cimitra

Active Directory Computer Object Management Scripts & Powershell Tutorial

How to Use This Tutorial

The tutorial was designed to provide practical application and learning. You will learn how to create Powershell scripts to accomplish some basic Active Directory tasks.

By completing this tutorial, you will have created the following Powershell scripts to accomplish the following tasks within Active Directory:

- Create a computer object
- Remove a computer object
- List all computer objects
- Rename a computer object

On the next page is the tutorial's table of contents.The first column of the table of contents lists the topics. The second column provides hyperlinks to each topic's learning resources.

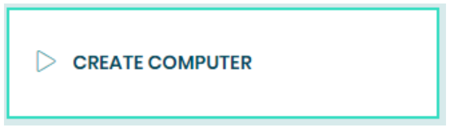
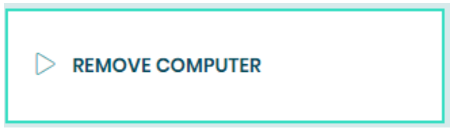
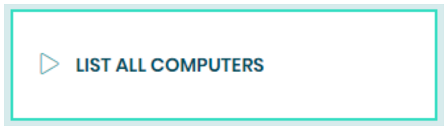
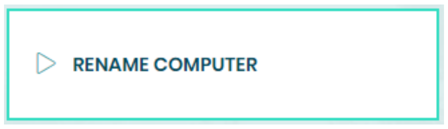
NOTE: Test run many of the scripts mentioned in this tutorial at:
https://app.cimitra.com
User: ad@cimitra.com
Password: 123
All the scripts can be run from Cimitra Apps in the **ACTIVE DIRECTORY | COMPUTERS** Folder

Questions or suggestions? Email tay@cimitra.com.

Table of Contents

Topic	Documentation

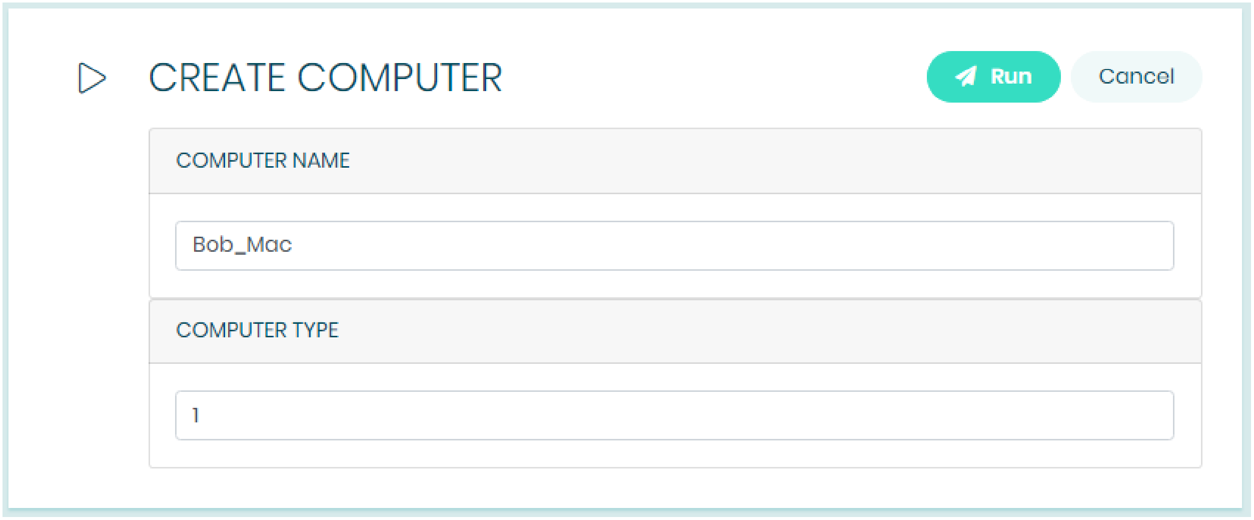
All scripts can be downloaded using one of two methods	**Download Scripts Here - Zip file (HTTP)**
	
	Or:
	
	`git clone https://github.com/cimitrasoftware/powershell_scripts.git`
	
	NOTE: Many of these scripts were updated in July of 2020. The newer versions of these scripts is located at:
	
	`https://github.com/cimitrasoftware/new-powershell-scripts`

	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
Calling External PowerShell Scripts	

Creating a Computer Object in Active Directory

Use Case

New computers come in regularly and need to be registered in Active Directory. The IT Active Directory specialist has now given a group of people on the help desk and in the receiving department rights to add computers into Active Directory using Cimitra.

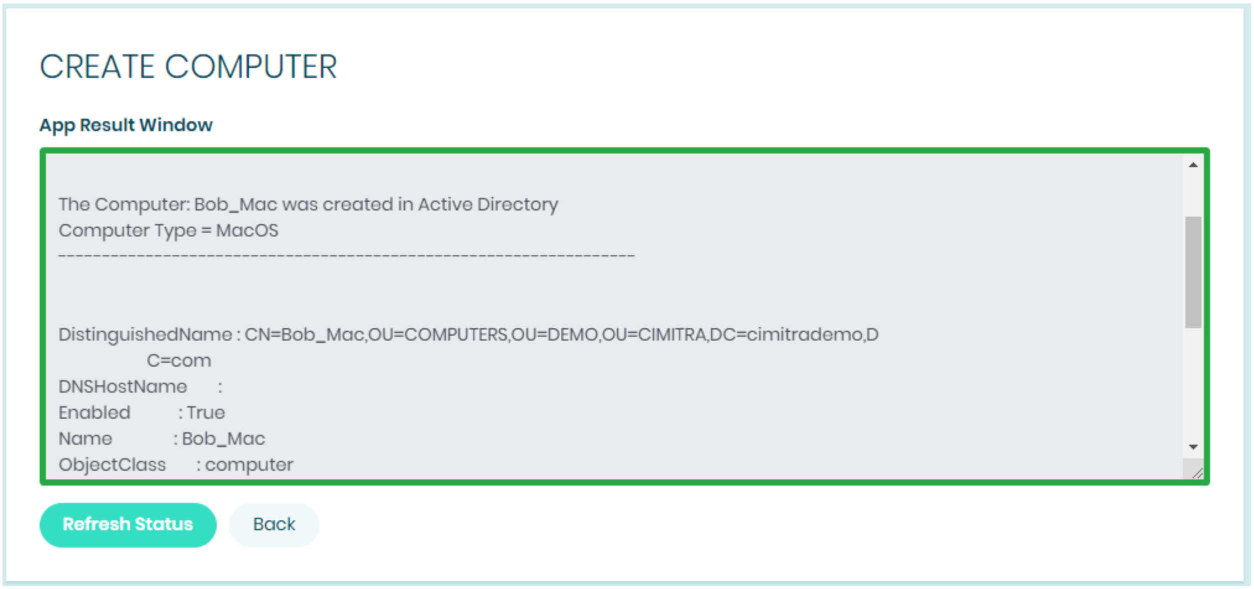

CREATE COMPUTER

COMPUTER NAME

Bob_Mac

COMPUTER TYPE

1


CREATE COMPUTER

App Result Window

```
The Computer: Bob_Mac was created in Active Directory
Computer Type = MacOS
-----------------------------------------------------------------


DistinguishedName : CN=Bob_Mac,OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,D
            C=com
DNSHostName    :
Enabled        : True
Name           : Bob_Mac
ObjectClass    : computer
```

Technical Overview

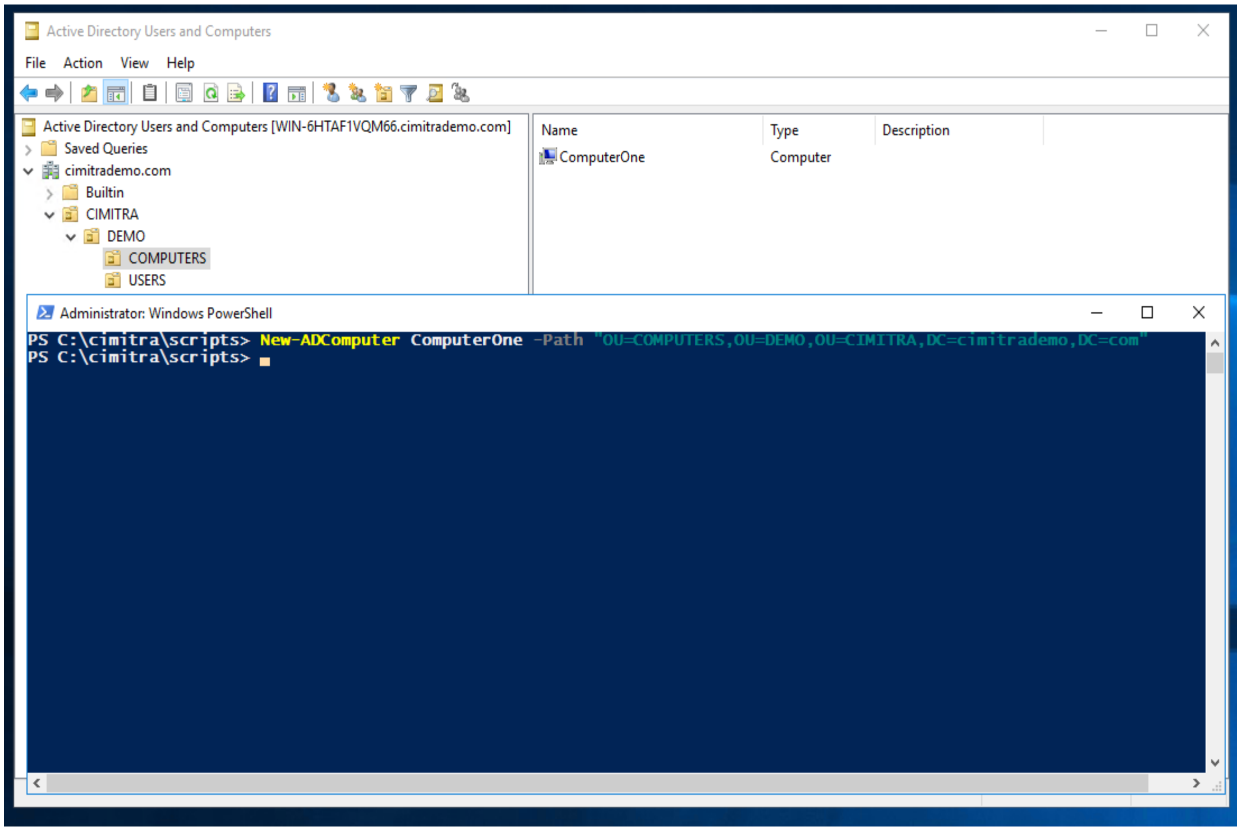
This PowerShell script will create a computer object in Active Directory at a specific location in Active Directory at the path. The type of computer will also be configurable. This script will be explained in 2 parts. Version 1 of the script shows how to create an object in Active Directory. Version 2 of the script allows for setting the computer type along with some nice usability and informational features.

