Everything you need to know about Office 365 Tenant to Tenant Migrations

Authored by Jeff Guillet, Mike Crowley, Gustavo Adolfo Velez Duque, Juan Carlos Gonzalez Martin, and Barbara Waskey.

Edited by Vasil Michev

Introduction

Office 365 has grown tremendously since it was first publicly released in 2011. It is now the de-facto collaboration suite for every organization, be it a small shop or large multinational corporation, a school or university, government or NGOs. Over the last few years, Microsoft has invested a lot in making the collaboration between different tenants easier, introducing features such as external file and folder sharing, sharing of cross-organizational presence information, guest user access, and more.

While these features do make cross-tenant collaboration easier, they are not always enough. The business world is ever-changing, and mergers, acquisitions and divestitures are just a fact of life. The reasons behind those can vary, but the end goal is usually the same – moving users and data between one or more Office 365 tenants. A variation of the Tenant to Tenant migration can also involve on-premises infrastructure or other cloud services. Unfortunately, Microsoft is yet to provide a toolset that addresses the challenges of this process, even though both the services and the underlying infrastructure is under their full control.

Several ISVs have stepped in and released products that can help with one or more of the tasks involved in a typical Tenant to Tenant migration scenario. Quadrotech's product for Tenant to Tenant migrations is called Cloud Commander, and you can learn more about it here. Because of the limited access ISVs have to Office 365, no perfect solution exists today and the tools available on the market often cover just some of the workloads.

Even in scenarios where all the underlying workloads are supported by the tool, the Tenant to Tenant migration process can still pose a challenge. There are many factors that need to be taken into consideration, so detailed planning and preparation is essential for successful migration. To alleviate some of the uncertainties, and better prepare you for this challenging task, we have asked few well-respected MVPs to share their experience and guidance on Tenant to Tenant migrations for this eBook. In addition, the last chapter offers some insight from Barbara Waskey, a certified PMP, put in the context of a recent migration she was involved with.

Jeff Guillet

Exchange MCSM | Office Server and Services MVP | EXPTA Consulting | www.expta.com | JGUILLET@EXPTA.COM

Jeff Guillet is an Exchange MCM and Office Servers and Services MVP for a number of years, with decades of experience in Windows Server and Exchange deployment and management. He is an author of several books and a regular speaker at high-profile conferences such as TechEd, IT Dev Connections and Microsoft Ignite. He is also an author of the EXPTA blog.

Mike Crowley

Mike Crowley is a seven times recipient of the prestigious Microsoft MVP Award for Exchange Server and Office Servers and Services. He has extensive experience in the government, education and private sectors. A certified Microsoft Trainer for a number of years, he also helps others on their path to obtaining Microsoft certifications. You can follow his blog here.

Gustavo Adolfo Velez Duque and Juan Carlos Gonzalez Martin

Veterans in the SharePoint world, Juan and Gustavo have each been awarded Microsoft MVP status for 10 consecutive years. Versed in both developing and administration, they often collaborate with each other as authors of numerous books and blog posts, or presenters at SharePoint and Office 365 events around the world. Gustavo's blog can be found here, and Juan's here.

It's their passion and dedication that makes the publication of the free Spanish digital magazine about SharePoint CompartiMOSS possible.

Barbara Waskey

Barbara Waskey, PMP is a Director for a Fortune 50 company specializing in IT Program Delivery, M&A, Strategy, and Intake. Barbara obtained her PMP certification in 2009. She has been the Project Manager for large and complex Merger and Acquisition programs. Barbara also has extensive experience in data and infrastructure migrations. Previously she worked in the Banking and Insurance Industries. Barbara brings a customer centric philosophy to IT Program Management. Combined with a high performing team approach, this reduces business impact and delivers high value results.

Contents

Introduction 1
Chapter 1: Overview of Tenant to Tenant Migrations 7
Planning the Tenant to Tenant Migration 8
Preparation 9
Record email addresses 11
Record delegates 11
Record MFA settings 11
Record Office 365 licensing 12
Enable auditing 12
Directory Synchronization 12
Mergers and acquisitions 14
Merge two cloud-only organizations 14
Merge a cloud-only organization into a hybrid organization 14
Merge two organizations without a forest trust in place 15
Merge two organizations with a forest trust 15
Divestments 15
Divesting from a cloud-only organization 15
Divesting from a hybrid organization 15
Uniqueness and SMTP domains 16
Migration strategies 17
Migrate >60-day email first, then migrate calendars, contacts, tasks, and delta emails 17
Migrate calendars, contacts, tasks, and email from <30-days first, then backfill older emails 17
Passwords 18
Authentication 19
DNS domains 20
Mail flow 22
Autodiscover 23
Public folders 24
Multi-Geo tenants 24
Communication Plan 25
Consider other workloads that rely on Exchange Online 26

Other Migration Tips and Tricks 26
Use IdFix 26
Cleaning up 27
Summary 27
Chapter 2: Core concepts and specific issues for Exchange and Skype for Business 28
Intended Audience 29
Core Concepts 29
Identity Management 29
AD Users, Mailboxes, Remote Mailboxes & Mail-Enabled Users 29
Active Directory Group Objects 31
Azure Active Directory User Objects 32
Azure Active Directory Group Objects 33
Azure AD Connect (and similar tools) 33
Identity Migration Scenario 34
Email Routing and Namespace Coexistence 35
Office 365 inter-tenant collaboration 36
Transferring Exchange Online Configurations 36
Exchange Online Service Readiness Scenario 36
Endpoint Readiness 37
Mailbox Migrations 37
Mailbox Migration and Cleanup Scenario 39
Documentation 40
Tools 40
Overcoming Specific Issues 40
Azure AD 40
Verifying Domain names 40
Mapping Users to a New AAD Connect Server 41
User Licensing 42
Azure AD RMS has encrypted files in the source tenant 42
Exchange Online 45
'Replyability', Suggested Contacts (Auto Complete Cache) 45
Outlook Profile Management 45
Skype for Business Online 46
Meeting Reoccurrences 46
Migrating Phone Numbers 46
Summary 47

Chapter 3: Migrating SharePoint Online and OneDrive for Business . . . 48
Approaches for a SPO and ODFB Tenant to Tenant migration 49
Third Party Migration tools 53
Criteria for Third-Party SPO and ODFB Tool selection 53
Usability 53
Mapping Features 54
Granular Configuration features. 55
Reporting Features 55
Automation Features and APIs supported 56
Pricing. 57
Other features to consider 58
SPO and ODFB Tenant to Tenant Migration Tools: A quick overview .59
What about other Office 365 services that use SharePoint Online? 60
What about Microsoft? 61
SPO and ODFB Tenant to Tenant migration on your own terms 64
Migrating data between tenants using SPO and ODFB APIs 64
Migrating data between tenants using PowerShell with CSOM API 66
Using the Microsoft Migration API 67
Conclusions 73
Chapter 4: Managing a Tenant to Tenant migration project 74
Story 74
Approach. 76
Communications and Champions. 78
Interviews and Inventory. 78
Education and Engagement 79
Design and Testing 80
Training. 81
Deployment strategy. 81
Conclusion. 83

Chapter 1: Overview of Tenant to Tenant Migrations

Jeff Guillet

Businesses have always changed. Over time, companies merge and separate. In the recent past, these would be treated as domain or forest migrations. There are many products and tools available to help admins with this process, including built-in Microsoft tools such as the Active Directory Migration Tool (ADMT), forest trusts, GAL sync and directory synchronization tools.

The wild popularity of Office 365 has created some interesting challenges for these types of moves. The guaranteed logical separation of business tenants hosted on the same physical platform makes it challenging to combine or separate tenants. There is no notion of a forest or domain trust within Azure Active Directory, and the lack of native migration tools for Tenant to Tenant migrations makes it that much more difficult.

There are several reasons why organizations may wish to perform a Tenant to Tenant migration:

- Mergers and acquisitions
- Divestitures
- Business realignment

Mergers and acquisitions can include migrating objects from one tenant to another (users in the Fabrikam tenant are migrated to the Contoso tenant), or it can involve migrating both tenants to a brand new third tenant (Contoso and Fabrikam users are migrated to a new MegaHoldings tenant).

Divestitures occur when a business unit is spun off as a new company into its own tenant. For example, a company that develops software may want to create a wholly owned and operated service and consulting company.

Tenant to Tenant migrations also may occur when an organization wants to realign the business due to regulatory or data residency reasons. An example might be a North American manufacturing company that has an arm of the business in Germany, where there are more restrictive data sovereignty regulations.

In this document I will cover some important items to consider when performing a Tenant to Tenant migration. I'll discuss planning the migration, directory synchronization, authentication, DNS domains, autodiscover, and mail flow design. Let's get to it.

Planning the Tenant to Tenant Migration

As with any important project, planning the migration is the most important step, and may take the longest amount of time. The more effort you put into planning, the smoother the migration should go and the less impact and surprises for your end users and the business.

You should know all the business and technical requirements before starting this endeavor. Every migration is different so it's impossible to make a completely comprehensive list of questions that should be asked, but some important things to know are:

- What are the business requirements?
- Where are the tenant geos currently located?
- Where will the target tenant be located? Is the goal to migrate A to B, or A and B to C?
- Are there any legal requirements, laws and regulations that affect the migration?
- Who are the key stakeholders?
- What is the project timeline?
- Are there any language, culture, or time zone considerations that will affect the migration?
- What is the expected end state? Operate as one organization, or independent organizations in the same single tenant?
- How will directory synchronization be configured?
- What are the authentication requirements?
- Where are the end-users located?
- What Office 365 licenses will be used?
- Which SMTP domains will be used post-migration?
- How will mail flow work?
- How will distribution and mail-enabled security groups be handled?
  - Consider Office 365 groups
  - Consider Teams
- Which email clients and access protocols will be used?
- Who will manage the migration?
- Who will manage the AD domain(s) and tenant(s) post-migration?
- What is the communication plan?

Before starting, there are two key things to know: the business and legal requirements, and the technical requirements. Sometimes these are at odds with each other. For example, in a global Tenant to Tenant merger there may be a business desire to host all data in a North American tenant, but the General Data Protection Regulation (GDPR) European privacy law may have a major impact on collecting and reporting data.

It is imperative that you know and understand the laws and regulations that apply to data and users involved in the migration. This information will drive some of the answers to the questions above. You may need to meet with business leaders and key stakeholders to explain how this affects the merger or divestment, and consequently the migration itself. Once you know this, you can choose the proper migration tools and strategy.

Preparation

There are several things you should do before you start any part of the Tenant to Tenant migration.

Read through the migration documentation for the migration tool of choice prior to starting the migration. You should have a clear understanding of all the steps of the migration, and the order in which to run them. If you have any concerns or confusion, get it sorted out with the vendor and/or the business. Make sure you have correct contact information for migration tool support, Office 365 support, and key stakeholders. Have phone numbers for each contact, in case email is not available.

Create Administrator accounts in both the source and target tenants for use in the migration. Some migration tools use dedicated migration accounts or may require more than one account in the source tenant to optimize the data throughput. Check with your vendor of choice.

You should create test accounts in both the source and target tenants. In a directory synchronization environment that means creating the account in AD and having them sync to Azure AD. License these accounts to ensure you can access email and other Office 365 services. Be sure to add some email, calendar, and contact items to each account. You can use these test accounts for test migrations.

Ensure the administrator can run remote PowerShell in both tenants.

- Install the **Azure Active Directory PowerShell for Graph** module. See https://docs.microsoft.com/en-us/powershell/module/Azuread/?view=azureadps-2.0 for installation instructions.
- Install the **Microsoft Exchange Online PowerShell Module**. This module includes all the latest Exchange Online PowerShell cmdlets and supports multi-factor authentication (MFA). It can be installed from the Hybrid pane of the Exchange Online Admin Center at https://outlook.office365.com/ecp.

There are two ways that the migration tool can access the source and target mailboxes to perform the migration. Most migration tools use a dedicated service account or accounts for the migration work. Delegation works by assigning full access permissions for the migration account(s) to each mailbox. Normally, this is done by running PowerShell scripts in the source and target tenants. **Impersonation** is an organization-wide elevated access mode which has several benefits for migrations:

- Impersonation does not require setting explicit permissions on the source and target accounts.
- With impersonation, each connection that the migration tool makes uses a separate throttling quota. With delegation, throttling quotas are based on the migration service account. Because of this, impersonation can greatly speed up migration rates.
- Impersonation allows more concurrent migrations.
- Impersonation does not require assigning Office 365 licenses to the migration service account(s).

Impersonation is configured at the organization level in both tenants using remote PowerShell. Here are the commands to enable it in a tenant.

```
Enable-OrganizationCustomization
New-ManagementRoleAssignment -Role ApplicationImpersonation -User <migration service account>
```

You should take an inventory of both the source and target tenants. This can be done using the Office 365 portal, the Azure portal, the Exchange Admin Center, and/or remote PowerShell.

- Record all the accepted domains for each tenant.
- Create a report of all mail-enabled objects in each tenant, including all their email addresses.

- Get mailbox statistics for users that will be migrated, such as number of items and mailbox sizes for each mailbox.
- Create a report of mailboxes that allows other users to access them (delegates).

Having this information can be invaluable later in the migration when you configure the new users in the target tenant. Most third-party tools don't take secondary and tertiary email addresses into account, so you'll need to remember to add those to the target users post-migration.

Record email addresses

The following remote PowerShell example will create a CSV report for all mail-enabled users, contacts, groups, and Public Folders in the tenant. It includes the object's DisplayName, WindowsLiveID (UserPrincipalName, or UPN), RecipientType, and all

```
Get-Recipient -ResultSize Unlimited | select DisplayName, WindowsLiveID,
RecipientType, EmailAddresses | Export-Csv AllEmailAddresses.csv
-NoTypeInformation
```

their email addresses.

Record delegates

The following remote PowerShell script will create a CSV report listing all users who have other users with explicit mailbox permissions to their mailbox. It includes the

```
Get-Mailbox -ResultSize Unlimited | Get-MailboxPermission | where
{$_.IsInherited -eq $false -and $_.User -ne "NT AUTHORITY\SELF"} | select
Identity, User, AccessRights | Export-Csv AllDelegates.csv -
NoTypeInformation
```

identity of the source mailbox, the user(s) who have explicit permissions, and the user's access rights.

Record MFA settings

If multi-factor authentication (MFA) is used in the source tenant, it will need to be reconfigured in the target tenant after the migration is complete. Microsoft Azure offers several reports you can use to view MFA usage within a tenant. Please refer

to https://docs.microsoft.com/en-in/azure/multi-factor-authentication/multi-factor-authentication-manage-reports.

Record Office 365 licensing

Office 365 licensing reports are another important item to review. If the Office 365 licenses change as part of the Tenant to Tenant migration (for example, moving from E5 to E3 licenses), features and functionality will change for the migrated users. You should ensure that the correct licenses are applied to the migrated users, so they continue to have access to the services and features they had in the old tenant. Microsoft MVP Alan Byrne wrote a very useful script to report on Office 365 licenses. Download it from the TechNet Script Gallery at https://gallery.technet.microsoft.com/scriptcenter/Export-a-Licence-b200ca2a.

It's important to note that Office 365 licenses cannot be migrated between tenants. You will need to purchase the correct number of types of licenses for the target tenant to assign to the migrated users.

Users in the source tenant that use Office 365 ProPlus will continue to validate their ProPlus licenses against the source tenant until the DNS domain cutover. After that, they will validate their ProPlus license against the target tenant. If users don't have a valid license for Office 365 ProPlus, it will stop working. This is another reason to ensure that the new accounts have proper licenses in the target tenant.

Enable auditing

Mailbox audit logging records mailbox access by mailbox owners, delegates, and administrators. This is important to enable before the migration so that activity logs are created and retained.

By default, mailbox auditing in Office 365 is disabled. That means mailbox auditing events won't appear in the results when you search the Office 365 audit log for mailbox activity. See this article to read how to enable mailbox audit logging. This article describes how to configure and perform an audit log search.

Directory Synchronization

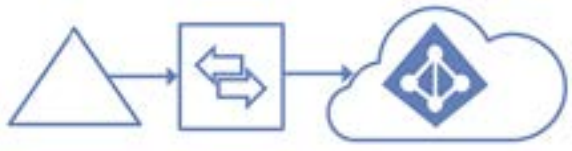
Directory synchronization syncs the on-premises Active Directory with Azure Active Directory in Office 365.

Some organizations are cloud-only, where there is no on-premises Active Directory to sync, and all accounts exist only in Azure Active Directory. Most enterprise organizations use directory synchronization to sync Active Directory on-premises to Azure Active

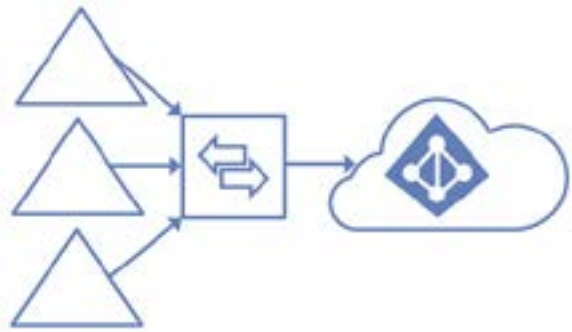
Directory. Directory synchronization is normally achieved using Azure Active Directory Connect (AAD Connect). Azure AD then syncs relevant directory information with the Exchange Online Directory Service (EXODS) for mail-enabled objects within Office 365.

