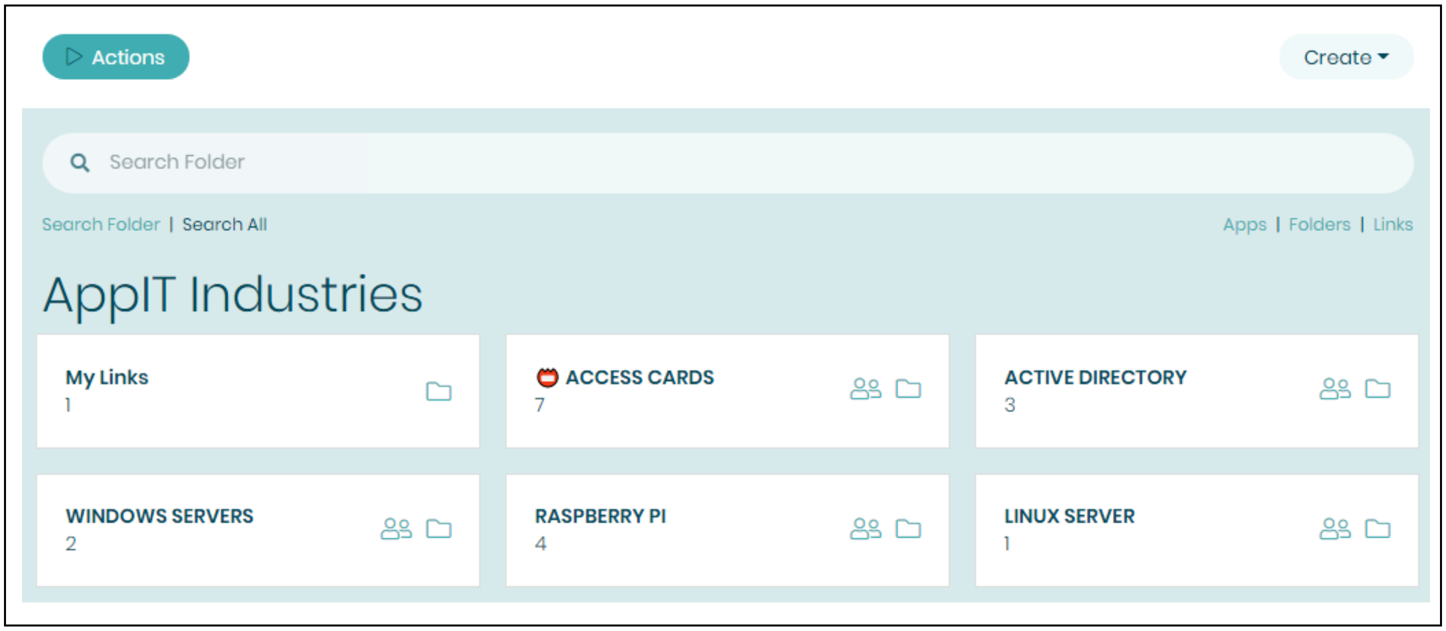
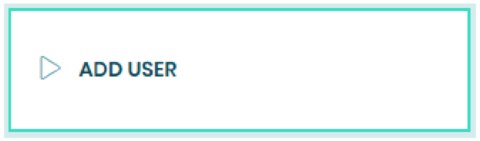
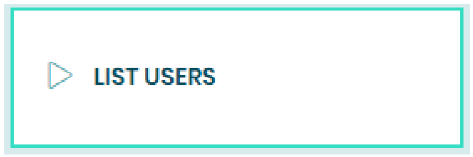
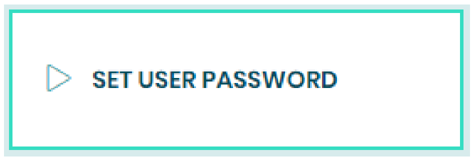
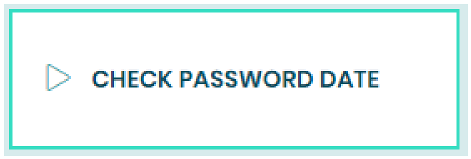
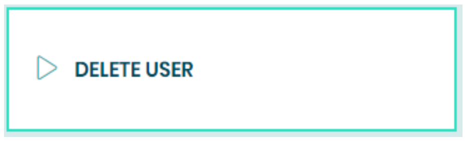
cimitra

Active Directory User Scripts



Topic	**Documentation**

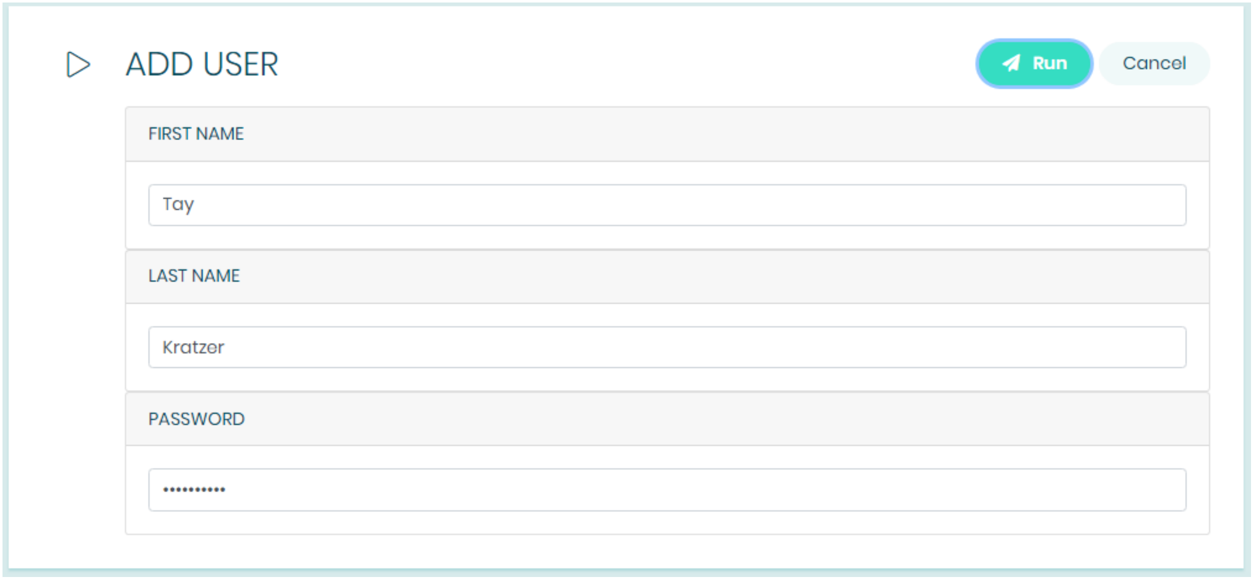
All scripts can be downloaded using one of two methods	**[Download Scripts Here - Zip file](#) (HTTP)**
	
	Or:
	
	`git clone https://github.com/cimitrasoftware/powershell_scripts.git`
	
	NOTE: Many of these scripts were updated in July of 2020. The newer versions of these scripts is located at:
	
	`https://github.com/cimitrasoftware/new-powershell-scripts`

 ADD USER	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
 LIST USERS	Use Case Command Documentation Command Integration Into Cimitra PowerShell Command - Copy/Paste
 SET USER PASSWORD	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
 CHECK PASSWORD DATE	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
 DELETE USER With an **Access Code**	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste
NEW USER	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste

Adding a User in Active Directory

Use Case

New users need basic access to the network. The HR department and Help Desk have been tasked with creating users so that they can immediately get access to some network workstations.



FIRST NAME

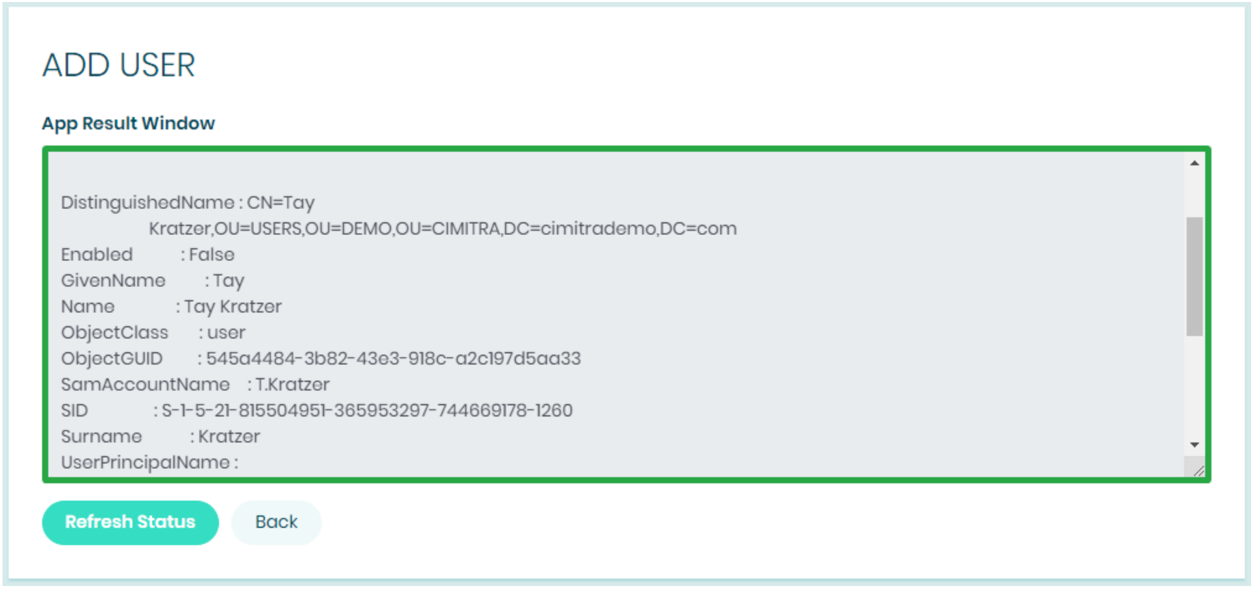
Tay

LAST NAME

Kratzer

PASSWORD

••••••••••



```
DistinguishedName : CN=Tay
            Kratzer,OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com
Enabled        : False
GivenName        : Tay
Name           : Tay Kratzer
ObjectClass      : user
ObjectGUID       : 545a4484-3b82-43e3-918c-a2c197d5aa33
SamAccountName   : T.Kratzer
SID            : S-1-5-21-815504951-365953297-744669178-1260
Surname          : Kratzer
UserPrincipalName :
```

Technical Overview

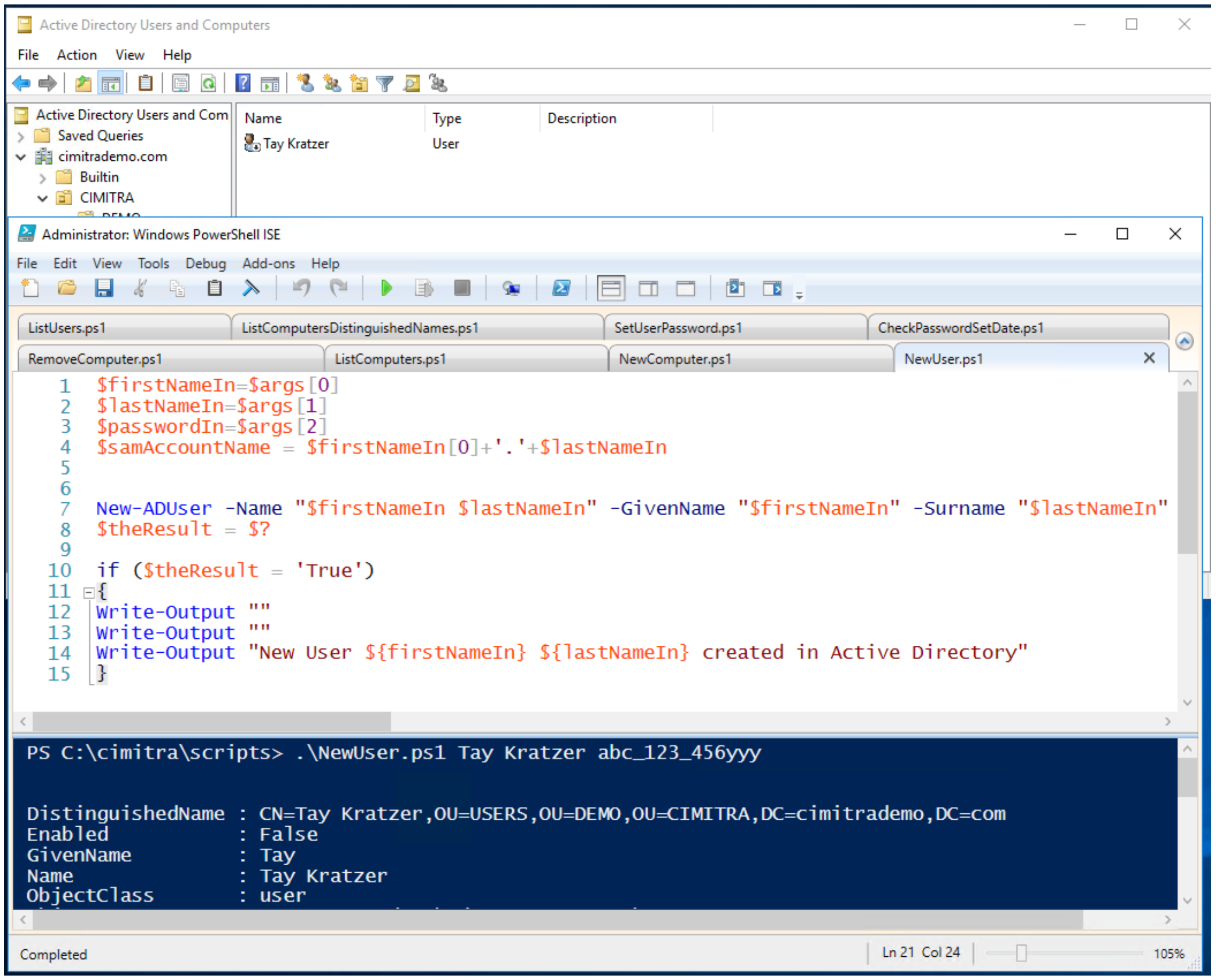
This PowerShell script will create a user object in Active Directory at a specific location in Active Directory as specified by the `-path` switch. The script takes in three command-line parameters that are converted by the script into variables within the script. These variables are then passed to Active Directory using the `New-AdUser` command along with supporting variables and required commands.