The PowerShell Command and Parameters

Command	Parameter	Parameter Description
New-AdComputer	computer name	Provide the name you wish to give to the computer you are adding to Active Directory
	-Path	Provide the LDAP type to the container to which the computer will be added in the directory surrounded by double quotation marks.
Full Command Followed By An Example		
New-AdComputer [computer name] -Path "[full LDAP path to object's container]"		
New-AdComputer ComputerOne -Path "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"		

Creating the PowerShell Script

PowerShell ISE script editor is a great way to create and modify PowerShell scripts. YouTube author: **Robert McMillen** has a great **7-minute introduction to PowerShell ISE** that could be of help to you to know how to use the PowerShell ISE.


Active Directory Users and Computers
File Action View Help
Active Directory Users and Computers [WIN-6HTAF1VQM66.cimitrademo.com]
Saved Queries
cimitrademo.com
Builtin
CIMITRA
DEMO
COMPUTERS
USERS
Name Type Description
ComputerOne Computer
Administrator: Windows PowerShell

```
PS C:\cimitra\scripts> New-ADComputer ComputerOne -Path "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"
PS C:\cimitra\scripts>
```

The script will be a very simple, single command-line argument. Running the script will look like this:

```
NewComputer.ps1 ComputerOne
```

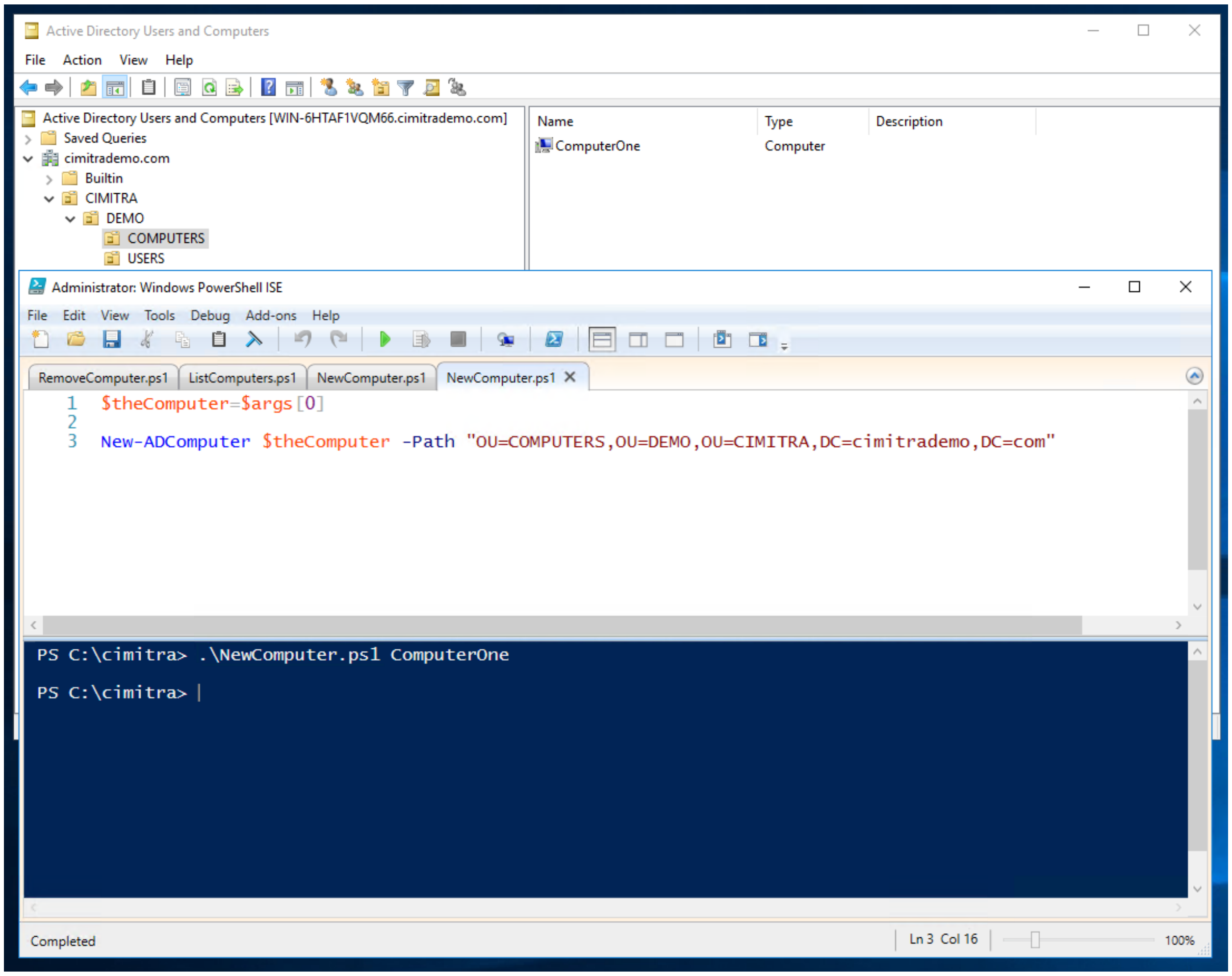
NewComputer.ps1: Version 1

# Take the first parameter passed into the script, store it as $theComputer
**$theComputer**=$args[0]

# Create a computer object, pass the $theCompter variable to the New-AdComputer command
New-ADComputer **$theComputer** -Path "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"

Here is an explanation of those two lines of the script:

Command	Explanation
**$theComputer**=$args[0]	Create a variable called **$theComputer** and assign the first parameter ($args[0]) passed into the script to **$theComputer**
New-ADComputer **$theComputer** -Path "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"	Use the PowerShell command, New-ADComputer, and pass the variable, **$theComputer**, which was created previously on the first line of the script. Also, indicate the LDAP path to create the computer object in with the command parameter -Path in this manner: -Path "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"



```
$theComputer=$args[0]

New-ADComputer $theComputer -Path "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"
```

```
PS C:\cimitra> .\NewComputer.ps1 ComputerOne

PS C:\cimitra> |
```

Adding an Attribute to a Computer in Active Directory

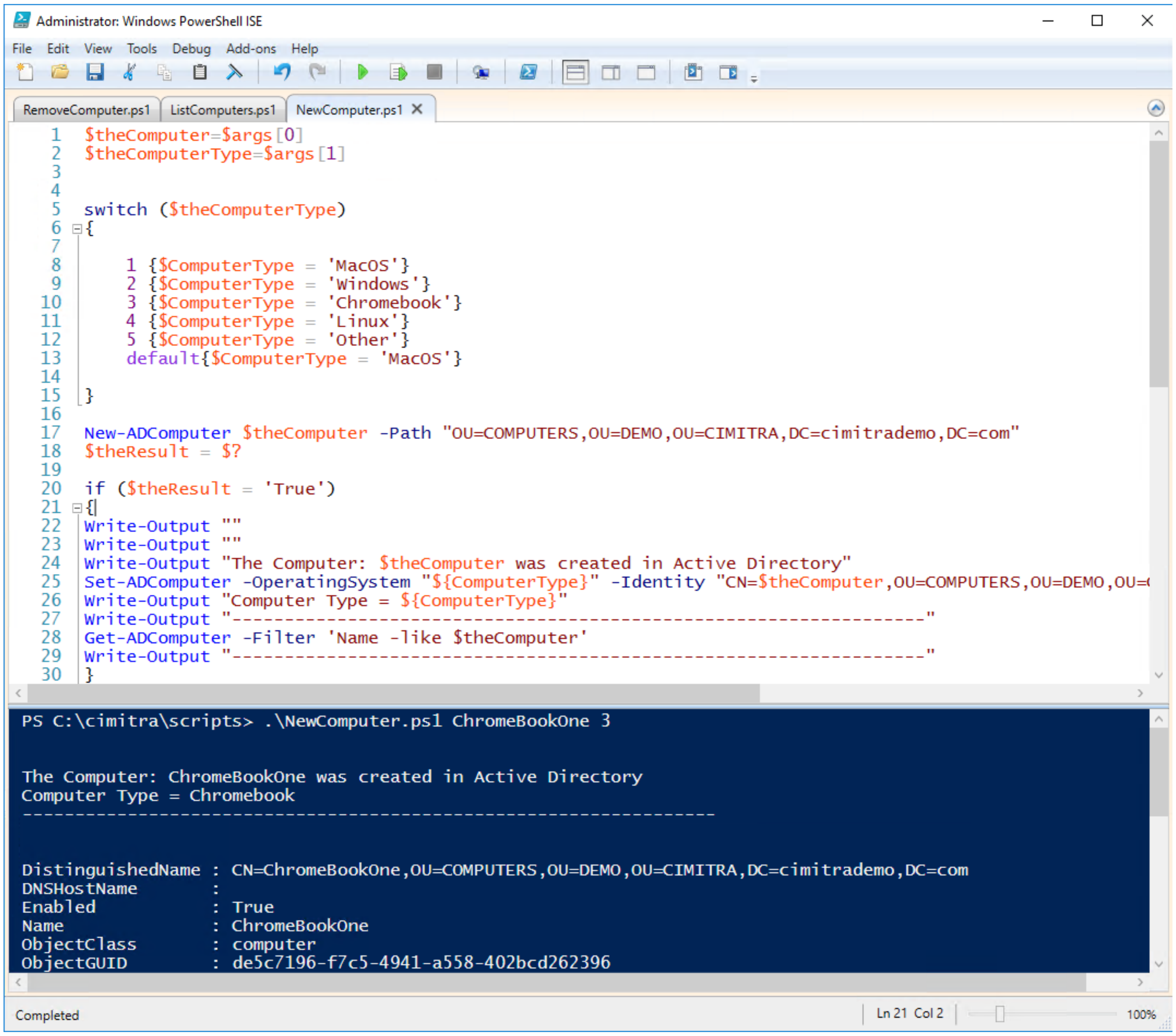
When a computer object is created in Active Directory, attributes can be added to that object. We will add one more command-line parameter to the **NewComputer.ps1** PowerShell script so we can very simply specify an operating system for the computer object. We have also added more functionality to the script. The original script (version 1) is combined with new additions in **bolded text** (version 2). Comments have also been added to explain each major section of the script.

