COMMENT

configurer une source de données DVLS avec le module PowerShell de Remote Desktop Manager

Devolutions | WHERE IT MEETS SECURITY

Remote Desktop Manager propose un module PowerShell complet permettant de gérer Devolutions Server, mais aussi le client de Remote Desktop Manager (RDM). Ainsi, vous pouvez provisionner un client de RDM presque uniquement à partir de la ligne de commande de PowerShell sans aucune interaction avec l'interface graphique.

> *La version de Remote Desktop Manager utilisée est 2022.1.27.0.*

Installation du module PowerShell de Remote Desktop Manager

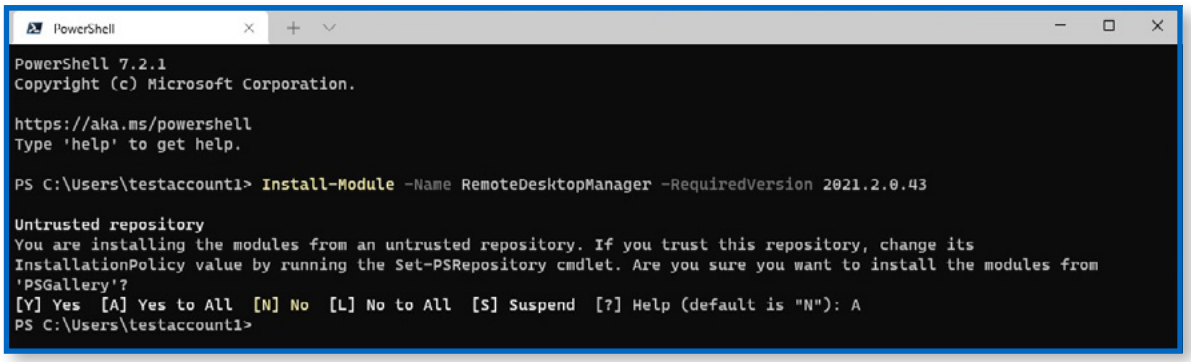
Vous devez installer le module PowerShell de Remote Desktop Manager à partir de la galerie PowerShell avant la configuration de RDM.

1. Lancez PowerShell 7 (le choix en tant qu'administrateur est facultatif). PowerShell 7.2.1 lancé dans le terminal Windows est utilisé pour cet article.

2. Installez la dernière version du paquet RemoteDesktopManager (**2021.2.0.43** au moment de la rédaction de cet article). Choisissez soit, **[Y] Yes** ou **[A] Yes to All** (le second indique que vous ne serez pas invité pour les installations ultérieures de modules à partir de ce référentiel) lorsque vous êtes invité à installer à partir de la galerie PowerShell.

```
Install-Module -Name RemoteDesktopManager -RequiredVersion 2021.2.0.43
```

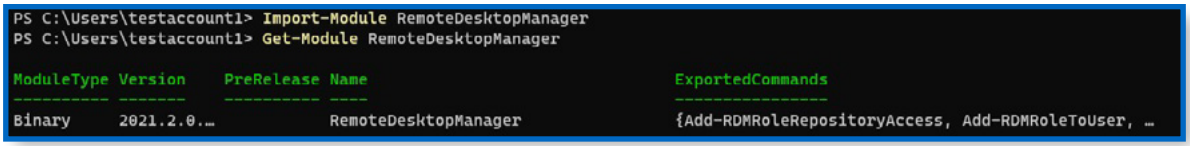
> *Il suffit de ne pas écrire le paramètre RequiredVersion pour installer la dernière version. De plus, vous pouvez choisir d'installer le module pour l'utilisateur actuel uniquement avec le paramètre Scope CurrentUser.*



Installation du module PowerShell de Remote Desktop Manager.

3. Importez le module PowerShell à partir du Import-Module, et vérifiez que le module est maintenant accessible à l'aide de Get-Module, comme illustré ci-dessous. Pour afficher toutes les commandes proposées, utilisez Get-Command -Module RemoteDesktopManager.

```
Import-Module RemoteDesktopManager
Get-Module RemoteDesktopManager
```



Importation du module et s'assurer qu'il est utilisable.

Ajout d'une source de données Devolutions Server avec l'authentification Windows

Un coffre local SQLite par défaut est créé et prêt à être utilisé lorsque vous lancez RDM pour la première fois. Une connexion à distance à Devolutions Server (DVLS) est utilisée pour stocker et gérer les autorisations liées aux coffres et aux entrées.

*Les cmdlets PowerShell fonctionnent sur le fichier **RemoteDesktopManager.cfg**, habituellement stocké dans un emplacement tel que "$($Env:LOCALAPPDATA)\\Devolutions\\RemoteDesktopManager". Vous devrez vous assurer que vos paramètres fonctionnent sur le même fichier si les modifications n'apparaissent pas.*

1. Puisque Devolutions Server (DVLS) est utilisé, la première étape consiste à ajouter la source de données DVLS. Pour ce faire, utilisez le cmdlet Get-RDMDataSource qui répertorie toutes les sources de données RDM connues localement. Ici, la source de données « **Local Data Source** » SQLite par défaut est affichée.

```
Get-RDMDataSource
```

```
PS C:\Users\testaccount1> Get-RDMDataSource

ID           : 4c30af75-054a-4aeb-8d1e-8a873c9038cb
IsConnected  : False
IsOffline    : False
Name         : Local Data Source
Type         : SQLite
```

Répertorie toutes les sources de données de RDM.

2. Créez la source de données à l'aide du cmdlet New-RDMDataSource pour ajouter une nouvelle source de données. Au départ, la source de données n'existe qu'en mémoire. L'objet créé doit par la suite être transféré à Set-RDMDataSource qui effectue les modifications à votre instance de RDM.

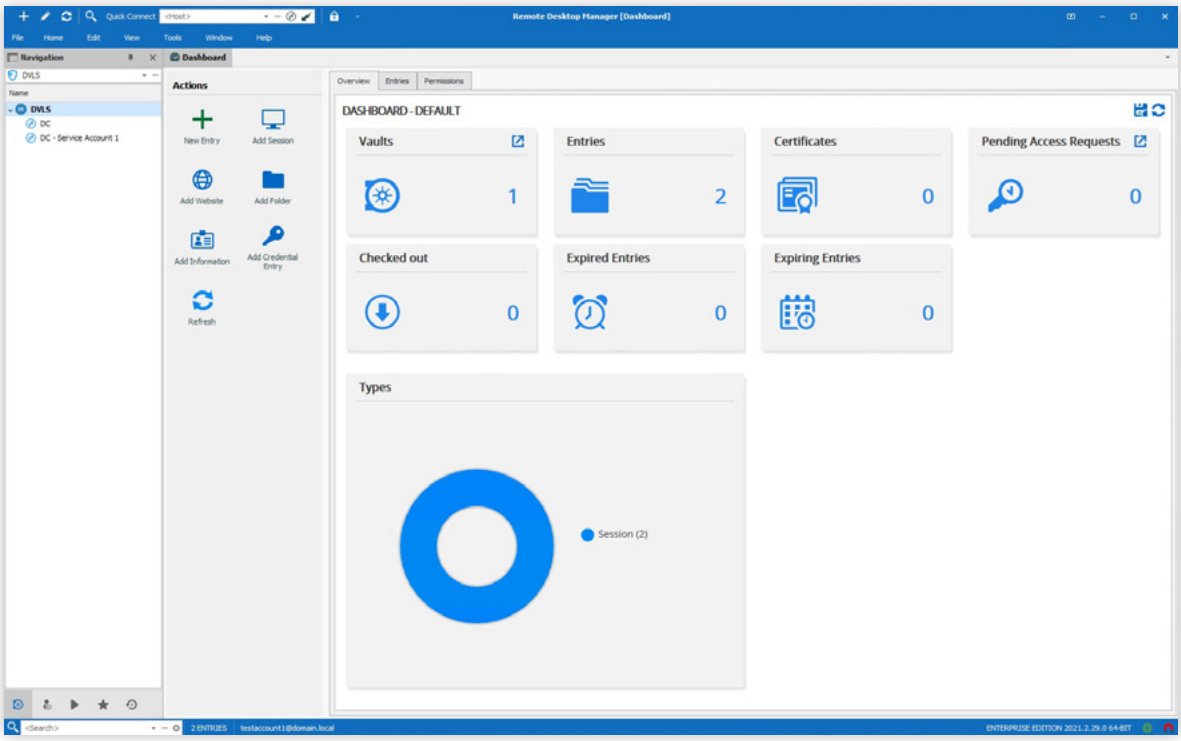
Comme cette instance de DVLS est connectée à un domaine Active Directory, cette source de données utilise l'authentification Windows pour connecter de manière transparente l'utilisateur.

```
$DataSource = New-RDMDataSource -DVLS -Name 'DVLS' -Server '<https://devsrv.domain.local/dps>'
Set-RDMDataSourceProperty -DataSource $DataSource -Property 'UseWindowsAuthentication' -Value
$True
$DataSource | Set-RDMDataSource | Update-RDMUI
```

```
PS C:\Users\testaccount1> $DataSource = New-RDMDataSource -DVLS -Name 'DVLS' -Server 'https://devsrv.domain.local/dps'
PS C:\Users\testaccount1> Set-RDMDataSourceProperty -DataSource $DataSource -Property 'UseWindowsAuthentication' -Value
$True
PS C:\Users\testaccount1> $DataSource | Set-RDMDataSource | Update-RDMUI
PS C:\Users\testaccount1>
```