Script Contents - Copy Paste | **`NewUser.ps1`**

```
# Read in parameters and assign them to variables
$firstNameIn=$args[0]
$lastNameIn=$args[1]
$passwordIn=$args[2]
$samAccountName = $firstNameIn[0]+'.'+$lastNameIn

# Create the new user
New-ADUser -Name "$firstNameIn $lastNameIn" -GivenName
"$firstNameIn" -Surname "$lastNameIn" -SamAccountName
"$samAccountName" -AccountPassword (ConvertTo-SecureString
"$passwordIn" -AsPlainText -force) -passThru -path
"OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"

# Catch the exit code from running the command
$theResult = $?

if ($theResult = 'True')
{
Write-Output ""
Write-Output ""
Write-Output "New User ${firstNameIn} ${lastNameIn} created
in Active Directory"
}

# Enable the account
Enable-ADAccount -Identity "CN=$firstNameIn
$lastNameIn,OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=co
m" -Confirm:$False

# Force an immediate password reset
```

```
Set-ADUser -Identity  "CN=$firstNameIn
$lastNameIn,OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=co
m" -ChangePasswordAtLogon $true
```

`NewUser.ps1` PowerShell Script in PowerShell ISE



```
$firstNameIn=$args[0]
$lastNameIn=$args[1]
$passwordIn=$args[2]
$samAccountName = $firstNameIn[0]+'.'+$lastNameIn


New-ADUser -Name "$firstNameIn $lastNameIn" -GivenName "$firstNameIn" -Surname "$lastNameIn"
$theResult = $?

if ($theResult = 'True')
{
Write-Output ""
Write-Output ""
Write-Output "New User ${firstNameIn} ${lastNameIn} created in Active Directory"
}
```

```
PS C:\cimitra\scripts> .\NewUser.ps1 Tay Kratzer abc_123_456yyy


DistinguishedName : CN=Tay Kratzer,OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com
Enabled           : False
GivenName         : Tay
Name              : Tay Kratzer
ObjectClass       : user
```

Script Integration into Cimitra

Script Integration Into Cimitra

This assumes that you have already created a Cimitra server and deployed a Cimitra Agent etc. to a Windows Server where the **`NewUser.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	**ADD USER**
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\NewUser.ps1**
User Defined Switches/Parameters	Click the "**+Add Switch**" option three times for switches as shown below. These switches correlate with the three command-line parameters we programmed into the script.

FIRST NAME SWITCH

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

LAST NAME SWITCH

Flag:	<LEAVE THIS FIELD BLANK>

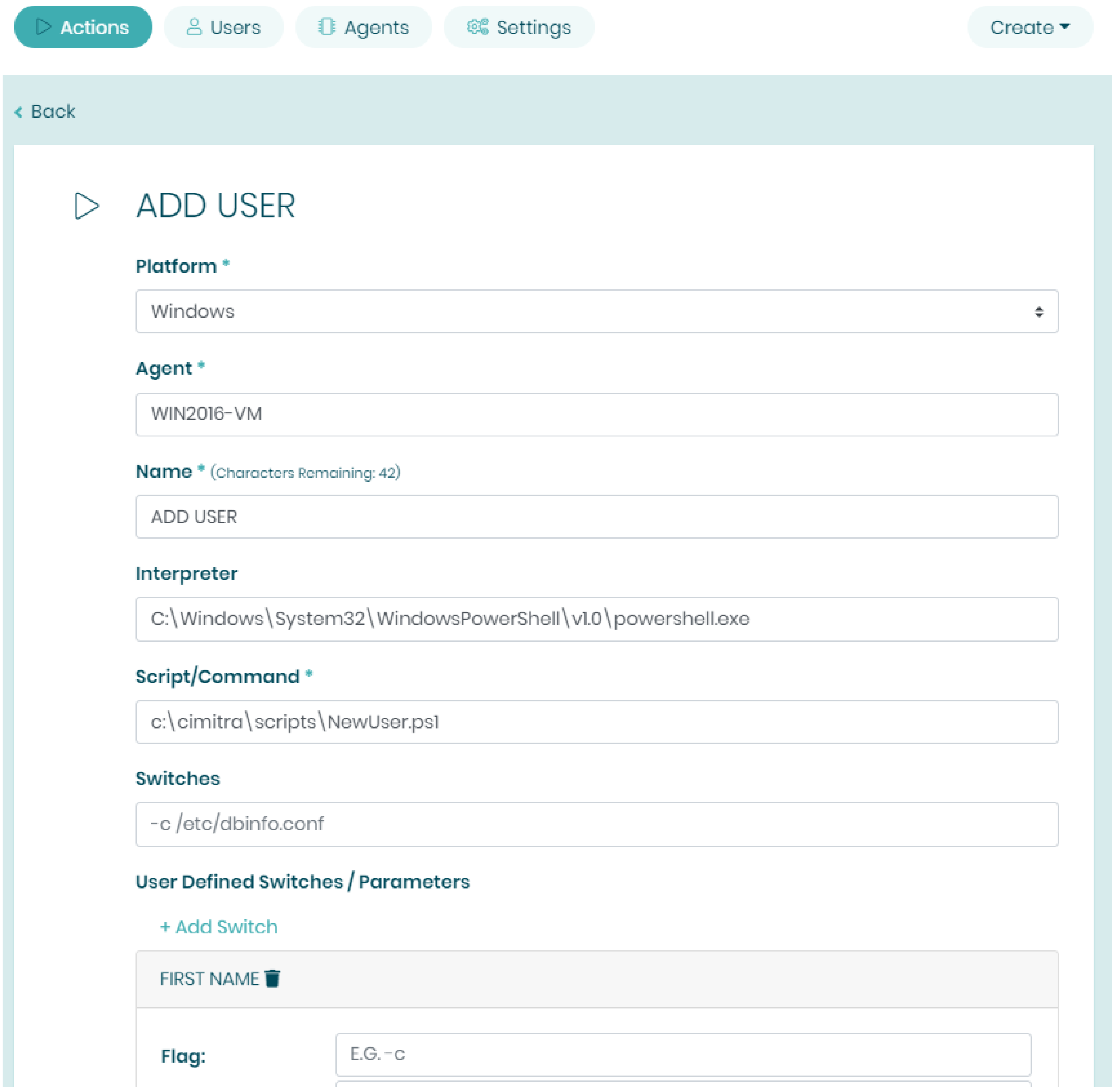
Parameter Name:	**LAST NAME**
Validating Regex: Allow: Letters, and Underscores	**/^[A-Za-z]+$/**

PASSWORD SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	PASSWORD
Validating Regex: Allow: Letters, numbers, dashes, and Underscores	**/^[A-Za-z0-9_]+$/**
Example:	(Letters, Numbers, Dash "-", Underscores "_")
Mask:	**ENABLE THIS**

INFORMATION FIELD

NOTE: The password should be 8 characters long, and include a number and an underscore or dash and one uppercase letter.



ADD USER

Platform *

Windows

Agent *

WIN2016-VM

Name * (Characters Remaining: 42)

ADD USER

Interpreter

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

Script/Command *

c:\cimitra\scripts\NewUser.ps1

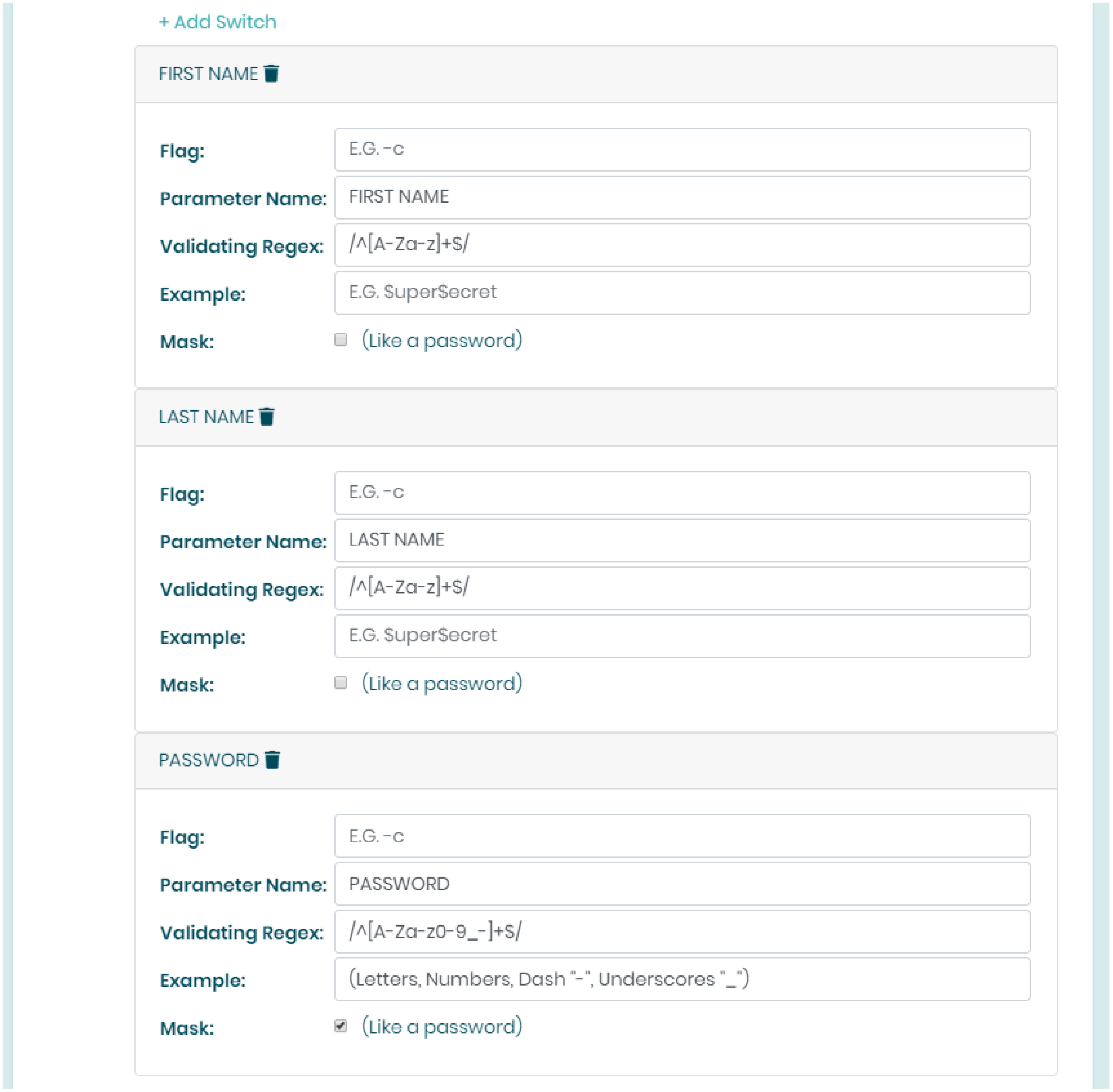
Switches

-c /etc/dbinfo.conf

User Defined Switches / Parameters

FIRST NAME

Flag: E.G. -c


+ Add Switch

FIRST NAME

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

LAST NAME

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

PASSWORD

Flag:	E.G. -c
Parameter Name:	PASSWORD
Validating Regex:	/^[A-Za-z0-9_-]+$/
Example:	(Letters, Numbers, Dash "-", Underscores "_")
Mask:	☑ (Like a password)

