Move tenant to a new subscription

Version 2 | March 2017

Contents

[Description 2](#_Toc452626644)

[Requirements 2](#_Toc452626645)

[How does it works? 2](#_Toc452626646)

[Export script 3](#_Toc452626647)

[PARAMETERS 3](#_Toc452626648)

[Example 3](#_Toc452626649)

[Import script 3](#_Toc452626650)

[Requirements 3](#_Toc452626651)

[Parameters 4](#_Toc452626652)

[Example 5](#_Toc452626653)

# Description

This set of scripts is used to export an account and import it in another subscription of the OMNIA platform.

The account migration process is based on 2 moments. First, you export the Data from the origin account and second, you import it to the destination account.

This is used with 2 scripts, import.ps1 and export.ps1.

The user should have access to the source and destination subscription.

# Requirements

Be sure that:

* you can access to the Sql Database from your current network
* you have the Azure Powershell installed
* you have the AzCopy (version 5.0.0.0) installed at:
  + %programfiles(x86)%\Microsoft SDKs\Azure\AzCopy
* you have co-admin rights to the destination Resource Group
* you are running powershell as Admin

Execution time is improved by running from an Azure Virtual Machine (blobs/sql never have to leave the Azure environment).

# How does it work?

* Export:
  + Export database using the [BCP Utility](https://msdn.microsoft.com/en-us/library/ms162802.aspx)
  + Extract the Tenant blobs using Azure Powershell
  + Extract the tenant image blobs using Azure Powershell
  + Extract the tenant users images using Azure Powershell
  + Extract the Tenant Table Storage using [azCopy](https://azure.microsoft.com/pt-pt/documentation/articles/storage-use-azcopy/)
  + Create an SQL Script (UsersInsertStatements.sql) to create users in the destination system.
  + The Exported files will be placed in the subfolders of the Exported folder
* Import:
  + Invoke the destination system API to create a new Tenant
  + Import data to Sql Database using the BCP utility
  + Using the powershell (Azure RM) imports:
    1. Images
    2. Blobs (Scripts, Reports, Commands)
  + Import table storage using azCopy
  + Run the script UsersInsertStatements.sql to create users in the destination system
  + Rebuild the Database Indexs

# Process Guidelines

* The migration should be scheduled with the customer previously – at the start of the process, the original tenant should be set to Inactive, and the connected connectors turned off.
* The migration should be performed on an Azure virtual machine, ideally on the same data center the subscriptions are in, as to save money and increase performance due to reduced latency.
* The infrastructure of the subscriptions can be improved, namely the database, to improve performance.
* The Connectors won’t be imported. After moving the account, the tenant admin should:
  + Recreate all the connectors in the tenant that existed in the original;
  + Download the license file and connector code, and configure all installed Omnia Connectors to point to the new URL and use the new files.
* Only the users existing in the source application will have privileges in the destination account. Example: If the Master Account doesn’t exist in the source tenant, the master account won’t have privileges in the destination account.
* The Source and Destination systems should be in the same Platform Version
* Tenants are **always** created as Demo type. Ensure you switch them to the correct type when the migration is finished, if they are Full or Template in the origin account.

## Export script

### Requirements

Before executing, you should be authenticated via Azure RM:

Login-AzureRmAccount

### Parameters

**tenant [text]**

The tenant code (GUID).

**WebsiteName [text]**

The full URL of the Azure website (https:\\xxx.azurewebsites.net format)

**SubscriptionName [text]**

The name of the Azure subscription (analogous to other -SubscriptionName in Azure Powershell)

**ResourceGroupName [text]**

The name of the Azure resource group (analogous to other -ResourceGroupName in Azure Powershell)

### Example

.\export.ps1 -tenant A0000000-B111-C222-D333-E44444444444 -WebsiteName https:\\waomnia12345.azurewebsites.net -SubscriptionName omnia12345 -ResourceGroupName omnia12345

## Import script

### Requirements

Before executing, you should be authenticated via Azure RM:

Login-AzureRmAccount

### Parameters

**tenant [text]**

The tenant code (GUID).

**WebsiteName [text]**

The full URL of the Azure website (https:\\xxx.azurewebsites.net format)

**SubscriptionName [text]**

The name of the Azure subscription (analogous to other -SubscriptionName in Azure Powershell)

**ResourceGroupName [text]**

The name of the Azure resource group (analogous to other -ResourceGroupName in Azure Powershell)

**shortcode [text]**

The tenant short code, that will be created in the destination subscription

**tenantname [text]**

The tenant name, that will be created in the destination subscription

**maxNumberOfUsers [int] (Optional – Default value 10)**

The maximum number of users that can be created in the new account

**subGroupCode [text] (Optional – Default value “DefaultSubGroup”)**

The Sub Group Code that the tenant will be part of. The sub group should already exist in the destination account

**tenantAdmin [text]**

The user that will be created as the tenant Admin

**tenantAdminPwd [text] (Optional)**

The password for the user that will be created as the tenant Admin. If omitted, user will be associated to the tenant, if they exist; otherwise, the password will be randomly generated and sent to their email (should be valid)

**oem [text]**

The code of the OEM the tenant will be assigned to. The OEM should already exist in the destination account

**master [text]**

An user with Administrator Role in the destination subscription

**masterpwd [text]**

The password of the user defined in ***master***.

### Example

.\import.ps1 -tenant A0000000-B111-C222-D333-E44444444444 -WebsiteName https:\\waomnia12345.azurewebsites.net -SubscriptionName omnia12345 -ResourceGroupName omnia12345 -shortcode tenantshortcode -tenantname 'My Tenant Name' -maxNumberOfUsers 10 -subGroupCode DefaultSubGroup -tenantAdmin admin@admin.com -tenantAdminPwd Password0 -oem omnia -master admin@admin.com -masterpwd Password0