dc=example,dc=org" -w
admin -f /ldap_ad_content.ldif
```

docker-compose-rfs.yml

Cette configuration démarre une Management Console, un RoboServer, une base de données MySQL, et ajoute également un système de fichiers du robot et le configure.

Synchronizer

Synchronizer n'est pas inclus dans les fichiers `docker-compose` pour Linux. Pour construire une image Docker Synchronizer Docker, utilisez la commande suivante :

```
docker build -f docker/synchronizer/Dockerfile . -t
synchronizer:@productVersion@
```

Si vous souhaitez désactiver le contrôle d'état du Docker JMX, supprimez les lignes du fichier Docker entre :

```
# ### commencez la découpe ici ### and # ### terminez la découpe ici ###
```

Les fichiers `docker-compose` suivants sont fournis pour un **environnement Windows**.

docker-compose.yml

Cette configuration démarre tous les composants disponibles de Kofax RPA , à l'exception de Document Transformation : Management Console, RoboServer, Synchronizer, Système de fichiers du robot, Kaplets, Kofax Analytics for RPA, et une base de données MS SQL.

docker-compose-dt.yml

Ce fichier `compose` est utilisé pour documenter les paramètres utilisés par le conteneur Windows Docker de Document Transformation. Si vous avez besoin de Document Transformation, vous pouvez utiliser le fichier `docker-compose-dt.yml` d'exemple comme suit. Avant d'exécuter le dossier, assurez-vous que vous avez suivi les conditions préalables à Document Transformation Service énumérées dans [Remarques pour Windows Docker](#).

```
docker-compose -f .\docker-compose-dt.yml up
```

Optimiser la taille du support de création Docker

Les informations contenues dans cette section ne s'appliquent qu'à Linux.

Comme toutes les images Docker sont créées avec le dossier racine de distribution comme support de création, la création des images peut prendre trop de temps. Pour optimiser la taille du support de création, des fichiers `dockerignore` sont ajoutés à la distribution. Comme chaque composant a des exigences différentes, un fichier `dockerignore` distinct est fourni pour chaque composant. Le fichier doit être situé à côté du `Dockerfile` et suivre la même convention de dénomination, sauf qu'il doit avoir un suffixe `.dockerignore`, tel que `Dockerfile.dockerignore`.

Pour utiliser cette fonctionnalité, la compilation avec `BuildKit` doit être activée. La version standard de Docker ne prend actuellement en charge que le `.dockerfile` global. Pour activer cette fonctionnalité, configurez `DOCKER_BUILDKIT=1` dans l'environnement ou définissez-le dans `/etc/docker/daemon.json`.

Si vous créez avec `docker-compose`, vous devez également définir `COMPOSE_DOCKER_CLI_BUILD=1` dans l'environnement. Si vous ne compilez pas avec `BuildKit`, vous pouvez renommer le fichier `Dockerfile.dockerignore` en `.dockerignore` et le déplacer à la racine de votre support de création. Si vous créez plusieurs composants, comme avec `docker-compose`, vous devez combiner le contenu `.dockerignore`. Toutefois, nous recommandons une compilation avec `BuildKit` car elle est plus efficace.

Utilisez la fonction Docker secrets pour stocker les mots de passe

Pour éviter de spécifier les mots de passe de connexion directement dans le fichier `docker-compose` sous Linux et Windows, vous pouvez utiliser la fonction Docker secrets pour stocker vos mots de passe dans un endroit fiable.

Vous pouvez utiliser la fonction secrets pour les variables d'environnement.

Dans l'exemple suivant, nous spécifions un mot de passe pour se connecter à la base de données MySQL, dans un environnement Linux.

1. Créez un fichier texte contenant un mot de passe. Un fichier doit contenir un mot de passe. Pour cet exemple, nous avons créé un fichier `mysqlpassword.txt` qui comprend un mot de passe pour la base de données MySQL.
2. Dans le fichier `docker-compose` que vous souhaitez utiliser, créez une section appelée `secrets` à la racine (sans indentation), spécifiez un nom de variable au deuxième niveau, et indiquez un chemin d'accès au fichier avec un mot de passe au troisième niveau d'indentation comme suit.

```
secrets:
  mysqlpassword:
    file: /kapow/linux64dist/mysqlpassword.txt
```

3. Dans la section `services > mysql-service > environment` du fichier `.yaml`, remplacez la variable `MYSQL_PASSWORD` par la variable `MYSQL_PASSWORD_FILE` et spécifiez le jeu de variables dans la section `secrets` comme suit :

```
services:
  mysql-service:
    image: mysql:5
    environment:
      - MYSQL_ROOT_PASSWORD=mysqlrootpassword
      - MYSQL_DATABASE=mysqldatabase
      - MYSQL_USER=mysqluser
      - MYSQL_PASSWORD_FILE=/run/secrets/mysqlpassword
    networks:
      - net
```

4. Au même niveau que la section `environnement` sous `services > mysqlservice`, créez une section `secrets` et spécifiez à nouveau la variable `secrets`.

```
secrets:  
- mysqlpassword
```

En utilisant cette procédure comme exemple, vous pouvez configurer la fonction `secrets` pour d'autres mots de passe dans le fichier `docker-compose`.

Configuration de la base de données

Pour Kofax RPA, nous recommandons de stocker votre configuration Management Console et les données de répertoire dans une base de données conteneurisée, et d'ajouter des bases de données externes pour le stockage des données et les journaux (d'audit).

Toutefois, nous vous recommandons de stocker les données de configuration interne de Management Console dans une base de données externe ou d'entreprise. Pour modifier la base de données Management Console, corrigez les variables d'environnement pour le contexte de la base de données et ajoutez le pilote JDBC approprié à l'image ou au conteneur.

Dans le Dockerfile pour Management Console, qui réside dans `docker/managementconsole` sur Linux et dans `docker-win\managementconsole` sur Windows, la ligne suivante ajoute le pilote JDBC actuel à l'image Management Console du dossier JDBC.

Linux : `ADD https://repo1.maven.org/maven2/mysql/mysql-connector-java/<version>/mysql-connector-java-<version>.jar /usr/local/tomcat/lib/jdbc/`

Windows : `AJOUTER https://clojars.org/repo/com/microsoft/sqlserver/sqljdbc4/4.0/sqljdbc4-4.0.jar${CATALINA_HOME}/lib/jdbc/, où CATALINA_HOME=${KAPOW_HOME}\tomcat\apache-tomcat-@tomcatVersion@ et KAPOW_HOME=c:\kapow`

Modifiez cette ligne ou ajoutez d'autres lignes dans le fichier Docker pour ajouter le bon pilote JDBC et la bonne version du connecteur de base de données pour votre Java. Utilisez la dernière version disponible du pilote.

Après avoir modifié le fichier Docker, vous devez reconstruire l'image en exécutant la commande suivante :

Linux : `docker build -f docker/managementconsole/Dockerfile . -t managementconsole:@productVersion@`

Windows : `docker build -f docker-win\managementconsole\Dockerfile . -t managementconsole:@productVersion@`

Ou exécutez la version simple :

```
docker-compose -p kapow up -d --build
```

Établir une connexion SSL à la base de données MySQL

Les informations contenues dans cette section ne s'appliquent qu'à Linux. Utilisez les informations de cette section pour établir une connexion SSL entre une Management Console et une base de données MySQL. Les exemples de `docker-compose` fournis par Kofax comprennent deux services : `mysql` et `managementconsole`. Le service `mysql` est composé sur la base de l'image du `docker mysql`. Le

service `managementconsole` est composé à partir de l'image docker de Tomcat. La procédure suivante explique comment établir une connexion SSL entre les deux services.

Lorsque vous démarrez un fichier docker-compose avec une connexion non sécurisée vers la base de données, certaines versions de Tomcat produisent un avertissement si SSL n'est pas configuré et explicitement désactivé. Si vous ne souhaitez pas établir la connexion SSL à la base de données, vous pouvez désactiver l'avertissement en spécifiant le paramètre `useSSL=false` dans la variable `CONTEXT_RESOURCE_URL` du fichier docker-compose, comme indiqué ici :

```
CONTEXT_RESOURCE_URL=jdbc:mysql://mysql-service:3306/scheduler?
useUnicode=yes&characterEncoding=UTF-8&rewriteBatchedStatements=true&useSSL=false
```

Pour établir une connexion SSL entre un site Management Console et une base de données MySQL, suivez les étapes suivantes :

1. Configurer SSL dans MySQL pour créer des certificats (y compris `ca.pem`) et des clés dans le dossier MySQL (`/var/lib/mysql`) de l'image `mysql` correspondante. Cette étape est nécessaire si vous votre propre certificat n'est pas signé.

MySQL fournit l'utilitaire `mysql_ssl_rsa_setup` qui vous aide à générer toutes les clés et certificats nécessaires. L'image `mysql` pour les fichiers docker-compose fournis par Kofax est configurée pour créer les certificats et les clés auto-signés nécessaires. Lorsque vous lancez un fichier docker-compose qui lance un service MySQL, le fichier `ca.pem` est créé dans le répertoire `/var/lib/mysql` de l'image.

2. Ajouter `ca.pem` au magasin de certificats.

Vous pouvez utiliser l'utilitaire Java `keytool` pour ajouter le certificat au magasin de certificats en utilisant la commande suivante.

```
keytool -importcert -alias MySQLCACert -file /<chemin du certificat>/
ca.pem -keystore /<chemin du magasin de certificats>/truststore -storepass
<mot de passe> -noprompt
```

3. Rendre le magasin de certificats accessible à partir du site Management Console.

Le magasin de certificats doit être accessible depuis le site Management Console sous forme de fichier local. Par exemple, vous pouvez modifier le fichier `Dockerfile.managementconsole` et inclure une commande COPY comme suit :

```
COPY truststore /usr/local/tomcat/
```

Un autre moyen consiste à monter un répertoire avec le fichier `ca.pem` dans l'image `mysql` comme volume de l'image `managementconsole`. Ensuite, lancez l'utilitaire `keytool` à partir du fichier `managementconsole.sh` situé dans le dossier `managementconsole.`).

4. Indiquez le chemin d'accès et le mot de passe du magasin de certificats dans la variable `CONTEXT_RESOURCE_URL` du fichier docker-compose. Par exemple :

```
CONTEXT_RESOURCE_URL=jdbc:mysql://mysql-service:3306/scheduler?
useSSL=true&useUnicode=yes&characterEncoding=UTF-8&rewriteBatchedStatements=
true&trustCertificateKeyStorePassword=password&trustCertificateKeyStoreUrl=
file:/usr/local/tomcat/truststore
```

Sauvegarder et restaurer

Les informations contenues dans cette section ne s'appliquent qu'à Linux. L'image Management Console contient deux scripts pour la sauvegarde et la restauration des informations de répertoire et de configuration.

backup.sh

Pour créer une sauvegarde à enregistrer dans `/kapow/backup`, exécutez la commande Docker suivante :

```
docker exec kapow_managementconsole-service_1 backup.sh [options]
```

Les options d'utilisation de script suivantes sont disponibles :

```
backup.sh [options]
```

Options

Option	Description
<code>-u, --username</code>	<i>Requis.</i> Le nom d'utilisateur à utiliser pour appeler une Management Console.
<code>-p, --password</code>	<i>Requis.</i> Le mot de passe à utiliser pour appeler une Management Console.
<code>-h, --host</code>	Le nom d'hôte pour une Management Console (par défaut : localhost).
<code>-i, --project <Id></code>	ID du projet à utiliser pour soutenir uniquement un projet spécifique.
<code>-c, --configurationOnly <Id></code>	ID pour sauvegarder uniquement la configuration et pas les données du projet.
<code>-n, --postfix</code>	Définit le suffixe pour le fichier de sauvegarde créé (par défaut <datetime>).

restore.sh

Pour restaurer une sauvegarde sauvegardée dans `/kapow/backup`, exécutez la commande docker suivante :

```
docker exec kapow_managementconsole-service_1 restore.sh [options]
```

Les options d'utilisation de script suivantes sont disponibles :

```
restore.sh [options]
```

Options

Option	Description
<code>-u, --username</code>	<i>Requis.</i> Le nom d'utilisateur à utiliser pour appeler une Management Console.
<code>-p, --password</code>	<i>Requis.</i> Le mot de passe à utiliser pour appeler une Management Console.
<code>-h, --host</code>	Le nom d'hôte pour une Management Console (par défaut : localhost).
<code>-f, --filename</code>	<i>Requis.</i> Le nom du fichier de sauvegarde à restaurer.
<code>-a, --path</code>	Le chemin d'accès au fichier de sauvegarde à restaurer (par défaut : <code>/kapow/backup/</code>).

Contrôles préalables au démarrage

Au démarrage d'un conteneur, l'image `ManagementConsole`, par défaut, effectue les vérifications suivantes.

1. Vérifie que la configuration de base de données fonctionne en se connectant à la base de données à l'aide de l'URI JDBC fourni. Il essaie également d'exécuter la requête de validation une fois.
2. Vérifie que la configuration LDAP fonctionne. Si LDAP est configuré, chacun des répertoires LDAP est vérifié en essayant de se connecter et de se lier, et en lançant une requête pour récupérer tous les groupes. Vous pouvez étendre ce test à des fins de débogage ou de validation en ajoutant un nom d'utilisateur de test à utiliser pour d'autres recherches.

Si une vérification échoue, l'image est réessayée pendant une période de temps configurable. Si l'un des contrôles ne réussit pas dans le délai d'attente configuré, le conteneur sort et vous pouvez contrôler vos paramètres de configuration.

Vous pouvez contourner un contrôle en fixant à zéro (0) le délai d'attente pour le contrôle. Voir la section [Variables d'environnement](#) pour obtenir plus d'informations.

Dossiers de données

Les journaux, les données de la base de données et la configuration des conteneurs sont stockés dans les dossiers énumérés ci-après. Pour éviter de faire grandir votre conteneur, ces dossiers doivent être montés sur un volume ou sur le système de fichiers local. Voir la documentation Docker sur les volumes.

Remarque Les journaux, la base de données et les autres données stockées sur ces volumes ne peuvent pas être réutilisés lors de la mise à niveau de Kofax RPA à la version supérieur.

RoboServer

Les données, les journaux et la configuration du conteneur RoboServer se trouvent dans le dossier suivant :

Linux : `/kapow/data`

Windows : `C:\kapow\data`

ManagementConsole

Les journaux Tomcat se trouvent dans le dossier suivant :

Linux : `/usr/local/tomcat/logs`

Windows : `C:\kapow\tomcat\apache-tomcat-@tomcatVersion@\logs`

Variables d'environnement

Vous pouvez trouver la liste des variables d'environnement pour les différents conteneurs Docker par rapport aux fichiers `docker-compose` dans le fichier `readme.md`, qui se trouve dans le dossier `docker` à l'intérieur de votre installation Kofax RPA.

Exécuter sur Docker Swarm avec Management Console en haute disponibilité

Remarque La procédure et la mise en place suivantes ne sont données qu'à titre d'exemple et ne doivent pas être considérées comme une recommandation.

L'exemple de cette section montre comment configurer Kofax RPA sur un essaim de Dockers avec plus d'une instance de Management Console (avec le mode [Haute disponibilité](#) activé).

Cette section contient un exemple de mise en place d'essaim de Dockers composé de deux nœuds, le directeur et le travailleur, qui offrent un niveau de tolérance aux pannes plus élevé que l'utilisation d'un seul nœud. L'exemple utilise une seule instance de MySQL avec un volume jetable.

Le Management Console peut s'exécuter en plusieurs instances dans un cluster, partageant les données de configuration, de journal et de répertoire par le biais d'une base de données, et partageant l'état volatile du cluster par le biais de la plateforme Hazelcast. Cette plateforme exige que chaque instance du Management Console puisse découvrir et se connecter aux autres instances du cluster. Les deux méthodes de découverte qui sont actuellement mises en œuvre sont la multidiffusion et le TCP. Dans cet exemple de procédure, nous utilisons la méthode de découverte du multicast (UDP). La méthode de découverte du TCP a également été testée avec succès.

Pour faire fonctionner l'UDP multicast dans un essaim de Docker, le plug-in Weave Net 2.5.1 est utilisé dans cette procédure.

Mettre en place un essaim minimal de Docker avec Management Console

Avant de continuer, assurez-vous que deux hôtes Docker s'exécutent avec le noyau Linux 3.8 ou supérieur. Dans cette procédure, les hôtes sont appelés `host1` et `host2`.

1. Installez le cluster d'essaim du Docker.

a. Créer un nœud de gestion sur `host1`.

```
host1$ docker swarm init --advertise-addr <host1_IP>
```

b. Rejoignez `host2` dans l'essaim en tant que nœud travailleur.

```
host2$ docker swarm join --token <token> --advertise-addr <host2_IP>
<host1_IP>:<port>
```

Remplacez `<token>` par le jeton récupéré lors de la première commande.

2. Installez le plug-in Weave Net sur les deux hôtes et activez la fonction multicast comme suit.