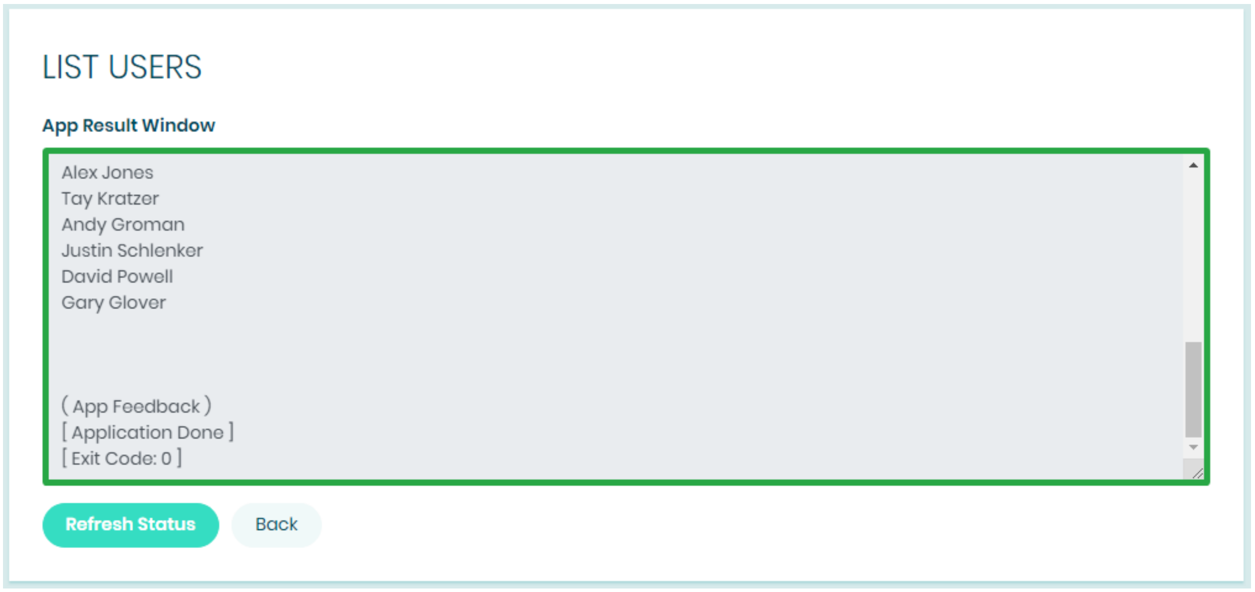
Assuming the Cimitra Agent and the PowerShell script are all in place, the Cimitra App should now be usable and shareable with others. A Cimitra App needs to be in a Cimitra Folder object in order to share the Cimitra App.

Listing Users in Active Directory

NOTE: Sometimes a simple PowerShell command to get some information out of PowerShell is all that you will need to perform. This section will show how to use a PowerShell command right inside a Cimitra App, without any need to establish a PowerShell script of any sort.

Use Case

The Help Desk and the HR department have been given rights to add users into Active Directory. Before they add a user, they want to make sure there are not any duplicate users. After they create the user, they may want to confirm that the user was created.



```
Alex Jones
Tay Kratzer
Andy Groman
Justin Schlenker
David Powell
Gary Glover



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

Technical Overview

The PowerShell Get-ADUser command is all that we will use..

PowerShell Command - Copy Paste

```
Get-ADUser "-Filter * -SearchBase
'OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com' | Select
Name"
```

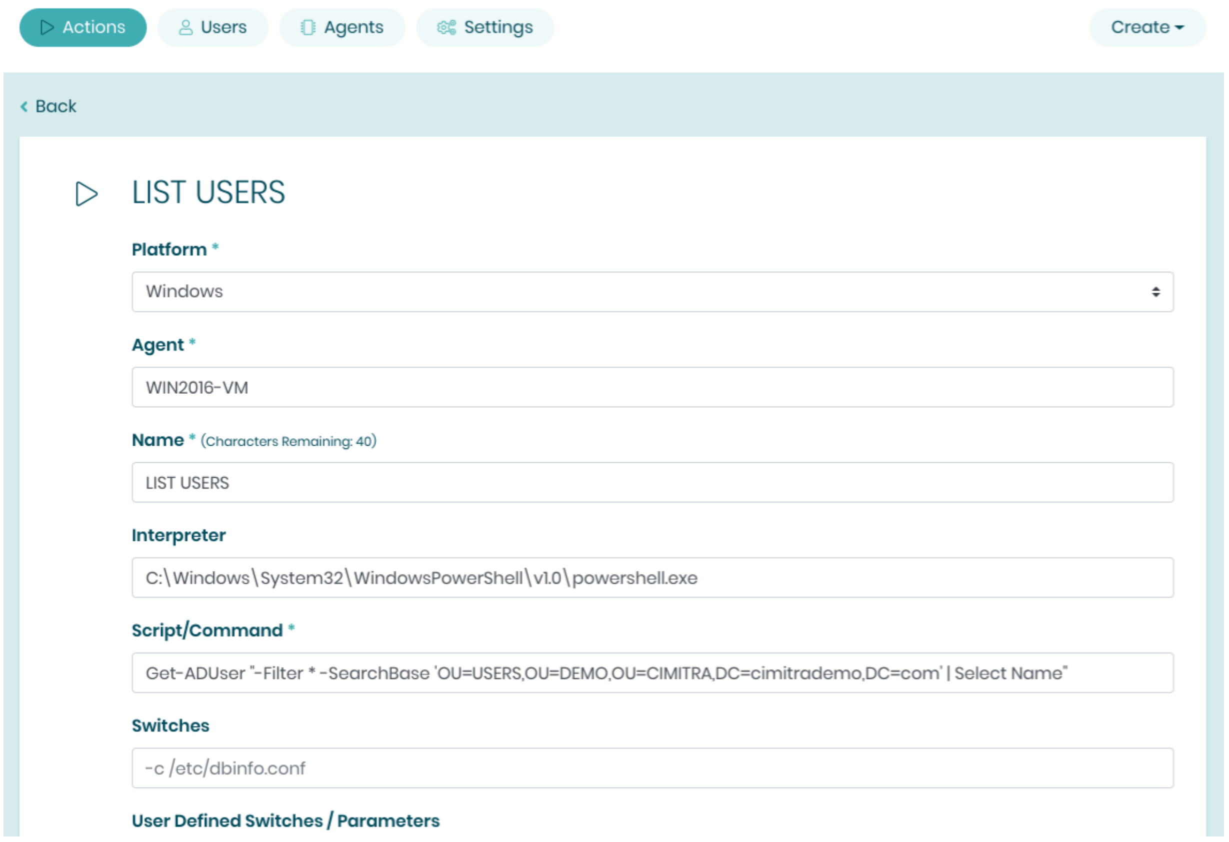
Command Integration Into Cimitra

This assumes that you have already created a Cimitra server deployed a Cimitra Agent etc. to a Windows Server where the PowerShell command will be executed.

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 USERS**
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	`Get-ADUser "-Filter * -SearchBase 'OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com' | Select Name"`



LIST USERS

Platform *
Windows

Agent *
WIN2016-VM

Name * (Characters Remaining: 40)
LIST USERS

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

Script/Command *
Get-ADUser "-Filter * -SearchBase 'OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com' | Select Name"

Switches
-c /etc/dbinfo.conf

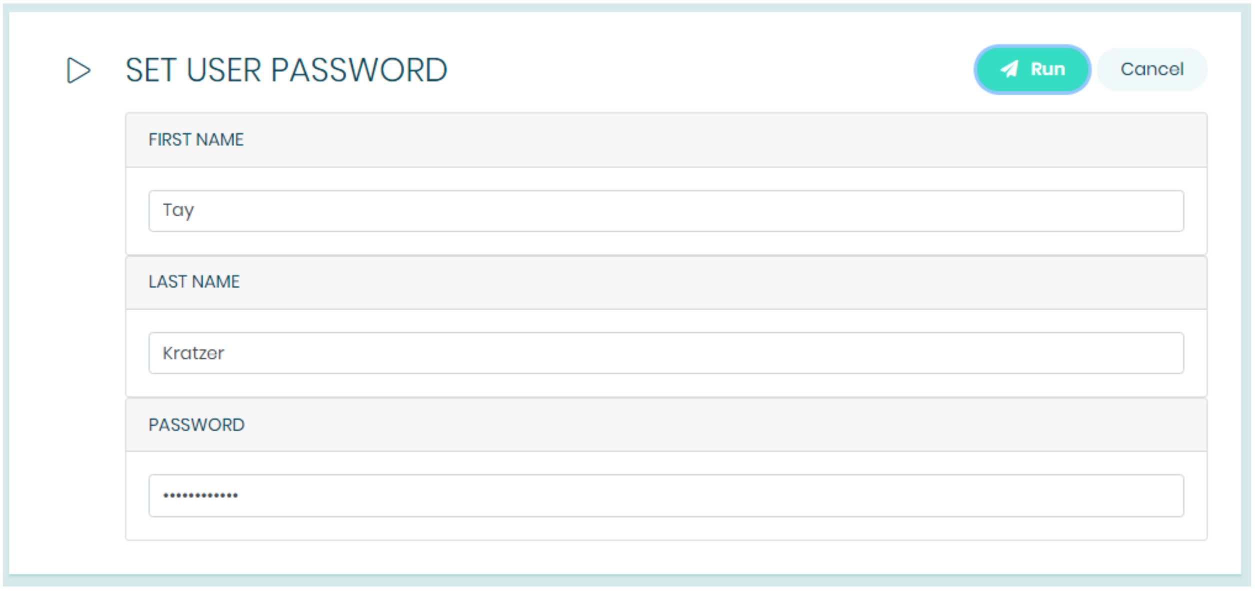
User Defined Switches / Parameters

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.

Setting a User’s Password in Active Directory

Use Case

The Help Desk often gets scenarios in which a user forgot their password. Particularly because some of their users only actually log into Active Directory every few weeks. The Help Desk needs the rights to change passwords, without the IT admin assigning rights and giving access to the User/Computer management console for Active Directory.



SET USER PASSWORD

FIRST NAME

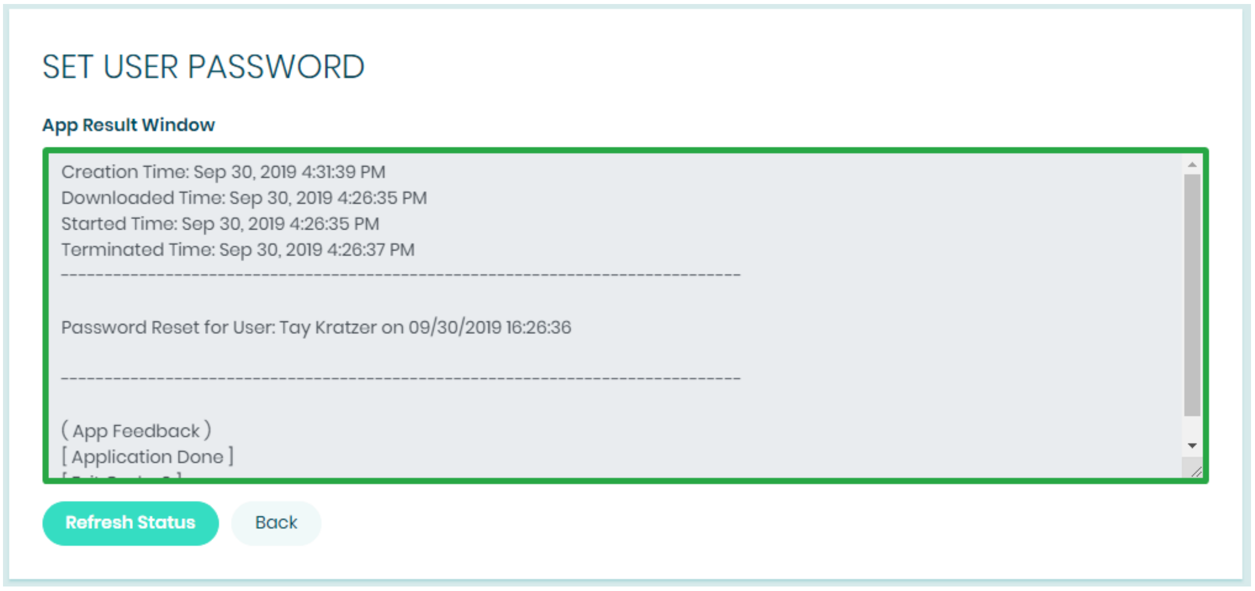
Tay

LAST NAME

Kratzer

PASSWORD

••••••••••••



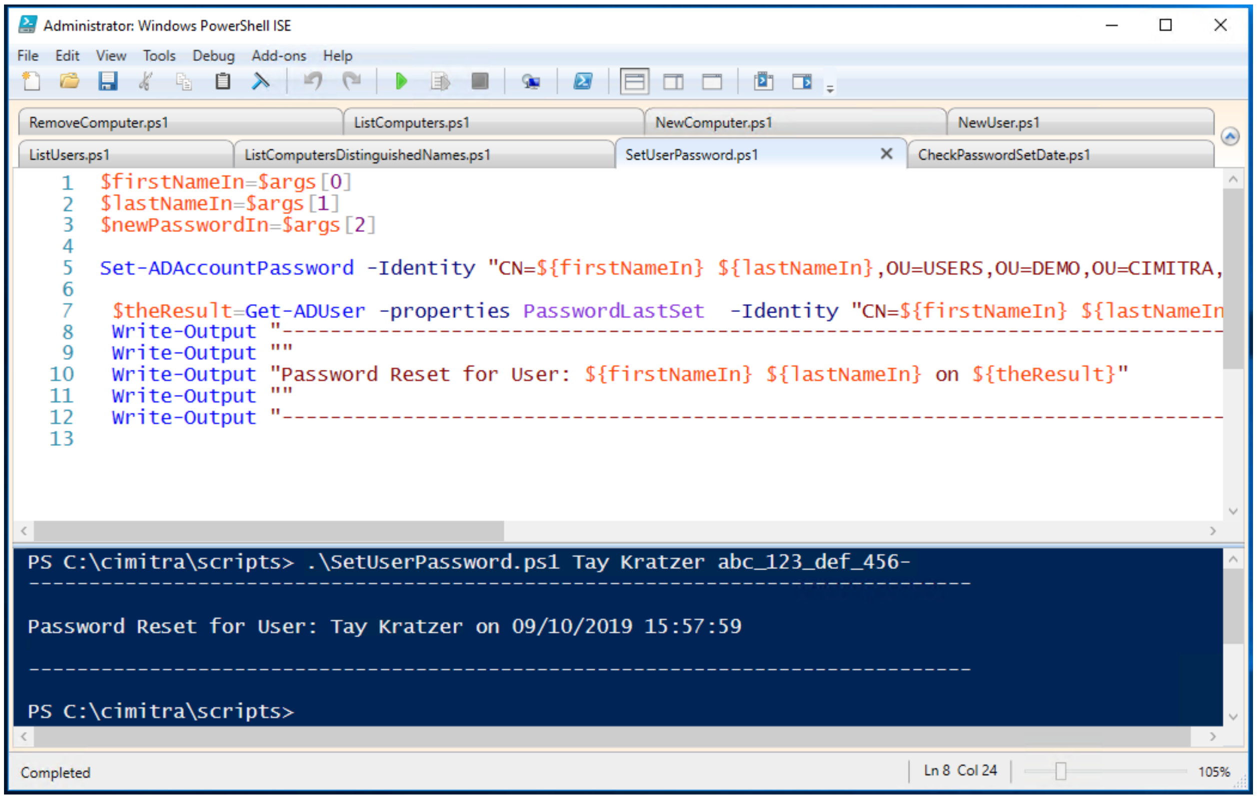
```
Creation Time: Sep 30, 2019 4:31:39 PM
Downloaded Time: Sep 30, 2019 4:26:35 PM
Started Time: Sep 30, 2019 4:26:35 PM
Terminated Time: Sep 30, 2019 4:26:37 PM
--------------------------------------------------------------------------------

