

COMMENT

tirer profit de la gestion
des secrets de PowerShell
avec Password Hub et Devolutions Server

La migration vers un nouveau système de gestion de mots de passe peut être ardue, mais le module PowerShell de gestion des secrets pour Password Hub et Devolutions Server simplifie tout. Les autres gestionnaires de mots de passe qui offrent des modules semblables en bénéficient : le module standardisé permet de remplacer aisément les systèmes par une solution de gestion de mots de passe sécurisée et intégrée.

Installation des modules PowerShell Secret de Devolutions Server et Hub

La première chose à faire avant d'utiliser les articles du Devolutions Secret Management, est d'installer Microsoft Secret Management, de même que les modules de Devolutions Hub et Devolutions Secret Store depuis la galerie PowerShell.

1. Lancez PowerShell 7 (le choix en tant d'administrateur est facultatif). PowerShell 7.2.1 lancé dans le terminal Windows est utilisé dans cet article.
2. Installez la dernière version des modules de [Microsoft.PowerShell.SecretManagement](#), [SecretManagement.DevolutionsHub](#), et [SecretManagement.DevolutionsServer](#) (version 1.1.2, v0.3.0 et v0.2 au moment de la rédaction de cet article). Choisissez soit **[Y] Yes** ou **[A] Yes to All** (le dernier signifie que vous ne serez pas averti des prochaines installations de module à partir de ce répertoire) lorsque vous êtes invité à l'installer depuis la galerie PowerShell.

```
Install-Module -Name Microsoft.PowerShell.SecretManagement,  
SecretManagement.DevolutionsHub, SecretManagement.DevolutionsServer
```

```
PS C:\Users\testaccount1> Install-Module -Name Microsoft.PowerShell.SecretManagement, SecretManagement.DevolutionsHub, S  
ecretManagement.DevolutionsServer  
  
Untrusted repository  
You are installing the modules from an untrusted repository. If you trust this repository, change its  
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from  
'PSGallery'?  
[V] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A  
PS C:\Users\testaccount1> |
```

Installation des modules requis



Si vous n'avez pas prévu d'utiliser Devolutions Hub ou Devolutions Server, vous pouvez ignorer cette étape.

3. Importez les modules PowerShell grâce au module d'importation (Import-Module), et assurez-vous que les modules sont prêts à l'utilisation avec Get-Module.

```
Import-Module Microsoft.PowerShell.SecretManagement,  
SecretManagement.DevolutionsHub, SecretManagement.DevolutionsServer  
Get-Module Microsoft.PowerShell.SecretManagement, SecretManagement.DevolutionsHub,  
SecretManagement.DevolutionsServer
```

```
PS C:\Users\testaccount1> Import-Module Microsoft.PowerShell.SecretManagement, SecretManagement.DevolutionsHub, SecretManagement.DevolutionsServer  
PS C:\Users\testaccount1> Get-Module Microsoft.PowerShell.SecretManagement, SecretManagement.DevolutionsHub, SecretManagement.DevolutionsServer
```

ModuleType	Version	PreRelease	Name	ExportedCommands
Binary	1.1.2		Microsoft.PowerShell.SecretManagement	{Get-Secret, Get-SecretInfo, Get-SecretVault, Register-DevolutionsHubSecretVault, Register-DevolutionsServerSecretVault}
Script	0.3.0		SecretManagement.DevolutionsHub	Register-DevolutionsHubSecretVault
Script	0.2		SecretManagement.DevolutionsServer	Register-DevolutionsServerSecretVault