Set-ADComputer is the new command that will define the contents of the **Operating System** value in Active Directory.

Script Contents for Copy/Paste

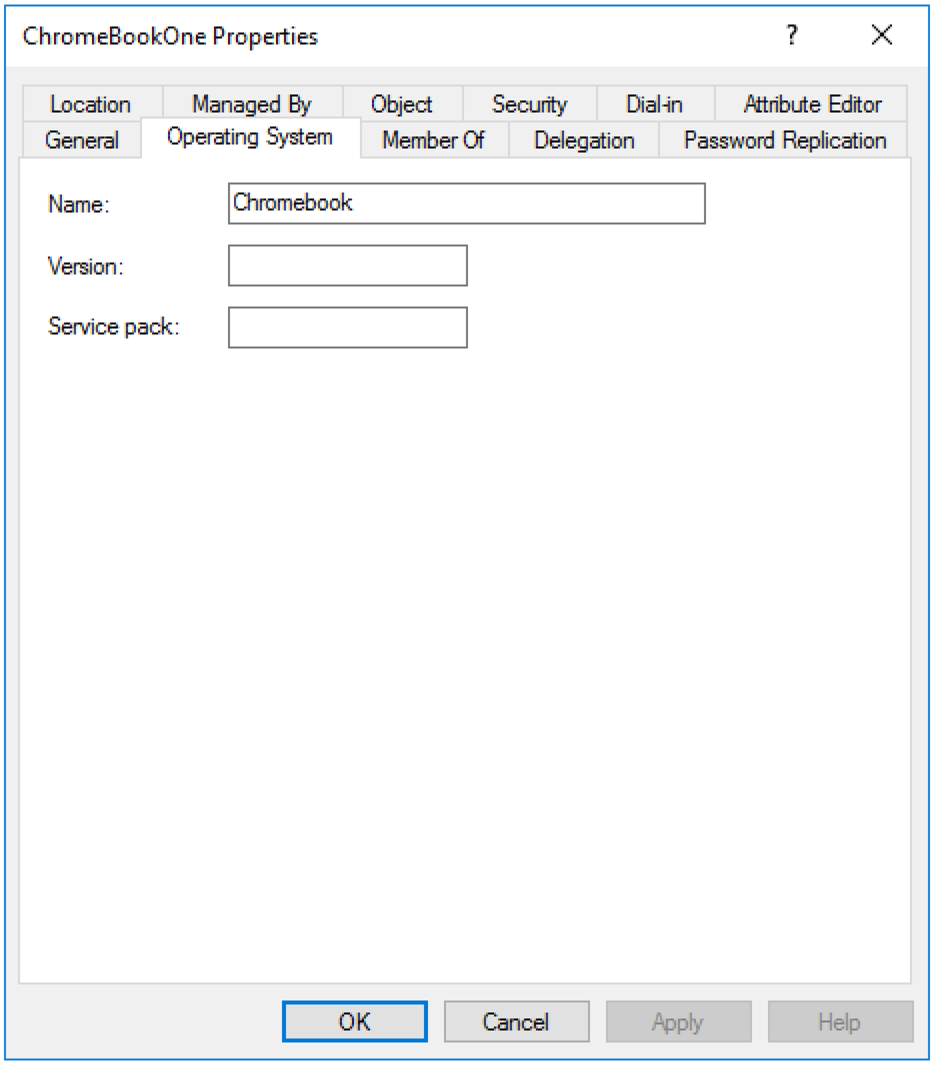
# Get the first command-line parameter

```
$theComputer=$args[0]
# Get the second command-line parameter
$theComputerType=$args[1]

# Correlate the number to a word with a switch statement
switch ($theComputerType)
{

   1 {$ComputerType = 'MacOS'}
   2 {$ComputerType = 'Windows'}
   3 {$ComputerType = 'Chromebook'}
   4 {$ComputerType = 'Linux'}
   5 {$ComputerType = 'Other'}
   default{$ComputerType = 'MacOS'}
}

# Add the Computer to Active Directory
New-ADComputer $theComputer -Path
"OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=co
m"

# Get the exit result from Active Directory
$theResult = $?

# If good result, display success, and update the OS
if ($theResult = 'True')
{
Write-Output ""
Write-Output ""
Write-Output "The Computer: $theComputer was created in Active
Directory"
Set-ADComputer -OperatingSystem "${ComputerType}" -Identity
"CN=$theComputer,OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimi
trademo,DC=com"
Write-Output "Computer Type = ${ComputerType}"
Write-Output "----------------------------------------------------"
Get-ADComputer -Filter 'Name -like $theComputer'
Write-Output "----------------------------------------------------"
}
```



```
$theComputer=$args[0]
$theComputerType=$args[1]


switch ($theComputerType)
{

    1 {$ComputerType = 'MacOS'}
    2 {$ComputerType = 'Windows'}
    3 {$ComputerType = 'Chromebook'}
    4 {$ComputerType = 'Linux'}
    5 {$ComputerType = 'Other'}
    default{$ComputerType = 'MacOS'}

}

New-ADComputer $theComputer -Path "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"
$theResult = $?

if ($theResult = 'True')
{
Write-Output ""
Write-Output ""
Write-Output "The Computer: $theComputer was created in Active Directory"
Set-ADComputer -OperatingSystem "${ComputerType}" -Identity "CN=$theComputer,OU=COMPUTERS,OU=DEMO,OU=
Write-Output "Computer Type = ${ComputerType}"
Write-Output "------------------------------------------------------------------"
Get-ADComputer -Filter 'Name -like $theComputer'
Write-Output "------------------------------------------------------------------"
}
```

```
PS C:\cimitra\scripts> .\NewComputer.ps1 ChromeBookOne 3


The Computer: ChromeBookOne was created in Active Directory
Computer Type = Chromebook
------------------------------------------------------------------


DistinguishedName : CN=ChromeBookOne,OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com
DNSHostName       :
Enabled           : True
Name              : ChromeBookOne
ObjectClass       : computer
ObjectGUID        : de5c7196-f7c5-4941-a558-402bcd262396
```



Name: Chromebook

Version:

Service pack:

Script Integration Into Cimitra

This assumes that you have already created a Cimitra server deployed a Cimitra Agent etc. to a Windows Server where the **NewComputer.ps1** script exists.

Create a new **Cimitra App** object and fill in the following properties as follows:

CIMITRA APP PROPERTIES

Property	Value
Platform	**Windows**
Agent	<The Cimitra Agent deployed to the Windows Server>

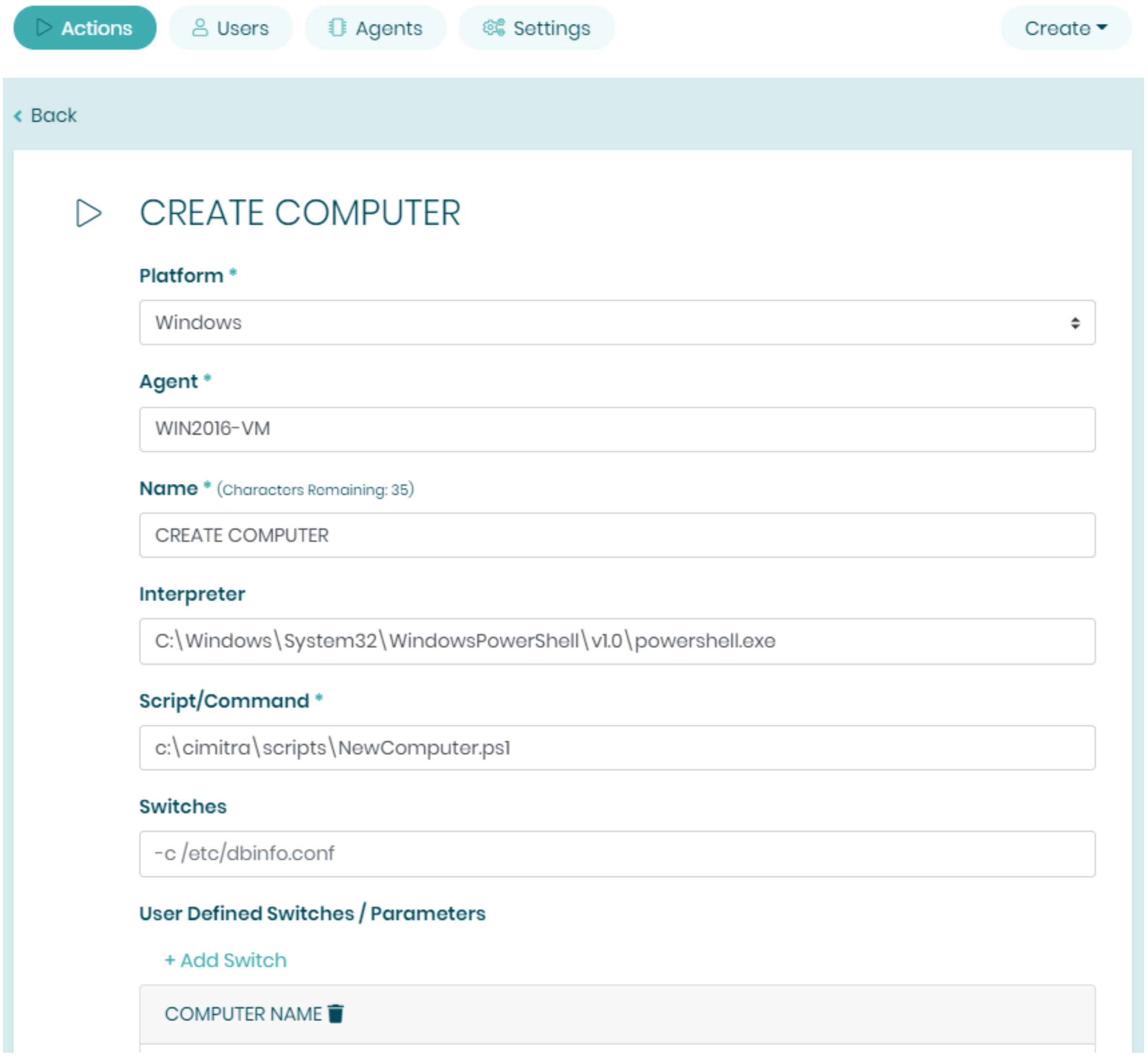
Name	CREATE COMPUTER
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	<Path to the Cimitra Script> Example: **c:\cimitra\scripts\NewComputer.ps1**
User Defined Switches/Parameters	Click the "**+Add Switch**" option twice to create two switches as shown below. These switches correlate with the two command-line parameters we programmed into the script.