Password Reset for User: Tay Kratzer on 09/30/2019 16:26:36

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

( App Feedback )
[ Application Done ]
```

Technical Overview

This PowerShell script will update the password for an existing user object in Active Directory. The script takes in three command line parameters that are converted by the script into variables within the script. These variables are then passed to Active Directory using the `Set-ADAccountPassword` PowerShell command along with supporting variables and required commands.

Script Contents - Copy/Paste - `SetUserPassword.ps1`

```
$firstNameIn=$args[0]
$lastNameIn=$args[1]
$newPasswordIn=$args[2]

Set-ADAccountPassword -Identity "CN=${firstNameIn}
${lastNameIn},OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=
com" -Reset -NewPassword (ConvertTo-SecureString
-AsPlainText "$newPasswordIn" -Force)

 $theResult=Get-ADUser -properties PasswordLastSet
-Identity "CN=${firstNameIn}
${lastNameIn},OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=
com" | Select-Object PasswordLastSet -ExpandProperty
PasswordLastSet
 Write-Output
"------------------------------------------------------------
-------------------"
 Write-Output ""
 Write-Output "Password Reset for User: ${firstNameIn}
${lastNameIn} on ${theResult}"
 Write-Output ""
 Write-Output
"------------------------------------------------------------
-------------------"
```



```
$firstNameIn=$args[0]
$lastNameIn=$args[1]
$newPasswordIn=$args[2]

Set-ADAccountPassword -Identity "CN=${firstNameIn} ${lastNameIn},OU=USERS,OU=DEMO,OU=CIMITRA,

 $theResult=Get-ADUser -properties PasswordLastSet  -Identity "CN=${firstNameIn} ${lastNameIn
 Write-Output "-------------------------------------------------------------------------------
 Write-Output ""
 Write-Output "Password Reset for User: ${firstNameIn} ${lastNameIn} on ${theResult}"
 Write-Output ""
 Write-Output "-------------------------------------------------------------------------------
```

```
PS C:\cimitra\scripts> .\SetUserPassword.ps1 Tay Kratzer abc_123_def_456-
-------------------------------------------------------------------------------

Password Reset for User: Tay Kratzer on 09/10/2019 15:57:59

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

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 **`SetUserPassword.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	**SET USER PASSWORD**
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\SetUserPassword.ps1**
User Defined Switches/Parameters	Click the "**+Add Switch**" option. There will be three switches as shown below. These switches correlate with the three command-line parameters we programmed into the script.

FIRST NAME SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**FIRST NAME**
Validating Regex: Allow: Letters	**/^[A-Za-z]+$/**

LAST NAME SWITCH

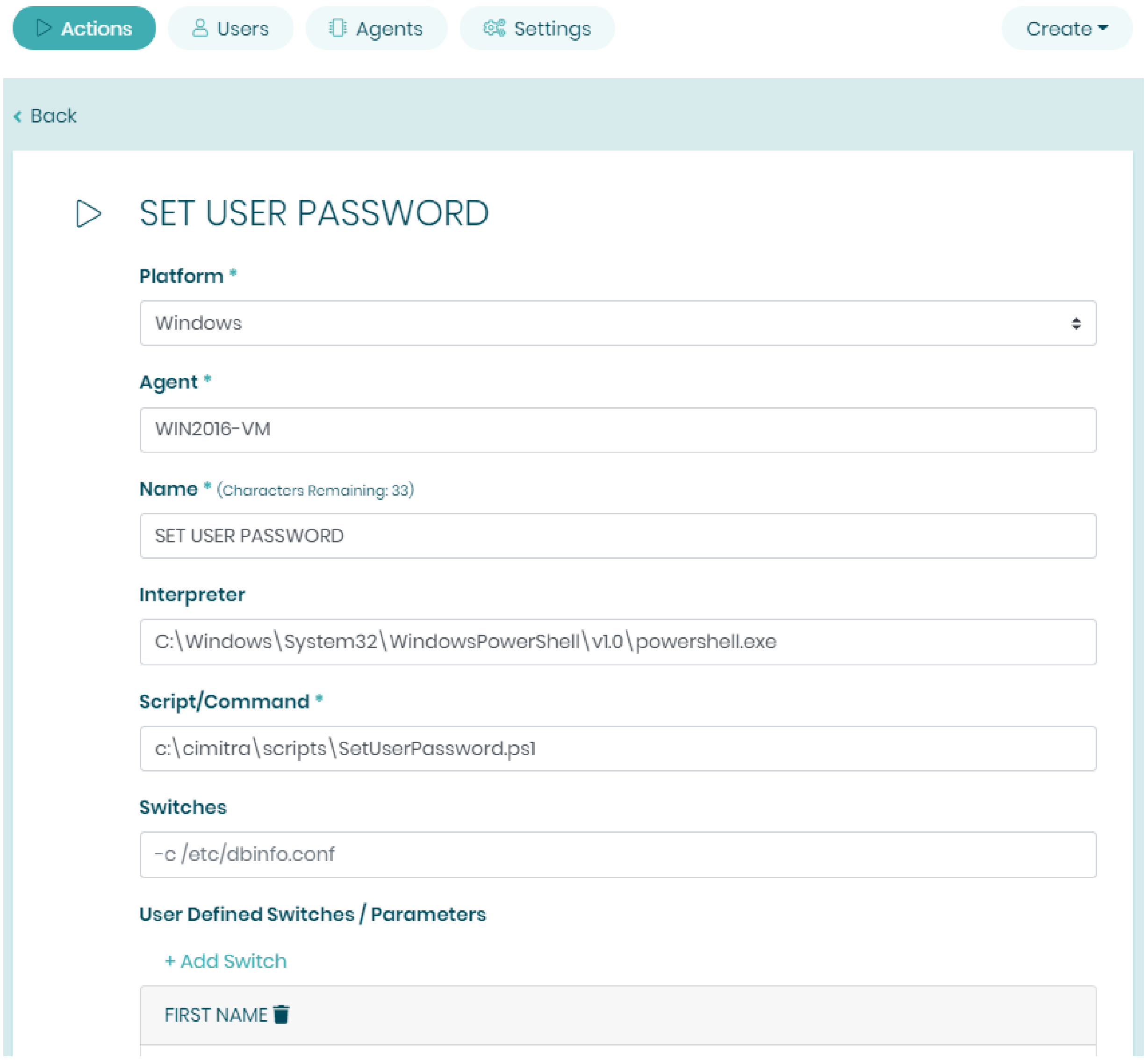
Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**LAST NAME**
Validating Regex: Allow: Letters	**/^[A-Za-z]+$/**

PASSWORD SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	PASSWORD
Validating Regex: Allow: Letters, numbers, dashes, and underscores	**/^[A-Za-z0-9_-]+$/**
Example:	(Letters, Numbers, Dash "-", Underscores "_")
Mask:	**ENABLE THIS**

INFORMATION FIELD

NOTE: The password should be 8 characters long, and include a number and an underscore or dash and an uppercase letter.