Vérification de l'installation adéquate des modules

Configuration des modules Secret Store

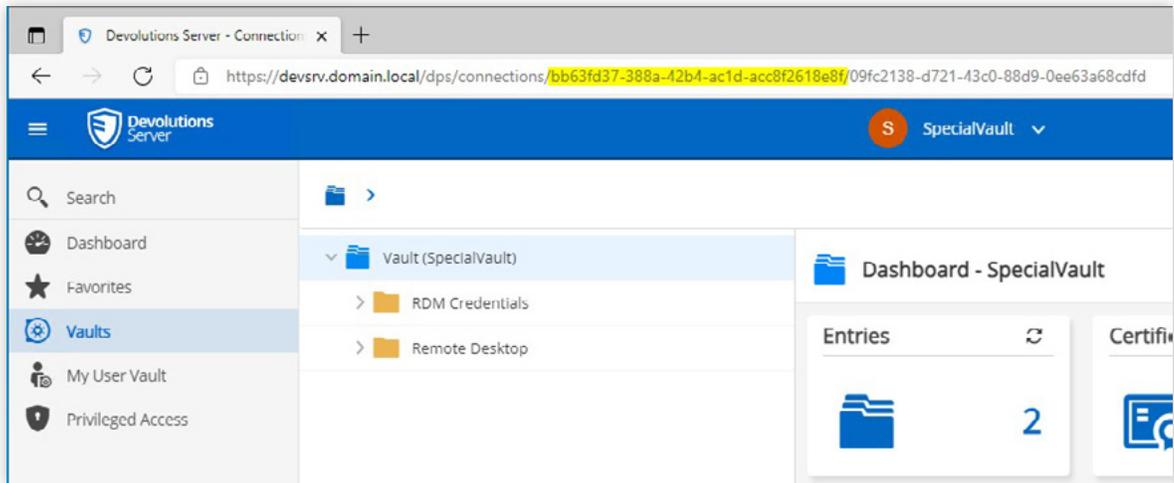
Vous devez tout d'abord configurer les modules avant leur installation pour qu'ils se connectent à leurs magasins de données correspondants.

Enregistrez les coffres secrets nécessaires dans le module Microsoft Secret Management comme indiqué ci-dessous pour Devolutions Server ou Devolutions Hub.

Cette action ne doit être effectuée qu'une seule fois, car un coffre enregistré est conservé dans les futures sessions PowerShell après l'importation des modules.

Connexion à Devolutions Server

1. La récupération de l'ID du coffre nécessite de repérer celui-ci dans l'interface Web de Devolutions Server en allant dans **Devolutions Server** → **Coffres** et en localisant l'ID dans l'URL comme indiqué ci-dessous.



Récupération de l'ID du coffre.

2. Ensuite, exécutez le code suivant pour enregistrer un coffre secret pour Devolutions Server, en indiquant l'URL du serveur ainsi que les identifiants ayant accès à l'ID du coffre fourni.

```

$Credentials = Get-Credential

Register-SecretVault -Name 'DevolutionsServer' -ModuleName 'SecretManagement.DevolutionsServer' -VaultParameters @{
    'Url'           = 'https://devsrv.domain.local/dps'
    'UserName'      = $Credentials.UserName
    'Password'     = ($Credentials.Password | ConvertFrom-SecureString -AsPlainText)
    'VaultId'      = 'bb63fd37-388a-42b4-ac1d-acc8f2618e8f'
}

```

```

PS C:\Users\testaccount1> $Credentials = Get-Credential

PowerShell credential request
Enter your credentials.
User: domain.local\testaccount1
Password for user domain.local\testaccount1: *****

PS C:\Users\testaccount1> Register-SecretVault -Name 'DevolutionsServer' -ModuleName 'SecretManagement.DevolutionsServer' -VaultParameters @{
>> 'Url'           = 'https://devsrv.domain.local/dps'
>> 'UserName'      = $Credentials.UserName
>> 'Password'     = ($Credentials.Password | ConvertFrom-SecureString -AsPlainText)
>> 'VaultId'      = 'bb63fd37-388a-42b4-ac1d-acc8f2618e8f'
>> }
PS C:\Users\testaccount1>

```

Enregistrement du coffre secret



Si vous utilisez le module PowerShell de Remote Desktop Manager, vous pouvez utiliser Get-RDMRepository pour obtenir la propriété VaultId nécessaire. Celle-ci est affichée comme la propriété Id renvoyée par l'applet de commande.