COMPUTER NAME SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**COMPUTER NAME**
Validating Regex: Allow: Letters, Numbers, and Underscores	**/^[A-Za-z0-9_]+$/**

COMPUTER TYPE SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**COMPUTER TYPE**
Validating Regex: Allow: Number 1-5 and only one digit	**/[1-5]/gi**
Example:	**1 = Mac, 2 = Windows, 3 = Chromebook, 4 = Linux, 5 = Other**



CREATE COMPUTER

Platform *
Windows

Agent *
WIN2016-VM

Name * (Characters Remaining: 35)
CREATE COMPUTER

Interpreter
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

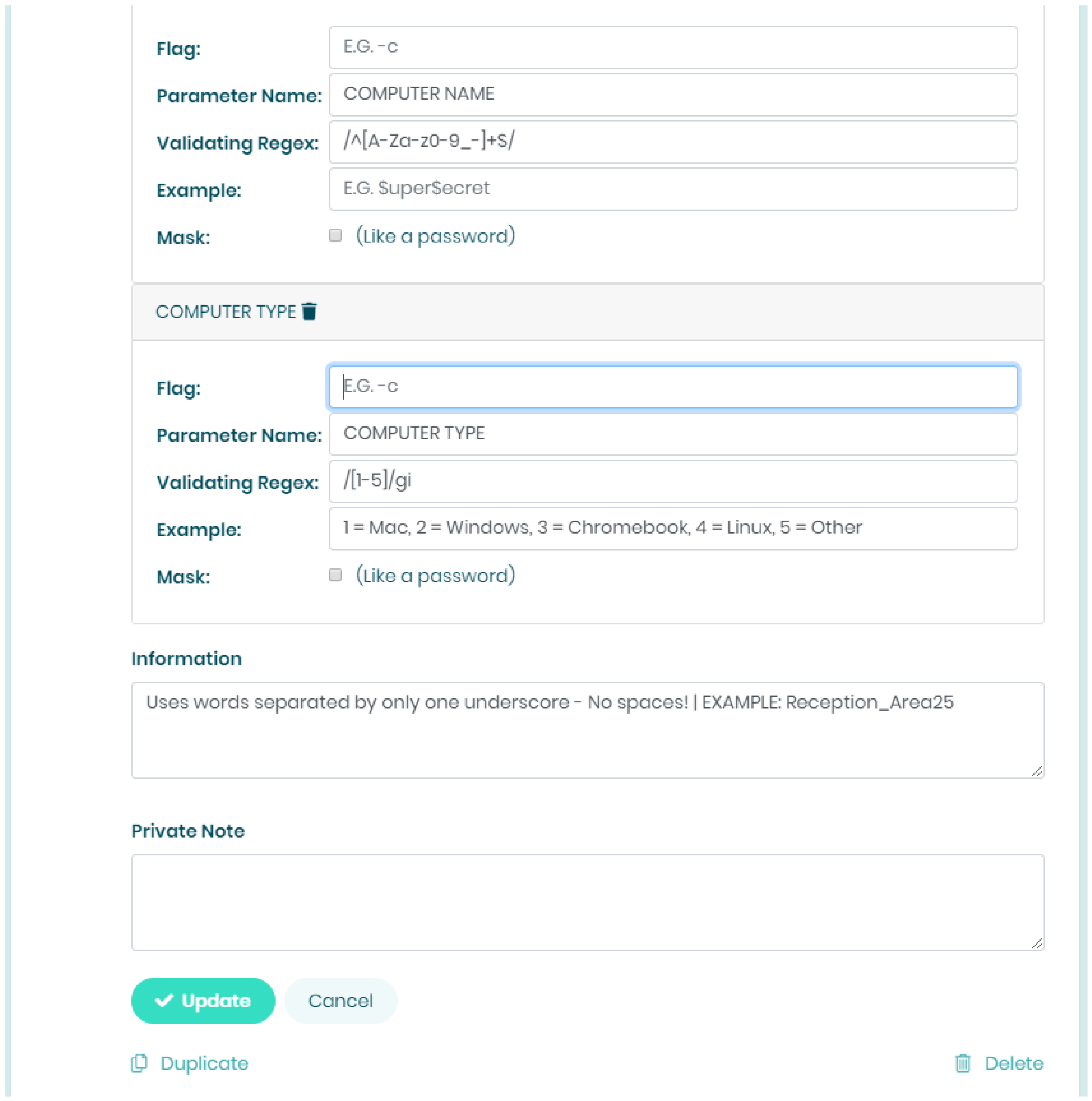
Script/Command *
c:\cimitra\scripts\NewComputer.ps1

Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters

+ Add Switch

COMPUTER NAME



Flag:	E.G. -c
Parameter Name:	COMPUTER NAME
Validating Regex:	/^[A-Za-z0-9_-]+$/
Example:	E.G. SuperSecret
Mask:	☐ (Like a password)

COMPUTER TYPE

Flag:	E.G. -c
Parameter Name:	COMPUTER TYPE
Validating Regex:	/[1-5]/gi
Example:	1 = Mac, 2 = Windows, 3 = Chromebook, 4 = Linux, 5 = Other
Mask:	☐ (Like a password)

Information

Uses words separated by only one underscore - No spaces! | EXAMPLE: Reception_Area25

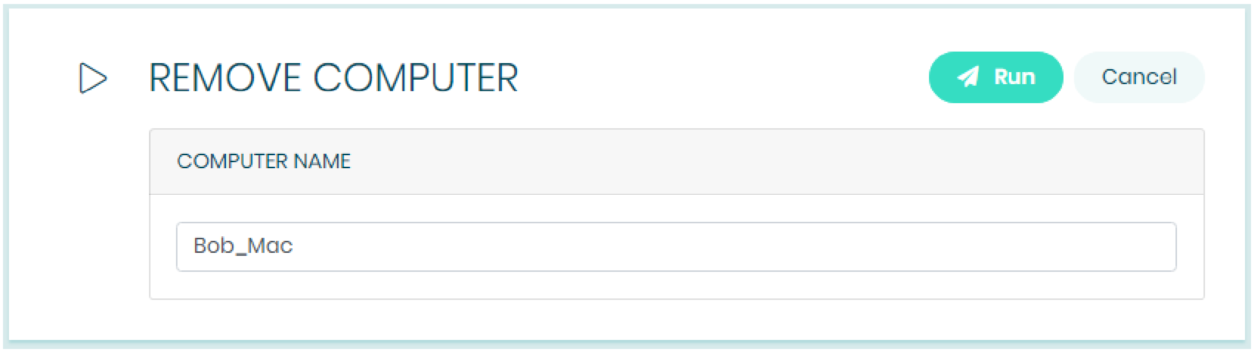
Private Note

Assuming the Cimitra Agent and the PowerShell script are all in place, the Cimitra App should now be usable and shareable with others. A Cimitra App needs to be in a Cimitra Folder in order to share the Cimitra App.

Removing a Computer in Active Directory

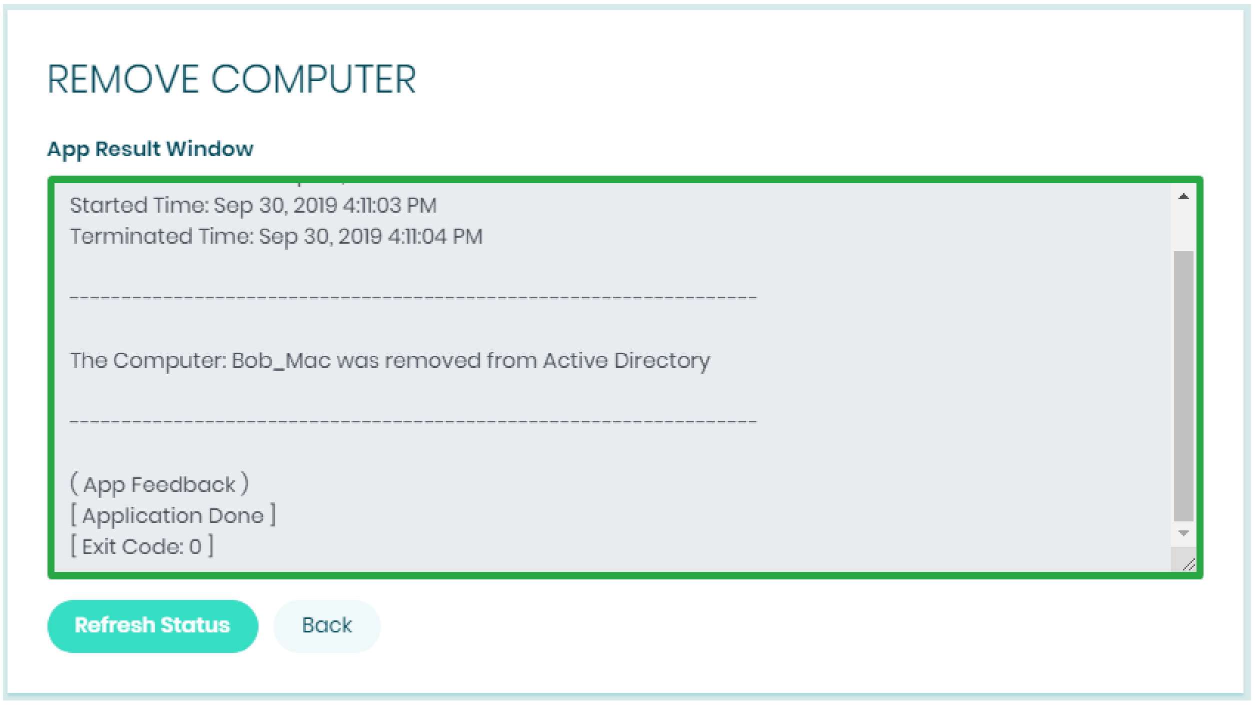
Use Case

Computers come and go in an organization very often. Being able to keep Active Directory clean of the clutter, it can be very helpful to allow other people on the Help Desk for example; to be able to remove computer objects.