SET USER PASSWORD

Platform *
Windows

Agent *
WIN2016-VM

Name * (Characters Remaining: 33)
SET USER PASSWORD

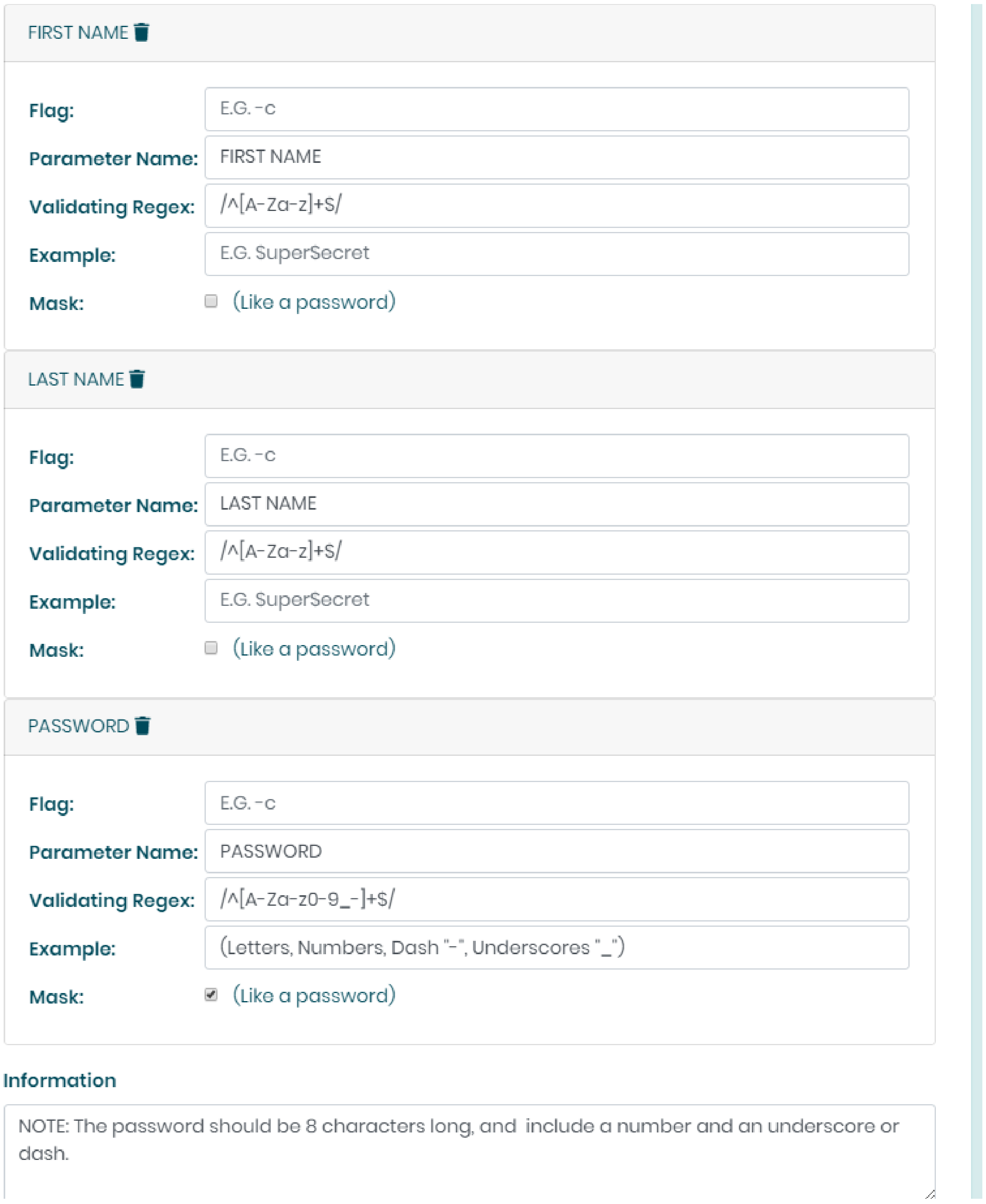
Interpreter
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

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

Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters

FIRST NAME



FIRST NAME

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

LAST NAME

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

PASSWORD

Flag:	E.G. -c
Parameter Name:	PASSWORD
Validating Regex:	/^[A-Za-z0-9_-]+$/
Example:	(Letters, Numbers, Dash "-", Underscores "_")
Mask:	✓ (Like a password)

Information

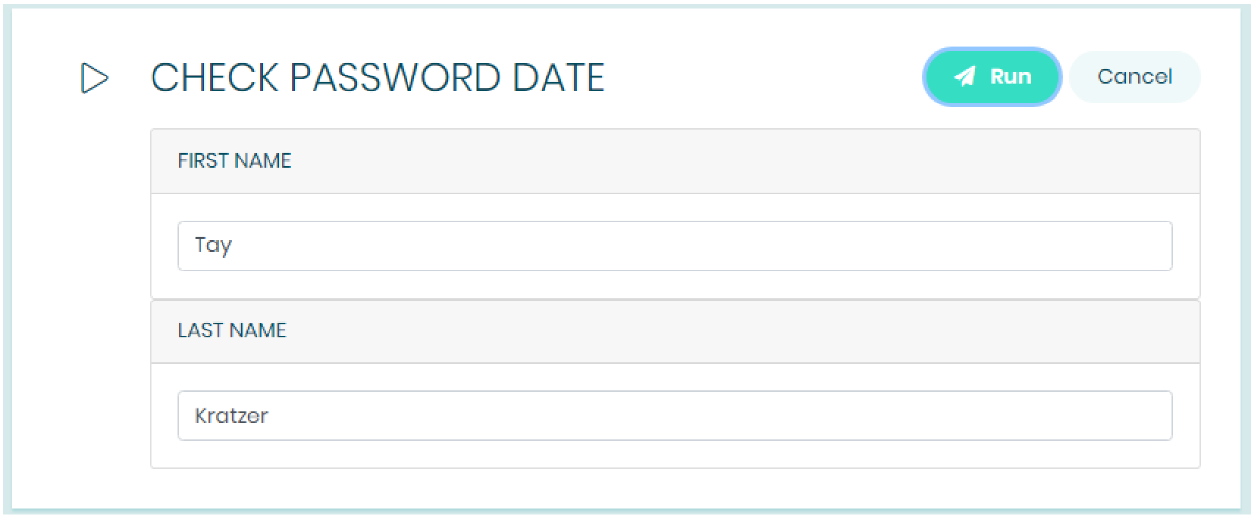
NOTE: The password should be 8 characters long, and include a number and an underscore or dash.

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.

Checking a User’s Password Reset Date

Use Case

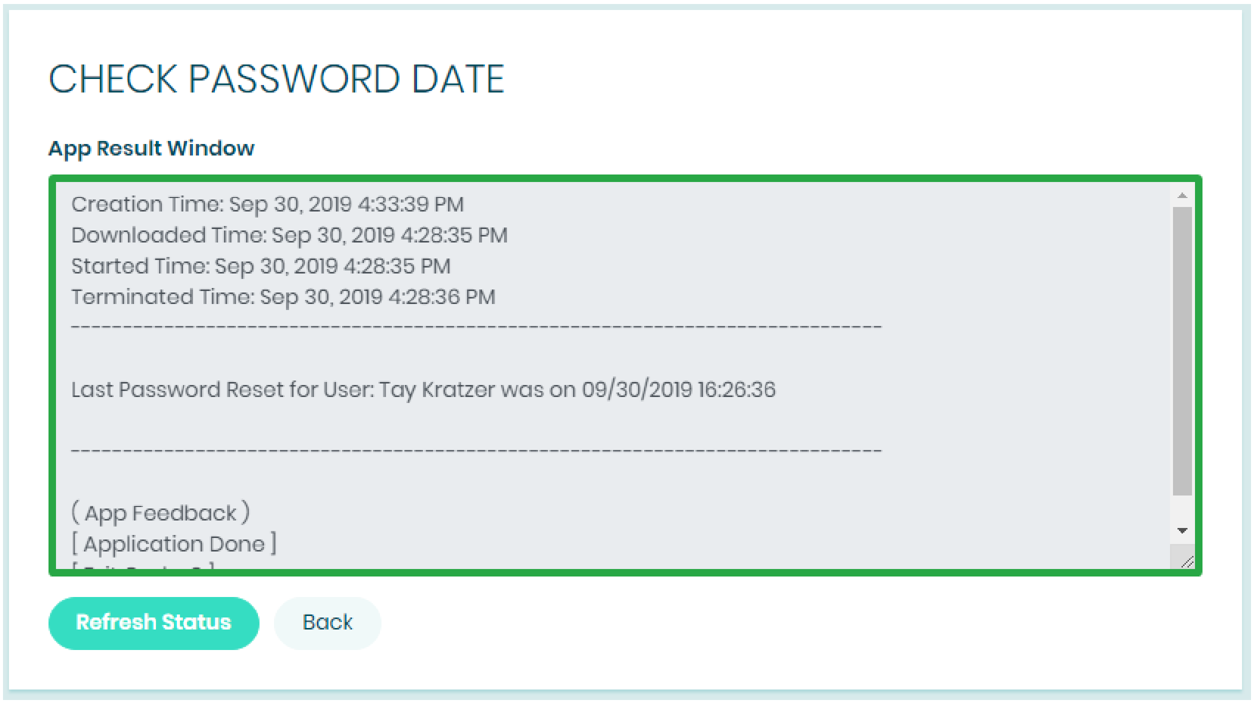
A user’s password was just changed inside of Cimitra, but the person using Cimitra wants to double-check their work and see that the password change actually happened.



CHECK PASSWORD DATE

FIRST NAME: Tay

LAST NAME: Kratzer



CHECK PASSWORD DATE

App Result Window

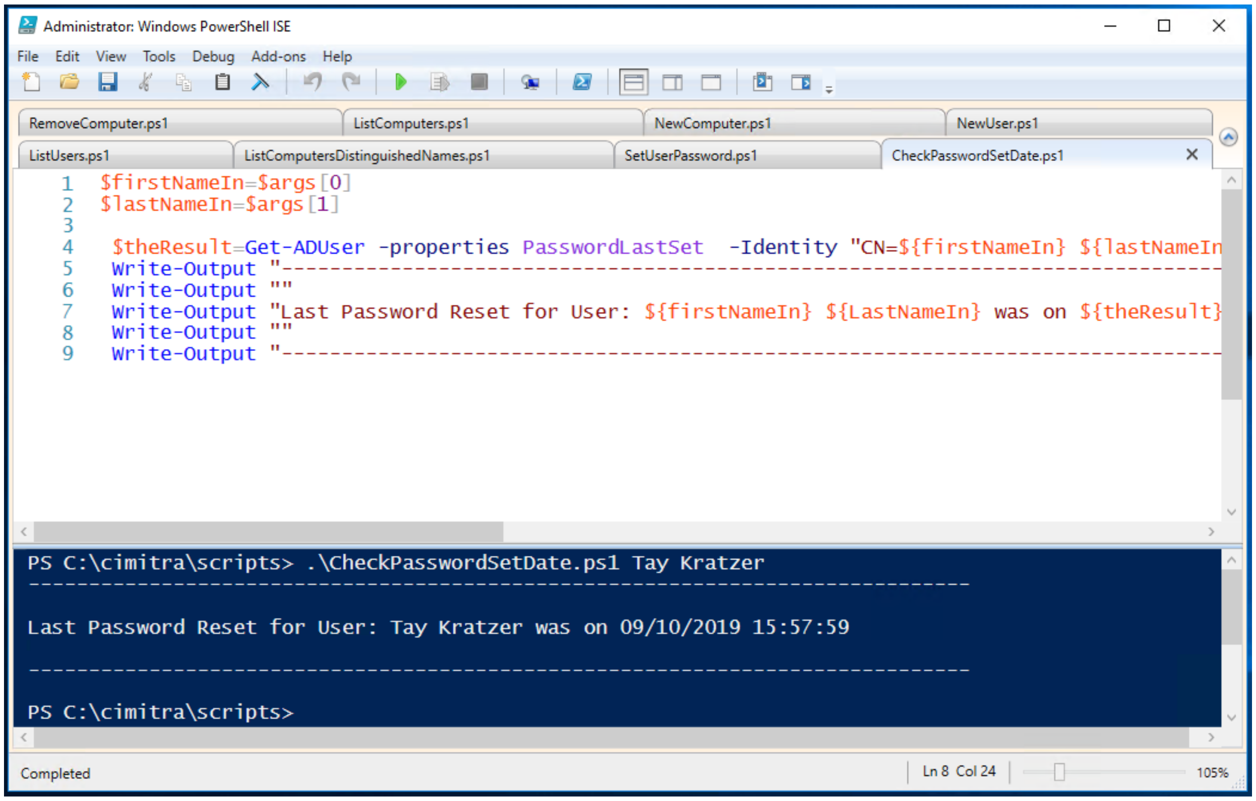
```
Creation Time: Sep 30, 2019 4:33:39 PM
Downloaded Time: Sep 30, 2019 4:28:35 PM
Started Time: Sep 30, 2019 4:28:35 PM
Terminated Time: Sep 30, 2019 4:28:36 PM
--------------------------------------------------------------------------------

Last Password Reset for User: Tay Kratzer was on 09/30/2019 16:26:36

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

( App Feedback )
[ Application Done ]
```

Technical Overview

This PowerShell script takes in two parameters to identify the user, they are the user's first name and last name. The script then uses PowerShell commands to extract a property called "**PasswordLastSet**" from the user's Active Directory.

