MDX Toolkit

Contents

À propos du MDX Toolkit

January 23, 2019

Le Mobile Device Experience (MDX) Toolkit est une technologie de conteneur d'application qui améliore l'expérience sur les appareils mobiles et vous permet de préparer des applications en vue de les déployer en toute sécurité avec Citrix Endpoint Management en ajoutant les informations suivantes aux applications :

- Le code requis pour prendre en charge des tâches de gestion d'applications mobiles, telles que le provisioning, l'authentification personnalisée, la révocation d'applications, les stratégies de contention des données, le cryptage des données et la mise en place d'un VPN par application.
- Des certificats de sécurité signés.
- Des informations sur les stratégies et d'autres paramètres Endpoint Management.

MDX Toolkit peut encapsuler de manière sécurisée des applications qui ont été créées au sein de votre organisation ou des applications mobiles développées par des tiers.

Vous utilisez la console Endpoint Management pour ajouter votre application à Endpoint Management. Lorsque vous ajoutez l'application, vous pouvez modifier la configuration de la stratégie, ajouter des catégories d'applications, appliquer des workflows et déployer des applications sur des groupes de mise à disposition.

Pour télécharger les composants Endpoint Management, consultez la section https://www.citrix.com/downloads/citrix-endpoint-management/.

Vous pouvez également encapsuler des applications à l'aide de notre outil cloud, MDX Service. Pour plus d'informations sur l'outil, consultez la section MDX Service.

- Problèmes connus dans le MDX Toolkit
- Problèmes résolus dans le MDX Toolkit

À propos de l'encapsulation d'applications

Vous pouvez encapsuler des applications Android ou iOS provenant de fournisseurs d'applications. Grâce à la distribution sur magasins d'applications publics, vous n'avez pas besoin de signer et d'encapsuler des applications développées par Citrix avec le MDX Toolkit. Ce processus optimise considérablement le déploiement d'applications. Étant donné que le serveur Endpoint Management prend déjà en charge le déploiement d'applications depuis des magasins d'applications publics, aucune mise à jour du serveur n'est requise. Cependant, vous pouvez utiliser l'outil MDX Toolkit pour encapsuler des applications tierces ou d'entreprise. Pour plus d'informations sur la distribution du magasin d'applications public, consultez la section Activation de la distribution sur des magasins d'applications publics.

> **Remarque :**
>
> Les éditeurs de logiciels indépendants (ISV) peuvent encapsuler les applications qu'ils développent et les mettre à disposition dans un magasin d'applications ou la galerie d'applications de productivité mobiles Citrix. Pour de plus amples informations, consultez le Guide du développeur MDX.

Le MDX Toolkit combine des fichiers applicatifs (.ipa, .app ou .apk) avec des composants Citrix et votre keystore ou certificat de signature pour produire une application MDX encapsulée.

> **Remarque :**
>
> Le MDX Toolkit prend en charge :
>
> - Les applications Android et iOS développées sur la plate-forme Xamarin.
> - Les applications développées à l'aide de l'infrastructure PhoneGap (Apache Cordova)
>
> Il s'agit des infrastructures tierces testées et prises en charge par Citrix de manière à fonctionner avec le MDX Toolkit.
>
> D'autres infrastructures tierces, telles que Swift, ne sont pas garanties de fonctionner, sauf mention explicite.

Le MDX Toolkit et le SDK XenMobile App pour iOS et Android incluent les outils suivants :

- Un outil GUI macOS capable d'encapsuler des applications iOS et Android.
- Un outil de ligne de commande macOS qui encapsule les applications iOS.
- Un outil de ligne de commande Java qui encapsule les applications Android.
- SDK XenMobile App : les développeurs tiers peuvent utiliser le SDK XenMobile App pour effectuer des actions sur des applications encapsulées en fonction de stratégies Endpoint Management. À titre d'exemple, si une stratégie Endpoint Management empêche le couper-coller dans une application de productivité mobile, un développeur peut empêcher la sélection de texte dans l'application. Pour de plus amples informations, consultez le Guide du développeur MDX.

Le MDX Toolkit et le SDK XenMobile App pour iOS et Android

Vous pouvez utiliser le MDX Toolkit pour encapsuler des applications personnalisées non publiques, compilées et natives pour Android et iOS. Ces applications doivent être créées à l'aide d'Objective-C sur iOS et de Java sur Android.

> **Remarque :**
>
> Le MDX Toolkit 10.7.10 est la dernière version qui prend en charge l'encapsulation des applications de productivité mobiles. Les utilisateurs accèdent aux versions 10.7.5 et ultérieures des

> applications de productivité mobiles depuis des magasins d'applications publics.

Pour l'encapsulation d'applications d'entreprise, commencez par une application iOS (.ipa) ou Android (.apk). Achetez vos applications tierces directement auprès du fournisseur. En effet, les applications iOS téléchargées depuis l'Apple Store sont cryptées et ne peuvent pas être encapsulées.

> **Remarque :**
>
> Le MDX Toolkit n'est pas pris en charge avec Windows Phone.

Problèmes résolus

January 23, 2019

Les problèmes suivants sont résolus dans le MDX Toolkit 18.12.0 :

- Les applications de stockage tierces ne peuvent pas télécharger de fichiers par intermittence lorsqu'elles sont encapsulées avec le MDX Toolkit. [CXM-58814]
- Les applications Android développées en interne sur la plate-forme Xamarin fonctionnent lentement lorsqu'elles sont encapsulées avec le MDX Toolkit. [CXM-58779]

Problèmes résolus dans la version 10.8.60

- Secure Mail pour iOS ne peut pas enregistrer de fichiers vidéo sur ShareFile. [CXM-42238]
- Lorsqu'un fichier auto-config proxy avec dnsResolve défini est configuré en mode Tunnel complet, la navigation sur Secure Web est sensiblement lente. [CXM-49567]
- Lorsqu'elle est encapsulée avec le MDX Toolkit, l'application Cisco Jabber se bloque lors de la connexion. [CXM-51052]
- Les applications d'entreprise peuvent rencontrer des problèmes de connectivité avec les ressources internes lorsque le mode VPN préféré est défini sur SecureBrowse. [CXM-52309]
- Certaines applications tierces peuvent se bloquer au démarrage après avoir été encapsulées avec le MDX Toolkit. [CXM-52311]
- Les applications qui spécifient android.support.multidex.MultiDexApplication ou android.app.Application comme classe d'application ne peuvent pas se connecter aux réseaux internes en mode de navigation sécurisée. [CXM-53126]
- Sur les appareils Android, plusieurs certificats sont générés et les certificats sont révoqués avant leur date d'expiration. [CXM-53428]
- Sur Android, Secure Mail se bloque lorsque les utilisateurs sont déconnectés de Secure Hub. [CXM-53930]
- Sur les appareils iOS, Secure Web et Secure Mail 10.8.45 se bloquent au lancement. [CXM-54089]
- Lorsque les utilisateurs inscrivent un appareil exécutant Secure Mail pour Android avec le portail d'entreprise Intune, Secure Mail cesse de fonctionner. [CXM-54178]

- Sur Android, lors de l'ouverture d'un PDF dans Quick Edit, une erreur s'affiche : « Erreur lors de l'initialisation du convertisseur PDF ». [CXM-54950]
- Lorsque vous encapsulez ShareFile 7.1 pour Android avec le MDX Toolkit, l'authentification unique (SSO) échoue lorsque les utilisateurs tentent d'accéder à du contenu via les connecteurs ShareFile. [CXM-55030]

Problèmes résolus dans la version 10.8.35

Android

- Sur les appareils qui utilisent l'API WebView, y compris Secure Web pour Android, vous ne pouvez pas joindre d'images à partir d'une galerie bien que la stratégie MDX Bloquer la galerie soit désactivée. [CXM-41475]
- Sur Android, Secure Web ne parvient pas à ouvrir tous les sites internes/externes. [CXM-47379]
- Sur Android, les applications internes encapsulées avec le MDX Toolkit 10.7.20 et versions ultérieures se bloquent au lancement. [CXM-47566]
- Certaines applications d'entreprise se bloquent au démarrage et le message d'erreur suivant s'affiche : « Unfortunately, the app has stopped ». [CXM-49901]

iOS

- Sur iOS, le MDX Toolkit 10.7.20 ne peut pas encapsuler certaines applications tierces conçues pour iPad uniquement. [CXM-44122]
- Sur iOS, les utilisateurs ne peuvent pas coller les liens de ShareFile dans d'autres applications MDX, à l'exception de Secure Mail. [CXM-44274]
- Sur iOS, après avoir encapsulé l'application SAP Fiori, la page d'ouverture de session ne s'affiche pas. [CXM-45542]
- Lorsque vous configurez la stratégie MDX Exporter contacts, dans Secure Mail sur les appareils iOS 10, vous ne pouvez pas partager de contacts ni vous synchroniser avec des contacts locaux. Au lieu de cela, le message Échec d'exportation du contact s'affiche. [CXM-44613]
- Sur iOS, les applications tierces se bloquent lors de l'appel de CFSocketConnectToAddress en mode sans restriction. [CXM-46592]
- Secure Web pour iOS se bloque lors de la navigation sur certains sites Web en mode de navigation sécurisée si votre réseau est tunnélisé. [CXM-47989]
- Pour les applications iOS Cordova tierces encapsulées avec la version 10.7.20 de MDX Toolkit, après l'activation de la stratégie Obscurcir le contenu de l'écran, un écran noir apparaît sur les appareils iOS au lieu d'un écran de code PIN. [CXM-48471]
- Une fois que vous mettez à niveau vers iOS 11.3, lorsque les stratégies "Coller" et "Couper et

copier” sont définies sur Restreint, vous pouvez effectuer l’opération de collage dans des applications non gérées. [CXM-50427]

Problèmes résolus dans la version 10.8.5

Android

- Sur Android, lorsque la stratégie BLOCK_DNS_FROM_UNMANAGED_APPS_IN_FULL_VPN est définie et qu’une application MDX demande un tunnel VPN complet alors qu’un VPN est déjà en cours d’exécution, une défaillance du réseau se produit. Les utilisateurs doivent fermer leur session, puis ouvrir une nouvelle session pour résoudre le problème. [CXM-42853]
- Après l’établissement de la connexion Micro VPN sur les appareils Android : lorsque les utilisateurs redémarrent l’appareil, le Micro VPN ne démarre pas. [CXM-43919]
- Sur les appareils fonctionnant sous Android, certaines applications ne fonctionnent pas si les connexions sont acheminées par tunnel vers le réseau interne. [CXM-44735]
- Une erreur se produit lors de la tentative d’encapsulation d’un fichier .apk avec le MDX Service et l’encapsulation échoue. [CXM-47060]

iOS

- Lorsque le chiffrement est désactivé, vous ne pouvez pas copier et coller dans des applications XenMobile Apps sur des appareils exécutant iOS. [CXM-43920]
- Sur iOS, après la mise à niveau de XenMobile Apps vers la version 10.7.30, si le niveau de journalisation est défini sur 11 ou plus, Secure Mail est extrêmement lent et se bloque s’il reste ouvert. [CXM-46721]

Problèmes connus

January 23, 2019

Voici une liste des problèmes connus dans la version 10.8.35 du MDX Toolkit.

- Lorsque la stratégie Échange de documents est définie sur Restreint, Secure Web cesse de fonctionner après avoir tenté de télécharger un fichier. [CXM-48447]

MDX Toolkit version 10.8.5

Il n’y a aucun problème connu dans le MDX Toolkit 10.8.5 pour les appareils Android. Pour accéder aux problèmes résolus, veuillez consulter Problèmes résolus.

MDX Toolkit version 10.7.20

- Lorsque le mode VPN est défini sur Navigation sécurisée, la configuration de OAuth pour Citrix Office365 peut échouer si la suite de chiffrement prise en charge par le serveur principal n'est pas prise en charge par NetScaler dans le déploiement. [CXM-41738]
- Sur Android, la stratégie des domaines Secure Web n'ouvre pas les adresses URL dans le navigateur Web par défaut. [CXM-43021]

MDX Toolkit version 10.7

- Sur iOS 11, les utilisateurs peuvent transférer les données d'une application MDX gérée vers une application non gérée à l'aide de la fonction glisser-déposer. [CXM-38106]
- Sur iOS 11, les applications non gérées sont toujours affichées dans la liste Ouvrir dans si la stratégie Échange de documents est définie sur Restreint. [CXM-38705]
- Sur iOS 11, les fichiers provenant de ShareFile ouverts dans des applications MDX wrappées apparaissent altérés. [CXM-38900]
- Sur iOS, les applications gérées n'apparaissent pas dans la liste Ouvrir dans lors de la première tentative dans les applications gérées. [CXM-38897]
- Sur iOS 10 et 11, si vous sélectionnez Ouvrir dans depuis des applications gérées MDX, un message d'erreur s'affiche. [CXM-38912]

Configuration système requise

January 23, 2019

Cet article fournit la configuration système requise par l'outil MDX Toolkit pour encapsuler des applications mobiles. Il fournit également la configuration requise spécifique pour chaque plate-forme d'application.

> **Important :**
>
> Le SDK XenMobile App 10.2 requiert désormais les composants suivants : JavaScriptCore.framework et LocalAuthentication.framework.

- **Java Development Kit (JDK) 1.7 ou 1.8 :** vous pouvez télécharger le JDK 1.8 depuis Java SE Development Kit Downloads sur le site Web Oracle. Pour obtenir les instructions d'installation, veuillez consulter la section JDK 8 and JRE 8 Installation Guide sur le site Web d'Oracle. Veillez à installer le JDK complet et définissez JDK 1.8 comme valeur par défaut.
- **macOS :** utilisez la version la plus récente. Le programme d'installation de l'outil MDX Toolkit et du SDK XenMobile App doit être exécuté sur macOS. Le programme d'installation comprend

des outils macOS qui encapsulent les applications iOS et Android, ainsi qu'un outil de ligne de commande Java qui encapsule les applications Android.
- **SDK XenMobile App :** utilisez la version la plus récente du SDK iOS et du Xcode ; génération de bitcode désactivée.

Autre configuration requise pour encapsuler des applications mobiles iOS

Pour obtenir l'accès à la configuration requise pour l'encapsulation d'applications pour iOS, vous devez vous enregistrer afin d'obtenir un compte de distribution Apple. Il existe trois types de comptes développeur iOS : Enterprise, Individual et University. Citrix recommande fortement les comptes iOS Developer Enterprise.

- **Comptes iOS Developer Enterprise :** seul type de compte Apple Developer qui vous permet de provisionner, déployer et tester un nombre illimité d'applications pour un nombre illimité d'appareils, avec ou sans encapsulation d'application. Veillez à distribuer votre certificat Developer à vos développeurs pour qu'ils puissent signer les applications.
- **Comptes iOS Developer Individual :** limité à 100 périphériques inscrits par an et ne permet pas l'encapsulation d'applications d'entreprise ni la distribution d'entreprise avec Citrix Endpoint Management.
- **Comptes iOS Developer University :** limité à 200 périphériques inscrits par an et ne permet pas l'encapsulation d'applications d'entreprise ni la distribution d'entreprise avec Endpoint Management.

> **Important :**
>
> Veillez à vérifier le délai d'expiration des profils de provisioning pour votre compte et à renouveler les profils avant qu'ils expirent. Si un profil utilisé pour encapsuler des applications expire, vous devez renouveler le profil, recommencer l'encapsulation des applications, puis réinstaller les applications sur les appareils utilisateur. Pour renouveler un profil de provisioning, ouvrez une session sur votre compte Apple Developer, accédez à **Certificates, Identifiers & Profiles**, puis sélectionnez **Provisioning Profiles**.

Téléchargez les outils de ligne de commande Xcode disponibles sur le site Web Xcode Apple Developer. macOS 10.10 n'installe pas les outils automatiquement. Pour installer les outils, suivez ces étapes :

1. Dans **Applications > Utilities**, cliquez sur Terminal pour utiliser l'interface de ligne de commande Mac (CLI).
2. Exécutez la commande suivante :

```
xcode-select --install
```

   Veillez à inclure deux tirets avant le mot install dans la commande.

3. Une fois les outils de ligne de commande Xcode installés, exécutez Xcode to pour installer les éléments pré-requis.

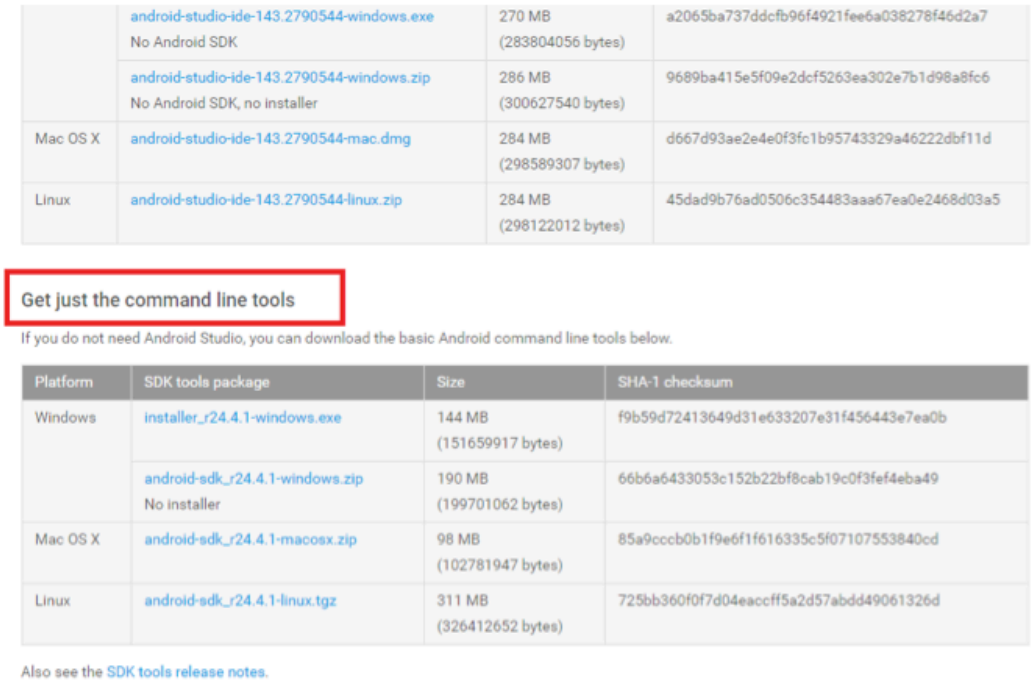
Autre configuration requise pour encapsuler des applications mobiles Android

Pour encapsuler des applications Android, vous avez également besoin d'un kit de développement logiciel (SDK) compatible Android et d'un keystore valide. Pour télécharger, créer et configurer correctement le SDK et le keystore, suivez ces instructions :

Kit de développement logiciel Android

Le MDX Toolkit est compatible avec le niveau d'API 26 du SDK Android.

1. Accédez au site Web de développeurs Google et téléchargez le SDK Android depuis la page de téléchargement des SDK. L'environnement Android Studio complet n'est pas requis. Vous pouvez télécharger les outils de ligne de commande depuis la section en bas de la page.



	android-studio-ide-143.2790544-windows.exe No Android SDK	270 MB (283804056 bytes)	a2065ba737ddcfb96f4921fee6a038278f46d2a7
	android-studio-ide-143.2790544-windows.zip No Android SDK, no installer	286 MB (300627540 bytes)	9689ba415e5f09e2dcf5263ea302e7b1d98a8fc6
Mac OS X	android-studio-ide-143.2790544-mac.dmg	284 MB (298589307 bytes)	d667d93ae2e4e0f3fc1b95743329a46222dbf11d
Linux	android-studio-ide-143.2790544-linux.zip	284 MB (298122012 bytes)	45dad9b76ad0506c354483aaa67ea0e2468d03a5

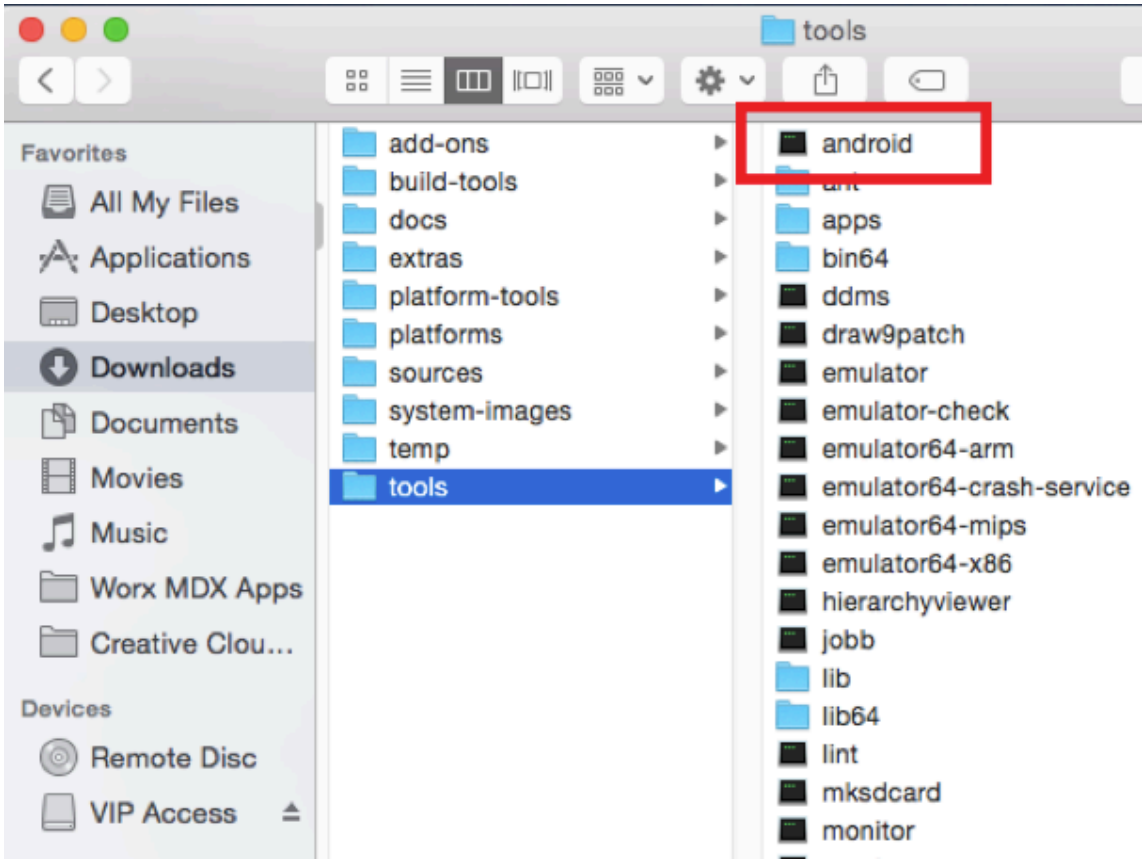
Get just the command line tools

If you do not need Android Studio, you can download the basic Android command line tools below.

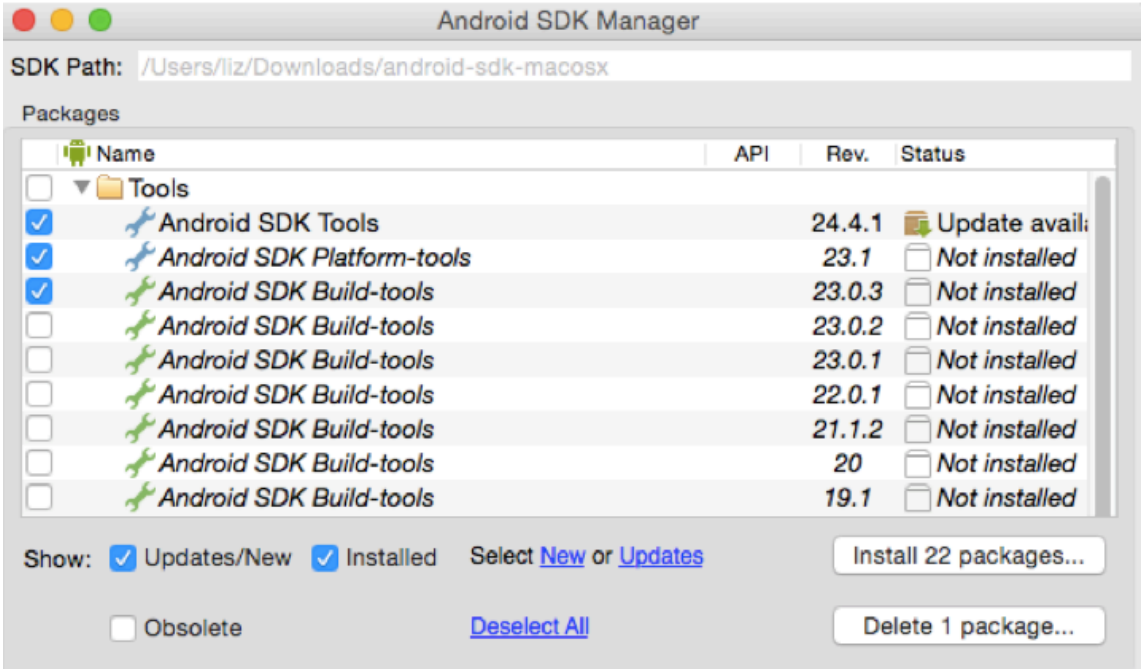
Platform	SDK tools package	Size	SHA-1 checksum
Windows	installer_r24.4.1-windows.exe	144 MB (151659917 bytes)	f9b59d72413649d31e633207e31f456443e7ea0b
	android-sdk_r24.4.1-windows.zip No installer	190 MB (199701062 bytes)	66b6a6433053c152b22bf8cab19c0f3fef4eba49
Mac OS X	android-sdk_r24.4.1-macosx.zip	98 MB (102781947 bytes)	85a9cccb0b1f9e6f1f616335c5f07107553840cd
Linux	android-sdk_r24.4.1-linux.tgz	311 MB (326412652 bytes)	725bb360f0f7d04eaccff5a2d57abdd49061326d

Also see the SDK tools release notes.

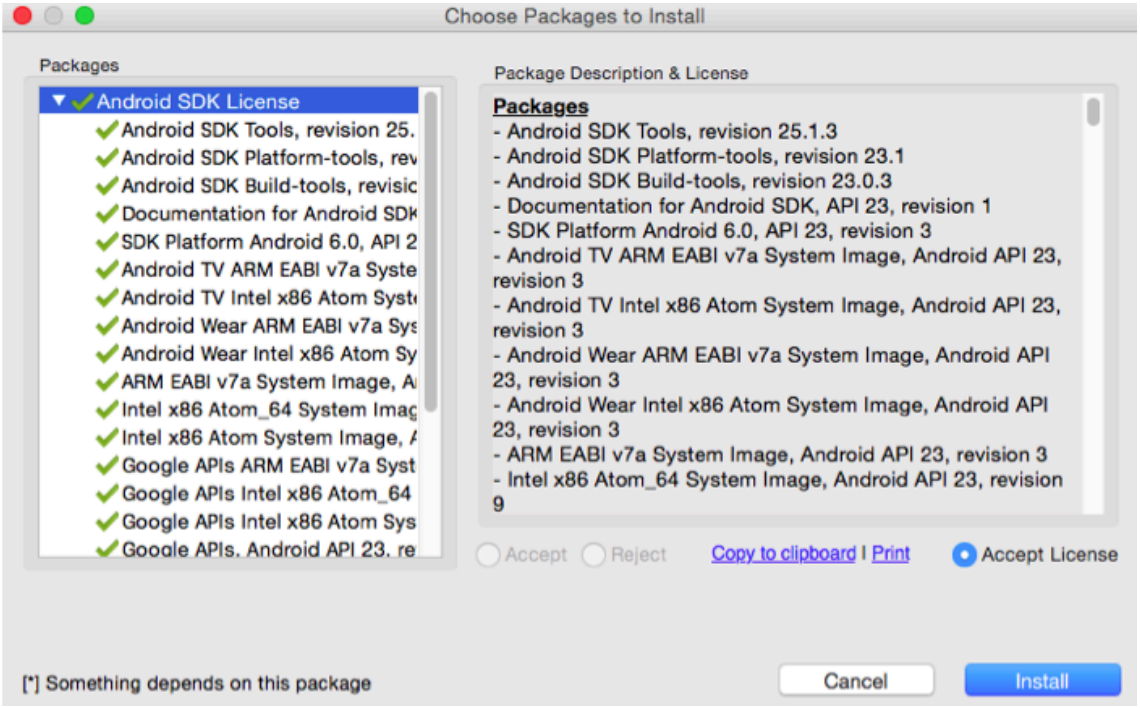
2. Installez les derniers outils, outils de plate-forme et outils de génération. L'installation requiert l'utilisation de l'outil Android dans **Android SDK > tools** pour démarrer le SDK Manager :
   - Dézippez le fichier SDK que vous avez téléchargé.
   - Accédez au dossier tools et cliquez sur **Android** pour exécuter le SDK Manager.



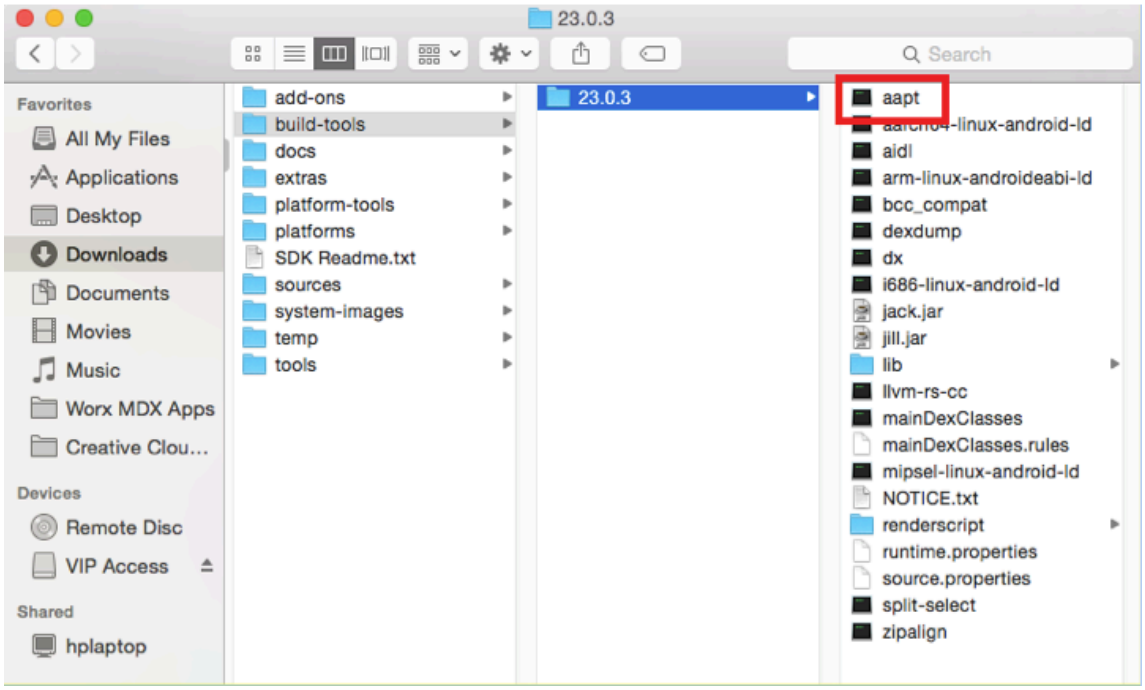
3. Dans le SDK Manager, sélectionnez les dernières versions de ce qui suit :
    - Android SDK Tools
    - Android SDK Platform
    - Android SDK Platform-tools
    - Android SDK Build-tools
4. Cliquez sur **Install Packages**.



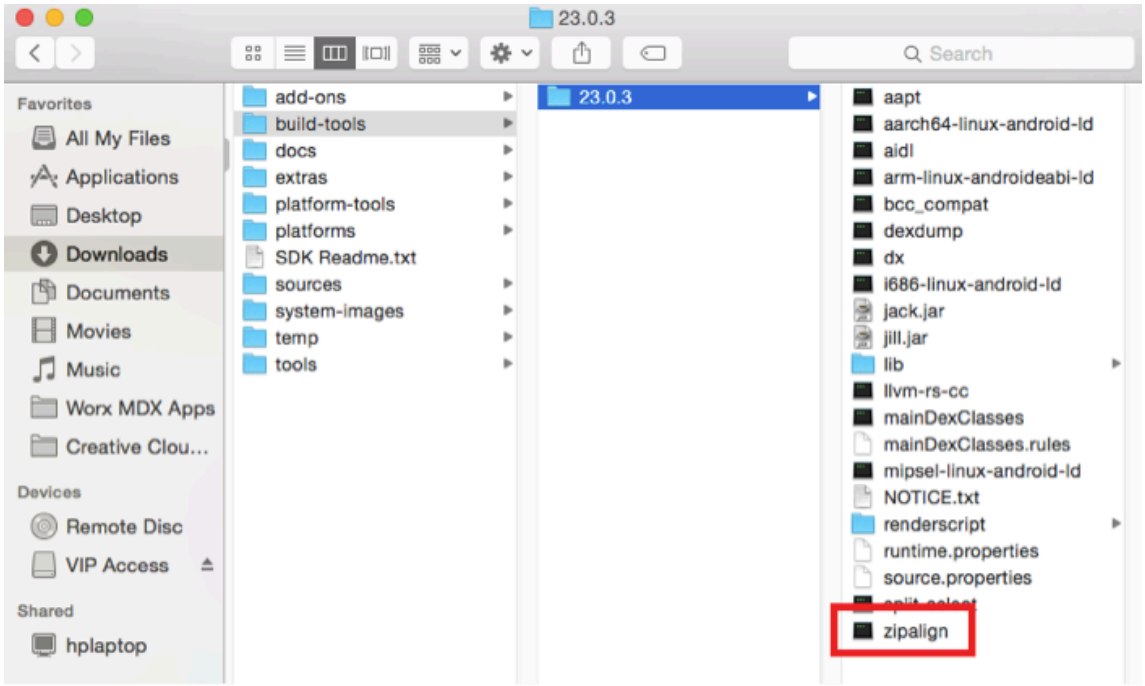
5. Sur l'écran **Choose Packages to Install**, cliquez sur **Accept License** pour tous les packages que vous installez et cliquez sur **Install**.



6. Pour vérifier que vous avez téléchargé les outils SDK et API appropriés, vérifiez que le fichier .aapt est présent dans **Android SDK > build-tools > 23.0.3**.



7. Lors de la mise à jour de votre SDK, vous devez supprimer tous les fichiers .aapt du dossier platform-tools. Assurez-vous que le fichier .aapt figure uniquement dans build-tools.
8. Si le fichier zipalign ne figure pas build-tools, copiez le fichier du dossier platform-tools sur le dossier build-tools, puis supprimez-le de platform-tools.



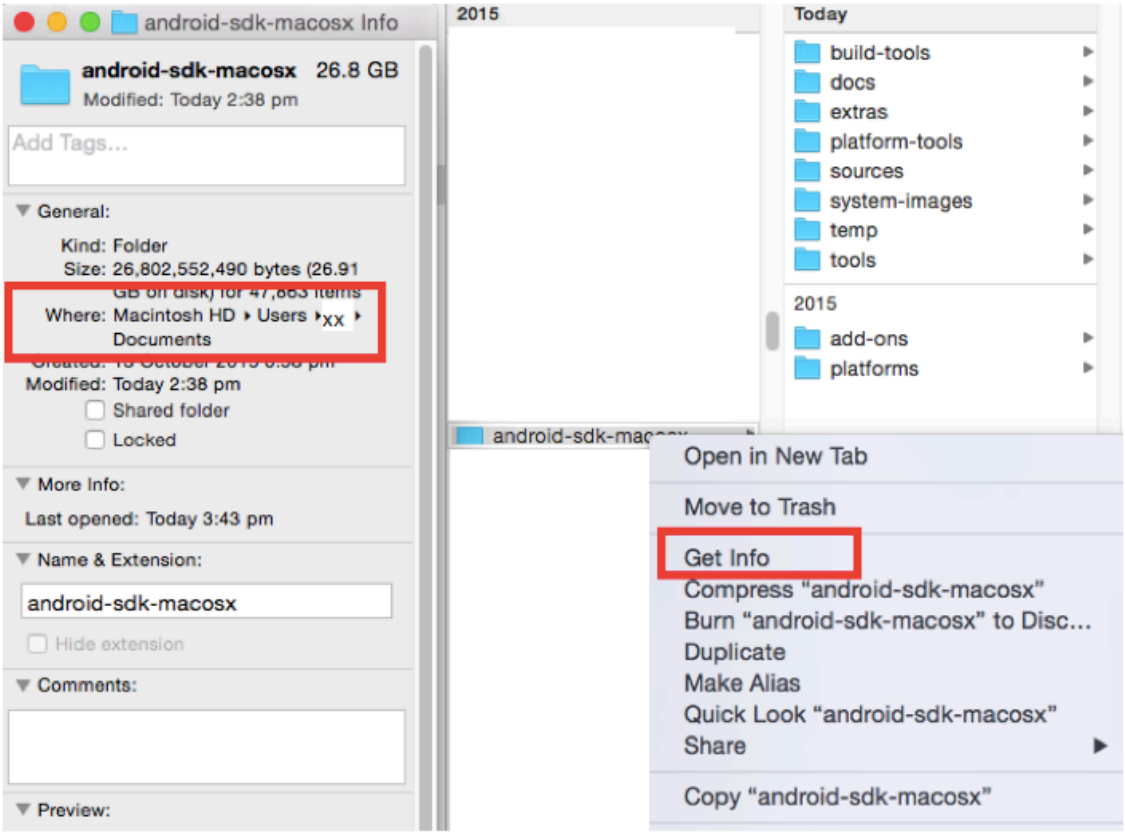
9. Ajoutez l'emplacement des nouveaux dossiers installés dans le fichier android_settings.txt qui se trouve dans le dossier d'installation du MDX Toolkit.

10. Dans **Applications > Citrix > MDX Toolkit**, ouvrez le fichier android_settings.txt et ajoutez le chemin d'accès complet des dossiers suivants :

    - **SDK Android**
    - **Android SDK > tools**
    - **Android SDK > platform-tools**
    - **Android SDK > build-tools > [version]**

    > **Remarque :**
    >
    > N'oubliez pas de supprimer le chemin **Android SDK > apktools** du fichier android_settings, car ce chemin n'est plus nécessaire.

    Pour trouver le chemin complet de votre dossier SDK, cliquez avec le bouton droit sur le fichier, sélectionnez **Get Info** puis vérifiez l'information Where dans le panneau Info.



11. Avant de modifier le fichier android_settings, faites-en une copie.

    a) Accédez à **Applications > Citrix > MDXToolkit > Android_settings**.

    b) Ajoutez les nouveaux chemins.

    c) Enregistrez le fichier en dehors du dossier **Applications > Citrix > MDX Toolkit**.

    d) Renommez le fichier android_settings d'origine dans le dossier **Applications > Citrix >**

MDXToolkit ; par exemple, android_settings.old.

e) Copiez le nouveau fichier android_settings avec les chemins ajoutés dans le dossier **Applications > Citrix > MDX Toolkit**.