COMPUTER NAME

Bob_Mac



```
Started Time: Sep 30, 2019 4:11:03 PM
Terminated Time: Sep 30, 2019 4:11:04 PM

------------------------------------------------------------------

The Computer: Bob_Mac was removed from Active Directory

------------------------------------------------------------------

( App Feedback )
[ Application Done ]
[ Exit Code: 0 ]
```

Technical Overview

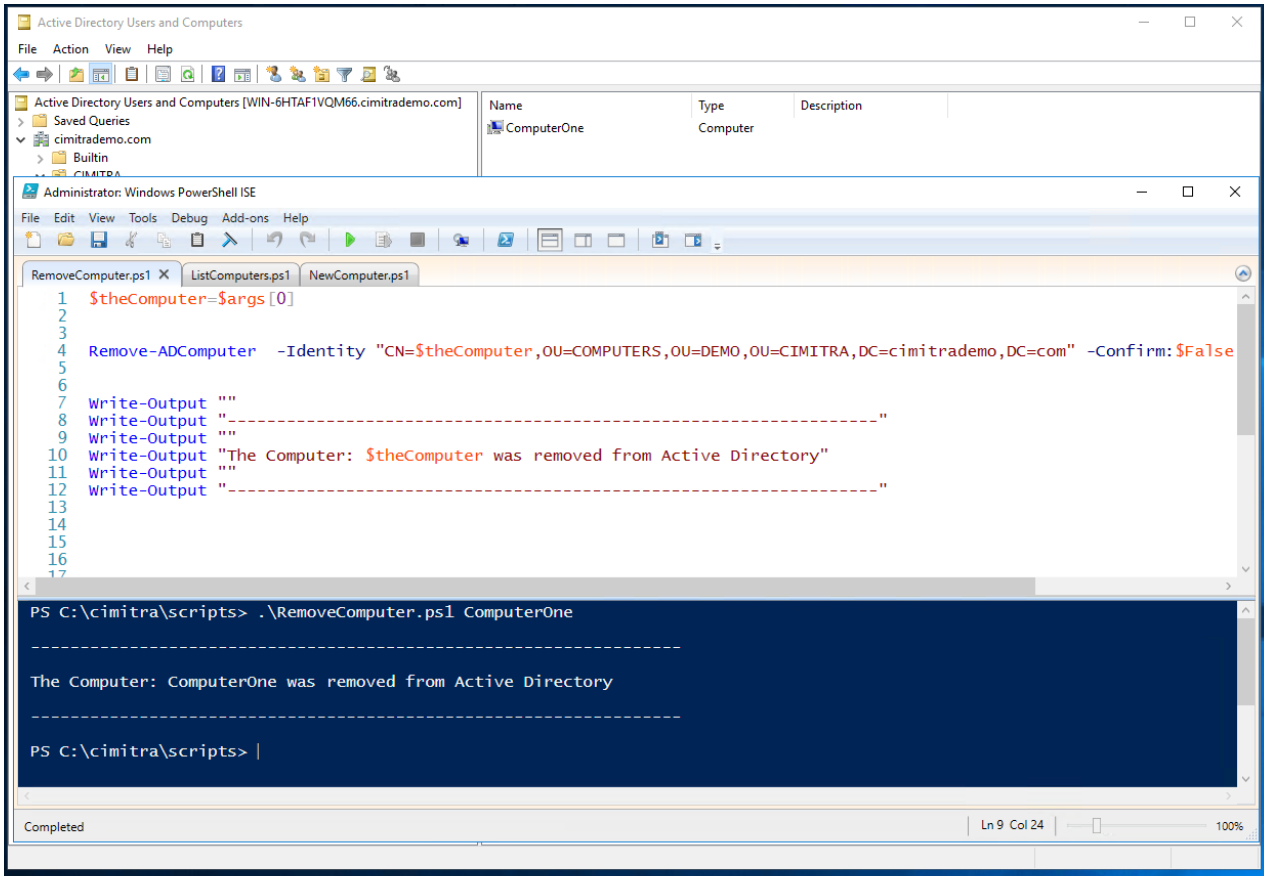
This script is rather straightforward. There are basically three directives we want to specify to Active Directory when specifying a delete. 1. Where the object is. 2. What the name of the object is. 3. Don't ask for confirmation (Cimitra doesn't allow for that).

Command	**Parameters**
Remove-ADComputer	<COMPUTER NAME>
	-Identity <LDAP type path location to create the computer object>
Remove-ADComputer	**`-Identity "CN=$theComputer,OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com" -Confirm:$False`**
ENTIRE COMMAND	`Remove-ADComputer  -Identity "CN=$theComputer,OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com" -Confirm:$False`

Script Contents for Copy/Paste

```
# Assign first parameter to script to: $theComputer
$theComputer=$args[0]

# Use Remove-ADComputer to remove the computer
Remove-ADComputer  -Identity
"CN=$theComputer,OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=ci
mitrademo,DC=com" -Confirm:$False

Write-Output ""
Write-Output "-------------------------------------------------------"
Write-Output ""
```

```
Write-Output "The Computer: $theComputer was removed from Active
Directory"
Write-Output ""
Write-Output "-----------------------------------------------------------"
```


Active Directory Users and Computers
File Action View Help
Active Directory Users and Computers [WIN-6HTAF1VQM66.cimitrademo.com]
Saved Queries
cimitrademo.com
Builtin
Name Type Description
ComputerOne Computer
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
RemoveComputer.ps1 ListComputers.ps1 NewComputer.ps1
Completed
Ln 9 Col 24
100%

```
$theComputer=$args[0]


Remove-ADComputer  -Identity "CN=$theComputer,OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com" -Confirm:$False


Write-Output ""
Write-Output "-------------------------------------------------------------------"
Write-Output ""
Write-Output "The Computer: $theComputer was removed from Active Directory"
Write-Output ""
Write-Output "-------------------------------------------------------------------"
```

```
PS C:\cimitra\scripts> .\RemoveComputer.ps1 ComputerOne

-------------------------------------------------------------------

The Computer: ComputerOne was removed from Active Directory

-------------------------------------------------------------------

PS C:\cimitra\scripts>
```

Script Integration Into Cimitra

This assumes that you have already created a Cimitra Server deployed a Cimitra Agent etc. to a Windows Server where the **RemoveComputer.ps1** script exists.

Create a new **Cimitra App** object and fill in the following properties as follows:

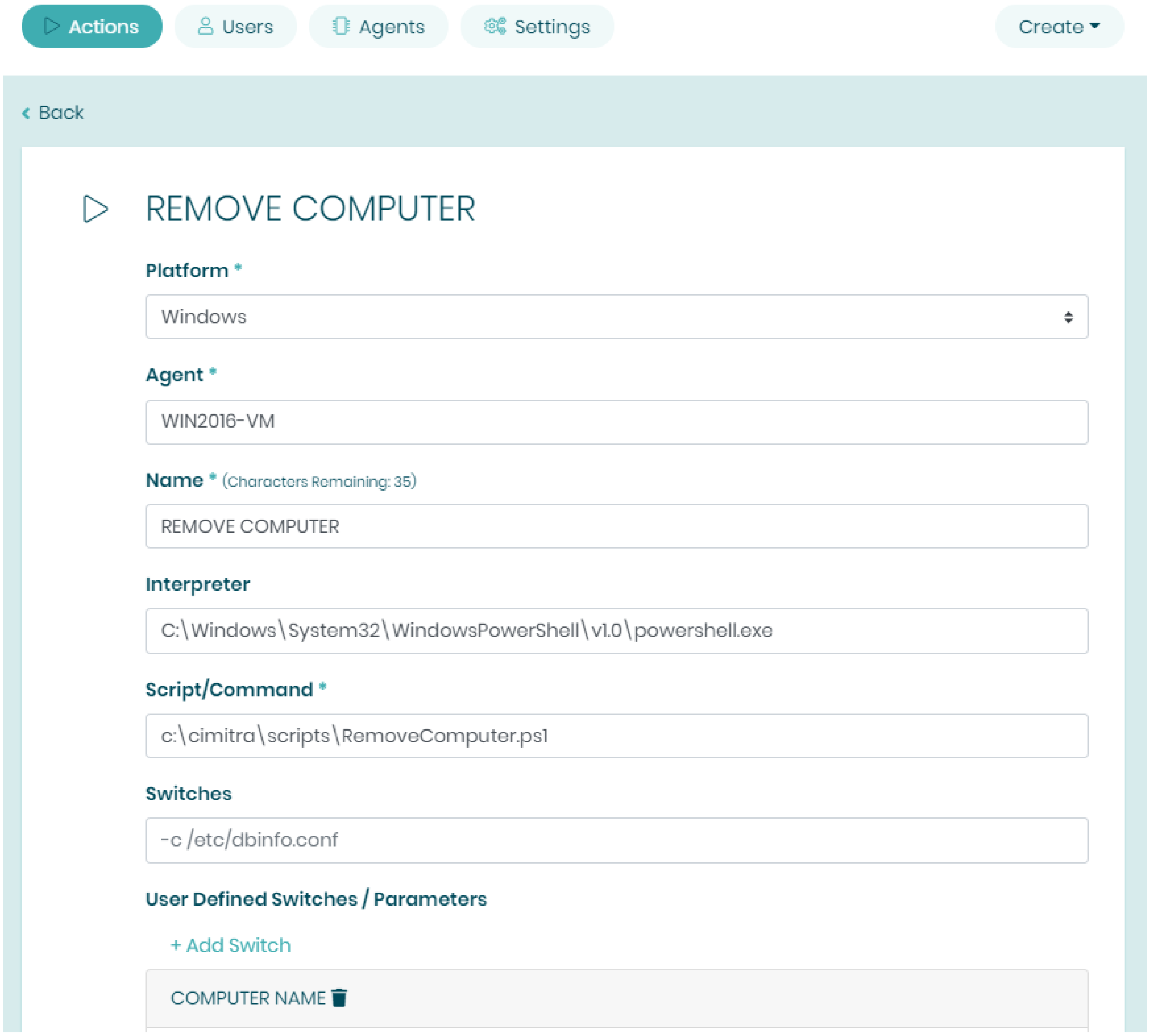
CIMITRA APP PROPERTIES

Property	Value

Platform	**Windows**
Agent	<The Cimitra Agent deployed to the Windows Server>
Name	**REMOVE COMPUTER**
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	<Path to the Cimitra Script> Example: **c:\cimitra\scripts\RemoveComputer.ps1**
User Defined Switches/Parameters	Click the "**+Add Switch**" option for one switch as shown below. This switch correlates with the command line parameter for the computer's name that we programmed into the script.