Script Contents - Copy Paste - **CheckPasswordSetDate.ps1**

```
firstNameIn=$args[0]
$lastNameIn=$args[1]

 $theResult=Get-ADUser -properties PasswordLastSet
-Identity "CN=${firstNameIn}
${lastNameIn},OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=
com" | Select-Object PasswordLastSet -ExpandProperty
PasswordLastSet
 Write-Output
"------------------------------------------------------------
-------------------"
 Write-Output ""
 Write-Output "Last Password Reset for User: ${firstNameIn}
${LastNameIn} was on ${theResult}"
 Write-Output ""
 Write-Output
"------------------------------------------------------------
-------------------"
```



```
$firstNameIn=$args[0]
$lastNameIn=$args[1]

 $theResult=Get-ADUser -properties PasswordLastSet  -Identity "CN=${firstNameIn} ${lastNameIn
 Write-Output "------------------------------------------------------------------------------
 Write-Output ""
 Write-Output "Last Password Reset for User: ${firstNameIn} ${LastNameIn} was on ${theResult}
 Write-Output ""
 Write-Output "------------------------------------------------------------------------------
```

```
PS C:\cimitra\scripts> .\CheckPasswordSetDate.ps1 Tay Kratzer
------------------------------------------------------------------------------

Last Password Reset for User: Tay Kratzer was on 09/10/2019 15:57:59

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

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 **`CheckPasswordSetDate.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	**CHECK PASSWORD DATE**
Interpreter	<Path to PowerShell Interpreter> Example:

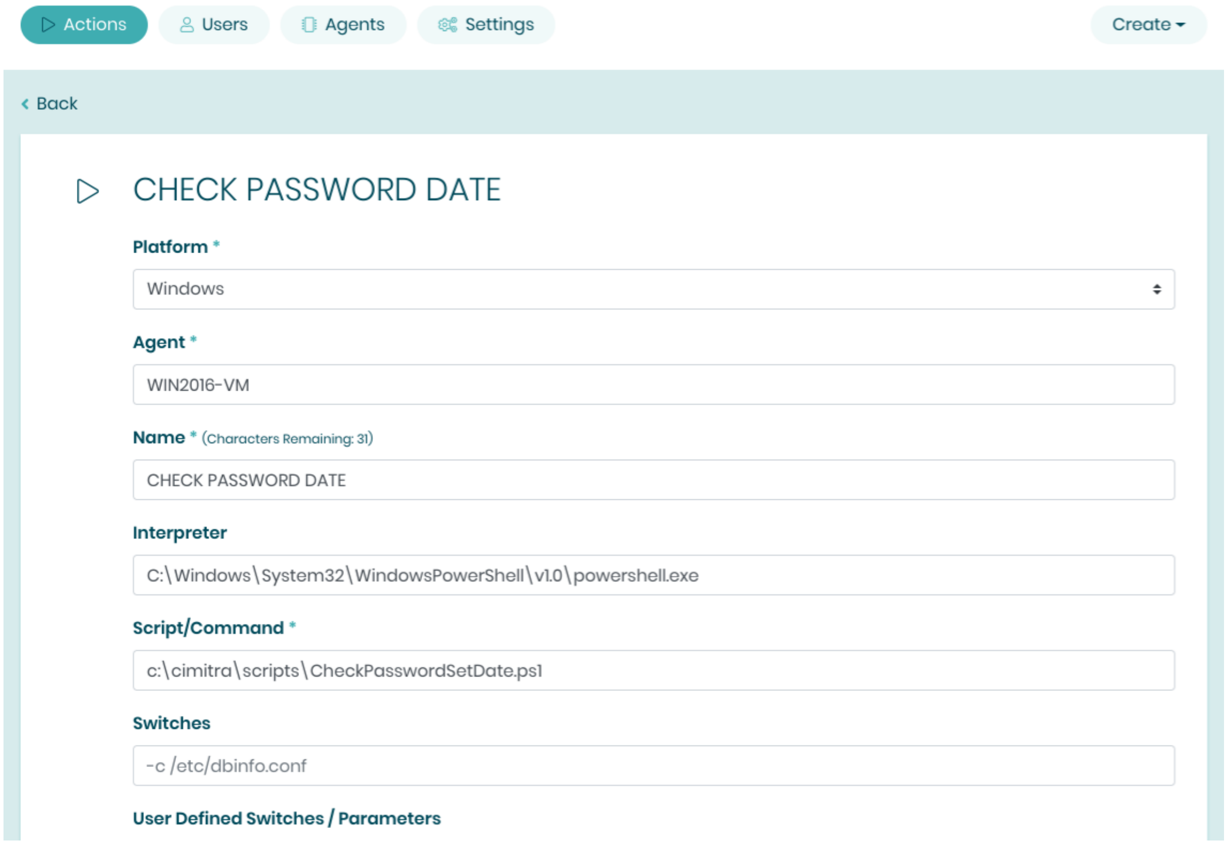
	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Script/Command	<Path to the Cimitra Script> Example: **c:\cimitra\scripts\CheckPasswordSetDate.ps1**
User Defined Switches/Parameters	Click the "**+Add Switch**" option. There will be two switches as shown below. These switches correlate with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

FIRST NAME SWITCH

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

LAST NAME SWITCH

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



CHECK PASSWORD DATE

Platform *
Windows

Agent *
WIN2016-VM

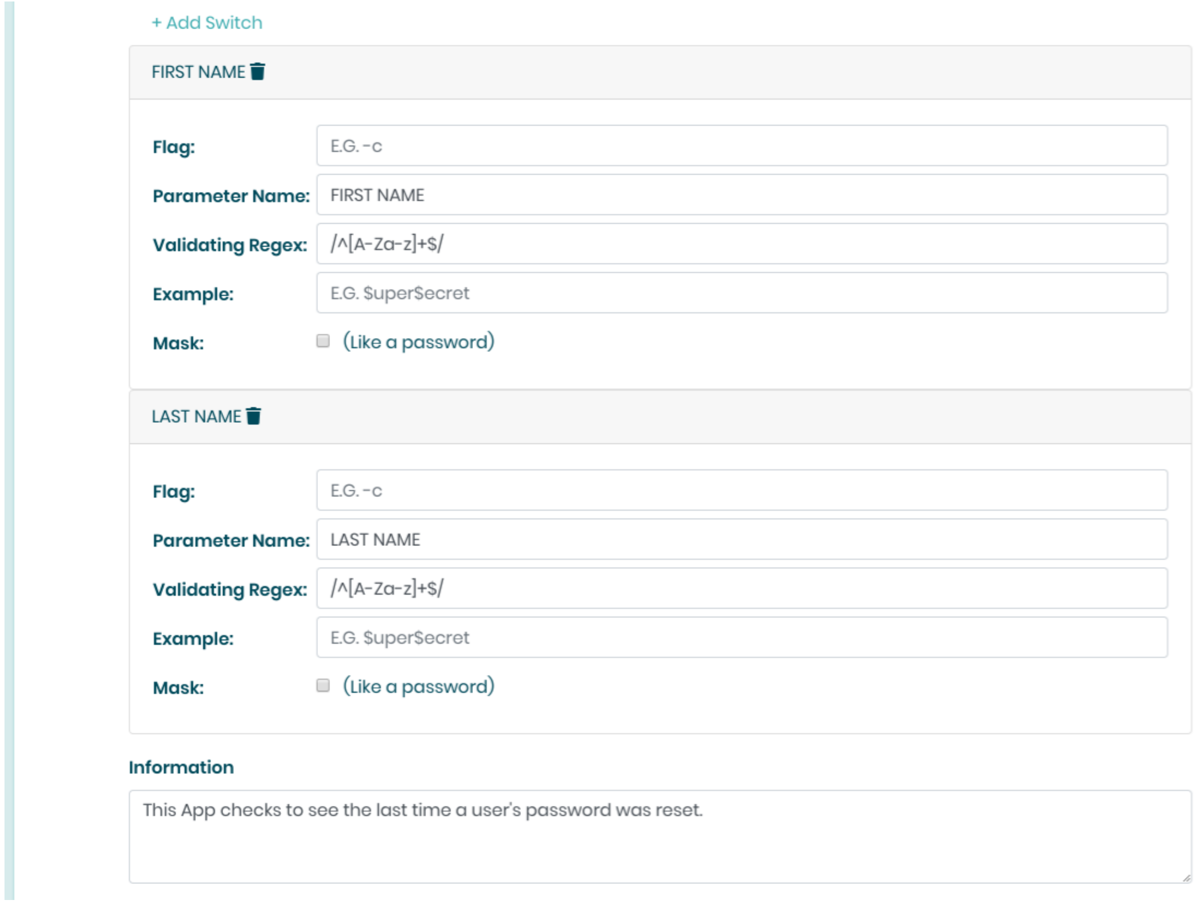
Name * (Characters Remaining: 31)
CHECK PASSWORD DATE

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

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

Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters



+ Add Switch

FIRST NAME

Flag:	E.G. -c
Parameter Name:	FIRST NAME
Validating Regex:	/^[A-Za-z]+$/
Example:	E.G. $uper$ecret
Mask:	☐ (Like a password)

LAST NAME

Flag:	E.G. -c
Parameter Name:	LAST NAME
Validating Regex:	/^[A-Za-z]+$/
Example:	E.G. $uper$ecret
Mask:	☐ (Like a password)

Information

This App checks to see the last time a user's password was reset.

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.

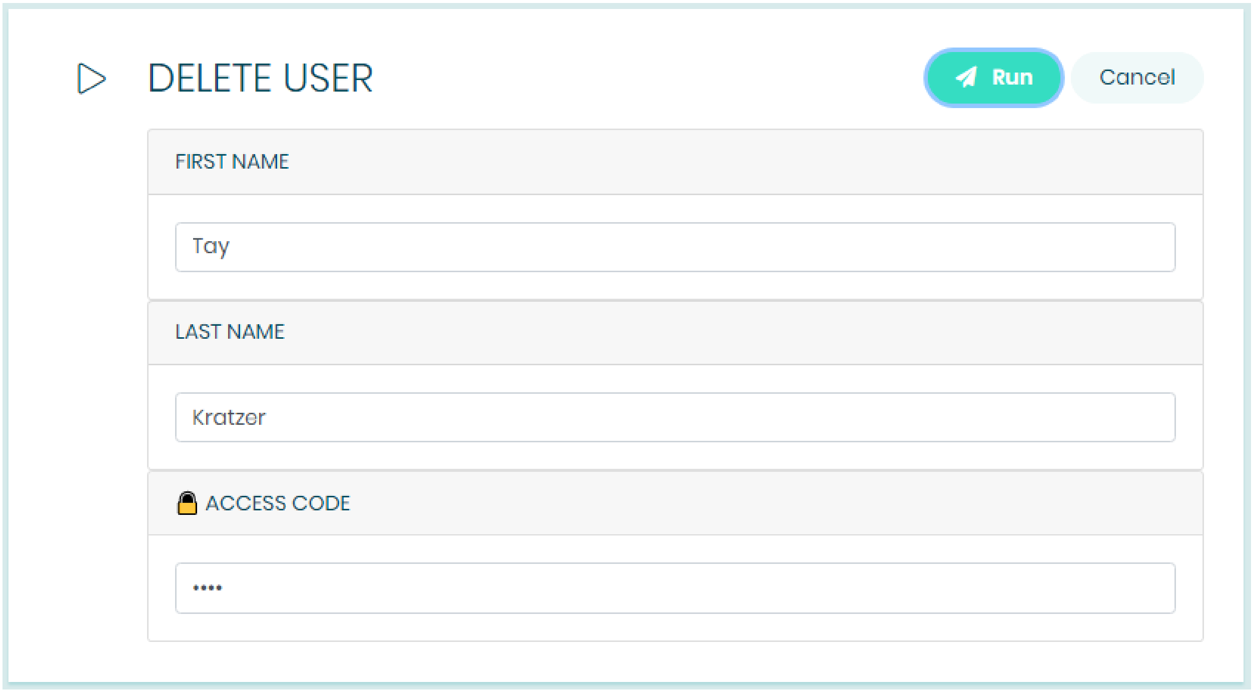
Removing a User From Active Directory

This Cimitra App has a unique feature we will refer to as an “**Access Code**”. The Access Code is a method for making a Cimitra App that needs a special hidden code in order for it to run. The key to making this feature work is that the script that you write must ignore the “Access Code”, because the Access Code is simply a mechanism for getting the Cimitra client to challenge the user with an Access Code to assure they have the correct access to run a script.

Here is the trick, the **Access Code** needs to be the last switch you create associated with the Cimitra App. So although the **Access Code** input is passed to the PowerShell script, it is ignored since it is never converted a variable with in the PowerShell script. Follow along, and you will see how it is done.

Use Case

Some people on the Help Desk you trust to delete users from Active Directory, in a few contexts in the Active Directory tree. You need to give them a method for doing so without giving them access to the native “Active Directory Users and Computers” console. You want to share the Cimitra App in a folder along with several other Apps, but you only want the user’s you have given the code to, to actually have access to run the App.

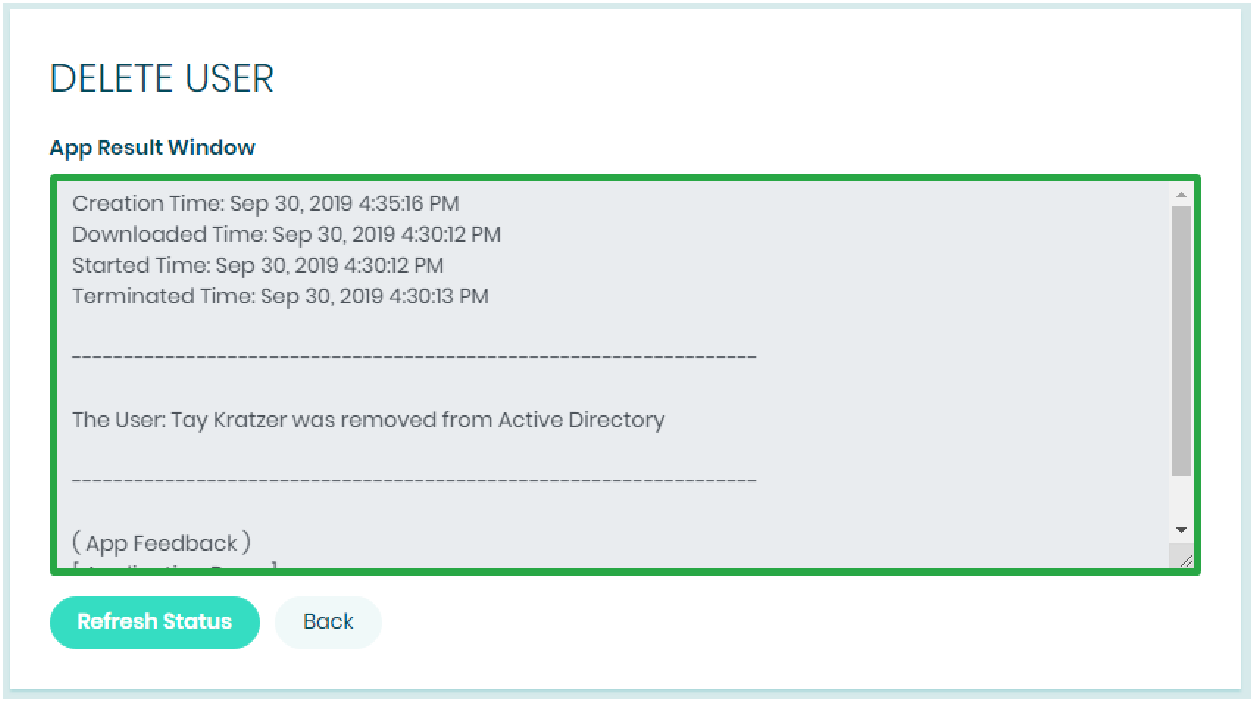


DELETE USER

FIRST NAME: Tay

LAST NAME: Kratzer

ACCESS CODE: ••••