L'exemple suivant affiche le fichier avec les chemins ajoutés :

```
//  For Android wrapping, not iOS
//
//  Created by Citrix Systems on 06/02/14.
//  Copyright (c) 2014 Citrix Systems, Inc. All rights reserved.
//
//  This file is intended for modifying the environment variables on
//  your machine for the purposes of Citrix application wrapping without
//  affecting the rest of your environment.
//
//  Please ensure all wrapping prerequisites are downloaded and installed
//  before continuing. A sample is given for you. Please find the correct
//  locations on your machine and add them to the path below. Please
//  separate paths with your OS specific separators,
//  i.e. (":" - Unix, ";" - Windows)
//
//  Sample Unix Path:
//  PATH =
/Users/Sample/Downloads/android-sdk-macosx/platform-tools:/Users/Sample/Downloads/android-sdk-macosx/build
-tools/19.1.0:/Users/Sample/Downloads/android-sdk-macosx/tools
//

//  To use this file, please delete the comments, "//", and append the correct
//  paths to the PATH variable below.

//PATH = /usr/bin:/usr/sbin
PATH =
/usr/bin:/usr/sbin:/usr/local/bin:/Users/TESTUSER/Documents/android-sdk-macosx/:/Users/TESTUSER/Documents/
android-sdk-macosx/tools/:/Users/TESTUSER/Documents/android-sdk-macosx/platform-tools/:/Users/TESTUSER/Doc
uments/android-sdk-macosx/build-tools/23.0.3/
```

Keystore valide

Un keystore valide contient des certificats signés numériquement que vous utilisez pour signer vos applications Android. Vous créez un keystore une fois et vous conservez ce fichier pour le réutiliser à des fins d'encapsulation. Si vous n'utilisez pas le même keystore lors de l'encapsulation des nouvelles versions d'applications que vous avez déjà déployées, les mises à niveau de ces applications ne fonctionnent pas. Les utilisateurs doivent supprimer manuellement les anciennes versions avant d'installer les nouvelles versions.

Un keystore peut contenir de multiples clés privées. Toutefois, le keystore ne contient en général qu'une clé.

Pour de plus amples informations sur les certificats, consultez la section Signing Your Applications.

Signez vos applications avec une clé répondant aux exigences suivantes :

- Taille de clé 2048 bits
- Algorithme de clé DSA ou RSA (keyalg)
- N'utilisez pas MD5.

Le MDX Toolkit signe les applications à l'aide de SHA1 pour prendre en charge les versions antérieures d'Android. Cet algorithme va bientôt être remplacé par SHA256. Si vous souhaitez signer votre application avec un autre algorithme, utilisez un autre outil.

Si vous ne souhaitez pas utiliser le debug keystore, créez un keystore. Pour créer un keystore, démarrez **Terminal** et entrez la commande suivante :

keytool -genkey -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000

Fournissez les informations demandées, telles qu'un mot de passe pour le keystore et le nom de domaine de votre organisation (exemple : exemple.com). La clé reste valide pendant 25 ans.

Pour signer une application, utilisez cette commande :

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore *clé-de-ma-version*.keystore *mon_application*.apk *nom_alias*

Vous pouvez maintenant encapsuler des applications Android. Pour plus de détails, consultez la section Encapsulation d'applications Android.

Installation du MDX Toolkit

January 23, 2019

Suivez ces étapes pour installer l'outil MDX Toolkit et le SDK XenMobile App pour iOS et Android.

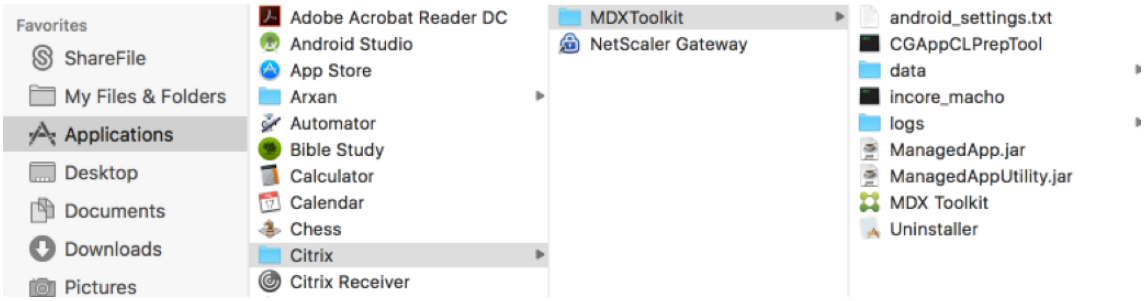
Effectuez les étapes suivantes sur un ordinateur exécutant macOS. Le programme d'installation comprend les outils suivants :

- Des outils macOS capables d'encapsuler des applications iOS et Android.
- Un outil de ligne de commande Java qui encapsule les applications Android. Vous pouvez également exécuter cet outil sur un ordinateur Windows.

> **Remarque :**
>
> supprimez la version précédente du MDX Toolkit avant d'installer la nouvelle version. Sauvegardez Android_settings.txt avant de désinstaller le Toolkit.

1. Ouvrez une session sur la page des téléchargements XenMobile.
2. Développez **Applications XenMobile et MDX Toolkit**.
3. Recherchez la version de MDX Toolkit que vous souhaitez installer, puis cliquez sur le lien pour lancer le téléchargement.
4. Ouvrez MDXToolkit.mpkg avec l'outil macOS Finder sur la version la plus récente de macOS et Xcode. Pour la version requise, consultez la section Configuration requise.

Le chemin d'installation par défaut est Applications/Citrix/MDXToolkit.



5. Si vous souhaitez exécuter l'outil de ligne de commande Java sur un ordinateur Windows, copiez ManagedApp.jar et ManagedAppUtility.jar dans un répertoire sur un ordinateur Windows qui répond aux conditions requises pour encapsuler des applications Android. Pour plus de détails, voir Configuration système requise.

6. Pour utiliser l'outil GUI pour encapsuler des applications Android, vous devez mettre à jour les informations de chemin du fichier android_settings.txt qui est installé dans Applications/Citrix/MDXToolkit. Si vous ne suivez pas ces étapes, l'outil GUI indiquera que les composants requis sont introuvables.

   > **Important :**
   >
   > lors de l'encapsulation d'applications Android, le MDX Toolkit peut échouer si les paramètres régionaux de l'ordinateur sur lequel vous exécutez l'outil MDX Toolkit ne sont pas réglés sur l'anglais.

   a) Copiez android_settings.txt dans un dossier auquel vous avez accès en écriture.

   b) Modifiez le fichier android_settings.txt avec un éditeur de texte. Pour utiliser Vim, vous pouvez utiliser la commande suivante. Entrez votre mot de passe utilisateur lorsque vous y êtes invité. Le fichier s'ouvre dans votre fenêtre de terminal.

   ```
   sudo vim /Applications/Citrix/MDXToolkit/android_settings.txt
   ```

   c) Mettez à jour le fichier avec le chemin d'accès au JDK et aux fichiers binaires SDK Android dans votre environnement.

   Ajoutez les lignes suivantes à la fin de la ligne "PATH =" dans votre fichier settings.txt (séparés par ":" sous Mac/Unix, et ";" sous Windows) :

   ```
   PATH = /bin:/usr/bin:/usr/sbin/sbin:/<Install Location> /adt-bundle-mac-x86_64-20130729/sdk:/<Install Location>/adt-bundle-mac-x86_64-20130729/sdk/tools:<Install Location>/adt-bundle-mac-x86_64-20130729/sdk/platform-tools:Documents/Android SDK/apktools
   ```

   d) Enregistrez le fichier mis à jour sous le même nom, android_settings.txt, puis copiez le fichier dans Applications/Citrix/MDXToolkit.

Vous pouvez être invité à entrer un mot de passe pour le copier sur ce dossier.

Le package d'installation contient un petit utilitaire permettant de supprimer le MDX Toolkit. L'utilitaire est installé dans l'emplacement suivant sur votre ordinateur : /Applications/Citrix/CGAppPrepTool/Uninstaller.app/Contents. Double-cliquez sur l'utilitaire pour démarrer le programme de désinstallation et suivez les instructions. Un message vous invite à entrer votre nom d'utilisateur et mot de passe lors de la suppression de l'outil.

Encapsulation des applications mobiles iOS

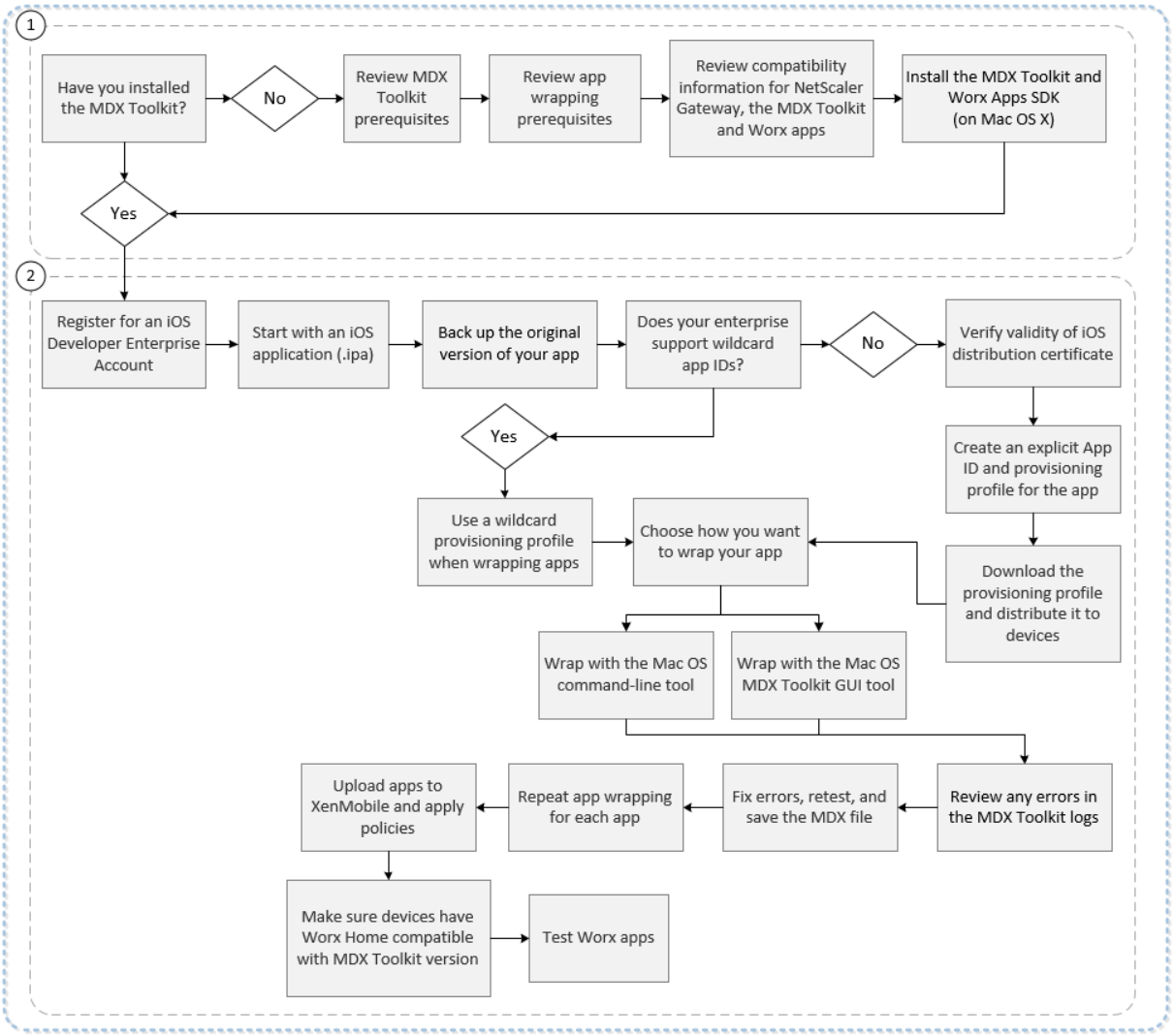
February 19, 2019

Cet article explique comment les administrateurs Citrix Endpoint Management encapsulent les applications d'entreprise tierces et comment les développeurs encapsulent les applications ISV. Pour encapsuler les applications mobiles iOS :

- Utilisez le service MDX. Pour plus d'informations, consultez la section MDX Service.
- Utilisez l'outil MDX Toolkit, qui comprend un outil d'interface graphique macOS et un outil de ligne de commande macOS. L'outil de ligne de commande macOS propose des options de personnalisation, peut être référencé depuis des scripts qui automatisent le processus d'encapsulation d'application, et vous permet de prédéfinir des stratégies MDX.

Le type de fichier pour une application encapsulée est .mdx. Vous devez charger le fichier .mdx vers la console Endpoint Management dans laquelle vous configurez ensuite les détails et les paramètres de stratégie spécifiques à l'application que Endpoint Management Store doit appliquer. Lorsque les utilisateurs ouvrent une session, l'application s'affiche dans le magasin. Les utilisateurs peuvent s'abonner, télécharger et installer l'application sur leur appareil.

La figure suivante présente les étapes d'encapsulation d'application, de l'installation de l'outil MDX Toolkit aux tests des applications de productivité mobiles. Les rubriques connexes sont répertoriées sous le diagramme.



Pour plus de détails sur la section 1, voir :

- Configuration système requise
- Autre configuration requise pour encapsuler des applications mobiles iOS
- Compatibilité Endpoint Management
- Installation du MDX Toolkit

Pour plus de détails sur la section 2, voir :

- Création de profils de provisioning
- Mises à niveau d'applications
- Stratégies et applications de productivité mobiles
- Encapsulation d'applications d'entreprise à l'aide de l'interface graphique
- Encapsulation d'applications d'entreprise iOS à l'aide de la ligne de commande
- Options de commande
- Prédéfinition de stratégies MDX pour applications iOS

- Identification des erreurs d'encapsulation des applications iOS
- Collecte des journaux système sur les appareils iOS
- Pour ajouter une application MDX à Citrix Endpoint Management

> **Important :**
>
> Assurez-vous que les appareils sont mis à jour avec une version de Secure Hub compatible avec la version du MDX Toolkit utilisé pour encapsuler les applications. Si ce n'est pas le cas, un message d'erreur d'incompatibilité s'affiche. Pour de plus amples informations, consultez la section Compatibilité Endpoint Management.

Déploiement d'appareils iOS via le programme DEP d'Apple

Inscrivez-vous au Programme de déploiement d'Apple pour pouvoir bénéficier du programme Apple Device Enrollment Program (DEP). Vous pouvez utiliser le programme DEP d'Apple pour déployer et gérer des appareils iOS et macOS dans Citrix Endpoint Management. Pour plus d'informations, notamment comment s'inscrire dans le Programme de déploiement d'Apple, consultez la section Déployer des appareils iOS et macOS via le programme DEP d'Apple.

Création de profils de provisioning

Toute application exécutée sur un appareil iOS physique, autre que des applications dans l'App Store d'Apple, doit être signée avec un profil de provisioning et un certificat correspondant. Il existe deux types de programmes de développement pour la distribution :

- iOS Developer Program (Ad-Hoc)
- iOS Developer Enterprise Program. Pour encapsuler les applications, Citrix vous recommande d'utiliser le programme Enterprise. Vous pouvez vous inscrire au programme sur le site Web d'Apple.

Le profil Enterprise vous permet d'exécuter une application sur un nombre illimité d'appareils. Le profil Ad Hoc vous permet d'exécuter une application sur un maximum de 100 appareils.

Apple ne prend plus en charge l'utilisation d'ID d'application génériques pour les nouveaux comptes Enterprise. Si votre compte Enterprise ne prend pas en charge les ID d'application génériques, vous devez créer plusieurs ID d'application explicites et profils de provisioning, comme suit.

1. Vérifiez que vous disposez d'un certificat de distribution iOS valide.
2. À partir du portail Apple Enterprise Developer, créez un ID d'application explicite pour chaque application que vous voulez encapsuler avec le MDX Toolkit. Exemple d'ID d'application acceptable : com.NomEntreprise.NomProduit.

3. À partir du portail Apple Enterprise Developer, accédez à **Provisioning Profiles > Distribution** et créez un profil de provisioning interne. Répétez cette étape pour chaque ID d'application créé à l'étape précédente.
4. Téléchargez tous les profils de provisioning.

Si votre compte Apple Enterprise prend en charge les ID d'application génériques, vous pouvez continuer à utiliser un profil de provisioning générique pour encapsuler les applications. Toutefois, si vous utilisez le service de notification push d'Apple (APNS) pour les notifications lorsque Secure Mail est exécuté en arrière-plan, vous devez utiliser un profil de provisioning et un ID App explicites.

Tout appareil sur lequel vous voulez installer l'application MDX doit disposer du profil de provisioning. Vous pouvez distribuer le profil auprès des appareils utilisateur en tant que pièce jointe à un e-mail. Les utilisateurs peuvent ajouter le profil sur leur appareil iOS en cliquant sur la pièce jointe.

Pour plus de détails sur les profils de provisioning et les certificats de distribution, consultez l'aide relative au compte de développeur Apple.

Pour plus d'informations sur le déploiement de profils d'approvisionnement sur des appareils iOS et sur la gestion des profils expirés, consultez l'article Endpoint Management dans Stratégie de profil de provisioning.

Mises à niveau d'applications

> Important :
>
> Avant d'effectuer la mise à niveau des applications, tenez compte de la façon dont les modifications apportées aux ID d'application ou l'utilisation d'un profil de provisioning d'ID d'application génériques partiels peuvent avoir un impact sur les mises à niveau des applications.

- Les applications encapsulées précédemment sont mises à niveau sur place, à moins que l'ID de l'application ait été modifié. Par exemple, si vous modifiez un Bundle ID de com.citrix.mail à com.exemple.mail, il n'existe pas de chemin de mise à niveau. L'utilisateur doit réinstaller l'application. Un appareil considère l'application en tant que nouvelle application. La nouvelle version et les versions antérieures de l'application peuvent résider sur un même appareil.
- Si vous utilisez un profil de provisioning partiel, tel que com.xxxx, pour encapsuler une application avec un Bundle ID qui inclut com.citrix, nous vous recommandons de supprimer les applications MDX encapsulées installées et d'installer les applications encapsulées avec le MDX Toolkit le plus récent. Si le Bundle ID com.citrix.mail devient com.exemple, les utilisateurs devront réinstaller l'application.
- Une mise à niveau sur place réussit si une application a été encapsulée avec un ID d'application générique complet et que la nouvelle version de l'application a un ID d'application qui correspond à l'application installée.

Stratégies et applications de productivité mobiles

> **Remarque :**
>
> Le MDX Toolkit 10.7.5 est la dernière version qui prend en charge l'encapsulation des applications de productivité mobiles. Vous ne pouvez pas utiliser les versions MDX 10.7.10 et ultérieures, ni le service MDX, pour encapsuler les applications de productivité mobiles version 10.7.5 ou ultérieure. Vous devez accéder aux applications de productivité mobiles à partir de magasins d'applications publics.

Citrix fournit un ensemble générique de stratégies par défaut qui s'appliquent à toutes les applications de productivité mobiles et un ensemble de stratégies spécifiques pour certaines des applications de productivité mobiles. Les noms des fichiers de stratégie sont basés sur le Bundle ID. Par défaut, le nom de fichier de stratégie pour une application Citrix Endpoint Management est au format com.citrix.app_policy_metadata.xml., où app est un nom tel que « mail ».

Si vous disposez d'un compte Apple Enterprise qui ne prend pas en charge les ID d'application génériques, vous devez changer l'identificateur d'entreprise dans le Bundle ID lorsque vous encapsulez Endpoint Management. Par exemple, le Bundle ID pour Secure Mail est com.citrix.mail. Dans cet identificateur, remplacez « Citrix » par l'identificateur de votre entreprise. Si l'identificateur de votre entreprise est « exemple », le Bundle ID est com.exemple.mail. Lorsque vous encapsulez cette application, le nom du fichier de stratégie est com.exemple.mail_policy_metadata.xml.

Pour déterminer quel fichier de stratégie à appliquer à une application, le MDX Toolkit recherche les fichiers dans l'ordre suivant et utilise le premier fichier qu'il trouve :

1. Un nom de fichier qui correspond à votre Bundle ID, tel que com.exemple.mail_policy_metadata.xml., comme décrit dans l'exemple précédent.
2. Un nom de fichier qui correspond au Bundle ID d'origine, tel que com.citrix.mail_policy_metadata.xml.
3. Un nom de fichier qui correspond au fichier de stratégie par défaut générique, policy_metadata.xml.

Créez votre propre ensemble de stratégie par défaut pour une application Citrix Endpoint Management en modifiant les fichiers correspondant à votre Bundle ID ou au Bundle ID d'origine.

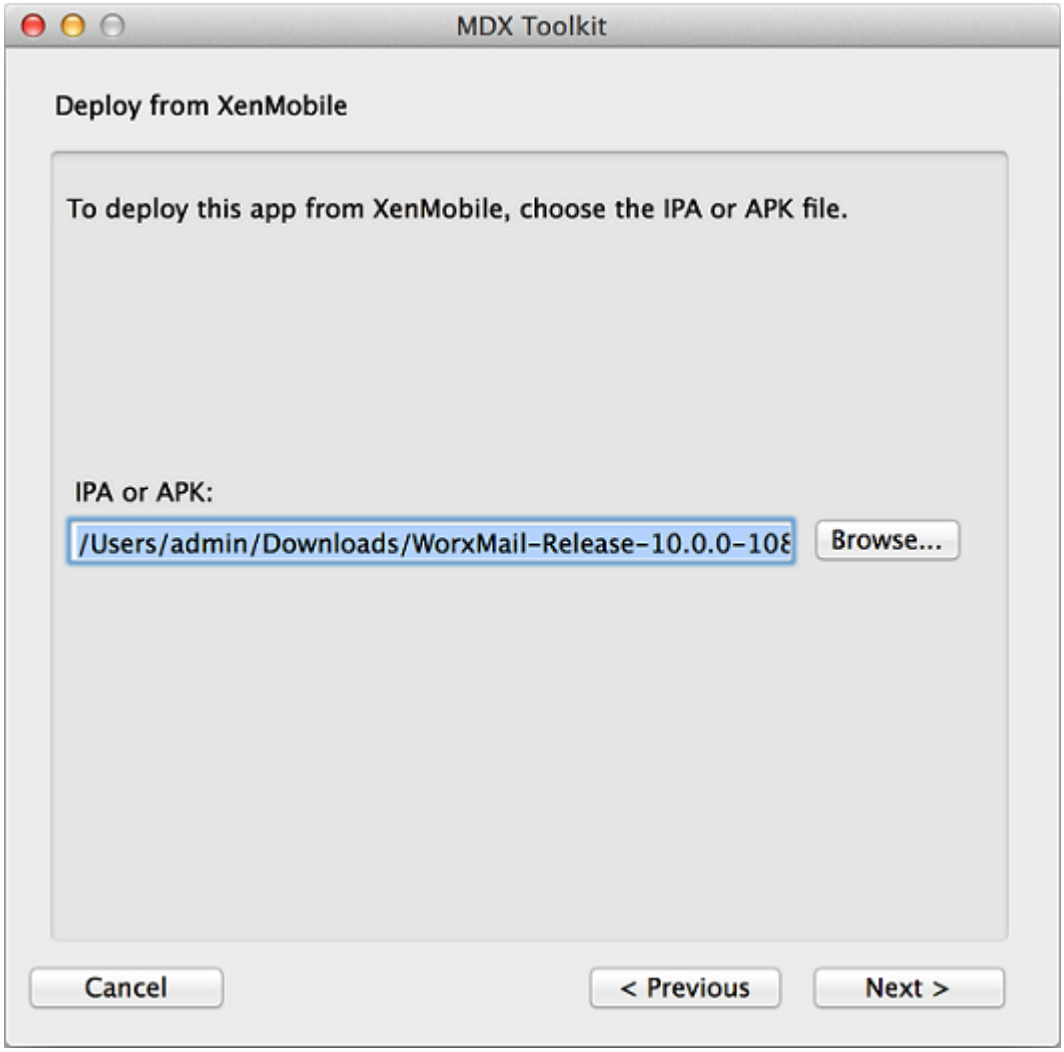
Encapsulation d'applications d'entreprise à l'aide de l'interface graphique

Les étapes suivantes décrivent la procédure générale à suivre pour l'encapsulation d'applications d'entreprise que vous déployez à partir de Endpoint Management. La procédure générale pour encapsuler des applications d'entreprise est décrite dans la section Encapsulation d'applications ISV à l'aide de l'interface graphique.

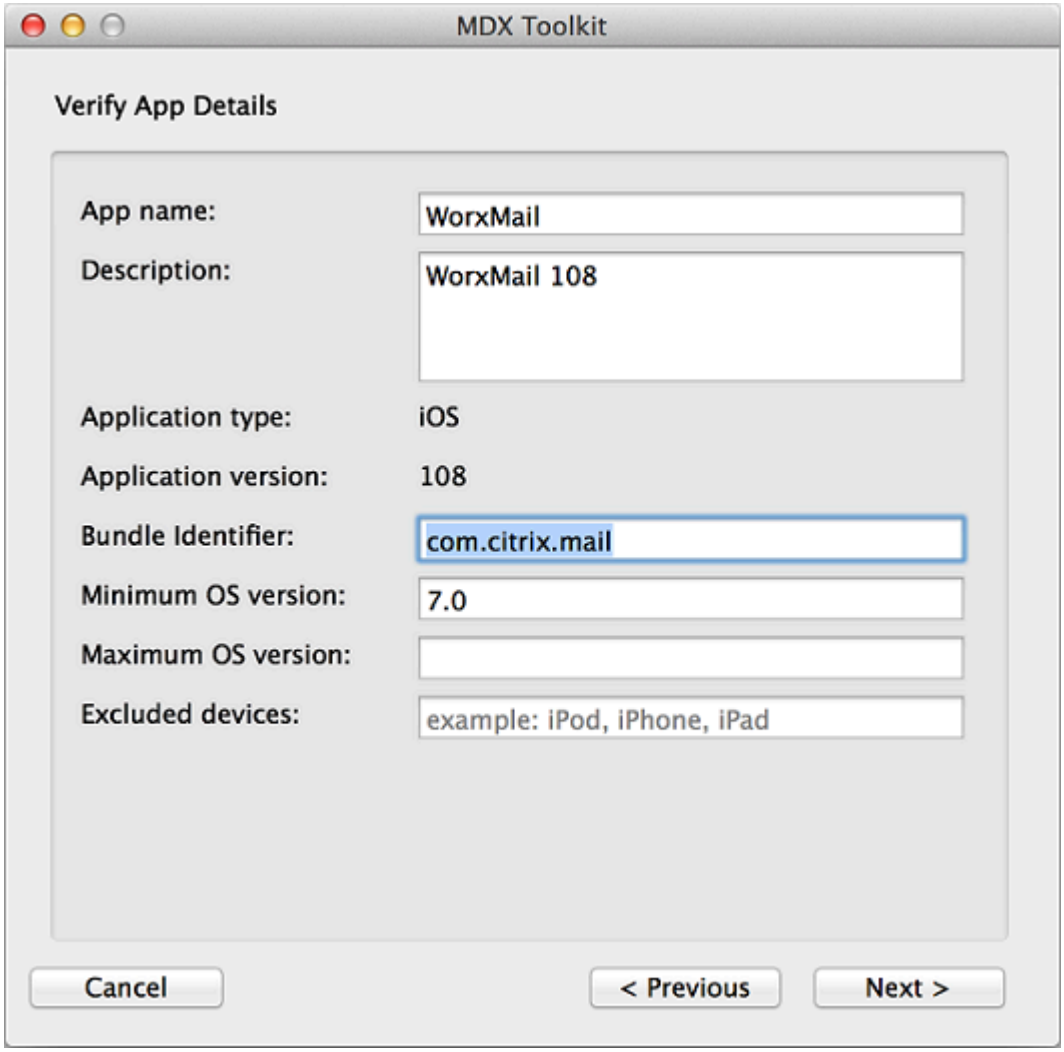
> Important :
>
> la clé privée et le certificat doivent être installés sur le trousseau d'accès de votre Mac avant d'utiliser l'interface graphique pour encapsuler des applications iOS. Si le certificat de distribution associé ne dispose pas de la clé privée installée dans le trousseau d'accès, l'interface graphique ne pré-remplit pas la liste iOS Distribution Certificate.
> Pour de plus amples informations, consultez la section « Réparation de votre trousseau lorsque le Toolkit ne peut pas trouver le certificat de distribution » dans cet article.

1. Avant d'utiliser le Toolkit pour encapsuler des applications, assurez-vous de sauvegarder la version d'origine de ces applications de façon à ce que vous puissiez y revenir si nécessaire.
2. Démarrez l'outil MDX Toolkit depuis votre dossier iOS Applications, sélectionnez **For IT administrators** (Pour les administrateurs informatiques), puis cliquez sur **Next** (Suivant).

3. Cliquez sur **Browse** (Parcourir), sélectionnez le fichier, puis cliquez sur **Next** (Suivant).



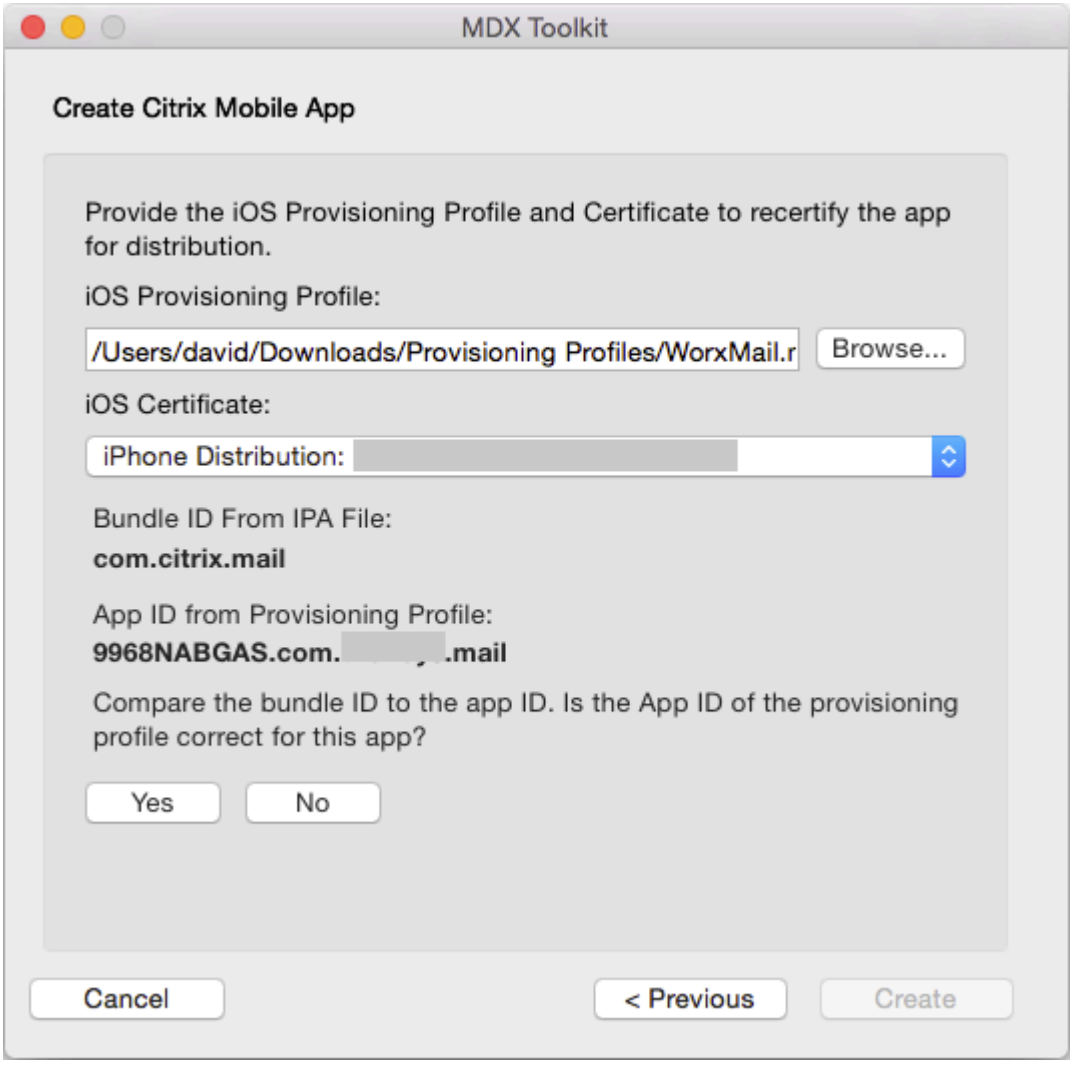
4. L'écran **Verify App Details** (Vérifier les détails de l'application) affiche les informations obtenues à partir de l'application. Si nécessaire, modifiez les informations pré-remplies. Si vous le souhaitez, vous pouvez spécifier des valeurs minimales et maximales pour la version du système d'exploitation et répertorier les types d'appareil sur lesquels l'application n'est pas autorisée à être exécutée. Vous pouvez également modifier les détails de l'application après le chargement de l'application dans Citrix Endpoint Management.



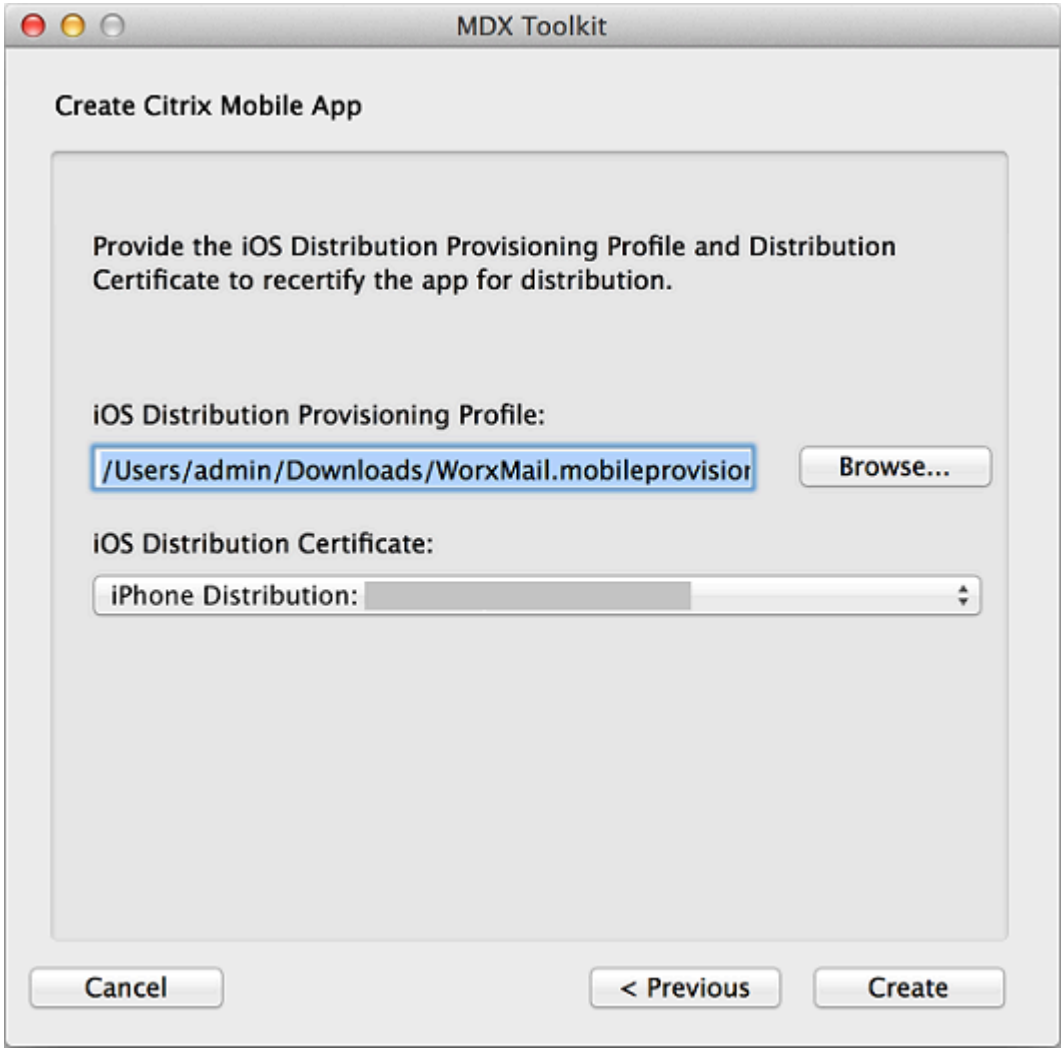
5. Dans l'écran **Create Citrix Mobile App** (Créer une application mobile Citrix), cliquez sur **Browse**, sélectionnez le profil de provisioning et sélectionnez un certificat de distribution. Si la liste iOS Certificate est vide, il se peut que vous deviez réparer le trousseau sur la machine sur laquelle vous exécutez l'outil MDX Toolkit. Pour de plus amples informations, consultez la section « Réparation de votre trousseau lorsque le Toolkit ne peut pas trouver le certificat de distribution » dans cet article.

6. Si vous avez sélectionné un profil de provisioning qui dispose d'un ID d'application explicite, l'outil vous invitera à confirmer l'ID d'application. Par exemple, le Bundle ID d'une application Citrix Endpoint Management est com.citrix.NomProduit. Le profil de provisioning que vous utilisez doit inclure l'identificateur de votre entreprise au lieu de « citrix ».