```
host1$ docker plugin install store/weaveworks/net-plugin:2.5.2 --grant-all-
permissions --disable
host1$ docker plugin set store/weaveworks/net-plugin:2.5.2 WEAVE_MULTICAST=1
host1$ docker plugin enable store/weaveworks/net-plugin:2.5.2
```

```
host2$ docker plugin install store/weaveworks/net-plugin:2.5.2 --grant-all-
permissions --disable
host2$ docker plugin set store/weaveworks/net-plugin:2.5.2 WEAVE_MULTICAST=1
host2$ docker plugin enable store/weaveworks/net-plugin:2.5.2
```

3. Sur votre nœud de gestion, créez un réseau Docker en utilisant le plug-in Weave Net en tant que pilote, comme suit.

```
host1$ docker network create --driver=store/weaveworks/net-plugin:2.5.2 weavenet
```


4. Construisez les images de Docker Management Console et RoboServer sur tous les nœuds de votre essaim Docker. Avec un répertoire Docker, vous pouvez y envoyer les images. Pour obtenir plus d'informations sur la création des images du Docker, voir [Outils Docker pour déploiement Kofax RPA](#).
5. Créez un fichier appelé **docker-compose.yml**, en adaptant les lignes suivantes à votre environnement.

```

version: '3.2'
networks:
  net:
    external:
      name: weavenet
services:
  loadbalancer:
    image: traefik
    command: --docker --docker.swarmmode --docker.watch --web --loglevel=DEBUG
    ports:
      - 80:80
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    networks:
      - net
    deploy:
      mode: global
      placement:
        constraints: [node.role == manager]
  mysql-service:
    image: mysql:5
    environment:
      - MYSQL_ROOT_PASSWORD=mysqlrootpassword
      - MYSQL_DATABASE=scheduler
      - MYSQL_USER=scheduler
      - MYSQL_PASSWORD=schedulerpassword
    networks:
      - net
  managementconsole-service:
    image: managementconsole:@productVersion@
    depends_on:
      - mysql-service
    environment:
      - CONTEXT_RESOURCE_VALIDATIONQUERY=SELECT 1
      - CONTEXT_RESOURCE_USERNAME=scheduler
      - CONTEXT_RESOURCE_PASSWORD=schedulerpassword
      - CONTEXT_RESOURCE_DRIVERCLASSNAME=com.mysql.jdbc.Driver
      - CONTEXT_RESOURCE_URL=jdbc:mysql://mysql-service:3306/scheduler?
useUnicode=yes&characterEncoding=UTF-8&rewriteBatchedStatements=true
      # enter your license here, or type it through the GUI in first login
      - CONFIG_LICENSE_NAME=
      - CONFIG_LICENSE_EMAIL=
      - CONFIG_LICENSE_COMPANY=@licenseCompany@
      - CONFIG_LICENSE_PRODUCTIONKEY=@licenseProduction@
      - CONFIG_LICENSE_NONPRODUCTIONKEY=@licenseNonProduction@
      - CONFIG_CLUSTER_JOINCONFIG=multicastCluster
      # change to fit your network
      - CONFIG_CLUSTER_INTERFACE=10.*.*.*
      - CONFIG_CLUSTER_MULTICAST_GROUP=224.2.2.3
      - CONFIG_CLUSTER_MULTICAST_PORT=54327
      - LOG4J_LOGGER_COM_HAZELCAST=ERROR, A
    deploy:
      replicas: 2
      labels:
        traefik.docker.network: weavenet

```

```

traefik.port: 8080
traefik.backend.loadbalancer.sticky: "true"
traefik.frontend.rule: PathPrefix:/
networks:
- net
roboserver-service:
image: roboserver:@productVersion@
depends_on:
- mysql-service
networks:
- net
environment:
- ROBOSERVER_ENABLE_MC_REGISTRATION=true
- ROBOSERVER_MC_URL=http://loadbalancer/
- ROBOSERVER_MC_CLUSTER=Production
- ROBOSERVER_MC_USERNAME=admin
- ROBOSERVER_MC_PASSWORD=admin
- ROBOSERVER_ENABLE_SOCKET_SERVICE=true
- WRAPPER_MAX_MEMORY=2048

```

Remarque Si vous copiez le fichier, veuillez à conserver la mise en page originale. Une mise en page incorrecte peut entraîner un fichier non valide.

À noter également :

- Remplacez **@productVersion@**, **@licenseCompany**, **@licenseProduction@**, et **@licenseNonProduction@** par les valeurs appropriées.
 - **CONFIG_CLUSTER_INTERFACE** doit correspondre au sous-réseau Weave Net dans votre essaim de Docker. Vous pouvez trouver le sous-réseau en utilisant la commande suivante sur votre nœud de gestion : le réseau de dockers `host1$ inspecte weavenet`
 - Le **réseau** commun fait référence au réseau externe **weavenet** que vous avez créé auparavant.
 - **Traefik** est utilisé comme load balancer avec des sessions persistantes.
 - Cette installation fait tourner une seule instance de MySQL sur un volume temporaire. Dans une véritable configuration de production, la base de données doit également être exécutée en cluster et avec un volume de données permanent.
 - Pour exécuter plusieurs instances du site RoboServer, vous pouvez ajouter des contraintes de déploiement au **service roboserver**.
 - Tous les services doivent utiliser **endpoint_mode: dnsrr** (le réglage par défaut **endpoint_mode: vip** n'est pas pris en charge par le plug-in Weave Net)
6. Déployez la pile de service sur le nœud de gestion comme suit.

```
host1$ docker stack deploy -c docker-compose.yml rpa
```

Conseil Démarrer un **service managementconsole** avec deux répliques sur une base de données vide peut conduire à une condition de course. Si vous rencontrez cette situation, changez les **répliques** des lignes : **1** et exécutez la commande précédente. Attendez que les services soient démarrés, changez de ligne, puis exécutez à nouveau la commande précédente.

Pour arrêter les services, utilisez la commande suivante.

```
host1$ docker stack down rpa
```

7. Après avoir déployé la pile, le Management Console peut être consulté à l'URL suivante.

```
http://host1/
```

Configuration avancée

Intégration Single Sign-On LDAP et CA

Cette rubrique décrit comment utiliser l'authentification Single Sign-On LDAP et CA.

En outre, Kofax RPA prend en charge LDAPS (LDAP sur SSL). Voir « LDAPS sécurisé » et « Liste de contrôle pour résoudre les erreurs de connexion SSL lors de l'utilisation de LDAPS » plus loin dans cette section.

Intégration de LDAP

Pour authentifier avec LDAP, activez l'authentification LDAP et modifiez la définition LDAP dans le fichier `login.xml`.

Pour activer l'authentification LDAP, réglez la propriété `useLdap` sur `true` comme suit :

```
<bean id="authenticationConfiguration"
  class="com.kapowtech.scheduler.server.spring.security.AuthenticationConfiguration">
  <property name="useLdap" value="true"/>
  <property name="useSiteMinder" value="false"/>
</bean>
```

Dans `login.xml`, vous pouvez trouver la définition suivante :

```
<bean id="ldapDirectories" class="com.kapowtech.mc.config.LdapDirectories" lazy-
init="true">
  <property name="directories">
    <list>
      <!-- List of security groups which will be application administrators.
      Users in these groups will have all permissions. Only users in
these groups can
      access the backup tab and create and restore backups -->
      <bean class="com.kapowtech.mc.config.LdapDirectory">

        <property name="adminGroups">
          <list>
            <value>KAPOWADMIN</value>
            <value>ENGINEERING</value>

          </list>
        </property>
        <property name="administratorGroups">
          <list>
            <value>RPAADMINISTRATORS</value>
          </list>
        </property>
        <property name="ldapServerURL" value="ldap://
ldap.kapowdemo.com:389"/>
        <property name="userDn" value="CN=LDAP
test,CN=Users,DC=kapowdemo,DC=local"/>
        <property name="password" value="change-me"/>
        <property name="userSearchBase"
value="OU=Users,OU=TheEnterprise,DC=kapowdemo,DC=local"/>
        <property name="userSearchFilter"
value="(userPrincipalName={0}@kapowdemo.local)"/>
        <property name="userSearchSubtree" value="true"/>
      </bean>
    </list>
  </property>
</bean>
```

```

        <property name="groupSearchBase" value="OU=Security
Groups,OU=TheEnterprise,DC=kapowdemo,DC=local"/>
        <property name="groupSearchFilter" value="(member={0})"/>
        <property name="groupRoleAttribute" value="cn"/>
        <property name="groupSearchSubtree" value="true"/>
        <property name="allGroupsFilter" value="(cn=*)"/>
        <property name="fullNameAttribute" value="displayName"/>
        <property name="emailAttribute" value="userPrincipalName"/>
        <property name="referral" value="follow"/>
    </bean>
</list>
</property>
</bean>

```

Ceci définit une liste de beans `ldapDirectory` appelée `ldap` et représente une liste de connexions aux serveurs LDAP. Kofax RPA prend en charge l'intégration LDAP multi-forêts, vous pouvez donc spécifier plusieurs connexions aux répertoires LDAP. Chaque bean définit un certain nombre de propriétés qui contrôlent l'intégration LDAP. Si vous connaissez la façon dont Tomcat s'intègre à LDAP, cela devrait également vous être familier.

Important Les noms de groupe doivent être uniques pour tous les serveurs LDAP de la liste.

LDAP sécurisé

Kofax RPA Prend en charge LDAPS (LDAP sur SSL) avec Management Console. Pour utiliser LDAPS, la propriété `ldapServerURL` doit être définie comme suit :

```
<property name="ldapServerURL" value="ldaps://<hostname>:<port>"/>
```

Par défaut, le port LDAPS est le 636.

Liste de contrôle pour le déploiement

Propriété	Description
<code>adminGroups</code>	Liste des groupes LDAP mis en correspondance avec le super-utilisateur de l'administration sur Management Console qui a accès à tout.
<code>administratorGroups</code>	Liste des groupes LDAP mis en correspondance avec les administrateurs RPA sur Management Console. Les utilisateurs appartenant à ces groupes LDAP disposent de tous les droits pour tous les projets (à l'exception des droits spéciaux des utilisateurs admin), comme la mise en correspondance des groupes LDAP avec les rôles dans Management Console.
<code>ldapServerURL</code>	URL du serveur LDAP. Elle utilise soit le protocole <code>ldap://</code> soit le protocole <code>ldaps://</code> .
<code>userDn</code>	DN (nom distingué) utilisé pour se connecter à LDAP afin d'authentifier d'autres utilisateurs.
Mot de passe	Mot de passe pour le compte userDN. Comme le mot de passe sera stocké en texte clair dans ce fichier, vous devez utiliser un compte qui n'a qu'un accès en lecture.
<code>userSearchBase</code>	Sous-répertoire dans l'arborescence LDAP où les utilisateurs peuvent être trouvés.
<code>userSearchFilter</code>	Filtre qui est appliqué pour trouver le nom d'utilisateur.
<code>userSearchSubtree</code>	Définir sur « vrai » si les utilisateurs peuvent être localisés dans le sous-répertoire de la base de données <code>userSearchBase</code> .
<code>groupSearchBase</code>	Sous-répertoire dans l'arborescence LDAP où les groupes peuvent être trouvés.

Propriété	Description
groupSearchFilter	Filtre appliqué pour identifier les utilisateurs de ce groupe.
groupRoleAttribute	Attribut qui contient le nom du groupe.
groupSearchSubtree	Mettre à « vrai » si les groupes peuvent être situés dans le sous-répertoire de groupeSearchBase
convertToUpperCase	Si les noms de groupes sont convertis en majuscules, vrai par défaut.
allGroupsFilter	Facultatif. Contrôle les groupes qui sont affichés lors de la création des autorisations de projet, voir ci-dessous.
fullNameAttribute	Attribut pour récupérer le nom complet de l'utilisateur.
emailAttribute	Attribut permettant de récupérer l'e-mail de l'utilisateur.
référence	Définir sur « suivre » pour permettre la redirection vers les sous-nœuds de l'arborescence LDAP.

Pour utiliser un compte LDAP et administrer une Management Console, vous devez ajouter l'un des groupes dont vous êtes membre au bean `administratorGroups` dans `login.xml`, comme décrit dans les [Autorisations de projets](#). Notez que toute personne membre d'un groupe figurant dans `administratorGroups` est l'administrateur de Management Console, vous pouvez donc créer un nouveau groupe LDAP à cet effet. Utilisez le nom de groupe en majuscules si `convertToUpperCase` est vrai.

Lorsque vous sélectionnez une autorisation de projet sur la Management Console, vous pouvez voir que tous les noms de groupes sont extraits du LDAP pour remplir la liste. Les groupes sont localisés en utilisant le `groupRoleAttribute` pour construire un filtre permettant de récupérer tous les groupes. Parfois, vous n'avez pas besoin que tous les groupes LDAP soient affichés ici, auquel cas vous pouvez passer outre ce comportement en proposant votre propre filtre. Cela se fait en ajoutant une propriété supplémentaire au `LdapLogin`.

`<property name="allGroupsFilter" value="(cn=*)" />`: Trouve tous les noms de groupes, si le nom du groupe est dans l'attribut `cn` (c'est la valeur par défaut).

Si vous souhaitez uniquement trouver les groupes commençant par la lettre « e », vous pouvez utiliser le code suivant

```
<property name="allGroupsFilter" value="(cn=E*)" />
```

Le filtre utilise des requêtes LDAP de base. Voir la documentation LDAP pour les requêtes plus complexes.

Liste de contrôle pour résoudre les erreurs de connexion SSL lors de l'utilisation de LDAPS

Si vous rencontrez des erreurs de connexion en utilisant LDAPS, vérifiez ce qui suit :

- LDAPS exige que le certificat présenté par le serveur LDAP soit approuvé par le java exécutant le tomcat. Importez le certificat public dans le magasin de clés Java que votre application utilise.
- Assurez-vous que les certificats sont importés dans le bon magasin de certificats, par exemple si vous avez plusieurs instances de JRE ou JDK.
- Assurez-vous que le bon magasin de certificats est utilisé. Si `-Djavax.net.ssl.trustStore` est configuré, il remplace l'emplacement du magasin de certificats par défaut.

- Si vous vous connectez à un serveur de messagerie, tel qu'Exchange, assurez-vous que l'authentification permet le texte brut.
- Vérifiez que le serveur cible est configuré pour servir le SSL correctement. Cela peut être fait avec l'outil de test du serveur SSL.
- Vérifiez si votre outil anti-virus dispose d'un « Scanning SSL » qui bloque le SSL et le TLS. Désactivez cette fonction ou définissez des exceptions pour les adresses cibles.

Intégration de Single Sign-On CA

Management Console prend en charge la pré-authentification en utilisant Single Sign-On CA. Avec Single Sign-On CA, l'identité de l'utilisateur est établie avant d'accéder à une Management Console, et l'identité de l'utilisateur est communiquée via une en-tête HTTP. L'identité de l'utilisateur doit apparaître sous la forme d'un nom distinctif LDAP pour une Management Console afin de résoudre les adhésions de l'utilisateur à un groupe LDAP.

L'intégration de Single Sign-On CA est désactivée par défaut. Vous pouvez l'activer en définissant la propriété `useSiteMinder` à `true` dans le fichier `login.xml` comme suit :

```
<bean
class="com.kapowtech.scheduler.server.spring.security.AuthenticationConfiguration"
id="authenticationConfiguration">
  <property name="useLdap" value="false"/>
  <property name="useSiteMinder" value="true"/>
</bean>
```

```
<bean class="com.kapowtech.mc.config.SiteMinderConfiguration"
id="siteMinderConfiguration">
  <property name="headerName" value="sm_userdn"/>
  <property name="accountAttribute" value="sAMAccountName"/>
  <property name="accountAttributePattern" value="(.*)" />
</bean>
```

Après avoir activé l'intégration de Single Sign-On CA, indiquez le nom de l'en-tête HTTP qui contient le nom distinctif de l'utilisateur. La propriété `accountAttribute` identifie lequel des attributs LDAP de l'utilisateur est utilisé comme nom de compte (la valeur par défaut est `sAMAccountName`, qui est le login Windows de l'utilisateur). La propriété `accountAttributePattern` précise comment le nom du compte est analysé à partir de la valeur de l'attribut, qui doit être une expression régulière (avec un seul ensemble de parenthèses identifiant le nom du compte), et la valeur `(.*)` dans la propriété `accountAttributePattern` signifie tout ce qui se trouve dans l'attribut. Pour extraire le nom de compte de l'adresse e-mail de l'utilisateur dans la configuration, vous pouvez spécifier ce qui suit :