Création de la nouvelle source de données DVLS avec l'authentification Windows.

3. Au lancement de RDM, une nouvelle option de source de données apparaît dans la liste déroulante de navigation. Choisissez **DVLS**. Les entrées autorisées pour ce compte sont automatiquement disponibles grâce à l'authentification Windows intégrée.



Connexion à la source de données DVLS.

Ajout d'un nouveau coffre (référentiel)

Jusque-là, un seul coffre (référentiel) a été utilisé. Il serait peut-être préférable de créer un nouveau coffre pour séparer les entrées. Vous pouvez créer un nouveau coffre local, mais comme DVLS est utilisé, il est recommandé de créer le nouveau coffre sur le serveur DVLS afin que plusieurs utilisateurs puissent y accéder.

1. Vous devez d'abord récupérer la source de données DVLS dans RDM (appelée DVLS dans cet exemple) avant l'ajout d'un nouveau coffre (référentiel) à votre source de données DVLS par PowerShell. Ensuite, configurez la source de données actuelle avec la source de données DVLS récupérée et mettez à jour l'interface utilisateur RDM pour afficher vos changements.

Vous devez être un administrateur de DVLS pour créer un coffre. Si vous avez modifié votre type de licence, quittez la session PowerShell et ouvrez-la à nouveau.

```
$DataSource = Get-RDMDataSource -Name 'DVLS'
$DataSource | Set-RDMCurrentDataSource | Update-RDMUI
```

```
PS C:\Users\testaccount1> $DataSource = Get-RDMDataSource -Name 'DVLS'
PS C:\Users\testaccount1> $DataSource | Set-RDMCurrentDataSource | Update-RDMUI
PS C:\Users\testaccount1>
```

Passer à la source de données de DVLS adéquate.

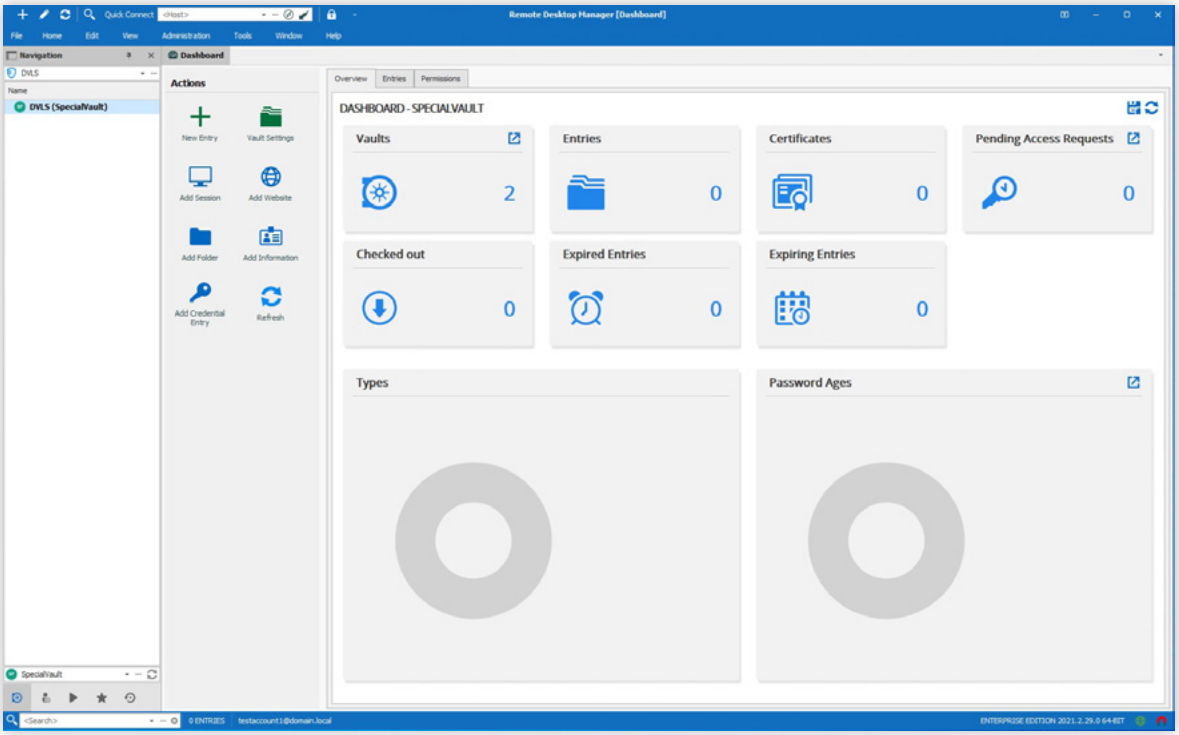
2. Créez par la suite un nouveau coffre (SpecialVault dans cet exemple) à l'aide du cmdlet New-RDMRepository. Un nom est proposé, ainsi que le paramètre – SetRepository. Celui-ci remplace l'appel supplémentaire de la commande à Set-RDMCurrentRepository.

```
$Vault = New-RDMRepository -Name 'SpecialVault' -SetRepository
Update-RDMUI
```

```
PS C:\Users\testaccount1> $Vault = New-RDMRepository -Name 'SpecialVault' -SetRepository
PS C:\Users\testaccount1> Update-RDMUI
PS C:\Users\testaccount1>
```

Création d'un nouveau coffre.

3. Assurez-vous que le nouveau coffre a été créé en lançant RDM et en naviguant vers le coffre SpecialVault récemment créé comme illustré ci-dessous. Il suffit de le choisir dans le menu déroulant de navigation en bas à droite.



Navigation vers le coffre récemment créé

Création de nouveaux dossiers dans SpecialVault

Une fois le coffre créé, il est préférable de classer les sessions (entrées) dans des dossiers individuels. Cette séparation logique favorise une gestion simplifiée.

1. Comme précédemment, assurez-vous que vous utilisez la bonne source de données et le bon coffre (référentiel), tel qu'indiqué dans le code ci-dessous.

```
$DataSource = Get-RDMDataSource -Name 'DVLS'
$DataSource | Set-RDMCurrentDataSource | Update-RDMUI
$Vault = Get-RDMRepository -Name 'SpecialVault'
Set-RDMCurrentRepository $Vault | Update-RDMUI
```

```
PS C:\Users\testaccount1> $DataSource = Get-RDMDataSource -Name 'DVLS'
PS C:\Users\testaccount1> $DataSource | Set-RDMCurrentDataSource | Update-RDMUI
PS C:\Users\testaccount1> $Vault = Get-RDMRepository -Name 'SpecialVault'
PS C:\Users\testaccount1> Set-RDMCurrentRepository $Vault | Update-RDMUI
PS C:\Users\testaccount1>
```

Connexion à la bonne source de données et au coffre (référentiel).