- Assurez-vous que le coffre fonctionne comme prévu avec l'applet de commande Test-SecretVault. Si True est renvoyé, le coffre s'est connecté avec succès.

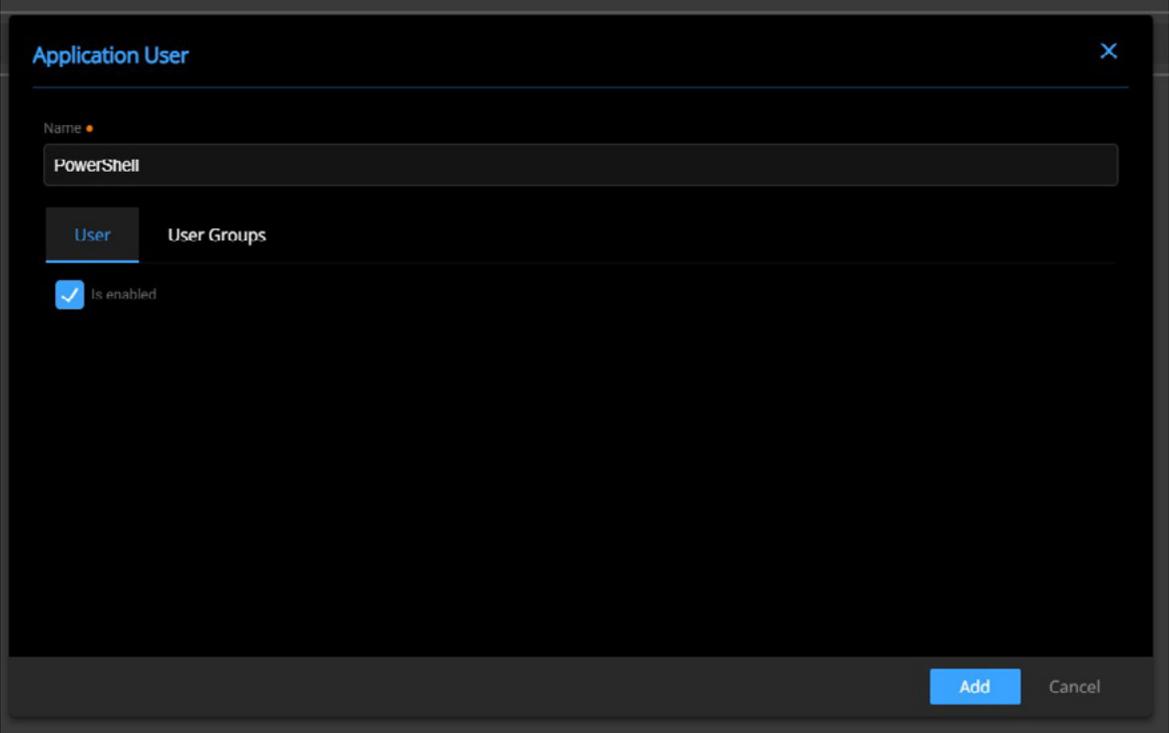
```
Test-SecretVault -Name 'DevolutionsServer'
```

```
PS C:\Users\testaccount1> Test-SecretVault -Name 'DevolutionsServer'  
True
```