COMPUTER NAME SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**COMPUTER NAME**
Validating Regex: Allow: Letters, Numbers, and Underscores	**/^[A-Za-z0-9_]+$/**

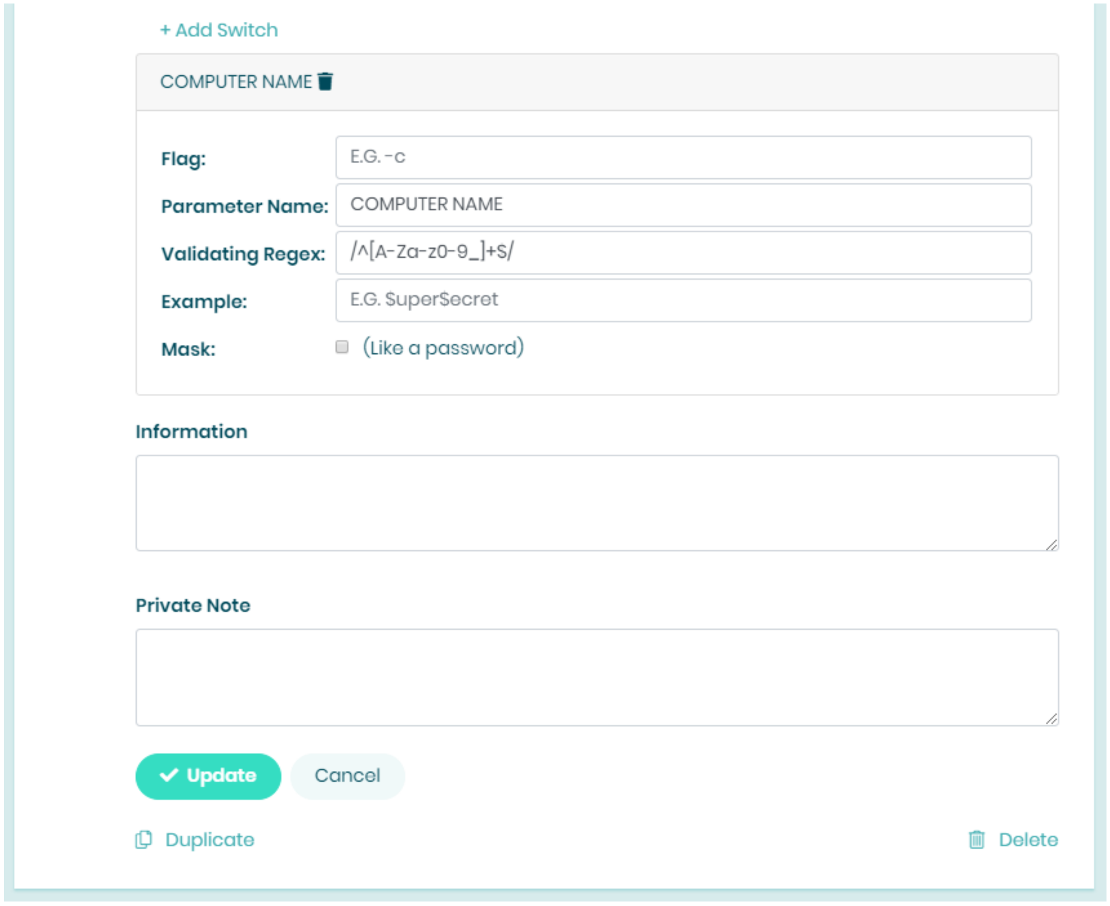


REMOVE COMPUTER

Platform *	Windows
Agent *	WIN2016-VM
Name * (Characters Remaining: 35)	REMOVE COMPUTER
Interpreter	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Script/Command *	c:\cimitra\scripts\RemoveComputer.ps1
Switches	-c /etc/dbinfo.conf

User Defined Switches / Parameters

COMPUTER NAME



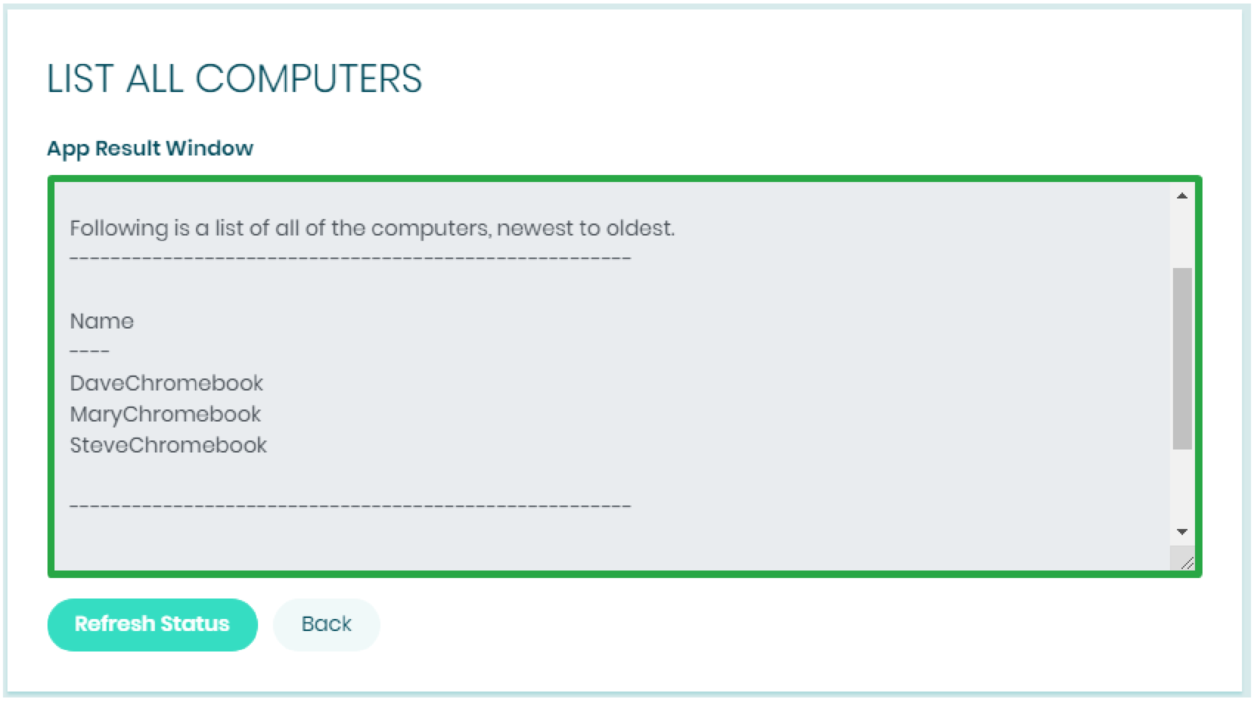
Flag:	E.G. -c
Parameter Name:	COMPUTER NAME
Validating Regex:	/^[A-Za-z0-9_]+$/
Example:	E.G. SuperSecret
Mask:	(Like a password)

Assuming the Cimitra Agent and the PowerShell script are all in place, the Cimitra App should now be usable and shareable with others. A Cimitra App needs to be in a Cimitra Folder in order to share the Cimitra App.

Listing Computers in Active Directory

Use Case

Computers come and go in an organization very often. Being able to keep Active Directory clean of the clutter, it can be very helpful to allow other people on the Help Desk for example; to be able to remove computer objects. But first, it might be helpful to see the list of computers in an Active Directory context.


LIST ALL COMPUTERS

App Result Window

```
Following is a list of all of the computers, newest to oldest.
-----------------------------------------------------

Name
----
DaveChromebook
MaryChromebook
SteveChromebook

-----------------------------------------------------
```

Technical Overview

This script takes no inputs. The script reads out all of the Computer objects in a certain context and puts them into an array, which is a computer version of a list. Then the array is reversed to show the newest computers first, and then the array is displayed. I found this gem on the Internet, I take no credit :)

Script Contents - Copy/Paste

```
Write-Output ""
Write-Output "Following is a list of all of the computers, newest to oldest."
Write-Output "-----------------------------------------------------"
# Create a function called “reverse”
function reverse
{
# Get a list of all computers, and assign it to $listOfComputers
```

```
 $listOfComputers = @(Get-ADComputer -Filter * -SearchBase
"OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"
| Select Name)

 [array]::reverse($listOfComputers)
 $listOfComputers
}
# Call the reverse function
reverse
Write-Output ""
Write-Output "--------------------------------------------------------"
```