2. Créez ensuite un dossier nommé RDM Crendentials avec le cmdlet New-RDMSession doté du paramètre -Type of Group qui indique un type de dossier. Transférez l'objet en mémoire récemment créé à Set-RDMSession afin d'appliquer les modifications à RDM et DVLS.

```
$RDMCredentialFolder = New-RDMSession -Name 'RDM Credentials' -Type 'Group'
Set-RDMSession -Session $RDMCredentialFolder -Refresh | Update-RDMUI
```

```
PS C:\Users\testaccount1> $CredentialFolder = New-RDMSession -Name 'RDM Credentials' -Type 'Group'
PS C:\Users\testaccount1> Set-RDMSession -Session $CredentialFolder -Refresh | Update-RDMUI
PS C:\Users\testaccount1>
```

Création d'un dossier d'entrée RDM Credentials.

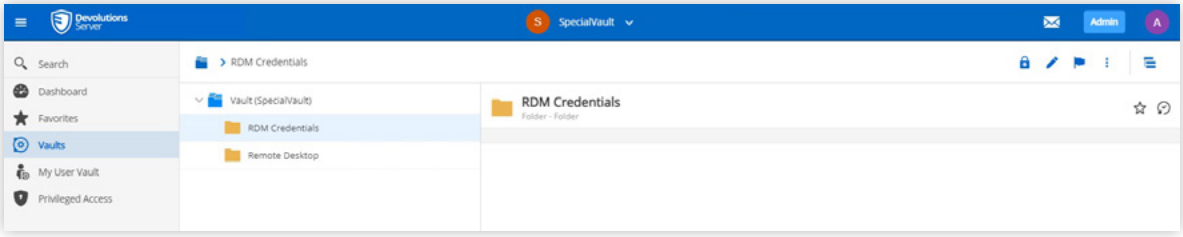
3. Créez un dossier Remote Desktop en suivant la même méthode.

```
$RemoteDesktopFolder = New-RDMSession -Name 'Remote Desktop' -Type 'Group'
Set-RDMSession -Session $RemoteDesktopFolder -Refresh | Update-RDMUI
```

```
PS C:\Users\testaccount1> $RemoteDesktopFolder = New-RDMSession -Name 'Remote Desktop' -Type 'Group'
PS C:\Users\testaccount1> Set-RDMSession -Session $RemoteDesktopFolder -Refresh | Update-RDMUI
PS C:\Users\testaccount1>
```

Création d'un dossier d'entrée Remote Desktop.

4. Il y maintenant deux dossiers créés dans le SpecialVault. Ceux-ci seront utilisés pour stocker les identifiants et les entrées de Bureau à distance (RDP).



Dossiers récemment créés.

Création d'une entrée

Vous avez besoin d'une session (entrée) à partir de laquelle vous pouvez vous connecter dans le nouveau coffre. Suivez les étapes ci-dessous pour créer une session enregistrée dans le SpecialVault.

1. Assurez-vous que vous utilisez la bonne source de données et le bon coffre (référentiel) comme indiqué ci-dessous.

```
$DataSource = Get-RDMDataSource -Name 'DVLS'
$DataSource | Set-RDMCurrentDataSource | Update-RDMUI
$Vault = Get-RDMRepository -Name 'SpecialVault'
Set-RDMCurrentRepository $Vault | Update-RDMUI
```

```
PS C:\Users\testaccount1> $DataSource = Get-RDMDataSource -Name 'DVLS'
PS C:\Users\testaccount1> $DataSource | Set-RDMCurrentDataSource | Update-RDMUI
PS C:\Users\testaccount1> $Vault = Get-RDMRepository -Name 'SpecialVault'
PS C:\Users\testaccount1> Set-RDMCurrentRepository $Vault | Update-RDMUI
PS C:\Users\testaccount1>
```

Connexion à la source de données et au coffre appropriés (référentiel).

2. Ensuite, configurez les informations de la session à créer. Tout d'abord, créez une entrée d'identifiants qui sera ensuite utilisée comme identifiants pour une connexion RDP déterminée. Cela sépare les identifiants d'une entrée pour simplifier la mise à jour d'un identifiant utilisé dans plusieurs sessions.

```
$RDMCredential = New-RDMSession -Name "RDPConnection" -Type 'Credential' -Group
'RDM Credentials'
$RDMCredential.Credentials.UserName = "domain.local\\TestAccount1"
Set-RDMSession $RDMCredential -Refresh
Set-RDMSessionPassword -ID $RDMCredential.ID -Password (ConvertTo-SecureString
'_eUDMYQr7gP22eJz' -AsPlainText -Force) | Update-RDMUI
```

```
PS C:\Users\testaccount1> $RDMCredential = New-RDMSession -Name "RDPConnection" -Type 'Credential' -Group 'RDM Credentia
ls'
PS C:\Users\testaccount1> $RDMCredential.Credentials.UserName = "domain.local\TestAccount1"
PS C:\Users\testaccount1> Set-RDMSession $RDMCredential -Refresh
PS C:\Users\testaccount1> Set-RDMSessionPassword -ID $RDMCredential.ID -Password (ConvertTo-SecureString '_eUDMYQr7gP22e
Jz' -AsPlainText -Force) | Update-RDMUI
PS C:\Users\testaccount1>
```

Création d'une entrée d'identifiants RDP à utiliser pour les entrées futures.

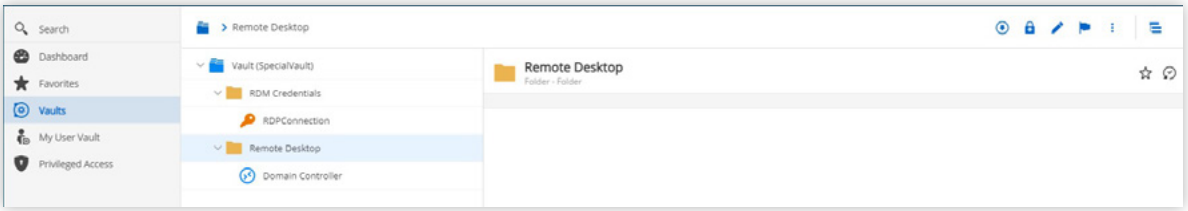
3. Lorsque les identifiants sont créés, ajoutez une session de Bureau à distance (RDP) standard (comme indiqué ci-dessous), et attribuez-lui l'ID de connexion précédemment créée.

```
$RemoteDesktop = New-RDMSession -Name 'Domain Controller' -Type 'RDPConfigured'
-Group 'Remote Desktop'
$RemoteDesktop.Host = 'dc.domain.local'
$RemoteDesktop.CredentialConnectionID = $RDMCredential.ID
Set-RDMSession $RemoteDesktop -Refresh | Update-RDMUI
```

```
PS C:\Users\testaccount1> $RemoteDesktop = New-RDMSession -Name 'Domain Controller' -Type 'RDPConfigured' -Group 'Remote
 Desktop'
PS C:\Users\testaccount1> $RemoteDesktop.Host = 'dc.domain.local'
PS C:\Users\testaccount1> $RemoteDesktop.CredentialConnectionID = $RDMCredential.ID
PS C:\Users\testaccount1> Set-RDMSession $RemoteDesktop -Refresh | Update-RDMUI
PS C:\Users\testaccount1>
```

Création de l'entrée RDP pour la connexion en cours.