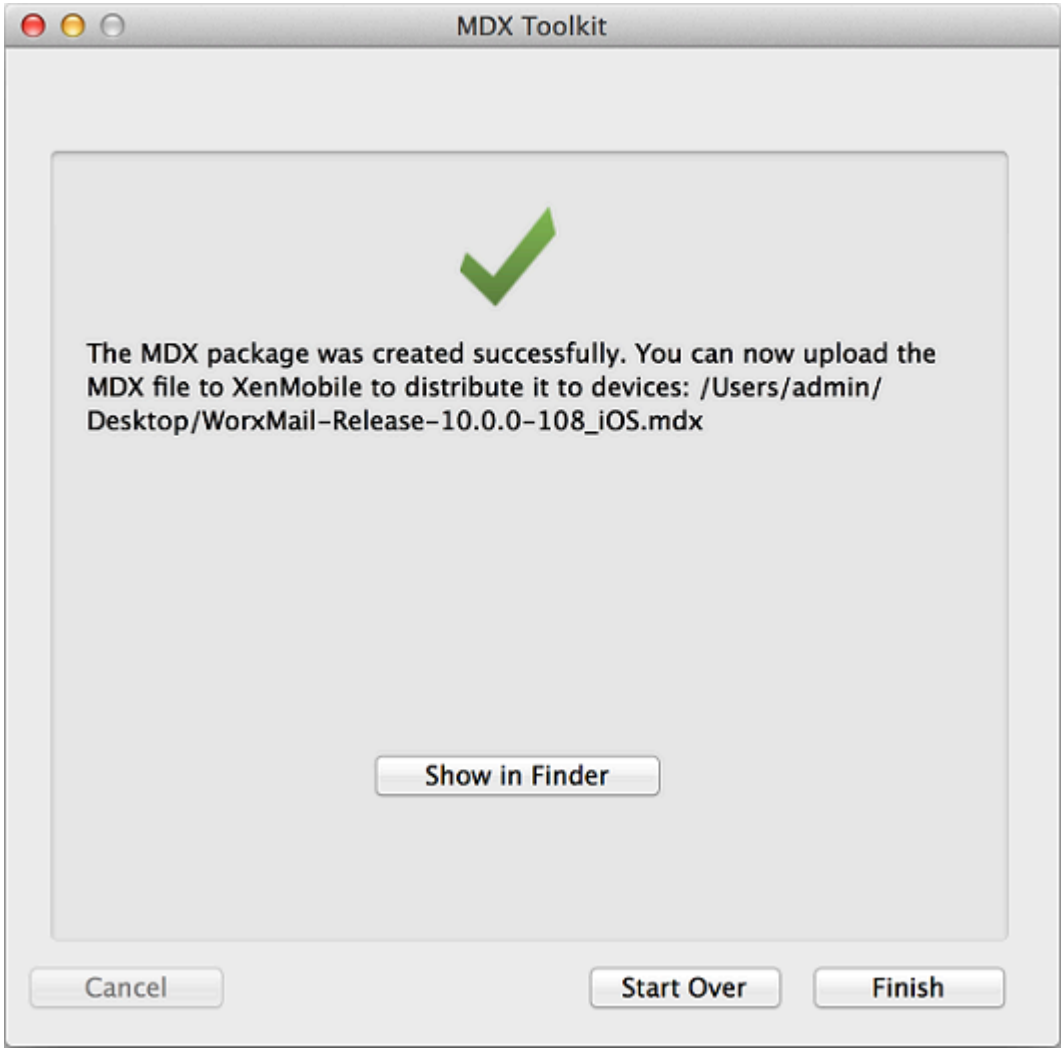
   Après avoir cliqué sur **Yes** (Oui), cliquez sur **Create** (Créer).



7. Si vous avez sélectionné un profil de provisioning qui dispose d'un ID d'application générique, l'outil affiche une liste d'ID d'application disponibles. Si l'ID d'application que vous souhaitez utiliser n'est pas répertorié, choisissez un autre profil de provisioning. Après avoir choisi un ID d'application, cliquez sur **Create** (Créer).



8. L'outil vous informe lorsque le package MDX a été créé. Pour encapsuler une autre application, cliquez sur **Start Over** (Redémarrer).



L'outil ajoute _iOS à la fin du nom de fichier de l'application iOS encapsulée.

Encapsulation d'applications d'entreprise iOS à l'aide de la ligne de commande

> **Remarque :**
>
> Achetez vos applications tierces directement auprès du fournisseur. En effet, les applications iOS téléchargées depuis l'Apple Store sont cryptées et ne peuvent pas être encapsulées.

Avant d'utiliser le Toolkit pour encapsuler des applications, assurez-vous de sauvegarder la version d'origine de ces applications de façon à ce que vous puissiez y revenir si nécessaire.

L'exemple suivant illustre une commande d'encapsulation d'application de base utilisant les paramètres par défaut. Modifiez les informations en gras pour votre système. La barre oblique inverse de fin indique que la commande continue à la ligne suivante. Supprimez ces symboles avant d'exécuter la commande.

Pour exécuter ces commandes, accédez au répertoire /Applications/Citrix/MDXToolkit/ depuis la ligne de commande.

Une ligne de commande d'encapsulation iOS de base se présente comme suit.

```
./CGAppCLPrepTool \
Wrap \
 -Cert CERTIFICATE \
 -Profile PROFILE \
-bundleID ID \
 -in INPUT_FILE \
 -out OUTPUT_FILE
```

Ce qui suit est un exemple de cette option de ligne de commande.

```
./CGAppCLPrepTool \
Wrap \
 -Cert  "iPhone Developer: Joe Admin (12MMA4ASQB)"  \
 -Profile  "team_profile.mobileprovision"  \
-bundleID  "com.CompanyABC.Sample"  \
 -in  "~/Desktop/SampleApps/Sample.ipa"  \
 -out  "~/Desktop/SampleApps/Sample.mdx"
```

Les exemples d'options que vous pouvez ajouter à la commande précédente comprennent :

```
-appName  "Wrapped Sample app"
```

```
-appDesc  "This is my newly wrapped iOS application."
```

Ces deux options se règlent par défaut sur la valeur obtenue depuis l'application, si possible.

Pour de plus amples informations sur les options disponibles, consultez la section Options de commande. Pour la documentation en ligne, utilisez l'option –help.

Options de commande

Commande wrap

- **Help:** affiche l'aide de cette commande.
- **In:** obligatoire. Chemin d'accès et nom de fichier de l'application en cours d'encapsulation.
- **Out:** facultatif. Chemin d'accès et nom de fichier pour le fichier .mdx. Si cette option est omise, le fichier porte le même chemin d'accès et nom de fichier que le fichier d'entrée, avec une extension .mdx.
- **Outbundle:** obligatoire lors de la génération d'un fichier .ipa pour le téléchargement dans Intune. Chemin d'accès et nom de fichier pour le fichier .ipa.

- **Cert:** obligatoire. Nom du certificat à utiliser pour signer l'application.
- **Profile:** obligatoire. Nom du profil de provisioning à utiliser pour signer l'application.
- **bundleID:** requis pour les comptes Enterprise qui ne prennent pas en charge les ID d'application génériques. Il s'agit de votre Bundle ID Apple. Le MDX Toolkit vérifie si le Bundle ID et le profil de provisioning sont compatibles.
- **Upgrade:** cette option est destinée aux applications d'ancienne génération et sera bientôt obsolète. Utilisée pour les mises à niveau sur place lorsque vous utilisez un profil de provisioning générique partiel. Cette option garantit que le nouveau fichier binaire est signé avec les mêmes droits que la version antérieure. Si les droits ne correspondent pas, les tentatives d'installation des mises à niveau à partir de Secure Hub échouent.
- **AppName:** facultatif. Nom de l'application, obtenu depuis l'application, si possible.
- **AppDesc:** facultatif. Description de l'application, obtenue depuis l'application, si possible.
- **MinPlatform:** facultatif. Version minimale de plate-forme prise en charge. Valeur par défaut : vide.
- **MaxPlatform:** facultatif. Version maximale de plate-forme prise en charge. Valeur par défaut : vide.
- **ExcludedDevices:** facultatif. Liste des types d'appareil sur lesquels l'application n'est pas autorisée à être exécutée. Valeur par défaut : vide.
- **PolicyXML:** facultatif. Fichier de définition et chemin d'accès de stratégie XML de remplacement. Valeur par défaut : définitions de stratégie intégrées. Exemple : -policyxml /Applications/Citrix/MDXToolkit/data/policy_metadata.xml. Pour de plus amples informations, veuillez consulter la section suivante, « Prédéfinition de stratégies MDX pour applications iOS ».
- **useNetworkOnlylib:** cette option encapsule l'application avec la version plus légère de la bibliothèque dynamique MDX uniquement sur le réseau. Une application encapsulée à l'aide de cette option ne peut être gérée que par Intune ou exécutée en mode non géré. Elle ne peut pas être gérée par MDX.
- **LogFile:** facultatif. Nom du fichier journal.
- **LogWriteLevel:** facultatif. Niveau de journalisation, 1 à 4.
- **LogDisplayLevel:** facultatif. Niveau de journalisation pour sortie standard, 0 à 4.

commande sign

- **Help:** affiche l'aide de cette commande.
- **In:** obligatoire. Chemin d'accès et nom de fichier de l'application en cours d'encapsulation.
- **Out:** facultatif. Chemin d'accès et nom de fichier pour le fichier .mdx. Si cette option est omise, le fichier porte le même chemin d'accès et nom de fichier que le fichier d'entrée, avec une extension .mdx.
- **Cert:** obligatoire. Nom du certificat à utiliser pour signer l'application.
- **Profile:** obligatoire. Nom du profil de provisioning à utiliser pour signer l'application.

commande setinfo

- **Help:** affiche l'aide de cette commande.
- **In:** obligatoire. Chemin d'accès et nom de fichier de l'application à modifier.
- **Out:** pour setinfo, le nom de fichier ou le chemin de sortie doit être différent de celui d'origine.
- **AppDesc:** facultatif. Description de l'application. La valeur reste inchangée si elle n'est pas spécifiée.
- **MinPlatform:** facultatif. Niveau de SDK minimal pris en charge. La valeur reste inchangée si elle n'est pas spécifiée.
- **MaxPlatform:** facultatif. Niveau de SDK maximal pris en charge. La valeur reste inchangée si elle n'est pas spécifiée.
- **ExcludedDevices:** facultatif. Liste des types d'appareil sur lesquels l'application n'est pas autorisée à être exécutée. La valeur reste inchangée si elle n'est pas spécifiée.
- **StoreURL:** facultatif. Adresse URL de l'application dans le magasin d'applications. La valeur reste inchangée si elle n'est pas spécifiée.
- **PolicyXML:** facultatif. Fichier de définition et chemin d'accès de stratégie XML de remplacement. Valeur par défaut : définitions de stratégie intégrées. Exemple : -policyxml /Applications/Citrix/MDXToolkit/data/policy_metadata.xml. Pour de plus amples informations, veuillez consulter la section suivante, « Prédéfinition de stratégies MDX pour applications iOS ».

Prédéfinition de stratégies MDX pour applications iOS

Pour les applications que vous encapsulez avec l'outil de ligne de commande MDX Toolkit, vous pouvez prédéfinir certaines stratégies MDX. Vous pouvez également configurer des stratégies dans la console Citrix Endpoint Management lorsque vous ajoutez les applications.

1. Mettez à jour les valeurs de stratégie dans le fichier de stratégie XML.

   Le programme d'installation de MDX Toolkit crée ce fichier de stratégie : Applications/Citrix/MDXToolkit/data/policy_metadata.xml

   > **Remarque :**
   >
   > Les fichiers de stratégie iOS diffèrent des fichiers Android. Pour prédéfinir des stratégies pour ces deux plates-formes, vous devez mettre à jour leurs fichiers de stratégie XML respectifs.

2. Lorsque vous encapsulez l'application à l'aide de la ligne de commande, vous devez inclure

   ```
   -policyxml /Applications/Citrix/MDXToolkit/data/policy_metadata.xml
   ```


Identification des erreurs d'encapsulation des applications iOS

Si vous rencontrez une erreur lors de l'encapsulation d'une application iOS, vous pouvez utiliser les journaux de l'outil MDX Toolkit pour identifier l'erreur. vous devez disposer des droits d'administrateur pour afficher les journaux de MDX Toolkit.

Lorsque vous exécutez le MDX Toolkit, l'outil enregistre un fichier journal dans l'emplacement suivant : **Applications > Citrix > MDXToolkit > Logs > Citrix.log**. Par défaut, l'outil enregistre les avertissements et les erreurs dans le journal.

Si une erreur se produit pour une application iOS, une ligne de commande avec des arguments s'affiche à la fin du journal. Vous pouvez copier la ligne de commande et l'exécuter dans le Terminal. Pour ce faire, dans **Applications > Utilities**, cliquez sur **Terminal** et utilisez l'interface de ligne de commande Mac pour évaluer la commande. Vous devrez peut-être vous reporter aux exigences de l'application pour évaluer l'erreur.

Lorsque vous utilisez l'outil de ligne de commande pour exécuter le processus d'encapsulation, vous pouvez spécifier les informations suivantes dans la ligne de commande : l'emplacement du fichier journal, le niveau d'affichage du journal et le niveau d'écriture dans le journal. Vous pouvez également spécifier le niveau de journalisation détaillée et un autre fichier journal dans la ligne de commande.

Sélection du profil de provisioning approprié

Lorsque vous encapsulez une application mobile iOS, vous recevrez peut-être un avertissement indiquant que l'application a été encapsulée avec succès, mais qu'elle peut contenir des erreurs. Des erreurs peuvent se produire si le profil de provisioning que vous avez choisi diffère du profil de provisioning que l'application a utilisé.

MDX Toolkit peut identifier certains problèmes relatifs au profil de provisioning. Par exemple, votre application peut nécessiter une ou plusieurs des fonctions suivantes :

- Application iCloud permettant d'utiliser le stockage de données iCloud pour votre application iOS
- Notification d'émission qui utilise le service de notification push d'Apple pour délivrer des messages à l'appareil iOS
- Droits d'accès spéciaux à keychain-access-groups afin d'accéder à l'élément de trousseau d'une autre application

Les journaux affichent les paires clé-valeur absentes de l'application. Pour chaque paire clé-valeur, vous pouvez décider si vous souhaitez corriger l'erreur. Si vous ne corrigez pas l'erreur, l'application risque de ne pas fonctionner correctement. De plus, en fonction de la paire clé-valeur, vous devez vérifier si vous pouvez corriger votre profil de provisioning. Il peut parfois arriver que vous ne soyez pas en mesure de corriger le profil de provisioning, mais vous pouvez publier l'application avec ce problème.

Pour de plus amples informations sur les profils de provisioning, consultez le site Apple Developer.

Réparation de votre trousseau lorsque le Toolkit ne peut pas trouver le certificat de distribution

Si le MDX Toolkit ne reconnaît pas votre certificat de distribution iOS, il existe peut-être un problème entre votre trousseau iCloud et le trousseau sur l'ordinateur exécutant le MDX Toolkit. Pour réparer votre trousseau local, suivez ces étapes.

1. Sur votre Mac, dans Préférences système, appuyez sur **iCloud**.
2. Désactivez la case à cocher Trousseau.

   Cette action supprime les trousseaux synchronisés localement de votre iCloud.
3. Ouvrez **Trousseaux d'accès**, qui se trouve dans le dossier Utilities du dossier Applications.
4. Supprimez le certificat de développeur iOS utilisé pour signer vos applications encapsulées. Il s'agit généralement du certificat « iPhone Distribution: Company Name » avec une clé privée associée.
5. Depuis le menu Trousseaux d'accès, choisissez S.O.S. Trousseau.
6. Dans la boîte de dialogue S.O.S. Trousseau, appuyez sur **Réparer**, puis sur **Démarrer**.
7. Une fois la réparation terminée, appuyez sur **Vérifier** et sur **Démarrer**.
8. Si la réparation réussit, réimportez votre certificat de distribution iOS dans l'application Trousseaux d'accès.
9. Démarrez l'outil MDX Toolkit. Les champs iOS Distribution Provisioning Profile et iOS Distribution Certificate doivent contenir vos informations.
10. Le cas échéant, resynchronisez votre trousseau avec iCloud : Dans Préférences système, appuyez sur iCloud, puis sélectionnez la case à cocher Trousseau.

Réattribution d'applications contenant le SDK Worx

Si vous disposez d'une application sur laquelle le SDK Worx a déjà été intégré l'aide de Xcode, il vous suffit de la resigner avec votre certificat d'entreprise ou profil de provisioning. Ce qui suit est un exemple de commande **Sign**.

```
1  $ /Applications/Citrix/MDXToolkit/CGAppCLPrepTool Sign -help
2
3  Command Line Interface for MDX Toolkit, version 10.4.1.290 (Env:
       Test)
4
5  2016-09-29 15:21:45.284 CGAppCLPrepTool[88453:5477658]
```



```
6
7       ------------------------------------------------------
8
9       Sign Command
10
11      ------------------------------------------------------
12
13      CGAppCLPrepTool Sign -in INPUTFILE -out OUTPUTFILE -Cert
            CERTIFICATE -Profile PROFILE
14
15      -Cert CERTIFICATE        ==> (Required)Name of the certificate to
            sign the app with
16
17      -Profile PROFILE         ==> (Required)Name of the provisioning
            profile to sign the app with
18
19      -in INPUTFILE            ==> (Required)Name of the input app file,
            ipa/mdx file
20
21      -out OUTPUTFILE          ==> (Optional)Name of the output app, ipa(
            if ipa is input)/mdx file
22
23      -upgrade                 ==> (Optional)Preserve in-place upgrade
            capabilty (not recommended for new apps)
24
25      ---------------EXAMPLE--------------------
26
27      Sign -Cert "iPhone Distribution: Company Name" -Profile "
            distributionprovisioanl.mobileprovision" -in "/Users/user1/
            Archives/citrix.ipa"
```

Collecte des journaux système sur les appareils iOS

Vous pouvez collecter les journaux système sur les appareils iOS en utilisant l'outil iPhone Configuration Utility ou Xcode. Vous pouvez ensuite envoyer par e-mail les fichiers à l'assistance Citrix pour vous aider à résoudre les problèmes liés aux applications.

Pour utiliser un outil Configuration Utility afin de collecter les journaux système sur des appareils iOS

1. Téléchargez et installez l'outil Apple Configurator (anciennement iPhone Configuration Utility) depuis Apple. Vous pouvez utiliser l'outil sur l'iPhone et l'iPad.

2. Assurez-vous que votre appareil répond à la configuration système requise et prend en charge les langues adéquates.
3. Exécutez le programme d'installation et suivez les invites de l'assistant.
4. Ouvrez l'outil Configurator.
5. Sous Devices, cliquez sur votre appareil.
6. Cliquez sur **Console**, puis cliquez sur **Clear** pour effacer les journaux existants.
7. Reproduisez le problème, cliquez sur **Save Console As**, puis envoyez par e-mail les journaux à l'assistance technique.

Pour utiliser Xcode pour collecter les journaux sur des appareils iOS

1. Sur le Mac, cliquez sur **Finder**, cliquez sur **Aller**, puis sur **Utilitaires**.
2. Dans le dossier **Utilitaires**, double-cliquez sur **Console**.
3. Dans la console, sous **Appareils**, cliquez sur l'appareil iOS à partir duquel vous souhaitez afficher les journaux de la console.
4. Reproduisez le problème.
5. Dans la console, effectuez l'une des opérations suivantes :
   - Dans la fenêtre principale, sélectionnez le message d'erreur récent.
   - Dans la barre de **menus**de la console, cliquez sur **Modifier**, puis cliquez sur **Tout sélectionner**.
6. Cliquez sur **Modifier**, puis sur **Copier**.
7. Ouvrez TextEdit, puis collez les journaux que vous avez copiés dans un nouveau fichier.
8. Joignez le fichier dans votre e-mail au service d'assistance.

Encapsulation d'applications mobiles Android

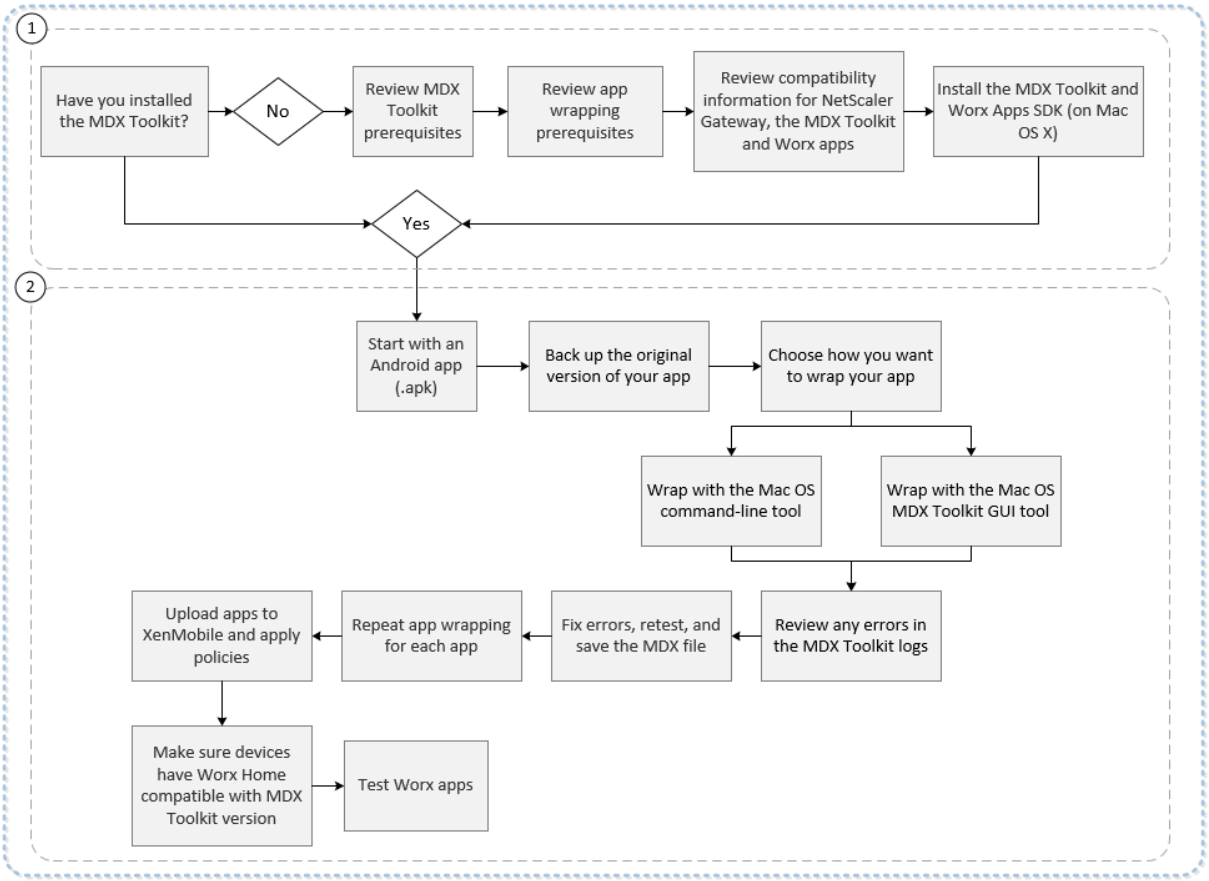
January 23, 2019

Cet article explique comment les administrateurs Citrix Endpoint Management encapsulent les applications d'entreprise tierces et comment les développeurs encapsulent les applications ISV. Pour encapsuler des applications mobiles Android, utilisez l'outil MDX Toolkit, qui comprend un outil d'interface graphique macOS et un outil de ligne de commande Java. L'outil de ligne de commande propose des options de personnalisation, peut être référencé depuis des scripts qui automatisent le processus de d'encapsulation d'application, et vous permet de prédéfinir des stratégies MDX.

Le type de fichier pour une application encapsulée est .mdx. Vous devez charger le fichier .mdx vers la console Endpoint Management dans laquelle vous configurez ensuite les détails et les paramètres

de stratégie spécifiques à l'application que Endpoint Management Store doit appliquer. Lorsque les utilisateurs ouvrent une session, l'application s'affiche dans le magasin d'applications. Les utilisateurs peuvent s'abonner, télécharger et installer l'application sur leur appareil.

La figure suivante présente les étapes d'encapsulation d'application, de l'installation de l'outil MDX Toolkit aux tests des applications de productivité mobiles. Les rubriques connexes sont répertoriées sous le diagramme.



Pour plus de détails sur la section 1, voir :

- Configuration système requise
- Autre configuration requise pour encapsuler des applications mobiles Android
- Compatibilité Endpoint Management
- Installation du MDX Toolkit

Pour plus de détails sur la section 2, voir :

- Encapsulation d'applications Android ISV à l'aide de la ligne de commande
- Encapsulation d'applications d'entreprise Android à l'aide de la ligne de commande
- Options de commande
- Prédéfinition de stratégies MDX pour applications Android
- Identification des erreurs d'encapsulation des applications Android

- Collecte des journaux d'application à partir de la ligne de commande
- Ajouter une application MDX

> **Important :**
>
> Assurez-vous que vos appareils sont mis à jour avec une version de Secure Hub compatible avec la version du MDX Toolkit utilisé pour encapsuler les applications. Si ce n'est pas le cas, les utilisateurs reçoivent un message d'erreur d'incompatibilité. Pour de plus amples informations, consultez la section Compatibilité Endpoint Management.

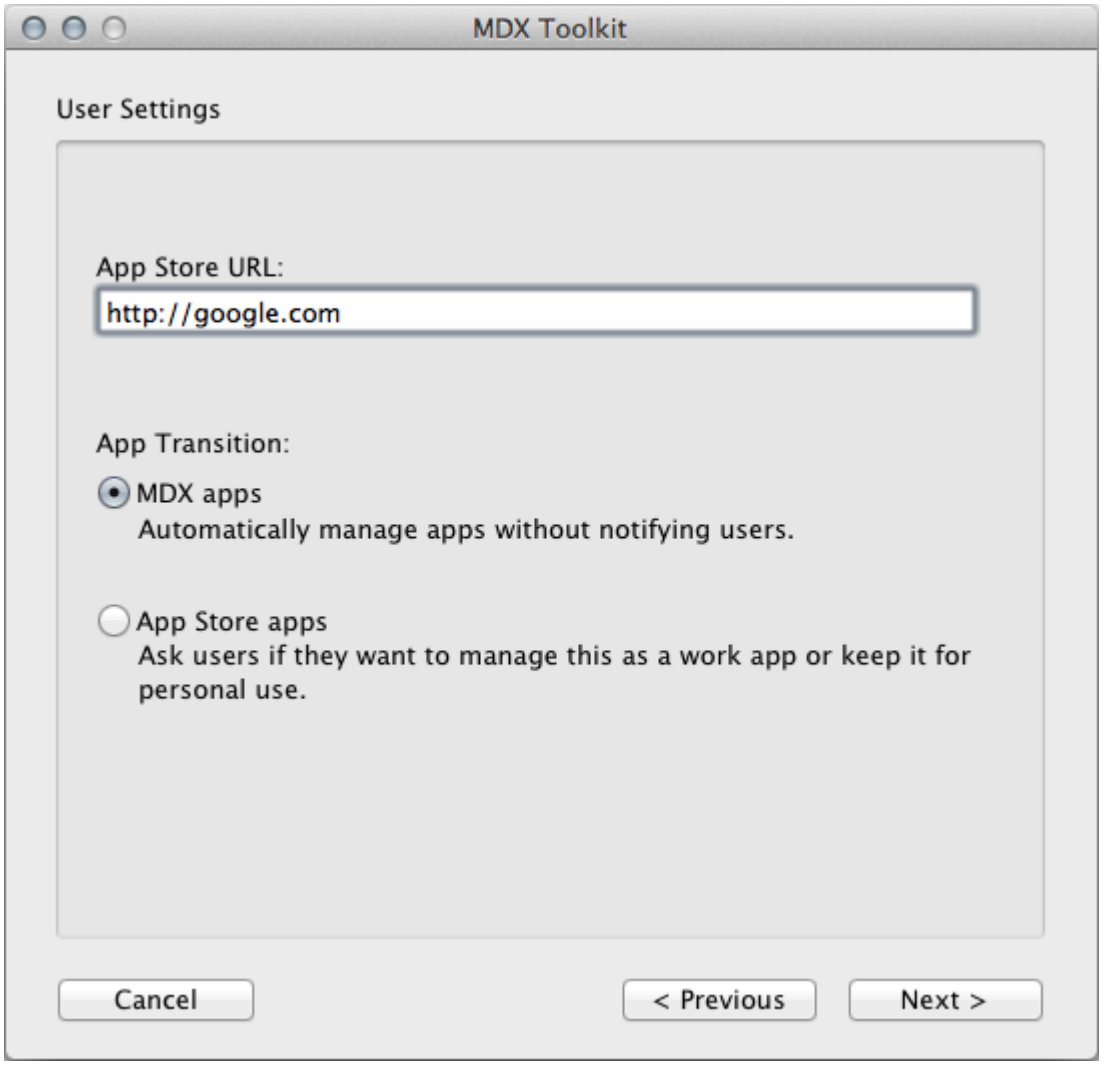
Encapsulation d'applications ISV à l'aide de l'interface graphique

Les étapes suivantes décrivent la procédure générale à suivre pour l'encapsulation d'applications ISV que vous déployez à partir de Google Play Store.

1. Avant d'utiliser le Toolkit pour encapsuler des applications, assurez-vous de sauvegarder la version d'origine de ces applications de façon à ce que vous puissiez y revenir si nécessaire.
2. Démarrez l'outil MDX Toolkit depuis votre dossier iOS Applications, sélectionnez **For Independent Software Vendors (ISVs)** (Pour les fournisseurs de logiciels indépendants (ISV)), puis cliquez sur **Next**.

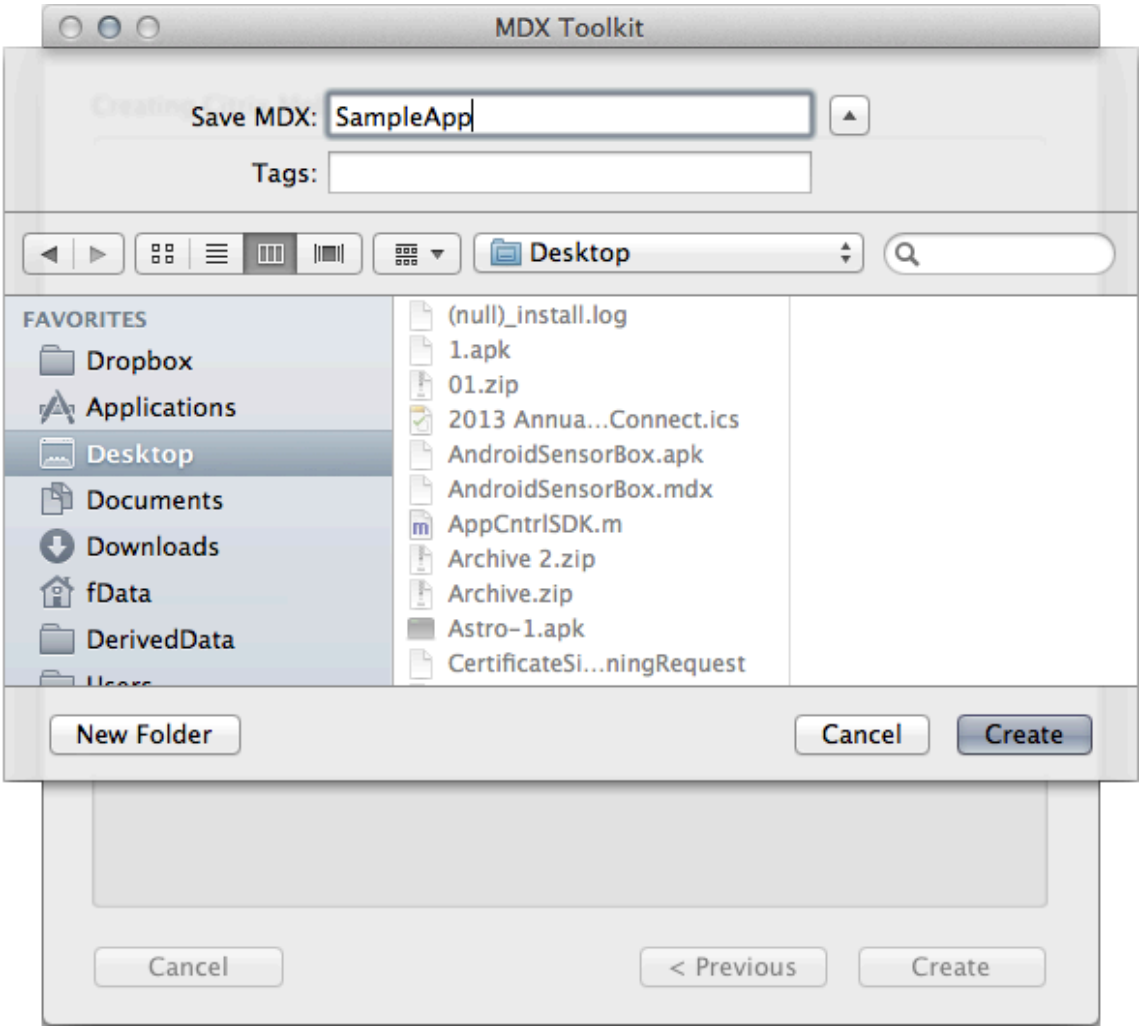
3. Dans l'écran **Deploy from App Store**, sélectionnez l'application et cliquez sur **Next**.
4. Dans l'écran **User Settings**, si vous disposez déjà de l'adresse URL du magasin d'applications, entrez-la. Si vous n'avez pas l'URL, entrez un espace réservé tel que `https://play.google.com/store/apps/details?id=com.citrix`. Vous pouvez mettre à jour l'adresse URL ultérieurement.

   Pour les applications Premium, sélectionnez **MDX apps**. Pour les applications General, sélectionnez **App Store apps**.



5. Dans l'écran **Verify App Details** (Vérifier les détails de l'application), mettez à jour les détails selon vos besoins.
6. Accédez à votre keystore, puis cliquez sur **Create**.

7. Enregistrez votre application.



Lorsque l'outil finit d'encapsuler l'application, le nom de fichier de l'application contient _andr.

Encapsulation d'applications d'entreprise Android à l'aide de la ligne de commande

Vous pouvez utiliser l'encapsulation d'applications d'entreprise pour encapsuler des applications personnalisées (internes) et certaines applications tierces. Vous devez acquérir des applications tierces directement à partir du fournisseur de l'application. Pour l'encapsulation d'applications d'entreprise, commencez par une application Android (.apk). Avant d'utiliser le Toolkit pour encapsuler des applications, sauvegardez la version d'origine de ces applications de façon à ce que vous puissiez y revenir si nécessaire.

L'exemple suivant illustre une commande d'encapsulation d'application de base utilisant les paramètres par défaut. L'application est signée avec le keystore fourni. Un keystore est un fichier qui contient les certificats utilisés pour signer votre application Android. Si le keystore contient plusieurs clés privées, vous pouvez spécifier l'alias de clé. Vous créez un keystore une fois. Puis, vous pouvez utiliser le keystore pour signer les applications que vous encapsulez. Si vous n'utilisez pas le

même keystore pour encapsuler la nouvelle version d'une application que vous avez déjà déployée, les mises à niveau de cette application ne fonctionneront pas. Les utilisateurs devront supprimer manuellement l'ancienne version avant de pouvoir installer la nouvelle version.

Modifiez les informations en gras pour votre système. La barre oblique inverse de fin indique que la commande continue à la ligne suivante. Veuillez supprimer ces symboles avant d'exécuter la commande.

> **Remarque :**
>
> Étant donné que le répertoire /Applications/ est restreint, vous devrez peut-être exécuter la commande suivante en mode super utilisateur. Pour ce faire, ajoutez sudo devant la commande. Vous serez invité à entrer votre mot de passe d'ordinateur lors de l'exécution à partir de ce répertoire restreint.

```
java -jar /Applications/Citrix/MDXToolkit/ManagedAppUtility.jar \
wrap \
-in ~/Desktop/SampleApps/Sample.apk \
-out ~/Desktop/SampleApps/Sample.mdx \
-keystore ~/Desktop/MyCompany.keystore \
-storepass MyKeystorePassword \
-keyalias MyCompanyKeyAlias \
-keypass MyKeyAliasPassword
```

Les exemples suivants sont des options que vous pouvez ajouter à la commande précédente, après modification des informations en gras :

- -appName "application encapsulée exemple"
- -appDesc "Ceci est mon application Android nouvellement encapsulée."

En outre, si le keystore n'est pas disponible lors du développement, utilisez la commande suivante pour créer une version commerciale d'une application mobile signée avec votre clé :

```
java -jar /Applications/Citrix/MDXToolkit/ManagedAppUtility.jar \
wrap \
-in ~/Desktop/SampleApps/Sample.apk \
-out ~/Desktop/SampleApps/Sample.mdx \
-keystore ~/Desktop/MyCompany.keystore \
-storepass MyKeystorePassword \
-keyalias MyCompanyKeyAlias \
-keypass MyKeyAliasPassword \
-createCert
```

Pour de plus amples informations sur les options disponibles, consultez la section Options de commande. Pour la documentation en ligne, utilisez l'option –help.

Encapsulation d'applications Android ISV à l'aide de la ligne de commande

Avant d'utiliser le Toolkit pour encapsuler des applications, assurez-vous de sauvegarder la version d'origine de ces applications de façon à ce que vous puissiez y revenir si nécessaire. Pour générer des applications ISV encapsulées pour Android, démarrez avec la commande d'encapsulation de base suivante.

```
java -jar /Applications/Citrix/MDXToolkit/ManagedAppUtility.jar \
wrap \
-in ~/Desktop/SampleApps/Sample.apk \
-out ~/Desktop/SampleApps/Sample.mdx \
-keystore ~/Desktop/MyCompany.keystore \
-storepass MyKeystorePassword \
-keyalias MyCompanyKeyAlias \
-keypass MyKeyAliasPassword \
-createCert
```

Pour encapsuler une application en tant qu'application ISV, vous devez définir le paramètre *–apptype* comme suit :

- **Premium:** pour encapsuler une application en tant qu'application Premium, dans laquelle certaines stratégies Citrix sont appliquées même pour les utilisateurs non gérés, ajoutez l'option suivante : -apptype Premium.
- **General:** pour encapsuler une application en tant qu'application General, qui ne contient aucune application de stratégie Citrix pour un utilisateur non géré, ajoutez l'option suivante : -apptype General

Si vous avez besoin de télécharger le fichier .apk encapsulé pour Google Play Store ou le serveur Web et que l'adresse URL est connue lors de l'encapsulation, ajoutez l'option -storeURL. Assurez-vous de définir également le paramètre apptype.

```
-storeURL "https://play.google.com/store/apps/details?id=com.zenprise"
```

Si vous ne connaissez pas l'adresse URL lors de l'encapsulation, vous pouvez modifier le fichier .MDX ultérieurement à l'aide de la commande suivante :

```
java -jar /Applications/Citrix/MDXToolkit/ManagedAppUtility.jar \
setinfo \
-in ~/Desktop/SampleApps/Sample.mdx \
-out ~/Desktop/SampleApps/wrapped/Sample.mdx \
-storeURL \
 "https://play.google.com/store/apps/details?id=com.zenprise"
```

Si vous avez personnalisé le fichier de stratégie, assurez-vous de pointer vers le fichier modifié :

`-policyxml /Applications/Citrix/MDXToolkit/data/policy_metadata.xml` Pour de plus amples informations sur les options disponibles, consultez la section Options de commande. Pour la documentation en ligne, utilisez l'option **–help**.

