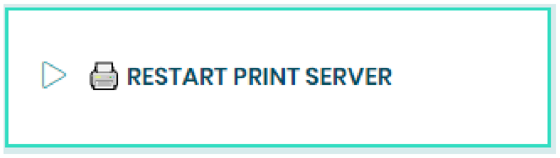
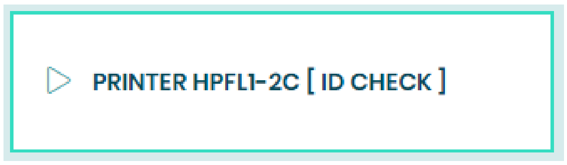
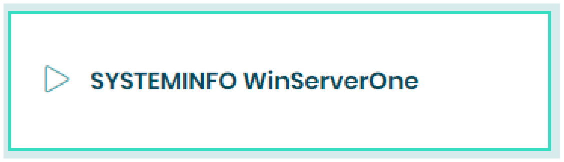
Windows Server Scripts

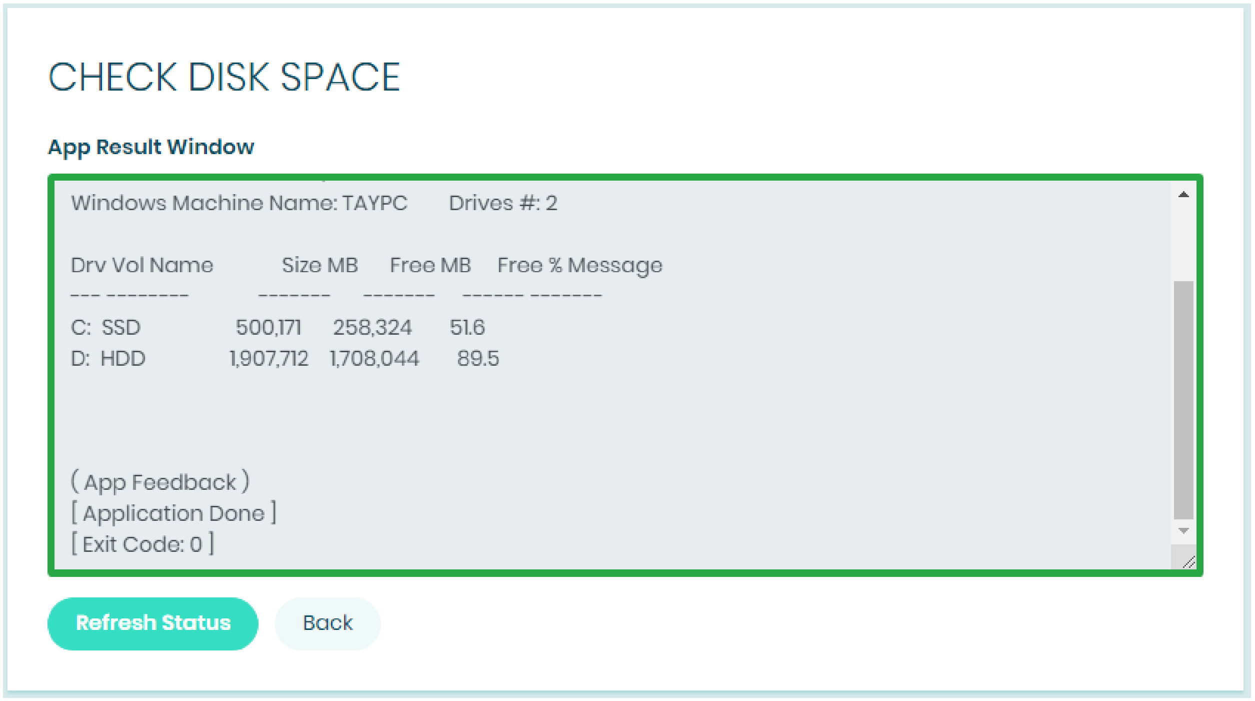
Topic	**Documentation**
▷ CHECK DISK SPACE - WinServerOne	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste Download Scripts (HTTP) GIT **`git clone https://github.com/cimitrasoftware/powershell_scripts.git`**
 ▷ RESTART PRINT SERVER	Use Case Script Integration Into Cimitra Script Contents - Copy/Paste Download Scripts (HTTP) GIT **`git clone https://github.com/cimitrasoftware/powershell_scripts.git`**
 ▷ PRINTER HPFL1-2C [ID CHECK]	Use Case Command Integration Into Cimitra Operating System Command - Copy/Paste
▷ IPCONFIG WinServerOne	Use Case Command Integration Into Cimitra Operating System Command - Copy/Paste

 SYSTEMINFO WinServerOne	Use Case Command Integration Into Cimitra Operating System Command - Copy/Paste

Checking Server Disk Space

Use Case

Multiple people are affected by a particular Windows server. When things such as a database dump or other actions are taken, the disk space on the Windows server is affected, which affects many other people who use this Windows server as a shared resource. Now multiple people can go in and see how the server's disk space is fairing, and how their action might have affected the disk space.