4. Assurez-vous que les entrées sont désormais dans DVLS. Pour ce faire, naviguez vers **Devolutions Server → Coffres → SpecialVault** et en développant les dossiers **RDM Credentials** et **Remote Desktop**.



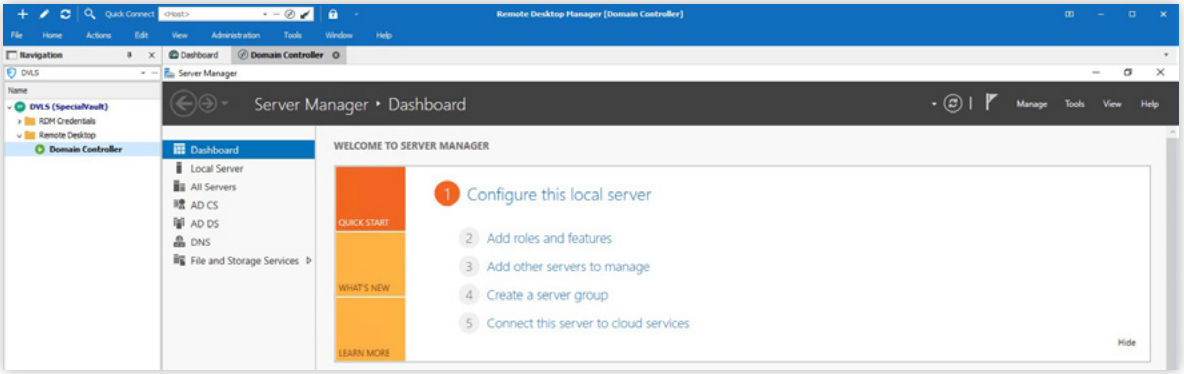
Assurez-vous que les entrées sont présentes dans le coffre.

5. Connectez-vous à la session avec l'entrée récemment créée.

```
Open-RDMSession -ID $RemoteDeskop.ID
```

```
PS C:\Users\testaccount1> Open-RDMSession -ID $RemoteDesktop.ID
PS C:\Users\testaccount1>
```

Ouverture d'une session à partir de la ligne de commande avec RDM



Affichage de la session ouverte dans RDM

En cas d'erreur, cette opération se termine par une erreur silencieuse.

Ajout d'un utilisateur

Vous avez peut-être déjà intégré un nouvel utilisateur, mais plutôt que de naviguer vers Devolutions Server, utilisez PowerShell pour créer le nouvel utilisateur dans DVLS.

> *L'utilisateur doit déjà exister dans Active Directory avant d'être ajouté dans DVLS.*

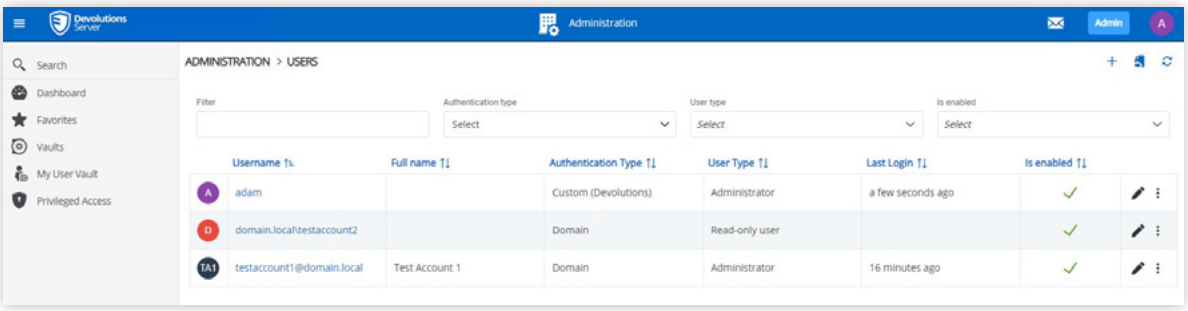
1. Ici, on transmet à New-RDMUser un compte de domaine Active Directory, configuré avec le paramètre -AuthentificationType of Domain. Appliquez les modifications à l'aide du cmdlet Set-RDMUser.

```
$User = New-RDMUser -Login 'domain.local\\testaccount2' -AuthentificationType 'Domain'
Set-RDMUser -User $User
```

```
PS C:\Users\testaccount1> $User = New-RDMUser -Login 'domain.local\testaccount2' -AuthentificationType 'Domain'
PS C:\Users\testaccount1> Set-RDMUser -User $User
PS C:\Users\testaccount1>
```

Création d'un nouvel utilisateur dans DVLS.

2. Le nouvel utilisateur est affiché ci-dessous dans l'interface Web de DVLS. Il est ajouté par défaut en tant qu'utilisateur en **lecture seule**.



Vérifier que le nouvel utilisateur du domaine DVLS a été ajouté.

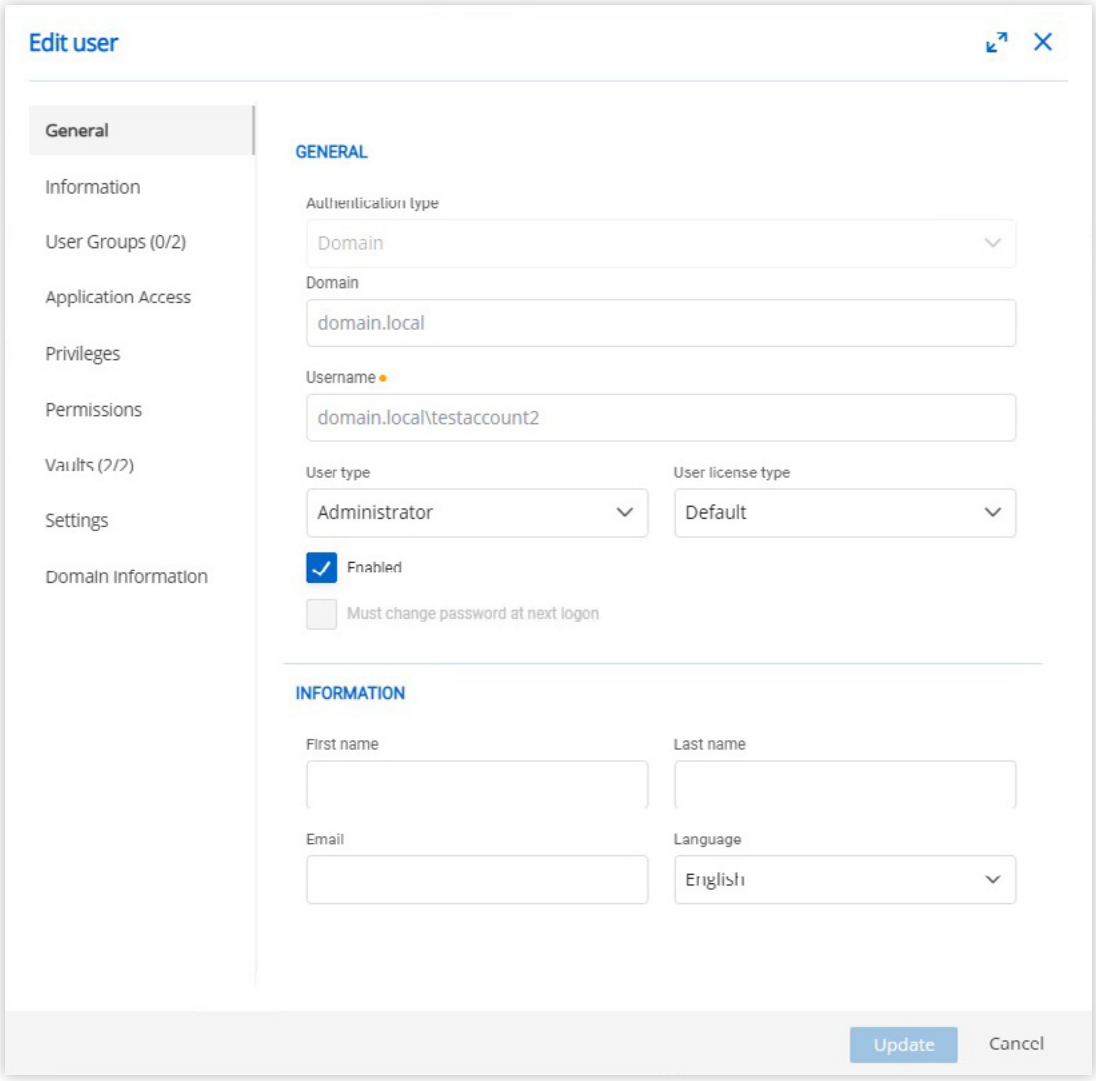
3. Vous pouvez indiquer un compte en tant qu'administrateur au besoin. Récupérez d'abord un utilisateur existant avec Get-RDMUser afin de définir IsAdministrator à $True. Puis, appliquez les modifications avec Set-RDMUser.

```
$User = Get-RDMUser -Name 'domain.local\\testaccount2'
$User.IsAdministrator = $True
Set-RDMUser -User $User
```

```
PS C:\Users\testaccount1> $User = Get-RDMUser -Name 'domain.local\testaccount2'
PS C:\Users\testaccount1> $User.IsAdministrator = $True
PS C:\Users\testaccount1> Set-RDMUser -User $User
PS C:\Users\testaccount1>
```