Options de commande

commande wrap

- **Help:** affiche l'aide de cette commande.
- **In:** obligatoire. Chemin d'accès et nom de fichier de l'application en cours d'encapsulation.
- **Out:** facultatif. Chemin d'accès et nom de fichier pour le fichier .mdx. Si cette option est omise, le fichier porte le même chemin d'accès et nom de fichier que le fichier d'entrée, avec une extension .mdx.
- **AppType:** facultatif. La valeur par défaut est MDXOnly. Pour générer des applications ISV, utilisez soit General soit Premium.
- **KeyStore:** chemin d'accès au fichier de keystore. Requis pour la signature du fichier .apk.
- **StorePass:** mot de passe du keystore. Requis pour la signature du fichier .apk.
- **KeyAlias:** nom de la clé dans le keystore. Requis pour la signature du fichier .apk.
- **KeyPass:** mot de passe pour la clé. Requis pour la signature du fichier .apk.
- **SigAlg:** facultatif. Algorithme à utiliser pour la signature.
- **AppName:** facultatif. Nom de l'application, obtenu depuis l'application, si possible.
- **AppDesc:** facultatif. Description de l'application, obtenue depuis l'application, si possible.
- **MinPlatform:** facultatif. Niveau de SDK minimal pris en charge. Valeur par défaut : vide.
- **MaxPlatform:** facultatif. Niveau de SDK maximal pris en charge. Valeur par défaut : vide.
- **ExcludedDevices:** facultatif. Liste des types d'appareil sur lesquels l'application n'est pas autorisée à être exécutée. Valeur par défaut : vide.
- **PolicyXML:** facultatif. Fichier de définition et chemin d'accès de stratégie XML de remplacement. Valeur par défaut : définitions de stratégie intégrées. Exemple :

  `-policyxml/Applications/Citrix/MDXToolkit/data/policy_metadata.xml`

  Pour de plus amples informations, veuillez consulter la section suivante, « Prédéfinition de stratégies MDX pour applications Android ».
- **StoreURL:** pour les applications ISV, adresse URL de l'application dans Google App Store. Valeur par défaut : vide.

commande sign

- **Help:** affiche l'aide de cette commande.
- **In:** obligatoire. Chemin d'accès et nom de fichier de l'application en cours d'encapsulation.
- **Out:** facultatif. Chemin d'accès et nom de fichier pour le fichier .mdx. Si cette option est omise, le fichier porte le même chemin d'accès et nom de fichier que le fichier d'entrée, avec une extension .mdx.
- **KeyStore:** obligatoire. Chemin d'accès au fichier de keystore.
- **StorePass:** obligatoire. Mot de passe du keystore.
- **KeyAlias:** obligatoire. Nom de la clé dans le keystore.
- **KeyPass:** obligatoire. Mot de passe pour la clé.
- **SigAlg:** facultatif. Algorithme à utiliser pour la signature.

commande setinfo

- **Help:** affiche l'aide de cette commande.
- **In:** obligatoire. Chemin d'accès et nom de fichier de l'application à modifier.
- **Out:** pour setinfo, le nom de fichier ou le chemin de sortie doit être différent de celui d'origine.
- **AppType:** facultatif. La valeur par défaut est MDXOnly. Pour générer des applications ISV, utilisez soit General soit Premium.
- **KeyStore:** chemin d'accès au fichier de keystore. Requis pour la signature du fichier .apk.
- **StorePass:** mot de passe du keystore. Requis pour la signature du fichier .apk.
- **KeyAlias:** nom de la clé dans le keystore. Requis pour la signature du fichier .apk.
- **KeyPass:** mot de passe pour la clé. Requis pour la signature du fichier .apk.
- **SigAlg:** facultatif. Algorithme à utiliser pour la signature.
- **AppName:** facultatif. Nom de l'application, obtenu depuis l'application, si possible.
- **AppDes:** facultatif. Description de l'application, obtenue depuis l'application, si possible.
- **MinPlatform:** facultatif. Niveau de SDK minimal pris en charge. Valeur par défaut : vide.
- **MaxPlatform:** facultatif. Niveau de SDK maximal pris en charge. Valeur par défaut : vide.
- **ExcludedDevices:** facultatif. Liste des types d'appareil sur lesquels l'application n'est pas autorisée à être exécutée. Valeur par défaut : vide.
- **StoreURL:** pour les applications ISV, adresse URL de l'application dans Google App Store. Valeur par défaut : vide.

- **PolicyXML:** facultatif. Fichier de définition et chemin d'accès de stratégie XML de remplacement. Valeur par défaut : définitions de stratégie intégrées. Exemple :

  ```
  -policyxml /Applications/Citrix/MDXToolkit/data/policy_metadata.xml
  ```

  Pour de plus amples informations, veuillez consulter la section suivante, « Prédéfinition de stratégies MDX pour applications Android ».

Prédéfinition de stratégies MDX pour applications Android

Pour les applications que vous encapsulez avec l'outil de ligne de commande MDX Toolkit, vous pouvez prédéfinir certaines stratégies MDX. Vous pouvez également configurer des stratégies dans la console Citrix Endpoint Management lorsque vous ajoutez les applications.

1. Mettez à jour les valeurs de stratégie dans le fichier de stratégie XML.

   Le programme d'installation de MDX Toolkit crée ce fichier de stratégie : Applications/Citrix/MDXToolkit/data/policy_metadata.xml

   > **Remarque :**
   >
   > Veuillez noter que les fichiers de stratégie iOS diffèrent des fichiers Android. Pour prédéfinir des stratégies pour ces deux plates-formes, vous devez mettre à jour leurs fichiers de stratégie XML respectifs.

2. Lorsque vous encapsulez l'application à l'aide de la ligne de commande, vous devez inclure

   ```
   -policyxml /Applications/Citrix/MDXToolkit/data/policy_metadata.xml
   ```

Identification des erreurs d'encapsulation des applications Android

Si vous rencontrez une erreur lors de l'encapsulation d'une application Android, vous pouvez utiliser les journaux de l'outil MDX Toolkit pour identifier l'erreur. vous devez disposer des droits d'administrateur pour afficher les journaux de MDX Toolkit.

Lorsque vous exécutez le MDX Toolkit, l'outil enregistre un fichier journal dans l'emplacement suivant : Applications/CitrixMDXToolkit/Logs/Citrix.log. Par défaut, l'outil enregistre les avertissements et les erreurs dans le journal.

Collecte des journaux d'application à partir de la ligne de commande

1. Installez le Android Debug Bridge à partir du site Web Android Developer. Pour plus de détails, voir Android Debug Bridge.
2. Entrez la commande suivante pour effacer les journaux existants : **"adb logcat -c"**

3. Reproduisez le problème.
4. Entrez la commande suivante pour capturer les journaux dans un fichier : **adb logcat -d > Name_of_Log_File.txt**

Synopsis des stratégies applicatives tierces MDX

February 19, 2019

Cet article décrit les stratégies MDX tierces pour applications iOS et Android. Le MDX Toolkit ne prend pas en charge Windows. Les notes comprennent des restrictions et des recommandations Citrix. Pour voir les stratégies prises en charge par le conteneur Android for Work, consultez la section connexe Android for Work.

Pour connaître les stratégies pour les applications de productivité mobiles Citrix, consultez Synopsis des stratégies MDX pour les applications de productivité mobiles.

> **Remarque :**
>
> Secure Hub actualise les stratégies au cours de certaines actions. Pour de plus amples informations, consultez la section Administration de Secure Hub.

Authentification

Code secret de l'appareil

- iOS : Oui
- Android : Non
- Paramètre par défaut : Désactivé

> Remarque :
>
> Cette stratégie s'applique uniquement aux appareils iOS 9, que Citrix ne prend plus en charge.

Code secret d'application

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Activé

Session en ligne requise

- iOS : Oui
- Android : Non
- Paramètre par défaut : Désactivé

Période hors connexion maximale

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 168 heures (7 jours)

Passerelle NetScaler Gateway alternative

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Vide

Sécurité de l'appareil

Bloquer les appareils jailbreakés ou rootés

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Activé

Exiger verrouillage de l'appareil

- iOS : Non
- Android : Oui
- Paramètre par défaut : Désactivé

Configuration réseau requise

Exiger Wi-Fi

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Désactivé

Réseaux Wi-Fi autorisés

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Vide

Accès divers

Période de grâce de mise à jour des applications (heures)

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 168 heures (7 jours)

> Remarque :
>
> Citrix recommande d'utiliser une valeur autre que zéro (0). La valeur zéro empêche immédiatement les utilisateurs, sans avertissement, d'utiliser une application en cours d'exécution jusqu'à ce qu'ils téléchargent et installent la mise à jour. Ce paramètre pourrait conduire à une situation dans laquelle les utilisateurs sont obligés de quitter l'application et, potentiellement, de perdre les modifications apportées à leur travail.

Effacer les données des applications après verrouillage

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Désactivé

Période d'interrogation active (minutes)

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 60 minutes (1 heure)

> Remarque :
>
> Utilisez une valeur inférieure à la valeur par défaut uniquement pour les applications à haut risque, sinon, les performances risquent d'être affectées.

Chiffrement

Activer le chiffrement

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

> Attention :
>
> Si vous modifiez cette stratégie après le déploiement d'une application, les utilisateurs devront la réinstaller.

Clés de cryptage

- iOS : Non
- Android : Oui
- Paramètre par défaut : Accès hors connexion autorisé est la seule option disponible.

Cryptage de fichiers privés

- iOS : Non
- Android : Oui
- Paramètre par défaut : SecurityGroup

Exclusions de cryptage des fichiers privés

- iOS : Non
- Android : Oui
- Paramètre par défaut : Vide

Limites d'accès pour les fichiers publics

- iOS : Non
- Android : Oui
- Paramètre par défaut : Vide

Inclusions de cryptage de bases de données

- iOS : Oui
- Android : Non
- Paramètre par défaut : Vide

Cryptage de fichiers publics

- iOS : Non
- Android : Oui
- Paramètre par défaut : SecurityGroup

Exclusions de cryptage de fichiers publics

- iOS : Non
- Android : Oui
- Paramètre par défaut : Vide

Migration de fichiers publics

- iOS : Non
- Android : Oui
- Paramètre par défaut : Write (WO/RW)

Exclusions de cryptage de fichiers

- iOS : Oui
- Android : Non
- Paramètre par défaut : Vide

Groupe de sécurité

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Vide

> Attention :
>
> Pour appliquer cette stratégie à une application existante, les utilisateurs doivent supprimer et réinstaller l'application.

Interaction des applications

Couper et copier

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Restreint

Coller

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Non restreint

Échange de documents (Ouvrir dans)

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Restreint

Liste d'exceptions d'ouverture restreinte

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Vide (pour Android) ; Applications Office 365 (pour iOS)

> Attention :
>
> N'oubliez pas de prendre en compte les incidences en matière de sécurité de cette stratégie. La liste d'exceptions autorise le déplacement de contenu entre des applications non gérées et l'environnement sécurisé MDX.

Échange de documents entrants (Ouvrir dans)

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Non restreint (pour Android et iOS) ; Tous (pour Android for Work)

Modèles d'URL d'application

- iOS : Oui
- Android : Non
- Paramètre par défaut : All registered app URL schemes are blocked.

Domaines d'URL exclus du filtrage

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Vide

URL autorisées

- iOS : Oui
- Android : Non
- Paramètre par défaut :
  +maps.apple.com
  +itunes.apple.com
  ^http:=ctxmobilebrowser:
  ^https:=ctxmobilebrowsers:
  ^mailto:=ctxmail:
  +^citrixreceiver:
  +^telprompt:
  +^tel:
  +^col-g2m-2:
  +^col-g2w-2:
  +^maps:ios_addr
  +^mapitem:

Domaines Secure Web autorisés

- iOS : Oui

- Android : Oui
- Paramètre par défaut : Vide

Restrictions applicatives

Bloquer la caméra

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Activé

Bloquer la galerie

- iOS : Non
- Android : Oui
- Paramètre par défaut : Désactivé

Bloquer la bibliothèque de photos

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer les enregistrements du micro

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Activé

Bloquer la dictée

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer les services de localisation

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Activé

Bloquer la composition de SMS

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Activé

Bloquer la capture d'écran

- iOS : Non
- Android : Oui
- Paramètre par défaut : Activé

Bloquer le capteur de l'appareil

- iOS : Non
- Android : Oui
- Paramètre par défaut : Activé

Bloquer NFC

- iOS : Non
- Android : Oui
- Paramètre par défaut : Activé

Bloquer iCloud

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer la recherche

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer la sauvegarde de fichiers

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer AirPrint

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer l'impression

- iOS : Non
- Android : Oui
- Paramètre par défaut : Activé

Bloquer AirDrop

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer les API Facebook et Twitter

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Obscurcir le contenu de l'écran

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer les claviers tiers

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Bloquer les journaux d'applications

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Désactivé

Accès au réseau

Accès réseau

- iOS : Oui
- Android : Oui
- Paramètre par défaut : pour les applications nouvellement téléchargées, la valeur par défaut est **Bloqué** pour toutes les applications, à l'exception de Secure Mail. Étant donné qu'Intune n'a pas d'état bloqué, la valeur par défaut pour Secure Mail est **Non restreint**.

Session micro VPN requise

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Non

Période de grâce requise pour la session micro VPN (minutes)

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 0 (pas de période de grâce)

Étiquette de certificat

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Vide

Liste d'exclusion

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Vide

Bloquer les connexions localhost

- iOS : Non
- Android : Oui
- Paramètre par défaut : Désactivé

Journaux d'applications

Sortie de journal par défaut

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Fichier

Niveau de journalisation par défaut

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 4 (Messages d'information)

Nombre maximal de fichiers journaux

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 2

Taille maximale du fichier journal

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 2 Mo

Rediriger les journaux système

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Géofencing

Longitude du point central

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 0

Latitude du point central

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 0

Rayon

- iOS : Oui
- Android : Oui
- Paramètre par défaut : 0 (désactivé)

> Remarque :
>
> Définir le rayon en mètres. Lorsqu'il est défini sur zéro, le géofencing est désactivé.

Analytics

Niveau de détail de Google Analytics

- iOS : Oui
- Android : Oui
- Paramètre par défaut : Complet

Rapports

Rapports Citrix

- iOS : Oui
- Android : Non
- Paramètre par défaut : Désactivé

> Remarque :
>
> Citrix peut également contrôler cette fonctionnalité avec un indicateur de fonctionnalité. L'indicateur de fonctionnalité et cette stratégie doivent être activés pour que cette fonctionnalité fonctionne.

Jeton de chargement

- iOS : Oui
- Android : Non
- Paramètre par défaut : Vide

Envoyer rapports uniquement via Wi-Fi

- iOS : Oui
- Android : Non
- Paramètre par défaut : Activé

Taille maximale de cache du fichier de rapport

- iOS : Oui
- Android : Non
- Paramètre par défaut : 2 Mo

Stratégies MDX pour les applications tierces pour Android

February 19, 2019

Cet article décrit les stratégies MDX pour applications tierces Android. Vous pouvez modifier les paramètres de stratégie dans la console Citrix Endpoint Management.

Authentification

Code secret d'application

Si cette option est définie **Activé**, un code PIN ou un code secret est requis pour déverrouiller l'application lorsque l'application démarre ou reprend après une période d'inactivité. La valeur par défaut est **Activé**.

Pour configurer l'inactivité du minuteur pour toutes les applications, définissez la valeur INACTIVITY_TIMER en minutes dans Propriétés du client sur l'onglet **Paramètres**. Par défaut, la valeur du délai d'inactivité est de 60 minutes. Pour désactiver le délai d'inactivité, de façon à ce qu'une invite de saisie du code PIN ou du code secret invite s'affiche uniquement lorsque l'application démarre, définissez la valeur sur zéro.

> **Remarque :**
>
> Si vous sélectionnez Secure offline pour la stratégie Clés de cryptage, cette stratégie est automatiquement activée.

Période hors connexion maximale (heures)

Définit la durée maximale pendant laquelle une application peut s'exécuter hors ligne sans ouverture de session réseau pour reconfirmer le droit et les stratégies d'actualisation. La valeur par défaut est **168 heures** (7 jours). La période minimale est d'une heure.

Il est rappelé à l'utilisateur de se connecter 30, 15 et 5 minutes avant l'expiration de la période ; après expiration , l'application reste verrouillée jusqu'à ce que l'utilisateur effectue une connexion réseau réussie.

Passerelle Citrix Gateway secondaire

> Remarque :
>
> Ce nom de stratégie dans la console Endpoint Management est **Passerelle NetScaler Gateway alternative**.

Adresse d'une autre passerelle Citrix Gateway (anciennement NetScaler Gateway) qui est utilisée pour l'authentification et les sessions micro VPN avec cette application. Il s'agit d'une stratégie facultative qui, lorsqu'elle est utilisée en conjonction avec la stratégie Session en ligne requise, oblige les applications à s'authentifier de nouveau à la passerelle spécifique. Ces types de passerelles ont généralement des exigences d'authentification et des stratégies de gestion du trafic différentes (meilleur contrôle). Si elle est laissée vide, la valeur par défaut du serveur est toujours utilisée. La valeur par défaut est vide.

Sécurité de l'appareil

Bloquer les appareils jailbreakés ou rootés

Si cette option est définie sur **Activé**, l'application est verrouillée lorsque l'appareil est jailbreaké ou rooté. Si elle est définie sur **Désactivé**, l'application peut fonctionner même si l'appareil est jailbreaké ou rooté. La valeur par défaut est **Activé**.

Exiger verrouillage de l'appareil

Si **Code secret ou code PIN de l'appareil** est sélectionné, l'application est verrouillée si aucun code PIN ou code secret n'est configuré sur l'appareil. Si **Séquence de verrouillage de l'écran de l'appareil** est sélectionné, l'application est verrouillée si l'appareil ne possède pas de verrou d'écran de modèle défini. Si cette option est définie sur **Désactivé**, l'application est autorisée à être exécutée même si aucun écran de verrouillage, code PIN ou code secret n'est défini sur l'appareil. La valeur par défaut est **Désactivé**.

Code secret ou code PIN de l'appareil requiert Android version 4.1 (Jellybean) au minimum. Si vous définissez la stratégie sur **Code secret ou code PIN de l'appareil**, une application ne peut pas être exécutée sur des versions antérieures.

Sur les appareils Android M, les options **Code secret ou code PIN de l'appareil** et **Séquence de verrouillage de l'écran de l'appareil** ont le même effet : dans les deux cas, l'application est verrouillée si l'appareil ne possède pas de code PIN ou de code secret, ou qu'une séquence de verrouillage de l'écran a été définie.

Configuration réseau requise

Exiger Wi-Fi

Si l'option **Activé** est sélectionné, l'application est verrouillée lorsque l'appareil n'est pas connecté à un réseau Wi-Fi. Si l'option **Désactivé** est sélectionné, l'application peut fonctionner si l'appareil est connecté, à un réseau 4G/3G, LAN ou Wi-Fi par exemple. La valeur par défaut est **Désactivé**.

Réseaux Wi-Fi autorisés

Liste séparée par des virgules des réseaux Wi-Fi autorisés. Si le nom du réseau contient des caractères non alphanumériques (y compris des virgules), il doit être entre guillemets. Les applications fonctionnent uniquement lorsqu'elles sont connectées à l'un des réseaux répertoriés. Si rien n'est spécifié, tous les réseaux sont autorisés. Cela n'affecte pas les connexions aux réseaux cellulaires La valeur par défaut est vide.

Accès divers

Période de grâce de mise à jour des applications (heures)

Définit la période de grâce pendant laquelle une application peut être utilisée une fois que le système a découvert qu'une mise à jour de l'application est disponible. La valeur par défaut est **168 heures** (7 jours).

> Remarque :
>
> L'utilisation d'une valeur zéro n'est pas recommandée car une valeur zéro empêche immédiatement une application en cours d'exécution d'être utilisée tant que la mise à jour n'est pas téléchargée et installée (sans que l'utilisateur en soit averti). Cela pourrait entraîner une situation dans laquelle l'utilisateur qui exécute l'application est obligé de quitter l'application (risque de perte de travail) afin de procéder à la mise à jour requise.

Effacer les données des applications après verrouillage

Efface les données et réinitialise l'application lorsqu'elle est fermée. Si cette option est définie sur **Désactivé**, les données d'application ne sont pas effacées lorsque l'application est verrouillée. La valeur par défaut est **Désactivé**.

Vous pouvez verrouiller une application pour les raisons suivantes :

- Perte du droit d'application pour l'utilisateur.
- Abonnement à l'application supprimé
- Compte supprimé
- Secure Hub désinstallé
- Nombre d'échecs d'authentification de l'application trop élevé.
- Appareil jailbreaké détecté (par paramètre de stratégie)
- Appareil verrouillé par une autre action d'administration

Période d'interrogation active (minutes)

Lorsqu'une application démarre, l'infrastructure MDX interroge Citrix Endpoint Management pour déterminer l'application en cours et l'état de l'appareil. En supposant que le serveur exécutant Endpoint Management peut être contacté, l'infrastructure renvoie des informations sur l'état de verrouillage et d'effacement de l'appareil et l'état d'activation ou de désactivation de l'application. Que le serveur puisse être contacté ou non, une autre interrogation est planifiée, basée sur l'intervalle d'interrogation. Une fois cette période expirée, une nouvelle tentative d'interrogation est effectuée. La valeur par défaut est **60 minutes** (1 heure).

> Important :
>
> Abaissez cette valeur uniquement pour les applications à haut risque, sinon, les performances risquent d'être affectées.

Chiffrement

Clés de cryptage

Permet aux secrets utilisés pour dériver des clés de cryptage d'être conservés sur l'appareil. Accès hors connexion autorisé est la seule option disponible.

Citrix vous recommande de définir la stratégie Authentification pour permettre une ouverture de session réseau ou un défi de mot de passe en mode déconnecté afin de protéger l'accès au contenu crypté.

Cryptage de fichiers privés

Contrôle le cryptage de fichiers de données privées dans les emplacements suivants : /data/-data/<nomapp> et /mnt/sdcard/Android/data/<nomapp>.

L'option **Disabled** signifie que les fichiers publics ne sont pas cryptés. L'option **SecurityGroup** crypte les fichiers privés à l'aide d'une clé partagée par toutes les applications MDX dans le même groupe de sécurité. L'option **Application** crypte les fichiers privés à l'aide d'une clé unique à cette application. La valeur par défaut est **Groupe de sécurité**.

Exclusions de cryptage des fichiers privés

Contient une liste séparée par des virgules de chemin d'accès de fichiers. Chaque chemin est une expression régulière qui représente un ou plusieurs fichiers cryptés. Les chemins de fichiers sont relatifs aux sandboxes internes et externes. La valeur par défaut est vide.

Les exclusions s'appliquent uniquement aux dossiers suivants :

- **Stockage interne :**

  /data/data/

- **Carte SD :**

  /storage/emulated/<SD Card Slot>/Android/data/

  /storage/emulated/legacy/Android/data/

Exemples

Fichier à exclure	Valeur dans l'exclusion de cryptage de fichiers privés
/data/data/com.citrix.mail/files/a.txt	^files/a.txt
Tous les fichiers texte dans /storage/emulated/0/Android/data/com.citrix.mail/files	^files/(.)+.txt$
Tous les fichiers dans /data/data/com.citrix.mail/files	^files/

Cryptage de fichiers publics

Contrôle le cryptage des fichiers publics. Si l'option **Désactivé** est sélectionnée, les fichiers publics ne sont pas cryptés. Si l'option **Groupe de sécurité** est sélectionnée, les fichiers publics sont cryptés à l'aide d'une clé partagée par toutes les applications MDX dans le même groupe de sécurité. Si l'option **Application** est sélectionnée, les fichiers publics sont cryptés à l'aide d'une clé unique à cette application.

La valeur par défaut est **Groupe de sécurité**.

Exclusions de cryptage de fichiers publics

Contient une liste séparée par des virgules de chemin d'accès de fichiers. Chaque chemin est une expression régulière qui représente un ou plusieurs fichiers qui ne sont pas cryptés. Les chemins d'accès aux fichiers sont relatifs au stockage externe par défaut et à tout autre stockage externe spécifique à l'appareil.

Les exclusions de cryptage de fichiers publics incluent des emplacements de dossiers externes uniquement.

Exemples

Fichier à exclure	Valeur dans Exclusions de cryptage de fichiers publics
Dossier Téléchargements sur une carte SD	Télécharger
Tous les fichiers MP3 du dossier Musique	^Music/(.)+.mp3$

Migration de fichiers publics

Cette stratégie est appliquée uniquement lorsque vous activez la stratégie Public file encryption (modifiée de **Disabled** à **SecurityGroup** ou **Application**). Cette stratégie est uniquement applicable aux fichiers publics existants et cryptés et spécifie le moment où ces fichiers sont cryptés. La valeur par défaut est **Écriture (WO/RW)**.

L'option **Désactivé** signifie que les fichiers existants ne sont pas cryptés. L'option **Écriture (WO/RW)** crypte les fichiers existants uniquement lorsqu'ils sont ouverts en accès en écriture seule ou en accès en lecture écriture. L'option **Quelconque** crypte les fichiers existants lorsqu'ils sont ouverts dans n'importe quel mode. Options :

- **Désactivé**. Ne crypte pas les fichiers existants.
- **Écriture (WO/RW)**. Crypte les fichiers existants uniquement lorsqu'ils sont ouverts en accès en écriture seule ou en accès en lecture écriture.
- **Tout**. Crypte les fichiers existants lorsqu'ils sont ouverts dans n'importe quel mode.

> Remarques :
>
> - Les nouveaux fichiers ou les fichiers non cryptés existants qui sont remplacés cryptent les fichiers de remplacement dans tous les cas.
> - Le fait de crypter un fichier public existant rend ce fichier non disponible pour d'autres applications qui ne possèdent pas la même clé de cryptage.

Groupe de sécurité

Laissez ce champ vide si vous souhaitez que toutes les applications mobiles gérées par Citrix Endpoint Management puissent échanger des informations entre elles. Définissez un nom de groupe de sécurité pour gérer des paramètres de sécurité pour des ensembles d'applications spécifiques (par exemple, Finance ou Ressources humaines).

> Avertissement
>
> Si vous modifiez cette stratégie pour une application existante, les utilisateurs doivent supprimer et réinstaller l'application pour appliquer la modification apportée à la stratégie.

Domaines Secure Web autorisés

Cette stratégie n'est en vigueur que pour les domaines non exclus par la stratégie de filtrage d'URL. Ajoutez une liste séparée par des virgules de noms de domaine complets (FQDN) ou de suffixes DNS redirigés vers l'application Secure Web lorsque l'échange de documents est Restreint.

Si cette stratégie contient des entrées, seules les URL avec des champs hôtes correspondant à au moins un élément de la liste (via la correspondance de suffixe DNS) seront redirigées vers l'application Secure Web lorsque l'échange de documents est restreint.

Toutes les autres URL seront envoyées au navigateur Web Android par défaut (ignorant la restriction de la stratégie d'échange de documents). La valeur par défaut est vide.

lnteraction des applications

Couper et copier

Bloque, autorise ou restreint les opérations de couper/coller sur le Presse-papiers pour cette application. Si ce paramètre est défini sur **Restreint**, les données copiées du Presse-papiers sont placées dans un Presse-papiers privé qui est uniquement disponible auprès des applications MDX. La valeur par défaut est **Restreint**.

Coller

Bloque, autorise ou limite les opérations de collage sur le presse-papiers pour cette application. Lorsque vous choisissez le paramètre **Restreint**, les données collées sur le Presse-papiers sont collectées depuis un Presse-papiers privé qui est uniquement disponible auprès des applications MDX. La valeur par défaut est **Non restreint**.

Échange de documents (Ouvrir dans)

Bloque, autorise ou restreint les opérations d'échange de documents pour l'application. Si ce paramètre est défini sur **Restreint**, les documents peuvent être échangés uniquement avec d'autres applications MDX et les exceptions d'application spécifiées dans la stratégie Liste d'exceptions d'ouverture restreinte. Si l'option **Non restreint** est sélectionnée, définissez les stratégies Cryptage de fichiers publics et Cryptage de fichiers privés sur **Désactivé** de façon à ce que les utilisateurs puissent ouvrir des documents dans les applications non encapsulées. La valeur par défaut est **Restreint**.

Liste d'exceptions d'ouverture restreinte

Lorsque la stratégie Échange de documents (Ouvrir dans) est définie sur **Restreint**, cette liste d'intents Android est autorisée à être transmise aux applications non gérées. Une connaissance des intents Android est nécessaire pour ajouter des filtres à la liste. Un filtre peut spécifier une action, un package, un schéma, ou une combinaison de ces derniers.

Exemples

```
{
 action=android.intent.action.MAIN }

{
 package=com.sharefile.mobile }

{
 action=android.intent.action.DIAL scheme=tel }
```

> Avertissement
>
> N'oubliez pas de prendre en compte les incidences en matière de sécurité de cette stratégie. La liste d'exceptions autorise le déplacement de contenu entre des applications non gérées et l'environnement sécurisé MDX.

Échange de documents entrants (Ouvrir dans)

Bloque, restreint ou autorise les opérations d'échange de documents entrants pour cette application. Si l'option **Restreint** est sélectionnée, les documents ne peuvent être échangés qu'avec d'autres applications MDX. La valeur par défaut est **Non restreint**.

Si ce paramètre est défini sur **Bloqué** ou **Restreint**, vous pouvez utiliser la stratégie Liste blanche d'échange de documents entrants pour spécifier les applications autorisées à envoyer des documents à cette application. Pour plus d'informations sur les autres interactions entre les stratégies, veuillez consulter la stratégie Bloquer la galerie.

Options : **Non restreint**, **Bloqué** ou **Restreint**

Restrictions applicatives

> Important
>
> Tenez compte des répercussions sur la sécurité des stratégies qui empêchent vos applications d'accéder ou d'utiliser les fonctionnalités du téléphone. Lorsque ces stratégies sont définies

> sur **Désactivé**, le contenu peut transiter entre des applications non gérées et l'environnement sécurisé.

Bloquer la caméra

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser directement la caméra. La valeur par défaut est **Activé**.

Bloquer la galerie

Si cette option est définie sur **Activé**, elle empêche une application d'accéder à la galerie sur l'appareil. La valeur par défaut est **Désactivé**. Cette stratégie fonctionne en conjonction avec la stratégie Échange de documents entrants (Ouvrir dans).

- Si le paramètre Échange de documents entrants (Ouvrir dans) est défini sur **Restreint**, les utilisateurs qui travaillent dans l'application gérée ne peuvent pas joindre des images à partir de la galerie, quel que soit le paramètre défini pour Bloquer la galerie.
- Si le paramètre Échange de documents entrants (Ouvrir dans) est défini sur **Non restreint**, l'expérience des utilisateurs qui travaillent dans l'application gérée sera la suivante :
  - Les utilisateurs peuvent joindre des images si Bloquer la galerie est défini sur **Désactivé**.
  - Les utilisateurs ne peuvent pas joindre des images si Bloquer la galerie est défini sur **Activé**.

Bloquer les enregistrements du micro

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser directement le microphone. La valeur par défaut est **Activé**.

Bloquer les services de localisation

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser directement les services de géolocalisation (GPS ou réseau). La valeur par défaut est **Désactivé** pour Secure Mail.

Bloquer la composition de SMS

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser la fonctionnalité de composition de SMS utilisée pour envoyer des SMS/messages texte à partir de l'application. La valeur par défaut est **Activé**.

Bloquer la capture d'écran

Si cette option est définie sur **Activé**, elle empêche les captures d'écran initiées par l'utilisateur lorsque l'application est en cours d'exécution. Elle obscurcit également l'écran de l'application lorsque l'utilisateur change d'application. La valeur par défaut est **Activé**.

Lors de l'utilisation de la fonctionnalité NFC (communication en champ proche) d'Android, certaines applications prennent une copie d'écran d'elles-mêmes avant de transmettre le contenu. Pour activer cette fonctionnalité dans une application encapsulée, modifiez la stratégie Bloquer la capture d'écran sur **Désactivé**.

Bloquer le capteur de l'appareil

Si cette option est définie sur **Activé**, empêche une application d'utiliser les capteurs de l'appareil (comme l'accéléromètre, le capteur de mouvement et le gyroscope). La valeur par défaut est **Activé**.

Bloquer NFC

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser la communication en champ proche (NFC). La valeur par défaut est **Activé**.

Bloquer les journaux d'applications

Si l'option est définie sur **Activé**, elle empêche une application d'utiliser la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile. Si cette option est définie sur **Désactivé**, les journaux d'application sont enregistrés et peuvent être collectées à l'aide de la fonctionnalité de prise en charge de la messagerie de Secure Hub. La valeur par défaut est **Désactivé**.

Bloquer l'impression

Si cette option est définie sur **Activé**, elle empêche une application d'imprimer des données. Si une application dispose de la commande Partager, vous devez définir Échange de documents (Ouvrir dans) sur **Restreint** ou **Bloqué** pour bloquer l'impression. La valeur par défaut est**Activé**.

Accès au réseau

Accès réseau

Les options des paramètres sont les suivantes :

- **Utiliser les paramètres précédents :** utilise par défaut les valeurs que vous aviez définies dans les stratégies précédentes. Si vous modifiez cette option, vous ne devez pas revenir à cette option. Notez également que les modifications apportées aux nouvelles stratégies ne prennent effet que lorsque l'utilisateur met à niveau l'application vers 18.12.0 ou version ultérieure.
- **Bloqué :** les API de mise en réseau utilisées par votre application échoueront. Conformément à la recommandation précédente, vous devriez traiter un tel échec de façon appropriée.
- **Sans restriction :** tous les appels réseau ont un accès direct et ne sont pas tunnélisés.
- **Tunnel VPN complet :** tout le trafic provenant de l'application gérée est tunnélisé via Citrix Gateway.

Session micro VPN requise

Si cette option est définie sur **Oui**, l'utilisateur doit disposer d'une connexion au réseau d'entreprise et d'une session active. Si elle est définie sur **Non**, une session active n'est pas nécessaire. La valeur par défaut est **Utiliser paramètres précédents**. Pour les applications nouvellement téléchargées, la valeur par defaut est **Non**. Le paramètre sélectionné avant la mise à niveau de cette stratégie reste en vigueur jusqu'à ce qu'une option autre que **Utiliser paramètres précédents** soit sélectionnée.

Période de grâce requise pour la session micro VPN (minutes)

Définit la période de grâce pendant laquelle une application peut continuer d'être utilisée une fois que le système a découvert qu'une mise à jour de l'application est disponible. La valeur par défaut est 168 heures (7 jours).

> Remarque :
>
> L'utilisation d'une valeur zéro n'est pas recommandée car une valeur zéro empêche immédiatement une application en cours d'exécution d'être utilisée tant que la mise à jour n'est pas téléchargée et installée (sans que l'utilisateur en soit averti). Cela pourrait entraîner une situation dans laquelle l'utilisateur qui exécute l'application est obligé de quitter l'application (risque de perte de travail) afin de procéder à la mise à jour requise.

Étiquette de certificat

Lorsqu'elle est utilisée avec le service d'intégration de certificat de StoreFront, cette étiquette identifie le certificat requis pour cette application. Si aucune étiquette n'est fournie, aucun certificat n'est disponible pour être utilisé avec une infrastructure de clé publique (PKI). La valeur par défaut est vide (aucun certificat utilisé).

Liste d'exclusion

Liste délimitée par des virgules de noms de domaine complets ou de suffixes DNS auxquels accéder directement plutôt que par une connexion VPN. Cela s'applique uniquement au mode **Tunnel - SSO Web** lorsque Citrix Gateway est configuré avec le mode inverse de split tunneling.

Bloquer les connexions localhost

Si cette option est définie sur **Activé**, les applications ne sont pas autorisées à établir des connexions localhost. Localhost est une adresse (telle que 127.0.0.1 ou ::1) pour les communications se produisant localement sur l'appareil. localhost contourne le matériel de l'interface réseau locale et accède aux services réseau s'exécutant sur l'hôte. Si l'option est définie sur **Désactivé**, cette stratégie remplace la stratégie Accès réseau, ce qui signifie que les applications peuvent se connecter en dehors du conteneur sécurisé si le périphérique exécute un serveur proxy localement. La valeur par défaut est **Désactivé**.

Journaux d'applications

Sortie de journal par défaut

Détermine quels supports de sortie sont utilisés par défaut par la fonctionnalité de journalisation des diagnostics de l'application Citrix Endpoint Management. Les supports possibles sont les suivants : fichier, console, ou les deux. La valeur par défaut est file.

Niveau de journalisation par défaut

Contrôle le niveau de détail par défaut de la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile. Plus le numéro est élevé, plus la journalisation est détaillée.

- 0 - Rien n'est enregistré
- 1 - Erreurs critiques
- 2 - Erreurs
- 3 - Avertissements
- 4 - Messages d'information
- 5 - Messages d'information détaillés
- 6 à 15 - niveaux de débogage de 1 à 10

La valeur par défaut est le niveau **4** (Messages d'information).

Nombre maximal de fichiers journaux

Limite le nombre de fichiers journaux conservés par la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile avant le déploiement. La valeur minimale est **2**. La valeur maximale est **8**. La valeur par défaut est **2**.

Taille maximale du fichier journal

Limite la taille en Mo des fichiers journaux conservés par la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile avant le déploiement. La valeur minimale est **1 Mo**. La valeur maximale est **5 Mo**. La valeur par défaut est **2** Mo.

Géofencing

Longitude du point central

Longitude (coordonnées X) du point central du périmètre dans lequel l'application est autorisée à être utilisée. Si l'application est utilisée en dehors du périmètre configuré, elle reste verrouillée. Doit être exprimée en format de degrés signés (DDD.dddd), par exemple « -31.9635 ». Les longitudes occidentales doivent être précédées d'un signe moins. La valeur par défaut est **0**.

Latitude du point central

Latitude (coordonnées Y) du point central du périmètre dans lequel l'application est autorisée à être utilisée. Si l'application est utilisée en dehors du périmètre configuré, elle reste verrouillée.

Doit être exprimée en format de degrés signés (DDD.dddd), par exemple « -43.06581 ». Les latitudes méridionales doivent être précédées d'un signe moins. La valeur par défaut est **0**.

Rayon

Rayon du périmètre (géofencing) dans lequel l'application est autorisée à être utilisée. Si l'application est utilisée en dehors du périmètre configuré, elle reste verrouillée.
Doit être exprimée en mètres. Lorsqu'il est défini sur zéro, le géofencing est désactivé. La valeur par défaut est 0 (désactivé).

Analytics

Niveau de détail de Google Analytics

Citrix collecte des données d'analyse pour améliorer la qualité de ses produits. Le fait de sélectionner Anonyme vous permet de ne pas inclure les informations identifiables de la société.