CHECK DISK SPACE

App Result Window

```
Windows Machine Name: TAYPC       Drives #: 2

Drv Vol Name          Size MB     Free MB     Free % Message
--- ---------         -------     -------     ------ -------
C:  SSD              500,171     258,324       51.6
D:  HDD            1,907,712   1,708,044       89.5



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

Technical Overview

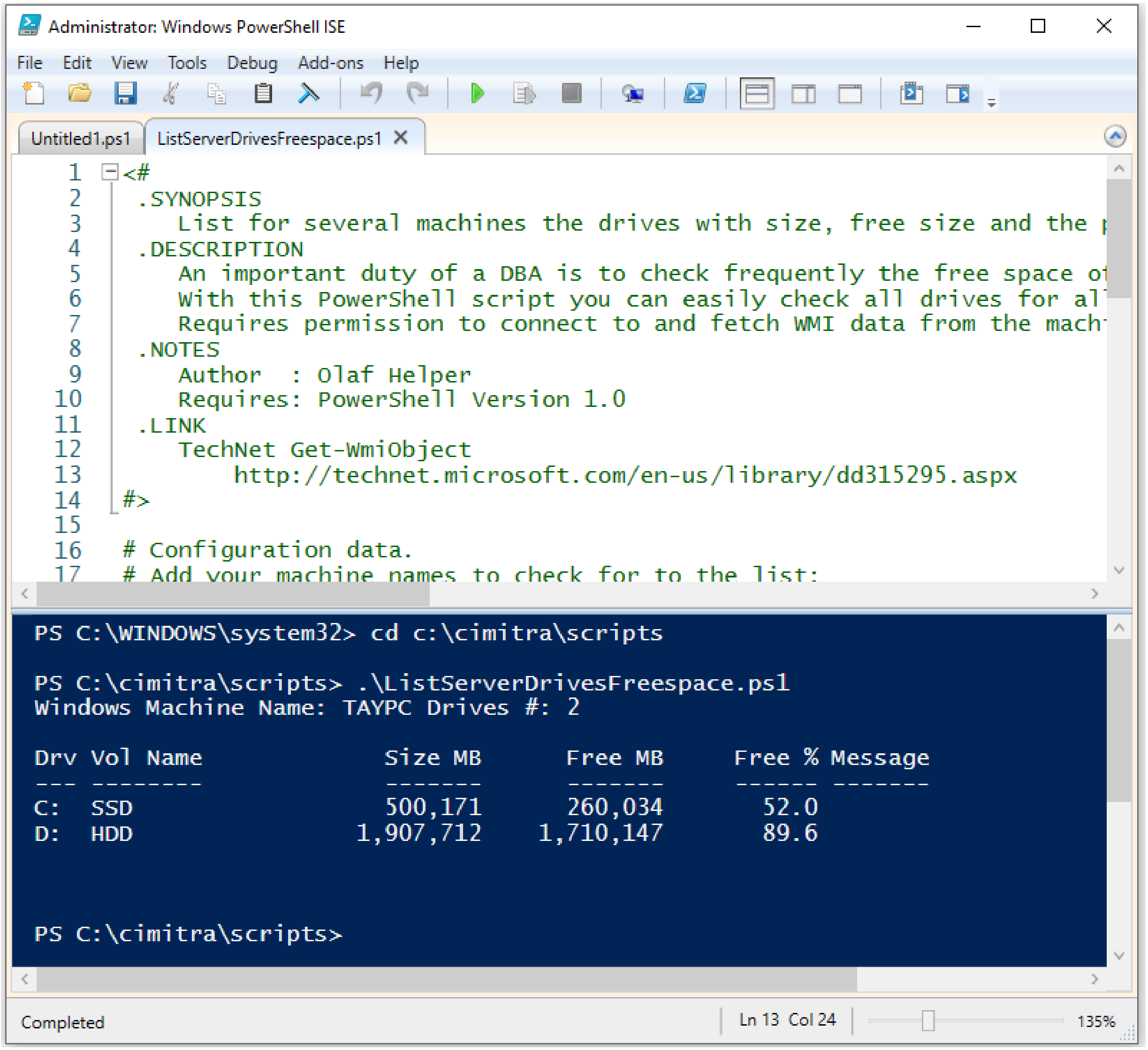
This script was obtained from Microsoft's free PowerShell script repository. It gets the disk space for all of the physical drives on a Windows box and then makes the output more human-friendly

Script Contents - Copy Paste | **ListServerDrivesFreespace.ps1**

```
<#
 .SYNOPSIS
    List for several machines the drives with size, free
size and the percentage of free space.
 .DESCRIPTION
    An important duty of a DBA is to check frequently the
free space of the drives the SQL Server is using to avoid a
database crash if a drive is full.
    With this PowerShell script you can easily check all
drives for all servers in the given list. You can configure
threshold value for Warning & Alarm level.
    Requires permission to connect to and fetch WMI data
from the machine(s).
 .NOTES
    Author  : Olaf Helper
    Requires: PowerShell Version 1.0
 .LINK
    TechNet Get-WmiObject

http://technet.microsoft.com/en-us/library/dd315295.aspx
#>

# Configuration data.
# Add your machine names to check for to the list:
[Array] $servers = (Get-WmiObject -Class
Win32_ComputerSystem -Property Name).Name;
[float] $levelWarn  = 20.0;  # Warn-level in percent.
[float] $levelAlarm = 10.0;  # Alarm-level in percent.

# Defining output format for each column.
```

```
$fmtDrive =@{label="Drv"      ;alignment="left"  ;width=3
;Expression={$_.DeviceID};};
$fmtName  =@{label="Vol Name" ;alignment="left"  ;width=15
;Expression={$_.VolumeName};};
$fmtSize  =@{label="Size MB"  ;alignment="right" ;width=12
;Expression={$_.Size / 1048576};; FormatString="N0";};
$fmtFree  =@{label="Free MB"  ;alignment="right" ;width=12
;Expression={$_.FreeSpace / 1048576}    ;
FormatString="N0";};
$fmtPerc  =@{label="Free %"   ;alignment="right" ;width=10
;Expression={100.0 * $_.FreeSpace / $_.Size};
FormatString="N1";};
$fmtMsg   =@{label="Message"  ;alignment="left"  ;width=12 ;
`
              Expression={     if (100.0 * $_.FreeSpace /
$_.Size -le $levelAlarm) {"Alarm !!!"} `
                          elseif (100.0 * $_.FreeSpace /
$_.Size -le $levelWarn)  {"Warning !"} };};

foreach($server in $servers)
{
    $disks = Get-WmiObject -ComputerName $server -Class
Win32_LogicalDisk -Filter "DriveType = 3";

    Write-Output ("Windows Machine Name: {0}`tDrives #: {1}"
-f $server, $disks.Count);
    Write-Output $disks | Format-Table $fmtDrive, $fmtName,
$fmtSize, $fmtFree, $fmtPerc, $fmtMsg;
}
```

`ListServerDrivesFreespace.ps1` PowerShell Script in PowerShell ISE


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1
ListServerDrivesFreespace.ps1

```
<#
 .SYNOPSIS
    List for several machines the drives with size, free size and the p
 .DESCRIPTION
    An important duty of a DBA is to check frequently the free space of
    With this PowerShell script you can easily check all drives for al
    Requires permission to connect to and fetch WMI data from the mach
 .NOTES
    Author   : Olaf Helper
    Requires: PowerShell Version 1.0
 .LINK
    TechNet Get-WmiObject
        http://technet.microsoft.com/en-us/library/dd315295.aspx
#>