Assurez-vous que le coffre peut se connecter

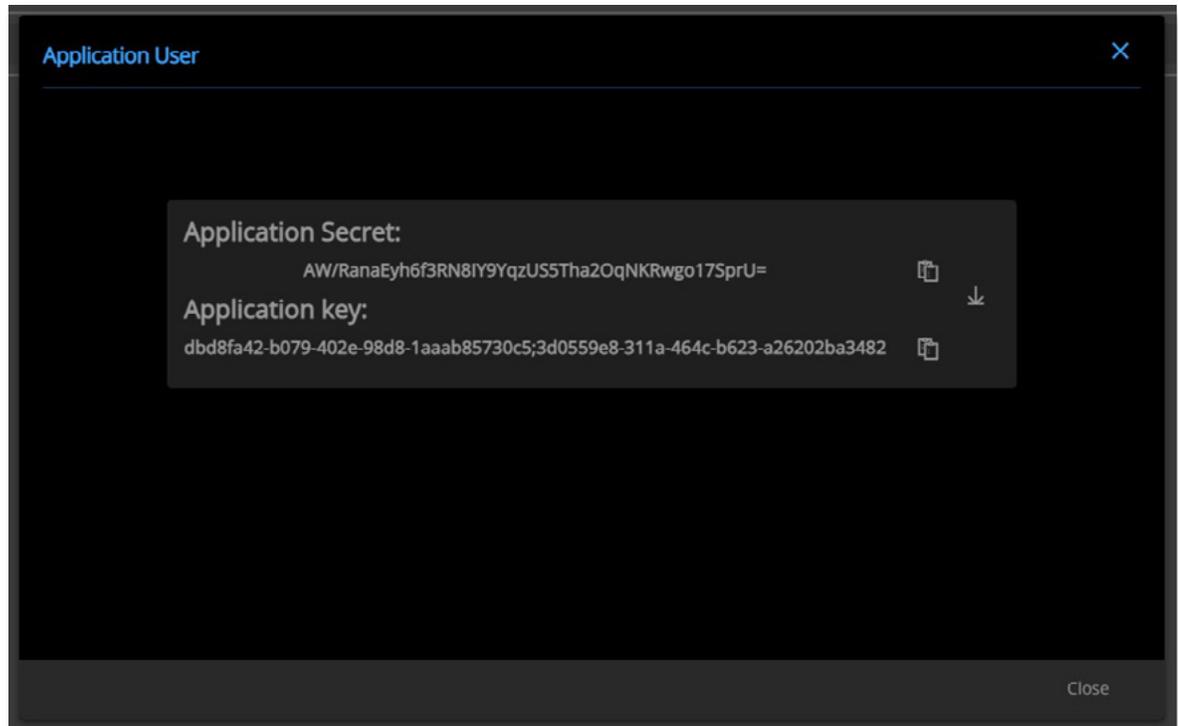
Connexion à Devolutions Hub

- En premier lieu, vous devez créer un Utilisateur de l'application. Connectez-vous à l'interface Web de Devolutions Hub et accédez à **Administration** → **Utilisateurs de l'application**. Cliquez ensuite sur le bouton **plus** pour ajouter un nouvel utilisateur en attribuant un nom et en veillant à ce que l'option **Est activée** est cochée.



Création d'un utilisateur d'application

Copiez le Secret d'application et la Clé d'application obtenus : ils seront utilisés pour connecter le module de gestion des secrets.