```
<bean class="com.kapowtech.mc.config.SiteMinderConfiguration"
id="siteMinderConfiguration">
  <property name="headerName" value="sm_userdn"/>
  <property name="accountAttribute" value="userPrincipalName"/>
  <property name="accountAttributePattern" value="([^\@]*)@.*" />
</bean>
```

`([^\@]*)@.*` correspondra à une adresse e-mail et extraira tout ce qui précède `@` comme le nom de compte.

Comme Single Sign-On CA utilise une partie de la configuration de connexion LDAP, vous devez ajouter un groupe d'utilisateurs au bean `administratorGroups` avant de pouvoir commencer à configurer la Management Console.

Il n'est pas possible de se déconnecter du système, car la présence de l'en-tête Single Sign-On CA signifie que vous êtes toujours authentifié. Pour vous déconnecter, fermez votre navigateur.

Limitations

L'intégration de Single Sign-On CA ne fonctionne que lorsqu'une Management Console est accessible via un navigateur. Cependant, si une Management Console est accessible par des applications qui ne sont pas des navigateurs, le mécanisme d'authentification Single Sign-On CA n'est pas utilisé et ces services nécessitent un ensemble d'identifiants (nom d'utilisateur et mot de passe) définis dans le Management Console. Ces clients comprennent, entre autres, les personnes suivantes :

Design Studio

Les développeurs de robots doivent accéder à Management Console pour obtenir une licence de siège de développeur, pour télécharger des robots et obtenir les configurations de base de données et autres paramètres stockés dans la Management Console. Pour cela, il faut accéder aux URL suivantes (par rapport au chemin de contexte où Management Console est déployé).

- /License/*
- /secure/*
- /IDESettings/*
- /rest/*
- /ws/*

L'accès aux URL suivantes est protégé par une authentification de base avec des mots de passe définis dans la Management Console.

Services REST

Les services REST sont protégés par une authentification de base par défaut, mais les robots peuvent être exposés sans authentification en tant que services REST. Ces services sont généralement invoqués par des applications externes. REST utilise l'URL /rest/*.

RoboServer

RoboServer est protégé par une authentification de base pour l'inscription à un cluster et pour les requêtes de ressources.

Service Desktop Automation

Le service Desktop Automation est protégé par une authentification de base pour l'enregistrement et les mises à jour de l'état.

Toute application basée sur Kofax RPA Java ou .Net API

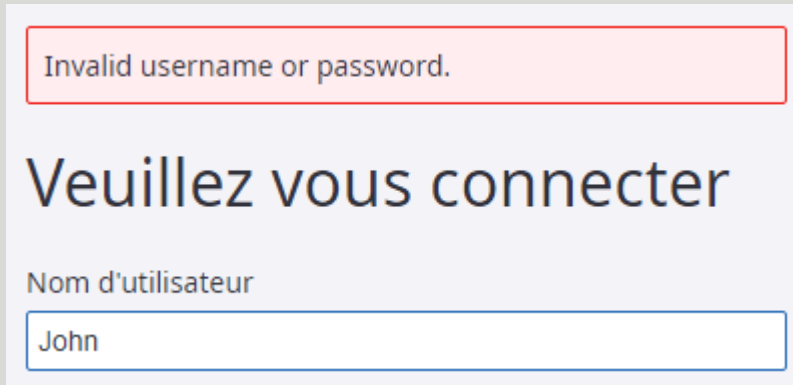
Toute application basée sur l'API Kofax RPA Java ou .Net est protégée par une authentification de base lors de l'accès à une Management Console.

Exemple : Utilisation

- **Jane** est l'administrateur désigné de Management Console. Comme elle est ajoutée au groupe d'**Administrateurs** dans l'Active Directory (AD), ce groupe est mentionné dans **login.xml**, et une fois qu'elle authentifie son navigateur avec Single Sign-On CA, elle est immédiatement connectée à l'adresse Management Console.



- **John** est un développeur de robots, il est membre du groupe **Users** de l'AD et actuellement il ne parvient pas à se connecter à l'adresse Management Console.



Invalid username or password.

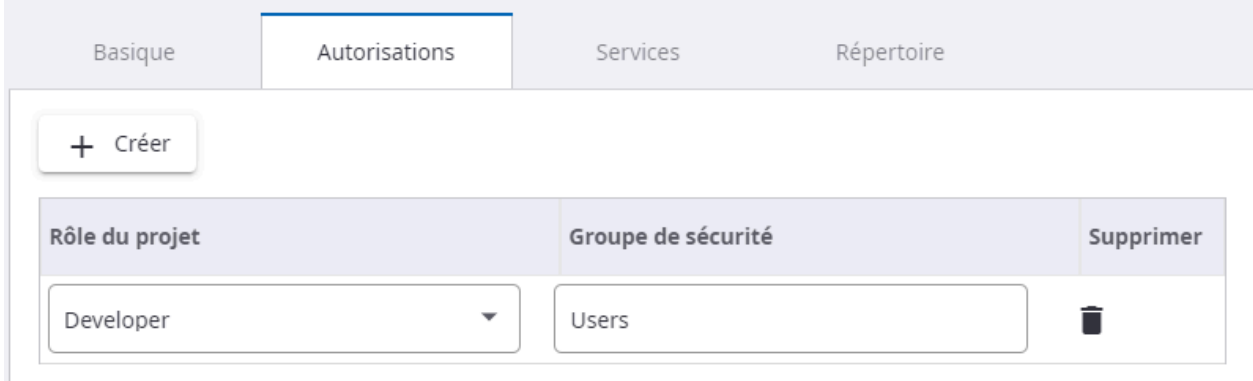
Veillez vous connecter

Nom d'utilisateur

John


Jane doit attribuer des privilèges au groupe **Utilisateurs** dans Management Console > **Administration** > **Projets** pour le projet sélectionné avant que John puisse se connecter. Par exemple, le groupe Utilisateurs peut se voir attribuer un rôle de développeur comme indiqué ici.

Modifier le projet Default project



Basique **Autorisations** Services Répertoire

+ Créer

Rôle du projet	Groupe de sécurité	Supprimer
Developer	Users	

John peut maintenant se connecter au site Management Console en utilisant Single Sign-On CA.



Lorsque John démarre Design Studio, il doit saisir ses identifiants pour obtenir une licence.

Jane peut également créer des utilisateurs de services locaux qui ne sont jamais autorisés à se connecter au site Management Console, comme pour le service RoboServer ou Desktop Automation Service. Notez ce qui suit :

- Les groupes que vous sélectionnez pour les utilisateurs du service doivent provenir uniquement d'Active Directory.
- Le groupe doit disposer des privilèges appropriés définis dans la Management Console.

Intégration de Single Sign-On SAML

Management Console prend en charge la pré-authentification des utilisateurs à l'aide de Single Sign-On SAML.

La procédure suivante est écrite en supposant que vous avez déjà créé et configuré une application SAML dans votre fournisseur d'identité. Pour un exemple de procédure de configuration, voir [Exemple de configuration OneLogin](#).

Remarque Management Console avec l'intégration SAML ne peut pas démarrer sans licence. Installez une licence valide avant de pouvoir utiliser l'application.

L'intégration du site Management Console avec Single Sign-On SAML est désactivée par défaut. Pour l'activer, dans le fichier `login.xml`, localisez et réglez la propriété `useSaml` sur `true`.

```
<bean
class="com.kapowtech.scheduler.server.spring.security.AuthenticationConfiguration"
id="authenticationConfiguration">
  <property name="useLdap" value="false"/>
  <property name="useSiteMinder" value="false"/>
  <property name="useWebSphere" value="false"/>
  <property name="useSaml" value="true"/>
</bean>
```

Après avoir activé l'intégration dans `login.xml`, vous devez configurer le fournisseur d'identité et les paramètres du fournisseur de services. Dans le fichier `saml.xml` situé dans le dossier `\WEB-INF\spring`, modifiez les propriétés suivantes pour qu'elles correspondent à votre configuration.

Après avoir modifié les fichiers `login.xml` et `saml.xml`, redémarrez Tomcat.

Propriété	Description
<code>forceAuthN</code>	Lorsqu'il est défini sur <code>vrai</code> , le fournisseur d'identité réauthentifie les utilisateurs et ne s'appuie pas sur les événements d'authentification précédents. Par défaut, il est défini sur <code>faux</code> .
<code>groupsAttributes</code>	Attribut de groupe. L'accès des utilisateurs au Management Console est géré par le biais de groupes auxquels un utilisateur particulier appartient. Les noms des groupes définis dans le fichier <code>saml.xml</code> doivent correspondre à ceux définis dans votre application SAML. Remarque Après avoir défini les groupes d'utilisateurs dans votre fournisseur d'identité, vous devez ajouter manuellement ces utilisateurs aux groupes dans le Management Console.
<code>e-mail</code>	Adresse e-mail de l'utilisateur Management Console.
<code>prénom</code>	Première partie du nom complet de l'utilisateur Management Console.
<code>nom de famille</code>	Deuxième partie du nom complet de l'utilisateur Management Console.

Propriété	Description
<code>idpName</code>	Nom du fournisseur d'identité. Les valeurs possibles sont OKTA, ONELOGIN et AZURE. Sinon, définissez-le sur DEFAULT.
<code>idpGroupNameSeparator</code>	Délimiteur à utiliser pour séparer les noms des groupes de fournisseurs d'identité. Ne prend effet que si le fournisseur d'identité spécifié avec <code>idpName</code> est OneLogin. Sinon, cette propriété est ignorée.
<code>entityId</code>	URL d'un Management Console, plus <code>saml/login</code> . Vous pouvez obtenir cette URL à partir de votre application SAML. Par exemple : <code>http://localhost:8080/ManagementConsole/saml/login</code>
<code>entityBaseURL</code>	URL de base d'un Management Console. Par exemple : <code>http://localhost:8080/ManagementConsole</code>
<code>maxAuthenticationAge</code>	Validité de Single Sign-On en quelques secondes. Fenêtre de temps accordée aux utilisateurs pour se connecter après leur authentification initiale auprès du fournisseur d'identité. Par défaut, elle est de 86 400 secondes, soit 24 heures. Utilisez cette propriété pour modifier la valeur par défaut.
<code>responseSkew</code>	Tolérance d'imprécision en secondes pour la comparaison des horloges entre le serveur du fournisseur d'identité et l'ordinateur où est déployé Management Console. Comme la synchronisation de l'horloge peut ne pas être entièrement précise, une tolérance de 60 secondes est appliquée par défaut. Utilisez cette propriété pour modifier la valeur par défaut.
<code>maxAssertionTime</code>	Validité des assertions traitées lors de Single Sign-On. Si le temps d'assertion atteint la limite configurée, l'authentification devient invalide. Par défaut, elle est limitée à 6000 secondes, soit 100 minutes. Utilisez cette propriété pour modifier la valeur par défaut.
<code>java.lang.String</code> du bean <code>HTTPMetadataProvider</code>	URL vers les métadonnées du fournisseur d'identité. Vous pouvez obtenir cette URL à partir de votre application SAML. Par exemple : <code>http://example.okta.com/saml/metadata/222670a0-2b96-48ef-975a-b7267446d09e</code> Certains fournisseurs d'identité, comme Microsoft Azure, n'exigent pas cette propriété.
<code>java.io.File</code> du bean <code>FilesystemMetadataProvider</code>	Chemin d'accès au fichier XML contenant les métadonnées du fournisseur d'identité. Utilisez cette propriété si votre fournisseur d'identité ne permet pas la lecture des métadonnées en temps réel. Certains fournisseurs d'identité, tels que OneLogin, n'exigent pas cette propriété.

Les snippets suivants du fichier `saml.xml` d'exemple contiennent les propriétés que vous devez configurer.

```
<bean id="samlEntryPoint" class="org.springframework.security.saml.SAMLEntryPoint"
  lazy-init="true">
  <property name="defaultProfileOptions">
    <bean class="org.springframework.security.saml.websso.WebSSOProfileOptions">
      <property name="includeScoping" value="false"/>
      <property name="forceAuthN" value="false"/>
      <property name="passive" value="false"/>
    </bean>
  </property>
  <property name="filterProcessesUrl" value="/saml/entry"/>
</bean>
```

```
<bean id="samlAuthenticationProvider" class="com.kapowtech.scheduler.server.spring.sec
  urity.GroupProvidingSAMLAuthenticationProvider" lazy-init="true"> <constructor-arg ref
  ="platformEMF"/>
  <property name="internalAuthenticationProvider"
  ref="internalAuthenticationProvider"/>
  <property name="customerNameMapper" value="false"/>
  <property name="customerNameMapperIdentifier" value=""/>
  <property name="groupsAttributes">
    <list>
      <value>FirstUserGroup</value>
      <value>SecondUserGroup</value>
      <value>groups</value>
    </list>
  </property>
  <property name="adminGroups">
    <list>
      <value>KapowAdmins</value>
    </list>
  </property>
  <property name="createUsers" value="true"/>
  <property name="assignGroups" value="true"/>
  <property name="consumer" ref="webSSOprofileConsumer"/>

  <property name="email" value="email"/>
  <property name="firstname" value="firstname"/>
  <property name="lastname" value="lastname"/>
  <property name="idpName" value="ONELOGIN"/>
  <property name="idpGroupNameSeparator" value=";"/>
  <property name="idpUserNameRegex" value="^[a-zA-Z0-9]*$"/>
  <property name="idpEmailRegex" value="^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$"/>
</bean>
```

```
<bean id="metadataGeneratorFilter"
  class="org.springframework.security.saml.metadata.MetadataGeneratorFilter">
  <constructor-arg>
    <bean class="org.springframework.security.saml.metadata.MetadataGenerator">
      <property name="entityId" value="http://localhost:8080/ManagementConsole/
  saml/login"/>
      <property name="requestSigned" value="false"/>
      <property name="entityBaseURL" value="http://localhost:8080/
  ManagementConsole"/>
      <property name="extendedMetadata">
        <bean
  class="org.springframework.security.saml.metadata.ExtendedMetadata">
          <property name="idpDiscoveryEnabled" value="false"/>
          <property name="signMetadata" value="false"/>
        </bean>
      </property>
    </bean>
  </constructor-arg>
</bean>
```

```

<bean id="webSSOprofileConsumer"
  class="org.springframework.security.saml.websso.WebSSOProfileConsumerImpl">
  <property name="maxAuthenticationAge" value="86400"/>
  <property name="responseSkew" value="600"/>
  <property name="maxAssertionTime" value="6000"/>
</bean>

<bean id="metadata"
  class="org.springframework.security.saml.metadata.CachingMetadataManager">
  <constructor-arg>
  <list>
    <bean class="org.opensaml.saml2.metadata.provider.HTTPMetadataProvider"
      lazy-init="true">
      <constructor-arg>
        <value type="java.lang.String">http://example.okta.com/saml/
metadata/222670a0-2b96-48ef-975a-b7267446d09e</value>
      </constructor-arg>
      <constructor-arg>
        <value type="int">10000</value>
      </constructor-arg>
      <property name="parserPool" ref="parserPool"/>
    </bean>

    <bean
      class="org.opensaml.saml2.metadata.provider.FilesystemMetadataProvider">
      <constructor-arg>
        <value type="java.io.File">classpath:security/idp.xml</value>
      </constructor-arg>
      <property name="parserPool" ref="parserPool"/>
    </bean>
  </list>
</constructor-arg>
</bean>

```

Exemple de configuration OneLogin

Cette section fournit un exemple de procédure sur la façon de configurer une application SAML dans le fournisseur d'identité OneLogin pour une utilisation avec Kofax RPA.

1. Sur le [site web OneLogin](#), créez un compte.
2. Lorsque le compte est créé, ouvrez la page {votre-nom-de-domaine}.onelogin.com et ouvrez la page Administration.
3. Dans le menu, cliquez sur **APPS** et sélectionnez l'option pour ajouter une nouvelle application. Sélectionnez le type de demande suivant : **Connecteur de test SAML (avancé)**.
4. Dans l'onglet **Configuration**, définissez les paramètres de l'application comme indiqué dans ce tableau. Laissez les autres paramètres tels quels.

Paramètre	Valeur
RelayState	http://<adresse IP>:8080/ManagementConsole/saml/login
Audience	http://<adresse IP>:8080/ManagementConsole/saml/login
Bénéficiaire	http://<adresse IP>:8080/ManagementConsole/saml/login

Paramètre	Valeur
ACS (Consumer) URL Validator	http://<adresse IP>:8080/ManagementConsole/saml/login
URL ACS (Consommateur)	http://<adresse IP>:8080/ManagementConsole/
URL de déconnexion unique	http://<adresse IP>:8080/ManagementConsole/saml/logout
URL de connexion	http://<adresse IP>:8080/ManagementConsole/saml/login
Initiateur du SAML	OneLogin
format nameID	E-mail
type d'émetteur	Spécifique
élément de signature	Réponse
méthode de chiffrement	TRIPLEDES-CBC

5. Dans l'onglet **Paramètres**, définissez les paramètres de l'application comme indiqué dans ce tableau.

Paramètre	Valeur
NameID	Partie du nom de l'e-mail
E-mail	E-mail
prénom	Prénom
groupe	Rôles des utilisateurs
nom de famille	Nom de famille

Sélectionnez également l'option **Inclure dans l'assertion SAML**.

6. L'onglet **SSO** contient l'URL de l'émetteur nécessaire pour configurer le fichier Kofax RPA `saml.xml`.

Assurez-vous que le **certificat X.509** est défini sur le **certificat de force standard (2018 bits)** et que l'**algorithme de signature SAML** est défini sur **SHA-1**.

Copiez la valeur de propriété de l'**URL de l'émetteur** et utilisez-la dans la section « Configuration des métadonnées IDP » de votre fichier `saml.xml`. Pour un exemple, voir ci-dessous.

7. Dans le menu, cliquez sur **UTILISATEURS > Rôles** et sélectionnez l'option pour ajouter un nouveau rôle.

Ajoutez les rôles qui correspondent à vos groupes Management Console et attribuez leur l'application créée auparavant. Le rôle de **KapowAdmins** est obligatoire sur Kofax RPA, alors assurez-vous de l'ajouter.

8. Dans le menu, cliquez sur **UTILISATEURS > Tous les utilisateurs** et sélectionnez les rôles requis pour l'utilisateur.

Vous avez maintenant configuré une application OneLogin.

9. Vous devez maintenant configurer les attributs de groupe dans `saml.xml` situé dans le dossier `\WEB-INF\spring` pour qu'ils correspondent à l'application OneLogin que vous avez créée. Après avoir modifié le fichier, redémarrez Tomcat.

Exemple

```

    <property name="groupsAttributes">
      <list>
        <value>group</value>
      </list>
    </property>
    <property name="adminGroups">
      <list>
        <value>KapowAdmins</value>
      </list>
    </property>
    ...
<property name="idpName" value="ONELOGIN"/>
    ...