# Configuration data.
# Add your machine names to check for to the list:
```

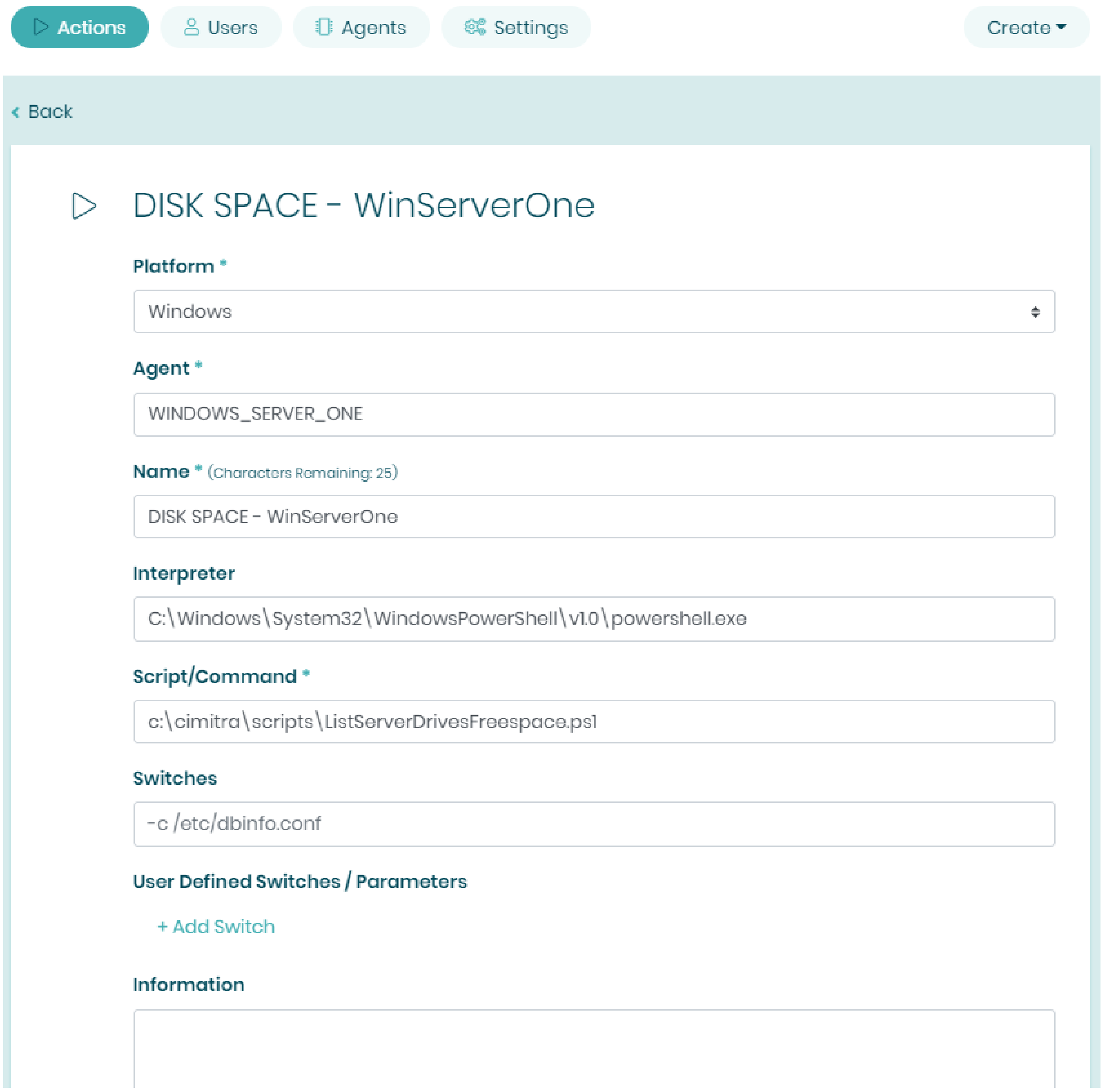
```
PS C:\WINDOWS\system32> cd c:\cimitra\scripts

PS C:\cimitra\scripts> .\ListServerDrivesFreespace.ps1
Windows Machine Name: TAYPC Drives #: 2

Drv Vol Name              Size MB      Free MB     Free % Message
--- --------              -------      -------     ------ -------
C:  SSD                   500,171      260,034       52.0
D:  HDD                 1,907,712    1,710,147       89.6



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 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	**CHECK DISK SPACE**
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	**C:\cimitra\scripts\ListServerDrivesFreespace.ps1**



DISK SPACE - WinServerOne

Platform *
Windows

Agent *
WINDOWS_SERVER_ONE

Name * (Characters Remaining: 25)
DISK SPACE - WinServerOne

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

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

Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters
+ Add Switch

Information

Restarting Windows Local Printer Services

Use Case

A Windows server or Windows 10 workstation is hosting printing services for a local printer. Often restarting the Print Spooler resolves printing issues. With Cimitra the IT admin has given access to restart the Print Spooler to persons who sit physically close to the printer and know the telltale signs that point to the Windows Print Spooler needing a restart.

Technical Overview

This a very simple command, that often fixes printing problems. You can find more complex scripts on the web to cover more conditions. This is simply one example that works well.

Script Contents - Copy Paste

```
# Cimitra Print Spooler Restart Script
# Version 1.0
# Release Date: 9/18/2020
# Author: Tay Kratzer tay@cimitra.com
# ---------------------------------------

### MODIFY THESE THREE VALUES TO USE SCRIPT FOR OTHER
WINDOWS SERVICES ###

###VARIABLES BEGIN###

$PROCESS_IN = "spoolsv"

$SERVICE_NAME = "Spooler"

$SERVICE_FRIENDLY_NAME = "Print Server"

###VARIABLES END###

# Make a temporary file
$TEMP_FILE=New-TemporaryFile

# Find the process spool service (spoolsv) and format output
as a list (fl)
```

```
Get-Process ${PROCESS_IN}* | fl 1> $TEMP_FILE

# Get just the line that has the word "Id" in it
$PID_LINE = Select-String -Path $TEMP_FILE -Pattern "Id"

# Remove the Temp file
Remove-Item -Path $TEMP_FILE -Force

# Remove all spaces in the line (easier for parsing
accurately with the next command)
$PID_LINE = $PID_LINE -replace " ", ""


# Get the first set column of data after the text "Id:",
which is the process PID
$PROCESS_PID = ($PID_LINE -split "Id:")[1]
Write-Output ""
Write-Output "=============================================="
Write-Output "Current ${SERVICE_FRIENDLY_NAME} Process ID:
[ $PROCESS_PID ]"
Write-Output "=============================================="
Write-Output "Restarting $SERVICE_FRIENDLY_NAME"

# Restart the Print Spooler Process
Restart-Service -Name ${SERVICE_NAME} -Force

# Get the PID again to confirm help to visually confirm that
the process was restarted

# Make a temporary file
$TEMP_FILE=New-TemporaryFile
```