Some organizations use Federated Identity Management (FIM) or Microsoft Identity Manager (MIM) instead of AAD Connect. The complexities of these products are outside the scope of this document, but will need to be considered as part of the migration project.

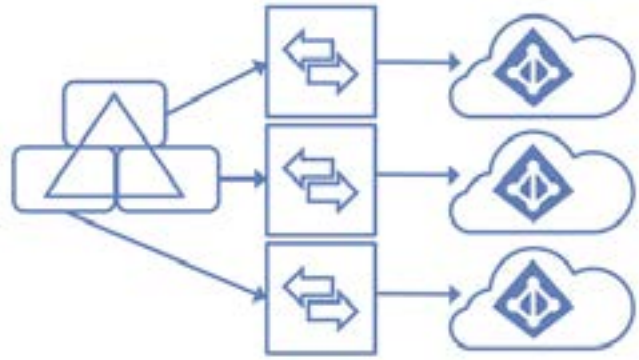
Previous Microsoft directory synchronization products include DirSync and AAD Sync, both of which are no longer supported. It is very important to upgrade older products or builds to the latest version of AAD Connect prior to migration to get the latest sync capabilities and security features. You can download the latest version of AAD Connect



here, and read the latest release notes here.



Azure AD Connect supports several different topologies, including:



- Single forest to a single Azure AD tenant:
- Multiple forests to a single Azure AD tenant:
- Single forest to multiple Azure AD tenants:

Please refer to this article for a review of the topologies supported by Azure AD Connect. Your business requirements will drive the technical directory and synchronization design.

Before starting your migration project, you should understand the current Active Directories and how they sync to Azure AD. Be aware of any custom OU, groups, or attribute filtering being performed by AAD Connect for the organizations.

For a Tenant to Tenant merger you will need to sync objects from both organizations' Active Directory to the same tenant before mailboxes can be migrated the target tenant. This almost always requires a directory consolidation project, which can take quite a lot of time. Be sure to account for this in your project timeline.

The target tenant being migrated to can be one organization's existing tenant or a brand-new tenant. The key thing to know is that each object can exist only once in an Azure AD tenant. See the Uniqueness and SMTP Domains section below.

Mergers and acquisitions
The following migration scenarios describe the directory synchronization requirements and how the mailbox migrations will be done for each scenario. Be sure to read the Uniqueness and SMTP domains and Migration strategies sections below for important information.

Merge two cloud-only organizations
Since there are no on-premises Active Directories in this scenario, you won't be using AAD Connect. You will need to create user accounts in the target tenant and license them, which gives them a new empty mailbox. This can be done manually, one-by-one, or using PowerShell. Microsoft does not provide native tools to migrate mailboxes between tenants, so you will need to use a third-party migration tool to migrate emails to the target mailboxes in the target tenant.

Merge a cloud-only organization into a hybrid organization
A hybrid organization syncs on-premises Active Directory with Azure AD. You will need

to create user accounts for all the cloud users in the hybrid organization's AD so that AAD Connect can sync them with the target organization's Azure AD. This can be done manually, one-by-one, or by using a custom PowerShell script. Once you license the new users to give them a new empty mailbox, you will use a third-party migration tool to migrate emails to the new target mailboxes in the target tenant.

Merge two organizations without a forest trust in place
In this scenario, both organizations have AD on-premises and each one is using AAD Connect to sync with their respective tenant's Azure AD. You will need to perform a directory consolidation to create user accounts for all the source users in the target organization's AD. AAD Connect will then sync those objects with the target organization's Azure AD. Once you license the new users to give them a new empty mailbox, you will use a third-party migration tool to migrate emails to the new target mailboxes in the target tenant.

Merge two organizations with a forest trust
Some organizations may already have a forest trust in place. In this case, there may already be some form of synchronization between on-premises Active Directories – mailbox users in domain B already exist in domain A as mail-enabled users (MEUs), and are synced to Azure AD in the target tenant. When you assign a license to these MEUs, Office 365 will convert them to mailbox users and give them a new empty mailbox. You will then use a third-party migration tool to migrate emails to the new target mailboxes in the target tenant.

Divestments
Another Tenant to Tenant migration scenario is for divestments. This involves spinning off a business unit from one tenant to a brand-new organization in a separate tenant.

Divesting from a cloud-only organization
Divesting from a cloud-only tenant involves creating the new target tenant, creating and licensing the new users in that tenant, and then using a third-party migration tool to migrate emails to the new target mailboxes in the target tenant. Finally, you will need to delete the users and their mailboxes from the source tenant.

Divesting from a hybrid organization
Divestment from an organization that has AD on-premises synced to Azure AD can be more complicated.

If the new organization will be cloud-only with no AD on-premises, start by creating the new target tenant, then create and license the new users in the target tenant to give them new empty mailboxes. Next, use a third-party migration tool to migrate emails to the new target mailboxes in the target tenant. Finally, delete the users and their mailboxes from the source tenant.

If the new organization will have a new on-premises AD, you will need to create the new Exchange-enabled forest, create the divested users in that AD, and configure a new instance of AAD Connect to sync to the new target tenant. Then, use a third-party migration tool to migrate emails to the target mailboxes in the target tenant. Finally, delete the users and their mailboxes from the source tenant.

Uniqueness and SMTP domains

One of the complexities in Tenant to Tenant migrations is that each object can exist only once in an Azure AD tenant. This is due to the multi-tenancy model used in Office 365. Even though users in one tenant cannot see users in another tenant, they all exist within the same Office 365 service, so each object (UserPrincipalName and sourceAnchor/immutableID) must be unique. In other words, you cannot sync the same user to multiple Azure AD tenants.

Likewise, SMTP domains can be associated with only one tenant in Office 365. This causes some challenges for the migration scenarios presented above. You won't be able to assign or sync email addresses with the same SMTP domain for users in a different tenant using the same email address. You'll need to remove the SMTP domain from the source tenant before you can assign and validate it on the new tenant.

Best practice is to make the Office 365 sign-on ID (UserPrincipalName, or UPN) match the user's primary SMTP email address. This greatly improves the single sign-on (SSO) experience and makes the provisioning and configuration of mobile devices easier. You will be unable to do this for the target users until the SMTP domain is removed from the source tenant and added to the target tenant. The primary email address and UPN for the users and groups can be updated to @contoso.com after the domain move is complete.

You might think that removing the SMTP domain from the source tenant and

transferring it to the target tenant would cause an email outage, but Exchange Online Protection will queue incoming emails for up to 24 hours. That should provide enough time to add the SMTP domain to the target tenant and perform domain validation. Once validated, EOP will deliver the emails to the tenant recipients, but you should be aware of the delay it may cause.

Migration strategies

There are several migration strategies that can be used for a Tenant to Tenant migration. The first is a "Big bang" migration. This is usually done for smaller migrations, where most of actual migration work and cutover is done overnight or a weekend. Obviously, this is only suitable for organizations with a small amount of data to migrate.

Enterprise migrations with large amounts of data take longer and usually require some period of coexistence – sometimes for months or even longer. In most projects, one of the project requirements is to limit disruption to users and the business. There are a couple of ways to do this, and the option you choose depends on your timeline.

Migrate >60-day email first, then migrate calendars, contacts, tasks, and delta emails

This is the most common way to migrate mailboxes between tenants. After cutover, users will sign into the new tenant and have access to all their email. However, this method takes the most amount of time before the cutover can occur.

You start by migrating all emails older than 60 days to the empty target mailboxes. If there's a lot of email data, this may take days, weeks, or even longer to complete. Once this is done, you're ready to perform the cutover and complete the migration.

Migrate calendars, contacts, tasks, and email from <30-days first, then backfill older emails

This method is useful when the cutover to the new tenant must be performed as soon as possible. Users have access to their latest emails, contacts, calendars, tasks, etc. right away. Older emails (>30 days for example) are backfilled into their mailboxes after the cutover.

You start by migrating all calendars, contacts, tasks, and the last 30 days of emails to the empty target mailboxes. Since this is a much smaller set of data, it shouldn't take long. Once this is done, you're ready to perform the cutover and complete the migration.

The domain cutover usually entails the following steps. In this example the source domain is fabrikam.com and the target domain is contoso.com.

- Disable AAD Connect in the fabrikam.com source tenant. See this article for details. Once AAD Connect is disabled, you can make changes to fabrikam.com objects in Azure AD directly, instead of on-premises AD.
- Add the fabrikam.com domain as a secondary UPN to the contoso.com Active Directory.
- Add fabrikam.com as an accepted domain on the Exchange hybrid mode server in the contoso.com domain.
- Change the sign-in IDs for all users in the fabrikam.com source tenant to use the service domain (fabrikam.onmicrosoft.com). This will also prevent these users from signing in to Office 365 and changing data during and after the cutover.
- Remove all the fabrikam.com email addresses from all mail-enabled objects. It's important to first record all email addresses, as mentioned earlier. You won't be able to remove the fabrikam.com domain from the tenant until all fabrikam.com email addresses are removed.
- Remove the fabrikam.com domain from the source tenant and add it to the contoso.com target tenant.
- Update the fabrikam.com user accounts in the contoso.com AD to use the fabrikam.com domain for their UPN, add their old fabrikam.com email addresses back, and synchronize with Azure AD.

Once the domain cutover is complete, you do a delta migration of all current email, contacts, calendars, tasks, etc. to the target mailbox to complete the migration. Users can see missing emails filling into their mailboxes while this process runs for hours, days, or weeks.

Since both migration strategies involve creating brand-new mailboxes in the target tenant and copying email from the source tenant to the new mailbox, Outlook clients will need to download that data to a new OST file. Outlook 2016/2013 provide the cached Exchange mode offline settings slider. The slider allows users to decide how much email to download and keep local on their computer. The default setting depends on disk size where the OST is kept, but it's normally set to 12 months for disks greater than 64GB. That means that post-migration, migrated users will download their calendars, contacts, and the last 12 months of emails. Keep this in mind if network bandwidth is at a premium.

Passwords

User passwords are normally not synchronized between the source Active Directory and the target Active Directory during directory consolidation. That means that the passwords on the new accounts in the target directory need to be set to a known value, and the migrated user must change their password at first sign-in.

Ensure that you create a process to give the new temporary password to the migrated users, along with instructions for resetting them, if required. Make sure this is part of your communication plan and that the help desk is prepared to handle support calls for this.

Authentication

There are several ways that users can authenticate to Office 365. Each has its own pros and cons. This document will not go into the details of each, but it's important to understand how each works at a high-level.

- **Cloud-Only Authentication** – Tenants that do not have Active Directory on-premises don't use AAD Connect. They authenticate directly with Azure AD and user accounts and passwords are stored and managed directly in Azure AD.
- **Password Sync** – AAD Connect doesn't actually sync passwords with Azure AD, it syncs hashes of password hashes. Users authenticate to Office 365 using Azure AD, but the passwords are stored and managed in on-premises Active Directory.
- **Pass-Through Authentication** – An on-premises PTA agent checks for and services Office 365 authentication requests. Users authenticate against on-premises AD where passwords are stored and managed. This method is useful for organizations that want to authenticate to Office 365 on-premises, but don't want to implement all the infrastructure required for AD FS.
- **Active Directory Federation Services (AD FS)** – AD FS is a claims-based authentication method used by Office 365 and other cloud services providers. It uses SAML 2.0 and WS-Fed protocols. Users authenticate against on-premises AD and are provided with a claims token. This token is used to access appropriate cloud services. A big differentiator is that AD FS can be used for Office 365 as well as many other cloud services, such as Salesforce or Workday. It also requires a fair amount of on-premises infrastructure – Windows Application Proxy (WAP) servers, AD FS servers, load balancers, and sometimes SQL servers.
- **Third-party authentication** (Okta, OneLogin, etc.) – These are third-party claims-based cloud services that provide a lot of the claims-based features that

AD FS provides, but without the on-premises infrastructure. There are a number of these providers and they all implement their solutions a bit differently. They typically have their own sync agent (besides AAD Connect) and usually offer an application portal website for access to cloud apps.

It's important to know which authentication methods are used by the tenants involved in the migration because they each have their own look and feel that users are familiar with. Since you can only use one type of authentication per Azure AD tenant, you may be required to communicate the changes to users affected by the migration. Be sure helpdesk staff understands the proposed authentication changes, so they can support end-users.

You should discuss authentication requirements with the business. Many organizations have chosen an authentication standard, such as AD FS, or a third-party authentication provider. There may be significant investments made in that provider, so you may not have a choice which to use, but you should know if the authentication standard can support the business needs of both tenants in the merger or divestment.

AAD Connect is used to configure Password Sync, Pass-Through Authentication, and AD FS. You can use it to change between authentication methods without much effort. You should discuss with the business if it makes sense to align authentication methods between the tenants before implementing the migration.

DNS domains

As mentioned earlier, a DNS domain can be registered in only a single Azure AD tenant. Careful planning is required to accommodate these requirements. Switching DNS domains between tenants represents the actual cutover for users.

Namespace planning is an important phase of a Tenant to Tenant migration. You will need to decide what the DNS namespace will be for the target tenant. Will it be one of the same DNS domains that already exist, or a new one?

One thing that remains constant is the tenant service domain namespace (i.e., contoso.onmicrosoft.com). Each tenant has a unique service domain namespace that

cannot change, and every mail-enabled object in Azure AD uses this as an additional SMTP address for internal mail routing. Your migration tool of choice will use these service domains throughout the migration because of this fact. It allows you to move vanity domain names between tenants without disrupting the migration.

As mentioned earlier, it is best practice to make the user sign in address (UserPrincipalName, or UPN) the same as the user's primary SMTP address. This provides a better SSO experience. A tenant can have more than one domain and DNS namespace associated with it. For example, the tailspin.onmicrosoft.com tenant can add the contoso.com and fabrikam.com DNS domains, assuming they can be validated, and do not belong to any other tenant. This provides you with the capability of using either domain for the UPN in the same tenant. You could allow users to sign-in to the target tenant using the source tenant's DNS domain once it has been cutover to the new tenant.

DNS domain cutover always requires some user downtime. You remove the DNS domain from the old tenant and add it to the new tenant. Before the domain can be associated to the target tenant, it needs to be validated. This usually involves registering a TXT record in the DNS domain to prove you are the owner of that domain. How this is done depends on your DNS provider. Some organizations host their own DNS, so it's fairly easy to update records. Others use a hosted DNS service where they need to enter support tickets for DNS changes. In this case, you need to be aware of service times and SLAs. You'll want these changes to be implemented as quickly as possible. Ensure the DNS provider can make these changes at the time you want.

You can speed up the domain transfer process by doing most of it ahead of time. A few days before the planned cutover, add the source domain to the target tenant on the Office 365 admin portal and get the validation TXT record. Enter this TXT record into the source domain's external DNS. This will give enough time for the TXT record to globally propagate, so it's one less thing you need to wait on. Keep in mind that you will be unable to complete domain validation until the domain has been removed from the source tenant.

You can use a global DNS propagation website, such as www.whatsmydns.net, to

confirm that DNS changes, including validation TXT records, are propagated across the world. Keep in mind that Office 365's DNS servers need to pick up the change before the DNS domain can actually be validated. This usually doesn't take more than a few minutes, but it has been known to take over an hour.

Mail flow

A DNS zone's mail exchanger (MX) records are used to tell the world where to send SMTP email for that domain. It is strongly recommended to reduce the time to live (TTL) for MX records prior to migration. I recommend setting the TTL value to 5 minutes, but some DNS providers have a higher minimum limit. This decreases the cutover time when you update MX records during the migration. It's not uncommon for MX records to have a TTL of 24 hours by default, which means it will take up to a day before the cached record expires and clients around the world start sending to the new updated record. Having a lower TTL also means that you can quickly fail back in case there's a mail flow problem.

Most organizations that use Office 365 use their tenant's Exchange Online Protection (EOP) namespace for their MX record. The tenant's EOP namespace typically takes the form of fabrikam-com.mail.protection.outlook.com. EOP processes the email for malware and spam and then delivers the safe message to the recipient.

Some organizations may use a third-party cloud provider as their SMTP gateway. In this case, the MX record will resolve to them. This provider typically scans emails for malware and spam, and might provide other services, such as email staging or journaling. The provider then relays the SMTP emails to EOP for scanning (again) and delivery to the recipient.

Other organizations use centralized mail flow, where their MX record resolves to an on-premises SMTP gateway, such as a Cisco IronPort or Barracuda. Centralized mail flow is used when all inbound and/or outbound email must pass through these third-party systems for processing. The systems may provide critical business functionality, such as advanced data loss protection (DLP) or message encryption functions. Office 365 features and functionality have improved greatly over the past few years and may provide the same or similar features as these products.

In the end, you'll want to carefully review how mail flow works for each tenant to understand the business and technical requirements. Understand the pricing, and decide how mail flow will work with each tenant. The Tenant to Tenant migration may be an excellent opportunity to simplify mail flow and reduce costs.