    <bean id="metadataGeneratorFilter"
      class="org.springframework.security.saml.metadata.MetadataGeneratorFilter" lazy-
      init="true">
      <constructor-arg>
        <bean
          class="org.springframework.security.saml.metadata.MetadataGenerator">
            <property name="entityId" value="http://<IP_address>:8080/
ManagementConsole/saml/login"/>
            <property name="requestSigned" value="false"/>
            <property name="entityBaseURL" value="http://<IP_address>:8080/
ManagementConsole"/>
            <property name="extendedMetadata">
              <bean
                class="org.springframework.security.saml.metadata.ExtendedMetadata">
                  <property name="idpDiscoveryEnabled" value="false"/>
                  <property name="signMetadata" value="false"/>
                </bean>
              </property>
            </bean>
          </constructor-arg>
        </bean>
      </bean>
    ...

    <!-- IDP Metadata configuration - paths to metadata of IDPs in circle of trust
is provided here
-->
    <bean
      class="org.springframework.security.saml.metadata.CachingMetadataManager"
      id="metadata" lazy-init="true">
      <constructor-arg>
        <list>
          <!-- <bean
            class="org.opensaml.saml2.metadata.provider.FilesystemMetadataProvider">
              <constructor-arg>
                <value type="java.io.File">classpath:security/idp.xml</
value>
              </constructor-arg>

```

```

        <property name="parserPool" ref="parserPool"/>
    </bean> -->
    <bean
class="org.opensaml.saml2.metadata.provider.HTTPMetadataProvider" lazy-
init="true">
        <constructor-arg>
            <value type="java.lang.String">https://
app.onelogin.com/saml/metadata/d237b5c5-b110-4f42-a646-7678ae08feae</value>
        </constructor-arg>
        <constructor-arg>
            <value type="int">10000</value>
        </constructor-arg>
        <property name="parserPool" ref="parserPool"/>
    </bean>
<!-- <bean
class="org.opensaml.saml2.metadata.provider.FilesystemMetadataProvider">
    <constructor-arg>
        <value type="java.io.File">classpath:security/idp.xml</
value>
    </constructor-arg>
    <property name="parserPool" ref="parserPool"/>
</bean> -->
</list>
</constructor-arg>
</bean>

```

Configurer le mode JMS

Pour activer le mode JMS sur Kofax RPA, suivez les étapes suivantes.

Remarque L'exécution de JMS n'est pas prise en charge lors de l'exécution en mode **Haute disponibilité** : ni lors de l'exécution Management Console en mode embarqué.

Les options de **Distribution des licences** et de la **Version seuil RoboServer** dans **Administration > RoboServers > Paramètres > Général** sont désactivées si Management Console est exécuté en mode JMS. La distribution des licences doit être définie sur Statique et Seuil, et la version RoboServer doit être définie sur 11.2.0.

1. Installez un répartiteur, tel qu'Apache ActiveMQ (voir <http://activemq.apache.org> pour obtenir de plus amples informations). Une fois le répartiteur installé, paramétrez JAVA_HOME pour qu'il utilise le dernier JDK et exécutez `activemq.bat`. Au lieu de définir manuellement JAVA_HOME et de lancer `activemq.bat`, vous pouvez créer le fichier `.bat` dans `... \apache-activemq-5.13.3 \bin` et l'exécuter. Le contenu du fichier `.bat` peut être le suivant.

```
set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_25\
```

```
activemq.bat start
```

Important Si vous souhaitez configurer l'accès pour les clients Kofax RPA utilisant le répartiteur ActiveMQ, configurez les utilisateurs comme décrit dans la documentation ActiveMQ et assurez-vous que l'utilisateur a les droits d'administrateur pour créer et utiliser toutes les destinations préfixées avec l'espace de nom utilisé par les clients (l'espace de nom par défaut est « Kapow »). De plus, l'utilisateur doit disposer de droits d'administrateur pour utiliser des destinations temporaires. L'utilisateur et le mot de passe doivent être configurés dans le fichier Management Console `configuration.xml` et soit sur la ligne de commande RoboServer soit en utilisant l'application [Paramètres du RoboServer](#) dans l'onglet Général.

Le courtier ActiveMQ conserve tous les messages de Kofax RPA. Assurez-vous que le répartiteur dispose de ressources suffisantes pour stocker les messages.

2. Installez Management Console sur Tomcat (voir [Tomcat Management Console](#)).

Modifiez `...\\webapps\\ManagementConsole\\WEB-INF>Configuration.xml` and set `<property name="jmsEnabled" value="true"/>`.

3. Modifiez les lignes suivantes

- Dans `<CATALINA_HOME>/webapps/ManagementConsole/WEB-INF`
`<property name="brokerUri" value="failover://(tcp://localhost:61616)? startupMaxReconnectAttempts=3"/>`
- Dans la ligne de connexion RoboServer
`-brokerUrl failover://(tcp://localhost:61616)? startupMaxReconnectAttempts=3`

Voir [Options du RoboServer en mode JMS](#) pour obtenir des informations sur les paramètres disponibles.

4. Configurez les RoboServers pour utiliser les services JMS au lieu des services Socket, par exemple en remplaçant `"-service socket:50000"` par `"-service jms:1"`. Pour chaque RoboServer, configurez l'URL du courtier en utilisant l'option `"-brokerUrl"` et spécifiez le cluster auquel il appartient. RoboServer La configuration peut être effectuée soit en ligne de commande, soit à l'aide de l'application [Paramètres du RoboServer](#).

Démarrez RoboServer. Par exemple, `RoboServer.exe -mcUrl http://admin:admin@localhost:8080/ManagementConsole -cluster Production -service jms:1 -brokerUrl failover://(tcp://localhost:61616)? startupMaxReconnectAttempts=3`

Remarque À partir de Kofax RPA version 10, tous les RoboServers doivent s'enregistrer automatiquement sur Management Console. Par conséquent, l'URL et les identifiants d'un Management Console ainsi que le nom de cluster doivent être spécifiés au démarrage du RoboServer (soit en ligne de commande comme dans l'exemple précédent, soit en utilisant l'application [Paramètres du RoboServer](#)).

5. Lancez un Management Console et supprimez tout RoboServer basé sur Socket des configurations de cluster dans la section **Administration > RoboServers**.

L'exécution simultanée de robots en mode JMS et en mode socket n'est pas prise en charge. Dans la configuration JMS, tous les RoboServers, les Management Consoles et les autres clients doivent fonctionner en mode JMS.

Utiliser MySQL dans ActiveMQ

Par défaut, ActiveMQ utilise KahaDB. Il est nécessaire de modifier les paramètres pour utiliser une base de données professionnelles. À ce sujet, nous utilisons MySQL comme exemple.

1. Localisez le fichier `conf\activemq.xml` dans votre dossier d'installation ActiveMQ.
 - Ajouter un bean de source de données MySQL.

Remarque Selon la version d'ActiveMQ, la source de données peut se trouver dans le paquet `dbcp` ou `dbcp2`.

- Commenter l'adaptateur de persistance pour le KahaDB.
 - Ajouter un adaptateur de persistance pour MySQL.
2. Ajouter le pilote de base de données MySQL (`mysql-connector-java-<version>-bin.jar`) au sous-répertoire `lib` du répertoire d'installation d'ActiveMQ. Remplacer `<version>` avec la version correcte du connecteur MySQL pour votre Java, telle que `mysql-connectorjava-8.0.16.jar`. Utilisez la dernière version disponible du pilote. Pour plus d'informations, voir <https://repo1.maven.org/maven2/mysql/mysql-connector-java/>.
 3. Créez un schéma ActiveMQ dans votre base de données MySQL.

Exemple de code pour les beans.

```
<beans ...>
  ...
</bean>

  <!-- MySql DataSource Sample Setup -->
  <bean id="mysql-ds" class="org.apache.commons.dbcp2.BasicDataSource" destroy-
method="close">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
    <property name="url" value="jdbc:mysql://localhost:3306/activemq"/>
    <property name="username" value="root"/>
    <property name="password" value="root"/>
    <property name="maxActive" value="200"/>
    <property name="poolPreparedStatements" value="true"/>
  </bean>

  <broker ...>
    ...

    <!-- Comment out the default storage adapter
    <persistenceAdapter>
      <kahaDB directory="{activemq.data}/kahadb"/>
    </persistenceAdapter>
    -->

    <persistenceAdapter>
      <jdbcPersistenceAdapter dataSource="#mysql-ds" />
    </persistenceAdapter>

    ...
  </broker>
  ...
</beans>
```

Exemple de code source de données MySQL.

```
<!-- MySql DataSource Sample Setup -->
<bean id="mysql-ds" class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close">
  <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
  <property name="url" value="jdbc:mysql://localhost:3306/activemq"/>
  <property name="username" value="root"/>
  <property name="password" value="root"/>
  <property name="maxActive" value="200"/>
  <property name="poolPreparedStatements" value="true"/>
</bean>
```

Configuration multi-répartiteurs

Vous pouvez configurer plusieurs répartiteurs en mode JMS pour augmenter la stabilité et permettre le basculement.

Remarque Le mode JMS n'est pas pris en charge en mode [haute disponibilité](#).

Effectuez les étapes suivantes pour configurer l'utilisation multi-répartiteurs. Dans cet exemple, nous utilisons MySQL comme base de données.

1. Installez ActiveMQ sur deux ordinateurs. Dans cet exemple, ils sont host1 et host2.
2. Ajoutez les fichiers suivants dans le sous-répertoire `lib` de chaque répertoire d'installation d'ActiveMQ.
 - `commons-dbc-1.4.jar` (voir <https://commons.apache.org/proper/commons-dbc/> pour obtenir plus de détails.)
 - `commons-pool-1.6.jar` (voir <https://commons.apache.org/proper/commons-pool/> pour obtenir plus de détails.)
3. Modifier le fichier `activemq.xml` sur chaque ordinateur ActiveMQ pour utiliser la base de données MySQL comme décrit dans [Utiliser MySQL dans ActiveMQ](#).
4. Modifiez la propriété `brokerUri` dans `<CATALINA_HOME>/webapps/ManagementConsole/WEB-INF/Configuration.xml` pour utiliser les deux hôtes ActiveMQ comme suit.

```
<property name="brokerUri" value="failover://(tcp://host1:61616,tcp://host2:61616)?startupMaxReconnectAttempts=3"/>
```
5. Démarrez les deux hôtes ActiveMQ avant de démarrer les RoboServers.
6. Spécifiez les deux hôtes ActiveMQ dans la chaîne de connexion RoboServer comme suit.

```
-brokerUrl failover://(tcp://host1:61616,tcp://host2:61616)?
startupMaxReconnectAttempts=3
```

Haute disponibilité :

Si une haute disponibilité (basculement) est requise, vous pouvez configurer plusieurs instances de Management Console pour qu'elles fonctionnent ensemble en tant que cluster. Les éléments suivants doivent être regroupés pour parvenir à un basculement complet.

Composantes des clusters

Composant	Description
Load balancer	<p>Un load balancer HTTP est nécessaire pour répartir les requêtes entre plusieurs serveurs Tomcat.</p> <p>Important Lors de la configuration du mode haute disponibilité, tous les autres services, tels que les RoboServers et les Kapplets, doivent être configurés pour accéder via l'équilibreur de charge au lieu de la connexion directe à Management Console.</p>
Base de données de la plateforme en cluster	<p>Le site Management Console stocke les planifications, les robots et autres dans la base de données de la plateforme. Dans un scénario de basculement, la base de données de la plateforme devrait s'exécuter sur un SGBD en cluster pour éviter un point de défaillance unique.</p>
Reproduction de la session Tomcat	<p>Bien que le site Management Console ne stocke aucune donnée directement dans la session de l'utilisateur (sauf pendant l'importation/exportation), la session contient les informations d'authentification de l'utilisateur.</p> <p>Si la réplication de session n'est pas activée, l'utilisateur devra se reconnecter si le Tomcat auquel il est actuellement connecté plante.</p>
Connecteur Apache Tomcat	<p>mod_jk est un module Apache utilisé pour connecter le conteneur de servlets Tomcat à des serveurs web tels qu'Apache, iPlanet, Sun ONE (anciennement Netscape) et même IIS en utilisant le protocole Apache JServ.</p>
Hazelcast	<p>Hazelcast (www.hazelcast.com) est utilisé pour regrouper des structures de données sur plusieurs JVM. Dans la Management Console, elle est utilisée pour regrouper les structures de données vitales et pour assurer l'intercommunication entre les instances d'application.</p> <p>Voici un exemple : Lorsque vous utilisez un robot sur un site RoboServer, un thread est nécessaire pour traiter les messages d'état renvoyés par le site RoboServer. Ce thread se déroule à l'intérieur d'une instance spécifique de Tomcat. Dans un environnement en cluster, un utilisateur qui tente d'arrêter le robot peut en fait générer la requête d'arrêt sur une autre instance Tomcat que celle qui exécute le robot. Dans ce cas, la requête d'arrêt est diffusée par Hazelcast à toutes les instances et l'instance qui exécute le robot la reçoit et agit pour arrêter le robot.</p>

Multiplés instances Management Console

Vous devez avoir deux ou plusieurs installations Tomcat identiques, et déployer la même version de `ManagementConsole.war` sur chacune d'entre elles. Assurez-vous que les fichiers `web.xml`, `Configuration.xml`, `login.xml` et `roles.xml` sont identiques dans toutes les instances.

Installer et configurer les composants

Cette rubrique décrit comment installer et configurer les composants nécessaires à la configuration de haute disponibilité en utilisant la mise en cluster de la diffusion multicast. Dans cette configuration, nous allons mettre en place deux ordinateurs hôtes : un hôte (hôte1) contient le serveur Tomcat et une base de données, un autre hôte (hôte2) contient le serveur Tomcat et le serveur Apache comme load balancer.

Procédure étape par étape

La procédure suivante vous aide à installer les composants pour une configuration à haute disponibilité.

1. Mettre en place une base de données sur l'ordinateur « host1 ».
2. Téléchargez Tomcat sur le site web d'Apache <https://tomcat.apache.org>
3. Installez Tomcat sur les deux hôtes et définissez le mot de passe de l'utilisateur. Voir [Déploiement de Tomcat](#) pour plus d'informations.
4. **Installez Management Console** sur Tomcat sur les deux hôtes.
5. Lancez l'application Tomcat sur les deux ordinateurs et assurez-vous qu'ils sont en ligne. Vous devez avoir deux installations Tomcat identiques, et déployer la même version de ManagementConsole.war sur chacune d'entre elles. Assurez-vous que les fichiers `web.xml`, `Configuration.xml`, `login.xml` et `roles.xml` sont identiques sur les deux installations.
6. Fermez les serveurs Tomcat.
7. Téléchargez le serveur Apache à partir du site web Apache : <http://httpd.apache.org/download.cgi#apache24>. Installez le serveur Apache sur « host2 » et démarrez le service Apache. Voir le document Apache pour obtenir plus d'informations : <http://httpd.apache.org/docs/>
8. Téléchargez le connecteur Apache `mod_jk` sur le site web d'Apache : <https://tomcat.apache.org/download-connectors.cgi>.
 - Décompressez les fichiers dans un répertoire de votre disque.
 - Copiez le fichier `mod_jk.so` dans le répertoire `<host2>\module` répertoire.
 - Modifier `<host2>\conf\httpd.conf` comme suit :

```
LoadModule jk_module modules/mod_jk.so
<IfModule mod_jk.c>
  JkWorkersFile "<apache>\conf\workers.properties"
  JkLogFile "<apache>\logs\mod_jk.log"
  JkLogLevel error
  JkLogStampFormat "[%a %b @d %H:%M:%S %Y] "
  JkRequestLogFormat "%w %V %T"
</IfModule>
JkMount /ManagementConsole/* loadbalancer
JkMount /ManagementConsole loadbalancer
```

- Créer un `<host2>\conf\workers.properties` avec le contenu suivant :

```
worker.list=host1, host2, loadbalancer

worker.host1.host=<ip address or host name>
worker.host1.port=8009
worker.host1.type=ajp13
worker.host1.lbfactor=1
worker.host2.host=<ip address or host name>
worker.host2.port=8009
worker.host2.type=ajp13
worker.host2.lbfactor=1
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=host1, host2
```

Où `<adresse IP ou nom d'hôte>` est l'adresse ou le nom d'hôte des ordinateurs hôtes qui exécutent Tomcat.

9. Pour les deux serveurs Tomcat, saisissez les lignes suivantes dans le fichier `conf\server.xml`.

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
```

```

address="0.0.0.0"
port="8009"
redirectPort="8443"
secretRequired="false" />

```

Pour plus d'informations, voir la [reproduction des sessions Tomcat](#).