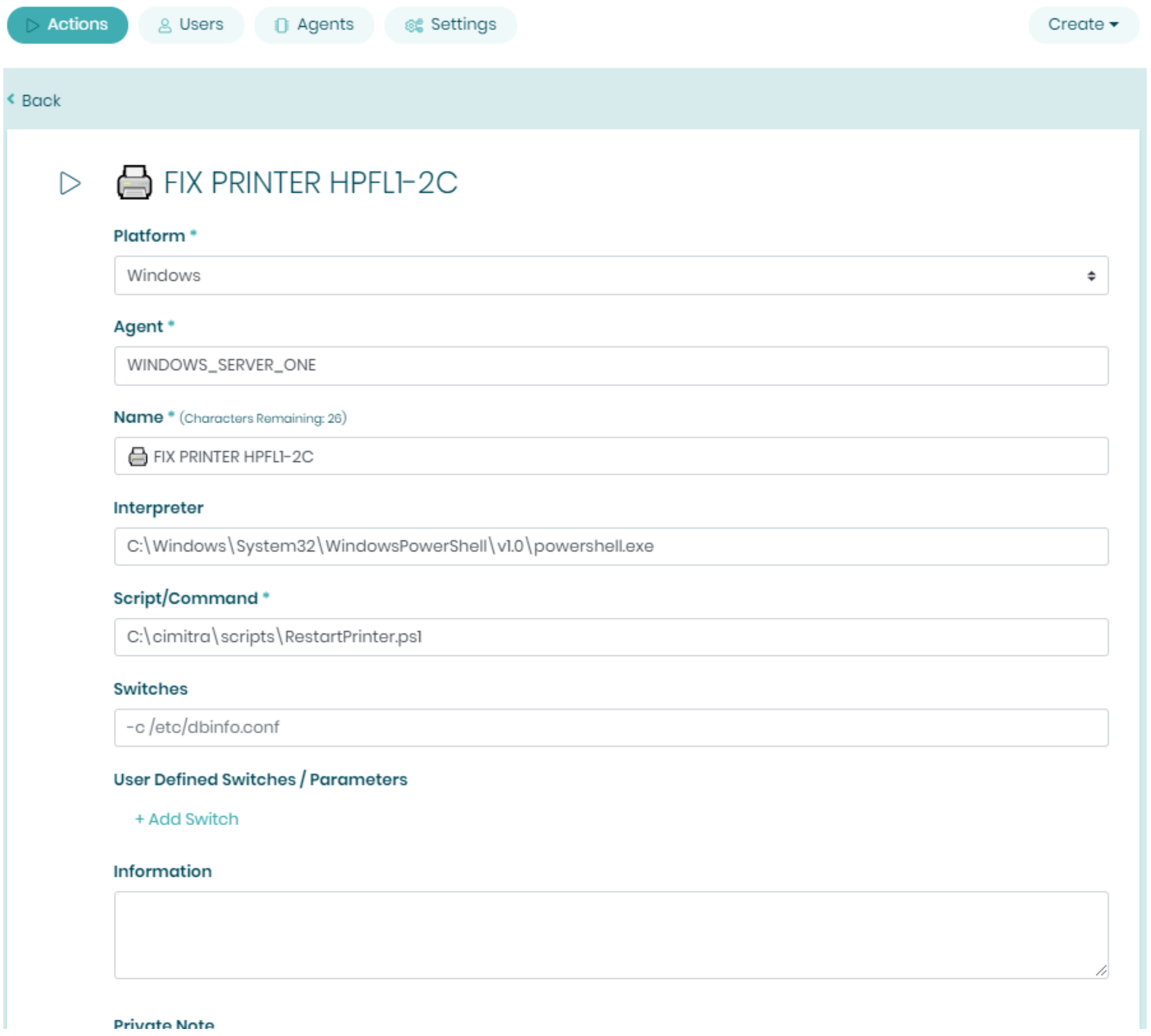
```
# Find the process spool service (spoolsv) and format output
as a list (fl)
Get-Process ${PROCESS_IN}* | fl 1> $TEMP_FILE

# Get just the line that has the word "Id" in it
$PID_LINE = Select-String -Path $TEMP_FILE -Pattern "Id"

# Remove the Temp file
Remove-Item -Path $TEMP_FILE -Force

# Remove all spaces in the line (easier for parsing
accurately with the next comand)
$PID_LINE = $PID_LINE -replace " ", ""


# Get the first set column of data after the text "Id:",
which is the process PID
$PROCESS_PID = ($PID_LINE -split "Id:")[1]

Write-Output "============================================="
Write-Output "Restarted ${SERVICE_FRIENDLY_NAME} Process ID:
[ $PROCESS_PID ]"
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 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	🖨 **FIX PRINTER HPFL1-2C** NOTE: I was able to grab emoticons from the web to represent the printer object. Emoticons may show up differently on different browsers, so be aware of this potential variance.
Interpreter	<Path to PowerShell Interpreter> Example: **C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe**
Script/Command	**C:\cimitra\scripts\RestartPrinter.ps1**



FIX PRINTER HPFL1-2C

Platform *
Windows

Agent *
WINDOWS_SERVER_ONE

Name * (Characters Remaining: 26)
FIX PRINTER HPFL1-2C

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

Script/Command *
C:\cimitra\scripts\RestartPrinter.ps1

Switches
-c /etc/dbinfo.conf

User Defined Switches / Parameters
+ Add Switch

Information

Private Note

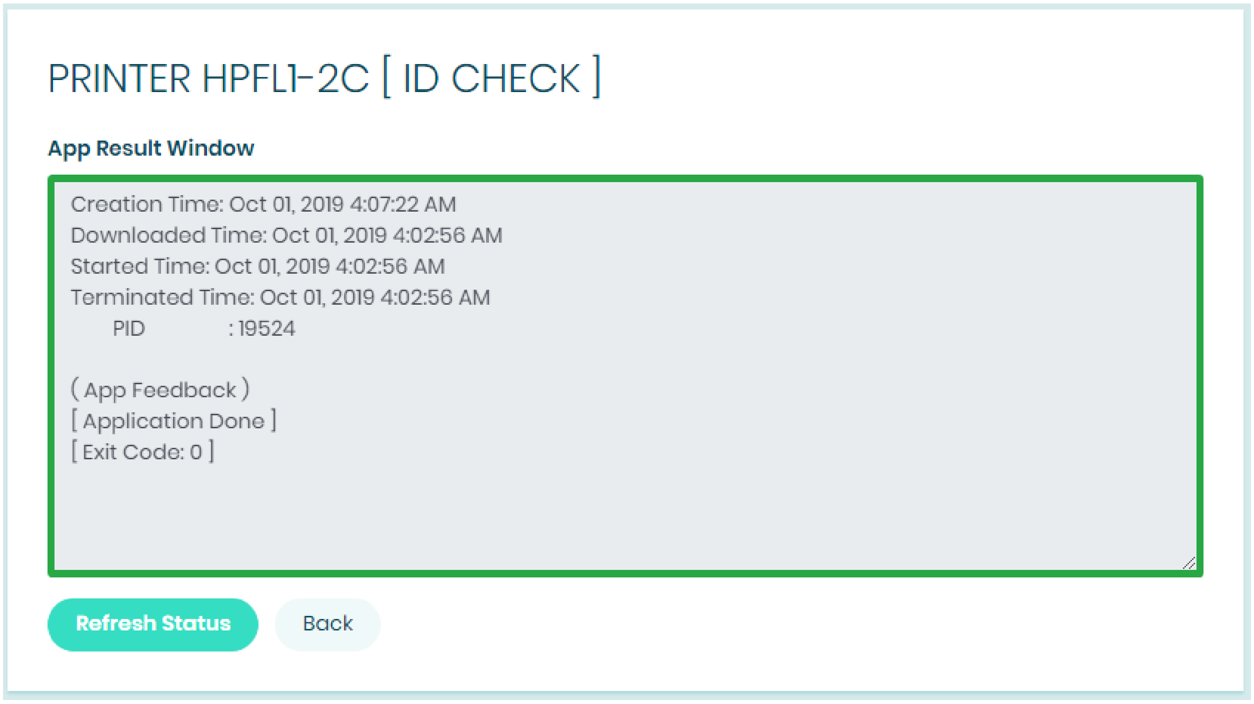
Assuming the Cimitra Agent and the PowerShell script are all in place, the Cimitra App should now be usable and shareable with others. A Cimitra App needs to be in a Cimitra Folder in order to share the Cimitra App.

Checking a Printer Spooler's PID

NOTE: Sometimes there is no reason to create a PowerShell script file for integration into Cimitra, particularly if there is no need for user input. This section will show how to use a Windows operating system command in-line in a Cimitra App, without any need to establish a PowerShell script of any sort.

Use Case

Since there is a way to restart the Print Spooler, it's also nice to see that when the Print Spooler restarted that there is a new process id (PID) associated with the Print Spooler. This is simply a sure indicator that the Print Spooler restart process happened.



