Finding Orphaned OneDrive for Business Accounts

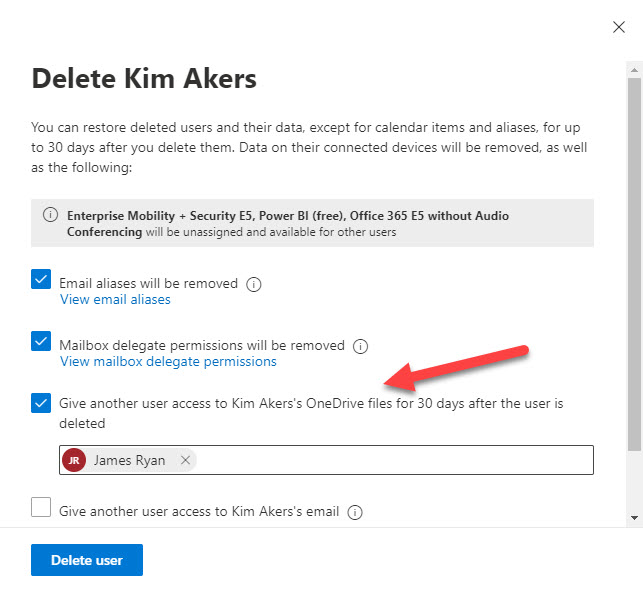
Posted on [**July 30, 2020**](https://office365itpros.com/2020/07/30/finding-orphaned-onedrive-for-business-accounts/) by [**Tony Redmond**](https://office365itpros.com/author/thoughtsofanidlemind/)

Use PowerShell to Assign New Users to Orphaned OneDrive Accounts

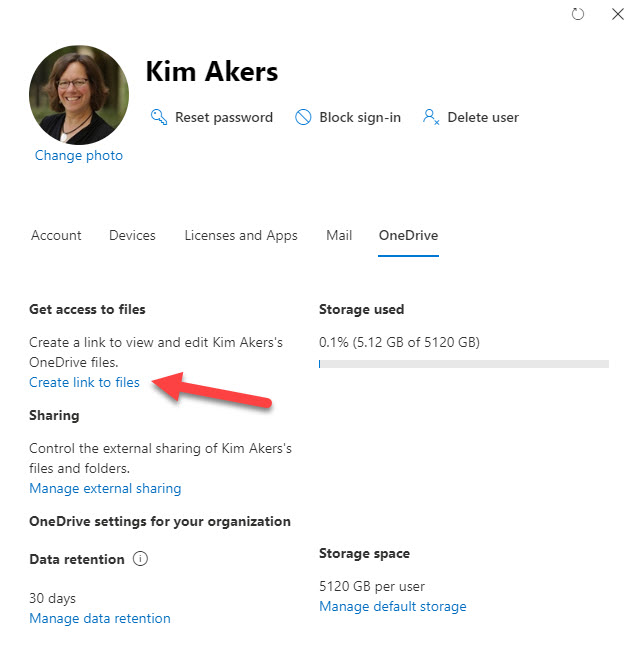
Given the growing importance of OneDrive for Business and user acceptance of features like Known Folder Move, which redirects [**well-known Windows folders like Documents and Pictures to OneDrive for Business**](https://docs.microsoft.com/en-us/onedrive/plan-onedrive-enterprise), lots of data ends up in user OneDrive accounts. And when those users leave the company, some action is usually needed to check the information in the account and recover anything which needs to be kept.

Making Sure OneDrive for Business Files are Kept

One way to do this is to assign another user access to the ex-employee’s OneDrive for Business account during the workflow to remove their Office 365 account (Figure 1). After they gain access, the user can move or copy information from the ex-employee’s files to their OneDrive for Business account or SharePoint Online sites.

