O365 High Speed Migration Technical Solution

Detailed View



**HCL Technologies**

**SharePoint Service Line – Modern Apps**

**Contents**

[**Background** 2](#_Toc458533766)

[**Approach** 2](#_Toc458533767)

[**System Requirement** 2](#_Toc458533768)

[**Prerequisites** 2](#_Toc458533769)

[**STEP 1 – Create Target Site Structure** 3](#_Toc458533770)

[**STEP 2 – Create packages** 3](#_Toc458533771)

[**STEP 3 – Upload Package** 5](#_Toc458533772)

[**STEP 4 – Submit to Migrate to O365** 6](#_Toc458533773)

[**References** 8](#_Toc458533774)

# **Background**

Migration of huge volumes of content from on-premises installations of SharePoint to multi-tenant O365 SharePoint Online used to be very time consuming using traditional CSOM APIs provisioned by Microsoft.

Migration of content volumes exceeding 500 GB which is heavily dependent on network factors like bandwidth, availability etc., will run into months.

Moreover, Microsoft imposes a number of restrictions on migrations using the traditional CSOM path.

Challenges using Traditional CSOM for Content Migration

* Limitations on the number of Client Side Object Model (CSOM) calls, a particular user can initiate on Office 365 within a given timeframe – referred as user based throttling.
* Limitations on the number of CSOM calls that can be initiated against a particular Office 365 tenant or SharePoint Online farm within a given timeframe – referred as tenant based throttling.
* Bandwidth limitations when copying content from on-prem to O365.

*Note: Depending on usage, Microsoft used to fine-tune thresholds so that users can consume the maximum number of resources without degrading the reliability and performance of SharePoint Online. Hence no strictly defined thresholds/limitations are shared.*

In order to address the problem of large-scale migrations to O365 SharePoint online (that can potentially have adverse cost and timeline implications), Microsoft has launched a set of High Speed Migration APIs which can significantly boost migration speed by leveraging Microsoft Azure.

# **Approach**

Microsoft provides four-steps approach using PowerShell scripts.

## **System Requirement**

**Supported Operating Systems:**

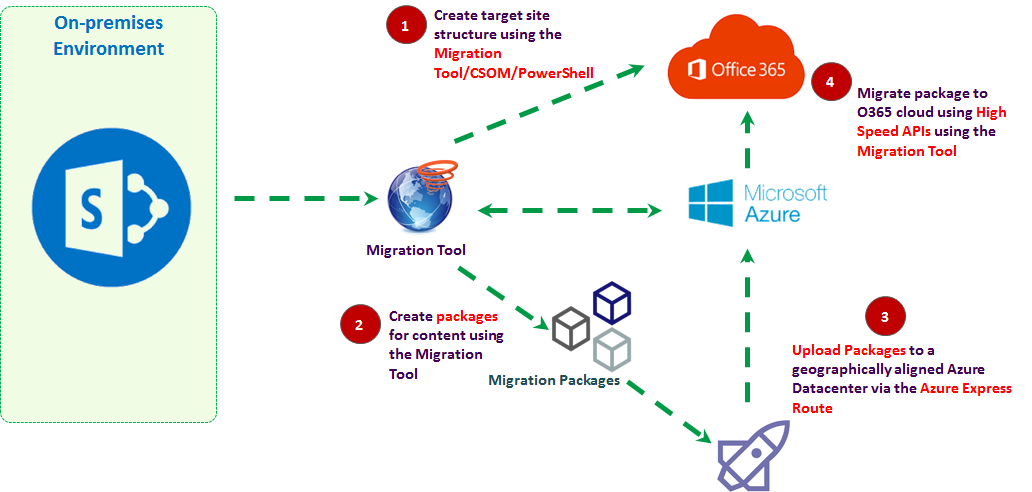
Windows 7 Service Pack 1, Windows 8, Windows Server 2008 R2 SP1, Windows Server 2008 Service Pack 2, Windows Server 2012, Windows Server 2012 R2

**Software:**

PowerShell 4.0

## **Prerequisites**

* An Azure subscription. [Azure Subscriptions](http://azure.microsoft.com/en-us/pricing/)
* An Azure Storage account.
* Provision your Office 365
* with your existing active directory [Office 365 integration with on-premises environments](https://support.office.com/en-US/article/Office-365-integration-with-on-premises-environments-263faf8d-aa21-428b-aed3-2021837a4b65)   
  OR
* Use one of the other options for adding accounts to Office 365 [Add users to Office 365 for business](https://support.office.com/en-us/article/Add-users-to-Office-365-for-business-435ccec3-09dd-4587-9ebd-2f3cad6bc2bc?ui=en-US&rs=en-US&ad=US).
* Download and install the SharePoint Online Management Shell MSI. Use the Control Pane to uninstall any previous versions.