```
Creation Time: Oct 01, 2019 4:07:22 AM
Downloaded Time: Oct 01, 2019 4:02:56 AM
Started Time: Oct 01, 2019 4:02:56 AM
Terminated Time: Oct 01, 2019 4:02:56 AM
      PID            : 19524

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

Technical Overview

This is a very simple operating system command (DOS command). We could have put the command into a batch file, but we didn't just to emphasize that often just a command sequence is all you need to create a powerful App in Cimitra.

Operating System Command - Copy Paste

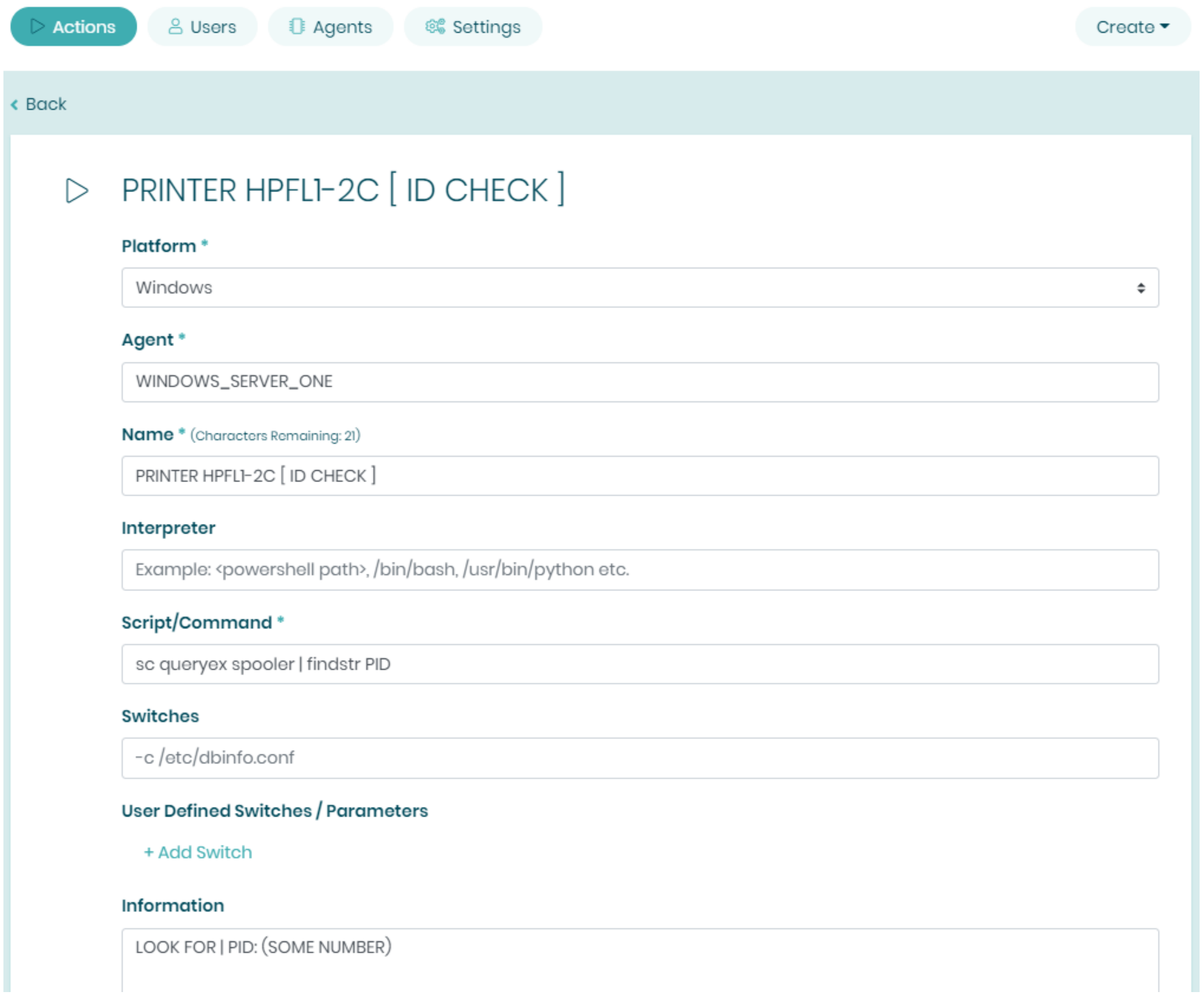
```
sc queryex spooler | findstr PID
```