Some Tenant to Tenant migration projects will involve hybrid customers with Exchange Server on-premises. Carefully review the Exchange on-premises and Exchange Online send and receive connectors to know how mail flow works. Look for partner connectors, and any special authentication requirements, such as mutual TLS. Special attention must be paid to the hybrid configuration during and after the migration project. The Hybrid Configuration Wizard (HCW) will usually need to be re-run for these organizations after the tenant cutover.

Make note of any hub transport rules, white lists, black lists, etc. in the source tenant. These will need to be recreated in the target tenant.

Having a good understanding of each tenant's mail flow is an important part of every Tenant to Tenant migration.

Autodiscover

The Autodiscover web service is used to configure a user's Outlook client and mobile device for connectivity to Exchange Online. Autodiscover is required to configure Exchange Web Services (EWS), a critical part of email connectivity which provides free/busy access and other services. You might be able to get email to work without Autodiscover, but it won't work well.

Each tenant has its own Autodiscover record in DNS, for example autodiscover.contoso.com. This is usually a CNAME record that resolves to the Autodiscover service provider for the tenant. If all users in the tenant are in Office 365, it usually resolves to autodiscover.outlook.com. If the organization is in hybrid mode, it usually resolves to the on-premises Exchange Server. Exchange Server will automatically redirect Autodiscover requests for Exchange Online users to autodiscover.outlook.com.

Outlook users on Windows PCs use the Autodiscover service when setting up Outlook for the first time, and during normal operation for free/busy lookups and to determine if their mailbox has been moved to another database within Exchange Online. Outlook for Mac users constantly depend on Autodiscover, since Outlook for Mac requires the EWS protocol for all connectivity.

Mobile devices that use Exchange ActiveSync (EAS) normally use Autodiscover only

once, when the device is first configured for email. Mobile devices that use Outlook for iOS or Android check Autodiscover more often during normal operation. Most mobile devices will not be affected by a Tenant to Tenant migration unless the user's UPN or password changes. In this case, users may need to reconfigure their email client. This should be part of your test plan and communicated to users prior to the migration.

The Tenant to Tenant migration is a good time to confirm Autodiscover is configure correctly. Review the Autodiscover records in each organization's DNS to know how they resolve and see if they require configuration updates. If both tenants' Autodiscover records resolve to autodiscover.outlook.com, everything should be fine since the service will automatically determine the correct configuration settings for each users' mailbox. You should still be prepared to tell users that they may need to reconfigure email on their mobile devices, just in case.

Public folders

If the source tenant uses Public Folders, these will either need to be migrated separately or eliminated. Microsoft does not have a migration solution, so you will need to use a third-party migration tool for this. Check with the migration tool provider for guidance.

Keep in mind that any mail-enabled Public Folders in the source domain will need to have their vanity domain email addresses removed before you can remove the domain from the tenant.

Multi-Geo tenants

Multinational companies that have offices around the world often have needs to store their employee data at-rest in specific regions to meet their data residency requirements. Microsoft announced Multi-Geo in Office 365 for these customers in September 2017, with general availability and pricing announcements scheduled for the first half of calendar year 2018.

Contrary to some people's thoughts, Multi-Geo is not a performance solution – it's a compliance and data sovereignty solution. Multi-Geo enables a single Office 365 tenant to span across multiple Office 365 datacenter geographies (geos) and gives customers the ability to store their Exchange Online and OneDrive data, at-rest, on a per-user basis, in their chosen geos.

Chapter 1: Figure 1 - Available geos in Multi-Geo. Photo courtesy of Microsoft

Multi-Geo is enabled using AAD Connect and PowerShell, and currently requires a nomination while it is in preview. While Multi-Geo is not necessarily part of a Tenant to Tenant migration strategy, it is mentioned here in this paper to dispel notions that it may play a part in migration performance or endpoints. Of course, things may always change by the time it reaches general availability.

Communication Plan

A communication plan is an important part of the migration project.

You should notify users of the upcoming migration and service changes. Send email to all users telling them the time and date of the migration, and what to expect. Make users aware of how to connect to Outlook on the web with their new sign on information in case they have a problem after migration.

Notify end users to let them know what to expect for their Outlook profile reconfiguration. You should know this when you run your test migrations.

Monitor migration progress with the tools provided by the vendor. Send out periodic progress reports during migration to management and migration team.

Consider other workloads that rely on Exchange Online

Most organizations use SMTP relays to send emails from on-premises application servers and appliances to internal and external users. Examples include cron job servers, copiers, scanners, and printers. You should review these devices to determine how they relay SMTP emails. If they use authenticated Office 365 accounts, they will need to be reconfigured to use new accounts in the target domain.

You may also need to change the endpoint they connect to. You can use tools like www.mxtoolbox.com to find the correct EOP endpoint for the target tenant.

After migration, the nickname cache will have to be cleared on all Outlook clients. See "How to reset the nickname and the automatic completion caches in Outlook" (https://support.microsoft.com/en-us/help/287623/how-to-reset-the-nickname-and-the-automatic-completion-caches-in-outlo) for an automated fix-it-tool that can be run by the end users. Alternatively, you can add the old legacy DN as an x.500 proxy address to all users. This should be included in the email address report you created.

Skype for Business uses Exchange Web Services (EWS) to read contacts and on-line meetings. Post-migration users will need to re-add their contacts (aka Buddy Lists) and may need to reschedule their online meetings. Be sure to include this in your pilot migration test plan.

Other Migration Tips and Tricks

Use IdFix

The Microsoft IdFix Directory Synchronization Error Remediation Tool is used to discover and remediate issues that would prevent AAD Connect from syncing objects properly. You should run this tool to help identify and fix errors in the on-premises

Active Directory prior to migration. Download it from https://www.microsoft.com/en-us/download/details.aspx?id=36832.

Remove Skype for Business licenses from users in the source tenant. This will remove the SIP addresses from those users.

Cleaning up

Tenant to Tenant migrations copy data from the source tenant, it doesn't move it. This is important because it means that the original data is still accessible in the source tenant. There are times when users may report that they think data is missing after a Tenant to Tenant migration. Tenant administrators will need to allow sign-ins for that old account to check. Then they can either reset the password of the user's account in the source domain and sign-in to that mailbox with Outlook on the web using the domain.onmicrosoft.com UPN, or they can give that UPN to the end-user, so they can look for themselves. Often, the data doesn't exist in the source mailbox like the user thought it did, but it's nice to know you can check.

If you allow the user to sign-in to the source mailbox themselves, care should be taken that they do not send any new emails from that account. New emails will go out with the service UPN and replies will go back to that source account which no one normally accesses.

Once the migration is complete and everyone agrees it was a success, the tenant can be canceled and decommissioned, and you can celebrate your achievement!

Summary

Tenant to Tenant migrations can be complex, but careful planning and understanding of the systems involved should reduce most risks.

Understand the business and legal requirements first, then develop a migration strategy. Remember to always gather reports on both tenants before you start the migration – it can be invaluable during the migration.

Always run test migrations to learn what the user experience will be like and create a communication test plan so there are no surprises.

Chapter 2: Core concepts and specific issues for Exchange and Skype for Business

Mike Crowley

Key concepts for Tenant to Tenant Migrations

Microsoft's Office 365 service facilitates migrations from many of their on-premises products, across a variety of technical landscapes, however their investments in "Tenant to Tenant" migrations are limited. For example, there is no "Hybrid Migration Wizard" allowing users to migrate user's email across tenants. Similarly, due to the nature of Office 365 architecture, there is no back-end "easy-button" or script Microsoft can run to move your users into another tenant.

Microsoft's long-time, on-premises customers benefit from the great amount of energy Microsoft has put towards their hybrid (cloud service + on-premises software) capabilities. With Office 365, Exchange mailboxes and Skype for Business accounts can be moved into and out of the service with minimal administrative effort and negligible user disruption. Azure Active Directory is also quite flexible in that it handles a variety of authentication and synchronization scenarios. While different in their nature, services like SharePoint Online, and much of the Azure family of products also have their own Microsoft-supported "hybrid" topologies.

To a lesser but still notable extent, Microsoft advertises hybrid, multi-vendor cloud topologies with other technology companies, such as Box, Gmail, Amazon Web Services and the like.

It is therefore sadly ironic that moving organizations between Office 365 tenants should be so difficult!

If you've done Office 365 migrations in the past, you may need to reset your expectations (and management's expectations). Unlike some migrations, where you barely even have to tell the users they are migrating, a Tenant to Tenant migration will probably require the users to pay closer attention - and in some cases even help out. In addition, it is virtually impossible to complete a Tenant to Tenant migration without the help of a for-purchase, 3rd party migration tool.

Intended Audience

The topics in this book assume you have had at least some experience with traditional on-premises to cloud migrations. Because each Tenant to Tenant migration has unique circumstances, we do not attempt to spell out each step that will be necessary. Instead, we focus on what makes Tenant to Tenant migrations unique, and point out the main areas you'll need to pay attention to as you migrate an organization from an existing tenant another.

Core Concepts

Before we address any specific Tenant to Tenant ideas or migration configurations, it's important to understand how those ideas and configurations work in a normal environment. In some cases, Tenant to Tenant migration topologies scale these configuration areas, and in others, we need to change their default behavior.

This section is not intended to be an in-depth, or even complete discussion on the topics within, but instead we'll shine a light on key concepts from the configuration areas that need the most attention. You will find that the points made here are at the center of the follow-on migration discussions, which is why a refresher is a good idea.

Identity Management

For this book, "Identity Management" refers to the creation and manipulation of user and group objects in the on-premises Active Directory, as well as the user objects in Azure Active Directory(ies) and any tools related to them, such as Azure AD Connect. This may be the most important topic in the whole book, because if you don't get it right, you could cause pop-ups on user computers, disconnect users from their data, or even connect users to the wrong data.

To manage the scope of each topic, we will make the assumption that if you have an on-premises environment, it is based on Microsoft's Active Directory. It is of course acknowledged that other directory services exist, but they are beyond the scope of this book.

It may also be assumed that Exchange Online is part of your Office 365 deployment, as many examples include concepts that apply to Exchange.

Active Directory Users, Mailboxes, Remote Mailboxes & Mail-Enabled Users

NOTE: The Active Directory service contains objects within domains. Domains are part of a forest, which is the outermost boundary to administrative and security configurations.

In truth, there is only one type of user object employed in Active Directory. When Exchange Server refers to different Recipient Types like Mailbox and Mail-Enabled user, etc. it is grouping users based on the value of their *msExchRecipientDisplayType* and *msExchangeRecipientTypeDetails* attributes. The following article includes some information on the possible values of these attributes: How msExchRecipientDisplayType and msExchangeRecipientTypeDetails Relate to Your On-Premises.

These Recipient Types represent a higher layer of abstraction than tools such as the Active Directory Users and Computers MMC console recognize, which is why you're often discouraged from editing recipients outside of the Exchange Server (or SharePoint or Skype for Business, etc.) administrative interfaces.

The Recipient Type attributes are labels that describe the object's behavior, but there are many more attributes involved, governing the specifics. Here are some significant attributes for the main Recipient Types:

UserMailbox

User Mailboxes are the most common recipient in Exchange. Email sent to these recipients is stored in their mailbox, which resides within an Exchange database. The mailbox is tied to their regular user account in Active Directory.

Attribute	*Why it's important*
homeMDB	The homeMDB attribute tells Exchange on which database they can find the user's mailbox. The absence or presence of this attribute will tell you if this user has a mailbox or not. When a user is migrated out of the Exchange Organization, this attribute is cleared on the source Organization and stamped on the target Organization.

MailUser

Mail Users are regular Active Directory accounts, but without a mailbox. These users appear in the Exchange Address Lists for the benefit of other mailbox-enabled users, but mail sent to these recipients is forwarded elsewhere. For example, a consultant for from an outside organization might be given an Active Directory account for accessing a file server, but mail sent to this individual is immediately forwarded on to unrelated environment, such as Gmail.

Attribute	*Why it's important*
TargetAddress	TargetAddress, or the ExternalEmailAddress is the destination email address for a user. Any mail sent to the MailUser object is immediately forwarded along to the value of this attribute.

RemoteMailbox

Remote Mailboxes are essentially Mail Users, however they imply a relationship with an actual Exchange mailbox somewhere else, such as Exchange Online.

Attribute	*Why it's important*
TargetAddress	Unlike the MailUsers scenario, the TargetAddress attribute for Remote mailboxes is represented via RemoteRoutingAddress in the Exchange tools. An example value might look like this: user1@yourtenant.mail.onmicrosoft.com

MailContact

Unlike the previous three recipient types, Contacts are actually a different type of Active Directory object. They are like mail users, with the exception that there is no underlying Active Directory account. These are not security principals, have no passwords, cannot be assigned permissions, etc. They exist for reference purposes only.

Attribute	*Why it's important*
TargetAddress	TargetAddress, aka ExternalEmailAddress is the destination email address for a user. Any mail sent to the MailContact object is immediately forwarded along to the value of this attribute.

During migrations, you may need to use one object type or another depending on the migration stage of a user. For example, a mailbox in a new environment might be represented as a contact or mail-user in the old environment, for the benefit of those still there.

If you'd like to learn more about objects and their attributes, the following article, which maps the Exchange labels to their underlying Active Directory attribute names can be of great help: Active Directory Attributes for Exchange 2007 Mail Objects.

Active Directory Group Objects

There are a few significant ideas concerning groups in Active Directory:

Types: Security vs Distribution

A group can be given a type of *security* or a *distribution* in Active Directory. Only the security type can be assigned permissions to a resource in Active Directory, but both can be used as an Exchange recipient. Be aware however that adding a user to a security group increases the size of their Kerberos token, which can lead to performance issues if

done too frequently. In addition, security groups represent a security principal in Active Directory, and it is best-practice to limit the number of those that exist.

Mail Enabled

Active Directory groups are ignored by Exchange unless they are mail-enabled. You can create a group that is mail-enabled upon creation in the Exchange tools, or you can enable existing AD groups. Mail-enabling a group adds exchange-specific attributes to the group, such as an email address.

Members

The membership of a group is recorded in an attribute called *member*, which is a multi-valued attribute listing the *DistinguishedName* of each member. These entries are automatically updated as necessary if a member is moved or renamed.

It is also worth pointing out that the members themselves have an attribute called *memberOf* which lists all the groups they are a member of. This attribute isn't editable however. It is a special type of attribute called a backlink that is automatically updated when manipulating the group itself. Editing group membership by a user's *MemberOf* tab in the Active Directory Users and Computers tool, is just reaching out and changing the membership of the group; not directly writing to the *memberOf* attribute.

Azure Active Directory User Objects

NOTE: Like an on-premises' Active Directory's forest, Azure Active Directory (AAD or Azure AD) has an administrative and security boundary, called a tenant.

Azure AD has similar object and attribute relationships to those within Active Directory, but it is important to understand that Azure AD is a web service that maps to various Active Directory forests behind the scenes. It is not literally Active Directory, and this means you can get yourself in trouble by making assumptions. For example, Azure AD has an attribute called the *immutableID*, which is not found in "regular" Active Directory. The *immutableID* is the base64 representation of a user's on-premises *objectGUID* attribute, which is maintained by the Azure AD Connect software.

You can explore Azure AD through a variety of interfaces, such as:

- The Office 365 admin portal, the Azure AD Portal, the Azure Portal
- Microsoft GRAPH API
- The Azure AD or Microsoft Online Services PowerShell Module

You can also view user information through the Office 365 products, such as the Exchange Online Admin Center. We mention this separately however because those products talk to a special back-end Active Directory which Microsoft syncs to Azure AD. So, you're not technically looking at Azure AD when you pull up a user object in Exchange Online, though it's the usually the same difference.

Be aware, there are often slight delays between a change in Azure AD and its appearance in Exchange Online (and vice versa), which is important to remember when troubleshooting or writing PowerShell automation scripts.

Azure Active Directory Group Objects
Like the on-premises Active Directory, Azure AD has security and distribution groups. There is also a confusingly named "Office 365 Group" which is an amalgamation of a traditional distribution group, Exchange shared mailbox and a SharePoint team site.

Azure AD Connect (and similar tools)
While some organizations use Office 365 without an on-premises Directory, many have an existing directory service. There is frequently deep integration between Azure AD and the on-premises AD, which means you'll need to figure out how to hand off this integration to the new Azure AD tenant at some point(s) during the migration.

Azure AD Connect is a small piece of software that runs on a server(s) in your on-premises environment. Its primary job is to create and update objects in Azure AD. Active Directory attributes are evaluated and processed through complex logic described in the product's synchronization rules, with the resultant transformations exported to Azure AD, typically every 30 minutes.

Azure AD Connect is a powerful tool and very flexible too. The following article describes all the different topologies that are supported by Microsoft, out of the box: Topologies for Azure AD Connect.

When migrating users between tenants, we will have to respect the rules laid out in this article. For example, did you notice that while multiple tenants are supported, there are large red 'X'es over scenarios that have the same user in multiple tenants? That's a problem we will need to overcome in a future chapter.

Another important thing to understand here is how objects are uniquely identified across the various directories (i.e. by their "anchor" attribute). Microsoft discusses this in the following article: Select how users should be identified in your on-premises directories.

Please do take time to read the above topics. This eBook will build on concepts described throughout, but it doesn't attempt to repeat what Microsoft's already done a great job of documenting.

