**Automated PST to Office 365 migration using MSPComplete, UploaderWiz and MigrationWiz**

Contents

[1. Description 3](#_Toc529810935)

[2. Requirements 3](#_Toc529810936)

[2. Usage 3](#_Toc529810937)

[3. Inputs and OutPUTS 4](#_Toc529810938)

[3.1. Upload-UW\_PstToAzureBlobContainer.ps1 4](#_Toc529810939)

[3.2. Create-MW\_AzureBlobContainerToO365 11](#_Toc529810940)

[3.3. Start-MW\_ AzureBlobContainerToO365Migrations.ps1 13](#_Toc529810941)

# Description

This document will outline the necessary requirements, usage, inputs and outputs when attempting to execute the scripts to automate the PST file migration process in MSPComplete and MigrationWiz by deploying our desktop agent UploaderWiz via email.

The migration process that is completely automated by these scripts is explained in this [PST to Office 365 Migration Guide](https://help.bittitan.com/hc/en-us/articles/115008258948-PST-to-Office-365-Migration-Guide-)

# Requirements

* Windows Operating System
* [BitTitan Powershell SDK](https://www.bittitan.com/downloads/bittitanpowershellsetup.msi).msi
* Microsoft Windows PowerShell 4.0+
* Microsoft .NET Framework 3.5
* Microsoft .NET Framework 4.6.2
* The ability to run PowerShell as an administrator
* Powershell execution policy set to RemoteSigned or Unrestricted. If already set to Unrestricted do not change to RemoteSigned!
* Ability to create a folder on the root of C:\
* API access granted to BitTitan MSPC account executing the script
* An Azure Subscription and an Azure Storage Account
* 3 MSPComplete endpoints:
* Azure Subscription – To be able to connect to an Azure subscription and create a blob container under an existing Azure storage account
* PST – To be able to upload all the PST files to the created Azure blob container
* Office 365 – To be able to migrate to Office 365

# Usage

1. Unzip the *AutomatedPSTtoOffice365.zip* file into a directory.
2. Make sure the ***BitTitanFunctions.psm1*** is in the same directory as the scripts.
3. Open the BitTitan Command Shell
4. Change directory to the directory where the zip file was extracted into
5. Run the following scripts in this order:

**.\Upload-UW\_PstToAzureBlobContainer.ps1**

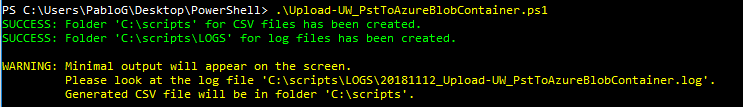
**.\Create-MW\_AzureBlobContainerToO365.ps1**

**.\Start-MW\_ AzureBlobContainerToO365Migrations.ps1**

# Inputs and OutPUTS

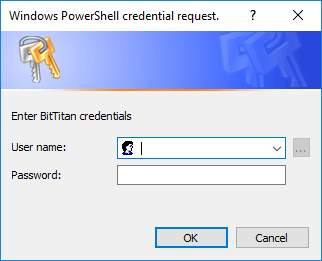
## Upload-UW\_PstToAzureBlobContainer.ps1

The first time the script *Upload-UW\_PstToAzureBlobContainer.ps1* is executed, it creates the folder *C:\scripts* to place all the CSV files generated during the script execution and a subfolder *C:\scripts\LOGS* for all the log files:

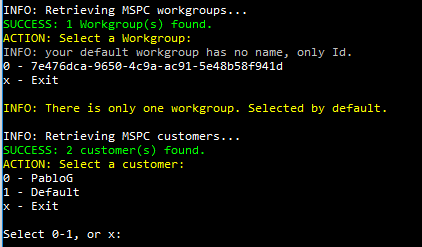


After that, the script prompts for the BitTitan credentials to connect to MSPComplete platform and retrieve two endpoints of type *AzureSubscription* and *PST*:

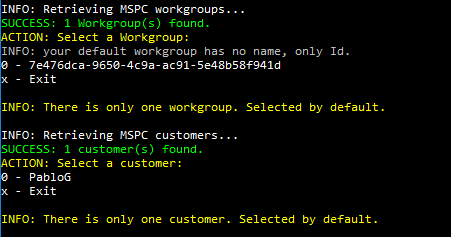




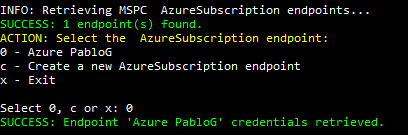
Once authenticated, it retrieves all the Workgroups created by the user and all where the user was invited to. The user will have to select the workgroup that contains the Customer (the default Workgroup does not has name but only ID) and after that, the script displays all the Customers under the selected Workgroup:



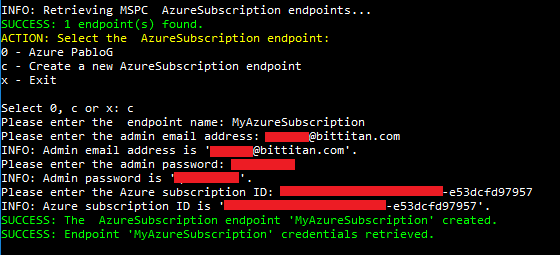
If there is only one workgroup, it gets selected automatically. The same for the Customer, if there is only one customer, it gets selected automatically:



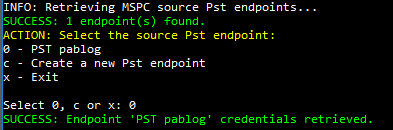
After selecting the Customer, the script display all *AzureSubscription* endpoints:



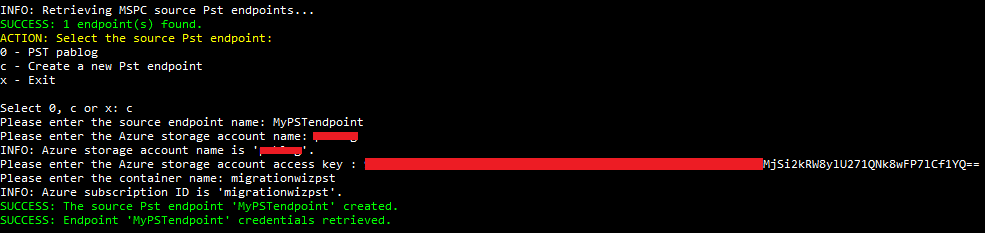
Select your endpoint or create a new one from the shell:



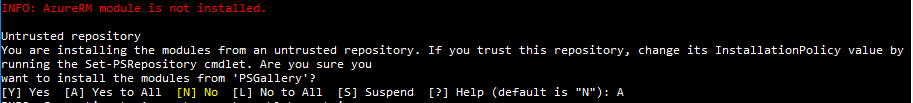
After that, the script displays all *PST* endpoints:



Select your endpoint or create a new one from the shell:



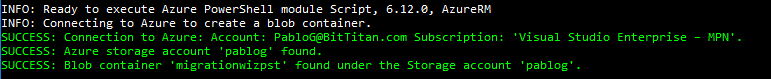
The script retrieves the credential from the *AzureSubscription* endpoint to connect to Azure. It firstly checks if the AzureRM modue is installed. If not, it downloads, installs and import the module:



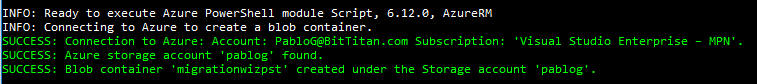
If the AzureRM module is already installed, the script is ready to execute Azure PowerShell:



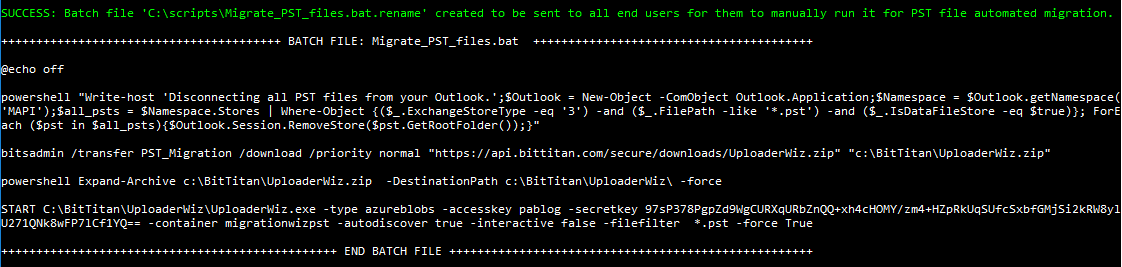
It connects to the Azure subscription, logs into the Azure storage account and tries to find a blob container called ‘migrationwizpst’:



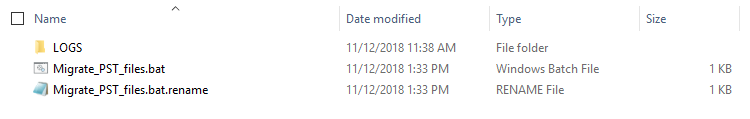
If there is no such blob container, it creates a new one.