Stratégies MDX pour les applications tierces pour iOS

February 19, 2019

Cet article décrit les stratégies MDX pour applications tierces iOS. Vous pouvez modifier les paramètres de stratégie directement dans les fichiers XML de stratégie ou dans la console Citrix Endpoint Management lorsque vous ajoutez une application.

Authentification

Code secret de l'appareil

Si cette option est définie **Activé**, un code PIN ou un code secret est requis pour déverrouiller l'appareil lorsque l'application démarre ou reprend après une période d'inactivité. Un code secret de l'appareil est requis pour crypter les données de l'application à l'aide du cryptage de fichier Apple. Les données pour toutes les applications sur l'appareil sont chiffrées. La valeur par défaut est **Désactivé**.

Code secret d'application

Si cette option est définie **Activé**, un code PIN ou un code secret est requis pour déverrouiller l'application lorsque l'application démarre ou reprend après une période d'inactivité. La valeur par défaut est **Activé**.

Pour configurer l'inactivité du minuteur pour toutes les applications, définissez la valeur **INACTIVITY_TIMER** en minutes dans **Propriétés du client** sur l'onglet **Paramètres**. Par défaut, la valeur du délai d'inactivité est de **60** minutes. Pour désactiver le délai d'inactivité, de façon à ce qu'une invite de saisie du code PIN ou du code secret invite s'affiche uniquement lorsque l'application démarre, définissez la valeur sur zéro.

> Remarque :
>
> Si vous sélectionnez **Secure offline** pour la stratégie Clés de cryptage, cette stratégie est automatiquement activée.

Session en ligne requise

Période hors connexion maximale (heures)

Définit la durée maximale pendant laquelle une application peut s'exécuter sans avoir à reconfirmer les identifiants liés à l'application ni à actualiser les stratégies de Citrix Endpoint Management. À l'expiration, une ouverture de session sur le serveur peut être déclenchée si nécessaire. La valeur par défaut est **168 heures** (7 jours). La période minimale est d'**une** heure.

Passerelle Citrix Gateway secondaire

> Remarque :
>
> Ce nom de stratégie dans la console Endpoint Management est **Passerelle NetScaler Gateway alternative**.

Adresse d'une autre passerelle Citrix Gateway qui devrait être utilisée pour l'authentification et les sessions micro VPN avec cette application. Il s'agit d'une stratégie facultative qui, lorsqu'elle est utilisée en conjonction avec la stratégie Session en ligne requise, oblige les applications à s'authentifier de nouveau à la passerelle spécifique. Ces types de passerelles ont généralement des exigences d'authentification et des stratégies de gestion du trafic différentes (meilleur contrôle). Si elle est laissée vide, la passerelle par défaut du serveur est toujours utilisée. La valeur par défaut est vide.

Sécurité de l'appareil

Bloquer les appareils jailbreakés ou rootés

Si cette option est définie sur **Activé**, l'application est verrouillée lorsque l'appareil est jailbreaké ou rooté. Si elle est définie sur **Désactivé**, l'application peut fonctionner même si l'appareil est jailbreaké ou rooté. La valeur par défaut est **Activé**.

Configuration réseau requise

Exiger Wi-Fi

Si l'option **Activé** est sélectionné, l'application est verrouillée lorsque l'appareil n'est pas connecté à un réseau Wi-Fi. Si l'option **Désactivé** est sélectionné, l'application peut fonctionner si l'appareil est connecté, à un réseau 4G/3G, LAN ou Wi-Fi par exemple. La valeur par défaut est **Désactivé**.

Réseaux Wi-Fi autorisés

Liste séparée par des virgules des réseaux Wi-Fi. Si le nom du réseau contient des caractères non alphanumériques (y compris des virgules), il doit être entre guillemets. L'application fonctionne uniquement lorsqu'elles sont connectées à l'un des réseaux répertoriés. Si rien n'est spécifié, tous les réseaux sont autorisés. Cela n'affecte pas les connexions aux réseaux cellulaires La valeur par défaut est vide.

Accès divers

Période de grâce de mise à jour des applications (heures)

Définit la période de grâce pendant laquelle une application peut continuer d'être utilisée une fois que le système a découvert qu'une mise à jour de l'application est disponible. La valeur par défaut est **168 heures (7 jours)**.

> Remarque :
>
> L'utilisation d'une valeur zéro n'est pas recommandée car une valeur zéro empêche immédiatement une application en cours d'exécution d'être utilisée tant que la mise à jour n'est pas téléchargée et installée (sans que l'utilisateur en soit averti). Cela pourrait entraîner une situation dans laquelle l'utilisateur est obligé de quitter l'application (risque de perte de travail) afin de procéder à la mise à jour requise.

Effacer les données des applications après verrouillage

Efface les données et réinitialise l'application lorsqu'elle est fermée. Si cette option est définie sur **Désactivé**, les données d'application ne sont pas effacées lorsque l'application est verrouillée. La valeur par défaut est **Désactivé**.

Vous pouvez verrouiller une application pour les raisons suivantes :

- Perte du droit d'application pour l'utilisateur.
- Abonnement à l'application supprimé
- Compte supprimé
- Secure Hub désinstallé
- Nombre d'échecs d'authentification de l'application trop élevé.
- Appareil jailbreaké détecté (par paramètre de stratégie)
- Appareil verrouillé par une autre action d'administration

Période d'interrogation active (minutes)

Lorsqu'une application démarre, l'infrastructure MDX interroge Citrix Endpoint Management pour déterminer l'application en cours et l'état de l'appareil. En supposant que le serveur exécutant Endpoint Management peut être contacté, l'infrastructure renvoie des informations sur l'état de verrouillage et d'effacement de l'appareil et l'état d'activation ou de désactivation de l'application. Que le serveur puisse être contacté ou non, une autre interrogation est planifiée, basée sur l'intervalle d'interrogation. Une fois cette période expirée, une nouvelle tentative d'interrogation est effectuée. La valeur par défaut est **60 minutes** (1 heure).

> Important :
>
> Abaissez cette valeur uniquement pour les applications à haut risque, sinon, les performances risquent d'être affectées.

Chiffrement

Activer le chiffrement

Si cette option est définie sur **Désactivé**, les données stockées sur l'appareil ne sont pas cryptées. Si elle est définie sur **Activé**, les données stockées sur l'appareil sont cryptées. La valeur par défaut est **Activé**.

> Attention :
>
> Si vous modifiez cette stratégie après le déploiement d'une application, les utilisateurs devront la réinstaller.

Exclusions de cryptage de bases de données

Liste d'exclusion de bases de données qui ne sont pas automatiquement cryptées. Pour éviter le cryptage de base de données pour une base de données spécifique, ajoutez une entrée à cette liste d'expressions régulières séparée par des virgules. Si un nom de chemin d'accès de fichier correspond à une des expressions régulières, la base de données est exclue du cryptage. Les modèles d'exclusion prennent en charge la syntaxe d'expressions régulières étendues Posix 1003.2. Le modèle de correspondance n'est pas sensible à la casse.

Exemples

\.db$,\.sqlite$ exclut tout nom de chemin d'accès à la base de données se terminant par « .db » ou « .sqlite ».

\/Database\/unencrypteddb\.db renvoie la base de données unencrypteddb.db dans le sous-dossier Database.

\/Database\/ renvoie toutes les bases de données contenant **/Database/** dans le chemin d'accès.

La valeur par défaut est vide.

Exclusions de cryptage de fichiers

Liste d'exclusion de fichiers qui ne sont pas automatiquement cryptés. Pour éviter le cryptage d'un ensemble spécifique de fichiers, ajoutez une entrée à la liste séparée par des virgules des expressions régulières. Si un nom de chemin d'accès de fichier ne correspond à aucune des expressions régulières, alors ce fichier est exclu du cryptage. Les modèles d'exclusion prennent en charge la syntaxe d'expressions régulières étendues Posix 1003.2. Le modèle de correspondance n'est pas sensible à la casse.

Exemples

\.log$,\.dat$ exclut tout nom du chemin d'accès à un fichier qui se termine par « .log » ou « .dat ».

\/Documents\/unencrypteddoc\.txt renvoie le contenu du fichier unencrypteddoc.txt dans le sous-dossier Documents.

\/Documents\/UnencryptedDocs\/.\.txt renvoie les fichiers contenant « .txt » dans le sous-chemin /Documents/UnencryptedDocs/.

La valeur par défaut est vide.

Groupe de sécurité

Laissez ce champ vide si vous souhaitez que toutes les applications mobiles gérées par Endpoint Management puissent échanger des informations entre elles. Définissez un nom de groupe de sécurité pour gérer des paramètres de sécurité pour des ensembles d'applications spécifiques (par exemple, Finance ou Ressources humaines).

> Avertissement
>
> Pour appliquer cette stratégie à une application existante, les utilisateurs doivent supprimer et réinstaller l'application.

Interaction des applications

Couper et copier

Bloque, autorise ou restreint les opérations de couper/coller sur le Presse-papiers pour cette application. Si ce paramètre est défini sur **Restreint**, les données copiées du Presse-papiers sont placées dans un Presse-papiers privé qui est uniquement disponible auprès des applications MDX. La valeur par défaut est **Restreint**.

Coller

Bloque, autorise ou limite les opérations de collage sur le presse-papiers pour cette application. Lorsque vous choisissez le paramètre **Restreint**, les données collées sur le Presse-papiers sont collectées depuis un Presse-papiers privé qui est uniquement disponible auprès des applications MDX. La valeur par défaut est **Non restreint**.

Échange de documents (Ouvrir dans)

Bloque, autorise ou restreint les opérations d'échange de documents pour cette application. Si l'option est définie sur **Restreint**, les documents peuvent être échangés uniquement avec d'autres applications MDX.

Si l'option **Non restreint** est sélectionnée, vous devez définir la stratégie Activer le cryptage sur **Activé** de façon à ce que les utilisateurs puissent ouvrir des documents dans les applications non encapsulées. Si l'application de réception n'est pas encapsulée ou que le cryptage est désactivé sur cette dernière, Citrix Endpoint Management décrypte le document.
La valeur par défaut est **Restreint**.

Liste d'exceptions d'ouverture restreinte

Lorsque la stratégie Échange de documents (Ouvrir dans) est définie sur **Restreint**, une application MDX peut partager des documents avec cette liste délimitée par des virgules d'ID d'applications non gérées, même si la stratégie Échange de documents (Ouvrir dans) est définie sur **Restreint** et que la stratégie Activer le cryptage est définie sur **Activé**.

com.microsoft.Office.Word,com.microsoft.Office.Excel,com.microsoft.Office.Powerpoint,
com.microsoft.onenote,com.microsoft.onenoteiPad,com.microsoft.Office.Outlook

Seules les applications Office 365 sont prises en charge pour cette stratégie.

> Avertissement
>
> N'oubliez pas de prendre en compte les incidences en matière de sécurité de cette stratégie. La liste d'exceptions autorise le déplacement de contenu entre des applications non gérées et l'environnement sécurisé MDX.

Échange de documents entrants (Ouvrir dans)

Bloque, restreint ou autorise les opérations d'échange de documents entrants pour cette application. Si l'option **Restreint** est sélectionnée, les documents ne peuvent être échangés qu'avec d'autres applications MDX. La valeur par défaut est **Non restreint**.

Si ce paramètre est défini sur **Bloqué** ou **Restreint**, vous pouvez utiliser la stratégie Liste blanche d'échange de documents entrants pour spécifier les applications autorisées à envoyer des documents à cette application.

Options : **Non restreint**, **Bloqué** ou **Restreint**

Modèles d'URL d'application

Les applications iOS peuvent envoyer des requêtes d'adresse URL à d'autres applications qui ont été enregistrées pour gérer des systèmes spécifiques (telles que « `http://` »). Cette fonctionnalité permet à une application de transmettre des requêtes d'aide vers une autre application. Cette stratégie permet de filtrer les schémas qui sont transmis dans cette application à des fins de gestion (c'est-à-dire les adresses URL entrantes). La valeur par défaut est vide, ce qui signifie que tous les modèles d'URL d'application enregistrés sont bloqués.

La stratégie doit être sous forme d'une liste séparée par des virgules de modèles dans laquelle chaque modèle peut être précédé d'un signe plus (+) ou moins (-). Les URL entrantes sont comparées aux modèles dans l'ordre indiqué jusqu'à ce qu'une correspondance soit trouvée. Lorsqu'une correspondance est trouvée, l'action exécutée est déterminée par le préfixe.

- Un préfixe de signe moins (-) bloque la transmission de l'adresse URL dans l'application.
- Un préfixe de signe plus (+) autorise la transmission de l'adresse URL dans l'application.
- Si aucun préfixe (+ ou -) n'est fourni avec le modèle, « + » (autoriser) est la valeur par défaut.
- Si une URL entrante ne correspond à aucun modèle dans la liste, elle est bloquée.

La table suivante contient des exemples de systèmes d'adresses URL d'application :

Modèle	Application qui requiert le modèle d'URL	Objectif
ctxmobilebrowser	Secure Web-	Autoriser Secure Web à traiter les URL HTTP: à partir d'autres applications.-
ctxmobilebrowsers	Secure Web-	Autoriser Secure Web à traiter les URL HTTPS: à partir d'autres applications.
ctxmail	Secure Mail-	Autoriser Secure Mail à traiter les URL mailto: à partir d'autres applications.
COL-G2M	GoToMeeting-	Autoriser une application GoToMeeting groupée à prendre en charge les demandes de réunion.
ctxsalesforce	Citrix for Salesforce-	Autoriser Citrix pour Salesforce à traiter les requêtes Salesforce.
wbx	WebEx	Autoriser une application WebEx groupée à prendre en charge les demandes de réunion.

Interaction des applications (URL sortante)

Domaines exclus du filtrage des URL

Cette stratégie exclut les URL sortantes de tout filtrage « URL autorisées ». Ajoutez une liste séparée par des virgules de noms de domaine complets (FQDN) ou de suffixes DNS pour les exclure de tout filtrage « URL autorisées ». Si cette stratégie est vide (valeur par défaut), les processus de filtrage « URL autorisées » définis sont des URL. Si cette stratégie contient des entrées, les URL avec des champs hôtes correspondant à au moins un élément de la liste (via la correspondance de suffixe DNS) sont envoyées directement à iOS, ignorant la logique de filtrage « URL autorisées ». La valeur par défaut est vide.

URL autorisées

Les applications iOS peuvent envoyer des requêtes d'adresse URL à d'autres applications qui ont été enregistrées pour gérer des systèmes spécifiques (telles que « `"http://"` »). Cette fonctionnalité permet à une application de transmettre des requêtes d'aide vers une autre application. Cette stratégie permet de filtrer les URL qui sont transmises depuis cette application vers d'autres applications à des fins de gestion (c'est-à-dire les adresses URL sortantes).

La stratégie doit être sous forme d'une liste séparée par des virgules de modèles dans laquelle chaque modèle peut être précédé d'un signe plus (+) ou moins (-). Les URL sortantes sont comparées aux modèles dans l'ordre indiqué jusqu'à ce qu'une correspondance soit trouvée. Lorsqu'une correspondance est trouvée, l'action exécutée est déterminée par le préfixe. Un préfixe de signe moins (-) bloque la transmission de l'adresse URL à une autre application. Un préfixe de signe plus (+) autorise la transmission de l'adresse URL à une autre application. Si aucun préfixe (+ ou -) n'est fourni avec le modèle, « + » (autoriser) est la valeur par défaut. Une paire de valeurs séparées par « = » indique une substitution où les occurrences de la première chaîne sont remplacées par celles de la seconde. Vous pouvez utiliser le préfixe « ^ » d'expression régulière pour rechercher la chaîne à ancrer au début de l'adresse URL. Si une URL sortante ne correspond à aucun modèle dans la liste, elle est bloquée.

Mode par défaut

+ maps.apple.com

+ itunes.apple.com

^http:=ctxmobilebrowser:

^https:=ctxmobilebrowsers:

^mailto:=ctxmail:

+^citrixreceiver:

+^telprompt:

+^tel:

+^lmi-g2m:

+^maps:ios_addr

+^mapitem:

+^sms:

+^facetime:

+^ctxnotes:

+^ctxnotesex:

+^ctxtasks:

+^facetime-audio:

+^itms-apps:

+^ctx-sf:

+^sharefile:

+^lync:

+^slack:

Si ce paramètre est laissé vide, toutes les URL sont bloquées, à l'exception des URL suivantes :

- http:
- https:
- +citrixreceiver: +tel:

La table suivante contient des exemples d'adresses URL autorisées :

Format URL	Description
^mailto:=ctxmail:	Toutes les URL mailto: s'ouvrent dans Secure Mail.
^http:	Toutes les adresses URL HTTP s'ouvrent dans Secure Web.
^https:	Toutes les adresses URL HTTPS s'ouvrent dans Secure Web.
^tel:	Permet à l'utilisateur d'effectuer des appels.
-//www.dropbox.com	Bloque les URL Dropbox envoyées depuis des applications gérées.
+^COL-G2M:	Autorise les applications gérées à ouvrir l'application cliente GoToMeeting.
-^SMS:	Bloque l'utilisation d'un client de chat.
-^wbx:	Empêche les applications gérées d'ouvrir l'application cliente WebEx.
+^ctxsalesforce:	Permet à Citrix pour Salesforce de communiquer avec votre serveur Salesforce.

Domaines Secure Web autorisés

Cette stratégie affecte uniquement les entrées de stratégie « URL autorisées » qui redirigeraient une URL vers l'application Secure Web (^ http:=ctxmobilebrowser: and ^https:=ctxmobilebrowsers:). Ajoutez une liste séparée par des virgules des noms de domaine complets (FQDN) ou des suffixes DNS autorisés à être redirigés vers l'application Secure Web. Si cette stratégie est vide (valeur par défaut), tous les domaines peuvent être redirigés vers l'application Secure Web. Si cette stratégie contient des entrées, seules les URL contenant des champs d'hôte correspondant à au moins un élément dans la liste (via la correspondance de suffixe DNS) sont redirigées vers l'application Secure Web. Toutes les autres URL sont envoyées sans modification à iOS, en ignorant l'application Secure Web. La valeur par défaut est vide.

Restrictions applicatives

> Important :
>
> Tenez compte des répercussions sur la sécurité des stratégies qui empêchent vos applications d'accéder ou d'utiliser les fonctionnalités du téléphone. Lorsque ces stratégies sont définies sur **Désactivé**, le contenu peut transiter entre des applications non gérées et l'environnement sécurisé.

Bloquer la caméra

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser directement la caméra. La valeur par défaut est **Désactivé**.

Bloquer la bibliothèque de photos

Si cette option est définie sur **Activé**, elle empêche une application d'accéder à la bibliothèque de photos sur l'appareil. La valeur par défaut est **Activé**.

Bloquer les enregistrements du micro

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser directement le microphone. La valeur par défaut est **Activé**.

Bloquer la dictée

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser directement les services de dictée. La valeur par défaut est **Activé**.

Bloquer les services de localisation

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser directement les services de géolocalisation (GPS ou réseau). La valeur par défaut est **Désactivé** pour Secure Mail.

Bloquer la composition de SMS

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser la fonctionnalité de composition de SMS utilisée pour envoyer des SMS/messages texte à partir de l'application. La valeur par défaut est **Activé**.

Bloquer la composition d'e-mail

Si cette option est définie sur **Activé**, empêche une application d'utiliser la fonctionnalité de composition d'e-mail utilisée pour envoyer des e-mails à partir de l'application. La valeur par défaut est **Activé**.

Bloquer iCloud

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser iCloud pour stocker et partager des données et des paramètres.

> Remarque :
>
> Le fichier de données dans iCloud est contrôlé par la stratégie Bloquer la sauvegarde de fichiers.

La valeur par défaut est **Activé**.

Bloquer la recherche

Si cette stratégie est **activée**, elle empêche une application d'utiliser la fonctionnalité de recherche. Cette dernière permet de rechercher du texte surligné dans le Dictionnaire, iTunes, l'App Store, les horaires des séances de cinéma, des lieux alentour et plus. La valeur par défaut est **Activé**.

Bloquer la sauvegarde de fichiers

Si cette option est définie sur **Activé**, elle empêche les fichiers de données d'être sauvegardés par iCloud ou iTunes. La valeur par défaut est **Activé**.

Bloquer AirPrint

Si cette option est définie sur **Activé**, empêche une application d'utiliser les fonctionnalités AirPrint pour imprimer des données sur des imprimantes compatibles AirPrint. La valeur par défaut est **Activé**.

Bloquer AirDrop

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser AirDrop. La valeur par défaut est **Activé**.

Bloquer les API Facebook et Twitter

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser les API de Facebook et Twitter sur iOS. La valeur par défaut est **Activé**.

Obscurcir le contenu de l'écran

Si cette option est définie sur **Activé**, lorsque les utilisateurs basculent entre applications, l'écran est obscurcit. Cette stratégie empêche iOS d'enregistrer le contenu de l'écran et d'afficher des miniatures. La valeur par défaut est **Activé**.

Bloquer les claviers tiers (iOS 11 et version ultérieures uniquement)

Si cette option est définie sur **Activé**, elle empêche une application d'utiliser des extensions de clavier tierces sur iOS 8. La valeur par défaut est **Activé**.

Bloquer les journaux d'applications

Si l'option est définie sur **Activé**, elle empêche une application d'utiliser la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile. Si cette option est définie sur **Désactivé**, les journaux d'application sont enregistrés et peuvent être collectées à l'aide de la fonctionnalité de prise en charge de la messagerie de Secure Hub. La valeur par défaut est **Désactivé**.

Accès au réseau

Accès réseau

> Remarque :
>
> **Tunnel - SSO Web** est le nom de **Secure Browse** dans les paramètres. Le comportement est le même.

Les options des paramètres sont les suivantes :

- **Utiliser les paramètres précédents :** utilise par défaut les valeurs que vous aviez définies dans les stratégies précédentes. Si vous modifiez cette option, vous ne devez pas revenir à **Utiliser paramètres précédents**. Notez également que les modifications apportées aux nouvelles stratégies ne prennent effet que lorsque l'utilisateur met à niveau l'application vers 18.12.0 ou version ultérieure.
- **Bloqué :** tous les accès réseau sont bloqués. Les API de mise en réseau utilisées par votre application échoueront. Conformément à la recommandation précédente, vous devriez traiter un tel échec de façon appropriée.
- **Sans restriction :** tous les appels réseau ont un accès direct et ne sont pas tunnélisés.
- **Tunnel VPN complet :** tout le trafic provenant de l'application gérée est tunnélisé via Citrix Gateway.
- **Tunnel - SSO Web :** l'URL HTTP/HTTPS est réécrite. Cette option permet uniquement le tunneling du trafic HTTP et HTTPS. Un avantage important de **Tunnel - SSO Web** est l'authentification unique (SSO) pour le trafic HTTP et HTTPS ainsi que l'authentification PKINIT. Sur Android, cette option a une charge de configuration faible et il s'agit donc de l'option préférée pour les opérations de type navigation Web.
- **Tunnel - VPN complet et SSO Web :** permet de basculer automatiquement entre les modes VPN selon les besoins. Si une demande réseau qui a échoué en raison d'une demande d'authentification qui ne peut pas être traitée dans un mode VPN spécifique est de nouveau tentée dans un autre mode.

Si l'un des **modes Tunnel** est sélectionné, un tunnel VPN par application dans ce mode initial est recréé sur le réseau d'entreprise et les paramètres de split tunneling Citrix Gateway sont utilisés. Citrix recommande un **tunnel VPN complet** pour les connexions qui utilisent des certificats clients ou des connexions SSL de bout en bout vers une ressource dans le réseau d'entreprise. Citrix recommande **Tunnel - SSO Web** pour les connexions qui nécessitent l'authentification unique (SSO).

Session micro VPN requise

Si cette option est définie sur **Oui**, l'utilisateur doit disposer d'une connexion au réseau d'entreprise et d'une session active. Si elle est définie sur **Non**, une session active n'est pas nécessaire. La valeur par défaut est **Utiliser paramètres précédents**. Pour les applications nouvellement téléchargées, la valeur par defaut est **Non**. Le paramètre sélectionné avant la mise à niveau de cette nouvelle stratégie reste en vigueur jusqu'à ce qu'une option autre que **Utiliser paramètres précédents** soit sélectionnée.

Période de grâce requise pour la session micro VPN (minutes)

Cette valeur détermine le nombre de minutes pendant lesquelles les utilisateurs peuvent utiliser l'application hors connexion avant que la stratégie Session en ligne requise n'empêche son utilisation (jusqu'à ce que la session en ligne soit validée). La valeur par défaut est **0** (pas de période de grâce). Cette stratégie ne s'applique pas à l'intégration avec Microsoft Intune/EMS.

Étiquette de certificat

Lorsqu'elle est utilisée avec le service d'intégration de certificat de StoreFront, cette étiquette identifie le certificat requis pour cette application. Si aucune étiquette n'est fournie, aucun certificat n'est disponible pour être utilisé avec une infrastructure de clé publique (PKI). La valeur par défaut est vide (aucun certificat utilisé).

Liste d'exclusion

Liste délimitée par des virgules de noms de domaine complets ou de suffixes DNS auxquels accéder directement plutôt que par une connexion VPN. Cela s'applique uniquement au mode **Tunnel - SSO Web** lorsque Citrix Gateway est configuré avec le mode inverse de split tunneling.

Journaux d'applications

Sortie de journal par défaut

Détermine quels supports de sortie sont utilisés par défaut par la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile. Les supports possibles sont les suivants : fichier, console, ou les deux. La valeur par défaut est **file**.

Niveau de journalisation par défaut

Contrôle le niveau de détail par défaut de la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile. Chaque niveau comprend des niveaux de valeurs moindres. Plage de niveaux possibles :

- 0 - Rien n'est enregistré
- 1 - Erreurs critiques
- 2 - Erreurs
- 3 - Avertissements
- 4 - Messages d'information

- 5 - Messages d'information détaillés
- 6 à 15 - niveaux de débogage de 1 à 10

La valeur par défaut est le niveau 4 (Messages d'information).

Nombre maximal de fichiers journaux

Limite le nombre de fichiers journaux conservés par la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile avant le déploiement. La valeur minimale est de 2. La valeur maximale est de 8. La valeur par défaut est 2.

Taille maximale du fichier journal

Limite la taille en Mo des fichiers journaux conservés par la fonctionnalité de journalisation des diagnostics de l'application de productivité mobile avant le déploiement. La valeur minimale est de 1 Mo. La valeur maximale est de 5 Mo. La valeur par défaut est 2 Mo.

Rediriger les journaux système

Si ce paramètre est défini sur **Activé**, il intercepte et redirige les journaux système ou de console d'une application vers la fonctionnalité de diagnostics des applications de productivité mobiles. Si ce paramètre est défini sur **Désactivé**, l'utilisation par une application des journaux système ou de console n'est pas interceptée.

La valeur par défaut est **Activé**.

Géofencing

Longitude du point central

Longitude (coordonnées X) du point central du périmètre dans lequel l'application est autorisée à être utilisée. Si l'application est utilisée en dehors du périmètre configuré, elle reste verrouillée.

Doit être exprimée en format de degrés signés (DDD.dddd), par exemple « -31.9635 ». Les longitudes occidentales doivent être précédées d'un signe moins. La valeur par défaut est **0**.

Latitude du point central

Latitude (coordonnées Y) du point central du périmètre dans lequel l'application est autorisée à être utilisée. Si l'application est utilisée en dehors du périmètre configuré, elle reste verrouillée.

Doit être exprimé en format de degrés signés (DDD.dddd), par exemple' 43.06581". Les latitudes méridionales doivent être précédées d'un signe moins. La valeur par défaut est **0**.

Rayon

Rayon du périmètre (géofencing) dans lequel l'application est autorisée à être utilisée. Si l'application est utilisée en dehors du périmètre configuré, elle reste verrouillée.

Devrait être exprimé en mètres. Lorsqu'il est défini sur zéro, le géofencing est désactivé. Lorsque la stratégie Bloquer les services de localisation est activée, le géofencing ne fonctionne pas correctement. La valeur par défaut est **0** (désactivé).

Analytics

Niveau de détail de Google Analytics

Citrix collecte des données d'analyse pour améliorer la qualité de ses produits. Le fait de sélectionner **Anonyme** permet aux utilisateurs de ne pas inclure les informations identifiables de la société. Le paramètre par défaut est Complète.

Rapports

Rapports Citrix

Si le paramètre est réglé sur **Activé**, Citrix collecte les rapports d'incident et les diagnostics pour aider à résoudre les problèmes. Si le paramètre est réglé sur **Désactivé**, Citrix ne collecte pas de données.

> Remarque :
>
> Citrix peut également contrôler cette fonctionnalité avec un indicateur de fonctionnalité. L'indicateur de fonctionnalité et cette stratégie doivent être activés pour que cette fonctionnalité fonctionne.

La valeur par défaut est **Désactivé**.

Jeton de chargement

Vous pouvez obtenir un jeton de téléchargement à partir de votre compte Citrix Insight Services (CIS). Si vous spécifiez ce jeton facultatif, CIS vous donne accès aux rapports d'incidents et aux diagnostics chargés à partir de vos appareils. Citrix a accès à ces mêmes informations. La valeur par défaut est vide.

Envoyer rapports uniquement via Wi-Fi

Si ce paramètre est **activé**, Citrix envoie des rapports d'erreur et des diagnostics uniquement lorsque vous êtes connecté à un réseau Wi-Fi. La valeur par défaut est **Activé**.

Taille maximale de cache du fichier de rapport

Limite la taille des packs de rapports d'incident et de diagnostics conservés avant de vider le cache. La valeur minimale est de 1 Mo. La valeur maximale est de 5 Mo. La valeur par défaut est 2 Mo.

Guide du développeur MDX

February 19, 2019

Citrix Endpoint Management est une solution d'entreprise qui vous permet de gérer des appareils, des applications et des données mobiles. Le principe de la gestion d'application mobile (MAM) de Endpoint Management consiste à injecter la fonctionnalité d'entreprise dans des applications préexistantes qui sont ensuite hébergées sur le magasin d'applications privé d'une entreprise, Apple App Store ou Google Play Store.

Pour ajouter la fonctionnalité d'entreprise de Endpoint Management à des applications mobiles, vous devez les encapsuler avec le MDX Toolkit. Le MDX Toolkit est une technologie de conteneur d'application qui améliore l'expérience sur les appareils mobiles et prépare les applications en vue de sécuriser le déploiement avec Endpoint Management en ajoutant les fonctionnalités MDX. Les fonctionnalités MDX comprennent les stratégies et les paramètres, les certificats de sécurité signés et le code de gestion des applications mobiles.

Le MDX Toolkit comprend le SDK de l'application MDX, qui procure un ensemble complet de fonctionnalités MDX à vos applications mobiles via la technologie de conteneur d'application Citrix MDX. Les API vous offrent les possibilités suivantes.

- Effectuer des actions dans des applications encapsulées sur la base de stratégies Endpoint Management. À titre d'exemple, si une stratégie Endpoint Management empêche le couper-coller dans une application MDX, vous pouvez empêcher la sélection de texte dans votre application. Votre application peut communiquer et partager des stratégies avec d'autres applications MDX.
- Détecter les activités dans vos applications MDX. Par exemple, vous pouvez vérifier si une application est encapsulée ou gérée.
- Ajouter des fonctionnalités personnalisées, telles que la sécurité et l'application des stratégies.
- Développer des applications mobiles qui s'exécuteront à l'intérieur ou à l'extérieur d'un environnement Citrix.

Non seulement les applications qui utilisent le SDK de l'application MDX peuvent être configurées de façon centralisée avec des stratégies MDX lorsqu'elles sont utilisées avec Endpoint Management, ces applications peuvent aussi fonctionner en mode autonome en dehors des environnements Citrix.

Nouveautés du SDK de l'application MDX 10.2

La version 10.2 du SDK de l'application MDX pour iOS comprend ces améliorations et ces mises à jour.

La stratégie Classe de protection des données minimum est masquée. Pour rendre la stratégie visible dans Citrix Endpoint Management, ouvrez le fichier policymetadata.xml associé à l'application (dans Applications/Citrix/MDXToolkit/data) et, dans la section **MinimumDataProtectionClass**, modifiez la valeur de **PolicyHidden** sur **false**. Après avoir encapsulé votre application, la stratégie s'affiche lorsque vous ajoutez l'application à Citrix Endpoint Management.

Intégration de l'encapsulation d'application avec processus de création Xcode. Les développeurs peuvent maintenant encapsuler et publier une application iOS dans le cadre du processus de création de Xcode. Pour plus d'informations, voir Intégration du SDK dans votre bibliothèque d'applications.

Prise en charge de coffre partagé dans les applications Android. Le SDK de l'application MDX inclut désormais l'API Android pour la fonctionnalité de coffre partagé MDX, ce qui vous permet de partager un contenu géré entre les applications. Par exemple, le coffre sécurisé permet le partage de certificats et de clés privées par le biais d'une application inscrite de façon à ce que les applications puissent obtenir un certificat depuis le coffre sécurisé plutôt que depuis Secure Hub. Pour de plus amples informations, consultez la section API XenMobile pour Android.

> **Remarque :**
>
> les développeurs doivent veiller à tester les applications encapsulées qui effectuent un traitement en arrière-plan, tel que l'actualisation du contenu sur un appareil verrouillé ou les synchronisations en arrière-plan.

Capacités MAM

La fonctionnalité d'entreprise ajoutée par Endpoint Management est contrôlée au travers de stratégies que les administrateurs mettent à jour au niveau de l'application à partir de la console Endpoint Management. Endpoint Management force les stratégies vers les appareils mobiles selon la planification établie par les administrateurs. Les stratégies gèrent des fonctionnalités, telles que les suivantes :

- **Authentification**. Lors de l'ouverture d'une application gérée, Endpoint Management peut demander aux utilisateurs d'entrer des informations d'identification d'entreprise ou un code PIN. Cette vérification d'informations d'identification peut être répétée régulièrement.

- **Mises à jour d'applications**. Endpoint Management notifie les utilisateurs lorsque des mises à jour d'applications gérées sont disponibles. L'administrateur peut rendre les mises à jour obligatoires dans un certain délai. Si un utilisateur n'accepte pas une mise à jour, l'ancienne version de l'application ne pourra plus être exécutée une fois que le délai aura expiré.
- **Verrouillage et effacement à distance**. Un administrateur peut verrouiller temporairement ou effacer de manière permanente des applications par application ou par périphérique.
- **Restrictions réseau et VPN**. Une stratégie Endpoint Management contrôle l'accès réseau : l'accès peut être bloqué, routé via un VPN complet, ou routé via un VPN proxy. Le routage VPN s'effectue via un boîtier Citrix Gateway hébergé par l'entreprise.
- **Restrictions de communication entre les applications**. Une stratégie Endpoint Management détermine si le partage de documents entre les applications est bloqué ou autorisé uniquement entre des applications gérées. Par conséquent, la fenêtre « Ouvrir dans » dans votre application peut ignorer les applications non gérées.
- **Contention des fonctionnalités**. Les stratégies Endpoint Management peuvent désactiver les capacités d'un appareil pour une application, parmi lesquelles la caméra, le micro et le capteur de localisation.

Composants Endpoint Management

Les composants Endpoint Management suivants offrent des fonctionnalités MAM.

- **Serveur Citrix Endpoint Management**

  Ce serveur d'entreprise ou résidant sur le cloud héberge Citrix Endpoint Management Store, le magasin d'applications interne. Les administrateurs peuvent charger des applications mobiles vers Endpoint Management et configurer les stratégies d'application et d'appareil.

- **Secure Hub**

  Les utilisateurs d'entreprise installent Secure Hub pour Android ou iOS sur leur appareil mobile, puis configurent l'application avec une adresse URL d'inscription et des informations d'identification. Lorsque Secure Hub s'ouvre, les utilisateurs sélectionnent les applications d'entreprise à partir de Citrix Endpoint Management Store. Une fois que les applications ont été téléchargées et installées sur l'appareil, Secure Hub est utilisé comme hub pour la gestion de ces applications, exécutant certaines tâches telles que l'authentification utilisateur et les mises à jour de stratégies administrées de façon centrale.

- **MDX**

  MDX est la source des fonctionnalités MAM. Le MDX Toolkit ajoute le code MDX à votre application mobile. Outre l'encapsulation d'applications, vous ne travaillez pas directement avec le code MDX.

- **MDX Toolkit et SDK de l'application MDX**

Le MDX Toolkit ajoute la fonctionnalité d'entreprise aux applications mobiles, un processus appelé encapsulation d'application. Le SDK de l'application MDX permet aux développeurs et aux intégrateurs de systèmes d'activer MDX pour leurs applications mobiles. L'encapsulation d'application effectue trois tâches principales. Tout d'abord, il injecte le code Citrix dans votre application qui implémente les fonctionnalités de gestion l'application. Le résultat de cette tâche est un nouveau fichier d'application. Ensuite, l'encapsulation d'application signe le nouveau fichier d'application avec un certificat de sécurité. Enfin, l'encapsulation d'application crée un fichier MDX, qui contient des informations de stratégie et d'autres paramètres. Dans certains cas, le fichier d'application signé est également directement contenu dans le fichier MDX.

Ce guide de développeur se concentre sur l'encapsulation d'application pour les éditeurs de logiciels (ISV).

Modes gérés et non gérés pour les applications ISV

Le SDK de l'application MDX propose un comportement d'application en mode double, ce qui vous permet de déployer des applications pouvant être exécutées avec ou sans l'infrastructure MDX. Les applications qui sont exécutées indépendamment de Secure Hub sont appelées applications non gérées. Lorsque ces applications répondent à certaines conditions, elles deviennent des applications gérées et s'exécutent sous le contrôle de Secure Hub.

Le comportement en mode double est différent du comportement des applications MDX déployées directement depuis le principal Endpoint Management. Ces applications nécessitent toujours la présence de Citrix MDX et l'autorisation d'un magasin Endpoint Management Store pour s'exécuter. Avec Intune, toutefois, ces applications peuvent être déployées et gérées en l'absence de Secure Hub ou de Endpoint Management Store.

Vous pouvez utiliser les API Endpoint Management pour spécifier le type de comportement en mode double nécessaire lors de l'intégration avec une application MDX. Vous pouvez développer deux versions d'une application, une non gérée et une autre gérée, ou développer une seule application à utiliser dans les deux cas et à inclure dans MDX. L'infrastructure MDX applique les comportements par défaut associés aux applications gérées et non gérées.

La transition d'une application non gérée vers le mode géré diffère selon que l'application est encapsulée en tant qu'application General ou application Premium :

- **Application General :** une application General est hébergée sur l'App Store d'Apple ou Google Play Store. Les utilisateurs qui ne disposent pas de Secure Hub peuvent télécharger et exécuter l'application normalement dans un mode non géré, comme n'importe quel magasin d'applications générique. Si un utilisateur non géré installe Secure Hub ultérieurement, l'application ISV passe au mode géré si ces conditions sont remplies.