Définir un utilisateur en tant qu'administrateur.

4. Assurez-vous que les modifications ont été effectuées en naviguant vers **Devolutions Server → Administration → Utilisateurs** et en cliquant sur le nom d'utilisateur. Dans la section **Type d'utilisateur**, notez que le type est désormais **Administrateur**.



Vérification des changements de permissions de l'utilisateur.

Ajout d'un nouveau rôle

Vous souhaiteriez peut-être attribuer au nouvel utilisateur les permissions du coffre à un groupe précis, l'ajouter à un groupe Active Directory et supprimer les permissions personnalisées ajoutées.

> *Le groupe doit déjà exister dans Active Directory avant d'être ajouté à DVLS.*

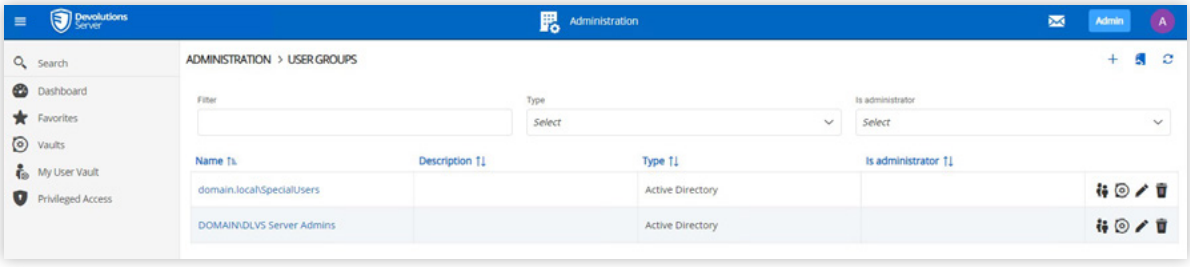
1. Dans l'exemple ci-dessous, créez un nouveau rôle à l'aide de New-RDMRole. Celui-ci est représenté par le groupe Active Directory existant domain.local\SpecialUsers. Ensuite, appliquez les modifications avec le cmdlet Set-RDMRole.

```
$Role = New-RDMRole -Name 'domain.local\\SpecialUsers'
Set-RDMRole -Role $Role
```

```
PS C:\Users\testaccount1> $Role = New-RDMRole -Name 'domain.local\SpecialUsers'
PS C:\Users\testaccount1> Set-RDMRole -Role $Role
PS C:\Users\testaccount1>
```

Création du nouveau rôle dans RDM.

2. Assurez-vous que le rôle est désormais dans **Devolutions Server → Administration → Groupe d'utilisateur**, comme indiqué ci-dessous.



Vérification du rôle récemment créé.

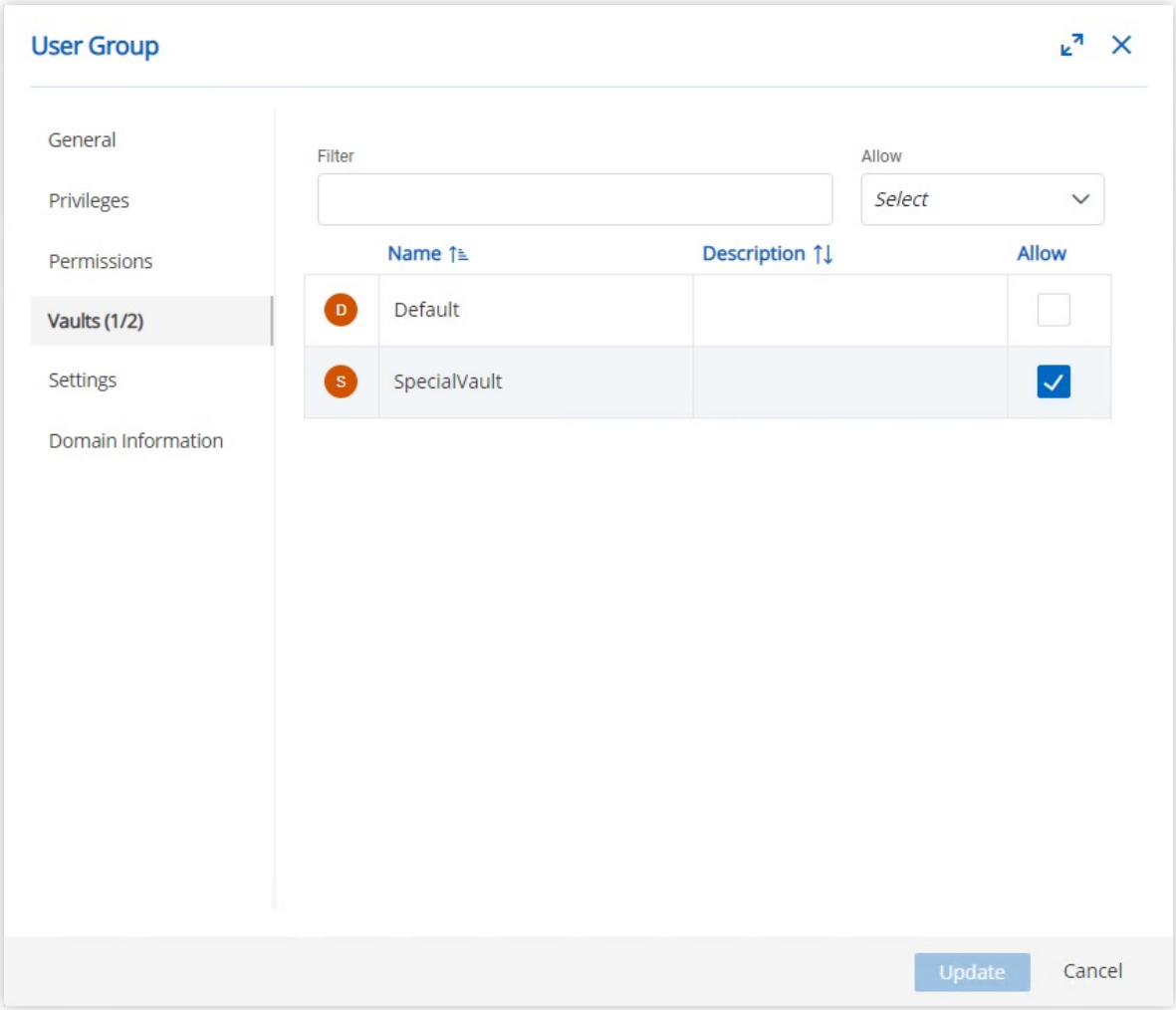
3. Ensuite, autorisez le rôle, domain.local\\SpecialUsers à accéder à SpecialVault.

```
$Vault = Get-RDMRepository -Name 'SpecialVault'
Set-RDMCurrentRepository -Repository $Vault | Update-RDMUI
Add-RDMRoleRepositoryAccess -Role $Role -Repository $Vault
Set-RDMRole -Role $Role
```

```
PS C:\Users\testaccount1> $Vault = Get-RDMRepository -Name 'SpecialVault'
PS C:\Users\testaccount1> Set-RDMCurrentRepository -Repository $Vault | Update-RDMUI
PS C:\Users\testaccount1> Add-RDMRoleRepositoryAccess -Role $Role -Repository $Vault
PS C:\Users\testaccount1> Set-RDMRole -Role $Role
PS C:\Users\testaccount1>
```

Accorder au rôle récemment créé l'accès à un coffre.