Several 3rd parties also include directory synchronization tools or services. Consult their documentation or professional services as necessary.

Identity Migration Scenario
Let's summarize some of what we've discussed in this section with a scenario.

Mike's Munchies, a company that produces a variety or delicious candy bars, recently completed a migration to Exchange Online, using Azure AD Connect and the usual Hybrid Configuration Wizard. Business is doing well, and they've decided to get into the popcorn game and have acquired Sarah's Snacks, an organization that has users in a cloud-only Exchange Online deployment. They want to consolidate all services into the existing tenant and completely shut down the tenant for Sarah's Snacks. Management doesn't like the idea of a weekend cutover, so we have been instructed to establish the components for a period of co-existence.

In this scenario, we're dealing with the following types of user objects:

Mike's Munchies
- On-premises Active Directory / Exchange has user objects configured as Remote Mailboxes
- Azure Active Directory / Exchange Online User Mailboxes

Sarah's Snacks
- Azure Active Directory / Exchange Online User Mailboxes

We'll hold off on the email aspect of this scenario for a future section, but from an identity perspective, we'd want to consider:
- Implementing address list synchronization (so that both organizations can see both sets of users in their Global Address List). This means Mike's Munchies' Exchange Online environment will have a mail user or remote mailbox for each not-yet-migrated user in the Sarah's Snacks environment. The inverse may also be desired. This would mean all users from Mike's Munchies would be represented in the Sarah's Snacks environment, along with any already-migrated Sarah's Snacks user.

- We may also want to "normalize" the Mike's Munchies environment, which means there would be on-premises RemoteMailbox users for Sarah's Snack staff as they are migrated. This normalization would allow users to be managed in the same manner, regardless of their company of origin.

The migration tool you use will probably specify the finer points here, such as how precisely this synchronization occurs, or whether these placeholder objects are RemoteMailboxes or MailUsers. Make sure you check your respective documentation.

Email Routing and Namespace Coexistence

As with the case of the abovementioned snack companies, some Tenant to Tenant migrations occur because one organization is acquiring another. In these cases, "namespace coexistence" may not be required, if the old email domain name will be retired. Users can shed the old domain as they are moved to the new tenant.

On the other hand, it is possible that the user needs to preserve the old email address, even if just for a period. This is a problem because Office 365 won't allow a domain to be registered to two tenants simultaneously. We will discuss the specifics of overcoming this in a future topic, but first, let us cover some basics on shared SMTP namespaces.

Any time you have multiple email systems (e.g. tenants) handling email for the same domain name, additional configuration is necessary to ensure mail is routed to the correct server and to avoid creating mail loops. Typically, this is done by ensuring users are reachable by an email address that is unique to each system, even if this isn't the primary address they send from. For example, this is exactly how Exchange Online and Exchange on-premises co-exist. Mail sent to Exchange on-premises is routed to Exchange Online if the object has been converted to a RemoteMailbox (i.e. migrated). When this occurs, Exchange updates the on-premises *RemoteRoutingAddress* during the final part of the migration (e.g. user@tenant.mail.onmicrosoft.com).

With on-premises to on-premises migrations, Mail User or Contact objects on one side map to mailboxes on the other to facilitate the per-user address mappings. This inverse relationship between objects also means users are listed in the Address Lists on both sides, which is nice.

Tenant to Tenant migrations may deviate from this depending on the requirements of your migration tool. For example, it may be necessary for mailboxes to exist on both sides, but with forwarding configured on one of them. Check with your vendor for specifics.

Office 365 inter-tenant collaboration

To minimize disruption during a migration, it's a good idea to migrate users with the other members of their department/agency/business unit. Collaboration features may have limited capabilities when users aren't in the same tenant, and intra-tenant collaboration tends to be the most critical.

There are however inter-tenant collaboration capabilities, such as calendar sharing and the availability service. This functionality is accomplished with Organization Relationships, which Microsoft discusses here, alongside non-Exchange considerations, such as Tenant to Tenant Skype and OneDrive sharing: Office 365 inter-tenant collaboration.

Transferring Exchange Online Configurations

Exchange Online recipient information was already addressed in the "Identity Management" topics, however there may be a number of email service related configurations in the source tenant that need to be preserved in the target. For example, these types of configurations should be evaluated:

- Accepted domains
- Anti-spam/malware settings
- Transport rules and Connectors
- Retention policies and tags
- Hold settings
- OWA Mailbox Policy and Mobile Device Mailbox policies

Microsoft has published an article on this topic, including some PowerShell code-samples, which can be found here: Move domains and settings from one EOP organization to another EOP organization. Migrations are also a great time to rationalize old configurations and "house clean" as appropriate. You may not want to move every configuration to the new environment.

Exchange Online Service Readiness Scenario
If we think back to Mike's Munchies' scenario, we'll want to make sure the target environment has the proper controls and mail flow capabilities established before we start migrating users. For example, if there are any IP allow/block lists, or perhaps popcorn recipe DLP templates need to be uploaded and added to transport rules to ensure their data is protected.

We may also consider moving any lesser-used accepted domains from the Sarah's Snacks tenant over to the Mike's Munchies tenant. Combined with the newly minted RemoteMailbox placeholder objects, mail sent to the old domains would align to these new objects in the Mike's Munchies tenant, and mail would be routed back to the Sarah's Snacks tenant. As mentioned earlier each RemoteMailbox would need its *RemoteRoutingAddress* attribute populated the Sarah's Snacks' tenant-specific routing domain (user@SarahsSnacks.mail.onmicrosoft.com) which is typically automated by a migration tool.

Endpoint Readiness
If your users are already connected to Office 365, there are probably no extra "readiness" tasks you'll need to perform, but as mentioned earlier, migrations are a great excuse to clean up and ensure your users are connecting with the latest software, using the most recent protocols. For example, in late 2017, the RPC over HTTP protocol reached end of life in Exchange Online. Microsoft provides a report in the Office 365 admin center, which you can use to determine users that may still be connecting with this protocol and take appropriate action. More information on that can be found in the following article: RPC over HTTP reaches end of support in Office 365 on October 31, 2017.

One other thing to remember is that new tenants may have different authentication settings, depending on if/how you've configured the Exchange Hybrid environment. To keep things simple during the migration, you'll probably want to mirror the settings. More information is provided in the following articles: Configure OAuth authentication between Exchange and Exchange Online organizations and A new architecture for Exchange hybrid customers enables Outlook mobile and security.

Mailbox Migrations
Typically, you will use a 3rd party product to migrate mailbox content from one tenant to another. If the migration tool you have chosen supports Tenant to Tenant migrations via the Mailbox Replication Service (MRS), it may work much like a on-

premises Hybrid migration. This is also the scenario we are discussing with Mike's Munchies. On the other hand, some migration tools use protocols like Exchange Web Services (EWS).

MRS is a better protocol for mailbox migrations, whereas EWS is a client-access protocol, which can be adopted for migrations. In the EWS case, a target mailbox needs to be present during the migration, since the tool is basically "uploading" messages into this mailbox based on what it sees in the source. In addition to the performance implications (EWS is slow) this complicates the address list synchronization story because mailboxes want to collect mail, not forward it. This can be overcome by shuffling attributes between multiple objects through automation, but again, consult your vendor's documentation on specifics.

If you're using an EWS based migration tool, the migration flow might look something like this:

1. Pre-stage AAD accounts and mailboxes in the target environment. You may need to hide these or otherwise ensure they don't receive new content if there are other users working in the target environment.
2. Use the migration tool to sync 95% of the user's mailbox content to the new tenant. Run "delta" syncs periodically until the mailbox is ready to be cut-over, which will likely be coordinated with the other Office 365 workloads (e.g. OneDrive, Skype for Business Online).
3. Cut over user mailboxes.
   a. Old (source) mailbox is removed and replaced with a mail-enabled user, contact, or forwarding is configured with the new tenant's onmicrosoft.com address as the *ExternalEmailAddress*. This ensures the user remains in the GAL for the benefit of those not yet migrated, and also ensures that mail sent to this user on the old address still reaches the user.
4. Update desktop profiles. Your migration tool may have a hyperlink for users to click, or package could be deployed to reconfigure the user's Outlook profile.
5. On-premises Active Directory attributes updates as necessary.
6. Users may need to sign out and back into Skype for Business Online, reconfigure the OneDrive client, using new credentials.

It's worth pointing out, you could also find yourself using an IMAP-based migration tool. Microsoft even provides one for free. You can read more about it here: [How to migrate mailbox data using the Exchange Admin Center in Office 365](#).

These are the least sophisticated tools around and usually only get mail items (calendar items, contacts, tasks, etc. are skipped). Other than that, they would potentially have similar capabilities to EWS migration tools.

Mailbox Migration and Cleanup Scenario
Again, the EWS/IMAP scenarios above differ from the story of Mike's Munchies and Sarah's Snacks. When we use MRS, the "target mailbox" is represented as a MailUser/ RemoteMailbox until MRS is ready to complete the move request for that specific user. Once a user is ready to go, the user will simultaneously become a mailbox in Mike's Munchies and a RemoteMailbox in Sarah's Snacks. MRS handles this conversion, and no manual attribute updates are required.

As recommended in an earlier section, mailboxes from Sarah's Snacks are migrated to Mike's Munchies by department. Their desktops are automatically reconfigured by the native MRS behavior/and the desktop reconfiguration elements of the 3rd party migration tool.

Assuming migrations are going well, you'll have other operational considerations during the project which will need to be addressed. For example, let's say that new hires joining the popcorn division will be provisioned into the Sarah's Snack tenant, unless their target business unit has already been migrated, in which case, we'd create them like any other new user within Mike's Munchies (represented via an on-premises remote mailbox, and an Office 365 Mailbox). We will however want to make sure the place holder object is added to the inverse environment as appropriate.

We may also need to discuss the management of groups. Groups that remain in the Sarah's Snacks environment will continue to function, even if they contain a mix of Mailbox and RemoteMailbox objects, however the end-user manageability of these groups will be limited if the group isn't in the same Exchange environment as a given user. We'll migrate department-specific groups with their respective department and leave company-wide groups for the end. Besides, many of the company-wide groups in Sarah's Snacks will be redundant in the Mike's Munchies environment (e.g. All-employees, HR, etc).

Once the last group is migrated, we will remove the primary domain name from the source tenant and add it to the Mike's Munchies tenant. While the old MX record may continue to function, the best practice is to update it with whatever value is presented when the domain is added to the Mike's Munchies tenant.

Documentation

There are some notable articles on the concept of Tenant to Tenant migrations on the web. You are encouraged to check them out, though Microsoft's contribution may prove to be overly simplistic:

- 5 Stages to Consider for a tenant-to-tenant Office 365 migration
- How to migrate mailboxes from one Office 365 tenant to another
- Office 365 to Office 365 Migration Guide - While Keeping the Same Domain Name
- Insert your migration tool's literature here!

Tools

As mentioned previously, migration tools are necessary for most Tenant to Tenant migration scenarios. Some of the well-known players in this space provide such tools, listed below:

- Quadrotech's Cloud Commander
- BitTitan's MigrationWiz
- CodeTwo Office 365 Migration
- Cloudiway's Office 365 Tenant Migration
- Binary Tree's Power365

Overcoming Specific Issues

Once you've read through the core concepts section as well as the various recommended reading articles, you're ready for some specific pointers, which are listed below.

Azure AD

Verifying Domain names

You cannot register the same DNS name in multiple Active Directory tenants simultaneously. This means that user objects in the destination environment will receive new *UserPrincipalName* (UPN), *Mail*, and *SIP addresses* attributes upon arrival. You will need to make some configuration changes in the old tenant to route email and other content over to the destination environment during the coexistence period.

3rd party identity providers may be able to smooth this process out for your users. For example, if users login to this 3rd party, there could potentially be a mapping within that service to translate the user's UPN. If you go this route, you'll want to pay close attention to the specific capabilities of these services to understand what the user experience will be across the various client applications and Office 365 services. For example, there may be a loss of SSO during certain points of the migration, or some email clients may not support this type of abstraction at all.

Mapping Users to a New AAD Connect Server

Assuming you have an on-premises Active Directory, you're likely synchronizing it to Office 365 with Azure AD Connect. As mentioned in an earlier section, a single user cannot be synced to two tenants from one forest simultaneously. You can however deploy Multiple AAD Connect servers to sync each tenant, provided you ensure there is no overlap in objects. AAD Connect can filter users by OU, group membership or user attribute. Pick the option that best suits your situation and ensure users are inversely excluded from each AAD Connect Server. Azure AD Connect filtering is discussed in the following article: Azure AD Connect sync: Configure filtering.

In addition to the mutual exclusivity requirement, your migration tool will probably copy user data to the target tenant in advance of the actual user cut-over (see the earlier comment about MRS vs IMAP/EWS). These target Azure AD objects don't yet have an *immutableID* attribute populated and therefore aren't yet mapped to the on-premises Active Directory. If your migration tool doesn't handle this aspect directly, you will want to pre-populate the *immutableID* attribute for those Azure AD Objects so that they immediately match up to the correct users in the on-premises Active Directory. Specifically, the *immutableID* is the Base64 equivalent of the users Active Directory Object GUID. If necessary, you can translate these values back and forth with PowerShell. Here is a code sample demonstrating the idea:

To ImmutableID

```
[guid]$guid = "c19936b8-e7b1-452a-9699-f60d3779ef55"
$ImmutableId =
[System.Convert]::ToBase64String($guid.tobytearray())
$ImmutableId
```

To GUID

```
$ImmutableId = "uDaZwbHnKkWWmfYNN3nvVQ=="
[guid]$ByteArray =
[System.Convert]::FromBase64String($ImmutableId)
$ByteArray
```

Azure AD will not allow you to modify this property if the user is currently being synchronized. Ensure these attributes have the correct values prior to implementing AAD Connect. If you need to get around this limitation on a case by case basis, one trick is as follows:

1. Pause AAD Connect by launching the desktop shortcut.
2. Delete the AAD user via Remove-MsolUser cmdlet. The user is now in the AAD Recycle Bin. Do not delete them from the recycle bin.
3. Restore the AAD user via Restore-MsolUser cmdlet. The user is now restored, with the immutableID attribute unlocked.
4. Set the attribute to the desired value via Set-Msoluser.

If your source environment is using linked mailboxes, or if you're also coordinating inter-forest Active Directory migration, your project is going to be a lot more complicated. Be aware however that you're able to override the Object GUID to Immutable ID mapping with the *msDS-ConsistencyGuid* attribute. For more information on this topic, see the following article: Using msDS-ConsistencyGuid as sourceAnchor.

User Licensing
You will likely need to contact Microsoft billing support in order to transfer your licenses to a new tenant. Microsoft has a business process that adds your existing licensing to the target tenant without removing them from your old tenant for a short duration (e.g. 30-90 days).

Aside from the business aspect of getting the licenses into the target tenant, you will need to map them to users. Your migration tool may handle this, or custom PowerShell scripting might be necessary. Though still in preview at time of writing, you may also consider leveraging Microsoft's group-based licensing feature. You can read about this feature here: Group-based licensing basics in Azure Active Directory.

Azure AD RMS has encrypted files in the source tenant
If the source Office 365 tenant employed the use of Azure AD RMS (now a component of Azure Information Protection) to protect user email or documents, you will want to take great care to ensure you don't lose the ability to access this data. Accessing content protected by RMS requires a "call home" to fetch a decryption key, therefore if the source tenant's RMS is deactivated prematurely, this data is lost forever.

Do not deactivate the RMS protection in the source tenant. Even if the related RMS licenses have expired or moved to another tenant, deactivating this will throw away your much-needed tenant key.

Before we get much deeper into this section, be aware of the following response to a support ticket opened in Feb. 2018:

Question	Is it possible to have Microsoft migrate Azure RMS settings and/or keys from one tenant to another?
Answer	"You can migrate from AD RMS to Azure RMS but if you are planning to switch from one Office 365 tenant to another and looking to switch the Azure RMS settings to another, it's not an option for now. You will have to do the entire setup again on the new Office 365 tenant including purchasing of licenses on the new tenant."

As you can see, we'll have to go at this alone. Fortunately, there are some options. But first, a quick note for enterprise customers with extensive or complex dependencies on Azure RMS.

As you can see above, Microsoft has not published any documentation on transitioning Azure AD RMS from one tenant to another. Though there are articles on topics such as *Migrating from AD [on-premises] RMS to Azure Information Protection* which might provide a good overview, as well as instructions for exporting the tenant key (through a paid support case) and importing a key through their Bring your Own Key (BYOK), this author cannot verify a complete migration is possible. If the source tenant has extensive RMS-protected content, it would be worth a phone call with your Microsoft Technical Account Manager to review potentially un-published options.

If, on the other hand, your Azure RMS utilization is of low to medium complexity, you can use Microsoft's Azure Information Protection PowerShell cmdlets to remove the protection templates in bulk, and then re-encrypt them again with PowerShell or instead use Microsoft's new Azure Information Protection Scanner service.

The below code samples will demonstrate the general idea, however if you want to decrypt files beyond those you yourself have personally protected, you'll need to first establish the prerequisites, such as "super user" accounts. You may also need to toggle back and forth between tenants using service principals on both sides. These concepts are detailed in the following article: Admin Guide: Using PowerShell with the Azure Information Protection client.