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 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	**PRINTER HPFL1-2C [ID CHECK]**
Interpreter	<LEAVE BLANK>
Script/Command	`sc queryex spooler | findstr PID`
Information	**LOOK FOR | PID: (SOME NUMBER)**



PRINTER HPFL1-2C [ID CHECK]

Platform *

Windows

Agent *

WINDOWS_SERVER_ONE

Name * (Characters Remaining: 21)

PRINTER HPFL1-2C [ID CHECK]

Interpreter

Example: <powershell path>, /bin/bash, /usr/bin/python etc.

Script/Command *

```
sc queryex spooler | findstr PID
```

Switches

-c /etc/dbinfo.conf

User Defined Switches / Parameters

Information

LOOK FOR | PID: (SOME NUMBER)

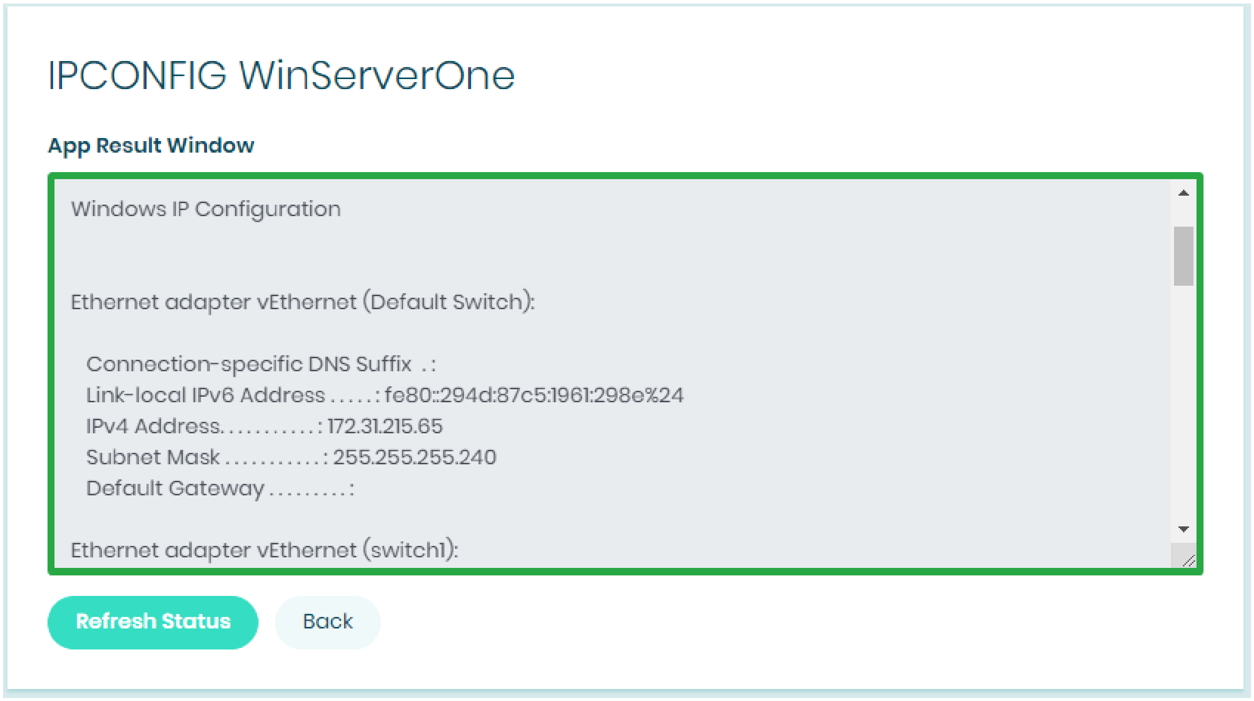
Assuming the Cimitra Agent is in place, the Cimitra App should now be usable and shareable with others. A Cimitra App needs to be in a Cimitra Folder in order to share the Cimitra App.

Checking a Server's IP Configuration

NOTE: Sometimes there is no reason to create a PowerShell script file for integration into Cimitra, particularly if there is no need for user input. This section will show how to use a Windows operating system command in-line in a Cimitra App, without any need to establish a PowerShell script of any sort.

Use Case

Very often someone needs to know the particular IP address of a server. This might not be something that you need to share with other people per-say, but the **`ipconfig`** command is something you might find yourself doing often.