```
Creation Time: Sep 30, 2019 4:35:16 PM
Downloaded Time: Sep 30, 2019 4:30:12 PM
Started Time: Sep 30, 2019 4:30:12 PM
Terminated Time: Sep 30, 2019 4:30:13 PM

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

The User: Tay Kratzer was removed from Active Directory

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

( App Feedback )
```

Technical Overview

This script has two parameters. They are the user's first and last name. Although in this example, you will create a third parameter in the Cimitra App, it will be ignored by the PowerShell script. The Cimitra App needs the third parameter to create the "**Access Code**" feature explained earlier.

Script Contents - Copy Paste - `RemoveUser.ps1`

```
$firstNameIn=$args[0]
$lastNameIn=$args[1]


# Use Remove-ADUser to remove the user
Remove-ADUser  -Identity "CN=$firstNameIn
$lastNameIn,OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=co
m" -Confirm:$False
```

```
Write-Output ""
Write-Output
"-------------------------------------------------------------
-------"
Write-Output ""
Write-Output "The User: ${firstNameIn} ${lastNameIn} was
removed from Active Directory"
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 **`RemoveUser.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	**DELETE USER**
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\RemoveUser.ps1
User Defined Switches/Parameters	Click the "**+Add Switch**" option. There will be three switches as shown below. The first two switches correlate with the two command-line parameters we programmed into the script. The last switch is used for the **Access Code** feature.

FIRST NAME SWITCH

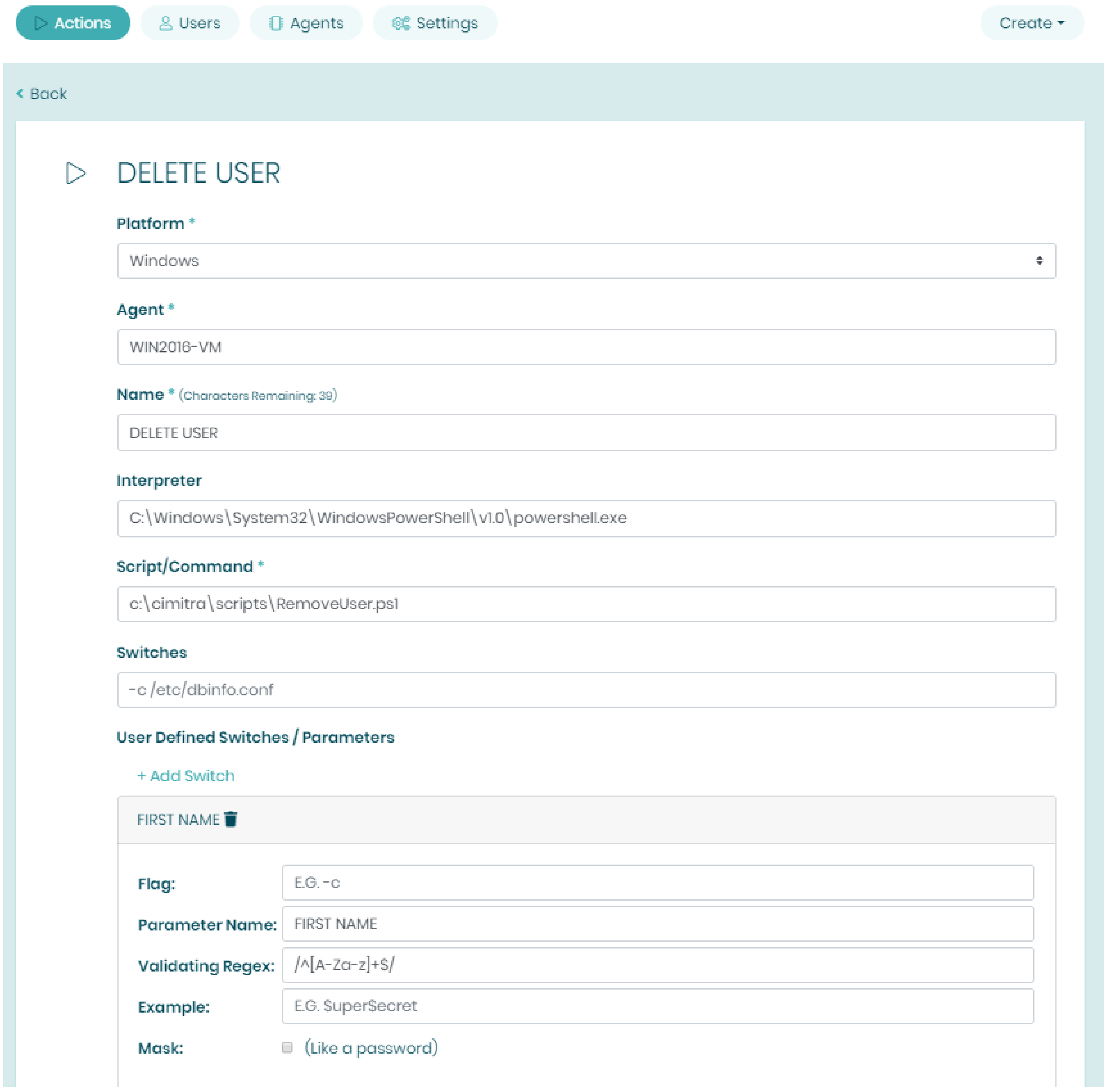
Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**FIRST NAME**
Validating Regex: Allow: Letters, and Underscores	**/^[A-Za-z]+$/**

LAST NAME SWITCH

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

ACCESS CODE SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	🔒 **ACCESS CODE**
Validating Regex: This is the trick to the "Access Code" functionality. The access code in this example is: **DoIT**	**/^DoIT$/**
Example:	
Mask:	**ENABLE THIS**



DELETE USER

Platform *
Windows

Agent *
WIN2016-VM

Name * (Characters Remaining: 39)
DELETE USER

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

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

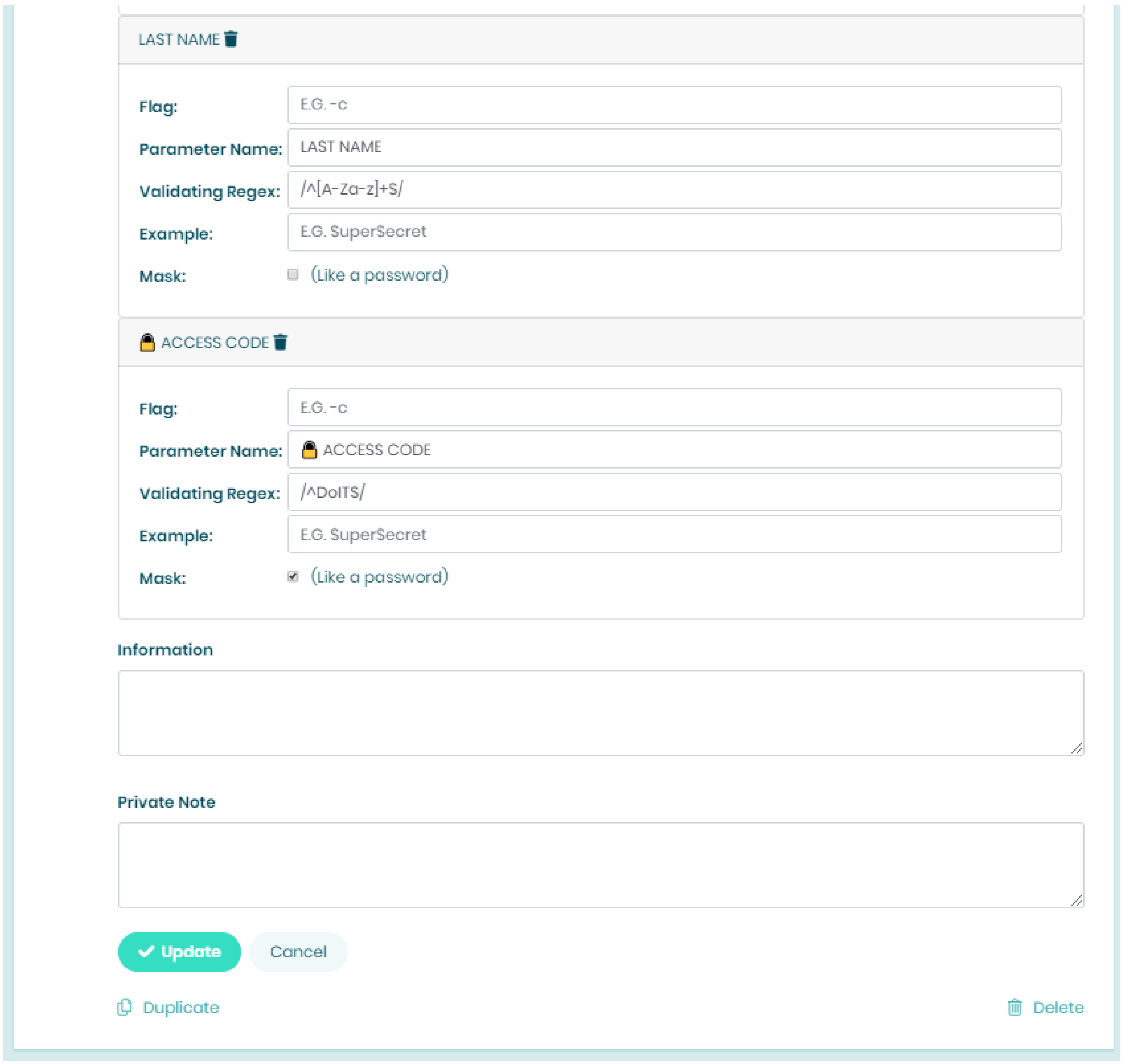
Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters

+ Add Switch

FIRST NAME

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



LAST NAME

Flag:	E.G. -c
Parameter Name:	LAST NAME
Validating Regex:	/^[A-Za-z]+$/
Example:	E.G. Super$ecret
Mask:	☐ (Like a password)

ACCESS CODE

Flag:	E.G. -c
Parameter Name:	ACCESS CODE
Validating Regex:	/^DoITS/
Example:	E.G. Super$ecret
Mask:	☑ (Like a password)

Information

Private Note

New User in Active Directory - Different Method

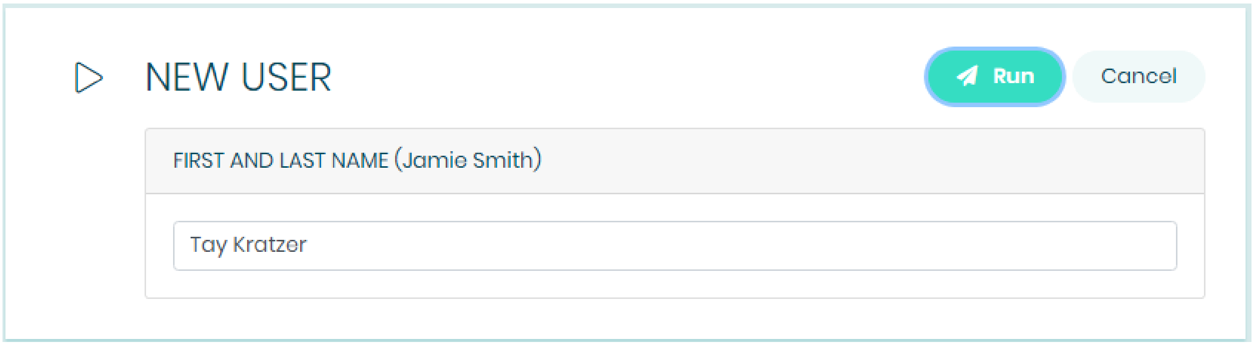
Use Case

New users need basic access to the network. The HR department and Help Desk have been tasked with creating users so that they can immediately get access to some network kiosks.

This method of adding a new user is different from the first method explained in this document. Basically this method does two things differently:

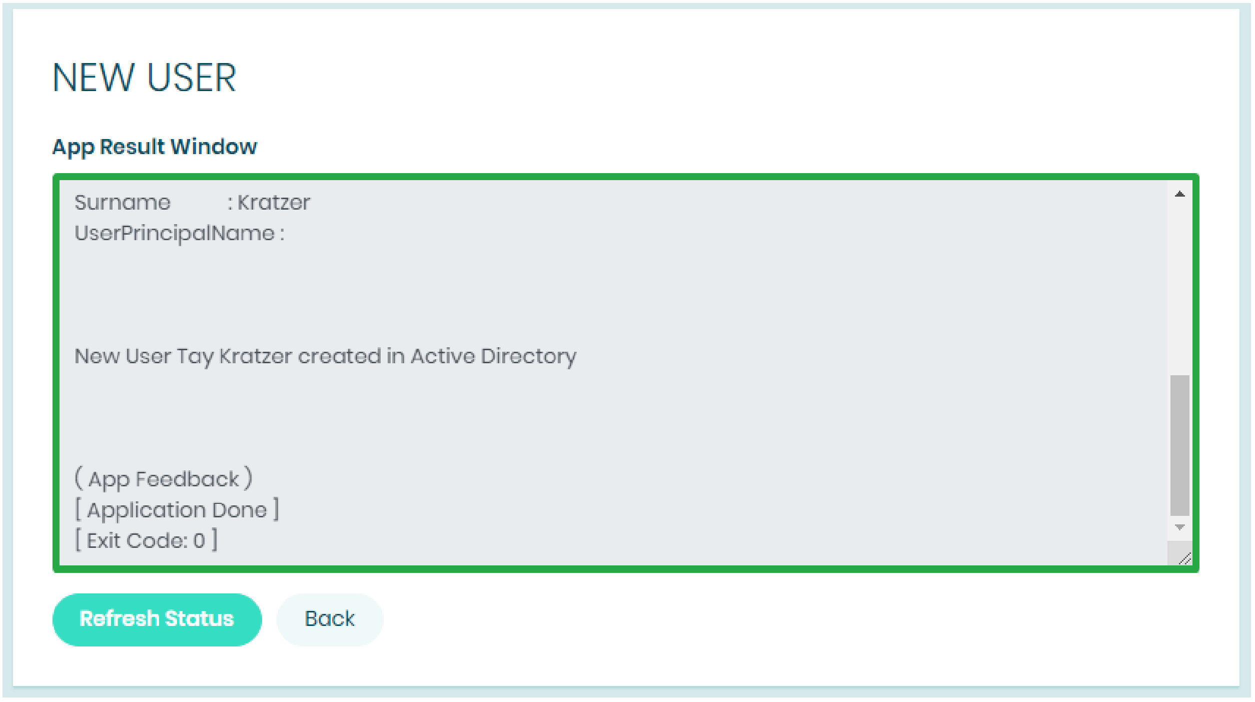
1. Only one field is used, and the First and Last name are parsed as the first and second words passed to the script.

2. A default password is also set for the user



FIRST AND LAST NAME (Jamie Smith)

Tay Kratzer



```
Surname          : Kratzer
UserPrincipalName :



New User Tay Kratzer created in Active Directory



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

Technical Overview

This PowerShell script will create a user object in Active Directory. The script takes in two command line parameters that are converted by the script into two variables in the script. These variables are then passed to Active Directory using the `New-AdUser` command along with supporting variables and required commands.

Script Contents - Copy Paste | **`NewUserNoPassword.ps1`**

```
$firstNameIn=$args[0]
$lastNameIn=$args[1]
$passwordIn = 'Changeme_123'
$samAccountName = $firstNameIn[0]+'.'+$lastNameIn


New-ADUser -Name "$firstNameIn $lastNameIn" -GivenName
"$firstNameIn" -Surname "$lastNameIn" -SamAccountName
"$samAccountName" -AccountPassword (ConvertTo-SecureString
"$passwordIn" -AsPlainText -force) -passThru -path
"OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=com"
$theResult = $?

if ($theResult = 'True')
{
Write-Output ""
Write-Output ""
Write-Output "New User ${firstNameIn} ${lastNameIn} created
in Active Directory"
}

Enable-ADAccount -Identity "CN=$firstNameIn
$lastNameIn,OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=co
m" -Confirm:$False

Set-ADUser -Identity  "CN=$firstNameIn
$lastNameIn,OU=USERS,OU=DEMO,OU=CIMITRA,DC=cimitrademo,DC=co
m" -ChangePasswordAtLogon $true
```

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 `NewUserNoPassword.ps1` script exists.

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

CIMITRA APP PROPERTIES

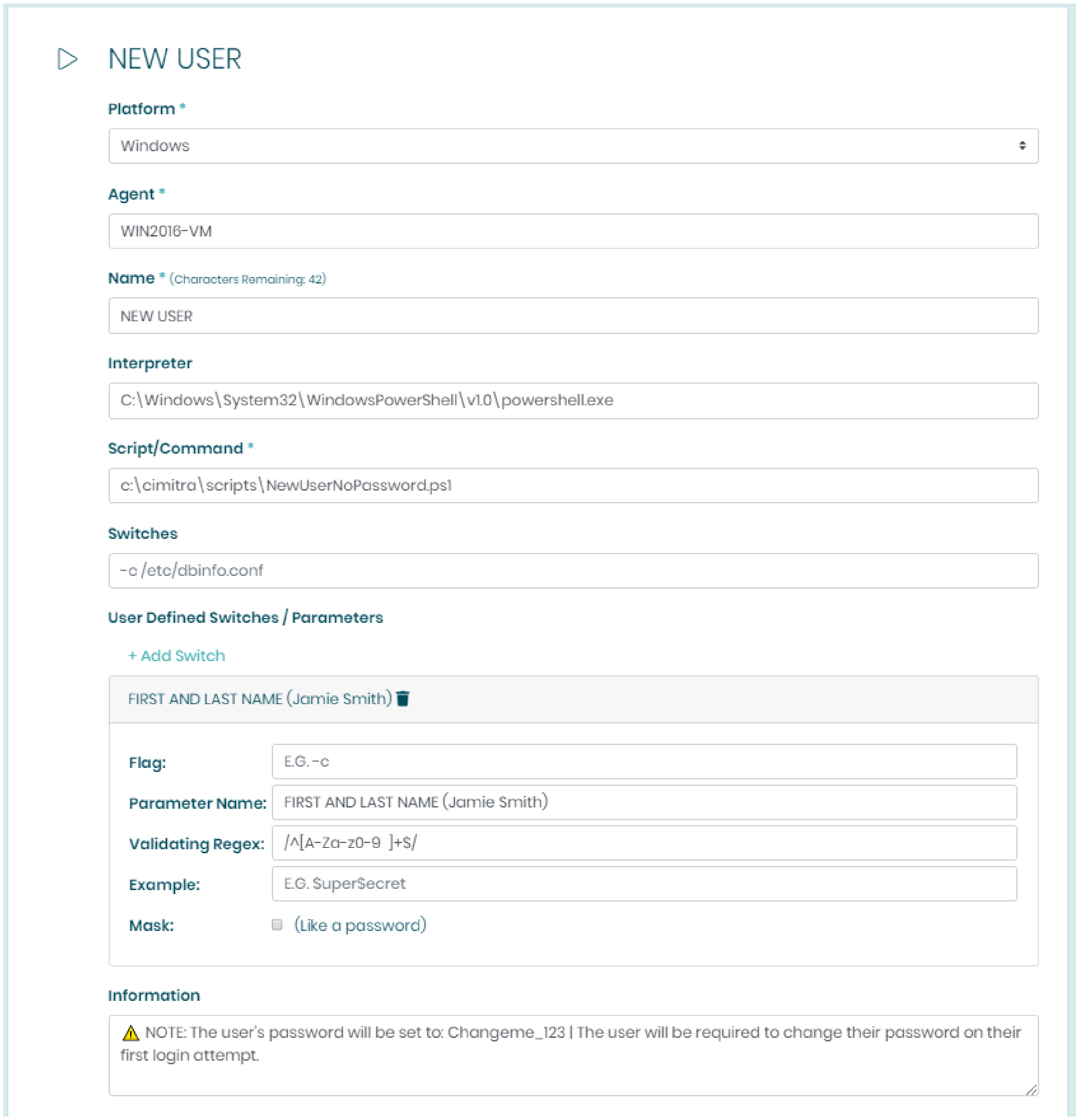
Property	Value
Platform	**Windows**
Agent	<The Cimitra Agent deployed to the Windows Server>
Name	**NEW USER**
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	<Path to the Cimitra Script> Example: **c:\cimitra\scripts\NewUserNoPassword.ps1**
User Defined Switches/Parameters	Click the "**+Add Switch**" option. There will be one switch as shown below. This switch is meant to have two words passed into it, the first name is the first word and the last name is the second word.

FIRST AND LAST NAME SWITCH

Flag:	<LEAVE THIS FIELD BLANK>
Parameter Name:	**FIRST AND LAST NAME (Jamie Smith)**
Validating Regex: Allow: Letters, Numbers, and spaces.	**/^[A-Za-z0-9]+$/**

INFORMATION FIELD

⚠ NOTE: The user's password will be set to: Changeme_123 | The user will be required to change their password on their first login attempt.



▷ NEW USER

Platform *

Windows

Agent *

WIN2016-VM

Name * (Characters Remaining: 42)

NEW USER

Interpreter

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

Script/Command *

c:\cimitra\scripts\NewUserNoPassword.ps1

Switches

-c /etc/dbinfo.conf

User Defined Switches / Parameters

+ Add Switch

FIRST AND LAST NAME (Jamie Smith)

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

Information

⚠ NOTE: The user's password will be set to: Changeme_123 | The user will be required to change their password on their first login attempt.