  - L'utilisateur ouvre une session sur un magasin d'entreprise de Citrix Endpoint Management au moins une fois.
  - L'utilisateur se trouve dans un groupe de mise à disposition Endpoint Management vers lequel l'application est déployée.
  - MDX abonne l'utilisateur.
  - Lorsqu'il y est invité, l'utilisateur confirme que son entreprise peut gérer l'application.

    Si un utilisateur bloque la gestion d'application d'entreprise, il peut continuer à exécuter l'application à des fins personnelles.

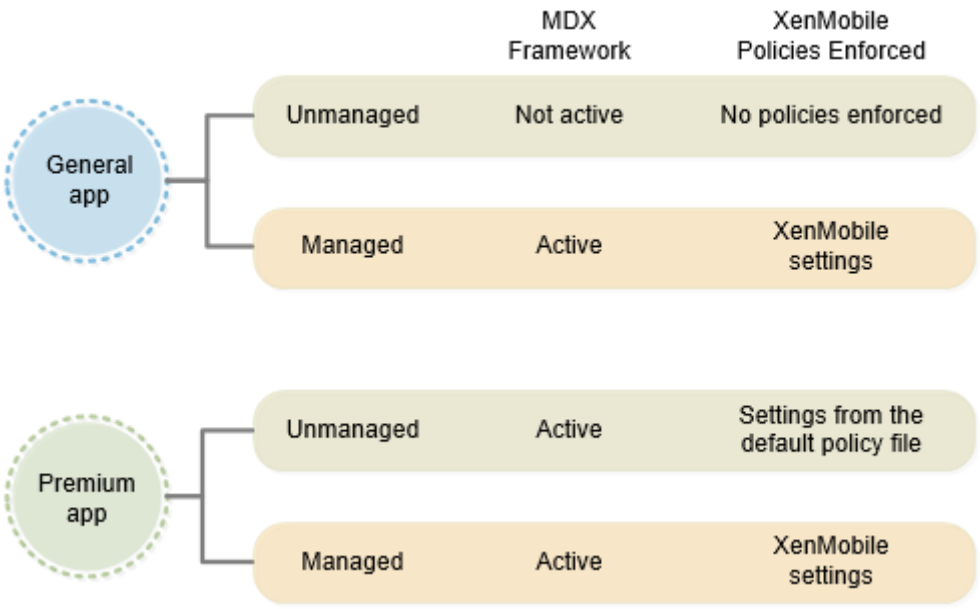
- **Application Premium :** une application Premium est une application destinée aux utilisateurs de votre entreprise. Les applications Citrix MDX sont des exemples d'applications Premium. Bien que les applications Premium s'exécutent généralement en mode géré, l'infrastructure MDX intégrée permet aux applications Premium de s'exécuter en mode non géré avec un ensemble par défaut de stratégies MDX que vous définissez au travers des fichiers de stratégie par défaut. Par conséquent, vous pouvez contrôler le comportement de l'application et utiliser les fonctionnalités MDX, même si l'utilisateur n'est pas associé à un compte d'entreprise.

  Si un utilisateur non géré installe Secure Hub ultérieurement, l'application passe de façon silencieuse au mode géré si ces conditions sont remplies.

  - L'utilisateur se trouve dans un groupe de mise à disposition Citrix Endpoint Management vers lequel l'application est déployée.
  - L'utilisateur ouvre une session sur Secure Hub si nécessaire.
  - MDX abonne l'utilisateur.

> **Remarque**
>
> Une application ne peut pas passer du mode géré au mode non géré.

Le diagramme suivant illustre les différences entre les applications General et Premium, selon qu'elles sont gérées ou non gérées.

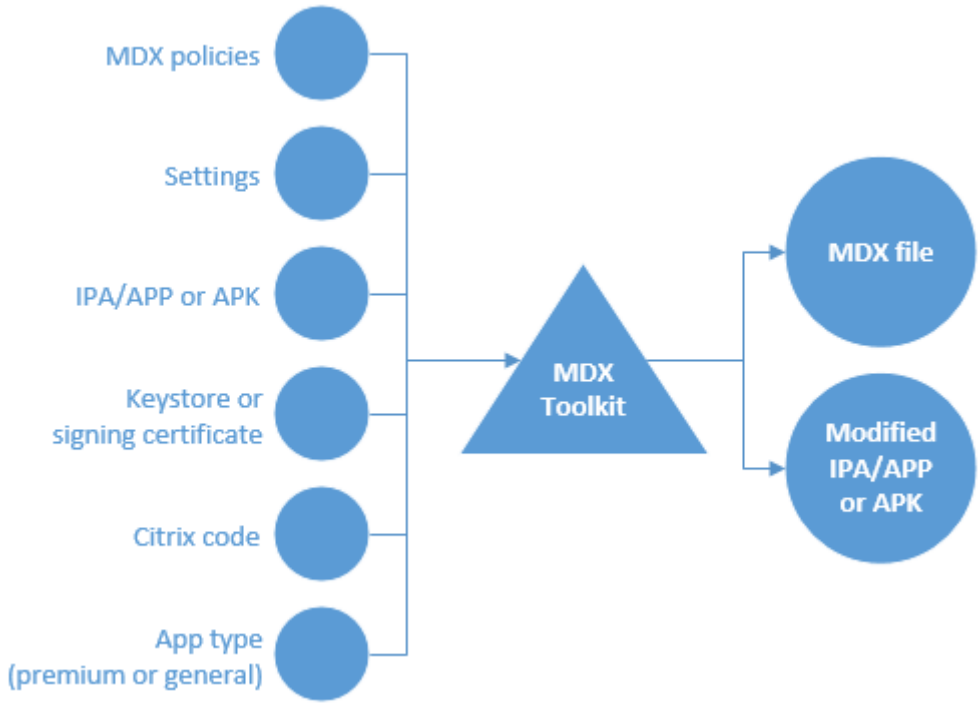


Encapsulation d'applications ISV

Cette section fournit des informations générales sur l'encapsulation d'applications pour les éditeurs de logiciels (ISV). L'encapsulation d'applications réalisée par les administrateurs d'entreprise est abordée dans la section À propos de l'outil MDX Toolkit.

Lorsque vous encapsulez des applications ISV, le MDX Toolkit crée deux fichiers : un fichier .mdx et le fichier applicatif (.ipa, .app ou .apk). Le MDX Toolkit vous permet d'incorporer l'URL du magasin d'applications dans le fichier .mdx que vous pouvez mettre à disposition des clients ou télécharger vers Citrix Ready Marketplace, comme décrit dans la section suivante. Vous pouvez alors mettre le fichier d'application à disposition via des magasins d'applications, en l'hébergeant vous-même, ou en le distribuant auprès de vos clients.

Comme illustré dans le diagramme suivant, le MDX Toolkit combine des fichiers applicatifs (.ipa, .app ou .apk) avec des composants Citrix et votre keystore ou certificat de signature pour produire un fichier .mdx et le fichier d'application modifié.



Les éléments ajoutés par l'encapsulation d'application ISV sont les suivants :

- Un fichier d'informations contenant les données requises par l'infrastructure SDK MDX lorsque l'infrastructure est liée à Secure Hub. Les informations de liaison correspondantes sont transmises à Secure Hub depuis le serveur Endpoint Management via le fichier .mdx ajouté à Endpoint Management. Les données contiennent différents éléments, tels qu'un ID d'application utilisé pour l'auto-identification et un ID de package utilisé pour les vérifications de mise à jour de l'application.
- Une empreinte numérique FIPS sur le OpenSSL FIPS Object Module Crypto incorporé dans l'application intégrée au SDK de l'application MDX.
- Pour iOS uniquement : un nouveau modèle d'URL qui est ajouté au fichier d'application et qui est également transmis à Secure Hub via le fichier .mdx qu'un administrateur ajoute à Endpoint Management.

À propos du programme Citrix Ready

Citrix évalue et certifie les applications ISV via le programme Citrix Ready. L'évaluation implique principalement des tests d'intégration de Endpoint Management. La certification garantit que les applications sont compatibles avec l'infrastructure Endpoint Management, ce qui permet aux entreprises d'utiliser vos applications en toute confiance.

Dans le cadre de ce programme Citrix Ready, vous pouvez publier les fichiers binaires de vos applications ISV certifiées directement dans l'App Store d'Apple ou Google Play Store. Cela signifie que vous n'avez pas besoin de distribuer les fichiers binaires aux entreprises, vous donnant un contrôle plus

important sur les mises à jour de l'application. En outre, les applications sont signées avec votre certificat ISV. Vous pouvez également choisir de distribuer vos applications certifiées directement auprès des entreprises ou de les héberger vous-même.

Vous pouvez également choisir de distribuer le bundle .mdx pour une application ISV : publiez le bundle sur Citrix Ready Marketplace ou distribuez le bundle directement auprès de vos clients Citrix Endpoint Management.

Pour plus de détails, consultez Citrix Ready.

Expérience utilisateur de l'application MDX

La manière dont les utilisateurs interagissent avec une application intégrée au SDK de l'application MDX dépend de la façon dont ils installent et lancent l'application.

L'utilisateur démarre avec Secure Hub

1. L'utilisateur ouvre Secure Hub et l'App Store d'Apple ou Google Play Store.
2. L'utilisateur ouvre une session sur Endpoint Management Store, puis s'abonne au magasin.
3. L'utilisateur télécharge et installe l'application à partir d'un magasin public.
4. Secure Hub invite l'utilisateur à se connecter, si nécessaire.
5. Si l'application était non gérée, elle passe au mode géré de manière silencieuse.

L'utilisateur démarre avec l'App Store d'Apple ou Google Play Store

Si Secure Hub est déjà présent sur l'appareil, les procédures sont les suivantes pour les applications General et Premium.

Application General

1. L'utilisateur lance l'application.
2. Si l'application détecte une installation de Secure Hub et que l'application y est autorisée, l'application invite l'utilisateur à confirmer la transition vers le mode géré.
3. Si l'utilisateur accepte que son entreprise gère l'application, Secure Hub invite l'utilisateur à se connecter, si nécessaire.
4. Une fois que MDX a inscrit l'application pour l'utilisateur, l'application passe en mode géré.

   Si l'application Secure Hub ne se trouve pas sur l'appareil ou si l'application n'est pas autorisée, l'application s'exécute en mode non géré, comme une application de magasin d'applications public standard.

Application Premium

1. L'utilisateur lance l'application.
2. Si l'application détecte une installation de Secure Hub et que l'application y est autorisée, l'application passe en mode géré de façon silencieuse. Si des informations d'identification Secure Hub sont requises, l'application informe l'utilisateur de la transition vers le mode géré et invite l'utilisateur à ouvrir une session.

Problèmes connus avec le SDK de l'application MDX

- L'encapsulation ne fonctionne pas pour les applications Android sauf si elles incluent des icônes.
- Certaines infrastructures d'applications ont des problèmes de compatibilité avec Citrix Endpoint Management. Pour de plus amples informations, reportez-vous à la section Prise en charge des infrastructures de développement d'application mobile (pour Android) et Prise en charge des bibliothèques tierces (pour iOS).
- Pour d'autres problèmes, consultez Problèmes connus.

Configuration système requise

February 19, 2019

Cet article décrit la configuration système requise pour le MDX Toolkit et le SDK de l'application MDX.

MDX Toolkit et SDK de l'application MDX (iOS et Android)

- Java Development Kit (JDK) 1.7 ou 1.8.

  Vous pouvez télécharger le JDK 1.8 depuis Java SE Development Kit Downloads sur le site Web Oracle. Pour obtenir les instructions d'installation, veuillez consulter la section JDK 8 and JRE 8 Installation Guide sur le site Web d'Oracle. Veillez à installer le JDK complet : définissez JDK 1.8 comme valeur par défaut.

- macOS 10.10

  Le programme d'installation de l'outil MDX Toolkit et du SDK de l'application MDX doit être exécuté sur un macOS. Le programme d'installation comprend des outils macOS qui encapsulent les applications iOS et Android, ainsi qu'un outil de ligne de commande Java qui encapsule les applications Android.

- Pour le SDK de l'application MDX : iOS 11 ou version ultérieure avec Xcode 9 ; génération de bitcode désactivée (nous vous recommandons d'utiliser la version la plus récente de Xcode disponible sur Apple).

  La génération de bitcode est activée par défaut. Vous devez désactiver cette option pour utiliser Xcode 9 avec le SDK de l'application MDX.

Autre configuration requise pour encapsuler des applications mobiles iOS

- Pour obtenir l'accès à la configuration requise pour l'encapsulation d'applications pour iOS, vous devez vous enregistrer afin d'obtenir un compte de distribution Apple. Il existe trois types de comptes développeur iOS : Enterprise, Individual et University. Citrix recommande fortement les comptes iOS Developer Enterprise.
  - Comptes iOS Developer Enterprise : seul type de compte Apple Developer qui vous permet de provisionner, déployer et tester un nombre illimité d'applications pour un nombre illimité d'appareils, avec ou sans encapsulation d'application. Veillez à distribuer votre certificat Developer à vos développeurs pour qu'ils puissent signer les applications.
  - Comptes iOS Developer Individual : limité à 100 périphériques inscrits par an et ne permet pas l'encapsulation d'applications d'entreprise ni la distribution d'entreprise avec Citrix Endpoint Management.
  - Comptes iOS Developer University : limité à 200 périphériques inscrits par an et ne permet pas l'encapsulation d'applications d'entreprise ni la distribution d'entreprise avec Endpoint Management.

> **Remarque :**
>
> Téléchargez les outils de ligne de commande Xcode disponibles sur le site Web Xcode Apple Developer. macOS 10.10 n'installe pas les outils automatiquement. Pour installer les outils, suivez ces étapes :

1. Dans **Applications > Utilities**, cliquez sur **Terminal** pour utiliser l'interface de ligne de commande Mac.
2. Exécutez la commande suivante :

   ```
   1 xcode-select --install
   ```

   Veillez à inclure deux tirets avant le mot install dans la commande.
3. Une fois les outils de ligne de commande Xcode installés, exécutez Xcode to pour installer les éléments pré-requis.

Autre configuration requise pour encapsuler des applications mobiles Android

- SDK Android, Niveau API 21 (version minimum prise en charge).
  - Téléchargez le SDK Android à partir de la page de téléchargement du SDK sur le site Web de développeurs Google.
  - Installez les outils SDK Android les plus récents, SDK Android Platform-tools et Android SDK Build-tools.

    Pour de plus amples informations, consultez la section Installing the Android SDK sur le site Web de développeurs Google.
  - Modifiez le fichier android_settings.txt qui se trouve dans le dossier d'installation du MDX Toolkit. Définissez la variable PATH pour inclure les outils de création du SDK Android à utiliser lors de l'encapsulation. Ajoutez le chemin d'accès aux sous-répertoires d'outils et d'outils de plate-forme du SDK Android. Exemple :

    PATH = /Users/Sample/Downloads/android-sdk-macosx/platform-tools:/Users/Sample/Downloads/an
    sdk-macosx/build-tools/28.0.2:/Users/Sample/Downloads/android-sdk-macosx/tools
- Keystore valide (contenant les certificats signés numériquement utilisés pour signer vos applications Android)

  Vous créez un keystore une fois et vous conservez ce fichier pour le réutiliser à des fins d'encapsulation. Si vous n'utilisez pas le même keystore lors de l'encapsulation de la nouvelle version d'une application que vous avez déjà déployée, les mises à niveau de cette application ne fonctionneront pas. Les utilisateurs devront supprimer manuellement la version précédente avant d'installer la version la plus récente.

  Un keystore peut contenir plusieurs clés privées, mais dans la plupart des cas, une seule clé existe.

  Pour de plus amples informations sur les certificats, consultez la section Signing Your Applications sur le site Web des développeurs Android.

  Vous devez signer vos applications avec une clé répondant aux exigences suivantes :
  - Taille de clé 2048 bits
  - Algorithme de clé DSA (keyalg)
  - Algorithme de signature DSA avec SHA1 (sigalg)

Développement d'applications Android

February 19, 2019

Vous pouvez utiliser l'API XenMobile dans vos applications mobiles pour permettre aux applications d'interagir avec Citrix Endpoint Management. Cet article explique comment intégrer le SDK de l'application MDX à votre bibliothèque d'applications et les étapes requises pour tester, certifier et publier vos applications.

Comment utiliser le SDK de l'application MDX

Voici quelques exemples d'utilisation de l'API.

Placer des restrictions sur les applications

Vous pouvez contrôler si votre application autorise l'accès à certaines fonctionnalités ou actions selon que les appels API indiquent que l'application est gérée ou encapsulée. Par exemple, si une application n'est pas gérée ou encapsulée, vous pouvez permettre à un utilisateur d'accéder à toutes les fonctionnalités et actions. Si une application est encapsulée, mais non gérée, vous pouvez restreindre certaines fonctionnalités ou actions. Si une application est encapsulée et gérée, vous pouvez placer des restrictions sur l'application.

Effectuer des actions basées sur les paramètres de stratégie Citrix Endpoint Management

Supposons que vous souhaitez afficher une notification pour les utilisateurs si un administrateur Citrix Endpoint Management définit la stratégie Exiger Wi-Fi sur Activé, ce qui signifie que l'application est autorisée à être exécutée sur un réseau sans fil. Vous pouvez utiliser l'API pour rechercher le paramètre de stratégie et baser les modifications de votre code sur la valeur de stratégie.

- Effectuer des actions basées sur des stratégies personnalisées

  Vous pouvez utiliser l'API pour lire des stratégies personnalisées dans vos applications. Par exemple, supposons que vous voulez autoriser les administrateurs Citrix Endpoint Management à afficher une notification dans l'application. Pour ce faire, créez une stratégie personnalisée qui est vide par défaut ou contient un message système fourni par un administrateur dans la console Citrix Endpoint Management. Lorsque l'application est gérée, elle peut détecter lorsque l'administrateur Citrix Endpoint Management modifie la valeur de stratégie. Si la valeur de stratégie contient un message, votre application affiche la notification.

Pour les définitions de l'API, consultez la section API pour Android.

Intégration du SDK dans votre bibliothèque d'applications à l'aide de Android Studio et Gradle

Pour ajouter le SDK de l'application MDX à vos applications Android, vous devez importer ou copier les bibliothèques Java de l'application MDX dans votre application, comme décrit dans cette section. Les étapes sont basées sur Android Studio et le système Gradle. Cette procédure ajoute la bibliothèque du SDK des applications MDX à votre application afin que ses classes et méthodes soient accessibles pour l'application.

1. Si vous n'avez pas encore installé la dernière version du MDX Toolkit, faites-le maintenant.
   a) Ouvrez une session sur la page des téléchargements Citrix Endpoint Management.
   b) Développez **Applications XenMobile et MDX Toolkit**.
   c) Recherchez la version de MDX Toolkit que vous souhaitez installer, puis cliquez sur le lien pour lancer le téléchargement.
   d) Ouvrez MDXToolkit.mpkg avec l'outil macOS Finder sur macOS 10.9.4 ou version ultérieure et Xcode 5.1 ou version ultérieure.

   Le chemin d'installation est Applications/Citrix/MDXToolkit.

   Les fichiers du SDK de l'application MDX se trouvent dans Applications/Citrix/MDXToolkit/-data/MDXSDK_Android.
2. Après l'installation de l'outil MDX Toolkit, installez Android Studio à partir du site Web de développeurs Android, puis effectuez les opérations suivantes :
   a) Dans le répertoire de projet, créez un dossier nommé libs.
   b) Ajoutez le fichier worxsdk.aar au dossier libs.
   c) Modifiez le projet « build.gradle » pour ajouter une règle qui effectue une recherche dans le dossier libs comme référentiel et inclut worxsdk.aar depuis le dossier libs comme dépendance.
   d) Créez votre fichier APK.

Exemple de worxsdk.aar :

```
1     // Top-level build file where you can add configuration options
         common to all sub-projects/modules.
2     buildscript {
3
4         repositories {
5
6             jcenter()
7          }
8
```



```
9           dependencies {
10
11              classpath 'com.android.tools.build:gradle:1.1.0'
12              // NOTE: Do not place your application dependencies here;
                    they belong
13              // in the individual module build.gradle files
14           }
15
16       }
17
18      allprojects {
19
20          repositories {
21
22              jcenter()
23              flatDir {
24
25                  dirs 'libs'
26               }
27
28           }
29
30       }
31
32      dependencies {
33
34          compile(name:'worxsdk', ext:'aar')
```

La bibliothèque OpenSSL peut entraîner des conflits avec des bibliothèques similaires dans les applications Android. Citrix vous recommande d'utiliser les versions Citrix des bibliothèques afin d'éviter les conflits.

Publication d'une application Android

Après avoir ajouté le SDK de l'application MDX à une application Android, effectuez les étapes suivantes pour encapsuler, tester, certifier et publier l'application. Lors de l'encapsulation d'applications à l'aide de l'interface de ligne de commande, vous devez inclure l'une des options suivantes :

- **–appType Enterprise**
- **-appType Premium**
- **-appType General**

La valeur par défaut est **–appType Enterprise**. Suivez les recommandations suivantes pour sélectionner le type d'application :

- **Enterprise :** les applications d'entreprise nécessitent la réinstallation de Secure Hub sur la machine utilisateur. Vous devez également publier l'application dans StoreFront et installer l'application via Secure Hub.
- **General :** les applications ISV General peuvent fonctionner sans Secure Hub (initialement). L'application peut passer en mode géré lorsqu'elle détecte Secure Hub sur la machine utilisateur, et si vous publiez une application correspondante. Lors de l'exécution d'une application gérée, les applications General fonctionnent de la même façon que les applications d'entreprise. Lors d'une exécution en tant qu'application non gérée, les stratégies Citrix ne sont pas appliquées.
- **Premium :** les applications ISV Premium peuvent s'exécuter sans que Secure Hub soit installé sur la machine utilisateur (initialement). L'application peut passer à une application gérée si elle détecte Secure Hub sur la machine utilisateur, et si vous publiez une application correspondante. Lors de l'exécution d'applications non gérées, MDX doit appliquer certaines stratégies, telles que les stratégies de contention des données (permettant l'accès réseau, la capture d'écran, ou bloquant l'appareil photo).

Si vous avez besoin de télécharger le fichier .apk encapsulé sur un magasin d'applications ou un serveur Web, et que vous connaissez déjà l'adresse URL, ajoutez l'option **-storeURL**. Vous pouvez également ajouter l'adresse URL ultérieurement, comme indiqué plus tard dans ces étapes.

Le MDX Toolkit produit un fichier .apk modifié et un fichier .mdx. Vous allez utiliser ces fichiers dans les étapes suivantes. Utilisez l'outil MDX Toolkit pour encapsuler le fichier .apk pour l'application. Pour de plus amples informations, consultez la section Encapsulation d'applications mobiles Android dans la documentation MDX Toolkit. Cet article contient toutes les commandes d'encapsulation, y compris celles qui sont spécifiques à des applications ISV.

> **Important :**
>
> L'option d'encapsulation des applications ISV à l'aide de l'interface utilisateur de l'outil MDX Toolkit n'est plus disponible. Vous devez encapsuler les applications ISV à l'aide de la ligne de commande.

Pour tester votre application

1. Installez le fichier .apk modifié sur un appareil Android à vérifier toutes les fonctions de l'application.
2. Utilisez la console Citrix Endpoint Management pour ajouter le fichier .mdx à Citrix Endpoint Management et le mettre à la disposition d'un appareil Android à des fins de test. Pour de plus amples informations, consultez la section Pour ajouter une application MDX à Citrix Endpoint Management. Sur cet appareil, testez la fonctionnalité MDX de votre application.

   Si vous avez ajouté des stratégies personnalisées, vérifiez que les stratégies apparaissent dans

la console Citrix Endpoint Management et fonctionnent comme prévu. Si vous avez modifié default_sdk_policies.xml, testez ces modifications. Pour de plus amples informations sur l'ajout de stratégies et la modification des valeurs par défaut des stratégies, veuillez consulter la section Valeurs par défaut des stratégies et stratégies personnalisées.

3. Corrigez les erreurs trouvées dans votre application, régénérez le fichier .apk, et encapsulez de nouveau l'application avec le MDX Toolkit.
4. Envoyez le fichier .apk d'origine (et non pas celui produit par le MDX Toolkit) à Citrix pour validation et certification.
5. Une fois que Citrix a attesté votre application, soumettez le fichier .apk généré par l'outil MDX Toolkit à Google Play Store pour approbation.
6. Une fois que Google a approuvé votre application, exécutez l'outil MDX Toolkit pour mettre à jour l'adresse URL de téléchargement de l'application dans le fichier .mdx. Voici un exemple de commande qui modifie l'URL :

```
java -jar /Applications/Citrix/MDXToolkit/ManagedAppUtility.jar \
setinfo \
-in ~/Desktop/SampleApps/Sample.mdx \
-out ~/Desktop/SampleApps/wrapped/Sample.mdx \
-storeURL \
 "https://play.google.com/store/apps/details?id=com.zenprise"
```

7. Fournissez le fichier .mdx final à un administrateur Citrix Endpoint Management, qui l'ajoutera à Citrix Endpoint Management et le publiera pour les utilisateurs. Ou, pour que votre application soit mise à disposition à une plus grande échelle, vous pouvez afficher votre application MDX vérifiée dans Citrix Ready Marketplace. Pour de plus amples informations, consultez la section Citrix Ready Worx Verified Program.

Considérations pour la mise à niveau des applications

Le logiciel Citrix Endpoint Management change de manière significative d'une version à l'autre. Pour bénéficier des dernières fonctionnalités et corrections de bogues, vous devez utiliser la dernière version de l'outil MDX Toolkit pour encapsuler votre application. Veillez à encapsuler le fichier .ipa ou .apk d'origine et non le fichier modifié qui a été préalablement généré par le MDX Toolkit.

Assurez-vous d'utiliser la version correspondante du SDK de l'application MDX.

Recommandations pour les applications Android

February 19, 2019

Les recommandations abordées dans cet article améliorent la compatibilité entre Citrix Endpoint Management et les applications mobiles pour les appareils Android.

SDK de l'application MDX et encapsulation

Si votre application utilise le SDK de l'application MDX, vous devez utiliser la version de l'outil MDX Toolkit correspondante pour l'encapsulation. Une différence de version entre ces deux composants peut entraîner un fonctionnement incorrect.

Pour éviter ce type d'incohérence, encapsulez l'application avec le type d'application Premium ou General. Cette configuration vous permet de mettre à disposition une application préencapsulée. Par conséquent, le client n'a pas besoin d'encapsuler l'application, ce qui évite l'utilisation d'un MDX Toolkit de version différente. Pour de plus amples informations sur l'encapsulation d'applications, consultez la section Encapsulation d'applications mobiles Android.

Ne pas bloquer le thread principal

Vous ne devriez pas utiliser de code de blocage lors de l'exécution sur le thread principal. Il s'agit d'une recommandation Google, mais elle est encore plus importante avec Citrix Endpoint Management. Certaines actions peuvent prendre plus de temps dans une application gérée ou même bloquer l'exécution d'autres threads.

Le code de blocage inclut notamment :

- Opérations de fichier ou de base de données
- Opérations de réseau

Pour plus de clarté, toutes les méthodes de cycle de vie des applications, comme onCreate, s'exécutent sur le thread principal.

Google fournit une API StrictMode qui peut vous aider à détecter le code de blocage. Pour plus d'informations, consultez ce billet de blog : https://android-developers.blogspot.com/2010/12/new-gingerbread-api-strictmode.html.

Écrire un code robuste

En particulier, vous devez vérifier les valeurs de retour ou intercepter les exceptions provenant des API d'infrastructure. Bien que ce soit une recommandation courante en programmation, elle est particulièrement importante pour les applications gérées.

Plusieurs API qui fonctionnent sans problème habituellement échouent si des stratégies Citrix Endpoint Management bloquent la fonctionnalité sous-jacente. Les fonctionnalités décrites précédemment en sont des exemples :

- Les API de réseau échouent comme s'il n'existait aucun réseau disponible.
- Les API de détection, telles que le GPS et l'appareil photo, renvoient une valeur null ou une exception.
- Les intents redirigés vers une application non gérée échouent.
- L'accès aux fichiers et aux bases de données peut échouer s'il est utilisé dans le thread principal. Pour de plus amples informations, consultez les sections Assurer la compatibilité du cryptage de données et Entropie utilisateur dans cet article.

Lorsque vous rencontrez un échec, votre application doit gérer le problème de façon appropriée plutôt que de se bloquer.

Hooking - Limites

MDX injecte les fonctionnalités dans une application Android binaire en modifiant le code DEX dans le APK. Il existe plusieurs limites :

- Citrix Endpoint Management peut ne pas gérer les classes d'infrastructure des versions de SDK Android antérieures à 4.0. Veillez à éviter ces classes obsolètes.
- La plupart des fonctionnalités sont injectées dans les API d'infrastructure Java/Android. Le code (C/C + +) natif n'est généralement pas géré. Cependant, même pour le code natif, le cryptage de fichier est toujours appliqué.
- Le code natif qui utilise JNI pour accéder à la fonctionnalité Java doit uniquement cibler le code dans l'application utilisateur. En d'autres termes, n'utilisez pas JNI pour invoquer directement les méthodes d'infrastructure Java ou Android. Utilisez plutôt le modèle de conception de proxy pour encapsuler la classe d'infrastructure souhaitée dans une classe Java de votre choix. Appelez ensuite votre classe depuis le code natif.

Assurer la compatibilité du cryptage de données

L'une des fonctionnalités principales de MDX est que toutes les données conservées sont cryptées de façon transparente. Vous n'avez pas besoin de modifier votre application pour bénéficier de cette fonctionnalité, et en fait, vous ne pouvez pas directement l'éviter. L'administrateur a la possibilité de désactiver le cryptage de manière sélective ou complètement, mais pas l'application.

Ceci est l'un des aspects plus complexes de MDX et il est important de comprendre les points suivants :

- Le cryptage de fichier est présent pour l'ensemble du code natif et Java qui s'exécute dans les processus gérés.
- Certaines API d'infrastructure, telles que les lecteurs média et la prise en charge de l'impression, s'exécutent dans des processus de système d'exploitation séparés. Si vous utilisez une telle API, il est possible que vous rencontriez des problèmes.

  - Exemple : votre application enregistre un fichier sur le disque (crypté) et transmet une référence au fichier à une API multimédia. L'API multimédia tente de lire le fichier mais ne comprend pas le contenu crypté. Elle échoue ou bloque l'application.
  - Exemple : vous créez un descripteur de fichier (qui démarre un fichier crypté) et vous l'attribuez à l'API de caméra. Le processus de caméra écrit directement les données non cryptées dans le fichier crypté. Lorsque votre application tente de lire ces données, les données sont décryptées, générant un texte illisible.

- Une solution pour gérer les processus séparés consiste à décrypter un fichier avant de le transmettre à l'API appropriée. Ou, si l'API écrit des données, vous laissez l'API écrire les données, puis vous les cryptez une fois que l'API a terminé. Quelques étapes sont nécessaires :

  1. Désignez une zone qui restera non cryptée. Vous devez documenter cette action pour votre client, car un administrateur Citrix Endpoint Management doit créer une stratégie d'exclusion de cryptage.
  2. Pour décrypter, il vous suffit de copier le fichier depuis l'emplacement normal (crypté) vers l'emplacement décrypté. Notez que vous devez effectuer une copie d'octets et non une opération de déplacement de fichier
  3. Pour crypter, effectuez la même opération dans le sens inverse. Copiez depuis des emplacements non cryptés vers des emplacements cryptés.
  4. Supprimez le fichier non crypté lorsqu'il n'est plus requis.

- Le mappage de mémoire n'est pas pris en charge pour les fichiers cryptés. Si vous appelez une API qui effectue le mappage de mémoire, elle échouera. Vous devez gérer l'erreur. Si possible, évitez d'utiliser de manière directe ou indirecte le mappage de mémoire. Un cas d'utilisation indirecte est la bibliothèque SqlCipher tierce.

  Si vous ne pouvez pas éviter le mappage de mémoire, l'administrateur doit définir une stratégie d'exclusion de cryptage qui ignore les fichiers concernés. Vous devez documenter cette stratégie pour votre client.

- Le cryptage ajoute une charge significative. Veillez à optimiser le traitement I/O des fichiers pour empêcher la dégradation des performances. Par exemple, si vous lisez et écrivez de manière répétée les mêmes informations, il se peut que vous souhaitiez mettre en place un cache au niveau de l'application.

- Les bases de données sont simplement des fichiers et elles sont également cryptées. Vous pouvez rencontrer un problème de performance ici aussi. La taille du cache de base de données standard est 2 000 pages ou 8 Mo. Si votre base de données est volumineuse, vous pouvez augmenter cette taille.

  Le mode SQLite WAL n'est pas pris en charge à cause des limites de mappage de mémoire.

Entropie utilisateur

Une option de cryptage Citrix Endpoint Management nécessite que l'utilisateur entre un code PIN avant que la clé de cryptage puisse être générée. Cette option est appelée entropie utilisateur. Elle peut entraîner un problème particulier pour les applications.

Plus spécifiquement, aucun accès fichier ou base de données ne peut être effectué jusqu'à ce que l'utilisateur entre un code PIN. Si une telle opération E/S est présente à un emplacement qui s'exécute avant que l'interface utilisateur du code PIN puisse être affichée, elle échouera toujours. Ce comportement a plusieurs incidences :

- Gardez les opérations de fichier et de base de données hors du thread principal. Par exemple, une tentative de lecture d'un fichier à partir de la méthode onCreate() de l'objet application échouera toujours.
- Les opérations en arrière-plan, telles que les fournisseurs de services ou de contenu, peuvent être exécutées même sans activité de l'application. Ces composants d'arrière-plan ne peuvent pas afficher l'interface utilisateur de code PIN et, par conséquent, ils ne peuvent pas effectuer d'accès fichier ou base de données. Veuillez noter que lorsqu'une activité s'exécute dans l'application, les opérations en arrière-plan sont autorisées à réaliser des opérations E/S.

Il existe plusieurs mécanismes d'échec si la clé de cryptage n'est pas disponible en raison de l'entropie utilisateur :

- Si le thread principal accède à une base de données avant que le code PIN soit disponible, l'application est arrêtée.
- Si un thread autre que le thread principal accède à une base de données avant que le code PIN soit disponible, ce thread est bloqué jusqu'à ce que le code PIN soit entré.
- Si un accès à un élément autre qu'une base de données est tenté avant que le code PIN soit disponible, l'opération échoue. Au niveau C, une erreur EACCES est renvoyée. Dans Java, une exception est envoyée.

Pour vous assurer que ce problème ne concerne pas votre application, testez-la avec l'entropie utilisateur activée. La propriété du client Citrix Endpoint Management, Encrypt secrets using Passcode, ajoute l'entropie utilisateur. Vous pouvez configurer cette propriété de client, qui est désactivée par défaut, dans la console Citrix Endpoint Management, sous **Configurer > Paramètres > Plus > Propriétés du client**.

Réseau et micro VPN

Les administrateurs disposent de plusieurs options de stratégie Citrix Endpoint Management pour le réseau. La stratégie Accès réseau empêche, permet ou redirige l'activité réseau de l'application :

> Important :
>
> La version 18.12.0 de MDX Toolkit inclut de nouvelles stratégies combinant ou remplaçant des stratégies plus anciennes.
> La stratégie Accès réseau combine Accès réseau, Mode VPN préféré et Autoriser le basculement vers le mode VPN. La stratégie Liste d'exclusion remplace Liste d'exclusion de split tunneling. La stratégie Session micro VPN requise remplace Session en ligne requise. Pour de plus amples informations, consultez la section Nouveautés dans le MDX Toolkit 18.12.0.

Les options sont les suivantes :

- **Utiliser les paramètres précédents :** utilise par défaut les valeurs que vous aviez définies dans les stratégies précédentes. Si vous modifiez cette option, vous ne devez pas revenir à **Utiliser paramètres précédents**. Notez également que les modifications apportées aux nouvelles stratégies ne prennent effet que lorsque l'utilisateur met à niveau l'application vers 18.12.0 ou version ultérieure.
- **Bloqué :** les API de mise en réseau utilisées par votre application échoueront. Conformément à la recommandation précédente, vous devriez traiter un tel échec de façon appropriée.
- **Sans restriction :** tous les appels réseau ont un accès direct et ne sont pas tunnélisés.
- **Tunnel VPN complet :** tout le trafic provenant de l'application gérée est tunnélisé via Citrix Gateway.

Limitation : Citrix Endpoint Management ne prend pas en charge les serveurs socket. Si un serveur socket est exécuté dans l'application encapsulée, le trafic réseau vers le serveur socket n'est pas tunnélisé via Citrix Gateway.

Prise en charge des infrastructures de développement d'application mobile

Certaines infrastructures d'applications ont des problèmes de compatibilité avec Citrix Endpoint Management.

- Avec PhoneGap, le service de géolocalisation n'est pas bloqué.
- SQLCipher ne fonctionne pas avec le cryptage car il utilise le mappage de mémoire. Une solution consiste à ne pas utiliser SQLCipher. Une autre solution consiste à exclure le fichier de base de données du cryptage à l'aide d'une stratégie d'exclusion de cryptage. Un administrateur Citrix Endpoint Management doit configurer la stratégie dans la console Citrix Endpoint Management.

Conseils de débogage

Lors du débogage d'une application encapsulée, suivez ces conseils.