IPCONFIG WinServerOne

App Result Window

```
Windows IP Configuration


Ethernet adapter vEthernet (Default Switch):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::294d:87c5:1961:298e%24
   IPv4 Address. . . . . . . . . . . : 172.31.215.65
   Subnet Mask . . . . . . . . . . . : 255.255.255.240
   Default Gateway . . . . . . . . . :

Ethernet adapter vEthernet (switch1):
```

Technical Overview

This is a very simple operating system command (DOS command). We could have put the command into a batch file, but we didn't just to emphasize that often just a command sequence is all you need

Operating System Command - Copy Paste

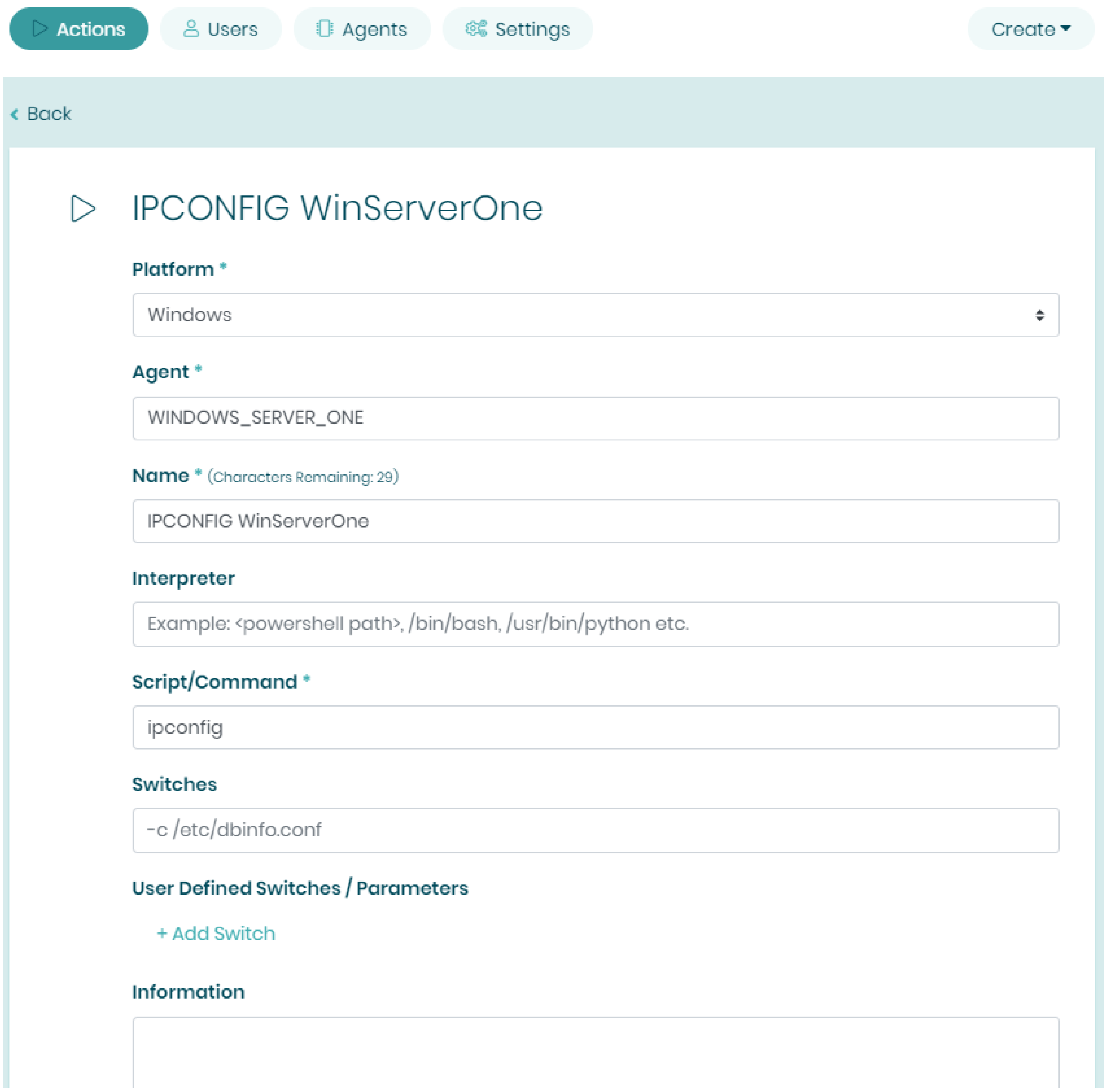
```
ipconfig
```

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 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	**IPCONFIG <Server Name>**
Interpreter	<LEAVE BLANK>
Script/Command	`ipconfig`



IPCONFIG WinServerOne

Platform *

Windows

Agent *

WINDOWS_SERVER_ONE

Name * (Characters Remaining: 29)

IPCONFIG WinServerOne

Interpreter

Example: <powershell path>, /bin/bash, /usr/bin/python etc.

Script/Command *

ipconfig

Switches

-c /etc/dbinfo.conf

User Defined Switches / Parameters

+ Add Switch

Information

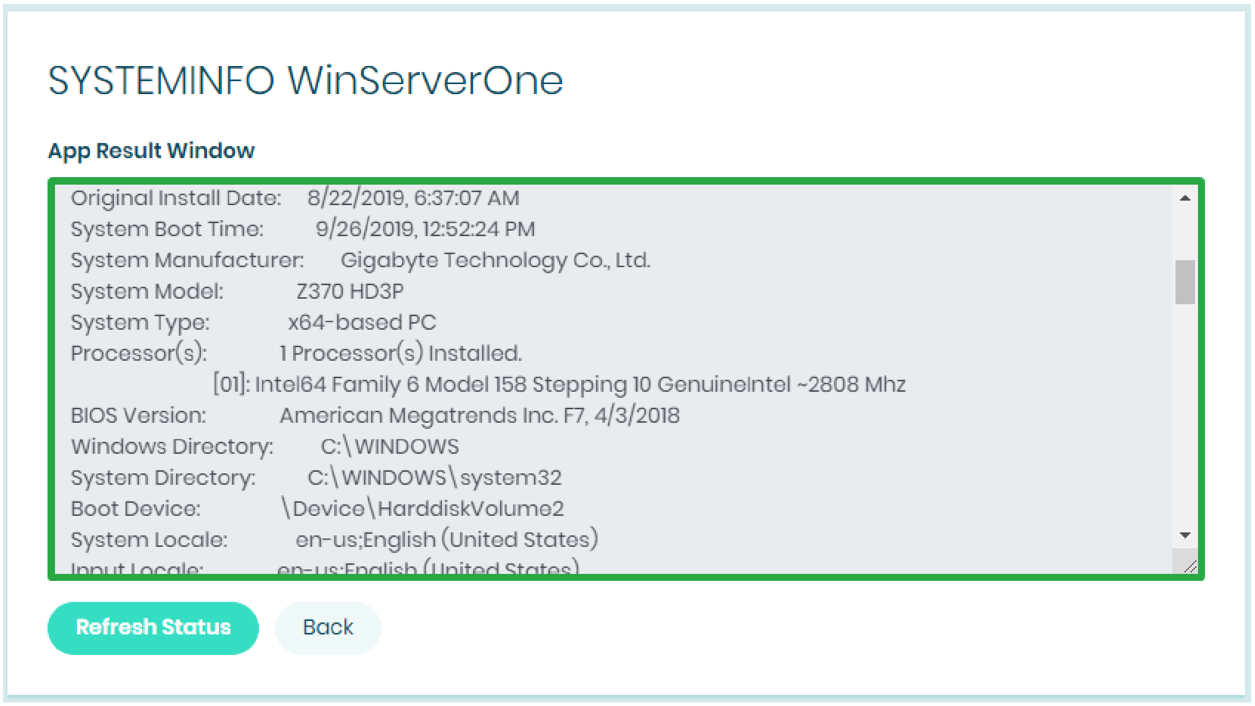
Assuming the Cimitra Agent is in place, the Cimitra App should now be usable and shareable with others. A Cimitra App needs to be in a Cimitra Folder in order to share the Cimitra App.

