



Active Directory Scripting

Summary: Creation of scripts for basics actions.

Version: 1.00

Contents

I	Preamble	2
II	Rules	3
III	Mandatory part	4
III.1	AD installation scripts	6
III.2	Data base scripts	6
III.3	User scripts	7
III.4	Group scripts	8
IV	Submission and peer-evaluation	10

Chapter I

Preamble

Active Directory (AD) work from graphical interface, like you experienced previously, but it also work perfectly by scripting action via a powerfull tool : PowerShell !

Time to experiment with PowerShell to control your AD installation !

In this subject, you will have to create a script for pretty much every action you made by clicking on a button during the last project. Why click on a screen when you can type hundreds of characters, and be pround of your beautiful result ?!

No ? I'm the only one who like scripts ? :'(

You will start to love it too, after this !

Chapter II

Rules

- Every action in the subject/evaluation must be executed from a PowerShell terminal
- Every script must be commented to explain what he's doing
- You should find a way to open a pop-up to ask the user informations he may need to give you

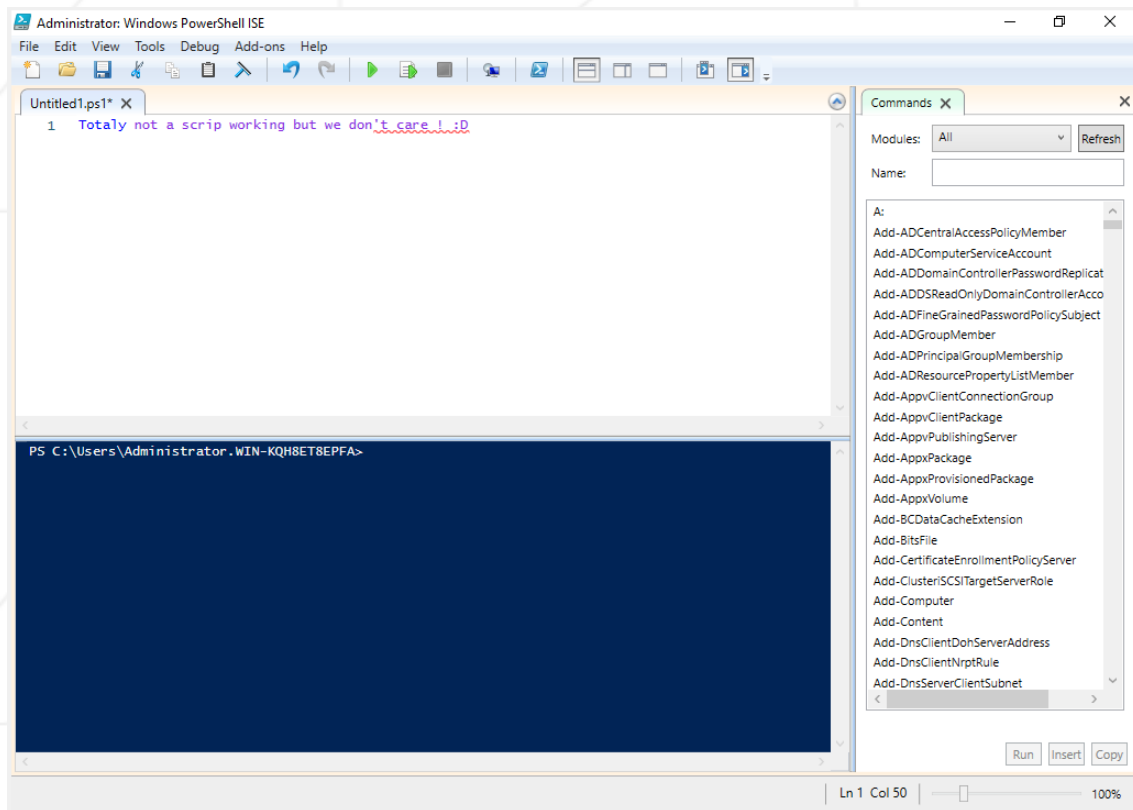
Chapter III

Mandatory part

Domolia is very happy with the installation you have done of Active Directory, but would like to have a "simpler" access to the features that the tool offers.