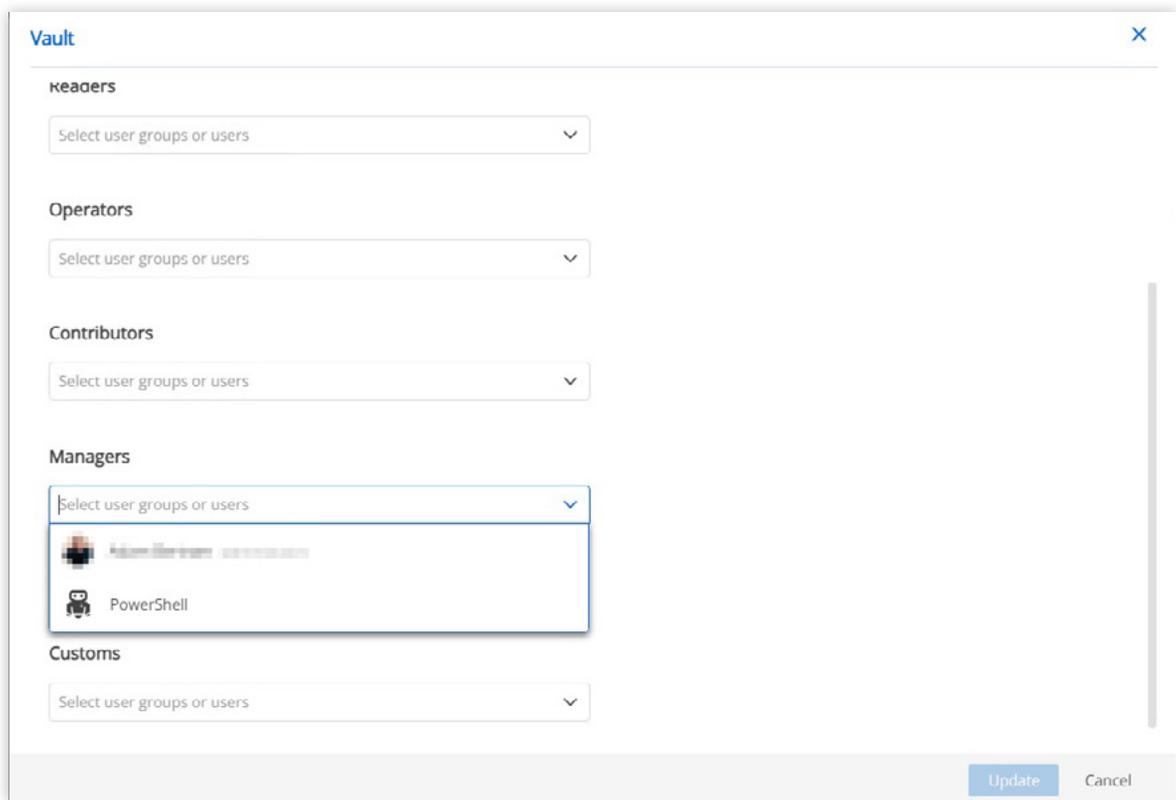


Récupération du Secret et de la Clé d'application

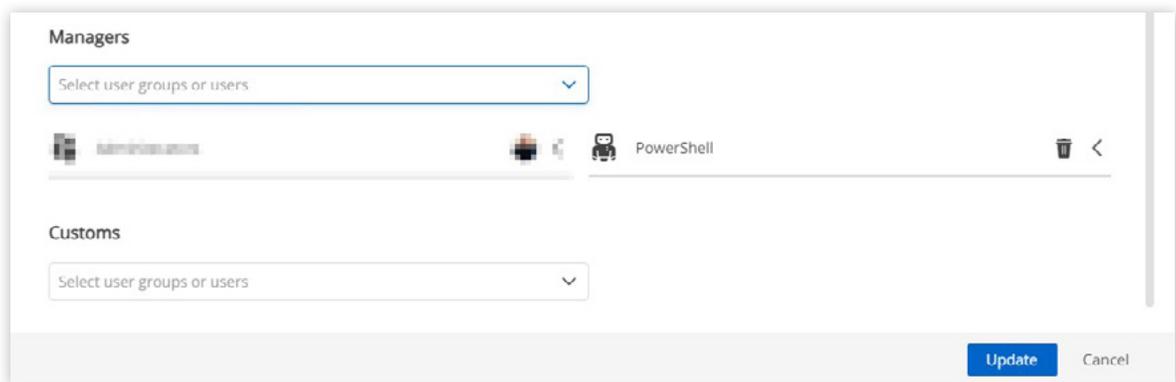


Le Secret d'application ne sera affiché qu'une seule fois. Veillez à le télécharger ou à le copier avant de fermer la fenêtre.

- 2.** Avant que l'utilisateur de l'application puisse accéder au coffre, vous devez au préalable accorder l'accès au compte. Accédez à **Administration** → **Coffres** → **Coffre par défaut** (utilisé dans ce tutoriel) et cliquez sur l'icône **Modifier** (crayon).
Pour les besoins de ce tutoriel, le compte est ajouté en tant que Gestionnaire.