Once the prerequisites are out of the way, you can proceed with the file transformations. One way to go about this would be as follows:

Activate and configure rights management in the new tenant

Enable the Azure information protection service in the new tenant and configure policies to match the source environment as necessary (using Get-AadrmTemplate/ Export-AadrmTemplate if you have a lot of templates). Instructions for this task can be in this article: Activating Azure Rights Management.

It is also suggested that you limit the users involved, until you are ready to roll it out more broadly. The above article also demonstrates this with the Set-AadrmOnboardingControlPolicy cmdlet.

NOTE: As of February 2018, this is enabled by default for new tenants. You'll want to check your tenant before you proceed, to see if RMS is already in use.

Identity the files that are protected by AAD RMS

Option 1: PowerShell

Open PowerShell and navigate to a root-level directory that contains protected content. Then, use the following sample to list the status of the protected files.

Group files according to RMS template used

```
$ProtectedFiles = Get-ChildItem -File -Recurse | Get-AIPFileStatus | Where-Object IsRMSProtected
$ProtectedFiles | Group-Object RMSTemplateName | select Name, Count
```

Option 2: AIP Scanner

The AIP scanner has some setup requirements which are described in the documentation: Deploying the Azure Information Protection scanner to automatically classify and protect files.

Once setup, run a scan in discover mode. There is a video tutorial on this topic here: https://youtu.be/UNN6FdDehVo?t=26m53s

Remove RMS protection from the files

Before you remove protection, you should document the previous file to template relationships. It is also advisable to work in small phases; perhaps one directory and one template at a time.

For example, if we are about to remove protection from the files with the Highly Confidential \ All Employees template, using the earlier Get-ChildItem results, the following command could be used to place a configuration backup on the desktop:

XML Backup

```
$HighlyConfiedntial | Export-Clixml $env:userprofile\desktop\HighlyConfiedntial.xml
```

Once you've done this, the following commands would remove the protection, overwriting the files in-place:

Remove Protection

```
$HighlyConfiedntial | foreach {Unprotect-RMSFile -File $_.FileName -InPlace}
```

NOTE: If you want to move the unprotected files to an intermediary location instead, use the -OutputFolder parameter instead of -InPlace.

To protect the files with the new tenant's templates, copy the xml file and run the following commands from a workstation logged into the new environment, or toggle to the new environment from the existing computer as described in the prerequisites article above.

Re-protect with target environment

```
$HighlyConfiedntial = Import-Clixml
$env:userprofile\desktop\HighlyConfiedntial.xml
$HighlyConfiedntial | Protect-RMSFile -File $_.FileName -
TemplateID 33ba341f-3394-41e1-9828-21d2fa9210d0 -InPlace
```

Exchange Online

'Replyability', Suggested Contacts (Auto Complete Cache)

If your migration process does not bring over the old x500 addresses for each user, group or other recipient, you will find that old emails cannot always be replied to. Users receive a Non-Delivery-Report (NDR) stating the "IMCEAEX" address cannot be resolved. In addition, the Suggested Contacts (formally Auto-Complete) feature will suggest a user by this now-invalid address; again, causing NDRs. The simple answer is to ensure these *proxyAddress* entries are migrated.

If choose not to do this, or didn't remember to do it for a given recipient, you can reverse engineer the IMCEAEX address in a bounce back message forwarded to you by your users. This article discusses how to decipher these: Convert NDR to LegacyExchangeDN / x500 Custom Address.

Outlook Profile Management

Most of the major migration tools have a method for updating the user's Outlook profile once the mailbox has been migrated to the new tenant.

If you choose to address this without a tool, you might consider a PRF file (if you're using Outlook 2013 or earlier). The more modern approach however is to use ZeroConfigExchange (ZCE) to create a new Outlook profile and the transition users to it. If there are custom themes, signatures or other profile-specific configurations, they will need to be addressed outside of ZCE. For more information on this approach, please see the following article: ZeroConfigExchange – Automating the Creation of an Outlook Profile for Exchange Online Accounts and Exchange On-Premises Environments.

Skype for Business Online

Meeting Reoccurrences

Users may have scheduled Skype for Business/Lync meetings in Outlook that now need a different URL after the Tenant to Tenant migration is completed. If the user's SIP address changes as part of this migration, they will need to recreate their meetings, unless you've acquired a migration tool that addresses this problem.

On the other hand, if the user's SIP address is going to be restored in the new tenant, wait until that happens, and then have them run the Skype for Business Online Meeting Migration Tool from their desktop. This will enumerate existing meetings and replace them with the correct URLs. You can read more about this in the following article: Meeting Update Tool for Skype for Business and Lync.

Migrating Phone Numbers

If you acquired phone numbers through Skype for Business Online, you will likely want to port those numbers over to the new tenant. At time of writing, there is a Microsoft article claiming that this cannot be done "outside the US", but doesn't explain how it can be done by at least US customers. Here is the link to the support article, should they ever update this topic: Can't transfer numbers from one Office 365 tenant in your organization to another in Skype for Business Online.

To port the numbers to the new tenant, you'll need to open a support ticket from within the old portal (where the numbers currently reside). You then must email ptn@microsoft.com with your case number and explain your intentions. They will ask for source and destination tenant GUIDs and the specific telephone numbers to be ported (you don't have to do all of them at once). They also require screenshots proving you own the environments and possibly other information. In this author's experience, it seemed like very immature process, so you should contact Microsoft at the address above for steps specific to your migration.

Summary

Compared to decades of on-premises tools and methodologies, Tenant to Tenant migrations may seem new and scary, but really they aren't! By its very nature, Software as a Service is limited, and as such migrations have a maximum limit to how complex they can be. We also know that the "as a service" part means that both source and target environments are well-maintained, and we don't have to worry about sizing or architecture. In addition, as the service is newish, Office 365 hasn't had decades to collect dozens of layers of configurations, left by generations of admins.

To be successful with Tenant to Tenant Migrations, you just need to understand your migration tool and its limitations very well and set expectations accordingly. Hopefully this "look under the hood" shows you that Tenant to Tenant migrations are mostly just a new way to employ technologies and concepts you already knew.

Chapter 3: Migrating SharePoint Online and OneDrive for Business

Gustavo Adolfo Velez Duque and Juan Carlos Gonzalez Martin

SharePoint Online (SPO) and OneDrive for Business (ODFB) are part of the core workloads present in Office 365, providing the ability to not only to store documents and information, but also to build modern workplace solutions and services on top of both services. Similar to the other Office 365 core workloads (Exchange Online and Skype for Business), we have the following common scenarios when considering the migration of SPO and ODFB information and contents structures between two or more Office 365 tenants:

- **Divestiture scenario**: A Business Group that currently uses Office 365 decides to disincorporate one or more companies that are part of the Group and each of these companies wants to keep using Office 365. They need to move all the SPO and ODFB data from the Business Group Office 365 tenant to the new subscriptions created for each company.
- **Merger or acquisition scenario**: Two or more companies already using Office 365 are integrated into a new company that will also use Office 365 (one of the existing Office 365 tenants or a brand new one). Under this scenario, it might be required that they migrate existing SPO and ODFB contents and information structures from the source Office 365 tenants to the definitive tenant.
- **Change of brand name scenario**: A company already using Office 365 decides to change business name and migrate all the data to a new Office 365 tenant that uses the new brand name.
- **Change of data residency scenario (change of data center)**: A company based on one continent, using an Office 365 provisioned to the corresponding datacenter region, decide to move their headquarters to another continent so they are required to provision a new Office 365 in a local datacenter due to compliance regulations. Under this scenario in which geo-replication is not an option, the company will need to migrate SPO and ODFB data to the new Office 365 tenant.

- **Common requirement to split data geographically**. Over the last few years, several countries in the world have passed legislation requiring that the data generated in the country, must stay in the country. In this case, the information in the main tenant should be identified and moved to the local tenant(s).
- **Data and contents structures migration scenario from a pre-production/ staging tenant to production tenant**: A company started to use Office 365 in a pre-production/staging/test tenant and needs to migrate data and contents structures to the definitive Office 365 subscription.

Approaches for a SPO and ODFB Tenant to Tenant migration

We can consider two main approaches when migrating SPO and ODFB information and content structures between two Office 365 tenants:

- **Manual approach**: Information and content structures are moved from the source to the target tenant following manual steps. This approach is time consuming and error-prone.
- **Automatic approach**: Once some configuration steps have been taken, information and contents structures are moved between tenants without the need of human intervention. This approach implies the use of a third-party migration tool or having a team of migration engineers that will design and build the required migration solution.

For each of these approaches we have one or more migration techniques (Table 1) that can be used in a SPO and ODFB Tenant to Tenant migration.

Migration Approach	Migration Techniques
Manual Migration	• Technique #1: Manual download and upload of the files and folders that need to be migrated. Download and upload operations are done by means of a modern Internet browser. • Technique #2: Local synchronization of the files and folders to be migrated using the ODFB Sync client and copying them to the target site / or ODFB using the ODFB Sync client.
Automatic Migration	• Technique #1: Migrate information and content structures using any of the Office 365 APIs available for migration scenarios. Those APIs can be used on a custom .NET solution or in a PowerShell script. • Technique #2: Move information and content structures between tenants using a third-party migration tool.

Chapter 3: Table 1.- Migration Techniques for each migration approach described.

Each migration technique has its advantages and disadvantages. Some of them are reflected in Table 2.

Migration Technique	Advantages	Disadvantages
Manual Migration Approach		
Manual Download / Upload	• Migration can be done without investing in a third-party tool or a custom solution.	• Document versions and metadata are not preserved. • File permissions are not migrated. • It's a time-consuming technique. • Lists and list items cannot be migrated - only document libraries. • Contents structures (Site Collections | Sites | Lists and Document Libraries) must be created first.

Migration Technique	Advantages	Disadvantages
ODFB Sync Client	• Migration can be done without investing in a third-party tool or a custom solution.	• Document versions and metadata are not preserved. • File permissions are not migrated. • This technique can consume a considerable part of the available bandwidth. • It's a time-consuming technique. • Contents structures (Site Collections | Sites | Lists and Document Libraries) must be created first.
Automatic migration approach		
Office 365 APIs	• Migration can be done without investing in a third-party tool or a custom solution. • It allows you to preserve document versions and metadata. • File permissions can be migrated. • Content structures can be recreated as part of the developed migration solution.	• A good knowledge of Office 365 APIs is required in order to build a custom migration solution. • A good knowledge of SPO and ODFB is required. Various particularities must be considered when designing and building the migration solution, for instance, the throttling performed by SPO and ODFB in scenarios with a lot of requests to both services. • Developing a custom migration solution requires significant time investment. • It may be necessary to hire services of an Office 365 Developer to code the migration solution. • Requires intensive testing to ensure the quality of the migrated data.

Migration Technique	Advantages	Disadvantages
Third-party migration tools	• A third-party migration tool has been specifically designed and built to simplify SPO and ODFB migration processes. • A third-party migration tool is supposed to use Office 365 APIs following best practices and development patterns. • The solution will preserve document versions and metadata. • File permissions can be migrated. • It provides the ability to re-create content structures. • A third-party migration tool is ready to handle SPO and ODFB particularities such as the built-in throttling mechanism. • Special SharePoint functionality, such as Workflows for example, can also be migrated.	• Budget is required to buy the third-party migration tool. • Some training on how to use the tool can be required. Alternatively, a migration consultant should be hired for the migration using the third-party tool.

Chapter 3: Table 2.- Advantages and disadvantages of each migration technique.

Although it's usually recommended to use an automatic migration approach for a SPO and ODFB Tenant to Tenant migration, there may be scenarios where a manual approach for migration is not feasible. Table 3 demonstrates when to use each approach depending on the migration scenario.

Criteria	Manual Approach	Automatic Approach (Custom Solution)	Automatic Approach (Third-party tool)
No budget restrictions		✔	✔
Budget restrictions	✔		

Budget restrictions, but there are resources (developers) available for the migration		✔	
Time restrictions for doing the migration			✔
No time restrictions for doing the migration	✔	✔	
Migration errors and issues should be minimized			✔
Batched migration is possible and easy to setup		✔	✔
Migration of metadata, document versions and permissions are not required	✔		
Migration of metadata, document versions and permissions are required		✔	✔
Migration of Lists is required		✔	✔
Migration of special SharePoint instruments (Workflows, Managed Metadata, etc.)		✔	✔

Chapter 3: Table 3.- When to use each migration approach.

SPO and ODFB Tenant to Tenant migration using Third Party Migration tools

Using a third-party migration tool is probably the most common approach when dealing with a SPO and ODFB Tenant to Tenant migration under any of the scenarios described in the introductory section. This is not only because the migration process can be much simpler, but also because these tools are designed and built to deal with the specific considerations present when moving SPO and ODFB contents between Office 365 tenants. The challenge here is not just to migrate the data using a third-party tool, but also to select the appropriate tool for your migration project. This is not an easy task - a quick search for SharePoint and ODFB migration tools using your favorite Internet search service will produce several pages with well-known, not-so well-known, and even unknown tools capable of performing this kind of migration under any scenario. A good summary of the available migration tools can be found in

the article here, published on the "Top SharePoint" website.

Due to the wide range of choices out there, an important activity that must be done in the planning stage of any SPO and ODFB Tenant to Tenant migration is to decide on the right tool for your migration. Some of key points to consider during the selection process are explained in the following sections of the chapter.

Criteria to select a Third-Party SPO and ODFB Tenant to Tenant Migration Tool
In this section, we will reflect on some of the key aspects and criteria to keep in mind when choosing a third-party tool to move SPO and ODFB contents between two or more Office 365 tenants.

Usability
The first thing to consider when selecting a SPO and ODFB Tenant to Tenant migration tool is the usability and simplicity of the solution. The ideal solution should have the following capabilities:

- The user experience should be intuitive, and the learning curve required to use it almost nonexistent. In this way, people in charge of the migration will find the tool easy to use from the very first moment.
- Tool configuration options in regards of security settings, required mappings, etc., are almost self-explanatory.
- The tool covers different migration scenarios in a simple way:
  - Copy files and folders from the source tenant to the destination tenant.
  - Copy lists / document libraries from the source tenant to the destination tenant.
  - Copy required users and groups from the source tenant to the destination tenant.
  - Copy full site collection and sites from the source tenant to the destination tenant.
  - Can copy special SharePoint elements such as Workflows, Managed Metadata and specific SPO settings.

Mapping Features
When you migrate SPO and ODFB contents and information structures from one Office 365 tenant to another, it is important that you plan to copy files or re-create those structures, but also keep the following settings:

- SPO Site templates used at the source tenant should remain the same at the target tenant.
- Existing Site Columns and Content Types that need to be replicated on the destination tenant due to business requirements in terms of the metadata used in the files that are going to be moved.
- SharePoint Groups, permissions levels existing in the source SPO Site Collections

```
$sUserName="<Office365_Admin_User>"
$sMessage="Type your Office 365 Credentials"
$O365Credentials = Get-Credential -UserName $sUserName -Message $sMessage
$SPOAdminUrl="https://<SPO_Admin_Center_Url>"
Connect-SPOService -Credential $O365Credentials -Url $SPOAdminUrl
Set-SPOTenant -SpecialCharactersStateInFileFolderNames Allowed
```

and Sites that are going to be recreated on the destination tenant.

- Users or groups (SharePoint groups or Security ones) who created and modified the contents in the source tenant that must be kept in the target tenant.

Therefore, your chosen tool for a SPO and ODFB Tenant to Tenant migration should provide features such as:

- **Site templates mapping between tenants** - so that the migration process can recreate sites based on the same (or a different one) site template in the destination tenant.
- **Content Types mapping between tenants** - in such way that metadata of the items and document remain consistent and associated to the lists and document libraries recreated in the destination tenant.
- **User and groups (SharePoint groups or Security groups) mapping between tenants** - to ensure that the version history of the items and documents remain consistent and associated to the corresponding users and groups in the destination tenant.
- **Permissions levels mapping** - so that permissions levels at the source tenant can be mapped with the permissions levels created at the target tenant.

Granular Configuration features

A comprehensive SPO and ODFB Tenant to Tenant migration tool should allow you to perform granular configurations such as:

- Setting the number of versions kept in the documents and list items that are going to be migrated.
- Keeping or removing custom permissions applied at different scopes (Site | List | Document Library | Document | List Item).
- Copying any workflow associated with the lists and document libraries being migrated.
- Creating different mapping templates (Users and Groups | Sites | Permissions levels) that can be reused across the identified migration scenarios.
- The use of an Azure Storage account for the migration, in case the Microsoft Migration API is used. The tool should allow you to use the default Azure Storage account linked to Office 365, or specify a custom one.

- The number of requests the tool can make to both SPO and ODFB. For instance, a high number can affect migration performance due to the throttling performed by SPO when it receives a high volume of requests.
- Full management of Metadata terms during migration between tenants.

Note: While this is not an extensive list of the granular settings a third-party migration tool should provide, it provides an initial idea of the kind of configurations that must be present.

Reporting Features

Moving SPO and ODFB data from one Office 365 tenant to another one using a third-party tool involves more than migrating data from the source tenant to the target tenant, the solution should also provide insights before and after the migration takes place. On the one hand, these insights allow you to identify any migration issues or errors that could take place in the migration so the migration team can remediate them beforehand. They can also help during the migration, by building a list of any problems as they occur, so that they can be fixed.