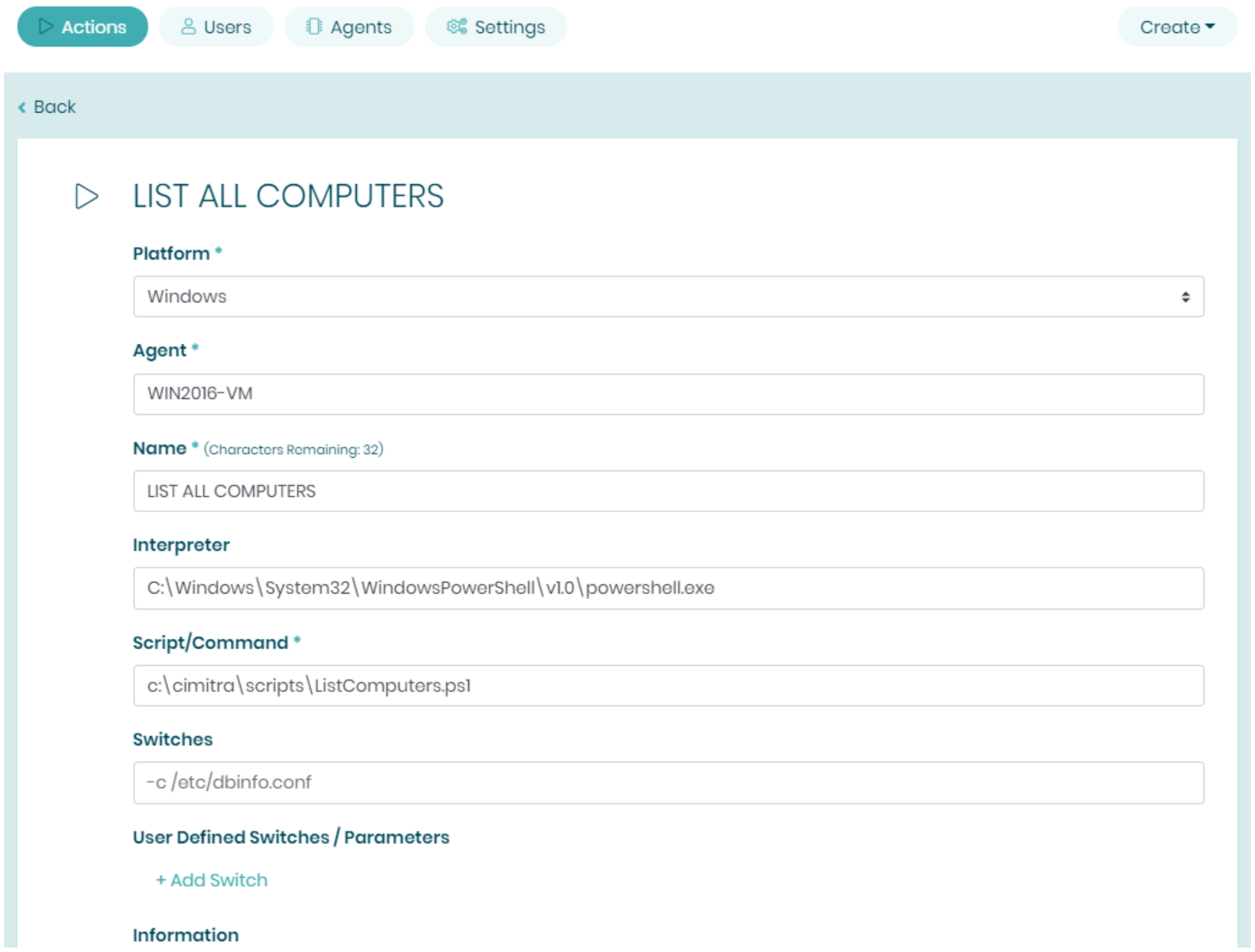
Script Integration Into Cimitra

This assumes that you have already created a Cimitra Server deployed a Cimitra Agent etc. to a Windows Server where the **ListComputers.ps1** script exists.

Create a new **Cimitra App** object and fill in the following properties as follows:

CIMITRA APP PROPERTIES

Property	Value
Platform	**Windows**
Agent	<The Cimitra Agent deployed to the Windows Server>
Name	**LIST COMPUTERS**
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	<Path to the Cimitra Script> Example: **c:\cimitra\scripts\ListComputers.ps1**



LIST ALL COMPUTERS

Platform *
Windows

Agent *
WIN2016-VM

Name * (Characters Remaining: 32)
LIST ALL COMPUTERS

Interpreter
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Script/Command *
c:\cimitra\scripts\ListComputers.ps1

Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters
+ Add Switch

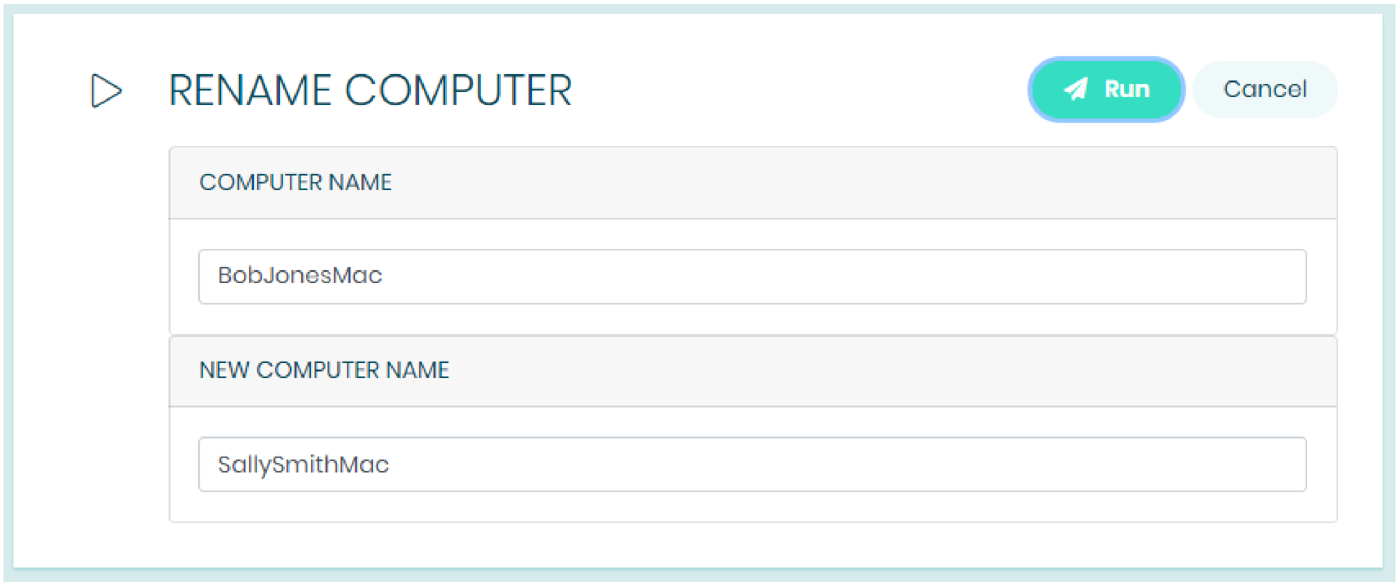
Information

Renaming a Computer in Active Directory

Use Case

Computers in an organization are given a name similar to this: BobJonesMac. When a computer is reassigned to a new user, the computer name should be renamed to reflect the new owner.

Renaming computers to reflect the correct user is very important for inventory purposes, however this task wasn't being done regularly since the task wasn't causing anyone to be locked out of anything. Now the IT hardware folks have been given the ability to rename the Computer object in Active Directory without having any rights in Active Directory.

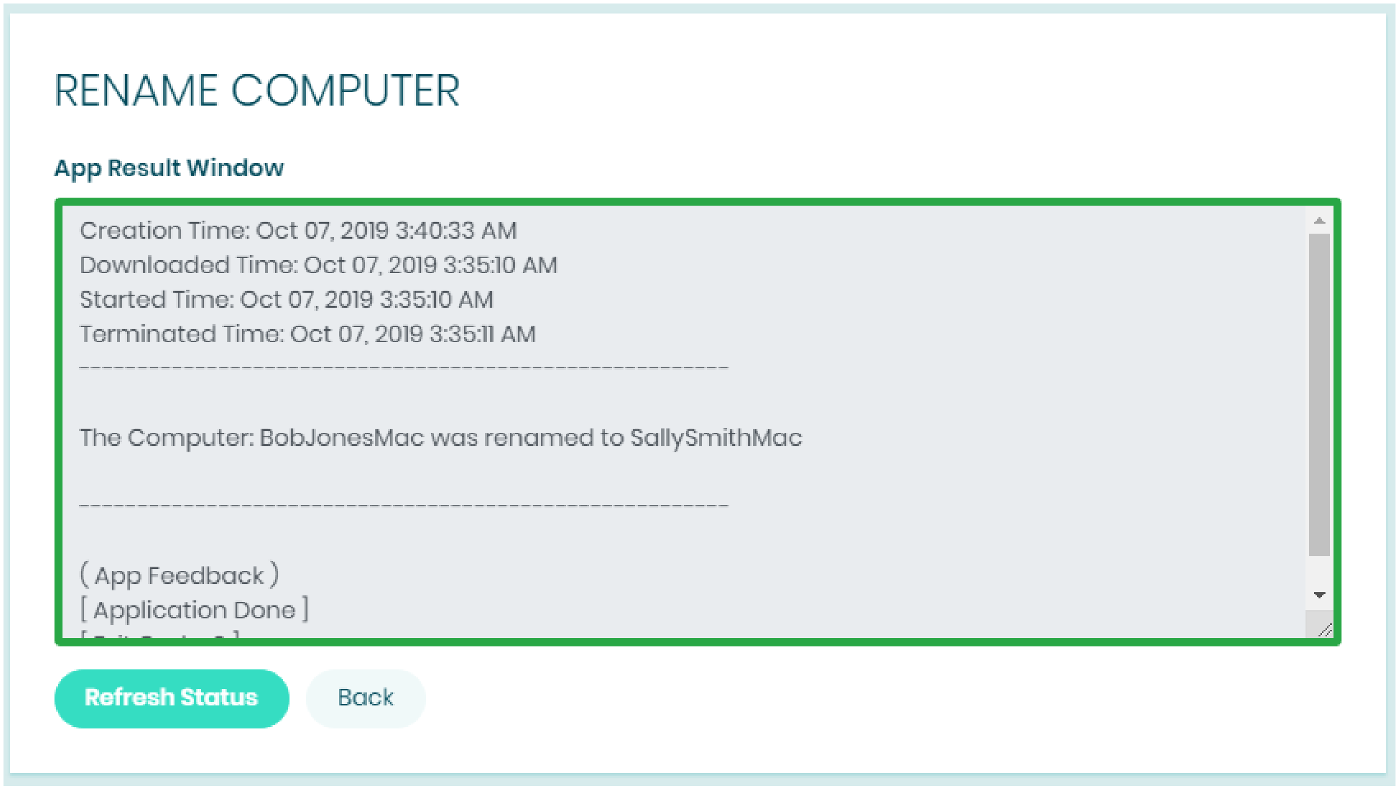


COMPUTER NAME

BobJonesMac

NEW COMPUTER NAME

SallySmithMac



```
Creation Time: Oct 07, 2019 3:40:33 AM
Downloaded Time: Oct 07, 2019 3:35:10 AM
Started Time: Oct 07, 2019 3:35:10 AM
Terminated Time: Oct 07, 2019 3:35:11 AM
------------------------------------------------------------


The Computer: BobJonesMac was renamed to SallySmithMac


------------------------------------------------------------


( App Feedback )
[ Application Done ]
```

Technical Overview

This script takes in two inputs. The current computer name and the new computer name. The script the script renames the computer.