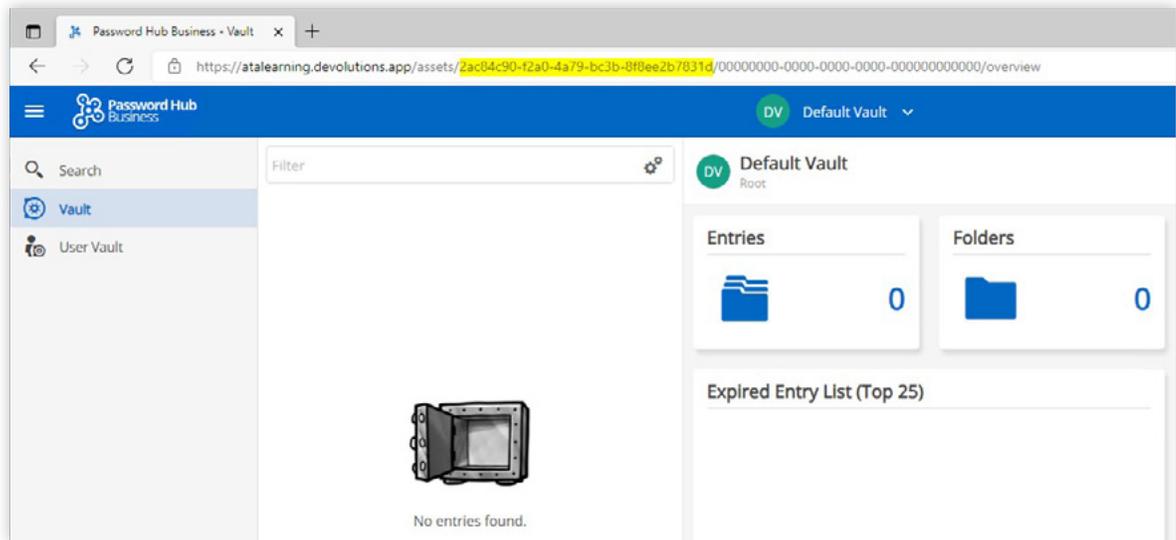


Sélection de l'utilisateur de l'application



Enregistrer les permissions mises à jour

3. La récupération de l'ID du coffre consiste à repérer celui-ci dans l'interface Web de Devolutions Hub en naviguant vers **Devolutions Hub → Coffres** et en localisant l'ID dans l'URL comme indiqué ci-dessous.



Récupération de l'ID du coffre

4. Ensuite, exécutez le code suivant pour enregistrer un coffre secret pour Devolutions Hub, en indiquant l'URL du serveur et les identifiants avec l'accès à l'ID du coffre fourni.

```
Register-SecretVault -Name 'DevolutionsHub' -ModuleName 'SecretManagement.DevolutionsHub'
-VaultParameters @{
    'Url' = 'https://atalearning.devolutions.app'
    'ApplicationKey' = 'dbd8-
fa42-b079-402e-98d8-1aaab85730c5;3d0559e8-311a-464c-b623-a26202ba3482'
    'ApplicationSecret' = 'AW/RanaEyh6f3RN8IY9YqzUS5Tha2OqNKRwgo17SprU='
    'VaultId' = '2ac84c90-f2a0-4a79-bc3b-8f8ee2b7831d'
}
```

```
PS C:\Users\testaccount1> Register-SecretVault -Name 'DevolutionsHub' -Module
Name 'SecretManagement.DevolutionsHub' -VaultParameters @{
>> 'Url' = 'https://atalearning.devolutions.app'
>> 'ApplicationKey' = 'dbd8fa42-b079-402e-98d8-1aaab85730c5;3d0559e8-311a-464c-b623-a26202ba3482'
>> 'ApplicationSecret' = 'AW/RanaEyh6f3RN8IY9YqzUS5Tha2OqNKRwgo17SprU='
>> 'VaultId' = '2ac84c90-f2a0-4a79-bc3b-8f8ee2b7831d'
>> }
PS C:\Users\testaccount1> |
```