With the blob container created, the scripts creates a bat file with the **.bat.rename** extension just to avoid being blocked by any security policy in your organization when sending the batch file as an email attachment:

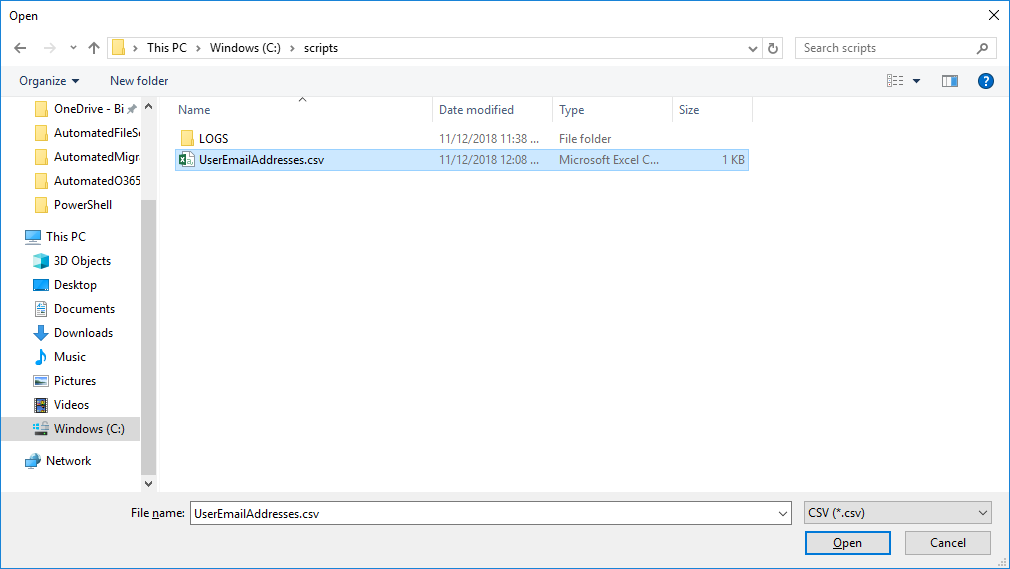


The batch file is placed under *C:\scripts:*

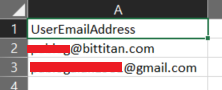
The script asks then if you want to send an email to all end users whose PST files must be migrated



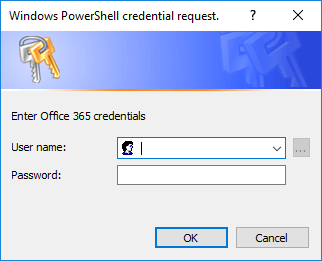
If you enter ‘y’, it opens a File Dialog window for you to select a CSV file with all the users email addresses:



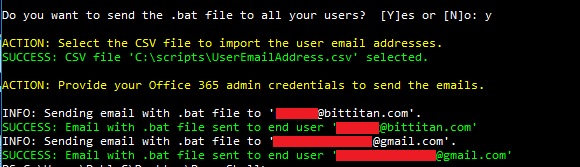
The CSV file must have a column *UserEmailAddress* with all the end users’ email addresses you want to send the batch file to (in the zip file there is a sample CSV file):



It will prompt for your Office 365 global admin credentials because the emails are going to be sent from Office 365:



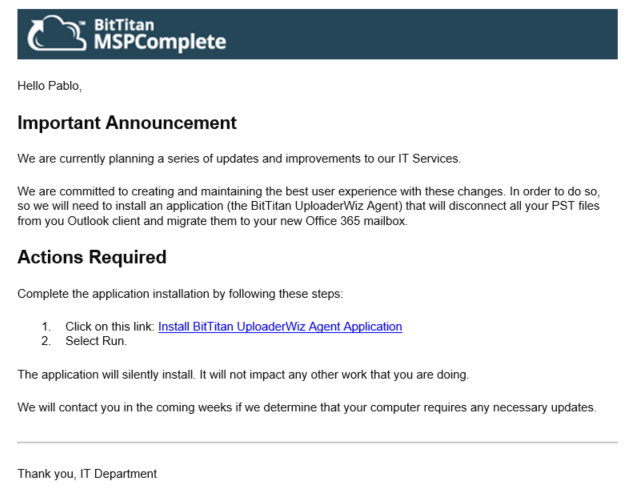
Once you provide the global admin credentials, it will send an email with the .bat filled attached to each of the users in the CSV file:



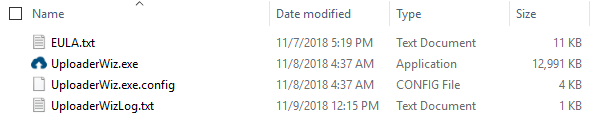
This is the email with subject

**Action required: Install the BitTitan UploaderWiz Agent on your computer**

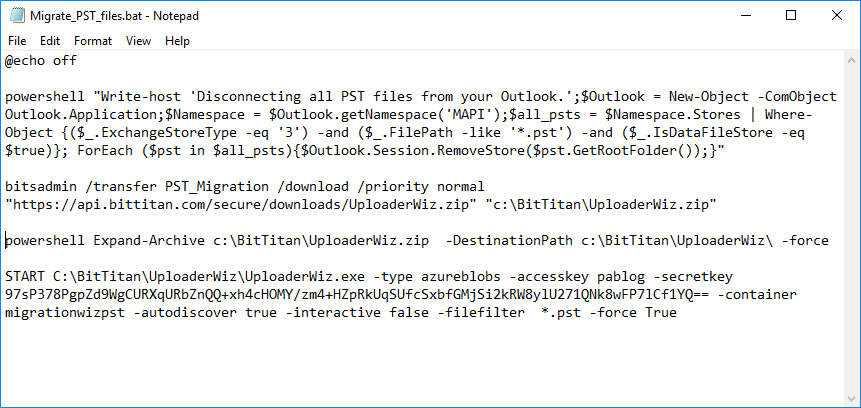
to be sent to all end users:



The end user will have to click on the link to save the *Migrate\_PST\_files.bat* file and click on the batch file. The batch file is going to disconnect all the PST files from the Outlook client, is going to download the latest version of the UploaderWiz agent and it is going to unzip it into the *UploaderWiz* folder in the folder containing the scripts:

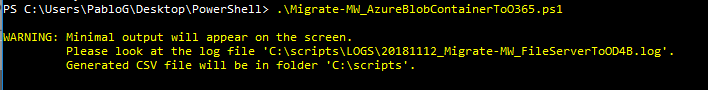


Finally it is going to run UploaderWiz with all the required parameters to discover all PST files across all hard drives and upload them to the Azure blob container created previously.

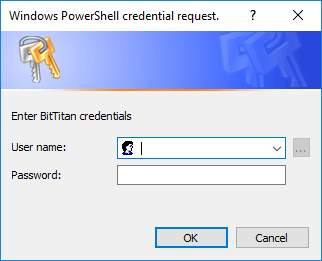


## Create-MW\_AzureBlobContainerToO365

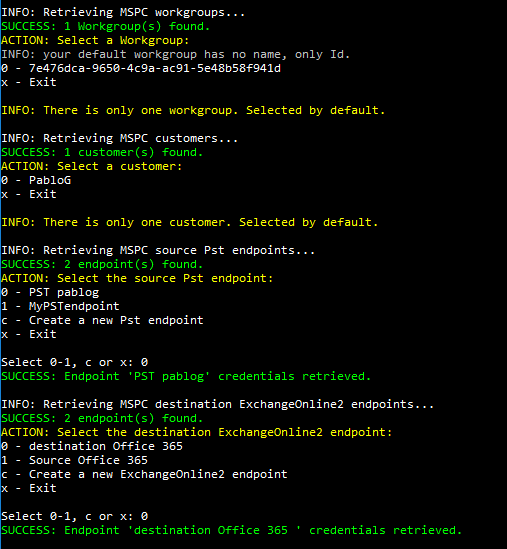
This script creates a MigrationWiz personal archive project to migrate the PST files from the Azure blob container to the destination Office 365 mailboxes:



The script starts prompting for the BitTitan credentials:



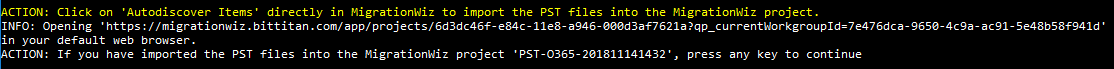
Once authenticated, it retrieves all the Workgroups created by the user and all where the user was invited to. The user will have to select the workgroup that contains the Customer (the default Workgroup does not has name but only ID) and after that, the script displays all the Customers under the selected Workgroup. If there is only one workgroup, it gets selected automatically. The same for the Customer, if there is only one customer, it gets selected automatically. The user then has to select a source PST Endpoint and a destination Office 365 (*ExchangeOnline2*) endpoints:



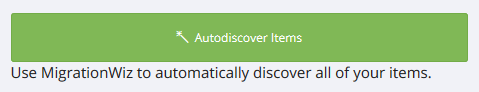
Once both endpoints have been selected, the script creates an empty MigrationWiz archive project:



At this moment, the script automatically launches your default web browser to open the MigrationWiz project

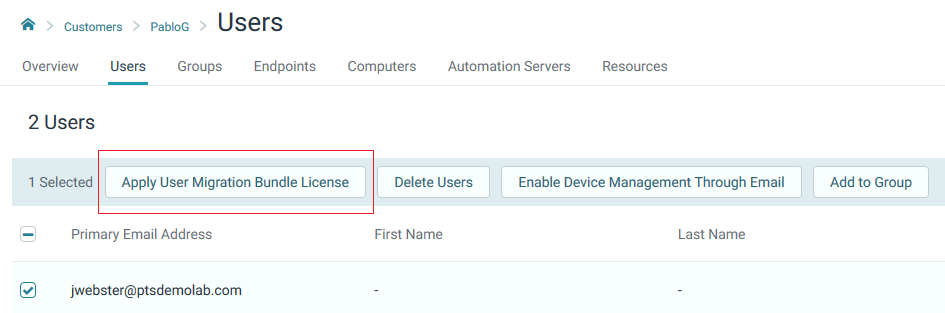


Because once the upload is completed you have to import all the PST files from the Azure blob container into the MigrationWiz project by clicking on the *Autodiscover Items* button:

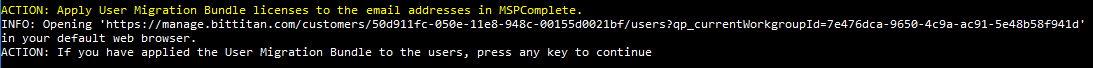


Press any key to continue once you have imported all the PST files into the project.

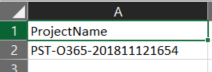
The script then automatically launches another tab in your default web browser to open the MSPComplete platform. You have to select the users you want to migrate and apply an User Subscription Bundle:



After that go back to the script and press any key to continue:

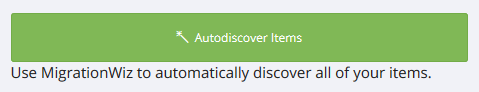


The name of the project just created is stored in a CSV file that will be used by the next script *Start-MW\_ AzureBlobContainerToO365Migrations.ps1* to submit all the migrations automatically:

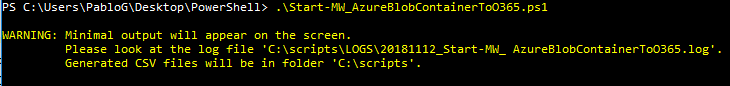


## Start-MW\_ AzureBlobContainerToO365Migrations.ps1

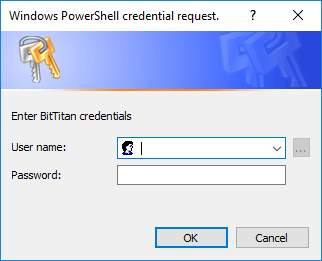
Before running this script, make sure you have imported all PST files from the Azure blob container into the MigrationWiz project created by the previous script *Create-MW\_AzureBlobContainerToO365.ps1* by clicking on the on the *Autodiscover Items* button:



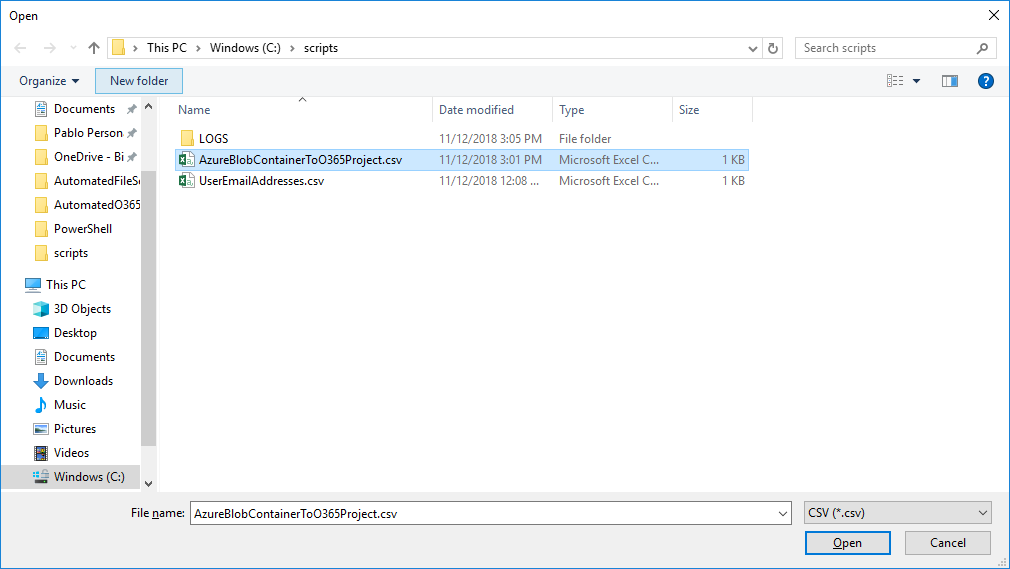
This script controls the submission of all migrations added to the MigrationWiz project*.* **The script assumes that all migration items have corresponding User Migration Bundle licenses applied before running the script.** If not, purchase and [apply a User Migration Bundle](https://help.bittitan.com/hc/en-us/articles/115014263488-Apply-User-Migration-Bundle-licenses-to-the-Customer-s-Users) per each Office 365 mailbox.



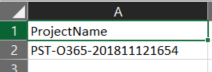
The script starts prompting for the BitTitan credentials:



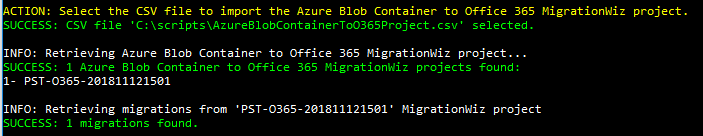
After that, it prompts for CSV file *AzureBlobContainerToO365Project.csv* created by the script *Create-MW\_AzureBlobContainerToO365.ps1* by opening a File Dialog window:



The CSV file contains the name of MigrationWiz project containing the PST to Office 365 mailbox migrations to be submitted:



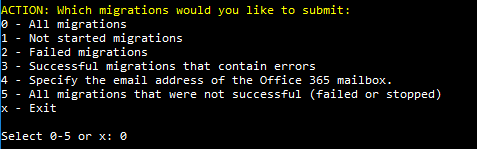
The script reads the MigrationWiz project name from the CSV file and retrieves all the migrations added to the project:



And it asks which migrations must be submitted:

1. All migrations
2. Not started migrations
3. Failed migrations
4. Successful migrations that contain errors
5. Specify the email address of the Office 365 mailbox
6. All migrations that were not successful (failed or stopped)

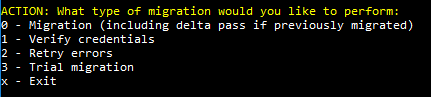
For example, you could enter ‘0’ the first time all the PST files are being submitted for migration.



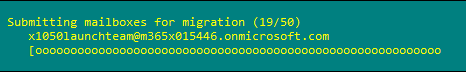
After that, select the type of migration pass you want to submit:

1. Migration (including delta pass if previously migrated)
2. Verify credentials
3. Retry errors
4. Trial migration

You can start with a *Verify credentials*,



And the script submits all migrations for verify credentials:



After displaying how many migrations were successfully submitted



The menu is displayed again, infinitely until you enter ‘x’ for exit.