**

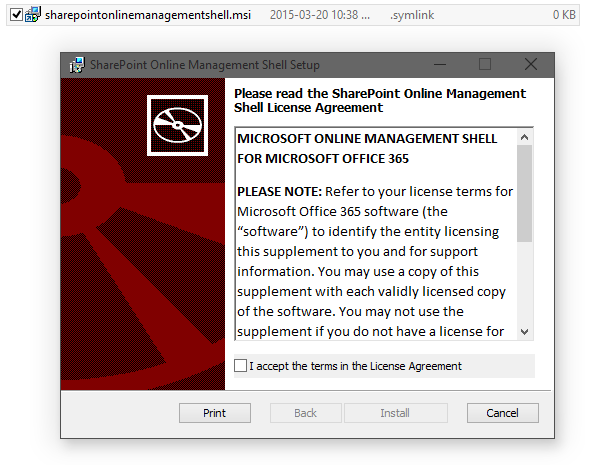
## **STEP 1 – Create Target Site Structure**

**Site structure** on the target environment including Sites/Subsites/Lists/Libraries etc is created via Migration Tool/CSOM/PowerShell.

## **STEP 2 – Create packages**

**Source Content** is packaged via PowerShell. Steps are as follows.

1. **Install SharePoint Online Management shell**



**Note:** Uninstall all previous versions. Open the SharePoint Online Management Shell as an Administrator.

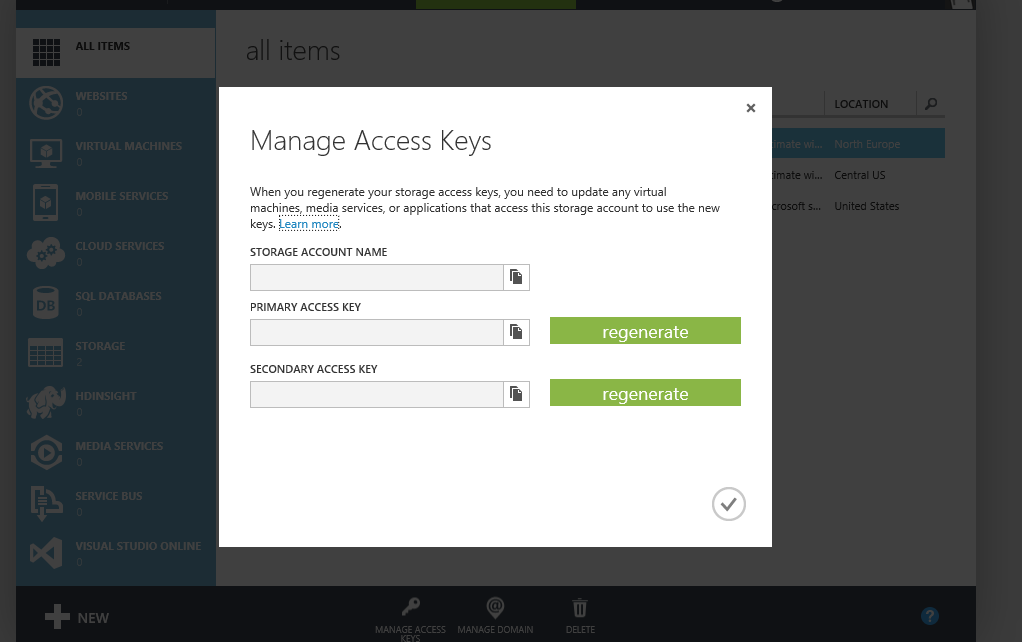
1. **Set up your working directory**

Create two empty folders before you start the migration process. These folders do not require a lot of disk space as they will contain only XML.

* Create a Temporary package folder
* Create a Final package folder

1. **Create an Azure Storage Account**

Keep Azure Storage account name and Key ready.



1. **Determine your locations and credentials**

$creds = (Get-Credential [admin@contoso.com](mailto:admin@contoso.com))

$sourceFiles = '\\fileshare\users\charles'

$sourcePackage = 'C:\migration\CharlesDocumentsPackage\_source'

$targetPackage = 'C:\migration\CharlesDocumentsPackage\_target'

$targetWeb = 'https://contoso-my.sharepoint.com/personal/charles\_contoso\_com'

$targetDocLib = 'Documents’

$azureAccountName = 'contosomigration’

$azureAccountKey = 'Insert Azure Account Key string Here’

$azureQueueName = 'migrationqueue’

1. **Create new content package**

After you determine your locations and credentials, the next step is to create a new migration package from a file share. The **New-SPOMigrationPackage** command reads the list of content targeted by the source path and will generate XML to perform migration.