- The migration tool should provide the ability to scan the data and content structures to identify migration issues such as:
  - Files and folders names not supported in the target tenant that could require additional configuration before starting the migration. For instance, special characters such as "%" or "#" could be used at the source tenant, but they may not be valid at the target tenant if the SPO / ODFB Administrator has not enabled the support for both characters. The following PowerShell cmdlets enable the use of the "#" and "%" characters in a SPO tenant.
  - Missing site templates at the target tenant.
  - Missing users and security groups at the target tenant.

  - Orphan or corrupted information.
- The same issues and errors that may arise when scanning the data and content structures to be moved to the target tenant should be reflected in the migration reports. The tool should provide these insights once the migration has been completed.
- In both cases (pre-migration scan and post-migration reports), the tool should provide not only detailed information about the issues or errors, but also valuable information to fix them.

Automation Features and APIs supported

The ability to automate migration processes incorporating PowerShell support and configuration files is very important. For instance, migrating all the ODFBs (or defining batches) from one tenant to another can be a very straightforward task, and should be automated. Of course, some automation features can be used to move regular SPO content.

It's also very important to check if the tool selected supports the current Microsoft migration APIs. At the time of writing this eBook, there are two main APIs intended for any ODFB and SPO migration scenario:

- **Client Side Object Model (CSOM) API** This API is mostly used when there is a need to migrate complex contents, and SPO information structures, where security settings, site columns and content types need to be migrated, not just information. This API could guarantee up to a maximum of 2GB / H upload rate. It's important to remember that CSOM API calls can be throttled or blocked by Microsoft.

Note: For more information about CSOM Blocking and Throttling, check out the official article available in the SharePoint Online Development documentation: https://docs.microsoft.com/en-us/sharepoint/dev/general-development/how-to-avoid-getting-throttled-or-blocked-in-sharepoint-online

- **The Office 365 Migration API** was designed by Microsoft to provide high performance in terms of the amount of data moved to both SPO and ODFB. This API can provide contents upload rates up to 30 GB / Hour.

In addition to the main APIs mentioned above, there are another two SPO and Office 365 APIs that could also be considered in the future:

- SharePoint REST API.
- Microsoft Graph API.

Pricing

Of course, another important factor to consider when selecting a third-party tool is pricing. You might be wondering if there are any free tools in the market suitable for this scenario; unfortunately, that's not the case. A quick look at any of the vendors in the Office 365 migration spectrum will highlight the following price schemes:

- 1 year licence with no limits in the amount of data and/or users to be migrated.
- Perpetual license, but with limitations in the amount of data to be migrated. The price for this option normally depends on the amount of data to be migrated.
- 1 year license with no limits in the amount of data to be migrated, but tool price depends on the number of users to be migrated: As the number of users to migrate increases, the tool price per user decreases.
- 1 year license with limits in the amount of data to be migrated, and pricing that is based on the number of users to be migrated: As the number of users to migrate increases, the tool price per user decreases.

Finally, the availability of a trial version of the tool is also important, so that you're able to test it before deciding if it is the right tool to be used for your migration scenario.

Other features to consider when selecting a SPO and ODFB Tenant-To-Tenant Migration Tool

So far, we have described some of the key features that should be present in a migration tool, but there are other aspects that, depending on the migration scenario, can also be important. The following table provides other elements to consider when selecting the most appropriate SPO and ODFB migration tool:

Element	Comments
Ability to perform incremental or delta migrations	The tool should provide the ability to make incremental or delta migrations once a migration has been completed. For instance, files and contents not migrated in the first pass can be migrated on a second pass - but without having to copy all the files and contents again.
Ability to perform granular migrations	The tool should allow you to migrate complex SPO objects such as Site Collections or sites, but also individual configurations or items like Workflows, Content Types, Site Columns, Permission Levels, Managed Metadata, etc.
Tool Documentation	Detailed documentation is needed not only to use the tool correctly, but also to know all the possibilities it can bring for SPO and ODFB migration scenarios.
Free helpdesk and tool support	The tool vendor should provide basic (and preferably free) helpdesk and support so common issues when using the tool can be solved / troubleshooted at no cost.
Tool premier support	A common offering many software vendors have is premier support services. This is offered to address issues and problems with the tool or under complex scenarios, customers can get rapid, detailed help and support from the premier support team.
Extensibility	Extensibility enables customers and partners to add additional functionality to support migration scenarios not covered by the tool as standard. A good example of extensibility could be the ability to create custom PowerShell cmdlets to add to the default set of tool cmdlets provided for scenarios where automation in the migration is very important. Some third-party tools have also developed their own scripting languages to automate repetitive work.

Chapter 3: Table 4. Other elements to consider when selecting the most appropriate SPO and ODB migration tool.

SPO and ODFB Tenant to Tenant Migration Tools: A quick overview

A quick search for SharePoint and ODFB migration tools using any Internet search service will produce numerous entries of possible tools, but how do you know solutions are worth shortlisting and exploring further? Table 5 provide a summary of

the best known third-party tools that could potentially be used in a Tenant to Tenant migration project, once they've been evaluated for your scenario.

Tool Vendor	Additional Information
Quadrotech	Tenant to tenant migration for Exchange Online, OneDrive for Business, SharePoint Online, and Teams. https://www.quadrotech-it.com/solutions/migration/cloud-to-cloud-migration/cloud-commander/
Sharegate	https://en.share-gate.com/blog/migration-from-to-office-365-sharegate ODFB Migration with Sharegate
Cloudiway	Tenant to Tenant migration: http://kb.cloudiway.com/onedrive-migration-between-2-tenants/ Pre-requisites and limitations: http://kb.cloudiway.com/onedrive-to-onedrive-migration-prerequisites/
CloudMigrator365	SharePoint Online Migration: https://www.cloudmigrator365.com/migrate-to-office-365/file/sharepoint/ ODFB Migration: https://www.cloudmigrator365.com/migrate-to-office-365/file/onedrive-for-business/
BinaryTree	General information about the tool: https://www.binarytree.com/prodcts/power365-saas/
AvePoint	DocAve High Speed Office 365 migration: https://www.avepoint.com/products/sharepoint-infrastructure-management/office-365-migration/

Chapter 3: Table 5. Overview of third-party migration tools that could be used in a Tenant to Tenant migration.

What about other Office 365 services that use SharePoint Online?

SharePoint Online is not only a core service in Office 365, but also a key building block of some other services and applications available in the platform. Indeed, the following Office 365 applications and services requires SPO to be fully functional:

- **Office 365 Video** is Microsoft's first attempt to create an enterprise video service for Office 365. Office 365 Video uses SPO to store the original video files that are

sent for processing to Azure Media Services, this means that any user can play the videos with a great user experience - no matter the device and network connection in use. Each time an Office 365 Video channel is created, a new SPO site collection is created behind the scenes

- **Office 365 Groups** was also the first attempt (ahead of Teams) to provide a centralized collaboration and productivity solution on top of the core Office 365 Workloads: Azure AD (AAD), SharePoint Online, Exchange Online and Skype for Business. Each time a Group is created in Office 365, a SPO Site Collection is provisioned for the Group. Group members can easily store files there, create lists and document libraries and subsites.
- **Microsoft Teams** is the chat-based workspace that Microsoft made generally available worldwide in March 2017. Teams uses Microsoft Groups as the membership service, which means each time a Team is created, a Group is created behind the scenes and all its building blocks (an AAD Object, an inbox in Exchange Online, and a SPO Site Collection). Again, all the documents generated as part of a Team activity are going to be stored in a SPO Site Collection. Teams also uses ODFB for storing files shared in 1:1 and 1:N chats.

Microsoft does not provide a way to migrate all the elements that are part of these services from one Office 365 Tenant to another. There are also no third-party tools capable of doing this kind of migration due to its inherent complexity, for instance - migrating an Office 365 Group between two tenants typically requires some/all the following steps (depending on usage):

- Create a new Office 365 Group at the destination with the same settings in regards of Group Name, Group Owners, Group Members, etc.
- Copy conversations from the source Office 365 Group to the target one.
- Copy Group events from the source Office 365 Group to the target one.
- Re-create in the target Office 365 Group site any content structure existing at the source Group site.
- Copy contents (documents, folders and list items) from the Office 365 Group source site into the Office 365 Group target site. This operation should also include

the possibility to copy the OneNote contents from the source Office 365 Group to the target Office 365 Group OneNote.
- Copy Tasks from the source Planner Group to the target Planet Group.

In summary, migrating Office 365 Groups between tenants is a complex process and this is also true for Teams. Therefore, talking about migrating these services from one Office 365 Tenant to another, implies a necessity to migrate the data in each individual building block separately, even though they are all part of the same service:
- Office 365 Group site contents can be easily migrated either using a third-party migration tool or Office 365 APIs.
- In the same way, Microsoft Teams site contents can be moved between two Office 365 tenants using Office 365 APIs or a third-party migration tool.
- Finally, migrating Office 365 Video contents to the target tenant can be done following the same pattern: first, Office 365 Video channels must be created at the destination tenant, and then the videos can be moved using the same approach described for Groups and Teams.

What about Microsoft?

Unfortunately, at the time of writing this eBook, there is no Tenant to Tenant SPO and ODFB Migration Tool provided by Microsoft. It's true that Microsoft announced a new SharePoint Migration Tool at their Ignite Conference held in Orlando (September 2017), but for now it only allows you to migrate data to SPO and ODFB coming from two main sources:
- A SharePoint 2013/2016 Farm.
- Corporate Files Server.

So far, we have talked about how third-party tools can handle a SPO and ODFB Tenant to Tenant migration, and about the lack of a Microsoft tool for this migration scenario. If using a third-party tool to handle a SPO and ODFB Tenant to Tenant migration is not an option for your scenario, then we will go on to consider other approaches leveraging the use of the available SPO and ODFB APIs in the next section of this chapter.

Moving content programmatically

The main concern when designing a data migration strategy from any kind of data

source, not only in a tenant-to-tenant/Office 365 scenario, is the transfer speed that can be achieved. There are only two ways to move data programmatically from a system to SPO and Office 365: using the Client-Side Object Model (CSOM) API, or the Microsoft Migration API, and (most of the time) the amount of information to move indicates which option should be used.

If the volume of data is small, (for example a Document Library with some hundreds of documents, or a local directory with a couple of GBs) it is probably a good idea to create a PowerShell script that uses the SharePoint CSOM to move the information to the destination SPO Document Library. The same can be said if the source of information is a legacy system (a BLOB in a Data Base or any other kind of system) or a different Office 365 Tenant. In this case, the work to be done simply doesn't justify the price involved in licensing third party tools.

On the other side, if the amount of data is larger, moving information with the CSOM API will be prohibitive due to the time it takes, as the performance of the CSOM is slow, very slow. Taking this into consideration, Microsoft offers a "Migration API" that considerably increases the data transfer (up to 5 times) to SPO and ODFB. Using this Migration API is totally free and requires, at most, only the costs to temporarily maintain the data in Azure Storage. Bear in mind that some commercial third-party tools and Microsoft own tools can also use this Migration API. It is also important to mention that the Microsoft Migration API is not suited to enable migrations between tenants, this will be explained in more detail later.

There are three different APIs to communicate programmatically with SharePoint Online in Office 365:

- CSOM API.
- SPO REST API.
- Microsoft Graph API.

Of course, each API has its own advantages and disadvantages. In summary, we can say that:

- The client-side object model (CSOM) provides client-side applications with access to a subset of the SharePoint object model, including core objects such as site collections, sites, lists, and list items. The CSOM API offers the most advanced way to work with SharePoint Online because it provides the biggest quantity of classes and methods, covering almost any aspect of the service. For traditional developers with experience working with C# or Visual Basic, there are no limits at all and the use of the CSOM is very intuitive. The main problem of CSOM is the speed of

operation and transfer of data: it depends totally on the physical distance (latency) in between the different components and the intrinsic SharePoint throttling mechanism that impedes to reach high data transfer rates. The CSOM provides both a synchronous and an asynchronous model for invoking server-side execution of commands: this prevents service calls from causing the Web browser to block user interaction for the duration of the call.

- The Representational State Transfer (REST) SPO interface is a WCF Data Service that allows you to use construct HTTP requests to query SharePoint data. Like all RESTful Web services, the SPO REST interface maps HTTP verbs, such as GET, POST and DELETE to data operations. The SPO REST interface is based on the REST-based Open Data protocol (OData) for Web-based data services, which extends the Atom and AtomPub syndication formats to exchange XML data over HTTP. Because OData is a platform-independent open standard, the SPO REST interface is the way to access SPO data from platforms on which the CSOM may be unavailable, such as from non-Windows–based operating systems. The REST implementation can also return the output in JavaScript Object Notation (JSON) format as an alternative to the ATOM feed. JSON is a compact representation of the returned results that can be easily parsed by JavaScript clients. Although it provides various methods to move data to and from SPO and ODFB, the REST API is less powerful than the CSOM API, and it is also more difficult to work with. It is not a simple task to get the correct and necessary authorization OAuth tokens, and the aforementioned speed issues are still in place - elevated by the fact that there are size limits when transferring information throughout HTTP(s).
- The Microsoft Graph API (and its subset, the Office Graph API) allows developers to build relationships and interactions between resources in Azure Active Directory, all services of Office 365, and other applications and data resources in a unified way. The Graph API is the evolution of the Office 365 Unified API into a single interface. The development platform exposes multiple Microsoft cloud service APIs through a single REST API endpoint. Microsoft Graph API provides continuous navigation between service entities such as users, groups, mail, messages, notes, tasks, and calendar, and the Office Graph. Data can be accessed from multiple Microsoft cloud services such as Exchange, OneDrive, SharePoint, OneNote, Planner and Azure Active Directory. The Graph API is under constant development, meaning that it receives monthly improvements from Microsoft, but it is still not as powerful as the CSOM when it comes to working with SharePoint.

SPO and ODFB Tenant to Tenant migration on your own terms

Migrating data between tenants using SPO and ODFB APIs

As we mentioned earlier, the use of CSOM and other traditional APIs in a SPO and ODFB migration assumes the risk of crushing the SharePoint Frontend servers, making them unresponsive to other users. With SPO, such an event can easily knock down the service for many tenants, not just the one where the migration is being done. To prevent this, Microsoft enforces throttling on CSOM calls just in case many requests happen in a short period of time. The subsequent effect is that migrations using this approach are slower because it will be necessary to re-try the calls that have been throttled by SPO.

That said, in cases where only small amounts of data need to be moved, the CSOM API is a valid approach. Defining migration batches is also a suitable technique to minimize the number of requests that are passed between the client and the Office 365 tenant. This decreases network traffic, and offers a smoother user experience. Both CSOM and REST APIs support batched requests, although the implementation can vary.

The SPO CSOM API was designed to apply a programming model where operations are sent to the server in batches. The client always performs a series of data operations on the ClientContext object that are submitted to the server in a single request when the method ClientContext.BeginExecuteQuery is called. The ClientContext.Load method can also be used to tell the client context object which object type to return when it executes a request.

The following piece of code implements a batch item creation in a SharePoint List (note that the example below is not fully functional):

```
static void BatchInsertCSOM_CS(ListItemCollection SourceItems)
{
  // Authorization not fully shown in code
  ClientContext destCtx = new ClientContext("https://[tenant].sharepoint.com");
  List destList = destCtx.Web.Lists.GetByTitle("MyList");
  foreach (ListItem oneSourceItem in SourceItems)
  {
    ListItemCreationInformation itemCreateInfo = new
    ListItemCreationInformation();
    ListItem newListItem = destList.AddItem(itemCreateInfo);
    newListItem["Title"] = oneSourceItem["Title"];
    newListItem.Update();
    destCtx.Load(newListItem);
```

```
    }
    destCtx.ExecuteQuery();
}
```

In this example, there is a collection of items used as input parameter. After creating the client context required to get the destination list, a loop traverses each source item and creates a new one in the destination list. Finally, the command "ExecuteQuery" is sent to create all the items in the destination list. This batch operation is sent in a single request to the SPO tenant.

The request batching process described here helps to improve performance and reduce network traffic in several ways. It leads to fewer Web service calls between the client and the SharePoint server, which reduces the amount of data moved: for example, it is possible to perform two list queries in a single request. Also, as a set of operations occur on the server in a single request, the data being acted on doesn't need to be moved between the client and the server for the intermediate operations— just the list of operations and the result set are passed between the client and the server.

Request batching requires a different mindset when creating queries from client-side code. Be aware that you do not have access to any results until you call ExecuteQueryAsync or ExecuteQuery and receive the call back with the results. If you want to implement conditional logic in the client-side code, that can't be expressed in the command list sent to the server, you will need to execute multiple queries. It is crucial to aim to group your operations and minimize the number of service calls. This means you may need to think about how you sequence your logic to take full advantage of request batching.

Note: SharePoint Server-Side Object Model (Full Trust) provides a batch method named SPWeb.ProcessBatchData which can be used to add, edit and delete list items in bulk. This method is very helpful to manage large number of items at once without making repeated calls to the database. The SharePoint CSOM API does not contain an equivalent to this method, so there is no mechanism/way to make list items operations in bulk.