Checking Windows System Information

NOTE: Sometimes there is no reason to create a PowerShell script file for integration into Cimitra, particularly if there is no need for user input. This section will show how to use a Windows operating system command in-line in a Cimitra App, without any need to establish a PowerShell script of any sort.

Use Case

The Windows systeminfo command has all kinds of goodies in it. This might not information that you share with other people, but it's very helpful information to have at hand.


SYSTEMINFO WinServerOne

App Result Window

```
Original Install Date:     8/22/2019, 6:37:07 AM
System Boot Time:          9/26/2019, 12:52:24 PM
System Manufacturer:       Gigabyte Technology Co., Ltd.
System Model:              Z370 HD3P
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 158 Stepping 10 GenuineIntel ~2808 Mhz
BIOS Version:              American Megatrends Inc. F7, 4/3/2018
Windows Directory:         C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume2
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
```

Technical Overview

This is a very simple operating system command (DOS command). We could have put the command into a batch file, but we didn't just to emphasize that often just a command sequence is all you need

Operating System Command - Copy Paste

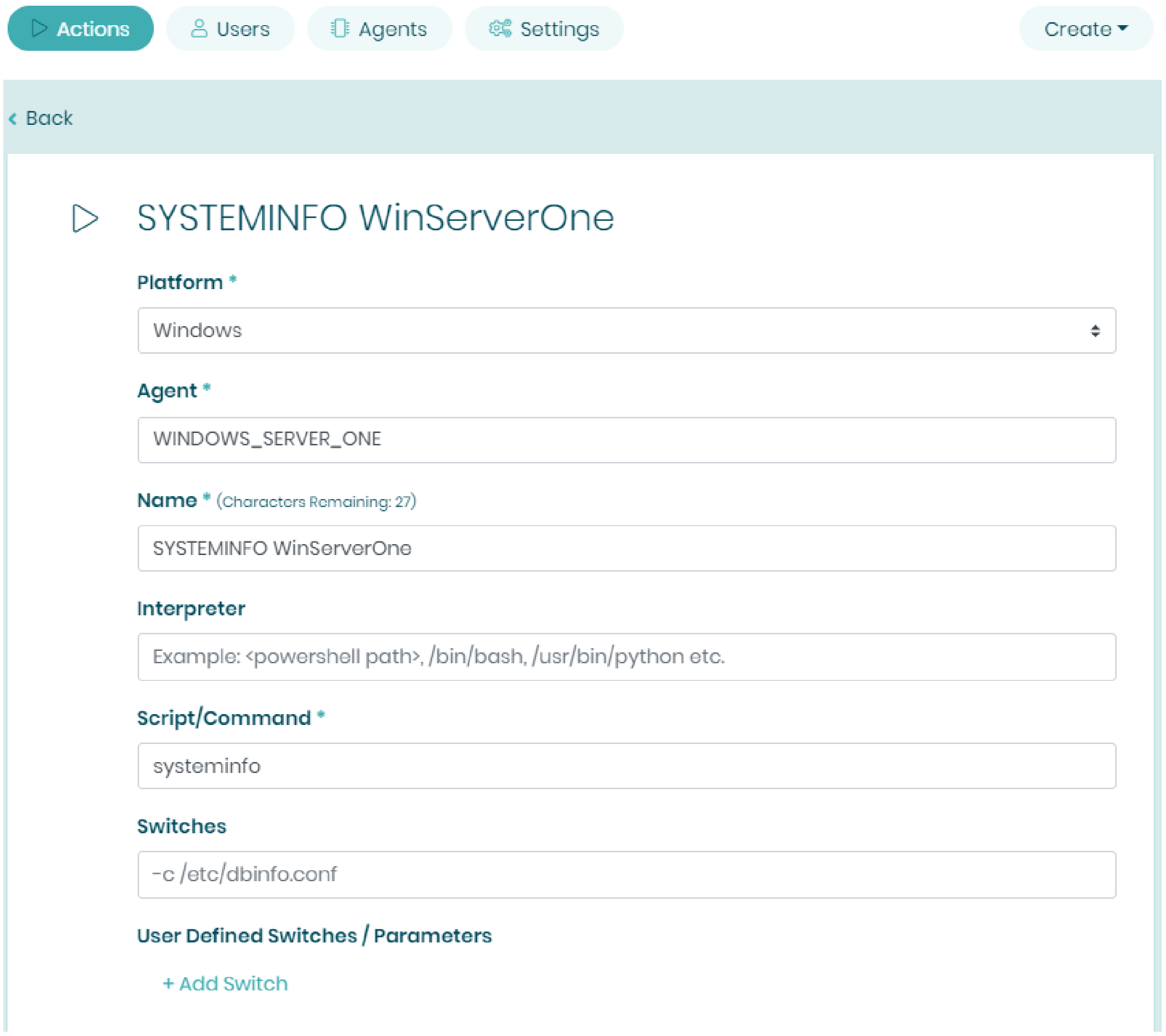
```
systeminfo
```

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 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	**SYSTEMINFO <Server Name>**
Interpreter	<LEAVE BLANK>
Script/Command	`systeminfo`



SYSTEMINFO WinServerOne

Platform *	Windows
Agent *	WINDOWS_SERVER_ONE
Name * (Characters Remaining: 27)	SYSTEMINFO WinServerOne
Interpreter	Example: <powershell path>, /bin/bash, /usr/bin/python etc.
Script/Command *	systeminfo
Switches	-c /etc/dbinfo.conf

User Defined Switches / Parameters

+ Add Switch

Assuming the Cimitra Agent is 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.