10. Sur chaque serveur Tomcat, modifiez le fichier `webapps\ManagementConsole\WEB-INF\Configuration.xml` comme suit. Notez que vous devez spécifier les adresses IP valides des ordinateurs hôtes de votre réseau.

```

<!-- Cluster configuration -->
<bean id="cluster" class="com.kapowtech.mc.config.ClusterConfig" >
<property name="port" value="5701"/>
<property name="interface" value="<mask for the IP address>"/>
<!-- Uncomment the line below to enable clustering via multicast. Your license
must support High Availability for this to work -->
<!--property name="joinConfig" ref="multicastCluster"/>
<!-- or uncomment this line to enable clustering via TCP-IP. Your license must
support High Availability for this to work-->
<property name="joinConfig" ref="tcpCluster"/-->
<property name="managementCenterUrl" value=""/>
</bean>

<!-- definition for a TCP cluster. You need to add peers to this list, so each
client can locate at least one other functioning cluster member -->
<bean id="tcpCluster" class="com.kapowtech.mc.config.TcpJoinConfig" lazy-
init="true">
<property name="peers">
<list>
<bean class="com.kapowtech.mc.config.TcpPeer">
<property name="host" value="<ip address or host name>"/>
<!-- port is only needed if the other machine is not using the same port as this
instance-->
<!--property name="port" value="5701"/-->
</bean>
<bean class="com.kapowtech.mc.config.TcpPeer">
<property name="host" value="<ip address or host name>"/>
<!-- port is only needed if the other machine is not using the same port as this
instance-->
<!--property name="port" value="5701"/-->
</bean>
</list>
</property>
</bean>

```

Pour plus d'informations, voir [Replication Hazelcast](#).

Vous pouvez désormais vous connecter au site Management Console sur le load balancer en allant sur `<host2>:80/ManagementConsole` où « `host2` » est le nom de l'ordinateur qui fait tourner le serveur Apache. Une fois connecté, allez dans **Administration > Nœuds à haute disponibilité** et vous devriez voir deux nœuds avec des adresses IP correctes.

Démarrage du load balancer

Cette section décrit comment déterminer si la demande a démarré correctement.

Si les fichiers `ManagementConsole.xml` (configuration du contexte) ou `web.xml` sont invalides, l'application ne peut pas être déployée sur Tomcat, et les requêtes renvoient normalement le code d'erreur 404 (car elle touche l'application ROOT de Tomcat qui n'a rien de déployé sur /`ManagementConsole/`).

Toute autre erreur rencontrée au démarrage de l'application est indiquée à l'utilisateur lors du chargement de l'application. De cette façon, vous ne devez pas toujours vérifier le journal pour savoir pourquoi l'application ne s'est pas chargée correctement. Cela n'est cependant pas très pratique car l'application renvoie 200 OK, même s'il y a des erreurs au démarrage. De plus, si l'authentification est activée, vous devez vous connecter avant de pouvoir voir les messages d'erreur.

Pour permettre aux load balancers de voir plus facilement si l'application a démarré correctement, vous pouvez faire une requête à l'URL `/ManagementConsole/Ping`. Cela renvoie soit le code d'état HTTP 200 si l'application s'est chargée correctement, soit 500 avec une trace de l'erreur dans la pile.

Reproduction de la session Tomcat

La réplication des sessions est configurée dans `/conf/server.xml`. Voici un exemple qui utilise le multicast, par exemple la découverte sur Tomcat.

```
<Cluster className="org.apache.catalina.cluster.tcp.SimpleTcpCluster"
  managerClassName="org.apache.catalina.cluster.session.DeltaManager"
  expireSessionsOnShutdown="false"
  useDirtyFlag="true"
  notifyListenersOnReplication="true"
  printToScreen="true">

  <Membership
    className="org.apache.catalina.cluster.mcast.McastService"
    mcastAddr="228.0.0.4"
    mcastPort="45564"
    mcastFrequency="500"
    mcastDropTime="3000"/>

  <Receiver
    className="org.apache.catalina.cluster.tcp.ReplicationListener"
    tcpListenAddress="auto"
    tcpListenPort="4002"
    tcpSelectorTimeout="100"
    tcpThreadCount="6"/>

  <Sender
    className="org.apache.catalina.cluster.tcp.ReplicationTransmitter"
    replicationMode="pooled"
    ackTimeout="150000"
    waitForAck="true"/>

  <Valve className="org.apache.catalina.cluster.tcp.ReplicationValve"
    filter=".*\.(gif|.*\.(js|.*\.(jpg|.*\.(png|.*\.(htm|.*\.(html|.*\.(css|.*
\.(txt;"/>

  <Deployer className="org.apache.catalina.cluster.deploy.FarmWarDeployer"
    tempDir="/tmp/war-temp/"
    deployDir="/tmp/war-deploy/"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>

  <ClusterListener
    className="org.apache.catalina.cluster.session.ClusterSessionListener"/>
</Cluster>
```

Vous devez également définir l'attribut `jvmRoute` sur l'élément `<Engine>` dans `server.xml` :

```
<Engine jvmRoute="tomcat2" name="Catalina" defaultHost="MyHost">
```

Remarque Si vous utilisez `mod_jk` comme load balancer pour les pauvres, la valeur de `jvmRoute` doit correspondre au nom indiqué dans les références du fichier `workers.properties` par la configuration `mod_jk`.

Consultez votre documentation Tomcat pour obtenir plus de détails.

Replication Hazelcast

Les paramètres Hazelcast les plus basiques peuvent être édités dans `Configuration.xml`, tandis que les paramètres plus avancés tels que le chiffrement SSL doivent être configurés dans `/WEB-INF/Hazelcast.xml`

Lorsqu'un Management Console démarre, il crée un nœud Hazelcast sur le port 5701 (ou le prochain port disponible si le 5701 est indisponible). Par défaut, ce nœud Hazelcast est lié à l'adresse IP 127.0.0.1. Vous devez changer l'adresse de liaison en un nom d'IP/hôte public avant qu'il puisse participer à un cluster. Cela se fait en modifiant la propriété de l'interface du cluster bean dans `Configuration.xml`. Cela pourrait ressembler à cela :

```
<bean id="cluster" class="com.kapowtech.mc.config.ClusterConfig" >
  <property name="port" value="26000"/>
  <property name="interface" value="10.0.0.*"/>
  .....
</bean>
```

L'astérisque (*) est utilisé comme joker, dans ce cas l'application essaiera de se lier à la première interface qui a une adresse IP commençant par 10.0.0. Il est possible, mais non recommandé, d'utiliser `*.*.*` car vous risquez de vous lier à 127.0.0.1, ou à une autre interface virtuelle.

Lorsque vous démarrez des instances supplémentaires de Management Console, leurs instances Hazelcast tenteront de trouver tout nœud Hazelcast existant et de rejoindre le cluster. Cette découverte peut se faire par multidiffusion ou par TCP/IP.

Pour utiliser la découverte multicast, vous devez modifier le cluster bean dans `Configuration.xml`. C'est ce que je fais en décommentant la ligne suivante :

```
<property name="joinConfig" ref="multicastCluster"/>
```

`multicastCluster` est une référence au bean `multicastCluster`, qui définit le groupe et le port de multicast. Vous pouvez le modifier pour l'adapter à la topologie de votre réseau.

Si votre réseau ne permet pas la multidiffusion, vous devrez utiliser le `tcpCluster`. Pour ce faire, il faut plutôt décommenter cette ligne :

```
<property name="joinConfig" ref="tcpCluster"/>
```

Le bean `tcpCluster` contient une liste de `TcpPeer`, un pour chaque autre nœud Hazelcast. Si vous utilisez le même port TCP pour tous les nœuds Hazelcast, vous n'avez pas besoin de spécifier un numéro de port (chaque nœud supposera que ses pairs fonctionnent sur le même port que lui). Si vous avez deux nœuds configurés dans un cluster TCP, cela pourrait ressembler à ceci :

```
<bean id="tcpCluster" class="com.kapowtech.mc.config.TcpJoinConfig">
  <property name="peers">
    <list>
```

```
<bean class="com.kapowtech.mc.config.TcpPeer">
  <property name="host" value="10.0.0.25"/>
</bean>
<bean class="com.kapowtech.mc.config.TcpPeer">
  <property name="host" value="10.0.0.26"/>
</bean>
</list>
</property>
</bean>
```

Remarquez que les deux nœuds sont dans la liste. Cela signifie que quel que soit le nœud qui démarre en premier, il sera en mesure de trouver son homologue. Il vous permet également d'utiliser des fichiers `Configuration.xml` identiques dans les deux applications. De plus, les numéros de ports TCP ne sont pas définis, de sorte que chaque pair essaiera de se connecter à l'autre sur le même port car il s'écoute lui-même.

Nœuds d'application

Vous pouvez vérifier que la demande est correctement regroupée en allant dans **Administration > Nœuds à haute disponibilité**.

La colonne **Interface** indique l'IP/hôte et le port que Hazelcast utilise pour la communication entre les clusters. La colonne « **Connecté à** » indique à quel nœud vous êtes actuellement connecté. Si vous arrêtez le serveur auquel vous êtes actuellement connecté, vous serez automatiquement redirigé vers une autre instance en direct par le load balancer.

Dans le menu contextuel d'un nœud, vous pouvez demander le vidage d'un thread, ce qui peut être utile pour le débogage.

Encodage URI

Si vous prévoyez de télécharger des robots dont les noms contiennent des caractères non ASCII, comme le ÆØÅ danois ou le ß allemand, vous devez configurer l'encodage URI de votre conteneur web en UTF-8.

Sur Tomcat, cela se fait sur la définition `<connector>` qui se trouve dans le fichier `server.xml`, à l'intérieur du dossier `/conf`. Ici, vous ajoutez l'attribut `URIEncoding="UTF-8"` comme ceci :

```
<Connector port="8080" URIEncoding="UTF-8"...../>
```

Chiffrement du mot de passe

Depuis la version 8.2, Management Console utilise un chiffrement par certificat (clé publique, clé privée) pour le stockage des mots de passe. Lorsque vous importez une version précédente, le mot de passe est automatiquement re-chiffré à l'aide du nouvel algorithme par certificat.

Le certificat et la clé privée correspondante sont stockés dans un magasin de clés Java, Management Console est livré avec un magasin de clés qui contient un certificat et une clé privée par défaut. Comme tous les clients ont le même magasin de clés, nous vous recommandons de créer votre propre magasin de clés, sinon n'importe qui pourra charger vos exportations et éventuellement obtenir vos mots de passe.

Créez votre propre magasin de clés

Si vous avez déjà commencé un Management Console, vous devez mettre à jour le certificat. La boutique doit être au format pkcs12, et peut être créée à l'aide de l'application keytool fournie avec le SDK Java (qui peut être téléchargé sur Oracle.com, actuellement disponible ici). La commande suivante crée un nouveau magasin de clés pkcs12 avec un certificat valide pendant 365 jours.

```
keytool -genkey -alias mc -keyalg RSA -validity 3650 -keystore mc.p12 -storetype pkcs12
```

On vous demandera votre mot de passe et les informations qui seront stockées dans la clé privée X.509. La commande va créer un fichier mc.p12 (la valeur de l'argument -keystore) dans le répertoire courant. -validity 3650 signifie que le certificat sera valide pendant 10 ans.

Remarque Nous ne recommandons pas d'utiliser un certificat délivré par une autorité de certification (CA) car pkcs12 détient à la fois la clé privée et le certificat public, et le mot de passe de la clé privée sera écrit en texte clair dans le cadre de la configuration de l'application.

Pour demander à Management Console d'utiliser le nouveau certificat, modifiez le fichier `Configuration.xml`. Le fichier se trouve dans l'archive web `ManagementConsole.war`, qui doit être décompressée. Voir la section Déploiement dans Tomcat pour obtenir de plus amples informations. Dans `Configuration.xml`, vous trouverez l'entrée suivante :

```
<bean id="keyStore" class="com.kapowtech.mc.config.KeyStoreConfig" >
  <property name="location" value="/WEB-INF/mc.p12"/>
  <property name="password" value="changeit"/>
  <property name="alias" value="mc"/>
</bean>
```

Vous devez indiquer ici l'emplacement, le mot de passe et l'alias du magasin de clés. Si vous copiez le magasin de clés dans `ManagementConsole.war`, l'emplacement doit être relatif à la racine de l'application. Si vous voulez faire référence à un magasin de clés stocké dans le système de fichiers, l'emplacement doit commencer par `file://`, et doit être une référence absolue à l'emplacement du magasin de clés.

Mise à niveau du magasin de clés

La première fois que Management Console démarre, il crée une somme de contrôle en utilisant la clé privée du magasin de clés, ce qui lui permet de détecter quand le magasin de clés a été remplacé, et de vérifier que les mots de passe peuvent effectivement être déchiffrés avec le certificat fourni. Si vous avez déjà lancé un Management Console avant d'installer votre propre magasin de clés, vous devez le configurer pour effectuer une conversion de mot de passe.

Important La mise à jour du magasin de clés n'est disponible que pour le magasin de clés Management Console qui stocke les mots de passe dans les objets de saisie de la planification. Il ne peut pas être utilisé pour mettre à jour les mots de passe stockés dans le magasin des mots de passe ou tout autre mot de passe chiffré.

Pour mettre à niveau le magasin de clés, copiez le fichier actuel du magasin de clés dans un nouvel emplacement, comme le dossier personnel de votre utilisateur, puis modifiez `Configuration.xml` pour créer un convertisseur de mot de passe avec une référence à l'ancien magasin de clés :

```
<bean id="oldKeyStore" class="com.kapowtech.mc.config.KeyStoreConfig" >
  <property name="location" value="file:///home/roboserver/mc.p12"/>
  <property name="password" value="changeit"/>
  <property name="alias" value="mc"/>
</bean>

<bean id="passwordConverter"
class="com.kapowtech.scheduler.server.service.PasswordConverter">
  <constructor-arg ref="oldKeyStore"/>
</bean>
```

Cette fonction configure un convertisseur de mot de passe pour utiliser le certificat précédent afin de déchiffrer les mots de passe et la somme de contrôle existants (vous devrez fournir l'emplacement, l'alias et le mot de passe corrects pour l'ancien magasin de clés), et utiliser la nouvelle clé privée (telle que configurée ci-dessus) pour re-chiffrer les mots de passe et créer une nouvelle somme de contrôle. La conversion a lieu lors du prochain lancement du site Management Console, la conversion a lieu pendant le lancement de la demande et peut prendre un certain temps s'il y a plusieurs planifications. Il n'est pas nécessaire de supprimer les beans `OldKeyStore` et `PasswordConverter` du fichier `Configuration.xml`, car la conversion du mot de passe n'est déclenchée que lorsque la somme de contrôle et le magasin de clés ne sont pas synchronisés, et après la conversion, la somme de contrôle correspond au nouveau magasin de clés).

Vérification des terminaux SSL

Lorsque vous créez un nouveau cluster, vous pouvez choisir de chiffrer la communication avec les RoboServers au moyen du protocole SSL, ce qui empêche toute personne d'« écouter » le réseau et d'extraire des informations critiques échangées entre les deux parties.

En plus du chiffrement, le protocole SSL permet la validation des terminaux. Cela permet de s'assurer que vous n'échangez pas d'informations critiques avec un tiers, soit en raison d'une mauvaise configuration, soit parce que votre DNS a été piraté. Pour que cela fonctionne, vous devez configurer RoboServer pour faire confiance à votre Management Console et configurer Management Console pour faire confiance à vos RoboServers.

Cela vous oblige à modifier les fichiers dans `ManagementConsole.war`, donc assurez-vous que votre serveur Tomcat n'est pas en cours d'exécution lorsque vous effectuez cette modification.

Certificats

Vous devrez créer deux certificats, un pour Management Console et un pour RoboServer, chaque certificat contient une clé privée et une clé publique. La création d'un certificat et l'exportation de la clé publique sont décrites ici. En général, il est conseillé de lire toute la section de l'aide qui traite des certificats, en particulier la section sur les certificats API client/serveur.

La vérification du terminal peut être séparée en deux parties : RoboServer fait confiance à Management Console et Management Console fait confiance à RoboServer. Chacune de ces parties est configurée individuellement, et vous n'avez pas à les configurer toutes les deux.