Migrating data between tenants using PowerShell with CSOM API

PowerShell for SharePoint Server (On-Premises, version 2010, 2013 and 2016) contains two specific cmdlets for data migration, Export-SPWeb and Import-SPWeb, but unfortunately they are not available for SPO, and subsequently, it is not possible to use them in a Tenant to Tenant migration. These cmdlets allow you to move a complete SharePoint Site, List, or Document Library in a safe and quick way, and the Export-SPWeb cmdlet is the foundation for the Migration API that will be described later in this chapter.

Another two handy and very popular cmdlets in SharePoint On-Prem, Copy-SPSite and Move-SPSite, used to make a copy of a site collection from an source content database to a specified destination content database, or to move it, are also not available in SPO.

Taking into consideration the huge amount of limitations for programmatic migrations to SPO, this means that the only leftover option to migrate data using PowerShell is to use the CSOM API with custom PowerShell Scripts, or creating custom PowerShell cmdlets that use the CSOM in a similar way. As an example, the following PowerShell code shows how to implement a batch item creation in a SPO list using CSOM:

```
Function BatchInsertCSOM_PS([ListItemCollection]$SourceItems)
{
  // Authorization not fully shown in code
  $destCtx = New-Object Microsoft.SharePoint.Client.ClientContext($URL)
  $destList = $destCtx.Web.Lists.GetByTitle("MyList")
  $destCtx.Load($destList)
  $destCtx.ExecuteQuery()
  ForEach($oneSourceItem In $SourceItems)
  {
    $itemCreateInfo = New-Object
Microsoft.SharePoint.Client.ListItemCreationInformation
    $newListItem = $destList.AddItem($itemCreateInfo)
    $newListItem["Title"] = $oneSourceItem["Title"]
    $newListItem.Update()
    $destCtx.Load($newListItem)
  }
  destCtx.ExecuteQuery();
}
```

Note: If the migration requirements are simply to re-create an information structure (Site Collections, Sites, List and Libraries, including their configuration such as rights structure, fields definitions, etc), the SharePoint Patterns and Practices (PnP) PowerShell Cmdlets (find out more here) can do the work in a relatively simple way by exporting the original source and importing it in the new SharePoint tenant. Take into consideration that PnP is not designed to migrate data between SPO tenants, it is designed to extract the definition of a SPO structure and provision it on a different site (located in the same tenant or in another tenant).

Using the Microsoft Migration API

From an Office 365 perspective, the most important reason for using the SharePoint Migration API is the high data transfer rate that can be achieved. According to Microsoft, it gives up to a five times speed increase over traditional methods. This

API is mainly designed to read content from a file system or on-premises SharePoint farm, therefore it is required that the content must reside on, and be accessible from, one on-premises location. Of course, this method clearly cannot be used for Tenant to Tenant migrations. When identified, an API command will build "packages" from the content and create XML files that define metadata and attributes. A second API command will then upload these packages and XML files to an Azure Storage location.

From the moment the content and XML files are uploaded, an Azure Migration job is added to a processing queue. From this point on, the process is fully automated by Azure and SPO: Microsoft tenancy bots scan the queues, and when an existing package is recognized the processing begins moving content, metadata and item-level permissions into a destination in a predefined SPO location or inside ODFB. On top of completion, exhaustive log files are pushed back into Azure for review and archive.

There are several factors that affect the performance of migrations:

- Size of the migration - For the creation of the migration package(s), the time to generate a package with many small files takes considerably longer than fewer larger files.
- Internet connection capacity and latency - Since the packages must be uploaded to Azure Storage, a slow internet connection will drastically affect the migration time. Microsoft offers the option to copy the packages to a hard drive and send it to the Microsoft data center, where it will be used by Azure (of course, costs must be taken in consideration). A second point to be aware of is that Azure and Office 365 Data centers must be preferably in the same location to minimize the connection latency during the migration.
- Single content databases are another factor. Microsoft has modeled the internal (Azure to SharePoint) migration process in such way that a single package is always moved into one content database at the time. In SPO and ODFB, content databases are auto-assigned for each Site Collection and ODFB. It could be the case that the entire tenancy resides on a single database, but there is no guarantee for that, in which case, the packages will be processed sequentially, never in parallel.

In a similar way, there are some limiting factors that should be considered when planning migrations to SPO and ODFB using the Migration API:

- The source data must be a SharePoint Server 2013 or 2016 farm, or an On-Prem File Share. The destination is always SPO and/or ODFB. Other sources and/or destinations are not supported, eliminating the possibility for the tool to be used in

an easy way for migrations between tenants.

- Only contents, metadata and item-level permissions can be migrated. Microsoft has promised that it will be possible to migrate other SPO objects in the future (Site Collections, Sites, Document Libraries, Lists, etc.). At the time of writing, this is not yet in the SPO road-map.
- The Migration API does not handle any other SharePoint object types, such as web parts, workflows, InfoPath forms, sites, lists, content types, and security configurations They should be manually created before the migration happens, using any kind of programming, or by using third-party tools.
- Incremental or delta migrations, a basic feature of several third-party migration tools - that allows users to keep using the source system until the end of the migration - is also not part of the Migration API.

To conduct the migration, you will need to have a local computer with remote access to any of the Front-End servers of the SharePoint On-Prem farm (or file share, if the migration is using this option), and the PowerShell cmdlets for Azure and SPO

installed. The "SharePoint Online Management Shell" can be downloaded from here,

and the "Microsoft Azure PowerShell" can be downloaded from the GitHub site here.

The following cmdlets can be used for migrating data from an on-premises environment to SPO and ODB. Note: these Cmdlets are currently not ready for a Tenant to Tenant migration scenario:

- ***Export-SPWeb*** - Creates the initial packages of Lists and Libraries of the SharePoint On-Premises farm. It is compulsory to specify the "*-NoFileCompression*" parameter so output will be created in a folder without compression and hence, the "*-Path*" should be given without .cmp. Also, the "*-ItemUrl*" parameter is required and must be a relative path (for example, if the absolute URL is *http://domain.*

com/DocLib, then *–ItemUrl* will be "*/DocLib*"), because at this time, the migration API is directed only for List and Document Library content

- ***New-SPOMigrationPackage*** - Creates the migration package grounded on source files in a local or network shared folder.
- ***ConvertTo-SPOMigrationTargetedPackage*** - This cmdlet translates a previously exported SharePoint package or file share-based package to the package needed by SPO.
- ***Set-SPOMigrationPackageAzureSource*** - Creates the Azure containers and

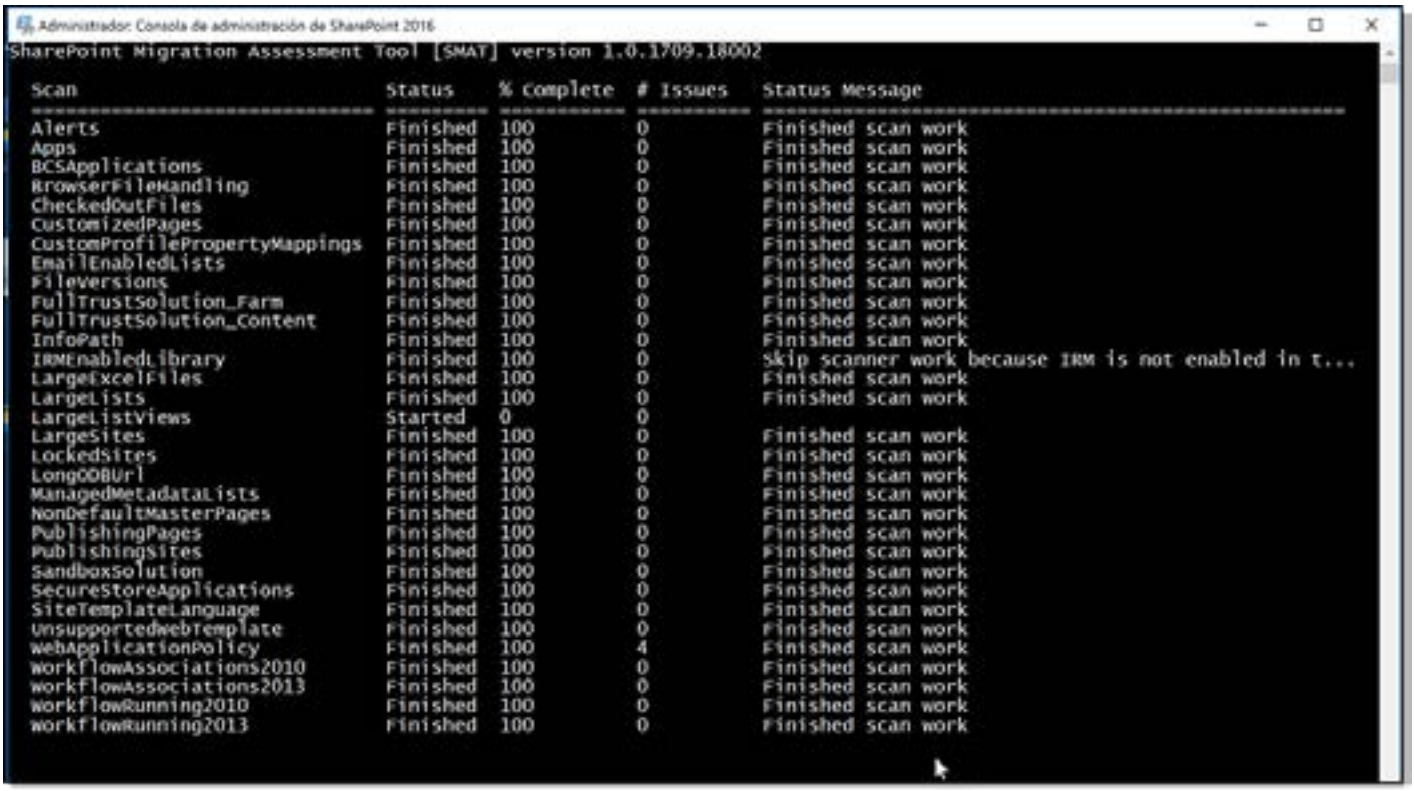
uploads the migration package files into the appropriate containers.

- ***Submit-SPOMigrationJob*** - Starts the migration job with a reference to the previously uploaded package in Azure Blob storage and the destination site collection.
- ***Remove-SPOMigrationJob*** - Allows you to delete a previously created migration job if necessary.
- ***Get-SPOMigrationJobProgress*** - Examines the progress of a submitted Migration job.
- ***Get-SPOMigrationJobStatus*** – Enables you to monitor the status of a submitted Migration job.

Microsoft has also released two tools to facilitate the migration of on-premises contents to SPO and ODFB using the SharePoint Migration API: the "SharePoint 2013 Migration Assessment Tool" (despites its name, it works also for SharePoint 2016 Server) and the "SharePoint Migration Tool".

The "SharePoint 2013 Migration Assessment Tool" (SMAT) is a console application to analyze the content data of a SharePoint Server 2013/2016 (it works as well with the first beta release of SharePoint Server 2019). The tool can help identify the impact of the migration to SPO, and has been designed to execute without affecting the daily working of a farm. This means that, if used in busy systems, it can take a couple of days to get a complete analysis of the farm. During the analysis, the tool will report the progress in the console window. After finishing, the complete log files can be found in the Logs directory, together with an operations summary, and several details about the problems that can impact the migration. Take into consideration that the tool automatically sends anonymized information to Microsoft, and if required, information about the organization can be added as well at the end of the analysis.

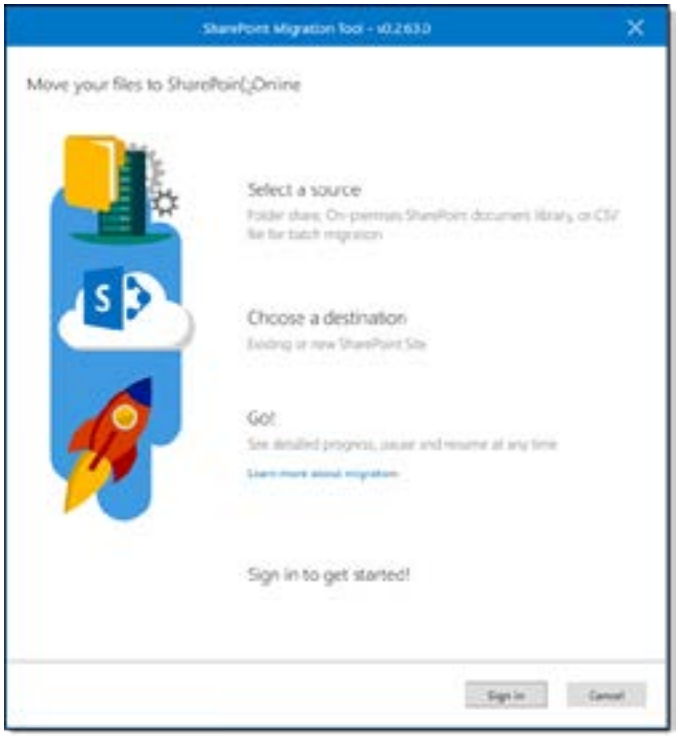
The tool can be downloaded from the following Microsoft site. After the download completes, and the files are unzipped in one of the Front-End servers of the farm, the tool can be started executing the "SMAT.exe" archive from a Command Prompt or PowerShell console.



Chapter 3: Figure 1.- SMAT Tool for SharePoint migrations.

The second tool: the SharePoint Migration Tool, coordinates all the migration work from a WYSIWYG interface, using the SharePoint Migration API. This means that you don’t need to know or use any underlying Azure or PowerShell Cmdlets, scripts, packages, manifests, etc. To use the tool: from one of the Front-End servers in the on-premises SharePoint farm, go to https://hrcppestorageprod.blob.core.windows.net/migrationtool/default.htm using a navigator, accept the use terms, and install the tool using the "Install" button.

When the installation process ends, the initial window of the tool appears automatically:



Chapter 3: Figure 2.- SharePoint Migration Tool.

Use the "Sign in" button to log into the SPO tenant. There are three options to start a migration:

- Using a List of Library from the local SharePoint farm as source
- Using a File share
- Initiate a "CSV file for bulk migration"

Note that there are no options to conduct a tenant-to-tenant migration. If the first option is selected, the following screen allows you to select the source List or Library, the account to be used, and the destination site and container. This creates a migration "Task". It is possible to create several tasks at once, as well as to save the task definitions to a CSV file. Certain configurations for the migration can also be

selected: if you only require scanning (it's good to know if there are any problems before beginning the real migration), if the version history should be migrated, how many versions should be kept, as well as a series of filters for migration of archives "created before", "modified before" and/or using predefined file extensions.

The "Migrate" button initiates the migration of all defined tasks. The tool initially runs a check of the environment, determines if there are any issues with the data, and migrates it to SPO. It's important to note that the source and destination must use the same language definition, otherwise the names of Content Types, Fields and Columns could be different, impeding the migration.

The migration of File Shares is similar: simply select the source file share instead of the source List/Library. Archives with read-write permissions will get "Contribute" rights in SharePoint, and archives with read-only permissions will get "Read" rights in SharePoint Online. If the Local Active Directory is synchronized with Office 365, all permissions will be mapped, but if not, the archives will receive the permission definition of the destination List/Library.

This tool also allows you to make migrations from several data sources to one SharePoint tenant using one CSV file defining the origin and destination data. The file should have the following format:

- The file must have the .csv extension, created by Excel or as a text archive.
- The field separator must be a comma.
- Each migration task must be in one line in the archive.
- The archive cannot have a header: the data must begin directly on the top.
- Each line must have six data fields. If one of the fields is irrelevant, it can be emptied but it must have a comma separation. The fields to use are:
  - Origin (Required) - Path to the file share or SharePoint URL.
  - Library/List Source (Optional) - SharePoint List/Library name to be migrated.
  - Sub-folder Source (Optional) - List/Library subfolder name.
  - Web Destination (Required) - SPO destination URL.
  - Library/List Destination (Required) - List/Library name in the destination SPO.
  - Sub-folder Destination (Optional) - Subfolder in the destination List/Library.

Because the "SharePoint Migration Tool" uses all the functionality described for the SharePoint Migration API, the data transfer speed, advantages and limitations

described earlier apply here. The tool simply adds the additional scan step to ensure that there are no migration conflicts, however it does liberate the people in charge of the migration from creating the whole infrastructure in Azure and battling with PowerShell commands and parameters.

Conclusions

Due to the various limitations and restrictions involved in tenant to tenant migrations for SharePoint Online or OneDrive for Business, third party tools are the most appropriate option for moving information and content structures between tenants. The lack of a Microsoft Tool, PowerShell cmdlets, and the inability to use the SharePoint Migration API all contribute to the complexities, and restrictions for this type of migration. All third-party tools are not alike, and it is crucial that you consider each possible solution for your migration scenario. Before you begin any steps toward migration, it's important to carefully review the most important features a third-party tool must provide for your tenant to tenant migration process.

Chapter 4: Managing a Tenant to Tenant migration project

Barbara Waskey

When an email migration or an acquisition project comes along, let's face it, things can get pretty chaotic. It can cause some highly negative emotions to come through.