4. Assurez-vous que les modifications ont été effectuées en accédant à **Devolutions Server → Administration → Groupes d'utilisateurs → Coffres**.



Vérifier que le rôle a maintenant un accès autorisé.

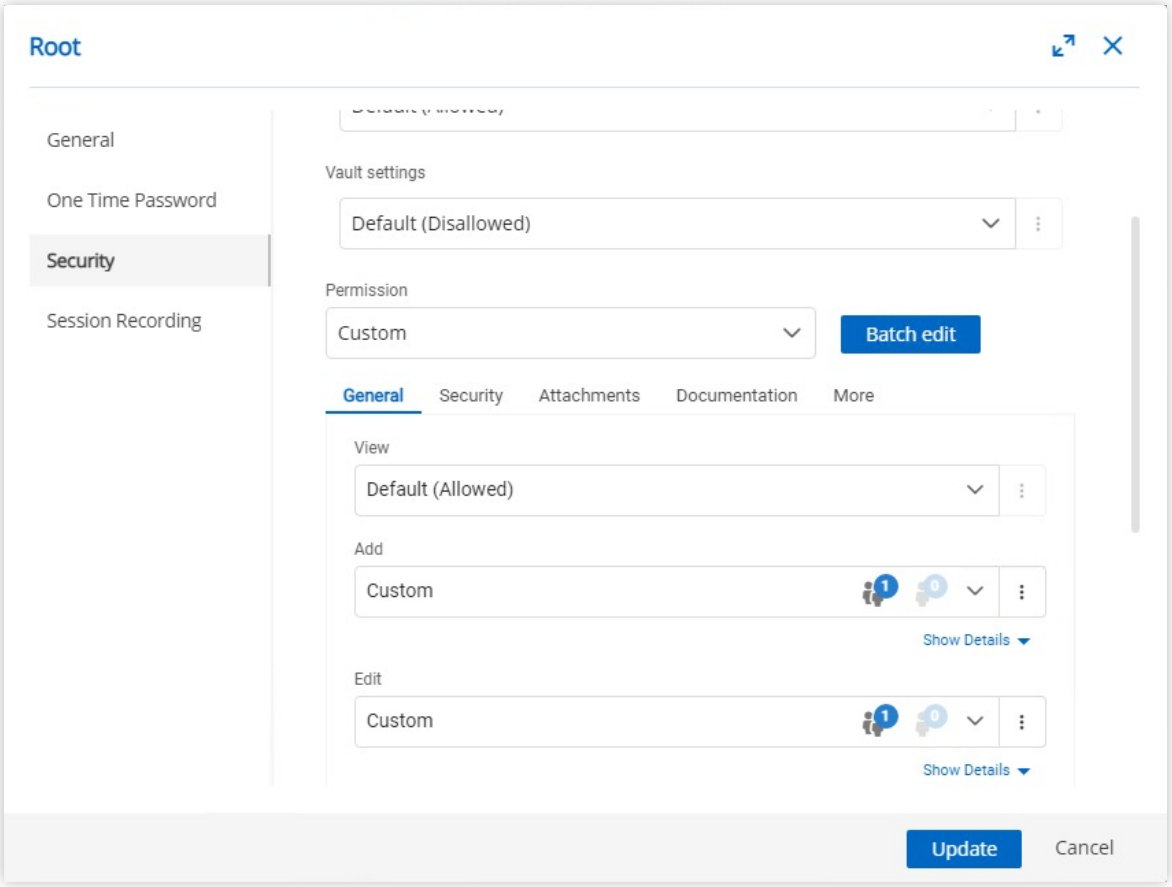
5. Modifiez les permissions du rôle afin de permettre l'ajout et la modification d'entrées dans SpecialVault.

```
$Permissions = @(
              [Devolutions.RemoteDesktopManager.Business.ConnectionPermission]@{
                     'Override' = 'Custom'
              'Right' = 'Add'
              'Roles' = 'domain.local\\SpecialUsers'
              'RoleValues' = 'domain.local\\SpecialUsers'
              }
              [Devolutions.RemoteDesktopManager.Business.ConnectionPermission]@{
              'Override' = 'Custom'
              'Right' = 'Edit'
              'Roles' = 'domain.local\\SpecialUsers'
                       'RoleValues' = 'domain.local\\SpecialUsers'
              }
)

$RDMRoot = Get-RDMRootSession
$RDMRoot.Security.RoleOverride = 'Custom'
$RDMRoot.Security.Permissions = $Permissions

$RDMRoot | Set-RDMRootSession
```

```
PS C:\Users\testaccount1> $Permissions = @(
>>    [Devolutions.RemoteDesktopManager.Business.ConnectionPermission]@{
>>      'Override' = 'Custom'
>>      'Right' = 'Add'
>>      'Roles' = 'domain.local\SpecialUsers'
>>      'RoleValues' = 'domain.local\SpecialUsers'
>>    }
>>    [Devolutions.RemoteDesktopManager.Business.ConnectionPermission]@{
>>      'Override' = 'Custom'
>>      'Right' = 'Edit'
>>      'Roles' = 'domain.local\SpecialUsers'
>>      'RoleValues' = 'domain.local\SpecialUsers'
>>    }
>> )
PS C:\Users\testaccount1> $RDMRoot = Get-RDMRootSession
PS C:\Users\testaccount1> $RDMRoot.Security.RoleOverride = 'Custom'
PS C:\Users\testaccount1> $RDMRoot.Security.Permissions = $Permissions
PS C:\Users\testaccount1> $RDMRoot | Set-RDMRootSession
PS C:\Users\testaccount1>
```

Modification des permissions pour le rôle.

6. Assurez-vous que les modifications ont été effectuées en accédant à **Devolutions Server → Coffres → Propriétés (clic droit sur la racine du coffre) → Sécurité → Permissions.**