Figure 1: Assigning access to a user’s OneDrive for Business account when their account is deleted

If you don’t delete the ex-employee’s account to regain the Office 365 license, you can [**create a link to access their OneDrive account**](https://docs.microsoft.com/en-us/microsoft-365/admin/add-users/get-access-to-and-back-up-a-former-user-s-data?view=o365-worldwide) by accessing the account in the Microsoft 365 admin center and going to the OneDrive tab in user properties (Figure 2). Again, once access is secured, you can review the files in the account and retrieve whatever needs to be kept.

Figure 2: Getting a link to a user’s OneDrive for Business account

Recovering OneDrive Accounts with PowerShell

Once an account is deleted, Office 365 keeps it for 30 days to allow mistaken deletions to be reversed. After this period elapses, the account is deleted, and workloads remove the information belonging to the account, such as the mailbox. Workloads have the liberty of processing deletions in their own ways, so it is possible to [**recover a OneDrive for Business account using PowerShell**](https://docs.microsoft.com/en-us/onedrive/restore-deleted-onedrive) in the period between 30 and 93 days after deletion because this is how the two-phase recycle bin process works.

The Influence of Retention Policies

If the ex-employee’s account comes within the scope of a retention policy, their OneDrive for Business account can be kept for even longer because the retention policy will kick in when OneDrive for Business tries to remove the account after 93 days. And as the years go by, it’s possible that a set of orphaned accounts might accumulate if retention policies keep accounts for a long time or do not delete the accounts after the retention period elapses.

Processing Orphan OneDrive for Business Accounts

To know if any orphan OneDrive for Business accounts exist, we can run some PowerShell to find OneDrive sites that aren’t connected by comparing the registered site owner to a hash table of Azure Active Directory accounts. If a match doesn’t exist, we have an orphan site. We can then add a user to orphan sites, perhaps a compliance administrator, to allow the contents of the sites to be examined. A [**complete script can be downloaded from GitHub**](https://github.com/12Knocksinna/Office365itpros/blob/master/FindOrphanOneDriveSites.PS1), but here’s the core code:

# Find Azure AD accounts

# Find OneDrive for Business accounts

$ODSites = Get-SPOSite -IncludePersonalSite $True -Limit All -Filter "url -like '-my.sharepoint.com/personal/'"

# Find Azure AD Accounts and create hash table for lookup

$AADUsers = Get-AzureADUser -All $True -Filter "Usertype eq 'Member'" |Select UserPrincipalName, DisplayName

$AADAccounts = @{}

$AADUsers.ForEach( {

$AADAccounts.Add([String]$\_.UserPrincipalName, $\_.DisplayName) } )

# Process the sites

ForEach ($Site in $ODSites) {

If (!($AADAccounts.Item($Site.Owner))) { #Allocate a new owner to the OneDrive site

Write-Host "Adding user to" $Site.URL

$Status = $Null

Try {

$Status = Set-SPOUser -Site $Site.URL -LoginName $NewSiteAdmin -IsSiteCollectionAdmin $True }

Catch {

Write-Host "Couldn't add" $NewSiteAdmin "to" $Site.URL }

If ($Status) { #Update output report file

$i++

$ReportLine = [PSCustomObject]@{ #Update with details of what we have done

Site = $Site.URL

"Previous Owner" = $Site.Title

OwnerUPN = $Site.Owner

"New Owner" = $NewSiteAdmin

LastModified = Get-Date($Site.LastContentModifiedDate) -format g

StorageUsage = $Site.StorageUsageCurrent }

$Report.Add($ReportLine) } # End If

} #End If

} # End ForEach

If ($i -gt 0) {

Write-Host $NewSiteAdmin "added to" $i "OneDrive for Business accounts - details in c:\temp\OrphanOneDrive.csv"

$Report | Export-CSV -NoTypeInformation c:\temp\OrphanOneDrive.csv }

Else {

Write-Host "No orphan OneDrive for Business accounts found" }

If any orphan OneDrive sites are found, the script generates a CSV file. The account added to the sites can be used to access the sites that seem to be of interest to check if any valuable information exists there. If nothing is found, or after anything interesting is retrieved, the site can then be removed.