Enregistrement du coffre secret

5. Assurez-vous que le coffre fonctionne comme prévu avec l'applet de commande Test-SecretVault comme indiqué ci-dessous.

```
Test-SecretVault -Name 'DevolutionsHub'
```

```
PS C:\Users\testaccount1> Test-SecretVault -Name 'DevolutionsHub'
True
PS C:\Users\testaccount1> |
```

Vérification que le coffre peut se connecter

Affichage des entrées du coffre

L'applet de commande `Get-SecretInfo` affiche les informations sur les entrées renvoyées afin de répertorier les entrées disponibles d'un coffre. Il est recommandé d'indiquer le paramètre `-Vault`, surtout si vous avez enregistré plusieurs coffres. Comme indiqué ci-dessous, deux entrées sont renvoyées.

```
Get-SecretInfo -Vault 'DevolutionsServer'
```



Le coffre référencé par `Get-SecretInfo` n'est pas le même que celui dans Devolutions Server ou Hub. Ici, le coffre est soit `DevolutionsServer` ou `DevolutionsHub`.

```
PS C:\Users\testaccount1> Get-SecretInfo -Vault 'DevolutionsServer'  
WARNING: [Close-DSSession] No session was previously established.  
  
Name                               Type                               VaultName  
----                               -  
RDM Credentials\RDPCConnection     PSCredential DevolutionsServer  
Remote Desktop\Domain Controller  PSCredential DevolutionsServer  
  
PS C:\Users\testaccount1> |
```

Récupération des entrées disponibles à partir de Devolutions Server

Modifiez le coffre indiqué pour la connexion de DevolutionsHub pour lister les entrées à partir de Devolutions Hub.

```
PS C:\Users\testaccount1> Get-SecretInfo -Vault 'DevolutionsHub'  
  
Name Type           VaultName  
---- -  
DC   PSCredential DevolutionsHub
```

Récupération des entrées de Devolutions Hub

Récupération des entrées d'un coffre

Utilisez l'applet de commande `Get-Secret` avec `Devolutions Server` pour récupérer une entrée individuelle du coffre comme illustré ci-dessous. Vous remarquerez que la valeur renvoyée est un objet `PSCredential` (la méthode privilégiée). Vous pouvez cependant préciser l'option `AsPlainText` pour récupérer la valeur non chiffrée.



Si aucun nom d'utilisateur ou mot de passe n'est attribué, les valeurs renvoyées seront vides.

```
Get-Secret -Name 'RDM Credentials\RDPConnection' -Vault 'DevolutionsServer'
```

```
PS C:\Users\testaccount1> Get-Secret -Name 'RDM Credentials\RDPConnection' -Vault 'DevolutionsServer'  
WARNING: [Close-DSSession] No session was previously established.  
  
UserName                Password  
-----                -  
domain.local\TestAccount1 System.Security.SecureString
```

Récupération d'une entrée de Devolutions Server

Tout comme pour `Devolutions Server`, il suffit d'utiliser la même commande `Get-Secret` pour récupérer un secret à partir de `Devolutions Hub`. Il faut cependant définir `DevolutionsHub` comme coffre.

```
Get-Secret -Name 'TestUser' -Vault 'DevolutionsHub'
```

```
PS C:\Users\testaccount1> Get-Secret -Name 'TestUser' -Vault 'DevolutionsHub'  
  
UserName                Password  
-----                -  
TestUser System.Security.SecureString
```

Récupérer une entrée de Devolutions Hub



Vous pouvez aussi passer par l'identificateur global unique de l'entrée, plutôt que par le nom, avec le même nom de paramètre pour récupérer une entrée. Cette méthode est normalement plus rapide.