- Déterminez si le problème est présent dans une version non encapsulée de l'application. Si le problème se produit avec l'application non encapsulée, utilisez les techniques de débogage normales.
- Essayez de désactiver plusieurs stratégies Citrix Endpoint Management.
  - Cela permet de localiser toute incompatibilité. La désactivation d'une stratégie signifie que MDX n'applique plus la restriction associée, ce qui vous permet de tester ces fonctionnalités comme si l'application n'était pas encapsulée.
  - Si la désactivation d'une stratégie résout le problème, il se peut que l'application ne vérifie pas la présence d'erreurs dans les API associées.
- Si une application non modifiée mais avec signature renouvelée ne s'exécute pas :
  1. Extrayez le contenu du APK à l'aide de JAR :
     jar xvf {some.apk}
  2. Supprimez le dossier META-INF :
     rm -rf META-INF
  3. Compressez de nouveau le contenu dans un nouveau APK à l'aide de JAR :
     jar cvf {/tmp/new.apk} *
  4. Signez le nouveau APK à l'aide de JARSIGNER :
     jarsigner -keystore {some.keystore} -storepass {keystorepassword} -keypass {keypassword} {/tmp/new.apk} {keyalias}
  5. Si l'application ne s'exécute toujours pas, vous ne pouvez pas encapsuler l'application à l'aide d'un certificat de signature différent de l'APK utilisé à l'origine.
- Si un .apk décompilé ou recompilé ne s'exécute pas :
  1. Décompilez et recompilez à l'aide de APKTOOL :
     apktool d {some.apk} -o {some.directory}
     apktool b {some.directory} -o {new.apk}
  2. Signez l'APK à l'aide de JARSIGNER comme décrit ci-dessus.
  3. Si l'application ne s'exécute toujours pas, il s'agit d'un bogue APKTOOL externe.
- Si l'encapsulation d'application ne fonctionne pas :
  1. Essayez de supprimer l'infrastructure APKTOOL et de réencapsuler.
     - Mac/Linux : **rm -rf ~/Library/apktool/framework**
     - Windows: **del /q /s C:\Users\{username}\apktool\framework**
  2. Comparez l'outil APKTOOL utilisé par le wrapper avec celui que vous avez utilisé pour décompiler et recompiler avec succès à l'étape précédente.
     - S'il s'agit de la même version d'APKTOOL, il existe un bogue dans le wrapper.
     - S'il s'agit d'une autre version d'APKTOOL, il peut s'agir d'un bogue dans le APKTOOL intégré dans le MDX Toolkit.
       a) Extrayez le contenu de ManagedAppUtility.jar.
       b) Remplacez par le contenu de l'APKTOOL.jar que vous avez utilisé pour encapsuler l'application avec succès dans l'étape précédente.

c) Compressez de nouveau le contenu dans un nouveau ManagedAppUtility.jar.
d) Encapsulez l'application pour confirmer le bogue dans l'APKTOOL intégré.

- Exécutez l'application encapsulée et capturez les informations du journal.
  1. Utilisez grep pour déterminer ce qui se produit dans l'application.
     Pour suivre les activités de l'application : grep "MDX-Activity"
     Pour suivre le verrouillage MDX de l'application : grep "MDX-Locked"

Pour afficher les deux journaux : egrep "MDX-Act MDX-Loc"

  2. Si une erreur indiquant que l'application ne répond pas s'affiche, vous pouvez extraire les traces ANR à l'aide d'ADB.
- Si un problème se produit lors de l'interaction avec de multiples applications, par exemple lors de l'utilisation de la fonction Ouvrir dans :
  1. Vérifiez que les stratégies de cryptage et les paramètres de groupe de sécurité sont les mêmes pour toutes les applications.
  2. Essayez une autre application. Il peut exister un bogue dans l'une des applications testées.
  3. Capturez les journaux de toutes les applications concernées. Notez que Secure Hub peut regrouper les journaux et envoyer les journaux par e-mail depuis les applications individuelles. À partir de l'écran de My Apps (Mes applications), balayez vers la droite pour accéder à l'écran d'assistance. Ensuite, cliquez sur le bouton Need Help (Besoin d'aide) dans le bas de l'écran.

Outre les outils mentionnés ci-dessus, les opérations suivantes peuvent également vous aider :

- Utilisez AAPT pour vider les informations sur l'application.

  aapt dump badging {some.apk}

- Utilisez la commande DUMPSYS sur l'appareil.

  adb shell dumpsys 2>&1 | tee {dumpsys.out}

- Utilisez DEX2JAR pour recompiler les classes en pseudo-Java.

  dex2jar {some.apk}

  Convertissez les classes depuis les applications encapsulées Dual-Dex :

  apktool d {some.apk} -o {some.dir}

  dex2jar {some.dir}/assets/secondary-1.dex

- Utilisez JD-GUI pour afficher le code pseudo-Java.
- Utilisez BAKSMALI pour décompiler les classes d'application depuis les applications encapsulées Dual-Dex.

- Décompilez l'APK encapsulé :

  apktool d {some.apk} -o {some.dir}

- Décompilez les classes de l'application qui ne sont pas décompilées par l'appel ci-dessus :

  baksmali {some.dir}/assets/secondary-1.dex -o {some.dir}/smali

API pour Android

February 19, 2019

L'API pour Android est basée sur Java. Cet article décrit les API Citrix Endpoint Management par fonctionnalité et fournit les définitions de l'API.

Gestion des applications :

- isManaged
- isWrapped

Stratégies MDX :

- getPoliciesXML
- getPolicyValue
- setPolicyChangeMessenger

Coffre partagé :

- MDXDictionary

Données utilisateur :

- getUserName

Class com.citrix.worx.sdk.MDXApplication

Méthodes

- **isManaged**

  public static boolean isManaged (Context context)

  Vérifie si l'application est actuellement gérée par MDX, ce qui signifie que l'application Citrix Secure Hub est installée sur l'appareil et que les stratégies Citrix Endpoint Management sont appliquées sur l'application. L'infrastructure principale de Endpoint Management (coffres de clés) est interrogée pour obtenir les clés partielles de cryptage des données (secrets) que MDX utilisera pour crypter les données de fichier de l'application. Renvoie true si l'application est gérée.

Les applications Premium non gérées utilisent les valeurs par défaut de stratégie Endpoint Management spécifiées dans Applications/Citrix/MDXToolkit/data/MDXSDK_Android/default_sdk_policies.xml. Les stratégies ne sont pas appliquées pour les applications General non gérées.

Paramètres

context – contexte Android qui effectue cet appel.

Exemple

```
boolean bIsManaged = MDXApplication.isManaged(context);
```

- **isWrapped**

  public static boolean isWrapped (Context context)

  Renvoie true si l'application est encapsulée avec le MDX Toolkit.

  Paramètres

  context – contexte Android qui effectue cet appel.

  Exemple

  ```
  boolean bIsWrapped = MDXApplication.isWrapped(context);
  ```

- **getUserName**

  public static String getUserName (Context context)

  Renvoie une chaîne contenant le nom d'utilisateur de l'utilisateur inscrit exécutant une application gérée par MDX, quel que soit l'état de connexion de l'utilisateur. Ne renvoie aucune valeur si l'utilisateur n'est pas inscrit, l'application n'est pas gérée, ou l'application n'est pas encapsulée.

  Paramètres

  context – contexte Android qui effectue cet appel.

  Exemple

  ```
  String userName = MDXApplication.getUserName(context);
  ```

Class com.citrix.worx.sdk.MDXPolicies

Méthodes

- **getPoliciesXML**

  public static String getPoliciesXML (Context context)

  Renvoie le contenu de default_sdk_policies.xml, avec une ligne par stratégie, précédé de (match) pour indiquer que la valeur dans le fichier XML de données correspond à la valeur renvoyée par **MDXPolicies.getPolicyValue()**. Renvoie une chaîne vide en cas d'échec.

Paramètres

context – contexte Android qui effectue cet appel.

Exemple

```
String policiesXML = MDXPolicies.getPoliciesXML(context);
```

- **getPolicyValue**

  public static String getPolicyValue (Context context, String policyName)

  Renvoie une **chaîne** qui contient la valeur courante de la stratégie nommée. Renvoie **null** si aucune valeur n'est trouvée.

  Paramètres

  context – contexte Android qui effectue cet appel.

  policyName – nom de la stratégie à rechercher. Le nom de la stratégie correspond à la valeur de l'élément *PolicyName* dans un fichier XML de stratégie.

  Exemple

  ```
  String value = MDXPolicies.getPolicyValue(context"DisableCamera");
  ```

- **setPolicyChangeMessenger**

  public static String setPolicyChangeMessenger (Context context, String policyName. Messenger messenger)

  Enregistre une messagerie pour recevoir un message lorsque la valeur de la stratégie donnée est modifiée. Lorsque MDX détecte qu'une valeur de stratégie a été modifiée dans la console Citrix Endpoint Management, MDX avertit cette messagerie. Vous pouvez utiliser les autres API pour relire les valeurs de stratégie et modifier l'application. Renvoie **null**.

  Paramètres

  context – contexte Android qui effectue cet appel.

  policyName – nom de stratégie à contrôler. Le nom de la stratégie correspond à la valeur de l'élément *PolicyName* dans un fichier XML de stratégie.

  messenger – messagerie qui recevra des messages lorsque la valeur de stratégie sera modifiée.

  Exemple

  ```
  MDXPolicies.setPolicyChangeMessenger(context, "DisableCamera", messenger
  );
  ```


Classe com.citrix.mdx.common.MDXDictionary

MDXDictionary est un conteneur qui lit et stocke les bundles Android cryptés de paires clé-valeur. Les applications de productivité mobiles dans le même groupe de sécurité MDX partagent un dictionnaire. Utilisez l'API de coffre sécurisé pour partager le contenu géré entre les applications qui ont le même dictionnaire MDX. Par exemple, vous pouvez partager des certificats et des clés privées par le biais d'une application inscrite de façon à ce que les applications puissent obtenir un certificat depuis le coffre sécurisé plutôt que depuis Secure Hub.

Les dictionnaires sont stockés sans cryptage quels que soient les paramètres de stratégie de cryptage de fichiers privés et de stratégie de cryptage de fichiers publics. Les développeurs doivent déverrouiller le coffre avant de récupérer les dictionnaires.

Constructeurs

- **public MDXDictionary(MDXDictionary source)**

  Construit une copie d'un MDXDictionary existant.

  Paramètres

  source – MDXDictionary qui doit être copié.

- **public MDXDictionary(String name, Bundle bundle, long sequence)**

  Construit un MDXDictionary à partir d'un nom, d'un bundle et d'un numéro de séquence. Si vous ne connaissez pas le numéro de séquence, utilisez la méthode **create() factory**.

  Paramètres

  name – nom du dictionnaire.

  bundle – bundle Android.

  sequence – numéro de séquence.

Méthodes

- **public static MDXDictionary create(Context context, String name)**

  Crée un dictionnaire en commençant par vérifier si un dictionnaire avec le même nom existe déjà. Si le dictionnaire n'existe pas, un nouveau dictionnaire est renvoyé. Sinon, le dictionnaire existant est renvoyé. Cette méthode ne renvoie jamais la valeur **null**.

  Paramètres

  context – contexte Android qui effectue cet appel.

name – nom du dictionnaire.

Exemple

// Crée une instance d'un dictionnaire.

```
MDXDictionary dict = MDXDictionary.create(getContext(), "app-settings")
;
```

- **public static boolean delete(Context context, String name)**

  Supprime un dictionnaire par nom. Renvoie **true** en cas de succès ; renvoie **false** en cas d'échec.

  Paramètres

  context – contexte Android qui effectue cet appel.

  name – nom du dictionnaire.

  Exemple

  // Crée une instance d'un dictionnaire.

  ```
  MDXDictionary.delete(getContext(), "app-settings");
  ```

- **public static MDXDictionary find(Context context, String name)**

  Recherche un dictionnaire existant. Renvoie un dictionnaire existant ; renvoie une valeur **null** si aucun dictionnaire n'est trouvé.

  Paramètres

  context – contexte Android qui effectue cet appel.

  name – nom du dictionnaire.

  Exemple

  ```
  MDXDictionary dict = MDXDictionary.find(getContext(),"app-settings");
  ```

  ```
  1  if( dict != null )
  2      {
  3
  4          // Use dictionary
  5       }
  ```

- **public boolean isNew( )**

  Vérifie s'il s'agit d'un nouveau dictionnaire ou d'un dictionnaire existant. Renvoie **true** si un dictionnaire n'existe pas déjà.

  Exemple

  ```
  MDXDictionary dict = MDXDictionary.create(getContext(), "app-settings")
  ;
  ```



```
if (dict.isNew())
    {

        // Dictionary was not found.
     }

else
    {

        // Existing dictionary was found.
     }
```

- **public boolean save(Context context)**

  Stocke un dictionnaire crypté. Si un dictionnaire avec le même nom existe, il sera remplacé. Renvoie **true** en cas de succès ; renvoie **false** en cas d'échec.

  Paramètres

  context – contexte Android qui effectue cet appel.

  Exemple

```
MDXDictionary dict = MDXDictionary.find(getContext(), "app-settings");
```

```
if( dict != null )
    {

        String certificate = getCertificate();
        dict.bundle.putString( &quot;secret-certificate&quot;,
            certificate );
        // Update bundle by overwriting the existing bundle.
        dict.save( getContext() );
     }
```

- **public boolean append(Context context)**

  Ajoute un dictionnaire crypté à un dictionnaire existant. Si aucun dictionnaire n'existe, le dictionnaire spécifié est stocké. Renvoie **true** en cas de succès ; renvoie **false** en cas d'échec.

  Paramètres

  context – contexte Android qui effectue cet appel.

  Exemple

```
MDXDictionary dict = MDXDictionary.find(getContext(), ";app-settings");
```



```
if( dict != null )
    {

        String certificate = getCertificate();
        Bundle bundle = new Bundle();
        bundle.putString( &quot;secret-certificate&quot;,
           certificate );
        dict.bundle = bundle;
        dict.append( getContext() );
        // Note that dict.bundle may not match the state of the
        // bundle that was stored. The stored bundle could be
        // larger.
     }
```

- **public boolean delete(Context context)**

  Supprime le dictionnaire. Renvoie **true** en cas de succès ; renvoie **false** en cas d'échec.

  Paramètres

  context – contexte Android qui effectue cet appel.

  Exemple

```
MDXDictionary dict = MDXDictionary.find(getContext(), "app-settings");
```

```
if( dict != null )
    {

        dict.delete( getContext() );
     }
```

Notes et considérations

- Les constructeurs renvoient une exception **IllegalArgumentException** lorsque des paramètres incorrects sont transmis.
- L'opération **create()** ne renvoie jamais la valeur null. Si la stratégie de cryptage est activée, l'utilisateur doit s'assurer qu'elle est déverrouillée avant l'appel de **create()**.
- L'opération **append()** peut échouer si un objet stocké qui peut être analysé ou sérialisé n'est pas un type de données Java ou Android connu. Secure Hub ne peut pas rétablir le dictionnaire car la classe n'est pas connue en interne par Secure Hub.
- L'opération **append()** ajoute son bundle à un bundle de dictionnaire existant. Si le bundle stocké est différent du bundle du dictionnaire, le bundle local ne reflète pas l'état du bundle

stocké. Une opération **find()** ou **create()** est nécessaire pour interroger l'état du bundle stocké précédemment.

Développement d'applications iOS

January 23, 2019

Vous pouvez utiliser l'API MDX pour activer vos applications mobiles pour Citrix Endpoint Management. Cet article explique comment intégrer le SDK de l'application MDX à votre bibliothèque d'applications et les étapes requises pour tester, certifier et publier vos applications.

Comment utiliser le SDK de l'application MDX

Voici quelques exemples d'utilisation de l'API.

- Placer des restrictions sur les applications

  Vous pouvez contrôler quand votre application autorise l'accès à certaines fonctionnalités ou actions selon que les appels API indiquent que l'application est gérée ou encapsulée. Par exemple, si une application n'est pas gérée ou encapsulée, vous pouvez permettre à un utilisateur d'accéder à toutes les fonctionnalités et actions. Si une application est encapsulée, mais non gérée, vous pouvez restreindre certaines fonctionnalités ou actions. Si une application est encapsulée et gérée, vous pouvez placer des restrictions supplémentaires sur l'application.

- Effectuer des actions basées sur les paramètres de stratégie Citrix Endpoint Management

  Supposons que vous souhaitez afficher une notification pour les utilisateurs si un administrateur Citrix Endpoint Management définit la stratégie Exiger Wi-Fi sur
  Activé. Cela signifie que l'application est autorisée à être exécutée à l'intérieur du réseau de votre organisation. Vous pouvez utiliser l'API pour rechercher le paramètre de stratégie et baser les modifications de votre code sur la valeur de stratégie.

- Effectuer des actions basées sur des stratégies personnalisées

  Vous pouvez utiliser l'API pour lire des stratégies personnalisées dans vos applications. Par exemple, supposons que vous voulez autoriser les administrateurs Citrix Endpoint Management à afficher une notification dans l'application. Pour ce faire, créez une stratégie personnalisée qui est vide ou contient un message système fourni par un administrateur dans la console Citrix Endpoint Management. Si l'application est gérée, elle peut détecter lorsque l'administrateur Citrix Endpoint Management modifie la valeur de stratégie. Si la valeur de stratégie contient un message, votre application affiche la notification.

Pour les définitions de l'API, consultez la section API pour iOS.

Intégration du SDK dans votre bibliothèque d'applications

Pour ajouter le SDK de l'application MDX à vos applications iOS, liez l'infrastructure SDK à votre application, comme décrit dans cette section. Le SDK de l'application MDX pour iOS, basé sur Objective-C, est une série de fichiers d'en-tête et une bibliothèque statique.

1. Si vous n'avez pas encore installé la dernière version du MDX Toolkit, faites-le maintenant.
   a) Ouvrez une session sur la page des téléchargements Citrix Endpoint Management.
   b) Développez **Applications XenMobile et MDX Toolkit**.
   c) Recherchez la version de MDX Toolkit que vous souhaitez installer, puis cliquez sur le lien pour lancer le téléchargement.
   d) Ouvrez MDXToolkit.mpkg avec l'outil Finder sur macOS X 10.9.4 ou version ultérieure et Xcode 7 ou version ultérieure.

      Pour Xcode 8 et versions ultérieures, un problème connu existe où le fichier de projet doit être nettoyé avant l'envoi de l'application sur l'appareil.

      Le chemin d'installation est Applications/Citrix/MDXToolkit.

   Les fichiers du SDK de l'application MDX se trouvent dans Applications/Citrix/MDXToolkit/-data/MDXSDK.

   Après avoir installé le MDX Toolkit sur votre ordinateur, intégrez l'infrastructure MDX à votre projet Xcode.

2. Ajoutez le dossier data/MDXSDK au projet Apple Xcode. Pour ce faire, vous pouvez faire glisser ce dossier vers le projet Xcode.

3. Modifiez une ligne de code dans le fichier d'en-tête précompilé dans le projet de l'application pour importer MDX.h depuis MDX.framework comme illustré dans l'exemple suivant.

```
#ifdef __OBJC__
_
//import MDX extensions
#import <AVFoundation/AVFoundation.h>
#import <SystemConfiguration/SCNetworkReachability.h>
#import <MDX/MDX.h>
#endif
```


Si vous n'incluez que la version réseau de l'infrastructure MDX, vous devez remplacer

```
#import <MDX/MDX.h.>
```

par

```
#import <MDXNetworkOnly/MDXNetworkOnly.h>
```

Si vous encapsulez une application qui émet explicitement un appel à une API exposée par l'infrastructure MDX SDK, les lignes MDX.h et MDXNetworkOnly.h sont facultatives.

Si une application émet explicitement un appel d'API SDK MDX, elle doit être liée au binaire MDX.Framework ou MDXNetworkOnly.Framework et l'incorporer lorsque l'application est créée.

Une application d'entreprise tierce préconfigurée et encapsulée par le MDX Toolkit ne nécessite aucune modification de compilation, car elle ne lance aucun appel d'API SDK MDX explicite.

Après avoir installé le MDX Toolkit sur votre ordinateur, intégrez l'infrastructure MDX à votre projet Xcode.

4. Faites glisser le fichier « data/MDXSDK/MDX.framework » (ou le fichier « data/MDXSDK/MDXNetworkOnly.framework ») vers la section Embedded Binaries du panneau General de l'espace de travail de l'application. Ce faisant, vous ajoutez cette infrastructure dynamique à l'infrastructure incluse dans le bundle d'application installé avec l'application. L'infrastructure est également automatiquement ajoutée à la liste des infrastructures qui sont liées à l'application.

Vous ne devez ajouter qu'une seule infrastructure MDX.

1. Faites glisser le fichier « data/MDXSDK/CitrixLogger.framework » vers la section Embedded Binaries du panneau General de l'espace de travail de l'application.
2. Ajoutez un script d'exécution pour supprimer les architectures des infrastructures incorporées qui n'apparaissent pas dans la liste des architectures valides Xcode. Cela répond à l'exigence d'Apple selon laquelle les infrastructures incorporées ne peuvent pas contenir les architectures iOS Simulator pour les versions d'applications Apple Store. Ce script gère automatiquement toutes les compilations cibles, qu'elles proviennent ou non de l'Apple Store.

```
echo "Strip unnecessary archs from Embedded Frameworks"
cd "${
 BUILT_PRODUCTS_DIR }
 /${
 FRAMEWORKS_FOLDER_PATH }
 "
for file in $(find . -type f -perm +111);
do
```



```
    if ! [[ "$(file "$file")" == \*"dynamically linked shared
        library"\* ]];
    then
        continue
    fi
    # Get architectures for current file
    archs="$(lipo -info "${
 file }
 " | rev | cut -d ':' -f1 | rev)"
    # Strip any archs from frameworks not valid for current app
        build
    for arch in $archs;
    do
        if ! [[ "${
 VALID_ARCHS }
 " == \*"$arch"\* ]];
        then
            lipo -remove "$arch" -output "$file" "$file" || exit 1
        fi
    done
done
```

3. Ajoutez un script à exécuter pour ajouter la ligne de commande SDKprep.
    - Sélectionnez votre projet dans Xcode, puis sélectionnez l'onglet **Build Phases**. Cliquez sur l'icône plus (+) située dans le coin supérieur gauche, puis cliquez sur **New Run Script Phase**.
    - Ouvrez le nouveau script d'exécution, puis saisissez le texte suivant dans le champ **Script**. Veillez à remplacer les variables PACKAGEID, APPTYPE, STOREURL et POLICYFILE par les valeurs qui s'appliquent à votre application. PACKAGEID est un identifiant unique pour votre application, généralement un UUID. Ceci n'est pas obligatoire car le MDX Toolkit génère un ID de package unique chaque fois que l'application est créée. Si vous fournissez un ID de package, assurez-vous qu'il est unique pour chaque nouvelle version d'application que vous encapsulez à l'aide de cette commande.
    - S'il s'agit d'une application Enterprise, utilisez le paramètre *-Apptype Enterprise*, qui correspond à la valeur par défaut. Pour les applications ISV, vous pouvez utiliser les valeurs Premium ou General.

> **Remarque :**
>
> les mots-clés pris en charge pour APPTYPE sont Enterprise, Premium et General.

```
    export PACKAGEID="your-project-PackageID"
    export APPTYPE="keyword"
    export STOREURL="http://your-store-URL"
```



```
    export DATE=‘date +%Y-%m-%d_%H-%M-%S‘
    export POLICYFILE=${
SRCROOT }
/${
EXECUTABLE_NAME }
/${
EXECUTABLE_NAME }
\_policy_metadata.xml
    /Applications/Citrix/MDXToolkit/CGAppCLPrepTool SdkPrep -in ”$
        {
CODESIGNING_FOLDER_PATH }
” -out ”/Users/<UserName>/Downloads/${
EXECUTABLE_NAME }
\_${
DATE }
.mdx” -storeUrl ”${
STOREURL }
” -appIdPrefix ”ABCDEFGH” -packageId ”${
PACKAGEID }
” -policyXML ”${
POLICYFILE }
” -appType ”${
APPTYPE }
” -entitlements ”${
CODE_SIGN_ENTITLEMENTS }
”
```

Exemple :

```
    export PACKAGEID=”a96d6ed5-6632-4739-b9b6-9ad9d5600732”
    export APPTYPE=”Enterprise
    export STOREURL=”http://example.com/12345”
    export DATE=‘date +%Y-%m-%d_%H-%M-%S‘
    export POLICYFILE=${
SRCROOT }
/${
EXECUTABLE_NAME }
/${
EXECUTABLE_NAME }
\_policy_metadata.xml
    /Applications/Citrix/MDXToolkit/CGAppCLPrepTool SdkPrep -in ”$
        {
CODESIGNING_FOLDER_PATH }
” -out ”/Users/<UserName>/Downloads/${
EXECUTABLE_NAME }
```



```
\_${
DATE }
.mdx" -storeUrl "${
STOREURL }
" -appIdPrefix "ABCDEFGH" -packageId "${
PACKAGEID }
" -policyXML "${
POLICYFILE }
" -appType "${
APPTYPE }
" -entitlements "${
CODE_SIGN_ENTITLEMENTS }
"
```

Paramètres	Description
-in *nom du fichier*	Chemin d'accès au fichier .app généré par Xcode. Le MDX Toolkit intègre des ressources spécifiques à MDX dans ce fichier.
-out *nom du fichier*	Chemin de destination pour le fichier .mdx. Utilisez ce fichier pour publier l'application sur le serveur Citrix Endpoint Management.
-storeURl *URL*	URL du magasin d'applications pour l'application, incorporée dans le fichier .mdx. Impossible d'utiliser ce paramètre avec -StoreURL.
-appType *mot-clé*	Les mots-clés sont"Enterprise," "Premium," and "General."

Paramètres	Description
-packageId *UUID*	ID de package unique pour cette application, généralement UUID. Non obligatoire, car le MDX Toolkit génère un ID de package unique chaque fois que l'application est créée. Si vous fournissez un ID de package, assurez-vous qu'il est unique pour chaque nouvelle version d'application que vous encapsulez à l'aide de cette commande. Un ID unique est associé à chaque profil de provisioning. Si vous ouvrez le profil de provisioning (.mobileprovision) dans un éditeur de texte, vous voyez la balise XML ci-dessous avec l'UUID. `<key>UUID</key> <string>4e38fb18-88b0-4806-acfa-e08bf38ec48d</string>`
-policyXML *nom du fichier*	Chemin d'accès au fichier du modèle de stratégie MDX pour votre application.
-entitlements *nom du fichier*	Obligatoire (introduit dans la version 10.3.10). Chemin d'accès au fichier de droits de licence pour l'application. Le MDX Toolkit ajoute à ce fichier une entrée de groupe de trousseau d'accès pour com.citrix.mdx. Ceci est nécessaire pour que votre application partage les secrets avec d'autres applications MDX signées avec le même certificat, à l'aide du trousseau iOS.
-appIdPrefix *préfixe*	Préfixe identifiant d'application - l'identifiant d'équipe associé à votre compte de développeur Apple.

4. Compilez votre projet et générez les fichiers binaires de l'application.
    - Créez votre application dans Xcode et vérifiez qu'elle est créée correctement.
    - Archivez votre application en sélectionnant **Product > Archive**.
    - Xcode Organizer s'ouvre automatiquement une fois que votre application est archivée.
    - Sélectionnez votre build archivée dans Organizer, puis cliquez sur Export.
    - Sélectionnez la méthode d'exportation, puis cliquez sur Next.

    Suivez les invites d'exportation de votre application dans un fichier IPA.

5. Compilez et archivez le projet pour générer le bundle d'application qui contient l'infrastructure MDX incorporée, c'est-à-dire le package .ipa. Xcode génère un fichier MDX correspondant que vous chargez sur le serveur Citrix Endpoint Management. Une fois que les étapes de création et d'archivage de Xcode ont créé le bundle IPA, exécutez la commande SetInfo sur le fichier MDX. Exécutez également l'option de commande –EmbedBundle pour insérer le fichier IPA final dans le fichier MDX. Ensuite, vous pouvez charger l'application sur Citrix Endpoint Management.

```
/Applications/Citrix/MDXToolkit/CGAppCLPrepTool SdkPrep -in "${
 CODESIGNING_FOLDER_PATH }
 " -out "/Users/<UserName>/Downloads/${
 EXECUTABLE_NAME }
 _${
 DATE }
 .mdx" " -embedBundle "/Users/deva/Desktop/{
 EXECUTABLE_NAME }
 .ipa"
```

6. Si vous avez configuré l'application afin qu'elle soit distribuée via le site Web iTunes Connect, vous pouvez également l'envoyer directement vers le magasin d'applications ou TestFlight.

Considérations pour la mise à niveau des applications

Le logiciel Citrix Endpoint Management peut changer de manière significative d'une version à l'autre. Pour bénéficier des dernières fonctionnalités et corrections de bogues, vous devez utiliser la dernière version de l'outil MDX Toolkit pour encapsuler votre application. Veillez à encapsuler le fichier .ipa ou .apk d'origine et non le fichier modifié qui a été préalablement généré par le MDX Toolkit.

Assurez-vous d'utiliser la version correspondante du SDK de l'application MDX.

Recommandations pour les applications iOS

February 19, 2019

Lors du développement d'applications iOS, utilisez ces recommandations pour améliorer la compatibilité entre Citrix Endpoint Management et les applications mobiles pour appareils iOS.

Infrastructure du SDK de l'application MDX et encapsulation

Si votre application utilise l'infrastructure du SDK de l'application MDX, vous devez utiliser la version de l'outil MDX Toolkit correspondante pour l'encapsulation. Une différence de version entre ces deux composants peut entraîner un fonctionnement incorrect.

Pour éviter ce type d'incohérence, encapsulez l'application en tant qu'application ISV et spécifiez un mode d'application Premium ou General. Cette configuration vous permet de mettre à disposition une application préencapsulée. Par conséquent, le client n'a pas besoin d'encapsuler l'application, ce qui évite l'utilisation d'un MDX Toolkit de version différente. Pour de plus amples informations sur l'encapsulation ISV, consultez la section Encapsulation d'applications mobiles iOS.

Utiliser des ID d'application explicites

Si votre compte iOS Developer Enterprise ne prend pas en charge les ID d'application génériques, vous devez créer un ID d'application explicite pour chaque application que vous voulez encapsuler avec le MDX Toolkit et créer un profil de provisioning pour chaque ID d'application.

Ne pas bloquer le thread principal

Vous ne devriez pas utiliser de code de blocage lors de l'exécution sur le thread principal. Il s'agit d'une recommandation Apple, mais elle est encore plus importante avec Citrix Endpoint Management. Certaines actions peuvent prendre plus de temps dans une application gérée ou même bloquer l'exécution d'autres threads. Les opérations de fichier, de base de données et de réseau sont des exemples d'opérations qui peuvent bloquer le thread en cours d'exécution et doivent être évitées sur le thread principal.

Écrire un code robuste

En particulier, vous devriez écrire des applications en suivant les recommandations indiquées dans les guides de programmation d'Apple, tels que le Guide de programmation d'applications Apple.

Utilisez uniquement des interfaces publiées par Apple.

Vérifiez toujours les valeurs renvoyées pour tous les appels API et gérez les exceptions qui peuvent être provoquées par l'appel API, pour garantir une récupération normale après erreur ou la fermeture normale de l'application. Bien que ce soit une recommandation courante en programmation, elle est particulièrement importante pour les applications gérées.

Plusieurs API qui fonctionnent généralement sans problème échouent si la fonctionnalité sous-jacente a été bloquée en raison de stratégies Citrix Endpoint Management. Les fonctionnalités décrites précédemment en sont des exemples :

- Les API de réseau échouent comme s'il n'existait aucun réseau disponible.
- Les API de détection, telles que le GPS et l'appareil photo, renvoient une valeur null ou une exception.

Les sélecteurs d'exécution Objective-C suivants renvoient une valeur null si la fonctionnalité sous-jacente a été bloquée en raison de stratégies Citrix Endpoint Management et doivent être traités en conséquence.

Classe d'objet : AVCaptureDevice

- **Nom du sélecteur :** devicesWithMediaType:

Classe d'objet : MFMailComposeViewController

- **Nom du sélecteur :** init:

Classe d'objet : MFMessageComposeViewController

- **Nom du sélecteur :** initWithNibName:bundle:

Classe d'objet : NSFileManager

- **Nom du sélecteur :** URLForUbiquityContainerIdentifier:

Classe d'objet : NSUbiquitousKeyValueStore

- **Nom du sélecteur :** defaultStore:

Classe d'objet : PHPhotoLibrary

- **Nom du sélecteur :** sharedPhotoLibrary:

Classe d'objet : UIImagePickerController

- **Nom du sélecteur :** availableCaptureModesForCameraDevice:

Classe d'objet : UIPasteboard

- **Nom du sélecteur :**

  dataForPasteboardType:

  valueForPasteboardType:

  items:

  dataForPasteboardType:inItemSet:

  valuesForPasteboardType:inItemSet:

Classe d'objet : UIPopoverController

- **Nom du sélecteur :** initWithContentViewController :

Classe d'objet : UINavigationController

- **Nom du sélecteur :**

  ctxInitWithRootViewController:

  ctxPopToViewController:animated:

Rediriger les interfaces d'exécution

Citrix Endpoint Management propose une invite à entrer le code PIN de sorte que vous n'avez pas besoin de le faire dans votre application.

Pour assurer la disponibilité de Citrix Endpoint Management, il est recommandé de ne pas rediriger ou remplacer les sélecteurs d'exécution Objective-C. En effet, Citrix Endpoint Management mélange les méthodes sous-jacentes de plusieurs sélecteurs de classes d'objet pour contrôler et/ou modifier le comportement d'exécution d'une application. Le tableau suivant dresse la liste des sélecteurs de classes Objective-C que Citrix Endpoint Management redirige :

Nom de la classe d'objet : NSURLProtectionSpace

- **Nom du sélecteur :** serverTrust

Nom de la classe d'objet : NSURLAuthenticationChallenge

- **Nom du sélecteur :** sender

Nom de la classe d'objet : NSURLConnection

- **Nom du sélecteur :**

  sendSynchronousRequest:returningResponse:error:

  initWithRequest:delegate:startImmediately:

  initWithRequest:delegate:

  connectionWithRequest:delegate:

Nom de la classe d'objet : NSURLConnectionDelegate

- **Nom du sélecteur :**

  connection:canAuthenticateAgainstProtectionSpace:

  connection:didReceiveAuthenticationChallenge:

  connection:willSendRequestForAuthenticationChallenge:

Nom de la classe d'objet : NSURLSessionConfiguration

- **Nom du sélecteur :**

  defaultSessionConfiguration

  ephemeralSessionConfiguration

Nom de la classe d'objet : ALAssetsLibrary

- **Nom du sélecteur :** authorizationStatus

Nom de la classe d'objet : AVAudioRecorder

- **Nom du sélecteur :**

  record

  prepareToRecord

  recordForDuration:

  recordAtTime:

  recordAtTime:ForDuration:

Nom de la classe d'objet : AVAudioSession

- **Nom du sélecteur :** recordPermission

Nom de la classe d'objet : AVCaptureDevice

- **Nom du sélecteur :**

  devices

  devicesWithMediaType:

Nom de la classe d'objet : AVAsset

- **Nom du sélecteur :** asseWithURL:

Nom de la classe d'objet : AVURLAsset

- **Nom du sélecteur :**

  initWithURL:options:

  URLAssetWithURL:options:

Nom de la classe d'objet : AVPlayerItem

- **Nom du sélecteur :**

  playerItemWithAsset:

  initWithURL:

  playerItemWithURL:

Nom de la classe d'objet : AVPlayer

- **Nom du sélecteur :**

  playerWithPlayerItem:

  initWithPlayerItem:

  initWithURL:

Nom de la classe d'objet : CLLocationManager

- **Nom du sélecteur :** startUpdatingLocation

Nom de la classe d'objet : UIScrollView

- **Nom du sélecteur :** setContentOffset:

Nom de la classe d'objet : MFMailComposeViewController

- **Nom du sélecteur :**

  canSendMail

  init

Nom de la classe d'objet : MFMessageComposeViewController

- **Nom du sélecteur :**

  canSendText

  initWithNibName:bundle:

Nom de la classe d'objet : NSFileManager

- **Nom du sélecteur :** URLForUbiquityContainerIdentifier:

Nom de la classe d'objet : NSUbiquitousKeyValueStore

- **Nom du sélecteur :** defaultStore

Nom de la classe d'objet : PHPhotoLibrary

- **Nom du sélecteur :** authorizationStatus

Nom de la classe d'objet : QLPreviewController

- **Nom du sélecteur :**

  setDataSource:

  canPreviewItem:

Nom de la classe d'objet : QLPreviewControllerDataSource

- **Nom du sélecteur :**

  numberOfPreviewItemsInPreviewController:

  previewController:previewItemAtIndex:

Nom de la classe d'objet : SLComposeViewController

- **Nom du sélecteur :** isAvailableForServiceType:

Nom de la classe d'objet : UIActivityViewController

- **Nom du sélecteur :**

  initWithActivityItems:applicationActivities:

  setExcludedActivityTypes:

Nom de la classe d'objet : UIApplication

- **Nom du sélecteur :**

  openURL:

  canOpenURL:

  setApplicationIconBadgeNumber:

Nom de la classe d'objet : UIDocument

- **Nom du sélecteur :**

  closeWithCompletionHandler:

  contentsForType:error:

Nom de la classe d'objet : UIDocumentInterActionController

- **Nom du sélecteur :**

  interactionControllerWithURL:

  setURL:

  setDelegate:

  presentPreviewAnimated:

  presentOpenInMenuFromBarButtonItem:animated:

  presentOpenInMenuFromRect:inView:animated:

  presentOptionsMenuFromBarButtonItem:animated:

  presentOptionsMenuFromRect:inView:animated:

Nom de la classe d'objet : UIDocumentMenuViewController

- **Nom du sélecteur :** initWithDocumentTypes:inMode:

Nom de la classe d'objet : UIImage

- **Nom du sélecteur :** imageNamed:

Nom de la classe d'objet : UIImagePickerController

- **Nom du sélecteur :** setSourceType:

  takePicture

startVideoCapture

isSourceTypeAvailable:

isCameraDeviceAvailable:

isFlashAvailableForCameraDevice:

availableCaptureModesForCameraDevice:

setMediaTypes

Nom de la classe d'objet : UINavigationController

- **Nom du sélecteur :**

  ctxInitWithRootViewController:

  ctxPushViewController:animated:

  ctxPopToViewController:animated:

Nom de la classe d'objet : UIPasteboard

- **Nom du sélecteur :**

  generalPasteboard

  pasteboardWithName:create:

  pasteboardWithUniqueName

  setValue:forPasteboardType:

  setData:forPasteboardType:

  setItems:

  addItems:

  dataForPasteboardType:

  valueForPasteboardType:

  numberOfItems

  pasteboardTypes

  pasteboardTypesForItemSet:

  containsPasteboardTypes:

  containsPasteboardTypes:inItemSet:

  items

  itemSetWithPasteboardTypes:

dataForPasteboardType:inItemSet:

valuesForPasteboardType:inItemSet:

string

strings

URL

URL

image

images

color

colors

Nom de la classe d'objet : UIPopoverController

- **Nom du sélecteur :** initWithContentViewController

Nom de la classe d'objet : UIPrintInteractionController

- **Nom du sélecteur :**

  isPrintingAvailable

  presentAnimated:completionHandler:

  presentFromBarButtonItem:animated:completionHandler:

  presentFromRect:inView:animated:completionHandler:

Nom de la classe d'objet : UIViewController

- **Nom du sélecteur :** presentViewController:animated:completion:

Nom de la classe d'objet : UIWebView

- **Nom du sélecteur :**

  loadRequest:

  setDelegate:

  UIWebViewDelegate

  webView:shouldStartLoadWithRequest:navigationType:

  webViewDidStartLoad:

  webViewDidFinishLoad:

  webView:didFailLoadWithError:

Nom de la classe d'objet : UIWindow

- **Nom du sélecteur :** makeKeyAndVisible

Nom de la classe d'objet : UIApplicationDelegate

- **Nom du sélecteur :**

  applicationDidFinishLaunching:

  application:didFinishLaunchingWithOptions:

  application:willFinishLaunchingWithOptions:

  applicationWillResignActive:

  applicationDidEnterBackground:

  applicationWillEnterBackground:

  applicationDidBecomeActive:

  applicationWillTerminate:

  application:openURL:sourceApplication:annotation:

  application:handleOpenURL:

  applicationProtectedDataWillBecomeUnavailable:

  applicationProtectedDataDidBecomeAvailable:

  application:performFetchWithCompletionHandler:

  application:handleEventsForBackgroundURLSession:completionHandler:

  application:didReceiveLocalNotification:

  application:didReceiveRemoteNotification:

  application:didReceiveRemoteNotification:fetchCompletionHandler:

  application:didRegisterForRemoteNotificationsWithDeviceToken:

  application:didFailToRegisterForRemoteNotificationsWithError:

  applicationSignificantTimeChange:

  application:shouldAllowExtensionPointIdentifier:

Nom de la classe d'objet : QLPreviewController

- **Nom du sélecteur :** allocWithZone:

Assurer la compatibilité du cryptage de données

L'une des fonctionnalités principales de MDX est que toutes les données conservées sont cryptées de façon transparente. Vous n'avez pas besoin de modifier votre application pour bénéficier de cette fonctionnalité, et en fait, vous ne pouvez pas directement l'éviter. L'administrateur Citrix Endpoint Management a la possibilité de désactiver le cryptage de manière sélective ou complètement, mais pas l'application.