Faire en sorte que RoboServer fasse confiance à Management Console

Vous devez maintenant configurer un Management Console pour utiliser la clé privée lors de la création de la connexion SSL vers un RoboServer. Cela se fait en modifiant `/WEB-INF/certs.xml` qui se trouve dans le fichier WAR. Indiquez l'emplacement et le mot de passe du certificat, qui pourrait ressembler à ceci :

```
<bean id="sslCertificationConfiguration"
  class="com.kapowtech.mc.config.SSLVerificationConfiguration">
  ...
  <property name="privateCertificateLocation" value="file:///home/roboserver/
client.p12"/>
  <property name="privateCertPassword" value="changeit"/>
</bean>
```

Management Console utilise désormais une clé privée pour établir des connexions SSL. Une fois que la clé publique Management Console est déployée dans le dossier RoboServer `/TrustedClients`, le RoboServer peut vérifier qu'il est connecté au bon Management Console. N'oubliez pas d'activer la fonction **Vérifier les certificats client de l'API** dans les paramètres de RoboServer, et de déployer la clé publique sur tous les RoboServers du cluster.

Faire confiance à la Management Console RoboServer

RoboServer est déjà livré avec un certificat API installé, vous devez donc créer un nouveau certificat et remplacer celui qui est préinstallé. Créez d'abord le certificat comme décrit ci-dessus, puis démarrez les Paramètres RoboServer et allez dans l'onglet **Certificats**. Cliquez sur le bouton **Changer**, sélectionnez le certificat et saisissez le mot de passe lorsque vous y êtes invité. RoboServer utilise désormais le nouveau certificat lors de la création d'une connexion SSL avec un client Management Console (et d'autres clients API).

Vous devez maintenant configurer la Management Console de manière à ne faire confiance qu'aux connexions SSL des RoboServers avec le bon certificat. Comme le certificat client de la Management Console, il est (partiellement) configuré dans `/WEB-INF/certs.xml`, en utilisant les deux options suivantes :

```
<bean id="sslCertificationConfiguration"
  class="com.kapowtech.mc.config.SSLVerificationConfiguration">
  <property name="verifyRoboServerCert" value="true"/>
  <property name="checkHostName" value="true"/>
  ...
</bean>
```

L'option de vérification des certificats RoboServer est un simple drapeau booléen (vrai/faux), car vous devez importer la clé publique RoboServer dans la base de données par défaut du JRE. Le magasin de clés par défaut du JRE est un fichier nommé `cacerts` situé dans `/jre/lib/security/`.

Pour importer la clé publique RoboServer dans `cacerts`, utilisez la commande suivante :

```
keytool -import -alias RoboServer -keystore cacerts -trustcacerts -file
server.pub.cer
```

Un mot de passe vous sera demandé, et ce mot de passe est `changeit` à moins que vous ne l'ayez déjà modifié. L'alias doit être unique, donc si vous avez créé un certificat séparé pour chaque RoboServer,

ajoutez un suffixe. Notez également que les références à cacerts et à server.pub.cer sont relatives dans cet exemple.

L'option checkHostName garantit que Management Console ne communique avec un RoboServer que s'il présente le bon certificat et est contacté en utilisant le nom d'hôte inscrit à l'intérieur du certificat RoboServer. Notez que localhost et 127.0.0.1 ne sont pas considérés comme le même hôte lorsque le nom d'hôte est vérifié.

Dépannage

Le dépannage peut être assez difficile car il n'y a pratiquement aucune information disponible si les connexions SSL ne peuvent pas être établies, mais il est important de savoir ce qui suit.

- Management Console ne démarre pas s'il ne trouve pas le certificat, ou si le mot de passe est erroné.
- Lorsque vous modifiez le certificat de RoboServer dans les paramètres de RoboServer, il vérifie que le mot de passe est correct avant de stocker le certificat.

Si un site Management Console ne peut pas se connecter à un site RoboServer, les informations suivantes peuvent vous aider à résoudre le problème :

- RoboServer s'exécute-t-il ? Essayez de vous connecter au socket pour en être sûr.
- Le nom d'hôte RoboServer est-il correct (si la fonction CheckHostName est activée) ?
- La clé publique v est-elle importée dans les cacerts ? Utilisez `keytool -list -v -keystore cacerts -alias RoboServer` si vous donnez `-alias` il liste tous les certificats.
- Le certificat public de Management Console a-t-il été copié dans le dossier RoboServer / `TrustedClients` ?
- Vérifiez la date d'expiration. La clé publique contient les données d'expiration de la clé privée, et peut être ouverte/visualisée aussi bien sous Windows que sous Linux.

Sessions simultanées pour un compte d'utilisateur

Par défaut, le système permet d'authentifier simultanément un seul compte d'utilisateur à partir de plusieurs endroits. Pour limiter la possibilité de sessions simultanées pour un seul compte utilisateur, ajustez les paramètres dans le fichier `authentication.xml` qui réside dans `WEB-INF/spring`.

Dans `authentication.xml`, localisez la section suivante et supprimez les balises de commentaire (marquées en gras ici) :

```
<!--  
<bean class="com.kapowtech.scheduler.server.spring.security.KapowConcurrentSessionControlAuthenticationStrategy" lazy-init="true"> <constructor-arg ref="sessionRegistry"/>  
<constructor-arg ref="platformEMF"/> <property name="maximumSessions" value="1"/> <property name="exceptionIfMaximumExceeded" value="true"/> </bean>  
-->
```

De plus, pour définir le délai d'attente pour mettre fin automatiquement à la session si l'utilisateur n'effectue aucune action, configurez la propriété `session-timeout` dans le fichier `web.xml` qui réside dans `WEB-INF`. Par défaut, le délai d'attente est de 30 minutes.

Utiliser le serveur Microsoft SQL avec sécurité intégrée

Si vous souhaitez exécuter Kofax RPA Management Console avec une base de données du serveur Microsoft SQL qui utilise une sécurité intégrée, ainsi que stocker des données dans une telle base de données, effectuez les étapes suivantes pour configurer l'environnement. Le pilote JDBC ne peut pas être stocké dans le répertoire Management Console, par conséquent les fichiers JAR et DLL doivent être placés dans les dossiers spécifiés.

Sur le serveur Tomcat

- Copiez le fichier JAR du Microsoft JDBC Driver for SQL Server dans le dossier **lib** du dossier d'installation de Tomcat.
- Copiez le fichier DLL du Microsoft JDBC Driver for SQL Server dans le dossier **bin** du dossier d'installation de Tomcat.

Sur les ordinateurs des développeurs pour les utilisateurs de Design Studio

- Copiez le fichier JAR du Microsoft JDBC Driver for SQL Server dans le dossier **lib** du dossier d'installation Design Studio.
- Copiez le fichier DLL du Microsoft JDBC Driver for SQL Server dans le dossier **jre\bin** du dossier d'installation Design Studio.

Sur les ordinateurs RoboServer

- Copiez le fichier JAR du Microsoft JDBC Driver for SQL Server dans le dossier **lib** du dossier d'installation RoboServer.
- Copiez le fichier DLL du Microsoft JDBC Driver for SQL Server dans le dossier **jre\bin** du dossier d'installation RoboServer.

Remarque Les utilisateurs utilisant Design Studio et des RoboServers doivent avoir des droits d'accès à la base de données et doivent s'exécuter sous Windows.

Configurer le fichier WAR de Management Console

L'installation complète de Kofax RPA contient le dossier `webApps` avec l'outil de ligne de commande `Configurator.jar` pour extraire la configuration d'un fichier WAR de Management Console ou l'appliquer à celui-ci.

Utilisation : `java -jar Configurator.jar <OPTIONS>`

Cet outil peut être utilisé pour appliquer les paramètres nécessaires aux Management Consoles, comme par exemple lors de la mise à jour de votre installation Kofax RPA sans configuration manuelle fastidieuse de chaque Management Console installée.

Vous pouvez

- Créez un modèle avec les paramètres de Management Console.
- Extrayez les paramètres du fichier WAR de Management Console dans l'installation existante de Management Console.
- Appliquez les paramètres aux Management Consoles nouvellement installées.

Le tableau suivant contient les arguments disponibles et leur description. Exécutez `java -jar Configurator.jar -h` à tout moment pour invoquer la référence de la commande.

Argument	Description
-h,--help	Affiche la référence de la commande.
-p,--use-properties <arg>	Indiquez le nom et le chemin d'accès au fichier de propriétés à utiliser en entrée.
-t,--template <arg>	Crée un fichier de propriétés vide ne contenant que des valeurs par défaut.
-w,--war-file <arg>	Indiquez le chemin d'accès et le nom du fichier WAR, y compris l'extension qui contient les paramètres de Management Console. Par défaut : Racine
-x,--extract <arg>	Extrayez les propriétés du fichier WAR de Management Console dans votre installation actuelle.

- Créez un modèle avec les paramètres de Management Console.
- Extrayez les paramètres du fichier WAR de Management Console dans l'installation existante de Management Console.
- Appliquez les paramètres aux Management Consoles nouvellement installées.

La procédure générale de configuration des Management Consoles installées est la suivante.

1. Installez et configurez manuellement une instance de Management Console.
2. [Extrayez les paramètres de configuration](#) du fichier WAR Management Console que vous avez configuré.
3. Modifiez le fichier de configuration pour l'adapter aux nouvelles instances du Management Console.
4. Installez un nouveau Management Console sur un serveur Tomcat avec le nouveau fichier WAR dans le dossier `webApps`. Copiez le pilote jdbc de la base de données que vous souhaitez utiliser dans le dossier `lib` de votre dossier d'installation Kofax RPA. Notez que le serveur Tomcat doit être arrêté.
5. [Appliquez les paramètres](#) à l'installation Management Console nouvellement créée.
6. Démarrez le serveur Tomcat.

Créer un modèle avec les paramètres de Management Console

Un fichier modèle Management Console est un fichier qui contient tous les paramètres de configuration de Management Console avec des valeurs vides. Vous pouvez ensuite remplir les valeurs dans le fichier et appliquer ces valeurs à un Management Console nouvellement installé. Le dossier se compose de plusieurs sections. Chaque titre de section contient un nom du fichier qui comprend les options de cette section. Par exemple, la première section contient `context.xml` dans son titre et la section est constituée des paramètres déclarés dans le fichier Tomcat `context.xml` ou dans le fichier `ManagementConsole.xml` s'ils sont déclarés en externe. Chaque section et option de ce dossier fait l'objet d'une description détaillée. L'ensemble de valeurs est le même que les variables d'environnement pour le conteneur du docker ManagementConsole. Voir [Variables d'environnement](#) pour plus d'informations.

Pour créer un fichier modèle, procédez comme suit.

1. Localisez le sous-dossier `webApps` dans votre dossier d'installation Kofax RPA.

2. Exécutez `Configurator.jar` comme suit :

```
java -jar Configurator.jar -t <target configuration file>
```

Exemple `java Configurator.jar -t config.properties`

Après avoir exécuté la commande ci-dessus, le dossier `webApps` devrait contenir le fichier `config.properties` avec toutes les options extraites de votre Management Console installé sur le Tomcat spécifié.

Extraire les paramètres du fichier WAR de Management Console existant

Si vous avez plusieurs Management Consoles déployées sur votre réseau, il faut beaucoup de temps pour mettre en place de nouvelles instances lors de la mise à niveau vers une nouvelle version de Kofax RPA. En utilisant l'outil `Configurator.jar`, vous pouvez extraire les paramètres de configuration de l'installation existante de Management Console pour les appliquer aux Management Consoles nouvellement installées.

Pour extraire les paramètres, procédez comme suit.

1. Localisez le sous-dossier `webApps` dans votre dossier d'installation Kofax RPA.
2. Exécutez `Configurator.jar` comme suit :

```
java -jar Configurator.jar -x <target configuration file> -w <path to war file>
```

Exemple `java Configurator.jar -x config.existing.properties -w ManagementConsole`

Notez que certaines valeurs de propriété ne sont pas extraites, comme les paramètres de la base de données, car le schéma de la base de données change d'une version à l'autre et vous devez utiliser une nouvelle base de données avec une nouvelle version de Management Console. Des remarques avec des descriptions sont ajoutées aux propriétés ignorées.

Lorsque vous spécifiez plusieurs répertoires LDAP sur votre réseau, indiquez le nombre de répertoires dans la propriété `login.ldap.directory.count` et remplacez `<n>` dans chacune des propriétés LDAP avec un numéro de série du répertoire que vous spécifiez commençant par 1.

Appliquer les paramètres au fichier WAR de Management Console

Après avoir créé un modèle et modifié ses valeurs ou après avoir extrait les paramètres de l'installation existante de Management Console, vous pouvez appliquer les paramètres spécifiés à un Management Consolenouvellement créé. Notez que vous devez appliquer les valeurs avant de démarrer le serveur Tomcat avec le Management Console installé. Les modifications sont appliquées au fichier WAR de Management Console.

Pour appliquer les valeurs spécifiées dans le fichier de propriétés, procédez comme suit.

1. Localisez le sous-dossier `webApps` dans votre dossier d'installation Kofax RPA.
2. Exécutez `Configurator.jar` comme suit :

```
java -jar Configurator.jar -p <source configuration file> -w <path to war file>
```

Exemple `java Configurator.jar -p config.properties -w ManagementConsole`

Configurer le Serveur du système de fichiers du robot

Le serveur Système de fichiers du robot (RFS) fournit un stockage partagé pour les RoboServers, les instances Design Studio et les agents Desktop Automation. Pour configurer un serveur RFS sur un serveur Tomcat, suivez les étapes suivantes.

La taille maximale du fichier qui peut être téléchargé dans le système de fichiers du robot est de 100 Mo.

1. Localisez le fichier `rfs.war` dans le dossier `webApps` de votre dossier d'installation Kofax RPA.
Par exemple, sur un système Windows, le dossier réside dans : `C:\Program Files\Kofax RPA 11.2.0.0\WebApps`
2. Copiez `rfs.war` dans le dossier `webapps` de votre dossier d'installation d'Apache Tomcat.
Par exemple, sur un système Windows, le dossier réside dans : `C:\apache-tomcat\webapps`
3. Redémarrez le serveur Tomcat.
Une fois que vous avez redémarré le serveur Tomcat, le dossier `webapps` dans le dossier d'installation de Tomcat doit contenir le sous-dossier `rfs`.
4. Dans un éditeur de texte, ouvrez le fichier `web.xml` situé dans le dossier `webapps\rfs\WEB-INF` du dossier d'installation de Tomcat.
5. Repérez le paramètre `mc-path` et spécifiez l'URL Management Console dans `param-value`. Par exemple, `http://localhost:50080`.
6. Localisez et réglez le paramètre `allow-absolute-paths` sur `true` ou `false`.
Si `allow-absolute-paths` est réglé sur `true`, vous pouvez créer des partages de fichiers RFS avec des chemins d'accès tels que `c:\files`, `z:\data`, et d'autres chemins auxquels l'utilisateur du service RFS peut accéder. Si `allow-absolute-paths` est réglé sur `false` et `data-path` sur un dossier spécifique, tel que `/data` sous Linux, le service ne permet d'accéder qu'aux partages dont le chemin d'accès se trouve dans le dossier spécifié, tel que `/data`.
7. Indiquez un dossier pour stocker les données temporaires d'exécution du robot dans `data-path`.
Par exemple, `C:/RFSData`. Notez que vous ne pouvez spécifier le chemin absolu que si `allow-absolute-paths` est configuré sur `true`.
Les actions temporaires sont créées et supprimées en tant que sous-dossiers du dossier spécifié.
8. Laissez les autres paramètres tels quels et redémarrez le serveur Tomcat.
9. Ouvrez le site Management Console et allez dans **Paramètres > Général > Serveur du système de fichiers du robot**.
Sélectionnez **Utiliser le serveur du système de fichiers du robot** et indiquez l'URL du serveur Tomcat où vous avez configuré le serveur RFS. Par exemple, `http://myserver.mydomain:8080/rfs`.

Vous pouvez désormais utiliser des systèmes de fichiers configurés pour partager et stocker des données utilisées et/ou produites par des robots. Pour ajouter une configuration pour un système de fichiers, voir « Système de fichiers du robot » dans *l'aide de Kofax RPA*.

Exemple : Dossier Carte vers le Système de fichiers du robot

Cet exemple fournit des étapes générales sur la façon de mapper un dossier sur Windows à un système de fichiers du robot dans Management Console et d'ajouter une étape à un robot dans Design Studio pour écrire des données dans ce Système de fichiers du robot.

Avant de suivre cet exemple, nous vous recommandons de lire la procédure « Configurer le serveur du Système de fichiers du robot » ci-dessus et « Système de fichiers du robot » dans l'*aide de Kofax RPA* car elles fournissent des informations détaillées sur la configuration et l'utilisation de la fonctionnalité du Système de fichiers du robot.

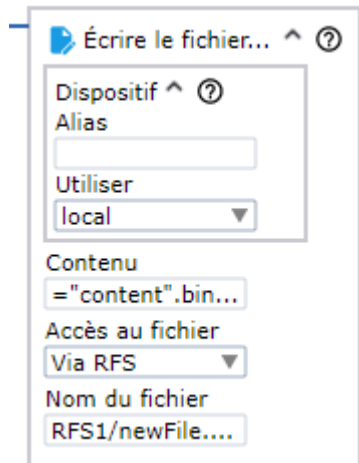
Cette procédure est écrite en supposant que vous avez effectué les étapes 1 à 7 de « Configuration du serveur du système de fichiers du robot ».