Récupération des mots de passe des entrées de coffre en texte brut

La propriété de mot de passe AsPlainText de Get-Secret ne retourne pas un mot de passe non chiffré : vous pouvez cependant utiliser la commande suivante pour déchiffrer la valeur du mot de passe. Cela fonctionne avec les entrées récupérées à partir de Devolutions Server et Devolutions Hub.

```
Get-Secret -Name 'RDM Credentials\RDPConnection' -Vault 'DevolutionsServer' | Select-Object  
-ExpandProperty Password | ConvertFrom-SecureString -AsPlainText
```

```
PS C:\Users\testaccount1> Get-Secret -Name 'RDM Credentials\RDPConnection' -Vault 'DevolutionsServer' | Select-Object -E  
xpandProperty Password | ConvertFrom-SecureString -AsPlainText  
WARNING: [Close-DSSession] No session was previously established.  
_eUDMYQr7gP22eJz
```

Récupération et convertissement d'un mot de passe d'une entrée en texte brut

Création d'entrées dans un coffre

L'applet de commande Set-Secret permet la création d'un nouveau secret dans Devolutions Server. Au moment de la rédaction de cet article, Set-Secret ne prend pas en charge la mise à jour des secrets existants. Une nouvelle entrée sera créée à chaque fois si un nom d'entrée existant est indiqué.

```
$Credentials = Get-Credential  
Set-Secret -Name 'ServiceUser' -Secret $Credentials -Vault 'DevolutionsServer'
```

```
PS C:\Users\testaccount1> $Credentials = Get-Credential  
  
PowerShell credential request  
Enter your credentials.  
User: domain.local\testaccount1  
Password for user domain.local\testaccount1: *****  
  
PS C:\Users\testaccount1> Set-Secret -Name 'ServiceUser' -Secret $Credentials -Vault 'DevolutionsServer'  
WARNING: [Close-DSSession] No session was previously established.
```

Création d'une nouvelle entrée dans Devolutions Server



Pour le moment, la fonctionnalité Set-Secret de Devolutions Hub ne fonctionne pas.

Intégration de la gestion des secrets Devolutions dans les scripts

Il est fréquent qu'un script côté serveur demande des identifiants pour gérer un processus. Plutôt que de coder en dur ces identifiants dans le script, utilisez les capacités de la gestion des secrets de Devolutions afin de récupérer les identifiants nécessaires, comme le montre l'exemple ci-dessous.

```
# Verify that the vault connection is available
If (Test-SecretVault -Name 'DevolutionsServer') {
    $Credential = Get-Secret -Name 'ServiceUser' -Vault 'DevolutionsServer'

    # Verify that a credential is returned
    If ($Credential) {
        # Retrieve the service details
        $Service = Get-Service -Name "ImportantService"

        # If the existing service username does not match that of the credential
        # update the service
        If ($Service.UserName -NE $Credential.UserName) {
            Write-Host "Updating Service Credential"
            Set-Service -Name "ImportantService" -Credential $Credential
        }
    }
}
```

```
PS C:\Users\testaccount1> Import-Module Microsoft.PowerShell.SecretManagement, SecretManagement.DevolutionsServer
PS C:\Users\testaccount1>
PS C:\Users\testaccount1> If (Test-SecretVault -Name 'DevolutionsServer') {
>> $Credential = Get-Secret -Name 'ServiceUser' -Vault 'DevolutionsServer'
>>
>> If ($Credential) {
>> $Service = Get-Service -Name "ImportantService"
>>
>> If ($Service.UserName -NE $Credential.UserName) {
>> Write-Host "Updating Service Credential"
>> Set-Service -Name "ImportantService" -Credential $Credential
>> }
>> }
>> }
WARNING: [Close-DSSession] No session was previously established.
WARNING: [Close-DSSession] No session was previously established.
Updating Service Credential
PS C:\Users\testaccount1> |
```

Exécution d'un script d'intégration du module de gestion des secrets de Devolutions