There are two required parameters to enter (others are optional)

* **SourcefilesPath**: points to the content you want to migrate
* **OutputPackagePath**: points to your Temporary folder

**Sample Syntax:**

New-SPOMigrationPackage -SourceFilesPath $sourceFiles -OutputPackagePath $sourcePackage

1. **Convert the content package for your target site**

After you have created the content package, the **ConvertTo-SPOMigrationTargetedPackage** command converts the xml generated in your temporary folder. It saves a new set of targeted migration package metadata files to the target directory. This is the final package.

**Note**: Your target site collection administrator credentials are used to gather data to connect to the data site collection.

There are six required parameters to enter (others are optional)

* **TargetwebURL**: points to the destination Web
* **SourceFilesPath:** points to the you want to migrate
* **SourcePackagePath**: points to your Temporary package folder
* **OutputPackagePath**: points to your final package folder
* **TargetDocumentLibraryPath**: the path to your destination library
* **Credentials**: SPO credential that has admin rights to the destination site

**Sample Syntax:**

# Convert package to a targeted one by looking up data in target site collection

ConvertTo-SPOMigrationTargetedPackage -SourceFilesPath $sourceFiles -SourcePackagePath $sourcePackage -OutputPackagePath $targetPackage -TargetWebUrl $targetWeb -TargetDocumentLibraryPath $targetDocLib -Credentials

## **STEP 3 – Upload Package**

In this step, migration package is created in containers inside Azure storage using your storage account credentials using the **Set-SPOMigrationPackageAzureSource** command. The package files are uploaded to the container, a snapshot of the files completed, and then it returns the connection strings to a PowerShell variable.

There are five required parameters (others are optional)

* **SourceFilePath**: points to the content you want to migrate
* **SourcePackagePath**: points to your final package folder
* **AccountName**: the name of the Azure storage account
* **Accountkey**: the key to the Azure storage account
* **AzureQueueName**: The name of the Azure queue BLOB (optional but recommended)

**Recommendation*:*** Take the output of the following command and store it in a variable for the next command.

$temporaryVariable = Set-SPOMigrationPackageAzureSource

**Sample Syntax:**

# Create azure containers and upload package into them, finally snapshotting all files

$al = Set-SPOMigrationPackageAzureSource -SourceFilesPath $sourceFiles -SourcePackagePath $targetPackage -AzureQueueName $azureQueueName -AccountName $azureAccountName -AccountKey $azureAccountKey

$al|fl

## **STEP 4 – Submit to Migrate to O365**

The final step is to use the **Submit-SPOMigrationJob** command to create a new migration job in the target site collection, and returns a GUID representing the JobID.

There are three required parameters (others are optional)