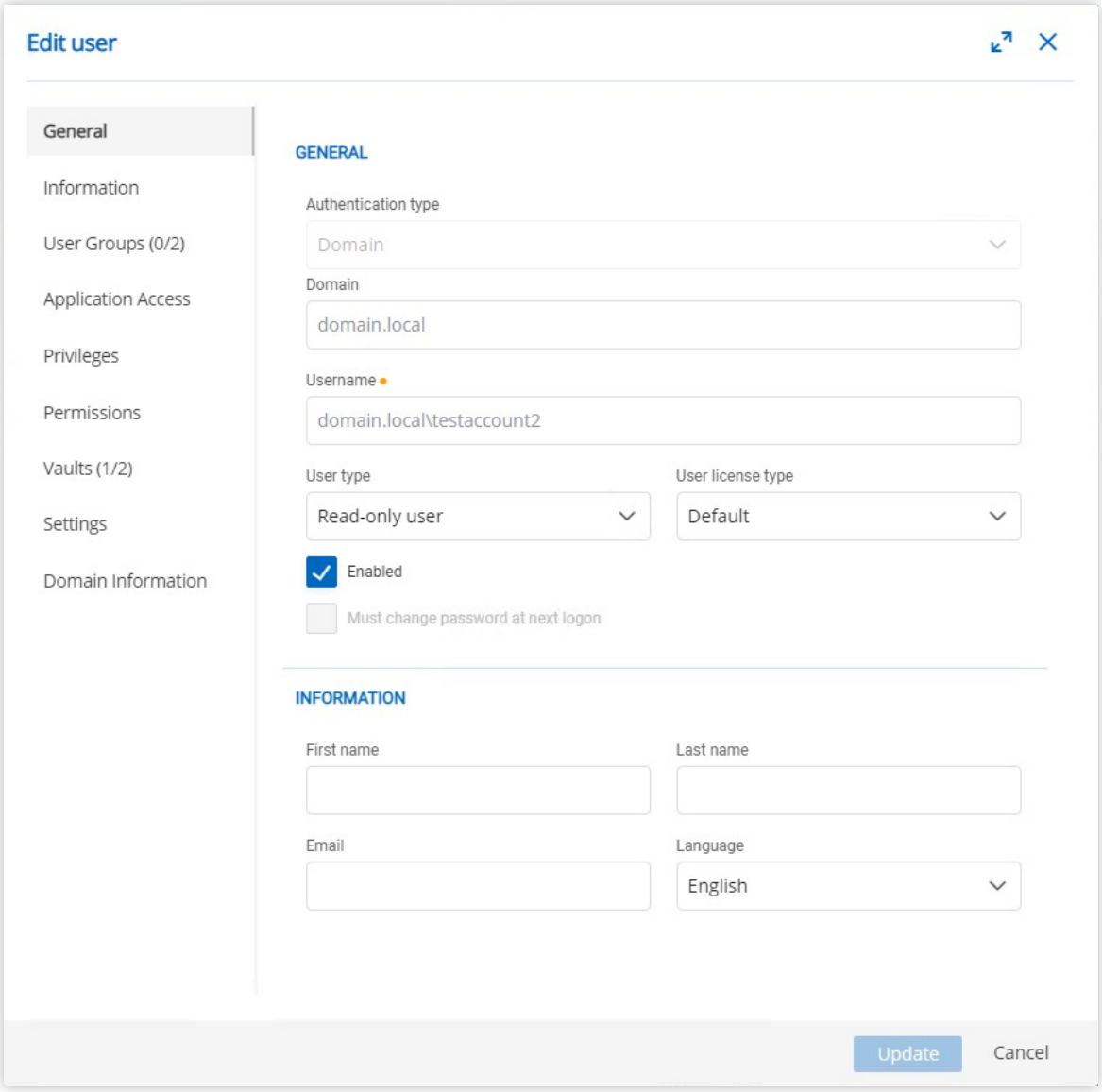
Vérifier que le rôle a des permissions personnalisées dans le coffre.

7. Pour conclure, supprimez les permissions personnalisées de l'utilisateur, ajoutées précédemment, car l'utilisateur a désormais ces mêmes permissions en raison de son appartenance au groupe Active Directory.

```
$User = Get-RDMUser -Name 'domain.local\\testaccount2'
$User.CanAdd = $False
$User.CanEdit = $False
Set-RDMUser -User $User
```

```
PS C:\Users\testaccount1> $User = Get-RDMUser -Name 'domain.local\testaccount2'
PS C:\Users\testaccount1> $User.CanAdd = $False
PS C:\Users\testaccount1> $User.CanEdit = $False
PS C:\Users\testaccount1> Set-RDMUser -User $User
PS C:\Users\testaccount1>
```

Retirer les permissions d'ajout et de modification personnalisées de l'utilisateur.



Vérifier que l'utilisateur n'a plus de permissions personnalisées.

Exportation et importation des sessions à des fins de sauvegarde

Comment faire pour créer une sauvegarde de vos sessions en cas de problème? Il est facile d'exporter un fichier XML contenant toutes les données nécessaires avec le cmdlet Export-RDMSession.

1. Assurez-vous que vous utilisez la bonne source de données et le bon coffre (référentiel), comme indiqué ci-dessous.

```
$DataSource = Get-RDMDataSource -Name 'DVLS'
$DataSource | Set-RDMCurrentDataSource | Update-RDMUI
$Vault = Get-RDMRepository -Name 'SpecialVault'
Set-RDMCurrentRepository $Vault | Update-RDMUI
```

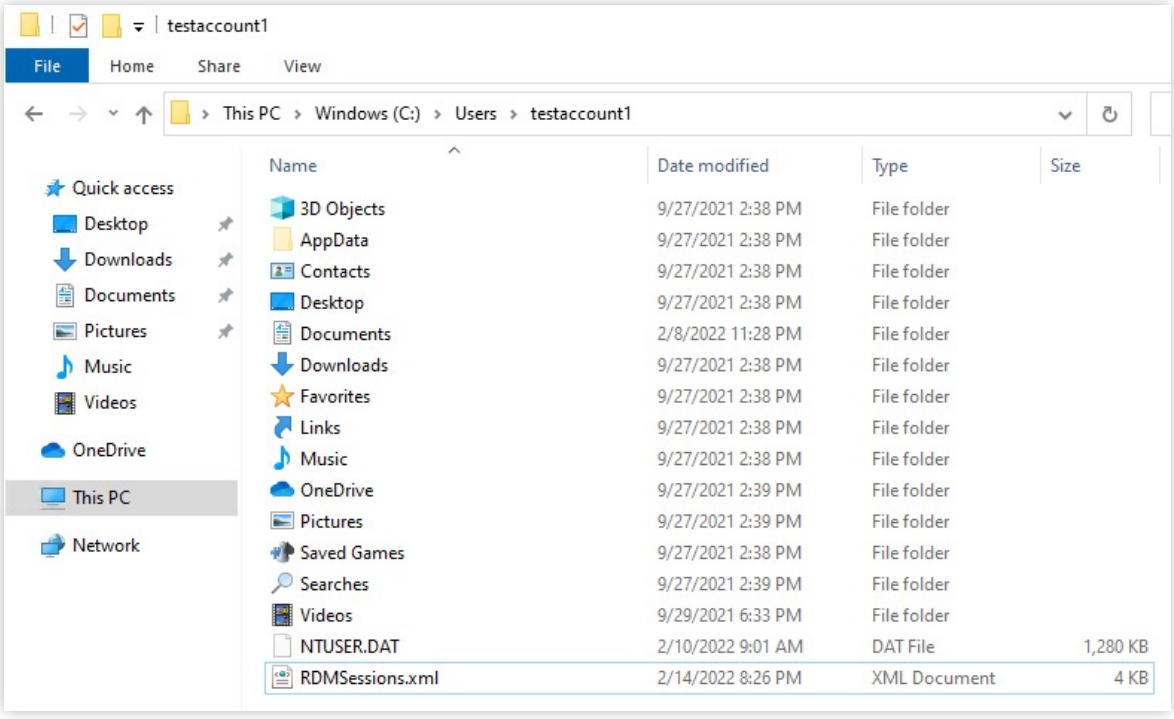
2. Ensuite, récupérez toutes les sessions, puis exportez le fichier XML vers le chemin donné. Dans cet exemple, le fichier comprend à la fois les identifiants et les groupes de sécurité.

```
$Sessions = Get-RDMSession
Export-RDMSession -Sessions $Sessions -Path "$($Env:USERPROFILE)\\RDMSessions.xml" -XML
-IncludeCredentials -IncludeSecurityGroups
```

```
PS C:\Users\testaccount1> $Sessions = Get-RDMSession
PS C:\Users\testaccount1> Export-RDMSession -Sessions $Sessions -Path "$($Env:USERPROFILE)\RDMSessions.xml" -XML -Includ
eCredentials -IncludeSecurityGroups
PS C:\Users\testaccount1>
```

Exportation des données de sessions XML.

3. Vérifiez que le fichier exporté existe, comme indiqué dans le fichier RDMSessions.xml ci-dessous.



Vérification de l'existence du fichier RDMSessions.xml.