1. Dans le fichier `web.xml`, dans le paramètre `data-path`, indiquez le chemin d'accès à un dossier pour stocker les données temporaires d'exécution du robot. Par exemple, `c:/rfs`.

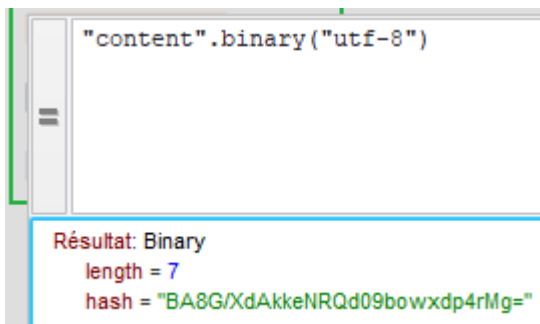
```
<init-param>
  <!-- the path to where local data is stored -->
  <param-name>data-path</param-name>
  <param-value>c:/rfs</param-value>
</init-param>
```

Redémarrez le serveur Tomcat pour appliquer les paramètres.

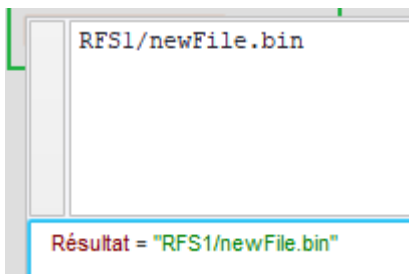
2. Dans **Management Console** > **Paramètres** > **Général** > **Serveur du système de fichiers du robot**, sélectionnez-le pour utiliser le système de fichiers du robot.
3. Dans **Management Console** > **Répertoire** > **Système de fichiers du robot**, ajoutez une configuration pour votre système de fichiers.
 - a. Dans l'onglet **Informations générales**, indiquez tous les paramètres requis.
Dans le paramètre **Nom du système de fichier**, spécifiez **RFS1**. Dans le paramètre **Chemin**, spécifiez : **rfs1_folder**.
Dans ce cas, `rfs1_folder` est le dossier racine de **RFS1**, donc le chemin absolu serait `c:/rfs/rfs1_folder`. Le dossier `rfs1_folder` sera créé automatiquement s'il n'est pas créé manuellement.
 - b. Dans l'onglet « **Jetons d'accès autorisés** », collez le jeton de votre instance Design Studio pour rendre le système de fichiers accessible à cette instance. Copiez le jeton de la fenêtre **Aide** > **À propos de** dans Design Studio.
4. Enregistrez la configuration.
5. Sur Design Studio, dans un robot, créez une étape **Écrire un fichier** avec les propriétés suivantes.



Contenu : Spécifiez une variable binaire contenant des données à écrire dans le système de fichiers du robot. Dans cet exemple, le « contenu » est la chaîne de caractères écrite dans le fichier. Le contenu du fichier doit être binaire.



Nom du fichier : Indiquez le chemin d'accès au fichier dans lequel les données sont écrites. Le nom du système de fichiers du robot est sensible à la casse.



- Après avoir exécuté l'étape, le fichier newFile.bin contiendra les données du « contenu ». Le fichier sera enregistré dans dossier `c:/rfs/rfs1_folder/`.

Chapitre 3

Exécuter les composants RPA en tant que services

Ce chapitre décrit comment exécuter différents composants RPA en tant que services à l'aide du programme `ServiceInstaller.exe`.

ServiceInstaller.exe expliqué

Pour exécuter un composant Kofax RPA en tant que service, vous devez d'abord l'installer à l'aide du programme `ServiceInstaller.exe`. Voici un exemple général qui présente les arguments de la ligne de commande du programme « `RPAComponent` » (bien qu'elle soit affichée sur plusieurs lignes ici, il s'agit d'une commande d'une seule ligne) :

```
ServiceInstaller.exe -i RPAComponent.conf wrapper.ntservice.account=Account
wrapper.ntservice.password.prompt=true wrapper.ntservice.name=Service-
name wrapper.ntservice.starttype=Start-method wrapper.syslog.loglevel=INFO
wrapper.app.parameter.1="First-Argument" wrapper.app.parameter.2="Second-argument"
```

wrapper.ntservice.account

Le compte de l'utilisateur qui doit exécuter un « `RPAComponent` ». Kofax RPA stocke la configuration dans le répertoire de l'utilisateur et il est important de choisir un utilisateur qui a la bonne configuration.

Pour exécuter « `RPAComponent` » comme un utilisateur de domaine, saisissez le compte dans le formulaire `domain\account`

Pour exécuter « `RPAComponent` » comme un utilisateur normal, saisissez le compte dans le formulaire `\account`

Remarque Pour des raisons de sécurité, n'utilisez pas le compte `LocalSystem` pour la connexion au service `RoboServer`. Si `LocalSystem` est utilisé, l'erreur suivante se produit lorsque les robots `WebKit` (par défaut) sont exécutés : « Impossible d'établir une connexion avec `WebKitBrowser`. Échec de connexion au bus. »

wrapper.ntservice.password.prompt

La valeur `vrai` invite l'utilisateur à saisir le mot de passe du compte. Si vous préférez saisir le mot de passe dans la ligne de commande, utilisez `wrapper.ntservice.password=<your-password>`.

wrapper.ntservice.name

Le nom du service à installer. Notez que le nom du service ne peut pas contenir d'espaces.

wrapper.ntservice.starttype

Précisez les valeurs suivantes.

- **AUTO_START** : si le service doit être démarré automatiquement lors du redémarrage du système.

- **DELAY_START** : si le service doit être lancé après un court délai.
- **DEMAND_START** : si vous souhaitez démarrer le service manuellement.

wrapper.syslog.loglevel

Rediriger la sortie de la console de « RPAComponent » vers le journal des événements.

wrapper.app.parameter.

Les arguments de « RPAComponent ». Vous pouvez en saisir autant ou aussi peu que nécessaire.

Lorsque le service est installé, l'utilisateur se voit accorder les droits de « connexion en tant que service ». Si le service ne démarre pas, vérifiez que le droit est accordé en ouvrant gpedit.msc et (sous Windows 10) naviguez jusqu'à **Administrative Tools > Local Security Policy > Local Policy > User Rights Assignment > Log on as a service > Properties** et ajoutez l'utilisateur.

Exécuter RoboServer et la Management Console en tant que service

Les deux RoboServer et Management Console sont lancés par le même programme du serveur, RoboServer, en fonction des arguments qui lui sont fournis au démarrage.

Voir la section [Paramètres RoboServer](#) dans [Démarez RoboServer](#) pour une description détaillée des arguments de la ligne de commande du programme RoboServer.

Les exemples suivants expliquent comment démarrer automatiquement un Management Console et RoboServer sous Windows et Linux.

Démarrer RoboServer sous Windows

Pour qu'un RoboServer démarre automatiquement sous Windows, ajoutez-le en tant que service Windows. Nous vous montrerons comment ajouter et supprimer des services Windows en utilisant le programme `ServiceInstaller.exe` inclus dans l'installation de Kofax RPA.

Ajouter des services Windows

Voici des exemples d'installation des RoboServers dans différentes configurations. Dans ces exemples, MC signifie Management Console et RS signifie RoboServer.

- Le script suivant installe les services qui démarrent les RoboServers avec des paramètres par défaut. Le nom du service peut être modifié selon les besoins.

```
ServiceInstaller.exe -i RoboServer.conf wrapper.ntservice.account=.
\<YOUR_USERNAME> wrapper.ntservice.password.prompt=true
wrapper.ntservice.name="RoboServer11.2.0_MC"
wrapper.ntservice.starttype=MANUAL wrapper.syslog.loglevel=INFO
wrapper.app.parameter.1="-p" wrapper.app.parameter.2="PORT
NUMBER FOR MC RS TO RUN ON" wrapper.app.parameter.3="-mcUrl"
wrapper.app.parameter.4="URL OF MC" wrapper.app.parameter.5="-cl"
wrapper.app.parameter.6="NAME OF CLUSTER"
```

- Ce script crée un service Windows qui ne fait que lancer la Management Console. C'est la configuration recommandée car la Management Console devrait s'exécuter, si possible, sous sa propre JVM. Le nom du service Windows peut être modifié selon les besoins.

```
ServiceInstaller.exe -i RoboServer.conf wrapper.ntservice.account=.
\
```

- Les scripts suivants installent des services qui lancent deux RoboServers : l'un sur le port 50000 et l'autre sur le port 50001. Le nom du service peut être différent :

```
ServiceInstaller.exe -i RoboServer.conf wrapper.ntservice.account=.
\
```

Supprimer les services Windows

Pour désinstaller un service, vous pouvez exécuter la commande suivante :

```
ServiceInstaller.exe -r RoboServer.conf wrapper.ntservice.name=Service-name
```

wrapper.ntservice.name

Le nom du service à supprimer.

Démarrer des serveurs sous Linux

La façon la plus simple de faire démarrer automatiquement un RoboServer sous Linux est d'utiliser crontab. Utilisez la commande suivante pour créer ou modifier la liste des tâches planifiées dans Linux pour l'utilisateur concerné :

```
crontab -u someUser -e
```

À la liste des tâches planifiées, ajoutez par exemple

```
@reboot $HOME/Kofax RPA_11.2.0/bin/RoboServer -mcUrl http://  
nomdutilisateur:motdepasse@localhost:8080/ManagementConsole -p 50000
```

ou

```
@reboot $HOME/Kofax RPA_11.2.0/bin/RoboServer -mcUrl http://  
nomdutilisateur:motdepasse@localhost:8080/ManagementConsole -p 50000 -MC
```

De cette façon, le programme RoboServer commence par les arguments de la ligne de commande indiqués après le redémarrage. Notez que vous devez identifier le répertoire bin sous le dossier d'installation réel.

Le nom d'utilisateur et le mot de passe par défaut sont `admin:admin`.

Exécuter Synchronizer en tant que service

Kofax RPA Synchronizer compare et synchronise l'état des objets entre la Management Console et votre répertoire. Pour plus d'informations sur Synchronizer, voir « Gestion du cycle de vie du robot » dans *Aide de Kofax RPA*.

Cette rubrique fournit un exemple de démarrage de Synchronizer en tant que service Windows.

Ajouter des services Windows

Voici un exemple d'installation de Synchronizer en tant que service. Voir [ServiceInstaller.exe expliqué](#) pour plus d'informations sur `ServiceInstaller.exe`.

```
ServiceInstaller.exe -i Synchronizer.conf wrapper.ntservice.account=domain  
\account wrapper.ntservice.password.prompt=true  
wrapper.ntservice.name="Synchronizer11.2.0_MC"  
wrapper.ntservice.starttype=MANUAL wrapper.syslog.loglevel=INFO  
wrapper.app.parameter.1="-c" wrapper.app.parameter.2="--mc-url"  
wrapper.app.parameter.3="http://localhost:8080/ManagementConsole"  
wrapper.app.parameter.4="--username" wrapper.app.parameter.5="admin"  
wrapper.app.parameter.6="--password" wrapper.app.parameter.7="admin"  
wrapper.app.parameter.8="--interval" wrapper.app.parameter.9="3"  
wrapper.app.parameter.10="--no-host-key" wrapper.app.parameter.11="false"  
wrapper.app.parameter.12="--private-key" wrapper.app.parameter.13="local"
```

Supprimer les services Windows

Pour désinstaller un service, vous pouvez exécuter la commande suivante :

```
ServiceInstaller.exe -r Synchronizer.conf  
wrapper.ntservice.name="Synchronizer11.2.0_MC"
```

wrapper.ntservice.name

Le nom du service à supprimer.

Chapitre 4

WebSphere Management Console

Management Console peut être déployé sur un serveur webSphere. La procédure suivante vous aide à configurer Management Console et votre logiciel webSphere pour qu'ils fonctionnent ensemble.

Remarque Certaines des étapes de configuration peuvent nécessiter le redémarrage du serveur webSphere.

1. Téléchargez l'installateur complet de Kofax RPA et procédez à l'installation. Sélectionnez l'option **Management Console WAR** lors de l'installation.
2. Ouvrez et connectez-vous à la console d'administration webSphere.
3. Définissez les Propriétés personnalisées suivantes de la machine virtuelle Java. Pour définir les propriétés, dans le volet gauche de la console d'administration webSphere, naviguez vers **Serveurs > Types de serveurs > Serveurs d'applications webSphere** et cliquez sur le serveur d'application que vous utiliserez pour exécuter Kofax RPA Management Console. Ensuite, rendez-vous sur **Process definition > Java Virtual Machine > Custom properties**.

Nom	Valeur
com.ibm.crypto.provider.DoRSATypeChecking	faux
com.ibm.websphere.persistence.ApplicationsExcludedFromJpaProcessing	*

4. Configurez le fichier ManagementConsoleWebsphere.war.

L'application Management Console se présente sous la forme d'une archive d'applications web (fichier WAR) appelée ManagementConsoleWebsphere.war, qui se trouve dans le dossier / webApps du dossier d'installation Kofax RPA. Avant de pouvoir le déployer en tant qu'application autonome sur webSphere, il doit être reconfiguré pour s'adapter à votre environnement.

Un fichier WAR est compressé à l'aide d'un fichier zip compressé. Pour accéder aux fichiers de configuration, vous devez extraire le fichier zip. Une fois que les fichiers de configuration sont mis à jour, vous redécompressez et déployez ManagementConsoleWebsphere.war sur votre serveur webSphere. Voir [Configurer ManagementConsole.war](#) pour plus d'informations.

- a. Extrayez le fichier login.xml situé à WEB-INF/login.xml dans le fichier ManagementConsoleWebsphere.war.
- b. Définissez le rôle de l'administrateur de Management Console en ajoutant le bean suivant au fichier login.xml. Voir [LDAP Integration](#) pour obtenir plus d'informations.

```
<bean id="webSphereConfiguration"
class="com.Kapowtech.mc.config.WebSphereConfiguration" lazy-init="true">
  <property name="adminGroups">
    <list>
      <value>KapowAdmin</value>
    </list>
  </property>
```



```
</bean>
```

Remarque Indiquez la valeur dans le bean sous une forme spécifiée dans le serveur webSphere, par exemple en ajoutant un préfixe ou un suffixe si nécessaire.

- c. Re-zip du fichier `ManagementConsoleWebsphere.war` en ajoutant le `login.xml` édité.
5. Ajouter le répertoire LDAP au domaine pour Kofax RPA sous **Sécurité > Sécurité mondiale > Répertoires fédérés**. Par exemple :
Répertoire `KapowDemo`, `DC=Kapowdemo`, `DC=local`.
6. Précisez une source de données sous **Ressources > JDBC > Data sources**. Nous vous recommandons vivement de créer une nouvelle base de données pour les tables utilisées par Management Console. Vous pouvez utiliser une base de données derby intégrée à la webSphere, mais nous vous recommandons d'utiliser MySQL ou une autre base de données pour entreprise. Voir [Créer une nouvelle base de données](#) pour plus d'informations.
7. Déployez le fichier `ManagementConsolewebsphere.war` comme suit :
 - a. Naviguer vers **Applications > Nouvelle application** et suivez la procédure pour installer une nouvelle application d'entreprise à partir d'un système de fichiers local.
 - b. En utilisant le **Chemin rapide** dans l'installation de l'application, modifiez les éléments suivants :
 - Nom de l'application, tel que `ManagementConsole`.
 - Target Resource JNDI Name au nom JNDI désigné de votre source de données configurée en liant la référence de la ressource `jdbc/Kapow/platform` à la source de données par défaut.
 - Context Root à `/mc`.
8. Pour l'application ManagementConsole, placez **Class loader order** sur **Classes loaded with local class loader first** sous **Enterprise Applications > ManagementConsole > Class loader**.
9. Sélectionnez **Enable application security** dans les paramètres de **Global Security**.
10. Pour en savoir plus sur le rôle de la sécurité, consultez le site **Enterprise Applications > ManagementConsole > Security role to user/group mapping**.

Une fois la procédure d'installation terminée, redémarrez le serveur webSphere et essayez de vous connecter à l'entreprise Management Console en tant qu'utilisateur: admin, mot de passe: admin.

Chapitre 5

Journal d'audit pour Management Console

Le journal d'audit de Management Console enregistre toutes les opérations de l'utilisateur dans le Management Console, comme les appels API. En configurant le fichier `log4j2.properties`, vous pouvez enregistrer les informations dans un fichier ou une base de données. Sur un système Windows, le fichier `log4j2.properties` se trouve ici : `User home\AppData\Local\Kofax RPA\version\Configuration`.