Domolia would like to have several scripts to perform operations on Active Directory. Therefore, she has provided you with the following list of action to automatize:

- AD installation and dependency
- Server promotion via drill creation
- Server promotion via existing domain connection
- Database backup
- Database loading
- User creation
- Reset password
- Modification of an attribute off a user
- Recovering information from a user
- Retrieve information from all users
- Creation of a group
- Modification of an existing group
- Creation of a distribution group
- Add a user to the group
- Remove a user from a group
- Import members of group A into group B
- Retrieve information from a group
- List the members of a group
- Retrieve information from all groups



You should found the PowerShell ISE, a tool given by Microsoft.
It's a powerfull editor devoted to PowerShell script.

III.1 AD installation scripts

Name	ADPackageInstaller.ps1
Description	Install ActiveDirectory and every dependencies needed for the domain controller role
Parameter	No parameter needed

Name	CreateNewForestDomainController.ps1
Description	Promote an AD server to Domain Controller by creating a new forest
Parameter	- DomainAddress - NetbiosName

Name	JoinExistingDomainController.ps1
Description	Promote an AD server to Domain Controller by joining an already existing domain controller
Parameter	- DomainAddress

III.2 Data base scripts

Name	SaveDataBase.ps1
Description	Create a data base to store every user and group from the Domain Controller
Parameter	- Path to save the result .CSV file - Desired delimiter - An undefined amount of parameter to compose the data base

Name	LoadDataBase.ps1
Description	Load a data base from a saving file
Parameter	- Path to .CSV file to load - File delimiter

III.3 User scripts

Name	UserCreation.ps1
Description	<p>In hashtable or in specification, don't care</p> <p>We assume that</p> <ul style="list-style-type: none">- mail address : name.surname@domainName.com- Basic password : TotallyN0tSecure -> Change at the first co- UserPrincipalName : email address <p>The default password must not be written in clear text in the script</p>
Parameter	<ul style="list-style-type: none">- Account name- Organisation Unit to join- Desired group

Name	ResetUserPassword.ps1
Description	Reset the password of the desired user
Parameter	<ul style="list-style-type: none">- Account name

Name	EditUserAttribute.ps1
Description	Modify an attribute of the user and define it at a new value
Parameter	<ul style="list-style-type: none">- Account name- Attribute name- Desired value

Name	ReadUserInformation.ps1
Description	Retreive user information from the server
Parameter	<ul style="list-style-type: none">- Account name- Filter the attribute to retrieve

Name	ReadDataBaseInformation.ps1
Description	Retreive every users informations from the server
Parameter	<ul style="list-style-type: none">- Account name- Filter the attribute to retrieve

III.4 Group scripts

Name	CreateGroup.ps1
Description	Create a new group
Parameter	- Group name - Organisation unit - Group scope - Description

Name	ModifyGroup.ps1
Description	Edit a group by modifying one attribute to a desired value
Parameter	- Group name - Attribute to edit - New attribute value

Name	ListUserInGroup.ps1
Description	Retreive an exhaustive list of user in the group
Parameter	- Group name

Name	CreateDistributionGroup.ps1
Description	Creation of a new distribution group for emailing multiples users at once
Parameter	- Group name - Organisation unit - Group scope - Description

Name	AddUserToGroup.ps1
Description	Add a user in the desired group. The script must block the adding of an unknown user
Parameter	- User name - Group name

Name	RemoveUserToGroup.ps1
Description	Remove a user from the desired group. The script must block the removal of an unknown user or a user that wouldn't be in the group
Parameter	- User name - Group name

Name	ImportGroup.ps1
Description	Import the content of a group inside another group
Parameter	- Origin group name - Destination group name

Name	ReadGroupInformation.ps1
Description	Retreive information(s) about a group. If no property name are given, the script must retrieve all properties
Parameter	- Group name - Optionnal : a property name

Name	ReadEveryGroupInformation.ps1
Description	Retreive information(s) from every group in the domain. If no property name are given, the script must retrieve all properties
Parameter	- Optionnal : a property name

Chapter IV

Submission and peer-evaluation

You have to turn every .ps1 script you have created.

The evaluator will only allow usage of those scripts.

For the peer-to-peer evaluation, you must prepare some empty from work VMs, on which the evaluator may try some of your script.



You may prepare those VMs by creating a shared folder, to just move scripts in it, instead of manually duplicate it from the newly cloned repo inside the VM.

You do not need to push any signature.

As the MV must be empty from work, you and the evaluator will install everything needed for this correction by using your script.