Ceci est l'un des aspects plus complexes de MDX et il est important de comprendre les points suivants :

- Le cryptage de fichier est présent pour l'ensemble du code natif qui s'exécute dans les processus gérés.

  La mise en œuvre du cryptage des données de fichier concerne l'ensemble du code natif et pas seulement le code des applications utilisant les infrastructures Apple et l'exécution d'Apple Objective-C. Tout cryptage de données de fichier mis en œuvre dans et uniquement pour l'exécution d'Objective-C peut être facilement détourné.

- Certaines API d'infrastructure, telles que la classe AVPlayer, la classe UIWebView et QLPreviewController, sont en fait mises en œuvre par les processus de service iOS dans un contexte d'exécution différent du processus d'application gérée de l'utilisateur.

  Ces processus de service ne sont pas capables de déchiffrer les données de fichier cryptées par MDX, donc l'application gérée doit fournir le processus de service avec une copie temporaire non cryptée des données qui est ensuite supprimée par l'application au bout de 5 secondes. Il est important que vous connaissiez cette limitation si vous utilisez ces classes car nous perdons le contrôle de la contention des données fournies à ces classes à cause de la mise en œuvre par Apple de ces classes spécifiques.

- Le mappage de mémoire est problématique pour le cryptage Citrix Endpoint Management car il implique l'appel par l'application des interfaces d'appel du système E/S du fichier.

  Une fois qu'un fichier est mappé en mémoire, les requêtes E/S pour le fichier sont gérées en dehors du contexte de l'application utilisateur, ignorant le cryptage de Citrix Endpoint Management. Tous les appels mmap(2) POSIX par une application gérée sont mappés comme MAP_PRIVATE et MAP_ANON et ne sont pas associés à une description de fichier. Une tentative de lecture de toutes les données mappées lors de l'appel mmap échoue pour toutes les données si une description de fichier est spécifiée. En effet, toute pagination suivante des données par le système d'exploitation entraîne la lecture des données cryptées sans qu'elles soient décryptées par Citrix Endpoint Management. Cette technique a réussi dans toutes les applications qui ont été testées avec Citrix Endpoint Management car le volume de données qui est mappé en mémoire est faible sans récupération des pages mémoire dans l'application.

- Le cryptage ajoute une charge significative. Les développeurs devraient optimiser le traitement

E/S disque pour empêcher la dégradation des performances. Par exemple, si vous lisez et écrivez de manière répétée les mêmes informations, il se peut que vous souhaitiez mettre en place un cache au niveau de l'application.

- Citrix Endpoint Management crypte uniquement les instances de la libsqlite.dylib d'Apple. Si l'application établit un lien direct et/ou incorpore une version privée de la libsqlite.dylib, les instances de bases de données de cette bibliothèque privée ne seront pas cryptées par Citrix Endpoint Management.
- Les bases de données Apple SQLite sont cryptées par Citrix Endpoint Management à l'aide de la couche VFS de SQLite.

  Vous pouvez rencontrer un problème de performance. La taille du cache de base de données standard est 2 000 pages ou 8 Mo. Si votre base de données est volumineuse, il peut être nécessaire qu'un développeur spécifie une commande pragma SQLite pour augmenter la taille du cache de la base de données. Dans Objective-C Core Data Framework, la commande pragma SQLite peut être ajoutée en tant que dictionnaire d'options lors de l'ajout de l'objet Persistent Store à l'objet Persistent Store Controller.
- Le mode SQLite WAL n'est pas pris en charge car la bibliothèque est de nouveau liée aux interfaces E/S de fichier et en interne utilise le mappage de mémoire de façon extensive.
- NSURLCache DiskCache est mis en œuvre par iOS à l'aide d'une base de données SQLite. Citrix Endpoint Management désactive le cache disque associé, car cette base de données est référencée par des processus de service iOS non gérés.
- Le tableau suivant dresse la liste des modèles de nom de chemin d'accès de fichier exclus codés en dur :

.plist	Exclu en raison d'un accès par les processus système iOS en dehors du contexte de processus.
.app	Sous-chaîne d'ancienne génération dans le nom du bundle de l'application. Cette sous-chaîne est obsolète, car un chemin d'accès de bundle d'application explicite est désormais exclu.
.db	Un fichier avec ce suffixe n'est pas crypté si le fichier n'est pas une base de données sqlite.

/System/Library	Les chemins d'accès aux fichiers dans le répertoire sandbox du bundle d'application et les chemins d'accès aux fichiers en dehors du sandbox des données d'application ne peuvent pas être cryptés. Sur iOS, l'application installée est en lecture seule et se trouve dans un répertoire différent de celui des fichiers de données de l'application que l'application produit et stocke lorsqu'elle est exécutée.
Library/Preferences	Les fichiers sont accessibles directement par iOS. Normalement, seuls les fichiers .plist sont présents dans ce répertoire.
/com.apple.opengl/	Les fichiers sont accessibles directement par iOS.
csdk.db	Base de données Citrix SSLSDK sqlite ancienne version
/Library/csdk.sql	Base de données Citrix SSLSDK sqlite
CtxLog_	Préfixe du nom du fichier journal Citrix
CitrixMAM.config	Nom de fichier interne MDX
CitrixMAM.traceLog	Nom de fichier interne MDX ancienne version
CtxMAM.log	Nom de fichier interne MDX
data.999	Nom de fichier interne MDX
CTXWrapperPersistentData	Nom de fichier interne MDX
/Documents/CitrixLogs	Répertoire de journaux MDX
/Document/CitrixLogs.zip	Nom du répertoire de journaux MDX compressés
Tout fichier dans le chemin d'accès au répertoire du bundle d'application	Répertoire en lecture seule des fichiers de l'application

- Citrix Endpoint Management remplace une instance de la classe Citrix Endpoint Management SecureViewController privée par des instances de la classe d'objet Apple Objective-C QLPreview-Controller au moment de l'exécution. La classe Citrix Endpoint Management SecureViewController est dérivée de la classe d'objet Apple Objective-C UIWebView. La classe d'objet QLPreviewController prend en charge en mode natif quelques formats de fichier que la classe d'objet

UIWebView ne prend pas en charge en mode natif, tels que les types audio et pdf.

- Pour obtenir les meilleures performances, les requêtes E/S de fichier doivent être émises à des offsets de fichier qui sont un multiple de 4 096 octets et doivent être émises pour une longueur qui est également un multiple de 4 096 octets.
- L'indicateur de mode de fichier O_NONBLOCK n'est pas pris en charge par le cryptage Citrix Endpoint Management. Cet indicateur est supprimé de la liste des modes lors du traitement par Citrix Endpoint Management.

Entropie utilisateur

Une option de cryptage Citrix Endpoint Management nécessite que l'utilisateur entre un code PIN avant que la clé de cryptage puisse être générée. Cette option est appelée entropie utilisateur. Elle peut entraîner un problème particulier pour les applications.

Plus spécifiquement, aucun accès fichier ou base de données ne peut être effectué jusqu'à ce que l'utilisateur entre un code PIN. Si une telle opération E/S est présente à un emplacement qui s'exécute avant que l'interface utilisateur du code PIN puisse être affichée, elle échouera toujours.

Pour vous assurer que ce problème ne concerne pas votre application, testez-la avec l'entropie utilisateur activée. La propriété du client Citrix Endpoint Management, Encrypt secrets using Passcode, ajoute l'entropie utilisateur. Vous pouvez configurer cette propriété de client, qui est désactivée par défaut, dans la console Citrix Endpoint Management, sous **Configurer > Paramètres > Plus > Propriétés du client**.

Compatibilité de la contention des données

- Les contrôleurs d'affichage à distance n'ont pas de contention de sécurité (par exemple, le cryptage de données, le blocage de la stratégie copier/couper/coller, et ainsi de suite) car un contrôleur d'affichage à distance s'exécute dans un contexte de processus différent de l'application gérée par MDX.
- L'action Copier est la seule action prise en charge depuis UIResponder. Les autres actions, telles que Couper et Supprimer, ne sont pas prises en charge.
- Airdrop est intercepté uniquement au niveau de l'interface utilisateur, et non pas à un niveau inférieur.
- MFI et Bluetooth ne sont pas interceptés.

Prise en charge des fichiers d'icônes

L'encapsulation MDX requiert la présence d'au moins une icône qui peut être utilisée comme icône springboard ou icône d'application. Les développeurs d'applications peuvent ajouter leurs icônes au

catalogue de logiciels, ou utiliser les clés CFBundleIcons ou CFBundleIconFiles du fichier Info.plist.

Le MDX Toolkit choisira la première clé dans la liste des emplacements plist connus du fichier Info.plist :

- CFBundleIcons
- CFBundlePrimaryIcon
- CFBundleIconFiles
- UINewsstandIcon
- CFBundleDocumentTypes

Si aucune de ces clés n'est trouvée dans Info.plist, le MDX Toolkit identifiera l'une des icônes suivantes dans le dossier racine du bundle d'application :

- Icon.png
- Icon-60@2x.png
- Icon-72.png
- Icon-76.png

Réseau et micro VPN

MDX gère actuellement uniquement les appels réseau directement émis par une application. Certaines requêtes DNS sont émises directement par l'infrastructure Apple et ne sont donc pas gérées par MDX.

Les administrateurs disposent de plusieurs options de stratégie Citrix Endpoint Management pour le réseau.

La stratégie Accès réseau empêche, permet ou redirige l'activité réseau de l'application :

> Important :
>
> La version 18.12.0 de MDX Toolkit inclut de nouvelles stratégies combinant ou remplaçant des stratégies plus anciennes.
> La stratégie Accès réseau combine Accès réseau, Mode VPN préféré et Autoriser le basculement vers le mode VPN. La stratégie Liste d'exclusion remplace Liste d'exclusion de split tunneling. La stratégie Session micro VPN requise remplace Session en ligne requise. Pour de plus amples informations, consultez la section Nouveautés dans le MDX Toolkit 18.12.0.
>
> Tunnel - SSO Web est le nom de Secure Browse dans les paramètres. Le comportement est le même.

Les options sont les suivantes :

- **Utiliser les paramètres précédents :** utilise par défaut les valeurs que vous aviez définies dans les stratégies précédentes. Si vous modifiez cette option, vous ne devez pas revenir à **Utiliser**

paramètres précédents. Notez également que les modifications apportées aux nouvelles stratégies ne prennent effet que lorsque l'utilisateur met à niveau l'application vers 18.12.0 ou version ultérieure.

- **Bloqué :** les API de mise en réseau utilisées par votre application échoueront. Conformément à la recommandation précédente, vous devriez traiter un tel échec de façon appropriée.
- **Sans restriction :** tous les appels réseau ont un accès direct et ne sont pas tunnélisés.
- **Tunnel VPN complet :** tout le trafic provenant de l'application gérée est tunnélisé via Citrix Gateway.
- **Tunnel - SSO Web :** l'URL HTTP/HTTPS est réécrite. Cette option permet uniquement le tunneling du trafic HTTP et HTTPS. Un avantage important de **Tunnel - SSO Web** est l'authentification unique (SSO) pour le trafic HTTP et HTTPS ainsi que l'authentification PKINIT. Sur Android, cette option a une charge de configuration faible et il s'agit donc de l'option préférée pour les opérations de type navigation Web.
- **Tunnel - VPN complet et SSO Web :** permet de basculer automatiquement entre les modes VPN selon les besoins. Si une demande réseau qui a échoué en raison d'une demande d'authentification qui ne peut pas être traitée dans un mode VPN spécifique est de nouveau tentée dans un autre mode.

Limitations :

- WkWebView n'est pas pris en charge.
- Les utilisateurs ne peuvent pas lire de vidéos hébergées sur des sites Web internes dans les applications iOS MDX encapsulées car les vidéos sont lues dans un processus de lecteur multimédia sur l'appareil que MDX n'intercepte pas.
- Le téléchargement en arrière-plan NSURLSession (NSURLSessionConfiguration backgroundSessionConfigurationWithIdentifier) n'est pas pris en charge.
- Nous bloquons le trafic UDP si la stratégie Accès réseau est définie sur **Bloqué**. Nous ne tunnélisons pas le trafic UDP si la stratégie Accès réseau est définie sur **Tunnel VPN complet**.
- Les applications encapsulées par MDX ne peuvent pas ajouter un serveur socket qui écoute les connexions entrantes. Toutefois, les applications encapsulées par MDX peuvent utiliser un socket client pour se connecter à un serveur.

Prise en charge des bibliothèques tierces

Certaines infrastructures d'applications ont des problèmes de compatibilité avec Citrix Endpoint Management.

- Les applications développées avec l'environnement de développement multi-plateformes Xamarin sont prises en charge. Citrix ne déclare pas officiellement la prise en charge d'autres environnements de développement multi-plateformes en raison d'exemples insuffisants d'utilisation et de test.

- SQLCipher ne fonctionne pas avec le cryptage car il utilise le mappage de mémoire. Une solution consiste à ne pas utiliser SQLCipher. Une autre solution consiste à exclure le fichier de base de données du cryptage à l'aide d'une stratégie d'exclusion de cryptage. Un administrateur Citrix Endpoint Management doit configurer la stratégie dans la console Citrix Endpoint Management.
- Les bibliothèques d'applications et tierces qui lient directement aux bibliothèques OpenSSL libcrypto.a et libssl.a peuvent entraîner une erreur de lien en raison d'un manque de symboles et des erreurs de lien en raison de définitions de symbole multiples.
- Les applications nécessitant une prise en charge du service de notification push d'Apple devront suivre les étapes spécifiques requises par Apple.
- Citrix Endpoint Management définit de manière explicite la version de la base de données SQLite sur 1 pour désactiver la prise en charge des fichiers WAL (Write Ahead Logging) et des fichiers mappés en mémoire dans les bases de données SQLite. Toute tentative d'accéder directement aux interfaces SQLite version 2 ou 3 échoue.

API pour iOS

February 19, 2019

L'API XenMobile pour iOS est basée sur Objective-C. Cet article décrit les API Citrix Endpoint Management par fonctionnalité et fournit les définitions de l'API.

Gestion des applications

- isAppManaged

Interaction avec Secure Hub

- isMDXAccessManagerInstalled
- logonMdxWithFlag
- isAppLaunchedByWorxHome

Stratégies MDX

- getValueOfPolicy

Coffre partagé

- getVaultDataFromVault
- saveVaultData
- updateAndSynchronizeVaultItem
- updateAndSynchronizeVaultItems
- deleteVault
- deleteVaultWithError

Données utilisateur

- managedUserInformation

Classe MdxManager

Méthodes

- **getValueOfPolicy**

  ```
  +(NSString*)getValueOfPolicy:(NSString*)policyName error:(NSError **)
  error;
  ```

  Pour les applications gérées, renvoie la valeur de stratégie définie par les administrateurs Citrix Endpoint Management. Pour les applications Premium non gérées, renvoie la valeur de stratégie définie dans Applications/Citrix/MDXToolkit/data/MDXSDK/default_policies.xml. Pour les applications General non gérées, renvoie **nul**.

  Paramètres :

  policyName – nom de la stratégie à rechercher dans defaultpolicies.xml.

  Exemple :

  ```
  +(NSString*)getValueOfPolicy:(NSString*)DisableCamera error:(NSError
  **)error;
  ```

- **isMDXAccessManagerInstalled**

  ```
  +(BOOL)isMDXAccessManagerInstalled: (NSError **)error;
  ```

  Vérifie si Secure Hub est installé, ce qui signifie que le contrôle MDX de l'application est activé, même si l'application n'est pas gérée. Renvoie **true** si Secure Hub est installé.

- **isAppManaged**

  ```
  +(BOOL)isAppManaged;
  ```

  Vérifie si l'application est actuellement gérée par MDX, ce qui signifie que le bundle de stratégie MDX est incorporé dans l'application en tant que fichier XML. L'infrastructure principale de Citrix Endpoint Management (coffres de clés) est interrogée pour obtenir les clés partielles de cryptage des données (secrets) que MDX utilisera pour crypter les données de base de données de l'application (iOS 9 et versions ultérieures). Renvoie **true** si l'application est gérée.

- **logonMdxWithFlag**

  ```
  +(BOOL)logonMdxWithFlag:(BOOL)force error:(NSError**)error;
  ```

  Initie une requête d'ouverture de session MDX avec Secure Hub.

- **isAppLaunchedByWorxHome**

  ```
  +(BOOL)isAppLaunchedByWorxHome;
  ```

  Vérifie si une requête d'URL inter-application provient de Secure Hub ou d'une autre application de l'appareil, ce qui est nécessaire si une application doit être informée des communications de contrôle de MDX. Sur iOS, les applications peuvent être inscrites pour certains modèles d'URL. Un modèle d'URL est la première partie d'une adresse URL, celle qui se trouve avant les deux-points. Si une URL commence par `http://`, le modèle est http.

  Les applications MDX communiquent avec Secure Hub à l'aide de modèles d'URL personnalisés. Par exemple, pour traiter les adresses URL mailto: à partir d'autres applications, Secure Mail requiert le modèle d'URL ctxmail. Pour traiter les adresses URL http ou https à partir d'autres applications, Secure Web requiert le modèle d'URL ctxmobilebrowser ou ctxmobilebrowsers, respectivement. Pour de plus amples informations sur la stratégie MDX Modèles d'URL d'application et la stratégie URL autorisées, consultez la section Stratégies MDX pour applications iOS.

  Renvoie des résultats corrects lors d'une interrogation à n'importe quel emplacement, à tout moment, pendant ou après les appels d'événement de délégation UIApplication suivants :

  - Lorsque l'application est chargée à partir du springboard ou d'un appel **openURL** :

    ```
    application:willFinishLaunchingWithOptions:

    application:didFinishLaunchingWithOptions:

    applicationDidFinishLaunching:
    ```

  - Lorsque l'application est activée ou réactivée par les utilisateurs à partir du springboard de l'appareil

    ```
    applicationDidBecomeActive:
    ```

  > **Important :**
  >
  > vous ne devez pas effectuer des requêtes durant applicationWillEnterForeground:.

  - Lorsque l'application est activée ou réactivée par un appel **openURL** :

    ```
    application:openURL:sourceApplication:annotation:

    application:handleOpenURL:
    ```

- **managedUserInformation**

  ```
  extern __attribute__((visibility ("default")))NSString *const kXenMobileUsername
  ; +(NSDictionary*)managedUserInformation;
  ```


Renvoie une chaîne contenant le nom d'utilisateur de l'utilisateur inscrit exécutant une application gérée par MDX, quel que soit l'état de connexion de l'utilisateur. Renvoie une chaîne vide si l'utilisateur n'est pas inscrit, l'application n'est pas gérée, ou l'application n'est pas encapsulée.

Classe XenMobileSharedKeychainVault

Méthodes

- **initWithVaultName**

  ```
  - (instancetype)initWithVaultName:(NSString*)vaultName accessGroup:(
  NSString*)accessGroup;
  ```

  Initialise un coffre partagé Citrix Endpoint Management.

  Utilisez l'API de coffre sécurisé pour partager le contenu géré entre les applications qui ont le même groupe de trousseaux d'accès. Par exemple, vous pouvez partager des certificats utilisateur par le biais d'une application inscrite de façon à ce que les applications puissent obtenir un certificat depuis le coffre sécurisé plutôt que depuis Secure Hub.

  Paramètres :

  vaultName – nom du coffre partagé Citrix Endpoint Management.

  accessGroup – nom du groupe de trousseaux d'accès. Il peut s'agir du groupe d'accès MDX par défaut, appelé TEAMID_A.appOriginalBundleID, ou d'un groupe de trousseaux d'accès que vous utiliserez pour partager les données entre les applications.

- **Propriétés du type de données du coffre**

  ```
  1    @property(nonatomic,readonly) BOOL exists;
  2
  3    @property(nonatomic,readonly) BOOL isAccessible;
  4
  5    @property(nonatomic,strong) NSMutableDictionary* vaultData
  ```

  Après avoir initialisé un coffre, les propriétés du type de données du coffre suivantes sont renvoyées :

  exists – indique si le coffre avec la valeur spécifiée *vaultName* a été trouvé.

  isAccessible – indique si le coffre se trouve dans le groupe *accessGroup* et est accessible.

  vaultData – contenu du coffre partagé. Lors de la première initialisation du coffre, *vaultData* est un dictionnaire de valeur nulle.

- **getVaultDataFromVault**


```
+ (NSDictionary*)getVaultDataFromVault:(NSString*)vaultName accessGroup
:(NSString*)accessGroup error:(NSError *__autoreleasing *)error;
```

Lit les données depuis le coffre partagé Citrix Endpoint Management. Il existe trois manières de lire les données du coffre, comme suit :

- Utilisez directement getVaultDataFromVault:accessGroup:error.
- Créez l'instance XenMobileSharedKeychainVault, puis lisez la propriété vaultData.
- Créez l'instance XenMobileSharedKeychainVault, puis rechargez les données du coffre à l'aide de l'erreur -(BOOL) loadDataWithError:(NSError *__autoreleasing *)error; et en lisant la propriété vaultData.

Pour un exemple de code, consultez la section Exemple de coffre partagé de cet article.

Paramètres :

vaultName – nom du coffre partagé Citrix Endpoint Management.

accessGroup – nom du groupe de trousseaux d'accès. Il peut s'agir du groupe d'accès MDX par défaut, appelé TEAMID_A.appOriginalBundleID, ou d'un groupe de trousseaux d'accès que vous utiliserez pour partager les données entre les applications.

- **saveVaultData**

```
+ (BOOL)saveVaultData:(NSDictionary*)vaultData toVault:(NSString*)
vaultName accessGroup:(NSString*)accessGroup error:(NSError *__autoreleasing
 *)error;
```

Enregistre les données dans le coffre partagé Citrix Endpoint Management. Il existe trois manières d'enregistrer les données du coffre, comme suit :

- Utilisez directement **saveVaultData:toVault:accessGroup:error:**.
- Utilisez **updateAndSynchronizeVaultItem:** ou **updateAndSynchronizeVaultItems** (décrit ci-après dans ce tableau).
- Utilisez- **(BOOL)synchronizeWithError:(NSError *__autoreleasing *)error;** en créant l'instance **XenMobileSharedKeychainVault**, en chargeant les données du coffre, en modifiant les données du coffre, puis en synchronisant les données.

Pour un exemple de code, consultez la section Exemple de coffre partagé de cet article.

Paramètres :

vaultData – données à enregistrer dans le coffre partagé Citrix Endpoint Management. Les données stockées dans le coffre partagé constituent un dictionnaire de paires clé/valeur, telles que @{@"username":@"andreo"}.

vaultName – nom du coffre partagé Citrix Endpoint Management.

accessGroup – nom du groupe de trousseaux d'accès. Il peut s'agir du groupe d'accès MDX par défaut, appelé TEAMID_A.appOriginalBundleID, ou d'un groupe de trousseaux d'accès que vous utiliserez pour partager les données entre les applications.

- **updateAndSynchronizeVaultItem**

  updateAndSynchronizeVaultItems

  ```
  - (BOOL)updateAndSynchronizeVaultItem:(NSString*)vaultItem withValue:(
  id)itemValue error:(NSError *__autoreleasing *)error;
  ```

  ```
  - (BOOL)updateAndSynchronizeVaultItems:(NSDictionary*)vaultItems error
  :(NSError *__autoreleasing *)error;
  ```

  Met à jour les données dans le coffre partagé Citrix Endpoint Management. Pour utiliser cette méthode, créez l'instance **XenMobileSharedKeychainVault**, puis synchronisez-la en ajoutant ou en mettant à jour des éléments de données de coffre. Par exemple, si l'entrée du coffre comporte {a:123, b:234, c:305} et que nous utilisons cette API avec les données de mise à jour {c:345, d:456}, cette API met les données du coffre à jour vers {a:123, b:234, c:345, d:456}. Pour un exemple de code, consultez la section Exemple de coffre partagé de cet article.

  Consultez la section **saveVaultData** ci-dessus pour en savoir plus sur les deux autres méthodes d'enregistrement des données du coffre.

  Paramètres :

  vaultItem – paire clé/valeur unique au format `@{ @";username::@"andreo"}` .

  vaultItems – liste des paires clé/valeur.

- **deleteVault**

  ```
  + (BOOL)deleteVault:(NSString*)vaultName accessGroup:(NSString*)accessGroup
   error:(NSError *__autoreleasing *)error;
  ```

  Supprime le coffre partagé spécifié.

  Paramètres :

  vaultName – nom du coffre partagé Citrix Endpoint Management.

  accessGroup – nom du groupe de trousseaux d'accès utilisé par le coffre que vous souhaitez supprimer.

- deleteVaultWithError

  ```
  -(BOOL)deleteVaultWithError:(NSError *__autoreleasing *)error;
  ```

  Supprime le coffre partagé renvoyé par l'instance **XenMobileSharedKeychainVault**. Vous devez libérer l'objet après sa suppression par **deleteVaultWithError**.

Exemple de coffre partagé

```
#import ”XenMobileSharedKeychainVault.h”

@interface ClassA ()
...
@property(nonatomic,strong) XenMobileSharedKeychainVault*
    XenMobileSharedKeychainVault;
...
@end

@implementation ClassA
...
@synthesize XenMobileSharedKeychainVault =
    _XenMobileSharedKeychainVault;


...
#ifdef USE_CLASS_INSTANCE_METHODS
-(XenMobileSharedKeychainVault*)XenMobileSharedKeychainVault
{

if(_XenMobileSharedKeychainVault==nil) {

_XenMobileSharedKeychainVault = [[XenMobileSharedKeychainVault alloc]
      initWithVaultName:<VAULT_NAME>
      accessGroup:kXenMobileKeychainAccessGroup];
 }

return _XenMobileSharedKeychainVault;
 }

#endif

-(void)read
{

NSError* error=nil;
#ifdef USE_CLASS_INSTANCE_METHODS
NSDictionary* vaultDictionary = nil;
if([self.XenMobileSharedKeychainVault loadDataWithError:&error]) {

vaultDictionary = [self.XenMobileSharedKeychainVault vaultData];
 }
```



```
41
42 #else
43 NSDictionary* vaultDictionary = [XenMobileSharedKeychainVault
44       getVaultDataFromVault:<VAULT_NAME>
45       accessGroup:kXenMobileKeychainAccessGroup error:&error];
46 #endif
47
48  }
49
50
51 -(void)save
52 {
53
54 NSError* error=nil;
55 /// check error handling here...
56
57 NSDictionary* dictToSave = @{
58  <VAULT_DATA_DICTIONARY_OBJECTS> }
59  ;
60 #ifdef USE_CLASS_INSTANCE_METHODS
61 #ifdef USE_CLASS_INSTANCE_METHODS_TO_UPDATE
62 BOOL result = [self.XenMobileSharedKeychainVault
63       updateAndSynchronizeVaultItems:dictToSave error:&error];
64 #else
65 self.XenMobileSharedKeychainVault.vaultData = [NSMutableDictionary
66       dictionaryWithDictionary:dictToSave];
67 BOOL result = [self.XenMobileSharedKeychainVault synchronizeWithError:&
      error];
68 #endif
69 #else
70 BOOL result = [XenMobileSharedKeychainVault
71       saveVaultData:dictToSave toVault:<VAULT_NAME>
72       accessGroup:kXenMobileKeychainAccessGroup error:&error];
73 #endif
74
75  }
76
77
78 -(void)delete
79 {
80
81 NSError* error=nil;
82 #ifdef USE_CLASS_INSTANCE_METHODS
83 BOOL result = [self.XenMobileSharedKeychainVault deleteVaultWithError:&
      error];
```



```
#else
BOOL result = [XenMobileSharedKeychainVault deleteVault:<VAULT_NAME>
      accessGroup:kXenMobileKeychainAccessGroup error:&error];
#endif

 }


...

@end
```

Valeurs par défaut des stratégies et stratégies personnalisées

February 19, 2019

Cet article explique comment vous pouvez utiliser les stratégies dans vos applications ISV encapsulées.

Modifier les valeurs par défaut de stratégie pour les applications Premium non gérées

Le SDK de l'application MDX comprend les fichiers de stratégie suivants qui spécifient les valeurs par défaut de stratégie pour les applications Premium non gérées uniquement.

- Android : Applications/Citrix/MDXToolkit/data/MDXSDK_Android/ default_sdk_policies.xml
- iOS : Applications/Citrix/MDXToolkit/data/MDXSDK/default_policies.xml

Toutes les stratégies dans ces fichiers sont désactivées. Les stratégies qui ne se trouvent pas dans le fichier sont ignorées pour les applications Premium non gérées.

Vous pouvez modifier les paramètres par défaut comme suit.

1. Effectuez une copie de sauvegarde des fichiers de stratégie par défaut que vous voulez modifier, dans le cas où vous en auriez besoin plus tard.
2. Pour modifier une stratégie par défaut pour des applications ISV, utilisez les valeurs de stratégie spécifiées dans la documentation du MDX Toolkit, dans Stratégies MDX pour applications Android et Stratégies MDX pour applications iOS.
3. Vous devez inclure le fichier de stratégie par défaut avec les ressources de votre application lorsque vous générez l'application Premium.

Créer des stratégies personnalisées

Les fichiers de stratégie dans le MDX Toolkit fournissent des définitions complètes des stratégies, y compris le nom de la stratégie et le texte d'aide affiché dans la console Citrix Endpoint Management. Lorsque vous encapsulez une application, ces stratégies sont inclues avec le fichier .mdx généré. Vous pouvez ajouter des stratégies personnalisées à ces fichiers, qui sont situés dans le dossier d'installation MDX Toolkit dans Applications/Citrix/MDXToolkit/data.

1. Effectuez une copie de sauvegarde des fichiers de stratégie que vous voulez modifier, dans le cas où vous en auriez besoin plus tard.
2. Pour ajouter des stratégies aux fichiers de stratégie XML, utilisez les formats fournis dans « Formats de stratégie » ci-après.
3. Lorsque vous encapsulez l'application, spécifiez l'emplacement de votre fichier de stratégie XML modifié, y compris l'option -policyxml à l'aide de la ligne de commande d'encapsulation :

   -policyxml /Applications/Citrix/MDXToolkit/data/policy_metadata.xml

   Pour de plus amples informations sur l'utilisation de la ligne de commande pour encapsuler des applications ISV, consultez les sections Encapsulation d'applications d'entreprise iOS à l'aide de la ligne de commande et Encapsulation d'applications Android ISV à l'aide de la ligne de commande.
4. Pour vérifier les noms, les descriptions et les valeurs de stratégie dans la console Citrix Endpoint Management, chargez l'application sur Endpoint Management.

Recommandations pour ajouter des stratégies

- Modifiez uniquement les éléments affichés en gras.
- La valeur de l'élément PolicyName correspond au nom appelé depuis l'application.
- La valeur de l'élément PolicyCategory correspond au nom de la catégorie dans laquelle la stratégie sera répertoriée dans la console Citrix Endpoint Management. Pour vérifier les noms des catégories, veuillez consulter les valeurs CategoryId dans la section **<Category>** des fichiers de stratégie MDX.
- La valeur de l'élément PolicyDefault est le paramètre par défaut de votre stratégie.
- L'élément POLICY_ID dans **<Title res_id="POLICY_ID">** correspond à un ID unique utilisé pour la stratégie. L'ID doit commencer par une lettre, ne peut pas contenir d'espaces et doit contenir uniquement des lettres, des chiffres ou un trait de soulignement.
- La valeur de l'élément Title est le nom de la stratégie qui s'affiche dans la console Citrix Endpoint Management.
- L'élément POLICY_DESC_ID dans **<Description res_id="POLICY_DESC_ID">** correspond à un ID unique utilisé pour la description de la stratégie. L'ID doit commencer par une lettre, ne

peut pas contenir d'espaces et doit contenir uniquement des lettres, des chiffres ou un trait de soulignement.

- La valeur de l'élément Description correspond à la description de la stratégie qui s'affiche dans la console Citrix Endpoint Management.

Chaîne

```
<Policy>
   <PolicyName>PolicyName</PolicyName>
   <PolicyType>string</PolicyType>
   <PolicyCategory>Category_ID</PolicyCategory>
   <PolicyDefault>Value</PolicyDefault>
   <PolicyStrings>
      <Title res_id=”POLICY_ID” >Sample String Policy</Title>
      <Description res_id=”POLICY_DESC_ID”>
          Please enter the policy value.
      </Description>
   </PolicyStrings>
</Policy>
```

Booléen

```
<Policy>
   <PolicyName>PolicyName</PolicyName>
   <PolicyType>string</PolicyType>
   <PolicyCategory>Category_ID</PolicyCategory>
   <PolicyDefault>false</PolicyDefault>
   <PolicyStrings>
        <Title res_id=”POLICY_ID” >Sample Boolean Policy</Title>
      <BooleanTrueLabel res_id=”POLICY_ON”>On</BooleanTrueLabel>
      <BooleanFalseLabel res_id=”POLICY_OFF”>Off</BooleanFalseLabel>
      <Description res_id=”POLICY_DESC_ID”>
          If On, the app does something.
          If Off, the app does something else.

            Default value is Off.
      </Description>
   </PolicyStrings>
</Policy>
```


Enum

```
<Policy>
   <PolicyName>PolicyName</PolicyName>
   <PolicyType>enum</PolicyType>
        <PolicyEnumValues>
         <PolicyEnumValue>
               <PolicyEnumValueId>Value1</PolicyEnumValueId>
            <PolicyEnumValueString res_id=”ID_1”>Yes</
               PolicyEnumValueString>
            </PolicyEnumValue>
            <PolicyEnumValue>
               <PolicyEnumValueId>Value2</PolicyEnumValueId>
            <PolicyEnumValueString res_id=”ID_2”>No</
               PolicyEnumValueString>
         </PolicyEnumValue>
         <PolicyEnumValue>
            <PolicyEnumValueId>Value3</PolicyEnumValueId>
            <PolicyEnumValueString res_id=”ID_3”>Maybe</
               PolicyEnumValueString>
            </PolicyEnumValue>
        </PolicyEnumValues>
   <PolicyCategory>Category_ID</PolicyCategory>
   <PolicyDefault>Value1</PolicyDefault>
   <PolicyStrings>
        <Title res_id=”POLICY_ID” >Sample Enum Policy</Title>
      <Description res_id=”POLICY_DESC_ID”>
          Sample policy description.

             Default value is Yes.
      </Description>
   </PolicyStrings>
</Policy>
```

Résolution des problèmes

February 19, 2019

Pour résoudre les problèmes qui peuvent se produire lorsque vos applications sont exécutées dans un environnement Citrix Endpoint Management, vous devez d’abord déterminer si le problème se produit lorsque l’application est non encapsulée ou encapsulée. Si ce problème se produit lorsque l’application est non encapsulée, il est spécifique à l’application.

Si ce problème se produit lorsque l'application est encapsulée

- Consultez les problèmes connus. Voir Problèmes connus avec le SDK de l'application MDX et Problèmes connus.
- Vérifiez que la version du SDK de l'application MDX que vous utilisez correspond à la version de l'outil MDX Toolkit que vous utilisez pour encapsuler l'application. Pour iOS, assurez-vous que la ligne correcte est ajoutée à votre projet. Cela permet de s'assurer que l'infrastructure a été ajoutée et que les API fonctionnent. Pour Android, assurez-vous que les bibliothèques pour tous les appareils que vous utilisez ont été ajoutées au projet et que worxsdk.aar a été ajouté aux dépendances du projet. Si vous rencontrez des problèmes supplémentaires avec l'intégration du SDK dans votre projet, veuillez contacter Citrix Ready ou le support Citrix.
- Déterminez si le problème est une erreur d'encapsulation d'application. Consultez les journaux de l'outil MDX Toolkit dans Applications/Citrix/MDXToolkit/logs.

  Les fichiers journaux contiennent les informations et la progression de l'encapsulation. Vérifiez dans ces journaux les messages d'erreur et les avertissements. Pour de plus amples informations, consultez les sections Identification des erreurs d'encapsulation des applications iOS et Identification des erreurs d'encapsulation des applications Android.

  Collectez les journaux à partir de Secure Hub : Dans Secure Hub, touchez Support, touchez Besoin d'aide?, puis touchez le nom de l'application. Secure Mail ouvre une fenêtre de nouveau message auquel est joint un journal de l'application sélectionnée. Vous pouvez ajouter plus d'informations sur le problème dans le message. Veuillez répertorier les étapes requises pour reproduire le problème et inclure le journal de la tentative d'encapsulation ainsi que des informations supplémentaires sur le problème.

  D'autres journaux de l'appareil peuvent être utiles. Voir Collecte des journaux système sur les appareils iOS et Collecte des journaux d'application à partir de la ligne de commande.

Si vous ne pouvez pas installer une application encapsulée sur un appareil

- Vérifiez que vous utilisez un keystore valide pour les applications Android ou un profil de provisioning et une paire de certificats valides pour iOS. Tenez compte des considérations spéciales pour les profils de provisioning et les certificats décrites dans Encapsulation d'applications mobiles iOS.

En cas de problème avec votre clé de certificat Apple

- Demandez une nouvelle émission du certificat depuis l'application Trousseau d'accès d'Apple. Cette opération génère une nouvelle clé privée. Vous devez ensuite télécharger le certificat et le profil de provisioning à partir du site Web des développeurs Apple.

Locations
Corporate Headquarters | 851 Cypress Creek Road Fort Lauderdale, FL 33309, United States
Silicon Valley | 4988 Great America Parkway Santa Clara, CA 95054, United States