Script Contents - Copy/Paste

```
# Change the context variable to match your system
# -------------------------------------------------

$context = "OU=COMPUTERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"

# -------------------------------------------------
$theComputerOldNameIn = $args[0]
$theComputerNewNameIn = $args[1]

Rename-ADObject -Identity "CN=$theComputerOldNameIn,$context"
-NewName "$theComputerNewNameIn"
$theResult = $?

if ($theResult)
{
Write-Output
"----------------------------------------------------------"
Write-Output ""
Write-Output "The Computer: $theComputerOldNameIn was renamed to
$theComputerNewNameIn"
Write-Output ""
Write-Output
"----------------------------------------------------------"
}
```

Script Integration Into Cimitra

This assumes that you have already created a Cimitra Server deployed a Cimitra Agent etc. to a Windows Server where the **ListComputers.ps1** script exists.

Create a new **Cimitra App** object and fill in the following properties as follows:

CIMITRA APP PROPERTIES

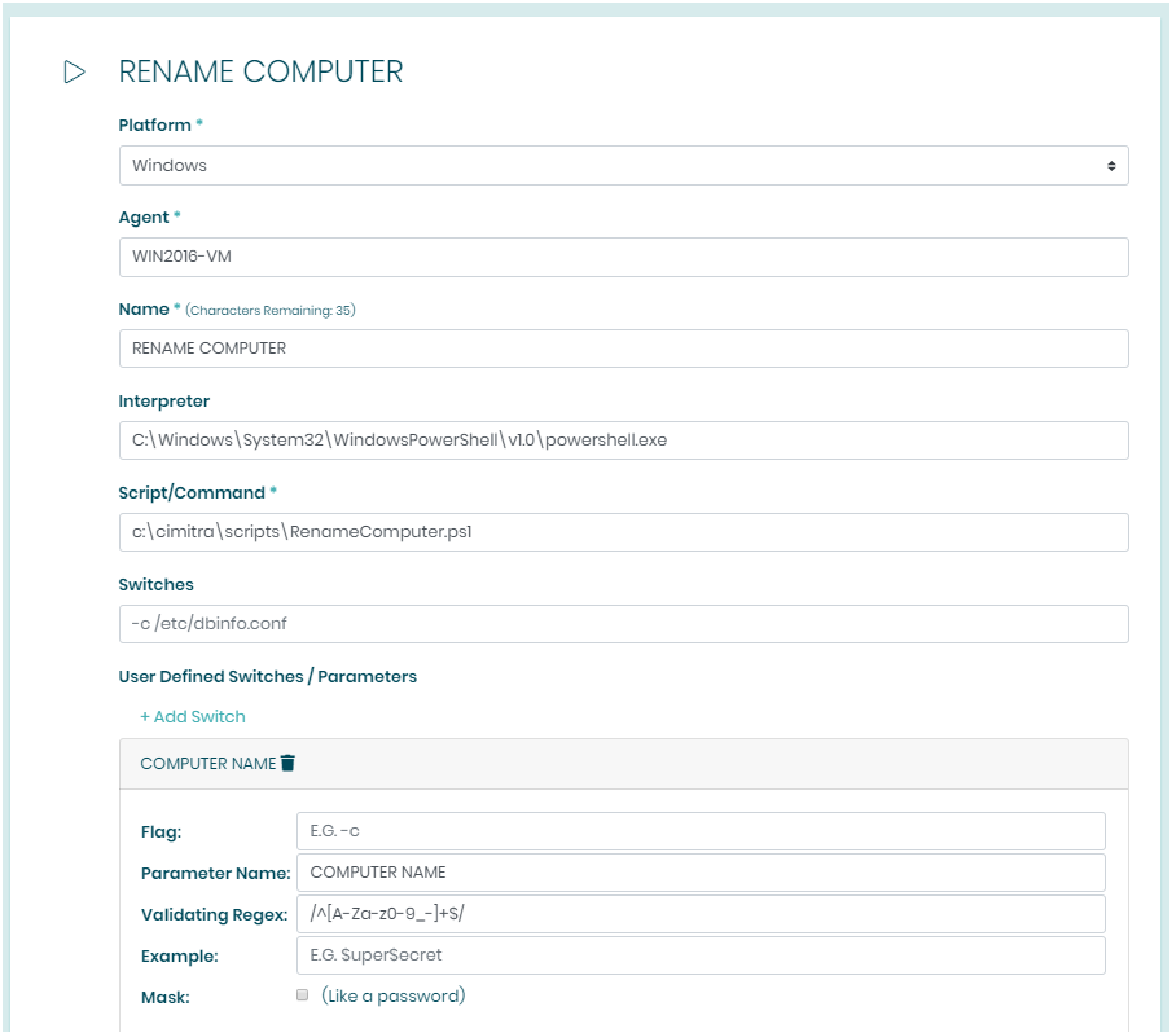
Property	Value
Platform	**Windows**
Agent	<The Cimitra Agent deployed to the Windows Server>
Name	**REMOVE COMPUTER**
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	<Path to the Cimitra Script> Example: **c:\cimitra\scripts\RemoveComputer.ps1**
User Defined Switches/Parameters	Click the "**+Add Switch**" twice for the two switches as shown below.

COMPUTER NAME SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**COMPUTER NAME**
Validating Regex: Allow: Letters, Numbers, and Underscores	**/^[A-Za-z0-9_-]+$/**

NEW COMPUTER NAME SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**NEW COMPUTER NAME**
Validating Regex: Allow: Letters, Numbers, and Underscores	**/^[A-Za-z0-9_-]+$/**



RENAME COMPUTER

Platform *
Windows

Agent *
WIN2016-VM

Name * (Characters Remaining: 35)
RENAME COMPUTER

Interpreter
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Script/Command *
c:\cimitra\scripts\RenameComputer.ps1

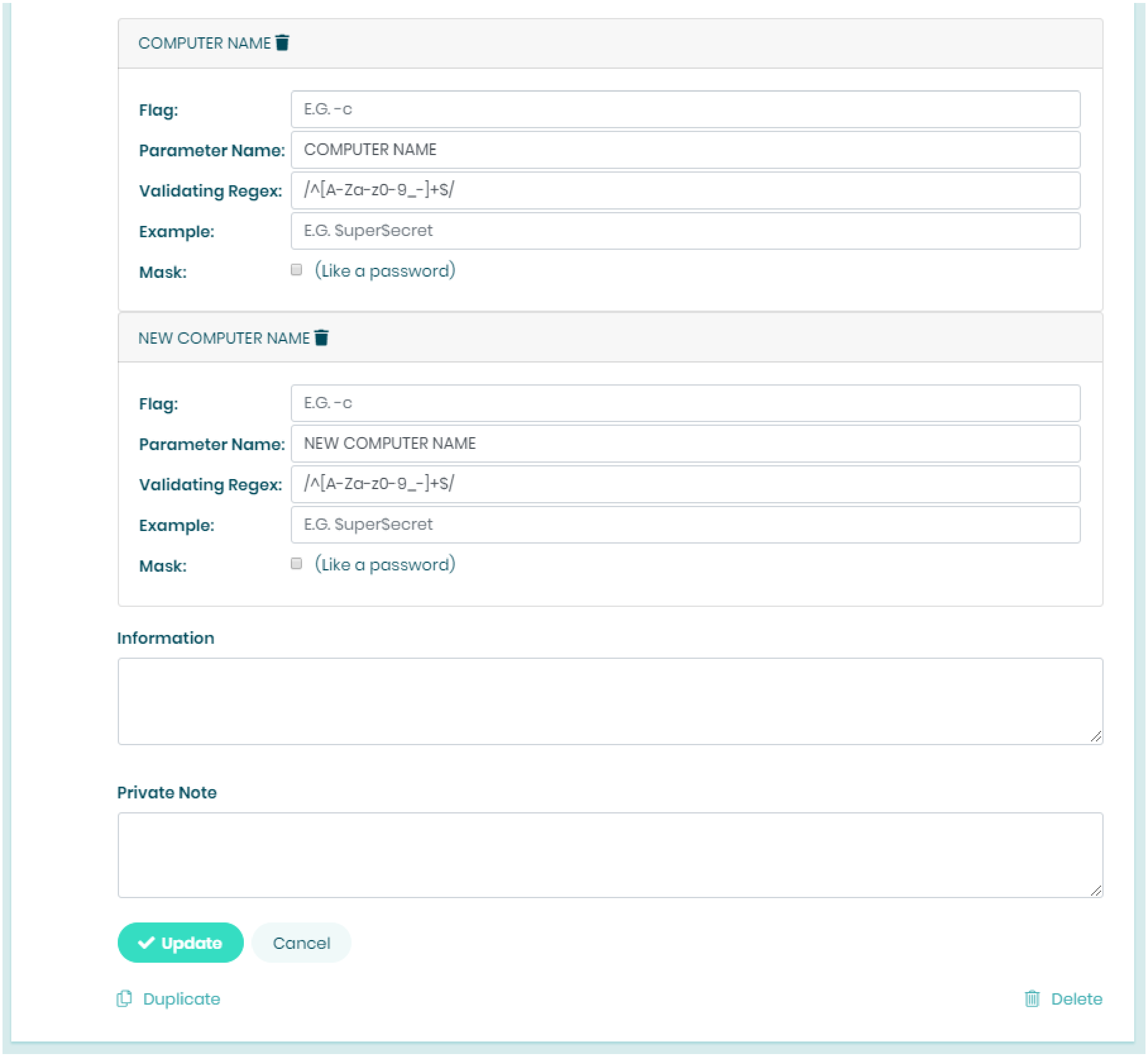
Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters

+ Add Switch

COMPUTER NAME

Flag:	E.G. -c
Parameter Name:	COMPUTER NAME
Validating Regex:	/^[A-Za-z0-9_-]+$/
Example:	E.G. SuperSecret
Mask:	☐ (Like a password)



COMPUTER NAME	
Flag:	E.G. -c
Parameter Name:	COMPUTER NAME
Validating Regex:	/^[A-Za-z0-9_-]+$/
Example:	E.G. SuperSecret
Mask:	(Like a password)
NEW COMPUTER NAME	
Flag:	E.G. -c
Parameter Name:	NEW COMPUTER NAME
Validating Regex:	/^[A-Za-z0-9_-]+$/
Example:	E.G. SuperSecret
Mask:	(Like a password)

Information

Private Note

Calling External PowerShell Scripts

You can use PowerShell syntax to call another PowerShell script. The ListComputers.ps1 script might be a handy script to call from the RemoteComputer.ps1 or the AddComputer.ps1 scripts at the end of these respective scripts. The way you do that is by adding this command to the end of the script in this manner:

```
.\ListComputers.ps1
```

This syntax assumes that the ListComputers.ps1 file is in the same directory as the calling script.