While most of this book is focused on the solutions and technology behind the process, there is an elephant in the room that cannot be ignored. That elephant is the human element. Charged with emotions that, when embraced, can make your migration a positive experience. Yet if ignored, your migration may still succeed, but will not be truly successful.

In this chapter, we will explore successful migration methods embraced by our user community. We will address not only the process, but also the human elements, which can make your migration a true success through and through.

Story

Our migration effort included converting network and voice for 90+ offices, replacing ~15,000 desktop devices, and migrating email and user data supporting a workforce of 15,000 employees. While both companies had great experience in acquisitions and data migrations, never had either tackled a project of quite this magnitude.

As most companies do, the initial focus was on the technology. This is an IT Project, right? How do we solution this? With the combining of the two companies, there was an alignment and collapsing of departments, bringing the human element to the forefront.

Enter the Project Manager (PM). For this particular type of project, you need a project manager with strong emotional intelligence. A respected leader, also known as an agent of change, best fills this role. As you will read later, cheerleading skills are also a bonus! For this migration we had two PMs. There was a need to collaborate between the two organizations, so creating balanced teams was important. As the teams were forming, the two PMs worked together to lead the team through several exercises.

Face to face meetings are highly recommended for any team building activity. Meeting with people in person reduces frustration and anxiety, as you bring more human elements into the conversation. Body language is also extremely helpful in understanding if your message is being well received or challenged. This feedback loop helps you better tailor your message. While one organization worked primarily from the same office, the other organization was a virtual team that rarely worked together face to face. The project itself was so large that conversations would scatter and divert into various directions, rarely coming to any productive conclusion. The other lingering issue was the perception that our team would be reduced once the project was complete, drawing out survival instincts and people fighting to maintain position or credibility in hopes of saving their job.

Bruce Tuckman, a famous psychologist, published an article in 1965, "Developmental Sequence in Small Groups" known by most people as the phases of team formation: Forming, Storming, Norming and Performing. These phases were certainly present during this project, and forming quickly turned to storming. Starting with the technology discussions, there was a highly charged brainstorming activity. Everyone had an opinion on how this should be done, usually based on how they did this in the past. Emotions ran high, and everyone wanted to demonstrate knowledge and experience to remain relevant. The atmosphere became competitive, feelings were hurt, people were humbled. The PMs were often the referees to keep the discussions positive and productive. After a while, the discussions stalemated. This was primarily because, while each was a good idea, they didn't seem to fit the problem at hand. Each team had their own set of predefined assumptions that they were basing their solutions on. Each of these solutions would have resulted in tremendous impact to the business. The challenge had to be broken down into logical requirements. Each requirement would need to be solved. Then, like a puzzle, assemble the solutions together to complete the picture.

The PMs shifted the focus into a requirements discussion based on what we knew about the business. The worst way to run a requirements session is to open it up with "What do you need?" Having skilled project managers that are familiar with how the business operates is extremely valuable. PMs should ask leading questions like: "How many offices are involved? How are teams formed? What does the organizational structure look like? Do you have any business or process flows that demonstrate the flow of information between teams? Do people work in the same office? Do people work from home? What methods of collaboration are in use with this company? Do they use things like Shared Mailboxes, or Public calendars? Thinking of the technologies at play and understanding how the business might use them is critical for understanding the requirements.

A good PM will consider all information provided to be important. If it isn't suiting the goal at hand, capturing it on a Parking Lot board will enforce that the topic is valid and important, and commit to coming back to it later. This helps in remembering the discussion topics for later, as well as facilitating a focused discussion when someone keeps diverting the conversation. And don't forget about the quiet engineer. Some people are not comfortable speaking up in a charged atmosphere. Be sure to draw them into the conversation and get their opinions and feedback. It may be necessary to follow-up with introverted members of the team in a one on one session - equally so for those who are more assertive. Encourage all points of view and when assumptions are brought up, make sure they are valid and apply to this particular problem. Avoid solutioning when talking about the requirements first. There are many ways to solve a problem, so weigh out the options carefully without bias for how one team may have always done things in the past.

Tips for running a successful requirements session:

- Identify the key players and their roles/expertise
- Limit the attendees to keep focus
- Request any helpful diagrams/documentation prior to the session
- Ask leading and open questions to get required details
- Engage all participants
- Keep a parking lot – don't spend too much time on an item
- Validate assumptions
- Assign a scribe and make the notes public

Approach

In our case the organizational structure was a matrix. Bricks and mortar buildings did not necessarily define teams, and users had both functional and traditional managers. An office by office approach was not "one size fits all", so it would not work in this situation. Email was not just a tool they utilized, it was a critical business function for some departments. Some matrixed departments utilized shared mailboxes to support critical business workflows. There were significant financial penalties incurred when these workflows did not complete in a timely manner. By laying out the problem statements and how they impacted the business, the team was forced to think about the migration in a new way. Suddenly the "way they had always done this" was not in the best interest of the business.

Our particular migration was further complicated by the need to also migrate users to

a new network and workstation. Due to the size and complexity of our organization, we were unable to simplify with an office-by-office approach. Our strategy needed to include multiple scenarios based on order of events. Sometimes the mailbox was the first to migrate, and other times it was the workstation. When these happened at the same time, it was much simpler from a technology perspective, but a lot for the user to absorb. Avoiding user overload was critical to the success of our project. As we scheduled our users, we aimed for as many of these as possible, and reserved the others for "one-off" migrations. These "one-off" migrations added a lot of administrative overhead but minimized the business impact.

Participation in the requirements session helped the engineers to better understand the business needs. They learned how the business might be impacted by these activities. We defined our approach with the goal of making the process easier for the business while improving the success of the migration. Simple things like engaging the business to clean up their mailboxes and files, reviewing how they operate, and understanding their workflow helped to identify gaps in the process. As the team was 'norming', we developed several strategies that they would employ to treat this migration more like an implementation project.

Our implementation approach included -

- Communication Strategy
- Interviews and Inventory
- Education and Engagement
- Design and Testing Strategy
- Training Strategy
- Deployment Strategy

Communications and Champions

One of the best ways to obtain trust from the business is to share vision early and often. Include key benefits of the migration that they will understand. Perhaps they are gaining enhanced security, cost savings around system management and contract consolidation, explain that the technology has improved, or they will obtain better collaboration tools and improved user experience. Including the business in the process gained support for our project and made us more successful. We announced the migration plans and acknowledged their fears and concerns about business disruption. We created posters throughout the planning phase advertising the project. Get people talking! We assured them that we would do everything possible to minimize disruption, but we needed their help. We recruited volunteers from organizations and offices (based off organizational charts, not phone trees) that could help share concerns, test

solutions, and champion our cause with the rest of the business. Much like a focus group would test a prototype, we wanted these champions to improve our process.

With a little help from HR, we created an employee listing that allowed us to see what departments people worked with, what building they worked from, identify work-at-home users, and capture any users currently on leave. Working with our champions we examined the organization and created logical groupings for migration strategies. This was considered an assumption, as we had not yet validated how much data this entailed, nor how much data we could manage in a migration period.

Interviews and Inventory

We first interviewed the champions to refine our interview and data collection process. We were shocked to learn how email was being used in everyday workflow, and how this migration could bring production to a grinding halt. The business had many business-developed applications (BDAs) that were created to automate processes that very few people knew about. Access databases were very common in our organization, and many were used to automate collaboration and communications through email. There were other processes that were integrated into critical system applications like electronic faxing. Had we not known about these, we would have caused business and customer disruption, potentially resulting in financial penalties.

We then worked with our distributed support teams to send out questionnaires to every single employee. Information gathered included what applications they use, where they collaborate with their team members and other teams, what other teams they work closely with. Do they store any information locally on their computer? System automation can be helpful as well if you can identify what is loaded on the machines and define drive mappings. Inventory of user data was critical to identify exactly how much data must be migrated. Inventory of mailbox sizes and email archives or PST files was required to later determine how many people could migrate in a defined window.

Some of our key inventoried items included - public folders, shared mailboxes,

distribution lists, fax mailboxes, delegated access (for administrative assistance) size of their mailbox, how many PST folders they have, where they store their data, what applications they use, and their drive mappings. Information was collected and loaded into a repository that would help us define the migration schedule bundles.

This information was also helpful for the team to resolve some of the other challenges with the collapsing of two organizations into one. As we looked at who worked together we began to see that teams were forming across the combined company and some of the departments were either no longer needed or changing in nature. Duplicate department names were addressed as well as any other name matches, as these are common in large mergers. Distribution lists also need to be examined and determined if they are still required and if there are any changes. It wasn't enough to do a one to one migration; we had to determine the new organizational structure and the functional organization. This strategy put a certain level of ownership on the business to further define their requirements. If they didn't tell us what they accessed, we couldn't promise that it would work once migrated. In our case, because this tied into the desktop conversion, detailed requirements were critical.

Education and Engagement

Encourage the urge to purge! Help users understand how their volume of data will directly impact the extended duration of the migration. Set goals for the users to get data levels within a defined limit (within compliance parameters that your organization has set forth). Give them time to comply but set benchmarks along the way as many will wait until the last minute. Create and distribute simple instructions that help them prioritize the largest files like attachments in email and calendar appointments. Get users to clean out mailboxes, archive off old calendar appointments, delete duplicate files. This is a great time to teach them about good habits, and the company's preferred method for collaboration, for example: email should not be used as a document repository!

The reality is that you will significantly shorten the duration of your migration program by minimizing the amount of data to be migrated. Do you have a fun way to engage your user community and share progress? Take an initial inventory and track your progress with periodic updates. A little public humiliation can be a great motivator and instill a sense of friendly competition among teams. Get leadership involved and report frequently on progress. Find creative ways to reward progress – I promise it will save you time and money in the long run.

Design and Testing

The rest of this book focuses on the technology, but this chapter is here to remind you that there is another design component. There was a careful scheduling that needed to take place to minimize critical business disruption. This was a very time intensive process that paid off with great success. The technical teams defined the volume of data that they could successfully manage during a weekend migration period. The amount of data each user was consuming was added to the employee listing. This needed to align closely with a real-estate strategy as we wanted to combine the desktop/office conversion with the email migration as closely as possible. While several teams were located in the same office, many were not. The distributed teams were highlighted on the list. Then groups were formed by office locations where possible, and coordination could be made with office migrations of network, voice and desktop computers. We planned for on-site support for the major office migrations, and this was extremely well received. Users were given clear instructions, but there are always issues that crop up, users that don't understand, or users that want you to walk them through it due to lack of confidence. Even with the best of planning and testing, we ran into some unique configuration issues that required technical support. This would have taken much longer to manage through remote support, so the local support teams were critical to our success.

Working closely with all migration teams involved, a master schedule was created. To maintain the velocity needed to minimize disruption and complete the email migrations as quickly as possible, we had to staff up the desktop migration teams. We were managing 3 separate desktop teams going office-to-office and updating workstations all while coordinating the email migration in the background with one combined team. Bringing all the pieces together was like choreographing a ballet with 10,000 dancers. If we hit a problem with a migration, it had the potential to impact other locations and users, so our timeline was critical. We had to perfect our process to minimize risk to the schedule.

Now to test our theories… A sample group was taken and walked through the process from end to end. Every step was documented, and every issue was captured. This

allowed the team to create a series of training materials and perfect the process. Gaps introduced by the migration were often solved with technology band aids or workarounds. Once refined, we moved to the next pilot group. They were walked through training materials and sample communications to educate and guide them through the process. Their feedback allowed us to further refine our migration processes, improve our communications and enhance our training materials.

Training

Depending on the actual user impact, you may need formal training sessions or maybe you only need quick reference guides that can be distributed on the desktop for migration day. We had a steep learning curve, so we made training mandatory. As users migrated from one company to the other, there was the potential to impact recurring calendar appointments. Orphaned appointments can be addressed through technical solutions and Cross Tenant Collaboration enhancements have minimized additional calendar impacts. The key message here: you must communicate and train the user on what actions they will need to take.

Your training strategy should also be somewhat iterative, in that you should start with smaller focused groups, and get their feedback on the materials. Did the documentation have what they needed to be successful? Was it written in laymen's terms for all users to understand? Does it provide a quick reference guide for frequently asked questions? Continue to enhance the training material and create feedback loops with your users to help identify the gaps. We utilized online surveys to poll the users for feedback on the entire migration process. We wanted to understand if the communication and training was adequate, and where we could improve the process. This allowed us to continue to refine the materials so that they worked for a broad audience with varying backgrounds.

Deployment strategy

Tied in with the communications strategy, we notified stakeholders and the business community frequently about our progress and what to expect during the migration period. Confirm your schedule with key stakeholders and your champions to avoid collision with critical business needs. For example –don't migrate Finance during Month End or Close. Don't migrate Development Teams during a release weekend. Don't migrate your Customer Service team all at the same time, especially if their highest call volume is Monday morning. Don't migrate your Executive teams the week

of a board meeting.

Share your schedule. Let people know when they are expected to move. If you hold a Go/No Go for each migration, this should be restricted to IT and the business lead to confirm readiness to proceed. A "No Go" will impact the entire schedule and groups will likely get shuffled causing business impact. This restarts the collision analysis and requires coordination of several teams to align.

Notify them with predefined communications at least 2-4 weeks prior to their migration date. Inform them of training opportunities and remind them if they are mandatory. Keep information flowing in small, useful bites. Don't overload them with too much information – keep it relevant for the time period. Have a "T Minus' countdown and send communications at T-2 weeks, T-1 week, T- 4, 3, 2 days and a "needs to know" reminder on the day before the migration and "Go Day." These communications can be clean up reminders, check lists, contact lists, and migration day instructions that will need to be printed out a day in advance.

If your project runs as long as ours did, watch out for project fatigue. Once the excitement wore off for our technical teams and the program shifted more into what felt like operations, it was easy to lose momentum. The PMs and Management should step in and share progress, milestones, achievements and remember to celebrate success along the way. Make sure you have "Cheerleaders" on your team that help to promote a positive work environment and continued gratitude. Recognize and reward those that are going above and beyond to maintain the progress and momentum.

Status reports throughout the project delivery can help with this as well. Share your actual counts completed, not just percentage. Four percent sounds small until you realize it equates to 650 mailboxes! With each migration wave, share your statistics. Examples may include # of mailboxes migrated in that wave, numbers migrated to date and percentage progress toward goal. Include your failure rate and issue count. Are you on track or maybe even ahead of schedule? What about other components like shared mailboxes, distribution lists and mobile devices?

Conclusion

So how did we do? In the end, it was the best team building experience I've ever been a part of. We focused on the fact that this was not a technology problem, but about people and perception. When you can engage and empower your users to influence the success of your project, you will reduce the business impact and improve the overall experience. Our team was now performing! We migrated about 15,000 accounts with less than a quarter of one percent failure rate requiring manual intervention. Our team was celebrated for our ability to break down walls and form a true team in the face of adversity. The project formed relationships that lasted well beyond the conclusion of the project. I hope that you can use some of the techniques that worked well for us in your migration!

Cloud Commander

Cloud Commander enables collaboration between Office 365 tenants and delivers a systematic, structured approach for migrating data between tenants. Cloud Commander currently offers Exchange Online, OneDrive for Business, SharePoint Online, and Microsoft Teams migrations. Find out more about the tool here.

Our Implementation Approach and Checklist

- Communication Strategy –
  - Announce early that this is coming, what to expect and how they can help
  - Acknowledge concern for disruption
  - Identify the business benefits
  - Enlist business support and champions
  - Engage champions to keep communication flowing
  - Share strategy and progress early and often
- Interviews and Inventory Strategy –
  - Start with technology champions – refine and simplify the process
  - Inventory mailbox sizes including PSTs and other archives
  - Capture distribution lists, delegated access, fax mailboxes, shared mailboxes, public folders
  - Inventory applications, document repositories, drive mappings
  - Uncover any business-developed applications (BDAs), including BDA's in the form of Access databases!
  - Inventory strategy – capture what must migrate and how big is it? Define migration parameters

  - Mobile phones
- Education and Engagement Strategy –
  - Involvement early and get the users to embrace the project
  - Remind users of Data Retention Policies
  - Teach healthy email habits
  - Encourage reduction of data
  - Track and report progress
- Design and Testing Strategy –
  - Establish technology Band-Aids or workarounds for known disruptions where possible
  - Establish limiting parameters like volume of data that can migrate in a defined time period
  - Determine schedule of migrations - who migrates with who & when
  - Determine support needed for migration activities
  - Will multiple teams improve velocity?
- Training Strategy –
  - Determine impact on users
  - Develop training materials and schedule classes
  - Declare importance of training (is it mandatory?)
  - Create checklists and reference guides
  - Encourage frequent feedback and incorporate into materials
- Deployment Strategy
  - Publish your schedule
  - Provide frequent updates and KPIs
  - Frequent, targeted, scheduled communications
  - Encourage frequent feedback and incorporate into process
  - Prevent project fatigue
  - Celebrate success

This eBook is sponsored by Quadrotech. Cloud Commander is their industry leading solution for tenant to tenant migration, relocation, or consolidation. Offering superior speeds, and the security of hosting within your own Azure environment if you choose, Cloud Commander can migrate Exchange Online, OneDrive for Business, Sharepoint Online, and Microsoft Teams workloads. **Find out more** here.