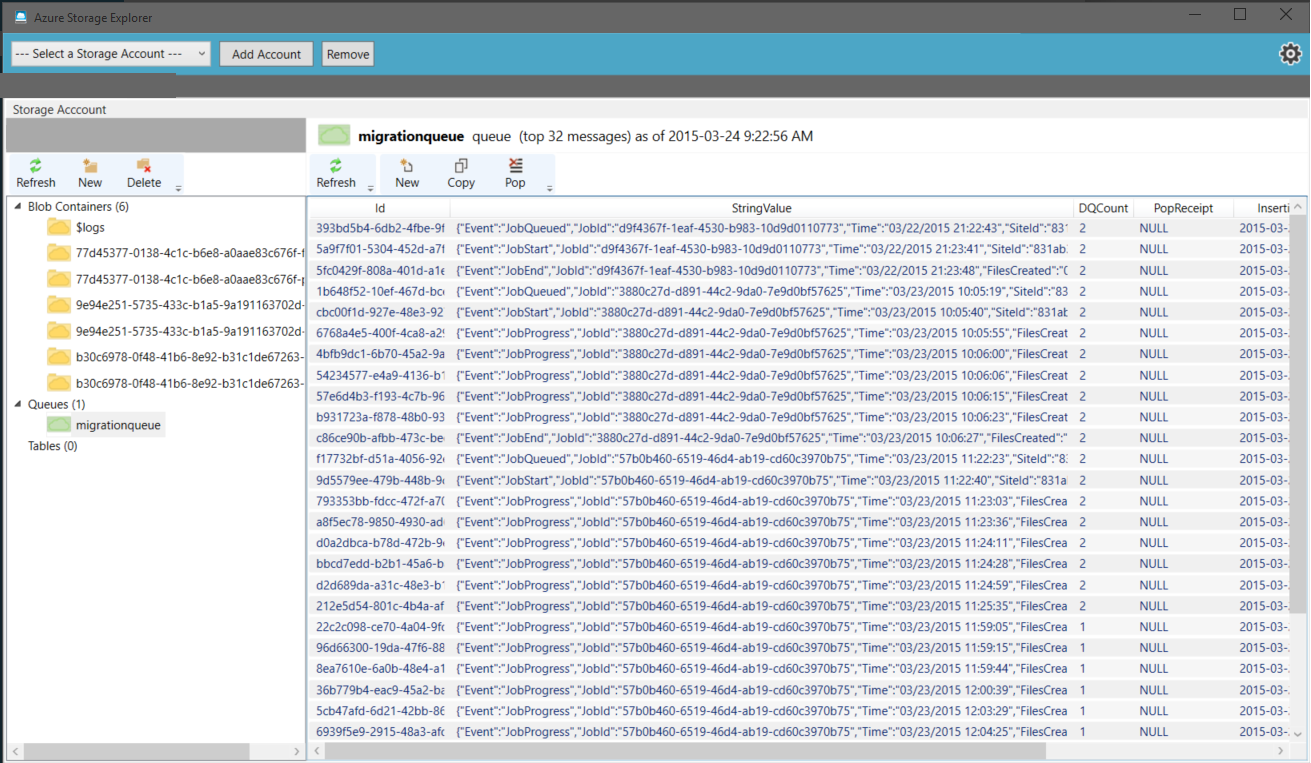
* **TargetwebURL**: point to the Web of the destination
* **MigrationPackageAzureLocations**: the temporary variable from the previous command
* **Credentials**: SPO credential that has admin rights to the destination site

**Sample Syntax:**

# Submit package data to site collection to create new migration job

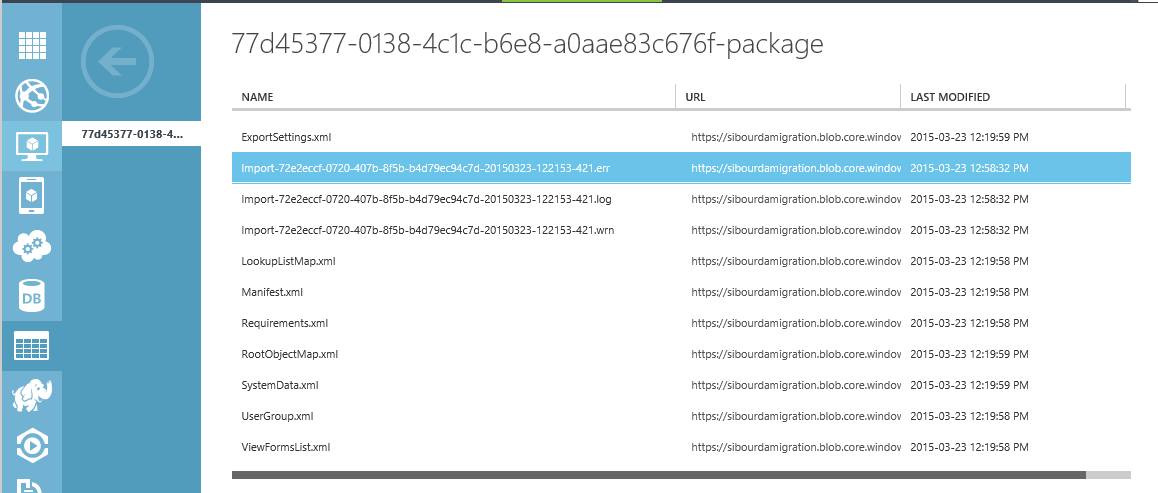
Submit-SPOMigrationJob -TargetWebUrl $targetWeb -MigrationPackageAzureLocations $al -Credentials $creds

Job Status can be checked in the Azure storage account queue. Real time updates will be posted there.



Manifest container in the Azure Storage can be checked for logs of everything that happened. It is now safe to delete those containers once migration is completed.

If there were errors or warnings, .err and .wrn files will be created in the manifest container.



# **References**

Please refer to the following URLs for further information on Disk Shipping and Azure Express Route options.

<https://azure.microsoft.com/en-in/documentation/articles/storage-import-export-service/>

<https://azure.microsoft.com/en-us/services/expressroute/>

<https://azure.microsoft.com/en-us/documentation/articles/expressroute-introduction/>