4. Enfin, ajoutez les sessions exportées dans DVLS avec le cmdlet Import-RDMSession.

```
Import-RDMSession -Path "$($Env:USERPROFILE)\\RDMSessions.xml"
```

```
PS C:\Users\testaccount1> Import-RDMSession -Path "$($Env:USERPROFILE)\RDMSessions.xml"


AlternateShell                              :
AlwaysAskForResources                       : False
ApplicationIntegrationMode                  : Default
AuthentificationLevel                       : Default
AutoReconnection                            : True
CommandLine                                 :
CommandLineWaitForApplicationToExit         : False
CommandLineWorkingDirectory                 :
Console                                     : False
DesktopComposition                          : False
DisableBitmapCache                          : False
DisableCursorSetting                        : False
DisableFullWindowDrag                       : False
DisableMenuAnims                            : False
DisableThemes                               : False
DisableWallpaper                            : False
```

Importation des sessions de RDM dans DVLS.

Exportation et importation de rôles à des fins de sauvegarde

Comment faire pour créer une sauvegarde de vos rôles et de leurs permissions? Il n'y a pas de cmdlet standardisé pour cette étape, mais vous pouvez utiliser les capacités intégrées de PowerShell pour exporter les données vers un fichier XML pour une importation ultérieure.

1. Assurez-vous que vous utilisez la bonne source de données et le bon coffre (référentiel), comme indiqué ci-dessous.

```
$DataSource = Get-RDMDataSource -Name 'DVLS'
$DataSource | Set-RDMCurrentDataSource | Update-RDMUI
$Vault = Get-RDMRepository -Name 'SpecialVault'
Set-RDMCurrentRepository $Vault | Update-RDMUI
```

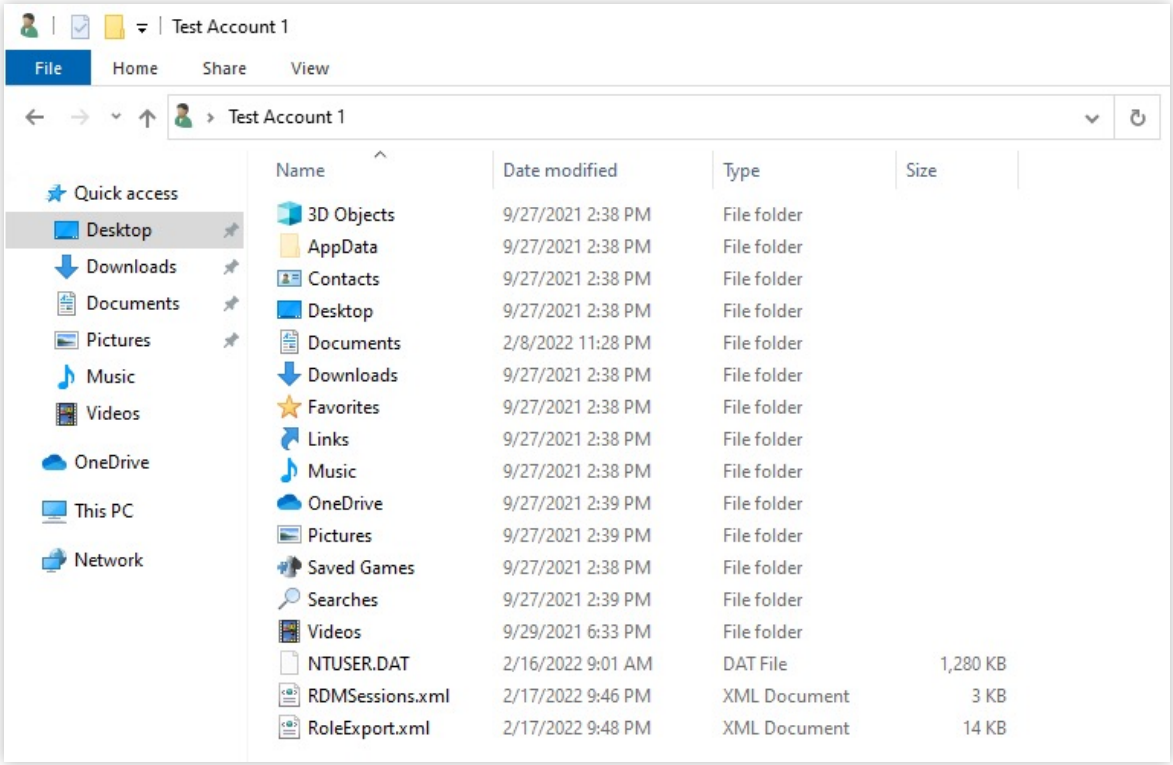
2. Ensuite, récupérez tous les rôles et exportez le fichier XML vers le chemin indiqué ci-dessous.

```
$Roles = Get-RDMRole
$Roles | Export-Clixml -Path "$($Env:USERPROFILE)\\RoleExport.xml"
```

```
PS C:\Users\testaccount1> $Roles = Get-RDMRole
PS C:\Users\testaccount1> $Roles | Export-Clixml -Path "$($Env:USERPROFILE)\RoleExport.xml"
PS C:\Users\testaccount1>
```

Exportation des rôles de RDM.

3. Confirmez que les rôles ont été exportés comme le montre le fichier RoleExport.xml ci-dessous.



Vérifier que le fichier des rôles exportés existe.

4. Voici un script simple qui prend les données de rôles exportées et les importe ensuite dans DVLS.

```
Import-Clixml -Path "$($Env:USERPROFILE)\\RoleExport.xml" | ForEach-Object {
       $Role = New-RDMRole -Name $PSItem.Name
Set-RDMRole -Role $Role
Set-RDMRoleProperty -Property "CanAdd" -Role $Role -Value $PSItem.CanAdd
Set-RDMRoleProperty -Property "CanDelete" -Role $Role -Value $PSItem.CanDelete
Set-RDMRoleProperty -Property "CanEdit" -Role $Role -Value $PSItem.CanEdit
Set-RDMRoleProperty -Property "CustomSecurity" -Role $Role -Value $PSItem.CustomSecurity
Set-RDMRoleProperty -Property "Description" -Role $Role -Value $PSItem.Description
Set-RDMRoleProperty -Property "Email" -Role $Role -Value $PSItem.Email

If ($PSItem.IsAdministrator) {
Set-RDMRoleProperty -Property "IsAdministrator" -Role $Role -Value $True
}

             Set-RDMRole -Role $Role
}
```

```
PS C:\Users\testaccount1> Import-Clixml -Path "$($Env:USERPROFILE)\RoleExport.xml" | ForEach-Object {
>>     $Role = New-RDMRole -Name $PSItem.Name
>>
>>     Set-RDMRole -Role $Role
>>
>>     Set-RDMRoleProperty -Property "CanAdd" -Role $Role -Value $PSItem.CanAdd
>>     Set-RDMRoleProperty -Property "CanDelete" -Role $Role -Value $PSItem.CanDelete
>>     Set-RDMRoleProperty -Property "CanEdit" -Role $Role -Value $PSItem.CanEdit
>>     Set-RDMRoleProperty -Property "CustomSecurity" -Role $Role -Value $PSItem.CustomSecurity
>>     Set-RDMRoleProperty -Property "Description" -Role $Role -Value $PSItem.Description
>>     Set-RDMRoleProperty -Property "Email" -Role $Role -Value $PSItem.Email
>>
>>     If ($PSItem.IsAdministrator) {
>>         Set-RDMRoleProperty -Property "IsAdministrator" -Role $Role -Value $True
>>     }
>>
>>     Set-RDMRole -Role $Role
>> }
PS C:\Users\testaccount1>
```

Importation de rôles exportés.

Si le rôle existe déjà, vous obtiendrez un message d'erreur (WARNING) : Unable to save user. De plus, vous devrez configurer à nouveau le rôle sur tout coffre dans lequel des autorisations personnalisées étaient précédemment définies.