Connexion au fichier

```
#Log4j2 log to file configuration example

name = PropertiesConfig
appenders = auditLogAppender

appender.auditLogAppender.name = auditLog
appender.auditLogAppender.type = File
appender.auditLogAppender.fileName=logFilePath/logFileName.log
appender.auditLogAppender.layout.type = PatternLayout
appender.auditLogAppender.layout.pattern=%d - %m%n

logger.auditLog.name = auditLog
logger.auditLog.level = INFO
logger.auditLog.appenderRef.auditLog.ref = auditLog
logger.auditLog.additivity = false
```

Se connecter à la base de données

Remarque Les instructions suivantes utilisent la base de données MySQL comme exemple, mais d'autres bases de données prises en charge peuvent être utilisées pour la journalisation en utilisant leurs pilotes JDBC spécifiques et leurs connexions URL.

Pour activer l'enregistrement d'audit dans la base de données MySQL du Management Console s'exécutant avec le RoboServer intégré, procédez comme suit :

1. Copiez le fichier JAR du connecteur MySQL dans le sous-dossier `lib` du dossier d'installation de Kofax RPA (où se trouvent les fichiers `Kapowtech-common.jar` et `platform.jar`). Par exemple, `mysql-connector-java-<version>.jar`. Utilisez la dernière version disponible du pilote pour votre Java. Pour plus d'informations, voir <https://repo1.maven.org/maven2/mysql/mysql-connector-java/>.
2. Créez une table de base de données dans laquelle vous voulez enregistrer les données. Voici un script MySQL permettant de créer des tables :

```
CREATE TABLE LOGS
(
  DATED    timestamp      NOT NULL,
  LEVEL    VARCHAR(10)    NOT NULL,
  MESSAGE  VARCHAR(1000) NOT NULL
```

```
);
```

Important Pour éviter toute perte d'information, assurez-vous que la colonne du message a une taille minimale de 600 varchars.

3. Ajoutez les lignes suivantes au fichier log4j2.properties :

```
#Log4j2 log to MySQL database configuration example

name = PropertiesConfig
appenders = auditLogAppender

appender.auditLogAppender.name = auditLogAppender
appender.auditLogAppender.type= JDBC
appender.auditLogAppender.connectionSource.type= DriverManager
appender.auditLogAppender.connectionSource.connectionString = jdbc:mysql://
localhost/YourDatabaseSchemaName
appender.auditLogAppender.connectionSource.username = user
appender.auditLogAppender.connectionSource.password = password
appender.auditLogAppender.connectionSource.driverClassName = com.mysql.jdbc.Driver

appender.auditLogAppender.tableName = LOGS

appender.auditLogAppender.columnConfigs[0].type = Column
appender.auditLogAppender.columnConfigs[0].name = DATED
appender.auditLogAppender.columnConfigs[0].pattern = %d{yyyy-MM-dd HH:mm:ss}

appender.auditLogAppender.columnConfigs[1].type = Column
appender.auditLogAppender.columnConfigs[1].name = LEVEL
appender.auditLogAppender.columnConfigs[1].pattern =%p

appender.auditLogAppender.columnConfigs[2].type = Column
appender.auditLogAppender.columnConfigs[2].name = MESSAGE
appender.auditLogAppender.columnConfigs[2].pattern =%msg

logger.auditLog.name = auditLog
logger.auditLog.level = INFO
logger.auditLog.appenderRef.auditLogAppender.ref = auditLogAppender
logger.auditLog.additivity=false
```

Remarque Définissez le nom du schéma de la base de données, le nom d'utilisateur, le mot de passe et le nom de la table que vous avez créée dans la base de données.

Important Le format d'horodatage utilisé dans l'exemple ci-dessus n'est pas universel. Le traitement correct de la requête dépend du type de base de données et du format horaire utilisé dans la base de données.

Assurez-vous que le format de l'horodatage répond aux exigences de la base de données.

Pour activer l'enregistrement d'audit dans la base de données MySQL du site Management Console s'exécutant avec le Tomcat, effectuez les étapes suivantes :

1. Copiez le fichier JAR du connecteur MySQL dans le répertoire `lib` sous Apache Tomcat. Par exemple, `mysql-connector-java-8.0.16.jar`. Utilisez la dernière version disponible du pilote pour votre Java. Pour plus d'informations, voir <https://repo1.maven.org/maven2/mysql/mysql-connector-java/>.

- Les étapes 2 et 3 sont les mêmes que pour le Management Console s'exécutant avec le RoboServer intégré. Le fichier `log4j2.properties` est situé dans le répertoire `tomcat\webapps\Management Console\WEB-INF\classes`.

Référence du journal d'audit

Cette section fournit une liste des opérations enregistrées lorsqu'elles sont exécutées avec succès ou échouent en raison de restrictions d'accès. Des exemples de fichiers journaux sont également fournis à titre de référence.

Événements de connexion

Les RoboServers utilisent des identifiants pour s'enregistrer sur le site Management Console, c'est pourquoi toutes les connexions d'utilisateur sont consignées. Lorsqu'un RoboServer démarre, un événement de connexion est enregistré dans le journal d'audit par l'utilisateur qui a eu accès au RoboServer.

Opérations enregistrées

Les opérations enregistrées sont regroupées par sections dans le site Management Console.

Exemple : Journaux

Voici des exemples de ce à quoi ressemblent les journaux d'exécution des robots dans différents scénarios. MySQL est utilisé comme base de données de journalisation et le message de journalisation est constitué d'un horodatage, d'un niveau de journalisation et d'un message détaillé.

Exécution du robot depuis l'appel REST

Un robot qui est lancé par un appel REST enregistre d'abord le nom du robot avec l'ID, l'ID d'exécution et l'ID de la tâche ; puis l'utilisateur qui a lancé le robot avec le nom du projet et l'ID de la tâche.

```
2019-11-27 16:42:26      INFO      Robot Wait60 with id = 17 execution id =
-1-9-67f20877bebb task id = 9 has requested to start
2019-11-27 16:42:26      INFO      admin run Robot f1/f2/f3/Wait60.robot from Project
Default project with task id = 9 from REST
```

Robot exécuté à partir d'un appel SOAP

```
2019-11-27 16:34:24      INFO      Robot Wait60 with id = 17 execution id =
-1-1-67f20877bebb task id = 1 has requested to start
2019-11-27 16:34:24      INFO      admin run Robot f1/f2/f3/Wait60 from Project
Default project with task id = 1 from SOAP
```

L'exécution du robot a commencé à partir de l'IU

Ce robot a été démarré à partir de la section Robots du site Management Console.

```
2019-11-27 16:35:56      INFO      Robot Wait60 with id = 17 execution id =
-1-2-67f20877bebb task id = 2 has requested to start
2019-11-27 16:35:56      INFO      admin run Robot Wait60 with id = 17 task id = 2
```

Planification déclenché en fonction du temps

Pas de journal.

Planification déclenché par l'utilisateur

Les messages du journal peuvent être référencés par l'ID de la planification et l'ID de la tâche. Les messages du journal d'exécution avec le postfix « - from schedule, started by user » indiquent également que ce robot a été déclenché par une exécution manuelle de la planification. Par exemple, la planification « MultipleTaskSchedule » contient 4 tâches robotisées : ExampleRobot1, ExampleRobot2, ExampleRobot2, ExampleRobot3. Lorsqu'un utilisateur exécute manuellement la planification, les messages du journal doivent se présenter comme suit :

```

2019-12-02 10:50:22      INFO      admin start Schedule MultipleTaskSchedule with id
= 373284858997185
2019-12-02 10:50:22      INFO      Robot ExampleRobot1 with task id = 53 has been
queued for schedule MultipleTaskSchedule with id = 373284858997185
2019-12-02 10:50:22      INFO      Robot ExampleRobot2 with task id = 54 has been
queued for schedule MultipleTaskSchedule with id = 373284858997185
2019-12-02 10:50:22      INFO      Robot ExampleRobot2 with task id = 55 has been
queued for schedule MultipleTaskSchedule with id = 373284858997185
2019-12-02 10:50:22      INFO      Robot ExampleRobot3 with task id = 56 has been
queued for schedule MultipleTaskSchedule with id = 373284858997185
2019-12-02 10:50:22      INFO      Robot ExampleRobot1 with id = 1 execution id =
3816-53-1dc33f3a2a44b task id = 53 has requested to start - from schedule, started by
user
2019-12-02 10:50:22      INFO      Robot ExampleRobot2 with id = 39 execution id =
3816-54-1dc33f3a2a44b task id = 54 has requested to start - from schedule, started by
user
2019-12-02 10:50:23      INFO      Robot ExampleRobot2 with id = 39 execution id =
3816-55-1dc33f3a2a44b task id = 55 has requested to start - from schedule, started by
user
2019-12-02 10:50:23      INFO      Robot ExampleRobot3 with id = 20 execution id =
3816-56-1dc33f3a2a44b task id = 56 has requested to start - from schedule, started by
user

```

Chapitre 6

Scripts SQL pour les tables Kofax RPA

Les scripts SQL pour créer et déposer des tables dans votre base de données se trouvent dans le répertoire `documentation\sql` de votre répertoire d'installation Kofax RPA. Par exemple, `C:\Program Files\Kofax RPA 11.2.0\documentation\sql` sur le système Windows. Le nom du fichier de script comprend le nom de la base de données à laquelle le script est destiné.

Remarque Les scripts SQL sont installés en même temps que la documentation Kofax RPA et lorsque vous installez Design Studio.

Scripts SQL des tables de base de données

Le répertoire `sql` contient quatre sous-répertoires avec des scripts différents comme suit :

- `altosoftsession` : Scripts pour la création et le dépôt de tableaux pour Single Sign-On avec Altosoft
- `logdb` : Scripts pour la création et la suppression de tables de logdb
- `mc` : Scripts pour créer et déposer des tableaux Management Console
- `statistiques` : Scripts pour créer et supprimer des tableaux de statistiques (Kofax Analytics for RPA)

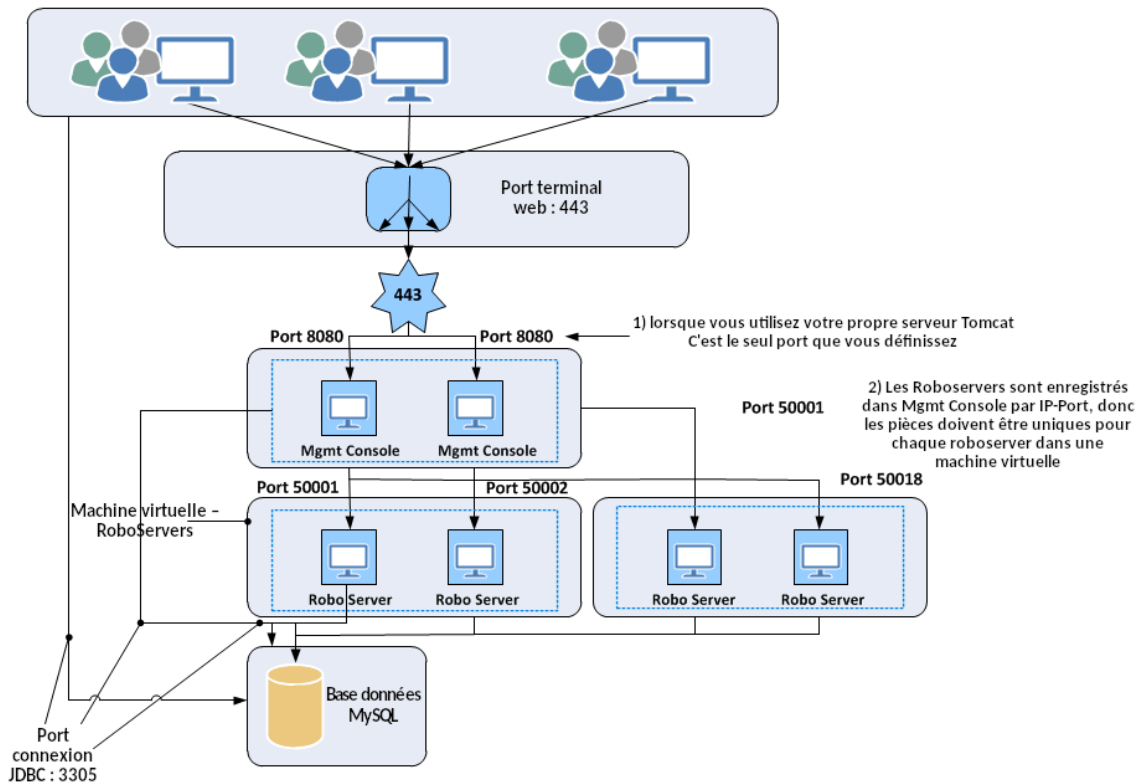
Management Console utilise un composant de planification tiers appelé Quartz. Le quartz nécessite également un certain nombre de tables qui doivent se trouver parmi les autres tables de la plateforme. Ces tableaux sont également créés automatiquement au démarrage de Management Console, ou peuvent être créés manuellement à l'aide des scripts.

Ce qui suit est un script de vérification de Quartz.

```
select count(*) from QRTZ_SIMPLE_TRIGGERS;
select count(*) from QRTZ_BLOB_TRIGGERS;
select count(*) from QRTZ_CRON_TRIGGERS;
select count(*) from QRTZ_TRIGGER_LISTENERS;
select count(*) from QRTZ_CALENDARS;
select count(*) from QRTZ_FIRED_TRIGGERS;
select count(*) from QRTZ_LOCKS;
select count(*) from QRTZ_PAUSED_TRIGGER_GRPS;
select count(*) from QRTZ_SCHEDULER_STATE;
select count(*) from QRTZ_JOB_LISTENERS;
select count(*) from QRTZ_TRIGGERS;
select count(*) from QRTZ_JOB_DETAILS;
```

Annexe A

Modèle de sécurité Kofax RPA



A. Connexion et authentification de l'utilisateur

<i>Catégorie</i>	Authentification et autorisation
<i>Description</i>	L'utilisateur fournit des identifiants de connexion pour l'application Kofax.
<i>Détails de sécurité</i>	<p>Kofax RPA prend en charge la synchronisation des utilisateurs/ groupes avec Active Directory/LDAP. Cela permet à Kofax RPA de tirer parti de l'infrastructure de l'entreprise pour l'authentification et la gestion des identifiants.</p> <p>Kofax RPA dispose également d'un mécanisme d'authentification et d'autorisation spécifique à l'application pour des raisons de commodité. Cela inclut la gestion et le stockage des identifiants. Les mots de passe stockés sont chiffrés.</p>

B. Le client transmet au(x) serveur(s) Kofax RPA

<i>Catégorie</i>	Données en transit
<i>Port</i>	80 ou 443
<i>Protocole</i>	HTTP ou HTTPS
<i>Description</i>	Les clients transmettent aux serveurs Kofax RPA.
<i>Détails de sécurité</i>	Toute la connectivité des clients Kofax RPA (Management Console et Design Studio) aux serveurs Kofax RPA se fait par HTTP/HTTPS. Le HTTPS doit être configuré pour une sécurité maximale.

C. Kofax RPA serveur(s) transmet à un autre Kofax RPA serveur(s)

<i>Catégorie</i>	Données en transit
<i>Port</i>	Configurable. Par défaut 80, 443, 50000, 50443, 49999, 49998
<i>Protocole</i>	HTTP/HTTPS, socket TCP/IP
<i>Description</i>	Les serveurs Kofax RPA transmettent vers/depuis une autre application ou un autre serveur Kofax.
<i>Détails de sécurité</i>	Tous les composants de Kofax RPA peuvent être configurés pour utiliser une communication chiffrée sécurisée (TLS 1.2) avec des certificats personnalisés.

D. Les serveurs Kofax RPA transmettent au serveur de base de données

<i>Catégorie</i>	Données en transit
<i>Port</i>	Varie en fonction du protocole
<i>Protocole</i>	TCP/IP
<i>Description</i>	Kofax RPA serveurs transmettent vers/depuis la base de données
<i>Détails de sécurité</i>	Les serveurs Kofax RPA se connectent à la base de données SQL. Généralement, le système de serveur de base de données est situé au même endroit ou est protégé physiquement de telle sorte que la transmission n'a pas besoin d'être chiffrée. Toutefois, si un tel chiffrement est nécessaire, vous pouvez chiffrer la connexion à la base de données via SSL.

E. Robot et stockage des données

<i>Catégorie</i>	Données au repos
<i>Description</i>	Les robots, les configurations et les métadonnées correspondantes sont stockées sur le Management Console. Les robots peuvent stocker les données des clients dans des bases de données.

Détails de sécurité

Les robots, les configurations et les métadonnées associées sont stockées dans la base de données Kofax, à laquelle on accède par un compte système configuré. Le chiffrement au niveau de la base de données est également disponible en utilisant la fonction de chiffrement de la base de données elle-même.

Que le chiffrement des systèmes de fichiers et/ou des bases de données soit activé ou non, les mots de passe (pour les systèmes externes ou les utilisateurs d'applications spécifiques), sont encore mieux protégés. Les mots de passe stockés dans le magasin des mots de passe ou en entrée dans un planning sont chiffrés à l'aide d'un certificat généré par le client. Nous utiliserons le chiffre choisi pour le certificat afin de chiffrer les mots de passe stockés. Par défaut, l'installation est fournie avec un certificat chiffré RSA 1024 bits, mais nous recommandons vivement au client de générer son propre certificat.